



RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université Abderrahmene Mira de Béjaïa

Faculté des Sciences exactes

Département d'Informatique

Ecole Doctorale d'Informatique

Mémoire de Magistère En Informatique

Option

Réseaux et Systèmes Distribués

Thème

CONCEPTION ET MODÉLISATION D'UN PROTOCOLE DE ROUTAGE GÉOGRAPHIQUE POUR LES RÉSEAUX DE CAPTEURS SANS FILS

Présenté Par

Mehdi BOULAICHE

Soutenu Devant le jury :

Président	Djamil	AÏSSANI	Professeur	Université de Béjaïa.
Rapporteur	Louiza	BOUALLOUCHE-MEDJKOUNE	MCA	Université de Béjaïa.
Examineur	Mohammed	BENMOHAMMED	Professeur	Université de Constantine.
Examineur	Abdellah	BOUKERRAM	MCA	Université de Sétif.
Invitée	Samra	BOULFEKHAR	MAA	Université de Béjaïa.

Promotion 2009/2010

A ma mère

A mon père

A toute ma famille

Table des Matières

Table des matières	iii
Liste des Tableaux	vii
Table des Figures	viii
Remerciements	ix
Résumé	x
Abstract	xi
	xii
Introduction générale	1
1 Introduction aux réseaux de capteurs sans fil	4
1.1 Introduction	4
1.2 Capture et capteurs	5
1.2.1 Définition	5
1.2.2 Architecture matériel d'un capteur	5
1.2.2.1 L'unité de capture	5
1.2.2.2 L'unité de traitement	5
1.2.2.3 L'unité de transmission/réception	6
1.2.2.4 L'unité de contrôle d'énergie	6
1.2.2.5 Système de localisation	6
1.2.2.6 Mobilisateur	6
1.2.2.7 Générateur d'énergie	7
1.2.3 Systèmes d'exploitation	7
1.2.3.1 TinyOS	7
1.2.3.2 SOS	7
1.2.3.3 Contiki	8
1.2.3.4 LiteOS	8
1.3 Réseaux de Capteurs Sans Fil (RCSF)	8
1.3.1 Architecture des RCSFs	9
1.3.2 Pile protocolaire	10
1.3.2.1 Les Couches	10

1.3.2.2	Plans de gestion	14
1.3.3	Caractéristiques des RCSF	15
1.4	Applications des RCSFs	17
1.4.1	Applications militaires	17
1.4.2	Applications environnementales	17
1.4.3	Applications dans le domaine de la santé	17
1.4.4	Applications domotiques	18
1.4.5	Applications industrielles	18
1.5	Contraintes de conceptions des RCSFs	18
1.5.1	Contraintes matérielles	19
1.5.2	La tolérance aux pannes	19
1.5.3	Scalabilité	19
1.5.4	Coût de production	20
1.5.5	Support de transmission	20
1.5.6	Consommation d'énergie	20
1.6	Les grands défis	22
1.7	Conclusion	23
2	Le routage dans les RCSFs	25
2.1	Introduction	25
2.2	Considérations de conception d'un protocole de routage dans les RCSFs	26
2.2.1	Consommation d'énergie	26
2.2.2	Scalabilité	26
2.2.3	Adressage	27
2.2.4	Robustesse	27
2.2.5	Topologie	27
2.2.6	Application	27
2.2.7	Hétérogénéité des nœuds/liens	28
2.2.8	Média de transmission	28
2.2.9	Connectivité	28
2.2.10	Agrégation de données	28
2.2.11	Qualité de service	29
2.3	Métriques de routage	29
2.3.1	Nombre de sauts minimum	29
2.3.2	Energie	29
2.3.3	Qualité de service	30
2.3.4	Robustesse	30
2.4	Classification des protocoles de routage dans les RCSFs	30
2.4.1	Classification selon l'organisation du réseau	31
2.4.1.1	Les protocoles à plat	31
2.4.1.2	Protocoles hiérarchiques	33
2.4.1.3	Protocoles basés sur la localisation	35
2.4.2	Classification selon l'établissement des chemins	36
2.4.2.1	Protocoles de routage proactifs	36
2.4.2.2	Protocoles de routage réactifs	37

2.4.2.3	Protocoles de routage hybrides	37
2.4.3	Classification selon le fonctionnement du protocole	38
2.4.3.1	Protocoles basés sur la QoS	38
2.4.3.2	Protocoles basés sur la négociation	39
2.4.3.3	Protocoles basés sur le multi-chemins	39
2.4.3.4	Protocoles basés sur la cohérence	40
2.4.3.5	Protocoles basés sur la requête	40
2.5	Conclusion	41
3	Les protocoles de routage géographique dans les RCSFs	42
3.1	Introduction	42
3.2	La localisation	43
3.2.1	Les méthodes basées mesure	43
3.2.1.1	Technologies de mesure	43
3.2.1.2	Méthodes de localisation	44
3.2.2	Les méthodes libres de mesure	45
3.3	Protocoles basés sur la position physique	45
3.3.1	Les algorithmes greedy	45
3.3.1.1	MFR (Most Forwarding with Radius)	45
3.3.1.2	Greedy Routing	46
3.3.1.3	Compass Routing	46
3.3.1.4	NFP (Nearest Forward Progress)	46
3.3.1.5	GEDIR	46
3.3.2	Routage basé sur graphe planaire	47
3.3.2.1	Construction d'un graphe planaire	48
3.3.2.2	Routage par faces (Face routing)	49
3.3.2.3	GFG (Greedy-Face-Greedy)	50
3.3.2.4	Power Progress	50
3.3.2.5	GPSR	50
3.3.2.6	GOAFR	51
3.3.3	Algorithmes hiérarchiques	54
3.3.3.1	GRID	54
3.3.3.2	Terminode	55
3.3.4	Algorithmes basés sur l'inondation dirigée	56
3.3.4.1	LAR	56
3.3.4.2	DREAM	57
3.4	Protocoles basés sur des coordonnées virtuelles	59
3.4.1	Routage greedy basé sur des coordonnées virtuelles	59
3.4.1.1	VCR (Virtual Coordinates based Routing)	59
3.4.1.2	ViP (Virtual Position)	61
3.4.1.3	VCP (Virtual Cord Protocol)	62
3.4.1.4	Aligned VCS (Aligned Virtual Coordinates System)	63
3.4.2	Algorithmes utilisant des ancrs pour construire des coordonnées virtuelles	64
3.4.2.1	VCap	64
3.4.2.2	ABVCap	66

3.4.2.3	VCost	67
3.4.2.4	GLIDER	68
3.5	Comparaison des protocoles de routage géographique dans les RCSFs	71
3.6	Conclusion	73
4	Nouveau protocole de routage géographique	74
4.1	Introduction	74
4.2	Objectifs	75
4.3	Modèle du réseau	75
4.4	Fonctionnement du protocole	76
4.4.1	Phase de communication de données	76
4.4.1.1	Construction du sous graphe G'	77
4.4.1.2	Calcul du plus court chemin	78
4.4.2	Phase de maintenance	80
4.5	Evaluation des performances de EGGR	81
4.5.1	Choix de l'environnement de simulation	83
4.5.2	Paramètres de simulation	83
4.5.3	Les métriques d'évaluation	84
4.5.3.1	Consommation d'énergie	85
4.5.3.2	Taux de succès	85
4.5.3.3	Durée de vie du réseau	85
4.5.3.4	La latence moyenne	85
4.5.4	Discussion des résultats	86
4.6	Conclusion	91
	Conclusion générale et Perspectives	93
	Bibliographie	95

Liste des tableaux

3.1	Comparaison entre les différents protocoles étudiés dans ce chapitre.	72
4.1	Format de la table des voisins.	77
4.2	Paramètres de simulation.	84

Table des figures

1.1	Architecture hardware générale d'un nœud capteur.	6
1.2	Un réseau de capteurs sans fil.	9
1.3	Nœuds capteurs dispersés dans un champ	10
1.4	Pile protocolaire d'un réseau de capteurs sans fil.	11
1.5	La consommation d'énergie d'un nœud capteur MicaZ.	21
2.1	Catégories des protocoles de routage	31
2.2	(a) Propagation de l'intérêt, (b) Etablissement du gradient, (c) Envoi des données.	33
2.3	Clusters avec un seul saut à la station de base (gauche) et clusters avec une connexion à la station de base multisauts (droit).	34
3.1	Les différentes méthodes de routage Greedy.	47
3.2	Graphe planaire.	48
3.3	Construction du graphe planaire.	49
3.4	Routage par face, les changements de faces surviennent dans les nœuds u , v , w	49
3.5	Exemple d'un trou.	51
3.6	Other face routing.	52
3.7	GOAFR.	54
3.8	Comparaison des deux schémas de LAR.	58
3.9	Exemple d'expected zone dans DREAM.	58
3.10	Obtention des coordonnées virtuelles.	60
3.11	Routage a travers les tiles.	70
4.1	Exemple de Fonctionnement de EGGR.	81
4.2	Moyenne d'énergie consommée par rapport au nombre de noeuds dans le réseau.	86
4.3	Moyenne d'énergie consommée dans le temps.	87
4.4	Taux de succès.	88
4.5	Durée de vie du réseau : durée de vie du dernier nœud qui meurt.	89
4.6	Durée de vie du réseau : durée de vie du premier nœud qui meurt.	90
4.7	Durée de vie du réseau : Nombre de nœuds en vie.	90
4.8	La latence moyenne.	91

Remerciements

Je tiens à remercier dans un premier temps le Dr. Louiza BOUALLOUCHE-MEDJKOUNE de l'université de bejaia (Algérie) pour m'avoir encadré tout au long d'année pratique de mon magistère, pour son aide et ses orientations.

Je tiens ensuite à remercier Melle Samra BOULFEKHAR pour ses précieux conseils qui m'ont été d'une grande aide.

J'adresse aussi mes sincères remerciements aux Pr. Djamil AÏSSANI, Pr. Mohamed BENMOHAMMED, Dr. Abdellah BOUKERRAM, pour l'intérêt qu'ils ont bien voulu porter à ce travail en acceptant de l'évaluer.

Je remercie tous les enseignants de l'école doctorale ReSyD, et tous les responsables qui ont participé dans la bonne démarche de nos études durant ces deux années de magistère.

Je remercie tous les étudiants de l'école doctorale ReSyD, pour l'environnement de travail très agréable durant ces deux dernières années de magistère.

Je remercie toutes les personnes qui ont lu et relu ce mémoire et en particulier, mesdames et messieurs les rapporteurs.

Enfin, et surtout, je remercie vivement toute ma famille qui m'a toujours supporté moralement et financièrement pendant toutes mes longues années d'étudiant.

Mehdi BOULAICHE.

Résumé

LES avancées récentes dans la technologie micro-électro-mécanique, les communications sans-fil et la micro-électronique, ont rendu possible le développement de capteurs multifonctionnels avec des coûts réduits capables de détecter, mesurer et rapporter des données physiques liées à l'environnement. Les réseaux de capteurs sont tout simplement un ensemble de ces capteurs, formant un réseau sans infrastructure. La petite taille de ces capteurs impose des limites sur leur capacité en termes d'énergie embarquée car ils sont généralement alimentés par des batteries non rechargeables et difficilement remplaçables. Et parce que la grande partie de l'énergie est consommée par le sous système radio (la communication), plusieurs solutions qui traitent tous les niveaux de la hiérarchie du réseau ont été proposées pour réduire la consommation d'énergie et étendre la durée de vie du réseau.

Dans ce travail, nous avons proposé un nouveau protocole de routage géographique qui conserve l'énergie des nœuds et qui garantit la livraison des paquets dans les réseaux de capteurs sans fil tout en se basant uniquement sur les connaissances locales des nœuds capteurs (le voisinage). Le protocole proposé *EGGR (Energy-aware and delivery Guarantee Geographic Routing)* permet une gestion plus efficace de la ressource énergétique lors de la communication des données dans le réseau, à travers l'introduction d'un mécanisme pour calculer le plus court chemin en termes de consommation d'énergie pendant le calcul du prochain nœud. La garantie de livraison est assurée par l'introduction d'un mécanisme permettant d'éviter les trous.

L'analyse de performances du protocole proposé EGGR montre qu'il réduit la consommation d'énergie et augmente le taux de livraison des paquets entre les nœuds capteurs et la station de base par rapport au protocole GPSR.

Mots clés : Réseaux de capteurs sans fil, Routage géographique, Localisation, Consommation d'énergie.

Abstract

RECENT advances in micro-electro-mechanic technology, wireless communication technologies, and in micro-electronic have enabled the development of multifunctional sensors at low cost capable of detecting, measuring, and reporting information about the environment . Wireless sensor networks are simply a set of these sensors, shaping a wireless network. The small size of the sensors confines the embedded energy because they are powered by non-rechargeable batteries, and even not easily replaceable. And because the most significant amount of energy is consumed by the sub-system radio (communication), several solutions, that address all network hierarchy layers, were proposed to reduce energy consumption and extend network lifetime.

The aim of this work is to propose a new geographic routing protocol that conserves nodes energy and that guarantees packets delivery in the wireless sensor network based only on local knowledge of sensor nodes (neighborhood). The proposed protocol EGGR (Energy-aware and delivery Guarantee Geographic Routing) manage energy resources efficiently during data communication in the network, by introducing a mechanism that calculates the shortest path in terms of energy consumption when setting up the next node. Delivery guaranty is assured by introducing a mechanism that avoids holes.

Our experimental results show that the proposed protocol EGGR decreases energy consumption and increases delivery rate between sensor nodes and the sink compared with GPSR protocol.

Keywords : Wireless sensor network, Routing, Geographic routing, Localization, Energy consumption.

Introduction générale

LES récentes avancées technologiques dans le domaine des télécommunications sans fils ont permis le développement à faible coût de minuscules systèmes micro-électromécaniques appelés " *capteurs* ", capables de détecter, mesurer et rapporter des données physiques liées à l'environnement. Les réseaux de capteurs sont tout simplement un ensemble de ces capteurs, formant un réseau sans infrastructure. Ainsi, dans ces réseaux, chaque nœud est capable de détecter son environnement et de traiter l'information au niveau local ou de l'envoyer à un ou plusieurs points de collecte. Leurs applications potentielles diversifiées sont la source de l'intérêt qu'ils suscitent aussi bien dans la communauté scientifique qu'industrielle. Néanmoins, plusieurs problèmes restent encore à traiter et plusieurs solutions proposées dans la recherche restent encore à améliorer.

Une problématique majeure dans les réseaux de capteurs est la maîtrise de l'énergie consommée par chaque nœud capteur. En effet, chaque nœud est alimenté par une batterie dont la capacité est limitée et qui n'est pas rechargeable. Aussi, pour qu'un réseau de capteurs ait une longévité maximale, il faut que la consommation d'énergie soit prise en compte à tous les niveaux de l'architecture réseau (physique, MAC, et réseau). Ainsi, les protocoles mis en œuvre doivent en particulier intégrer cette contrainte.

La plupart des protocoles de routage dans les réseaux de capteurs nécessitent la localisation des nœuds capteurs. En général, ces informations sont nécessaires pour calculer la distance entre deux nœuds particuliers de sorte que la consommation d'énergie puisse être estimée. Puisque il n'y a aucun système d'adressage pour les nœuds dans les réseaux de capteurs (comme les adresses IP) et comme ils sont déployés dans une région d'une manière aléatoire, l'information de localisation de ces nœuds peut être utilisée dans le routage des données d'une manière efficace en termes d'énergie. Par exemple, si la région à sentir est connue en utilisant la localisation des nœuds capteurs pour diffuser la requête seulement à cette région particulière, alors il est facile d'éliminer, de manière significative, le nombre de transmissions.

Notre travail s'intéresse particulièrement à la conception d'un nouveau protocole de routage géographique qui répond aux deux problèmes majeurs du routage géographique : la consommation d'énergie et la garantie de livraison des données vers la station de base.

De ce fait, nous allons, dans un premier temps, réaliser une étude critique des différents protocoles antérieurs qui traitent ces deux problèmes. Puis, nous allons concevoir un nouveau protocole de routage géographique qui minimise la consommation d'énergie et qui garantit la livraison des paquets entre la source et la destination en se basant uniquement sur les connaissances locales des nœuds voisins.

Organisation du mémoire

Ce mémoire se compose de quatre chapitres :

- Le premier chapitre présente des généralités sur les réseaux de capteurs sans fil avec une description de leurs architectures et leurs caractéristiques principales ainsi que leurs domaines d'application. Nous discutons par la suite les facteurs et les contraintes influant sur leurs conceptions, et nous exposons enfin les grands défis de ces réseaux.
- Dans le deuxième chapitre, nous présentons tout d'abord, les différentes considérations qui doivent être prises en compte lors de la conception d'un nouveau protocole de routage pour les réseaux de capteurs sans fil, ainsi que les métriques généralement utilisés. La deuxième partie du chapitre, quant à elle, est consacrée au concept du routage dans les réseaux de capteurs, où nous exposons les classifications existantes pour les protocoles de routage proposés pour ces réseaux, en citant quelques exemples et en apportant une étude critique à chaque classe de routage.
- Après l'introduction de la classe du routage géographique dans le chapitre précédent, nous consacrons le troisième chapitre à l'étude détaillée de cette approche. Nous exposons en premier lieu, les différentes méthodes de localisation dans les réseaux de capteurs sans fil. Puis, nous présentons un ensemble de protocoles pour les deux catégories existantes (protocoles qui utilisent des coordonnées physiques et ceux qui utilisent des coordonnées virtuelles). A la fin de ce chapitre, une étude comparative entre les différentes caractéristiques des protocoles étudiés est présentée.
- Le quatrième chapitre est consacré à la description détaillée du protocole de routage "*EGGR*" (*Energy-aware and delivery Guarantee Geographic Routing*) proposé comme solution à la problématique du routage géographique dans les réseaux de capteurs sans fil (la consommation d'énergie et la garantie de livraison), tout en détaillant son principe de fonctionnement, puis analysant ses performances en le comparant avec le protocole GPSR.

- Le travail s’achève par une conclusion mettant l’accent sur les perspectives et les directions de recherche induites par les résultats obtenus.

Chapitre 1

Introduction aux réseaux de capteurs sans fil

1.1 Introduction

De nombreuses avancées techniques et technologiques dans les domaines de la micro-électronique, de la micro mécanique, et des technologies de communication sans fil permettent de créer de petits objets communicants équipés de capteurs à un coût raisonnable. Ces nouveaux objets appelés nœuds ou capteurs sont équipés d'une unité de capture, d'une unité de traitement, de mémoires et d'une radio pour communiquer. Enfin, pour l'alimentation, ces nœuds possèdent une pile ou un système de récupération d'énergie dans l'environnement.

Cette technologie rend possible le déploiement de réseaux de capteurs sans fil. Les réseaux de capteurs ont de nombreuses perspectives d'application dans des domaines très variés : applications militaires, domotique, surveillance industrielle ou de phénomènes naturels, et relevé de compteurs. Chaque application a ses propres contraintes. Dans tous les domaines, le rôle d'un réseau de capteurs est cependant à peu près toujours le même. Les nœuds doivent surveiller certains phénomènes grâce à leurs capteurs puis envoient les informations à un puits. Le puits (ou sink en anglais) est un nœud particulier doté d'une puissance de calcul supérieure et d'une quantité d'énergie potentiellement infinie.

Dans ce chapitre, nous allons présenter un ensemble de généralités sur les réseaux de capteurs sans fil, leurs architectures, leurs caractéristiques, ainsi que leurs domaines d'applications. Nous discuterons les principaux facteurs qui influencent la conception des réseaux de capteurs. Et nous terminerons par les grands défis des réseaux de capteurs sans fil.

1.2 Capture et capteurs

La capture est une technique utilisée pour collecter les informations concernant un objet physique ou un processus [DP10], comportant les occurrences des évènements (c-à-d : le changement d'un état comme la température ou bien l'humidité). Un objet qui effectue cette opération est appelé capteur.

1.2.1 Définition

D'une perspective technique, un capteur est un dispositif qui traduit les paramètres ou bien les évènements dans le monde physique aux signaux qui peuvent être mesurés et analysés [DP10]. Donc, un capteur est un dispositif qui convertit l'énergie dans le monde physique en une énergie électrique qui peut être passée à un système de calcul ou de contrôle.

1.2.2 Architecture matériel d'un capteur

La qualité, le volume, et la fréquence des données capturées qui peuvent être extraites du réseau sont influencés par les ressources physiques disponibles dans le nœud capteur. Pour cela, la conception et l'implémentation d'un nœud capteur sans fil est une étape critique. Un nœud capteur est généralement composé de quatre composants de base : l'unité de capture, l'unité de traitement, l'unité de transmission/réception, et l'unité de l'énergie [AV10]. Des composants additionnels peuvent être intégrés dans le nœud capteur selon l'application.

1.2.2.1 L'unité de capture

C'est le composant principal dans un nœud capteur sans fil qui le distingue des autres systèmes embarqués. L'unité de capture peut comporter plusieurs unités de capture et chaque unité de capture est responsable de collecter les informations d'un certain type, comme l'humidité, la température, etc. Elle est composée généralement de deux sous unités : un capteur et un ADC (analog to digital converter) [AV10]. Les signaux analogiques produits par le capteur sont convertis en signaux numériques par l'ADC.

1.2.2.2 L'unité de traitement

C'est le contrôleur principal du nœud capteur sans fil. Elle gère les procédures qui permettent au nœud capteur d'effectuer les opérations de capture, exécuter les algorithmes associés, et collaborer avec les autres nœuds capteur via une communication sans fil [AV10].

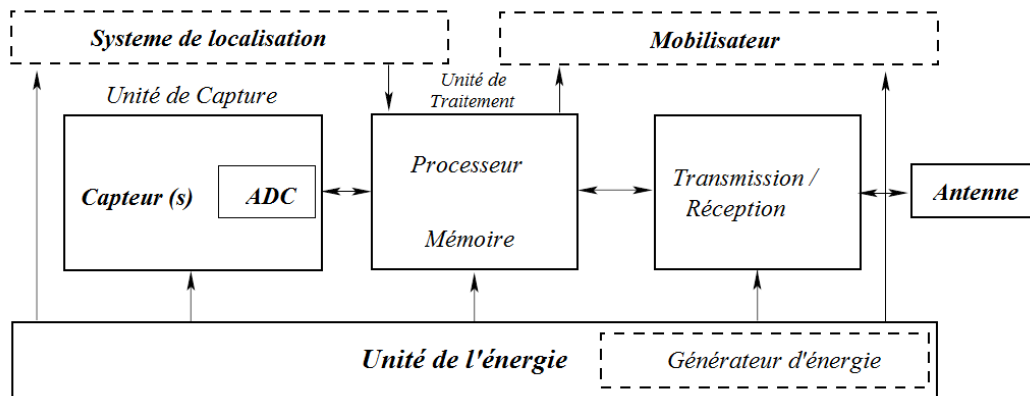


FIG. 1.1 – Architecture hardware générale d'un nœud capteur.

1.2.2.3 L'unité de transmission/réception

La communication entre chaque deux nœuds s'effectue par l'unité de transmission/réception (*Transceiver*). Elle implémente les procédures qui convertissent les bits transmises en ondes radio fréquences et les récupèrent dans l'autre coté [AV10].

1.2.2.4 L'unité de contrôle d'énergie

Un des composants les plus importants des nœuds capteurs sans fil est l'unité de contrôle d'énergie. Chaque composant dans le nœud capteur sans fil est alimenté par l'unité de l'énergie et la capacité limitée de cette unité nécessite des opérations efficaces en énergie pour toutes les tâches effectuées par chaque composant [AV10].

1.2.2.5 Système de localisation

Plusieurs applications réseau, tâches de capture, et techniques de routage nécessitent la connaissance de la position physique du nœud. Donc, il est nécessaire que le nœud capteur soit équipé par un système de localisation. Ce système peut être un module GPS ou bien un module software qui implémente des algorithmes de localisation qui fournissent des informations de la position via des calculs distribués [AV10].

1.2.2.6 Mobilisateur

Le mobilisateur est utilisé pour mouvoir le nœud lorsque il est nécessaire. Le mobilisateur peut aussi opérer en interaction avec l'unité de capture et le processeur pour contrôler le mouvement du nœud capteur [AV10].

1.2.2.7 Générateur d'énergie

Lorsque l'énergie de la batterie s'épuise, un générateur d'énergie additionnel est utilisé pour les applications lorsque le réseau nécessite une longue durée de vie [AV10].

1.2.3 Systèmes d'exploitation

Un système d'exploitation dans un nœud capteur est une couche software qui réside logiquement entre le hardware du nœud et l'application. Sa principale fonction est de permettre aux applications d'interagir avec les ressources hardware, ordonnancer les tâches, et leur donner les priorités [DP10]. Les tâches additionnelles sont :

- Gestion de la mémoire
- Gestion de l'énergie
- Gestion des fichiers
- Le réseau

Dans ce qui suit nous allons présenter quelques exemples des systèmes d'exploitation utilisés dans les nœuds capteurs sans fil.

1.2.3.1 TinyOS

TinyOS [xWG05, GLC07] est le plus utilisé dans les réseaux de capteurs sans fil. Son architecture compacte le rend approprié pour plusieurs applications [DP10]. L'architecture consiste en un ordonnanceur et un ensemble de composants qui peuvent être connectés entre eux par des interfaces bien définies. Les composants sont classifiés en composants de configuration et modules. Un composant de configuration spécifie comment deux ou plusieurs modules sont connectés entre eux, tandis que les modules sont les blocks de base qui construisent le programme de TinyOS. La composition de plusieurs configurations dans un seul code exécutable produit l'application de TinyOS [DP10].

1.2.3.2 SOS

SOS [HKS⁺05] est un système d'exploitation qui consiste en un noyau et un ensemble de modules pouvant être chargés et déchargés. Fonctionnellement, un module est similaire à un composant dans TinyOS ; il implémente une tâche ou une fonction spécifique [DP10]. Une application SOS est composée d'un ou plusieurs modules qui interagissent entre eux. Contrairement à un composant TinyOS qui a un emplacement statique dans la mémoire, un module dans SOS est indépendant ; ce qui lui permet de connecter dynamiquement les modules entre eux [DP10].

1.2.3.3 Contiki

Contiki [DGV04] est un système d'exploitation hybride. Par défaut, son noyau fonctionne comme un noyau orienté événement. Les services communiquent entre eux par l'envoi des événements. Chaque service gère son propre état dans une mémoire privée et le noyau maintient un pointeur sur l'état du processus [DP10].

1.2.3.4 LiteOS

LiteOS [CASH08, CA06] est un système d'exploitation qui supporte les applications multiples [DP10]. Il est basé sur la séparation entre le système d'exploitation et les applications qu'il exécute. Contrairement aux autres systèmes d'exploitation, LiteOS ne fournit pas les composants et les modules qui doivent être connectés ensemble pour construire une application [DP10]. LiteOS permet plusieurs appels système.

1.3 Réseaux de Capteurs Sans Fil (RCSF)

Un capteur n'a pas uniquement un composant de capture, mais aussi des capacités de traitement, communication, et stockage. Avec ces améliorations, un nœud capteur n'est pas seulement responsable de la capture uniquement, mais aussi de l'analyse, la corrélation, et l'agrégation de ses données et des données de ses voisins [DP10]. Quand plusieurs capteurs surveillent en coopération un large environnement physique, ils forment un Réseau de Capteurs Sans Fils (RCSF). Les nœuds capteurs ne communiquent pas entre eux seulement mais aussi avec la station de base en utilisant ses radios sans fil permettant le traitement distant, la visualisation, l'analyse, et les systèmes de stockage. La figure suivante présente deux champs de capteurs surveillant deux régions géographiques différentes, et qui se connectent à l'Internet via leurs stations de base [DP10].

Les capacités des nœuds capteurs dans le RCSF peuvent varier [DP10], c à d, les nœuds simples surveillent un seul phénomène physique, tandis que les nœuds complexes peuvent combiner plusieurs techniques de capture (acoustique, optique, magnétique etc.). Les nœuds complexes peuvent effectuer des fonctions de traitement et d'agrégation extensives. Finalement, quelques capteurs peuvent avoir des technologies additionnelles comme les GPS qui leurs permettent de déterminer leurs positions exactement. Cependant, de tels systèmes consomment davantage d'énergie.

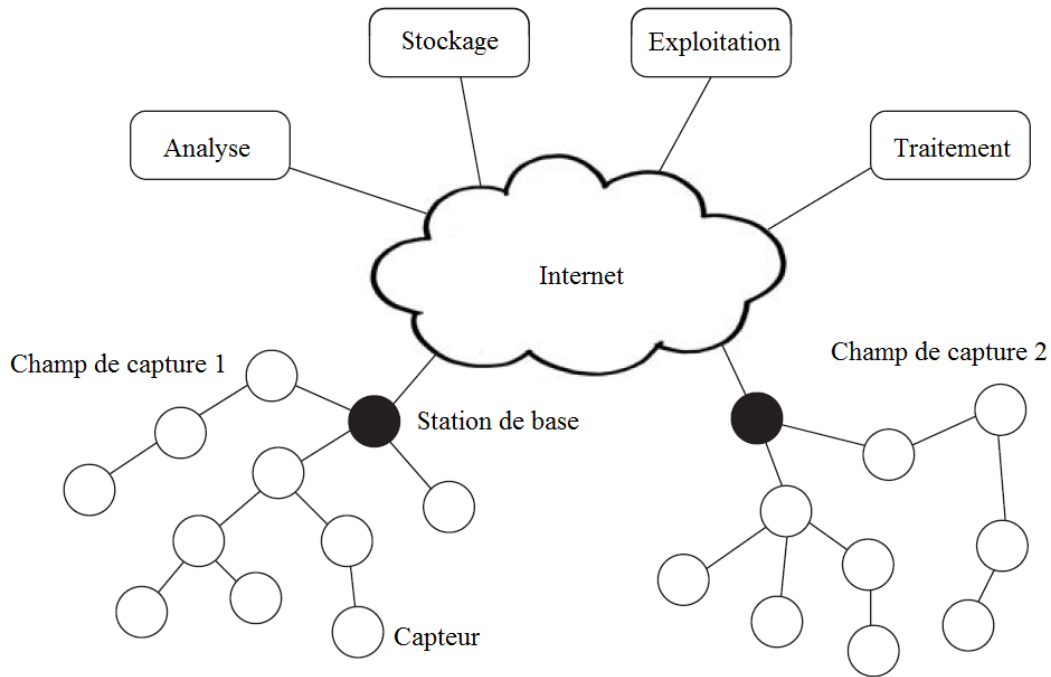


FIG. 1.2 – Un réseau de capteurs sans fil.

1.3.1 Architecture des RCSFs

Les nœuds capteurs sont généralement dispersés dans un champ de capture, chacun de ces nœuds capteurs est capable de collecter les données et de les router vers la station de base. Les données sont routées vers l'utilisateur final via une architecture sans infrastructure multi-sauts à travers la station de base comme illustré dans la figure 1.3

Dans les réseaux de capteurs sans fils, les nœuds capteurs ont la double fonctionnalité. Ils sont initiateurs de données et routeurs de données [AV10]. Pour cela la communication est faite pour deux raisons :

- *Fonction de source* : les nœuds sources avec les informations des évènements font les fonctionnalités de communication pour transmettre leurs paquets vers la station de base.
- *Fonction de routeur* : les nœuds capteurs participent aussi à l'acheminement des paquets reçus à partir des autres nœuds vers la prochaine destination dans le chemin multi-sauts jusqu'à la station de base.

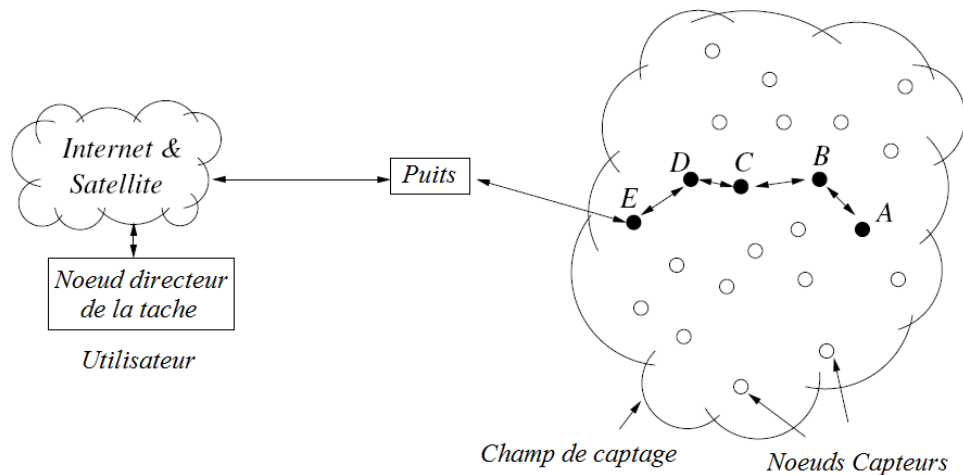


FIG. 1.3 – Nœuds capteurs dispersés dans un champ

1.3.2 Pile protocolaire

La pile protocolaire utilisée par la station de base ainsi que tous les autres nœuds capteurs est illustrée par la figure 1.4. Cette pile prend en charge le problème de la consommation d'énergie, intègre le traitement des données transmises dans les protocoles de routage, et facilite le travail coopératif entre les capteurs [AV10]. Elle est composée de la couche physique, la couche liaison de données, la couche réseau, la couche transport, la couche application, ainsi que le plan de synchronisation, plan de localisation, plan de gestion de la topologie, plan de gestion de l'énergie, plan de gestion de la mobilité, et plan de gestion de la tâche .

1.3.2.1 Les Couches

La couche physique

Elle traite les techniques de la modulation, la transmission, et la réception. Elle est responsable de la sélection de la fréquence, la génération de la porteuse de la fréquence, la détection du signal, la modulation, et le chiffrement des données.

La couche liaison de données

Comme l'environnement est bruyant et les nœuds peuvent être mobiles, la couche liaison de données est responsable de la fiabilité de la communication à travers les

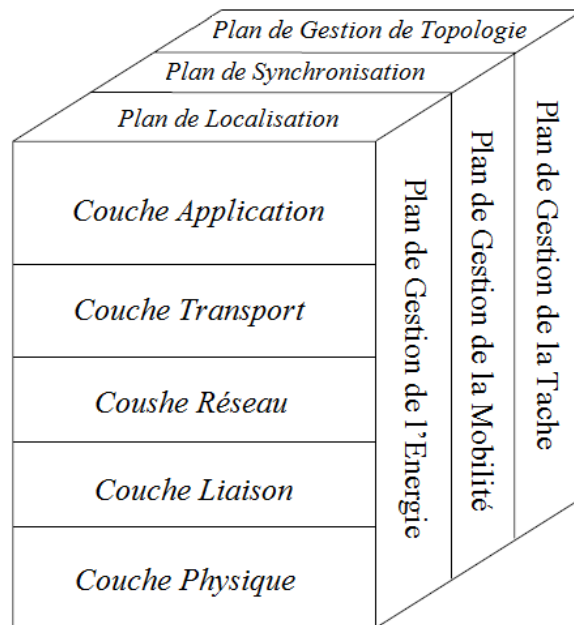


FIG. 1.4 – Pile protocolaire d'un réseau de capteurs sans fil.

techniques de contrôle des erreurs et la gestion des accès au niveau de la sous-couche MAC en minimisant les collisions avec les autres nœuds.

- **MAC** : Un protocole MAC dans un réseau de capteurs sans fil multi-sauts auto-organisé doit accomplir deux buts. Le premier but est la création de l'infrastructure du réseau [AV10]. Comme il y a des milliers de nœuds capteurs dispersés dans un champ de capture, le schéma MAC doit établir la liaison de communication pour le transfert de données. Le deuxième objectif est le partage efficace des ressources de communication entre les nœuds capteurs [AV10]. Ces ressources comportent le temps, l'énergie, et la fréquence. Dans n'importe quel schéma d'accès au media, l'efficacité énergétique est la plus importante [AV10].

Un protocole MAC doit supporter l'opération du mode de sauvegarde de l'énergie du nœud capteur. Le moyen le plus évident pour la conservation de l'énergie est d'éteindre le Transmetteur/Receveur quand il n'est pas demandé. Malgré que cette méthode conserve l'énergie significativement, elle peut influencer sur la connectivité du réseau. Premièrement, quand le Transmetteur/Receveur est éteint, le nœud ne peut pas recevoir de paquets de ses voisins et devient déconnecté du réseau. De plus l'opération d'éteindre et de mise sous tension du Transmetteur/Receveur a un overhead en terme de consommation de l'énergie à cause des procédures de démarrage et de fermeture pour le Hardware et le Software [AV10].

- **Contrôle d’erreurs** : Une autre fonction de la couche liaison de données est le contrôle d’erreurs de la transmission de données. Deux modes importants de contrôle d’erreurs dans la communication réseau : forward error correction (FEC), et automatic repeat request (ARQ), et hybride ARQ [AV10].

La couche réseau

Les nœuds capteurs sont dispersés dans le champ de capture soit près soit à l’intérieur du phénomène comme illustré dans la figure 1.3. Les informations collectées concernant le phénomène doivent être transmises vers la station de base qui peut être localisé loin du champ de capture. Cependant, la limitation du rang de communication des nœuds capteurs empêche la communication directe entre chaque nœud capteur et la station de base. Cela nécessite un protocole de routage sans fil multi-sauts efficace entre les nœuds capteurs et la station de base en utilisant des nœuds intermédiaires comme des relais. Les techniques de routage existantes, qui ont été développées pour les réseaux ad hoc sans fil, ne remplissent pas les exigences des réseaux de capteurs sans fil. La couche réseau des réseaux de capteurs sans fil est généralement conçue selon les principes suivants :

- L’efficacité en énergie est toujours une considération importante.
- La plupart des réseaux de capteurs sont centrés-données (*data-centric*).
- En plus du routage, le relais des nœuds peut agréger les données provenant de différents voisins en effectuant un traitement local.
- A cause de la densité des nœuds capteurs dans un réseau de capteurs, un ID pour chaque nœud ne peut pas être disponible et les nœuds doivent être adressés en se basant sur leurs données et leurs positions.

Un problème important pour le routage dans les réseaux de capteurs est qu’il peut être basé sur des requêtes centrées-données (*data-centric*). En se basant sur les informations demandées par l’utilisateur, le protocole de routage doit traiter les différents nœuds qui vont fournir les informations demandées. Par exemple, ”les surfaces dont la température est supérieur à 21°C” est une requête plus commune que ”la température lue par le nœud N° 47”.

Une autre fonction très importante de la couche réseau est de fournir l’interconnexion externe avec les autres réseaux comme d’autres réseaux de capteurs, systèmes de commande et de contrôle, et l’internet. Dans un scénario, les stations de base peuvent être utilisés comme une Gateway vers les autres réseaux, tandis que, un autre scénario est de créer un backbone par la connexion des stations de base ensemble et de rendre ce backbone accessible à d’autres réseaux via une gateway [AV10, ASSC02].

La couche transport

La couche transport est nécessaire lorsque le réseau est conçu pour être accessible par Internet ou bien par un réseau externe, TCP avec ses mécanismes de fenêtre de transmission courants ne traite pas les défis posés par l'environnement des réseaux de capteurs sans fil. Contrairement aux protocoles comme TCP, les schémas de communication end-to-end dans les réseaux de capteurs ne sont pas basés sur l'adressage global. Ces schémas doivent considérer que l'adressage basé sur les données ou bien sur la position est utilisé pour indiquer la destination des paquets de données. Des facteurs, comme la consommation d'énergie et le passage à l'échelle, et des caractéristiques comme le routage centré-données signifient que les réseaux de capteurs nécessitent différentes manipulations dans la couche transport [AV10].

Pour la communication à l'intérieur du réseau de capteurs, les protocoles de la couche transport sont nécessaires pour deux fonctionnalités principales : la fiabilité et le contrôle de congestion. Comme les nœuds capteurs sont limités en termes de traitement, stockage, et consommation en énergie, les protocoles de la couche transport aspirent à exploiter les capacités de collaboration de ces nœuds capteurs et déplacent l'intelligence à la station de base à la place des nœuds capteurs [AV10].

La couche application

La couche application contient l'application principale ainsi que les différentes fonctionnalités de gestion. En plus du code de l'application, qui est spécifique pour chaque application, le traitement des requêtes et les fonctionnalités de gestion du réseau résident aussi dans cette couche.

L'architecture en couche a été adoptée dans le développement des RCSFs grâce à son succès avec internet. Cependant, les applications en large échelle des applications des RCSFs révèlent que le canal sans fil a un impact significatif sur les protocoles des couches de haut niveau. En plus, les contraintes des ressources, et la nature spécifique de l'application des RCSFs conduisent aux solutions inter-couches [AV10].

En plus des fonctionnalités de communication dans la pile en couche, les RCSFs sont aussi équipés par plusieurs fonctionnalités qui aident le fonctionnement des solutions proposées. Dans un RCSF, chaque capteur est équipé par sa propre horloge locale pour les fonctions internes. Chaque événement qui est relié à l'opération du capteur comportant la capture, le traitement et la communication est associé à une information de la date contrôlée par l'horloge locale. Comme les utilisateurs sont intéressés par les informations de collaboration des différents capteurs, l'information de la date

associée aux données au niveau de chaque nœud capteur doit être cohérente. De plus, le RCSF doit être capable d'ordonner correctement les événements capturés par des capteurs distribués pour modéliser précisément l'environnement physique. Ces nécessités de datation ont conduit au développement des protocoles time synchronisation dans les RCSFs [AV10].

Les informations collectées doivent être associées par la position des nœuds capteurs pour fournir une vue précise du champ observé. D'ailleurs, les RCSFs peuvent être utilisés pour suivre quelques objets qui nécessitent aussi les informations de la position pour les incorporer avec les algorithmes de suivi. En plus, les services basés sur la localisation et les protocoles de communication nécessitent des informations sur les positions. C'est pourquoi, les protocoles de localisation ont été incorporés dans la pile de communication.

Finalement, plusieurs solutions de gestion de la topologie sont nécessaires pour maintenir la connectivité et la couverture du RCSF. Les algorithmes de gestion de la topologie fournissent des méthodes efficaces pour le déploiement du réseau et qui résultent en une longue durée de vie et des informations de couverture efficaces [AV10].

Les protocoles de clustering sont utilisés pour organiser le réseau en clusters pour améliorer le passage à l'échelle et augmenter la durée de vie du réseau [AV10].

1.3.2.2 Plans de gestion

Les plans de gestion de l'énergie, la mobilité, et de la tâche montrent l'énergie, le mouvement, et la distribution de la tâche entre les nœuds capteurs.

Plan de gestion de l'énergie

Le plan de gestion de l'énergie montre comment le nœud gère son énergie, par exemple le nœud capteur peut éteindre son receveur après la réception d'un message de l'un de ses voisins pour éviter d'avoir des messages dupliqués, aussi lorsque le niveau de l'énergie du nœud capteur est bas, le nœud capteur diffuse à ses voisins que son énergie est basse et ne peut pas participer dans le routage des messages, l'énergie restante est réservée à la capture des données [AV10, KB04, ASSC02].

Plan de gestion de la mobilité

Le plan de gestion de la mobilité détecte et registre le mouvement des nœuds capteurs, donc une route vers l'utilisateur est toujours maintenue, et les nœuds peuvent garder une trace de leurs voisins. Avec la connaissance de ces voisins les nœuds peuvent balancer l'utilisation de leurs énergies et tâches [AV10, KB04, ASSC02].

Plan de gestion de la tâche

Le plan de gestion de la tâche balance et ordonnance les tâches de capture données à une région spécifique. Il n'est pas nécessaire de faire une tâche de capture dans cette région par tous les nœuds capteurs en même temps, donc quelques nœuds font la tâche de capture plus que d'autres, selon leurs niveaux de l'énergie [AV10, KB04, ASSC02].

Ces plans de gestion sont nécessaires afin que les nœuds capteurs puissent travailler ensemble de manière efficace en termes d'énergie, routage de données dans un réseau de capteurs, et partage de ressources entre les nœuds capteurs. Sans ces plans de gestion, chaque nœud travaillera individuellement. Il serait très efficace que les nœuds capteurs puissent collaborer entre eux pour prolonger la durée de vie du réseau [AV10, KB04, ASSC02].

1.3.3 Caractéristiques des RCSF

Forte densité

Les réseaux de capteurs sont caractérisés par une grande densité des nœuds déployés dans la région à surveiller. Un réseau peut contenir des centaines voire des milliers de nœuds capteurs.

Energie limitée

Les RCSFs visent la consommation d'énergie puisque l'alimentation de chaque nœud est assurée par une source d'énergie limitée et généralement irremplaçable à cause de l'environnement hostile où il est déployé. De ce fait, la durée de vie d'un RCSF dépend fortement de la conservation d'énergie au niveau de chaque nœud.

Modèle de communication

Les nœuds dans les RCSFs communiquent selon un paradigme plusieurs-à-un (*many to one*). En effets, les nœuds capteurs collectent des informations à partir de leur environnement et les envoient toutes vers un seul nœud qui représente le centre de traitement.

Absence d'adressage fixe des nœuds

Les nœuds dans les réseaux sans fil classiques sont identifiés par des adresses IP. Cependant, cette notion n'existe pas dans les RCSFs. Ces derniers utilisent un adressage basé sur l'attribut du phénomène capté, on parle donc de l'adressage basé-attribut. En effet, les requêtes des utilisateurs ne sont généralement pas destinées à un seul nœud, mais plutôt, à un ensemble de nœuds identifiés par un attribut. Par exemple, identifier un ensemble de nœuds par "les nœuds qui captent le volume du CO₂ dépassant 0,0375 % dans l'atmosphère".

Ressources limitées

Les capteurs sont des objets communicants à taille très réduite ce qui limite leurs ressources en termes de : mémoire disponible, puissance de traitement, bande passante et particulièrement en quantité d'énergie embarquée.

Sécurité physique limitée

A l'instar des réseaux ad hoc, les réseaux de capteurs sont plus touchés par le paramètre de sécurité que les réseaux filaires classiques. Cela se justifie par les contraintes et limitations physiques qui minimisent le contrôle sur les données transférées.

Auto organisation du réseau et la dynamique de la topologie

La mobilité des nœuds capteurs, la défaillance des nœuds dûe à différentes raisons, et le rajout de nouveaux nœuds provoquent une topologie instable. Donc, un réseau de capteurs doit être capable de s'auto-organiser pour continuer ses applications.

1.4 Applications des RCSFs

Les RCSFs peuvent consister en plusieurs différents types de capteurs, sismique, magnétique, visuel, infrarouge, acoustique, et radar, qui sont capables de surveiller une large variété de conditions : température, humidité, pression, vitesse, direction, mouvement, niveau de bruit, la présence ou l'absence de certains types d'objets [AV10]. Par conséquent, plusieurs applications sont possibles.

1.4.1 Applications militaires

Comme dans le cas de plusieurs technologies, le domaine militaire a été un moteur initial pour le développement des réseaux de capteurs sans fil. Le déploiement rapide, le coût réduit, l'auto-organisation et la tolérance aux pannes des réseaux de capteurs sans fil sont des caractéristiques qui rendent ce type de réseaux un outil appréciable dans un tel domaine [AV10].

Un réseau de capteurs sans fil déployé sur un secteur stratégique ou difficile d'accès, permet par exemple d'y surveiller tous les mouvements (amis ou ennemis), ou d'analyser le terrain avant d'y envoyer des troupes (détection d'agents chimiques, biologiques ou de radiations).

1.4.2 Applications environnementales

Les capacités de coordination autonomes des RCSFs sont utilisées dans la réalisation d'une large variété d'applications environnementales. Quelques applications environnementales des RCSFs comportent le suivi des mouvements des oiseaux, de petits animaux et des insectes, surveillance des conditions environnemental qui affectent les récoltes, irrigation, détection chimique / biologique, agriculture de précision, contrôle biologique, détection de feu de forêt, recherche météorologique ou géophysique, détection d'inondation, et études de pollution [AV10].

1.4.3 Applications dans le domaine de la santé

Les événements dans les dispositifs biomédicaux implantés et les capteurs smart intégrés rendent l'utilisation des réseaux de capteurs pour les applications biomédicales possibles. Certaines applications sanitaires pour les réseaux de capteurs sont : contrôle intégré du patient, diagnostique, administration des médicaments dans les hôpitaux, contrôle des mouvements et des processus internes d'insectes ou d'autres petits animaux,

contrôle de données humaines physiologiques, et le contrôle et le suivi des docteurs et des patients à l'intérieur d'un hôpital [AV10].

1.4.4 Applications domotiques

Comme la technologie avance, les nœuds capteurs smart et les déclencheurs peuvent être enterrés dans des appareils comme des aspirateurs, des fours à micro-ondes, des réfrigérateurs et des lecteurs DVD aussi bien que des systèmes de contrôle d'eau [AV10]. Ces nœuds capteurs à l'intérieur des dispositifs domestiques peuvent interagir entre eux et avec le réseau externe via l'internet ou le satellite. Ils permettent aux utilisateurs finaux de gérer facilement les dispositifs domestiques localement et à distance.

1.4.5 Applications industrielles

Les réseaux de capteurs câblés ont été utilisés pour une longue période de temps dans les champs industriels. Cependant, le coût associé au déploiement des capteurs câblés limite l'applicabilité de ces systèmes. Les RCSFs est une solution prometteuse pour ces systèmes grâce à leur simplicité de déploiement, la haute granularité, et la haute exactitude fournies à partir des unités de communication sans fil.

Certaines applications commerciales contrôlent : la construction de claviers virtuels, la qualité de produit, la construction d'espaces de bureau intelligents, l'environnemental d'immeubles de bureaux, le robot dans des environnements automatiques industriels, les jouets interactifs, les musées interactifs, la commande de processus d'usine et automatisation, les régions sinistrées, le diagnostic de machine, le transport, l'instrumentation d'usine, les déclencheurs, le vol de voiture [AV10].

1.5 Contraintes de conceptions des RCSFs

La conception et la réalisation des réseaux de capteurs sans fil est influencée par plusieurs paramètres, parmi lesquels nous citons les contraintes matérielles, la tolérance aux pannes, la scalabilité, le coût de production, la topologie du réseau, le support de transmission et la consommation d'énergie. Ces facteurs importants servent comme directives pour le développement des algorithmes et protocoles utilisés dans les réseaux de capteurs, ils sont considérés également comme métriques de comparaison de performances entre les différents travaux dans le domaine.

1.5.1 Contraintes matérielles

Un nœud capteur contient quatre composants de base : l'unité de captage, l'unité de traitement, l'unité de transmission, et l'unité de contrôle d'énergie. Il peut contenir également, suivant son domaine d'application, des éléments supplémentaires tels qu'un système de localisation, ou bien un système générateur d'énergie. Toutes ces unités peuvent exiger leur intégration dans un boîtier de taille minimale inférieure à un centimètre cube, et avec un poids très léger qui permet aux nœuds de rester suspendus dans l'air, si l'application l'exige [AV10, KB04, ASSC02].

Autre que la taille, il existe d'autres contraintes exigeantes pour la construction des nœuds capteurs, qui doivent :

- Consommer le minimum d'énergie
- Opérer dans une haute densité
- Avoir un coût de production réduit
- Etre autonome et pouvoir opérer sans assistance
- Etre adaptatif à l'environnement

1.5.2 La tolérance aux pannes

La défaillance ou le blocage des nœuds dans un réseau de capteurs peut être engendré par plusieurs causes, notamment l'épuisement d'énergie, l'endommagement physique, ou les interférences liées à l'environnement. La panne d'un nœud capteur conduit à la déconnection de ce dernier du réseau. Donc, la panne d'un nœud ne doit pas affecter le fonctionnement du réseau entier. Le niveau des pannes qui sont autorisés pour que le réseau puisse continuer à fonctionner définit sa tolérance aux pannes [AV10, KB04, ASSC02]. Plus spécifiquement, la tolérance aux pannes est la capacité de soutenir les fonctionnalités du réseau sans aucune interruption à cause des pannes d'un nœud. Les protocoles et les algorithmes conçus pour les RCSFs visent à traiter les pannes fréquentes des nœuds capteurs à travers la redondance [AV10].

1.5.3 Scalabilité

Le nombre des nœuds capteurs déployés pour surveiller un phénomène physique peut être de l'ordre de plusieurs centaines ou milliers de capteurs. Pour cela, les protocoles réseau développés pour ces réseaux doivent être capables de manipuler ce grand nombre de nœuds d'une manière efficace [AV10, KB04, ASSC02].

1.5.4 Coût de production

Comme le réseau de capteur consiste en un grand nombre de nœuds capteurs, le coût d'un nœud capteur est très important pour justifier le coût total du réseau entier. Pour cela, le coût de chaque nœud capteur doit être réduit. De plus, un nœud capteur peut avoir des unités supplémentaires. Aussi, il peut être équipé par un système de localisation, mobilisateur, ou générateur d'énergie. Toutes ces unités augmentent le coût du nœud capteur. Donc le coût d'un capteur est un problème très important étant donné le nombre de fonctionnalités [AV10, KB04, ASSC02].

1.5.5 Support de transmission

Le bon fonctionnement d'un RCSF dépend de la communication fiable entre les nœuds dans le réseau [AV10]. Dans un réseau de capteurs, les nœuds sont liés via un moyen de communication sans fil, et ceci, en utilisant un support optique, ou des fréquences radio. Cependant, il faut s'assurer de la disponibilité du moyen de transmission choisi dans l'environnement de captage, afin de permettre au réseau d'accomplir la totalité de ses tâches. Pour les liens de communication via les fréquences radio, les bandes ISM (*Industrial Scientific Medical bands*) peuvent être utilisées, ces bandes de fréquence sont employées pour assurer des communications libres de la charge dans le domaine industriel, scientifique ou médical, et ceci dans la plupart des pays du monde [AV10, KB04, ASSC02]. Pour les réseaux de capteurs, les unités de transmission intégrées au niveau des nœuds doivent être de petite taille et à faible consommation d'énergie.

1.5.6 Consommation d'énergie

Comme les nœuds capteurs sont des composants micro-électroniques, ils ne peuvent être équipés que par des sources limitées d'énergie (<0.5 Ampère-heure, 1.2 V) [AV10, KB04]. De plus, dans certaines applications, ces nœuds ne peuvent pas être dotés de mécanismes de rechargement d'énergie, par conséquent, la durée de vie d'un nœud capteur dépend fortement de la durée de vie de la batterie associée. Dans les réseaux de capteurs, l'efficacité en consommation d'énergie représente une métrique de performance significative, qui influe directement sur la durée de vie du réseau en entier. Pour cela, les concepteurs peuvent au moment du développement des protocoles négliger les autres métriques de performance telle que la durée de transmission et le débit, au détriment du facteur de consommation d'énergie. Les tâches principales d'un nœud capteur dans un champ de capture sont : la détection des événements, le traitement local des données, et

la transmission des données [AV10]. Donc la consommation d'énergie peut être divisée en trois domaines : capture, transmission, et traitement des données. La figure 1.5 présente la consommation d'énergie d'un nœud capteur MicaZ.

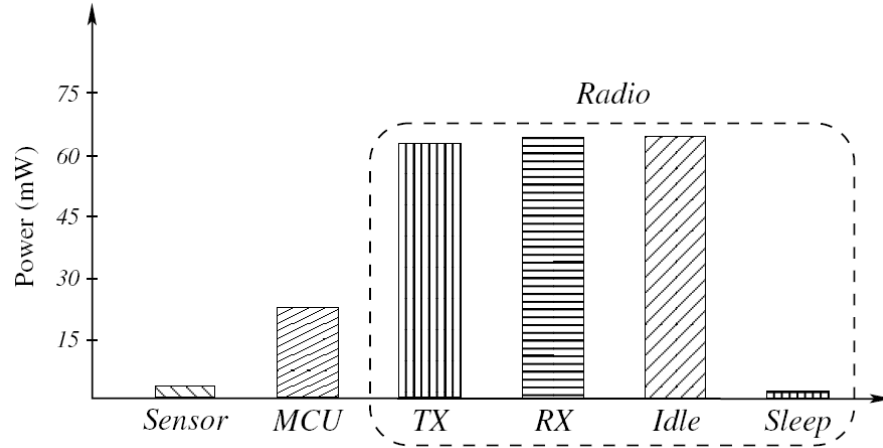


FIG. 1.5 – La consommation d'énergie d'un nœud capteur MicaZ.

- *Capture* : la tâche de capture est effectuée par les composants d'acquisition qui traduisent les phénomènes physiques en signal électrique. L'énergie consommée lors de la capture peut être réduite en utilisant des composants à faible consommation mais en réduisant ainsi leurs performances. Alors, la solution la plus envisageable consiste à réduire les durées de capture et donc supprimer les captures jugées redondantes et inutiles.
- *Traitement de données* : le traitement comporte le contrôle des composants de capture et l'exécution des protocoles de communication sur les données collectées. Les microprocesseurs sont à la charge de ce traitement. Le choix de ces derniers est en fonction du scénario de l'application et la consommation d'énergie souhaitée, et en général un compromis est réalisé entre le niveau de performance et la consommation d'énergie.
- *Communication* : l'énergie de communication représente la plus grande quantité d'énergie totale consommée au niveau d'un nœud capteur. La communication est principalement liée aux protocoles développés pour la couche liaison et la couche réseau. Pour minimiser l'énergie consommée par la communication, on doit intervenir au niveau de ces deux couches. Pour ce faire, plusieurs approches de communication avec conservation d'énergie ont été proposées pour la couche réseau ainsi que pour la couche liaison et en particulier pour la sous couche MAC.

1.6 Les grands défis

Les solutions efficaces en énergie sont extrêmement nécessaires pour chaque aspect de la conception du RCSF pour délivrer les avantages potentiels du phénomène du RCSF [AV10]. Pour cela, dans les solutions existantes et les futures solutions pour les RCSFs, l'efficacité en énergie constitue un grand défi. En plus de l'efficacité en énergie, il existe plusieurs autres défis.

Intégration des RCSFs et de l'Internet

Le standard 6LoWPAN a été développé pour intégrer le standard IPv6 avec les nœuds capteurs à basse énergie [AV10]. En conséquence, l'entête du paquet IPv6 est compressé à une taille appropriée pour les motes capteurs. Cependant, des défis significatifs dans l'intégration continue entre les RCSFs et l'Internet existent toujours au niveau des couches plus hautes de la pile protocolaire. De plus, le routage de bout en bout entre un nœud capteur et un équipement Internet n'est pas faisable en utilisant les solutions existantes. Aussi, les solutions existantes de la couche transport pour les RCSFs ne sont pas compatibles avec les protocoles TCP et UDP [AV10].

La communication en temps réel et multimédia

La majorité des solutions développées pour les RCSFs focalisent sur l'efficacité en énergie. Cependant, les développements récents dans les robotiques et la nécessité d'effectuer des tâches spécialisées ont conduit au développement des WSNs (*wireless sensors and actor networks*) et CPS (*cyber-physical systems*). Dans ces réseaux, les garanties de l'opération en temps réel doivent être fournies à travers les protocoles du réseau [AV10].

La pile protocolaire

La plupart des protocoles MAC sont adaptés aux topologies où les nœuds sont statiques. Cependant, le développement dans les MEMS (*Micro Electro-Mechanical Systems*) et la technologie robotique ont permis la production des nœuds mobiles. En conséquence, l'aide à la mobilité au niveau de la couche MAC est nécessaire pour les applications des RCSFs. Dans la couche transport de nouvelles solutions, qui traitent les besoins de livraison de la multimédia sur les RCSFs, doivent être développées. L'optimisation inter-couche des couches transport, routage, liaison, et physique doit être

examinée et les limites théoriques doivent être identifiées pour développer de nouveaux protocoles de communication inter-couche pour un transport fiables dans les RCSFs [AV10].

Synchronisation et localisation

Malgré que plusieurs protocoles traitent plusieurs aspects de synchronisation et de localisation, il reste plusieurs problèmes de recherche ouverts. A titre d'exemple, les protocoles de synchronisation existants fournissent une précision qui est acceptable pour la plupart des applications des RCSFs. Cependant, cette précision n'est pas acceptable pour le développement de WMSN. Les contraintes de la communication en temps réel et les exigences temporelles de la multimédia nécessitent des méthodes de synchronisation avec une faible surcharge (*overhead*) [AV10].

RCSFs dans les environnements difficiles

Les avantages prometteurs des RCSFs ont ouvert de nouveaux secteurs d'applications dans des environnements stimulants tels que souterrain et sous-marin, qui exigent des protocoles de communication qui sont capables de faire face aux caractéristiques et aux dommages du moyen de propagation et les caractéristiques environnementales [AV10].

1.7 Conclusion

Le développement des réseaux de capteurs a été rendu possible grâce aux avancées importantes des technologies sans fil et de l'électronique digitale. Les RCSFs sont composés d'un nombre important de nœuds, contenant chacun un capteur, un processeur et une interface de communication et sont soumis à des contraintes qui proviennent de faibles ressources physiques disponibles au sein de chaque capteur et concernant aussi bien la réserve de l'énergie, la capacité de calcul ou la taille de la mémoire. Ces réseaux peuvent se révéler très utiles pour de nombreuses applications militaires ou civiles lorsqu'il s'agit de collecter et traiter des informations provenant de l'environnement.

Aujourd'hui, les RCSFs constituent un champ de recherche très vaste vu leur importance. Pour l'aspect informatique, les travaux se portent sur le développement de nouvelles techniques de sécurité, méthodes de synchronisation, de nouveaux protocoles

et algorithmes de communication notamment le routage efficace en terme de consommation de l'énergie. L'objectif de ces recherches est d'optimiser les performances des RCSFs tout en prolongeant leurs durées de vie.

La consommation d'énergie est un facteur à prendre en considération dans la conception des réseaux de capteurs. Puisqu'une grande partie de cette ressource critique est utilisée pendant la communication, plusieurs protocoles de communication en particulier ceux relatifs à l'acheminement de données ont été proposés afin de minimiser considérablement la dissipation de l'énergie. L'objectif du prochain chapitre est de présenter une étude des différentes familles de protocoles de routage dédiés aux réseaux de capteurs.

Chapitre 2

Le routage dans les RCSFs

2.1 Introduction

Dans un réseau de capteurs, les données collectées par les nœuds capteurs sont propagées vers une station de base qui connecte le réseau de capteurs avec les autres réseaux où les données peuvent être visualisées et analysées. Comme les RCSFs contiennent un grand nombre de nœuds capteurs qui couvrent une surface large, des approches de communication indirectes (multi-sauts) sont nécessaires. Les nœuds capteurs doivent servir comme des relais ou des nœuds d'acheminement pour les autres nœuds. Le processus d'établissement des chemins entre la source et la station de base est appelé routage. La couche réseau de la pile protocolaire est responsable de cet acheminement. Lorsque les nœuds d'un réseau sont déployés d'une manière déterministe (c-à-d, ils sont placés dans des emplacements prédéterminés), la communication entre ces derniers et la station de base peut s'effectuer en utilisant des routes prédéterminées. Cependant, lorsque les nœuds sont déployés de manière aléatoire, la topologie résultante est imprévisible. Dans ce cas, il est nécessaire que ces nœuds s'auto-organisent, autrement dit, ils doivent coopérer pour déterminer leurs positions, identifier leurs voisins, et découvrir des chemins vers la station de base.

Dans ce chapitre nous allons discuter les différents défis de routage dans les réseaux de capteurs sans fil, ainsi que les métriques utilisées par les protocoles de routage pour le choix des routes. Puis, nous allons donner les différentes approches de classification des protocoles de routage dans les réseaux de capteurs sans fil.

2.2 Considérations de conception d'un protocole de routage dans les RCSFs

Le routage est l'un des principaux problèmes dans les RCSFs et plusieurs solutions ont été proposées pour traiter ce problème.

2.2.1 Consommation d'énergie

Le principal objectif des protocoles de routage est la délivrance efficace des informations entre les nœuds capteurs et la station de base. A cause de la limitation des ressources énergétiques des nœuds capteurs, les données doivent être délivrées d'une manière efficace en énergie sans affecter la précision des données [AV10]. La consommation d'énergie pour le routage dans les RCSFs peut être classifiée comme suit :

- *Découverte des voisins* : plusieurs protocoles de routage nécessitent l'échange des informations entre les voisins. Par exemple, les protocoles de routage géographiques nécessitent la connaissance des positions des nœuds voisins. Les nœuds consomment l'énergie dans l'échange de ces informations qui augmente l'overhead du protocole. Pour améliorer l'efficacité énergétique du protocole, les échanges des informations doivent être minimisés sans affecter la précision du protocole [AV10].
- *Communication vs traitement* : il est connu que le traitement est moins coûteux que la communication en terme de la consommation d'énergie. En plus, dans les RCSFs, le but est de délivrer les informations au lieu de délivrer des paquets individuels [AV10]. Par conséquent, en plus des techniques d'acheminement des paquets, le traitement doit être intégré avec le routage pour améliorer la consommation d'énergie. Par exemple, les données provenant de nœuds multiples peuvent être agrégées dans un seul paquet pour minimiser le volume du trafic sans affecter le contenu des informations.

2.2.2 Scalabilité

La densité élevée des nœuds, qui constituent un RCSF, rend la connaissance globale de la topologie du réseau par chaque nœud impossible. Pour cela, des protocoles distribués qui fonctionnent avec une connaissance limitée de la topologie doivent être développés pour améliorer le passage à l'échelle [AV10].

2.2.3 Adressage

La densité élevée des nœuds qui constituent un RCSF, rend l'affectation d'une adresse unique à chaque nœud impossible. Pour cela, la majorité des protocoles de routage ad hoc ne sont pas valables pour les RCSFs tant que ces solutions nécessitent des adresses uniques pour chaque nœud dans le réseau. En conséquence, de nouveaux mécanismes d'adressage ou de nouvelles techniques de routage qui ne nécessitent pas des identificateurs pour chaque nœud sont nécessaires [AV10].

2.2.4 Robustesse

Les RCSFs s'appuient sur les nœuds dans le réseau pour délivrer les données de manière multi-sauts. Les protocoles de routage sont hébergés dans ces nœuds non pas dans des routeurs comme dans l'internet. Donc, les protocoles de routage doivent fournir la robustesse pour les pannes des nœuds et interdire l'apparition de points de panne simples où les informations sont perdues si un nœud tombe en panne [AV10].

2.2.5 Topologie

Le déploiement d'un RCSF peut être soit prédéterminé soit de façon aléatoire. La topologie prédéterminée peut être exploitée pour concevoir des protocoles de routage plus efficaces, mais ce n'est pas toujours le cas. En conséquence, les nœuds individuels sont généralement inconscients de la topologie initiale du réseau. Cependant, la position relative des voisins d'un nœud et la position relative des nœuds dans le réseau affectent significativement la performance du routage [AV10]. Pour cela, les protocoles de routage doivent fournir la conscience de la topologie de telle façon que le voisinage de chaque nœud soit découvert et sur lequel les décisions de routage se font. De plus, la topologie peut être changée pour plusieurs raisons. En conséquence, le protocole de routage doit être adaptable à ces changements.

2.2.6 Application

Dans les applications de surveillance, les nœuds communiquent généralement leurs observations avec la station de base d'une manière périodique. Donc, des routes statiques peuvent être maintenues. Dans les applications qui sont basées sur l'évènement lorsqu'un évènement se produit, des routes doivent être générées pour délivrer les informations de l'évènement. De plus, la localisation de l'évènement n'est pas fixée et donc de nouvelles routes doivent être générées pour chaque évènement. Il est évidemment clair que les techniques de routage sont reliées à l'application[AV10].

2.2.7 Hétérogénéité des nœuds/liens

De nombreux travaux supposent que tous les nœuds sont homogènes en termes de calcul, de paramètres et de puissances de communication. Cependant, selon l'application, un nœud peut avoir un rôle ou des possibilités différentes. Par exemple, certaines applications peuvent nécessiter des capteurs de natures différentes pour surveiller la température, la pression et l'humidité de l'environnement, détectant le mouvement par l'intermédiaire de signatures acoustiques, et capturant l'image ou le cheminement visuel d'objets mobiles. Ces capteurs peuvent contenir des fonctionnalités différentes. De même, la capture et la transmission des données peuvent être effectuées par ces capteurs avec des caractéristiques différentes. Par exemple, les protocoles hiérarchiques nécessitent des nœuds "cluster-heads" différents des nœuds capteurs classiques. Ces cluster-heads peuvent être plus puissants que les autres nœuds capteurs en terme d'énergie, de bande passante et de mémoire. Par conséquent, ces cluster-heads sont chargés de la transmission des données à la station de base [AKK04].

2.2.8 Média de transmission

Dans un réseau de capteurs multi-sauts les nœuds communiquant sont liés par un médium sans fil. En général, la largeur de bande requise des données sera basse, sur l'ordre de 1-100 kb/s. Les problèmes classiques liés au canal sans fil (par exemple un taux d'erreur plus élevé en raison de la présence d'interférences) peuvent affecter le fonctionnement du réseau [AKK04].

2.2.9 Connectivité

La densité parfois élevée des nœuds dans les réseaux de capteurs exclut complètement l'isolement entre eux. Par conséquent, on s'attend à ce que des nœuds capteurs soient fortement reliés. Ceci, n'élimine pas le risque de partitionnement du réseau dû aux échecs des nœuds capteurs [AKK04].

2.2.10 Agrégation de données

Selon la densité des nœuds dans l'espace couvert, les données mesurées peuvent être identiques (lorsqu'elles sont relevées par des capteurs très proches les uns des autres). Pour réduire le nombre de transmissions et donc économiser de l'énergie, les données identiques peuvent être agrégées. L'agrégation de données est la combinaison des données de différentes sources selon une certaine fonction d'agrégation, par exemple,

qui supprime les valeurs identiques, transmet le minimum, le maximum ou la moyenne [AKK04].

2.2.11 Qualité de service

Dans quelques applications, les données devraient être fournies au cours de certaines périodes du moment où elles sont senties, sinon elle sont inutiles, et par conséquent, la latence définie pour la livraison des données est une autre condition pour les applications qui sont soumises à des contraintes de temps. Cependant, dans plusieurs applications, la conservation d'énergie, qui est directement liée à la durée de vie du réseau est considérée relativement plus importante que la qualité des données envoyées. Pendant que l'énergie s'épuise, le réseau exige de réduire la qualité des résultats afin de réduire la diminution d'énergie dans les nœuds, et par conséquent augmenter la durée de vie du réseau [AKK04].

2.3 Métriques de routage

Dans n'importe quel schéma utilisé dans le réseau (périodique, basé sur l'évènement, ou basé sur la requête), la conception d'un protocole de routage est dirigé par les ressources disponibles dans le réseau et les besoins des applications [DP10]. Pour cela, des métriques de routage sont utilisées pour exprimer une variété d'objectifs d'un protocole de routage.

2.3.1 Nombre de sauts minimum

La métrique la plus utilisée dans les protocoles de routage des RCSFs est le nombre de sauts minimum. Dans cette technique le protocole de routage essaye de trouver un chemin vers la destination qui comporte un nombre minimum de nœuds relais sachant que tous les liens ont le même coût [DP10].

2.3.2 Energie

Sans doute l'aspect le plus important pour le routage dans les RCSFs est l'efficacité en énergie. Cependant, il y a plusieurs interprétations de l'efficacité énergétique [DP10] :

- *Energie minimum consommée par paquet* : le but est de minimiser la quantité totale de l'énergie dépensée pour la propagation d'un paquet de la source

jusqu'à la destination. L'énergie totale est donc la somme des énergies consommées par chaque nœud à travers une route pour la réception et la transmission d'un paquet.

- *Variance minimum dans les niveaux d'énergie des nœuds* : tous les nœuds dans le réseau sont considérés importants et le défi est de distribuer la consommation d'énergie à travers tous les nœuds dans le réseau de façon équitable. Le but de cette technique est de maximiser la durée de vie du réseau entier.
- *Capacité énergétique maximum* : dans cette approche on a une vue centrée sur la capacité énergétique des nœuds capteurs. Le protocole de routage favorise les routes qui ont la capacité énergétique la plus grande de la source jusqu'à la destination.

2.3.3 Qualité de service

Le terme qualité de service (QoS) réfère aux mesures de performance définies dans le réseau y compris la latence (délais), le débit, la gigue (variation dans la latence), et la perte des paquets (taux d'erreur). Le choix de la métrique de QoS dépend du type de l'application [DP10].

2.3.4 Robustesse

Plusieurs applications de capture préfèrent utiliser des routes qui restent stables pour une longue période de temps. Un nœud peut mesurer ou estimer la qualité de la liaison avec chacun de ses voisins, puis sélectionner un voisin, qui augmente la probabilité de succès d'une transmission, comme prochain saut. Dans les réseaux avec des nœuds mobiles, un protocole de routage peut aussi utiliser la métrique de stabilité de liaison, qui mesure combien probablement une liaison sera disponible dans le futur [DP10].

2.4 Classification des protocoles de routage dans les RCSFs

Il y a plusieurs façons de classifier les protocoles de routage. La figure 2.1 présente trois différentes classifications basées sur la structure (ou l'organisation) du réseau, le processus d'établissement des routes, et le fonctionnement du protocole [DP10]. En se basant sur la structure du réseau les protocoles de routage dans les RCSFs peuvent être divisés en trois types : protocoles de routage à plat, hiérarchiques, et géographiques.

Ces protocoles peuvent être aussi classifiés selon le mode d'établissement de la route : réactif, proactif, ou hybride. Finalement, les protocoles de routage diffèrent aussi dans leurs fonctionnements. Les protocoles qui sont basés sur la négociation, le multi-saut, la requête, la QdS, et la Cohérence.

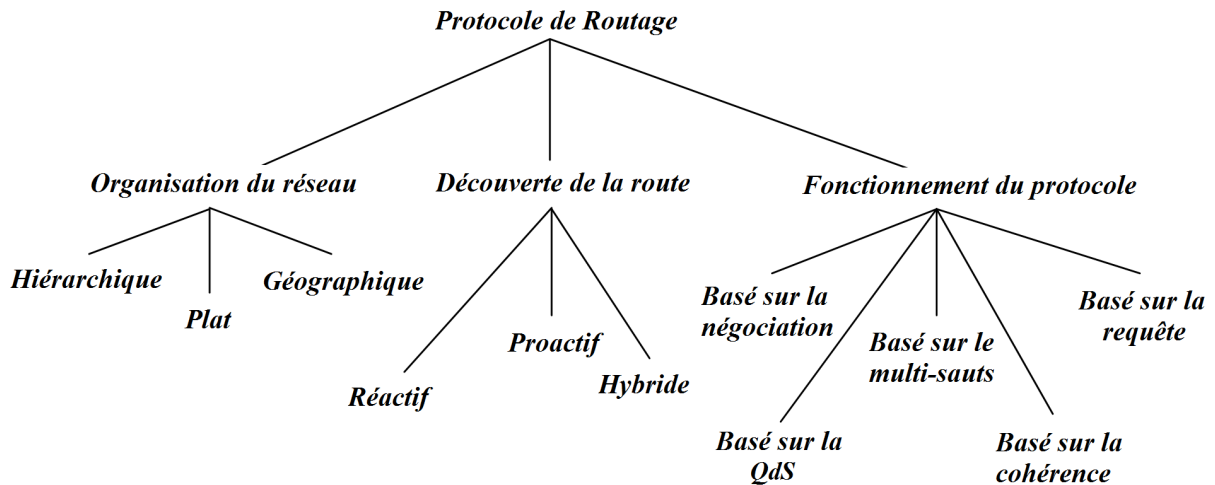


FIG. 2.1 – Catégories des protocoles de routage

2.4.1 Classification selon l'organisation du réseau

Selon l'organisation du réseau, les protocoles de routage peuvent être classifiés en trois classes : les protocoles de routage à plat considèrent tous les nœuds ont le même rôle, au contraire, dans les protocoles de routage hiérarchiques différents nœuds peuvent effectuer différents rôles dans le processus de routage. Les protocoles de routage basés sur la position s'appuient sur les informations de localisation des nœuds pour faire les décisions de routage.

2.4.1.1 Les protocoles à plat

Dans les protocoles de routage à plat, tous les nœuds capteurs jouent le même rôle concernant le routage et tous les nœuds collaborent pour effectuer le processus de routage. Comme le nombre de nœuds est grand, il n'est pas possible d'assigner un identificateur individuel pour chaque nœud ; ceci conduit aux approches de routage centrées-données dans lesquelles une station de base envoie une requête à un groupe particulier de nœuds dans une région et attend la réponse. Donc, dans les approches de routage centrées-données, les données sont plus importantes que les identités des nœuds. Les informations sont référées par l'utilisation des attributs du phénomène. Par

exemple, la requête "*donner moi la température de la région R*" doit être disséminée vers les nœuds capteurs de la région R en même temps, les données générées par la région R doivent être acheminées vers l'utilisateur qui a envoyé la requête.

Les protocoles de routage à plat présentent plusieurs avantages ; ils ont un overhead minimal pour maintenir l'infrastructure, ils ont la capacité de découverte de routes multiples entre les nœuds communicants pour accomplir la tolérance aux fautes. Les solutions centrées-données réduisent la consommation d'énergie dans le réseau par la sélection des routes spécifiées à des intérêts spécifiques à partir de la station de base [AV10].

L'inconvénient majeur de cette approche est que la majorité des informations générées dans les nœuds sont concentrées à la proximité de la station de base ce qui causera l'épuisement rapide des batteries de ces nœuds et donc la déconnexion entre la station de base et les autres nœuds capteurs [AV10].

Les protocoles qui sont basés sur l'architecture à plat sont : flooding, gossiping , SPIN, directed diffusion, energy-aware routing protocol, rumor routing, gradient-based routing (GBR), CADR, COUGAR, ACQUIRE, Shortest PathMinded SPIN (SPMS) , et solar-aware routing, Minimum Cost Forwarding Algorithm (MCFA), Routing Protocols with RandomWalks [AV10] [KB04] [AY05].

Directed diffusion

Dans l'algorithme de SPIN [and99] le flux de trafic est initié par les nœuds capteurs et se termine généralement dans la station de base [AV10]. Cependant, ce type de trafic n'est pas préféré lorsque l'utilisateur (la station de base) demande une information spécifique. Pour cela, *Directed Diffusion* a été proposé pour répondre à ce problème.

L'algorithme *Directed Diffusion* [and00b] est basé sur le paradigme de dissémination de donnée, suivant lequel, la station de base diffuse son intérêt, qui présente une description des tâches affectées à tous les nœuds capteurs du réseau, telle que montré dans la figure 2.2(a). Ces descripteurs englobent des paires d'attributs qui décrivent les tâches requises par la station de base. Chaque nœud capteur stocke dans son cache une entrée d'intérêts, cette entrée contient un champ d'estampille et d'autres champs de gradients, le gradient indique le nœud à partir duquel l'intérêt a été reçu. Ce champ gradient est utilisé pour construire le chemin inverse vers la station de base. Comme l'intérêt est propagé à travers le réseau, les gradients à partir du nœud source de donnée vers la station de base seront déterminés (figure 2.2(b)). Quand des données seront disponibles au niveau du nœud source, il les envoie à la station de base suivant le chemin ayant un gradient fort pour l'intérêt diffusé comme illustré dans la figure 2.2(c) [AV10].

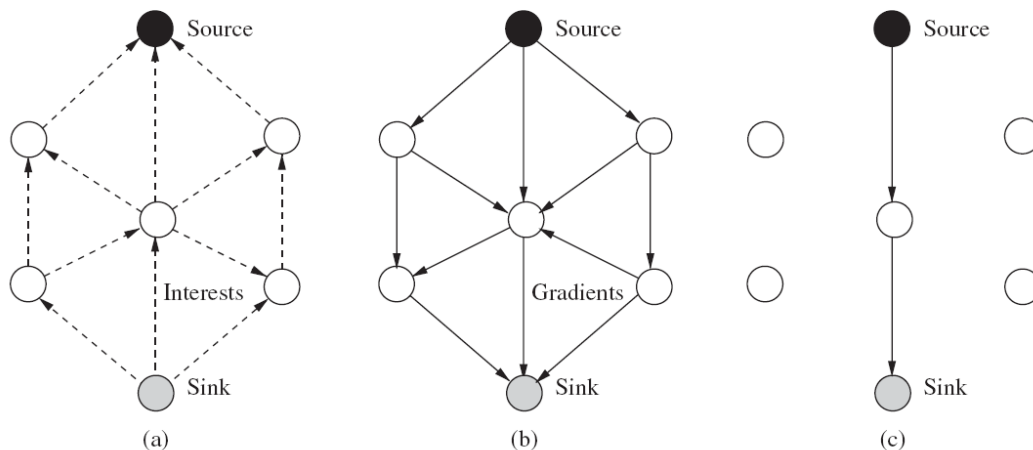


FIG. 2.2 – (a) Propagation de l'intérêt, (b) Etablissement du gradient, (c) Envoi des données.

Les messages d'intérêt indiquent les données demandées dans un temps donné. Par exemple, "à 08 :00 la valeur de la température lue est 21°C".

2.4.1.2 Protocoles hiérarchiques

Dans les protocoles basés sur l'architecture à plat, la majorité des informations générées dans les nœuds sont concentrées près de la station de base [AV10]. Les nœuds qui sont localisés proche de la station de base routent plus d'informations que les autres nœuds dans le réseau. En conséquence, ces nœuds meurent rapidement et produisent une déconnection entre la station de base et le RCSF. Donc, l'architecture à plat limite le passage à l'échelle des protocoles.

Les protocoles de routage hiérarchique sont basés sur le groupage des nœuds en clusters pour traiter les inconvénients des architectures à plat, et surtout le passage à l'échelle et l'efficacité en énergie. L'idée de base du routage hiérarchique est que les nœuds capteurs communiquent directement avec un nœud chef dans leur propre cluster qui s'appelle le cluster-head. Ces cluster-heads, qui peuvent être plus puissants en énergie que les autres nœuds réguliers, sont responsables de la propagation des données vers la station de base. Cette approche peut réduire significativement la communication et la consommation d'énergie dans les nœuds capteurs [AV10].

Le défi dans cette approche réside dans la sélection du cluster-head, la formation des groupes (clusters), et l'adaptation à la dynamique du réseau comme la mobilité ou les échecs des cluster-heads. La figure suivante présente deux variations de cette approche. Lorsque tous les cluster-heads communiquent directement avec la station de base, le

défi est réduit à la formation des clusters. Lorsque les cluster-heads ne communiquent pas directement avec la station de base, alors un protocole de routage doit aussi établir les routes multi-sauts à partir de tous les cluster-heads vers la station de base [DP10].

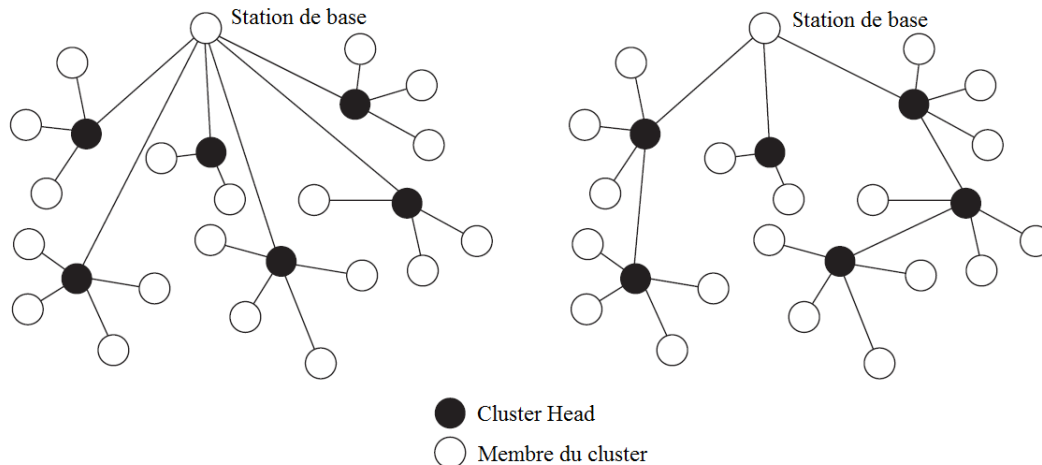


FIG. 2.3 – Clusters avec un seul saut à la station de base (gauche) et clusters avec une connexion à la station de base multisauts (droit).

L'avantage des protocoles de routage hiérarchique et qu'ils facilitent le processus de routage par l'utilisation des clusters, et facilitent aussi l'agrégation des données provenant des différents capteurs ; ce qui réduit la charge du trafic sur le réseau et donc augmente la durée de vie des nœuds et du réseau [DP10].

L'inconvénient major des protocoles de routage hiérarchiques et que les cluster-heads traitent beaucoup de trafic ; ce qui augmente la consommation d'énergie dans ces nœuds. Le processus de formation des clusters augmente aussi la consommation d'énergie [AV10, DP10].

Parmi les protocoles hiérarchiques proposés low-energy adaptive clustering hierarchy (LEACH), power-efficient gathering in sensor information systems (PEGASIS), threshold-sensitive energy-efficient sensor network (TEEN), and adaptive threshold-sensitive energy-efficient sensor network (APTEEN), Energy-aware routing for cluster-based sensor networks [AV10, KB04].

LEACH

LEACH [and00a] est un protocole basé sur les groupes, dans lequel les chefs de groupes élus collectent les données à partir de tous les nœuds capteurs appartenant à

leur groupe, agrègent les données rassemblées par des procédures de fusion, et transmettent ces données directement à la station de base. LEACH change le rôle de chef de groupe périodiquement sur tous les nœuds pour propager la consommation d'énergie dans la communication avec la station de base. Les chefs de groupes élus demeurent ainsi pour une période de temps appelée "round". Au début de chaque "round", chaque nœud détermine la possibilité d'être un chef de groupe pendant la période en cours. S'il décide de l'être, il annonce sa décision à tous les nœuds voisins. Les autres nœuds qui décident de ne pas être un chef de groupe, se joignent à l'un des chefs élus après la réception de leurs décisions. La sélection du chef de groupe adéquat se base sur plusieurs paramètres prédéterminés tels que le rapport signal/bruit (SNR). Un round est décomposé en quatre phases : phase d'annonce, phase de création des groupes, phase d'établissement de l'ordonnancement, et la transmission des données [AV10, KB04].

Le fonctionnement de LEACH, basé sur le groupage, améliore l'efficacité en énergie du RCSF. Pendant la phase de transmission, seulement les chefs de groupe sont actifs tout le temps. Un membre dans un groupe est actif seulement pendant le slot de temps qui lui a été alloué pendant la phase de l'ordonnancement.

2.4.1.3 Protocoles basés sur la localisation

L'information de la position ou de la localisation est très importante dans plusieurs applications dans les RCSFs [AV10]. Comme l'intérêt dans telles applications est le phénomène physique dans l'environnement, il est très important d'associer les observations capturées avec les positions des nœuds capteurs. Pour fournir l'information de la position de chaque nœud capteur, un GPS peut être intégré dans le système de bord embarqué. Comme l'information de la position est nécessaire pour plusieurs applications, il est naturel d'utiliser cette information dans le routage. Les protocoles de routage géographique exploitent l'information de la position de chaque nœud pour fournir un routage efficace. Comme la destination dans les RCSFs est généralement fixée, la position de la destination peut être utilisée pour développer des règles pour router les données près de cette position [AV10].

L'avantage des protocoles basés sur la localisation est que seulement l'information de la position est nécessaire pour les décisions du routage et il n'est pas nécessaire de maintenir une table de routage ou d'établir des chemins de bout en bout entre la source et la destination [DP10].

D'un autre côté, la performance des protocoles de routage géographiques dépend de la connaissance exacte de la position des nœuds. Une erreur dans la détection de la position peut causer des erreurs dans le routage [AV10].

Les protocoles basés sur la localisation sont : Routage Greedy, Routage par Face, GPSR, MECN (Minimum Energy Communication Network), Small MECN, Geographical Forwarding Schemes for Lossy Links, PRADA, Geographic adaptive fidelity (GAF), geographic and energy-aware routing (GEAR) [AV10, KB04].

Le routage géographique dans les réseaux de capteurs sans fil fera l'objet du prochain chapitre.

2.4.2 Classification selon l'établissement des chemins

Les protocoles de routage sont responsables de l'identification ou l'établissement des routes de la source vers la destination voulue. Ce processus de découverte de routes peut être utilisé pour distinguer entre les différents types de protocoles de routage. On distingue trois catégories de protocoles : réactifs, proactifs et hybrides [DP10].

2.4.2.1 Protocoles de routage proactifs

Les protocoles de routage proactifs établissent les chemins avant qu'ils soient demandés. L'avantage principal de cette approche est que les routes sont disponibles n'importe quand elles sont demandées et il n'y a pas de délai encouru pour rechercher des routes. Cependant, cette approche consomme de l'énergie dans l'échange régulier de messages qui ne sont certainement pas tous nécessaires (seules certaines routes seront utilisées par les applications en général) [DP10]. Parmi les protocoles proactifs, on distingue le protocole OLSR.

Optimized Link State Routing (OLSR)

Le protocole OLSR [CGMT04] est un protocole basé sur l'algorithme de l'état de lien [DP10]. Dans cette approche, les nœuds diffusent périodiquement les informations de la mise à jour de la topologie à tous les autres nœuds du réseau, leurs permettant d'avoir une connaissance complète de la topologie du réseau et de déterminer des chemins immédiatement vers n'importe quelle destination.

Pour identifier les voisins et détecter les changements dans le voisinage, un nœud dans OLSR diffuse périodiquement un message HELLO, qui contient l'identité du nœud et une liste de tous les voisins connus de ce nœud. Pour obtenir les informations du réseau entier, les informations de la topologie doivent être inondées à travers tous le réseau [DP10].

2.4.2.2 Protocoles de routage réactifs

Contrairement aux protocoles de routage proactifs, les protocoles de routage réactifs ne maintiennent aucune route jusqu'à ce qu'elles soient demandées et utilisées explicitement. Lorsqu'une route est demandée, le protocole de routage cherche une route vers la destination. En général, cette recherche se fait par l'inondation d'un paquet. L'avantage de cette approche est qu'elle ne génère du trafic de contrôle que lorsqu'il est nécessaire. Le principal inconvénient de cette approche est que l'inondation est un mécanisme coûteux qui fait intervenir tous les nœuds du réseau en très peu de temps et qu'un délai est nécessaire pour l'établissement des routes [DP10]. Parmi les protocoles réactifs, on distingue le protocole AODV.

Ad Hoc On-Demand Distance Vector (AODV)

Le protocole AODV [PR99] s'appuie sur un mécanisme de diffusion *route discovery*, qui est utilisé pour établir les entrées de la table de routage dynamiquement dans les nœuds intermédiaires [DP10].

Le nœud source diffuse vers ses voisins un paquet RREQ qui contient l'adresse de la source et de la destination, un ID de diffusion, un compteur de sauts, et deux nombres de séquence. A la réception d'un paquet RREQ, un nœud qui possède une route vers la destination spécifiée répond par l'envoi d'un paquet RREP vers le voisin d'où le paquet a été reçu. Autrement, le paquet RREQ est rediffusé vers les nœuds voisins intermédiaires. Le paquet RREP contient l'adresse de la source et de la destination, un nombre de séquence de la destination, et le compteur de sauts. Un nœud intermédiaire qui reçoit le paquet RREP propage ce paquet si et seulement si : 1. Il s'agit de la première copie de RREP, 2. Le RREP contient un nombre de séquences de la destination plus grand que le RREP précédant, 3. Le nombre de séquence de la destination est le même que le précédent RREP, mais le nombre de sauts est plus petit. Cela réduit le nombre de RREP voyageant vers la source et assure que l'information de routage pour la route la plus courte (en termes de nombre de sauts) arrive à la source [DP10].

2.4.2.3 Protocoles de routage hybrides

Ces protocoles combinent les deux approches pour tirer profit de leurs avantages. Généralement, le réseau est divisé en deux zones et le principe est d'utiliser une approche proactive pour avoir des informations sur les voisins les plus proches, qui se trouvent au maximum à deux sauts d'un nœud donné. Une approche réactive est utilisée pour au delà de cette zone prédéfinie afin de rechercher des routes.

2.4.3 Classification selon le fonctionnement du protocole

Les protocoles de routage peuvent être classifiés selon leurs fonctionnalités en cinq catégories :

2.4.3.1 Protocoles basés sur la QdS

Le développement de nouvelles applications comportant la communication multi-média nécessite d'autres métriques de performance comme le débit, le délai, et la gigue qui doivent être prises en compte dans la conception du protocole. Donc les exigences de la qualité de service de ces applications doivent être fournies en plus de l'efficacité énergétique dans les RCSFs. Les protocoles de routage basés sur la QdS utilisent des métriques additionnelles autres que la consommation d'énergie pour construire les routes ce qui donne des capacités additionnelles pour les RCSFs. Cependant, ces garanties additionnelles augmentent le coût en termes de la consommation d'énergie [AV10].

Parmi les protocoles basés sur la QdS, on distingue SAR (Sequential Assignment Routing), Minimum Cost Path Forwarding, et SPEED [AV10].

Sequential Assignment Routing (SAR)

SAR [K. 00] est le premier protocole pour les RCSFs qui contient la notion de QdS dans les décisions de routage [AV10, AY05]. C'est une approche multi-sauts orientée table [AV10, AY05]. Le but principal de SAR est de créer des arbres multiples commençant d'un nœud racine dont il est de longueur d'un seul saut de la station de base. Chaque arbre augmente à l'extérieur en évitant les nœuds avec une basse QdS et une réserve basse en énergie. Chaque nœud spécifie deux paramètres pour chaque chemin vers la station de base :

- **Ressource énergétique** : la ressource énergétique d'un chemin est estimée comme le nombre maximum de paquets qui peuvent être envoyés par le nœud, si le nœud a une utilisation exclusive du chemin.
- **Métrique de QdS additive** : chaque chemin est aussi associé à une métrique de QdS additive, qui est reliée à l'énergie et le délai dans chaque lien. Une grande métrique de QdS signifie une basse QdS.

En utilisant les chemins multiples, un nœud peut sélectionner le chemin de routage selon la QdS de chaque chemin et la priorité du paquet. La QdS d'un chemin est déterminée comme une métrique additive, qui est une fonction de l'énergie et du délai. Comme la métrique de QdS augmente, la QdS de la route diminue. Lorsque un nœud

à un paquet à transmettre, il calcule une métrique de QoS pour le paquet, qui est un produit de la QoS du chemin et le niveau de priorité du paquet. Par conséquent, les chemins avec une grande QoS sont utilisés pour les paquets ayant une grande priorité.

2.4.3.2 Protocoles basés sur la négociation

Ces protocoles utilisent des descripteurs de données de haut niveau et des messages de négociation afin d'éliminer la transmission des données redondantes. Les décisions de communication sont aussi prises en se basant sur les ressources disponibles. La négociation permet aux nœuds de prendre des décisions adéquates suivant leurs ressources énergétiques disponibles. Cependant, l'échange des messages entre les nœuds cause la congestion du réseau ainsi qu'une perte additionnelle d'énergie. La famille des protocoles SPIN est un bon exemple de ce type de protocoles [AKK04].

SPIN

Le mécanisme de négociation de SPIN [and99] est effectué à travers trois types de messages : Avertissement (ADV), requête (REQ), et données (DATA). Avant d'envoyer un paquet de données, un nœud avertit ses voisins par la diffusion d'un paquet ADV. Le paquet ADV contient une description du paquet de données DATA à transmettre, qui a une taille plus petite que le paquet de données. Après, si un voisin est intéressé par le paquet ADV, il répond par un message REQ. Finalement, le paquet de données DATA est envoyé au nœud qu'il a demandé.

2.4.3.3 Protocoles basés sur le multi-chemins

Ce sont des protocoles de routage utilisant des chemins multiples plutôt qu'un chemin simple afin d'augmenter la performance du réseau. La tolérance aux fautes (résilience) d'un protocole est mesurée par la probabilité qu'un chemin alternatif existe entre une source et une destination quand le chemin primaire est défaillant. Ceci peut être augmenté en maintenant des chemins multiples entre la source et la destination aux dépens d'une consommation d'énergie plus importante et d'une génération accrue du trafic. Ces chemins alternatifs sont maintenus par l'envoi périodique des messages. Par conséquent, la fiabilité du réseau peut être augmentée en maintenant les chemins alternatifs les plus récents [AKK04]. L'inconvénient de cette approche est qu'un overhead et une perte additionnelle d'énergie viennent s'ajouter pour maintenir ces chemins alternatifs. *Directed diffusion* est un bon exemple de ce type de protocoles [AKK04].

2.4.3.4 Protocoles basés sur la cohérence

En générale, les nœuds capteurs coopèrent entre eux pour le traitement de différentes données inondées sur le réseau. Deux exemples de traitement de données sont proposés ; protocoles basés sur la cohérence et protocoles basés sur la non-cohérence. Dans les protocoles basés sur la non-cohérence, les nœuds traitent les données localement avant de les transmettre aux autres nœuds pour un traitement supplémentaire. Les nœuds qui effectuent le traitement supplémentaire sont appelés les aggregators. Dans le routage cohérent, les données sont acheminées vers les aggregators après un traitement minimum [AKK04]. SWE et MWE sont deux exemples de ce type de protocoles [AKK04].

Single Winner algorithm (SWE)

Dans l'algorithme de SWE [SP00] un seul nœud aggregator est élu pour le traitement complexe. L'élection d'un nœud est basé sur la réserve d'énergie et les capacités de calcul de ce nœud.

2.4.3.5 Protocoles basés sur la requête

Dans ce type de routage, les nœuds destinataires propagent une requête de données de la part de la station de base à travers le réseau et le nœud qui détient ces données les envoie au nœud demandeur en suivant le chemin inverse de la requête. Tous les nœuds détiennent des tables contenant les intérêts sollicités par la station de base. Les nœuds envoient les données correspondantes aux requêtes dès qu'ils les capturent. Les protocoles *Directed Diffusion* et *Rumor Routing* sont deux exemples de ce type de routage [AKK04].

Routage Rumor

Le routage Rumor [D. 02] est une variation de *directed diffusion* qui essaye de combiner les caractéristiques des deux techniques orientée évènement et orientée requête [DP10]. Dans le routage Rumor, chaque nœud maintient une table des ses voisins et une table des évènements qui contient les informations acheminées pour tous les évènements connus [DP10]. Une fois un nœud détecte un évènement, l'évènement est ajouté à cette table et un agent est généré. L'agent est un paquet qui traverse le réseau pour propager l'information concernant cet évènement. Lorsqu'un agent arrive à un nœud, le nœud peut utiliser le contenu de l'agent pour mettre à jour sa table.

2.5 Conclusion

Le routage est en général un composant crucial pour chaque réseau multi-sauts, et plus particulièrement pour les réseaux de capteurs sans fil, à cause de leurs caractéristiques telles que les contraintes de ressources et la liaison moins fiable. Plus spécifiquement, les protocoles de routage doivent opérer efficacement pour éviter l'épuisement des ressources limitées des RCSFs (en particulier l'énergie) et ils doivent être capables de s'adapter aux changements imprévisibles du réseau, tels que le changement dans la topologie et la densité.

Dans ce chapitre, nous avons discuté les différentes approches de classification de protocoles de routage dans les RCSFs ; la classification selon l'organisation du réseau, l'établissement de la route, et le fonctionnement du protocole. L'objectif du prochain chapitre est de faire une étude sur les différents protocoles de routage géographique dans les RCSFs tout en mettant en évidence le fonctionnement, les points forts et les limites de chaque protocole.

Chapitre 3

Les protocoles de routage géographique dans les RCSFs

3.1 Introduction

Les nœuds dans les réseaux de capteurs sans fil, s'appuient sur des batteries ayant des capacités limitées. Donc le critère le plus important pendant la conception des protocoles de communication est d'optimiser la consommation d'énergie pour étendre la durée de vie du nœud capteur et ainsi étendre la durée de vie du réseau. Dans ce contexte, les protocoles de routage basés sur les informations géographiques des nœuds capteurs ont été proposés. Les protocoles de routage géographiques utilisent les informations des positions des nœuds comme des adresses de ces derniers. Cette solution nécessite que les nœuds capteurs connaissent leurs positions géographiques qui peuvent être obtenues en équipant tous les nœuds capteurs par des dispositifs comme GPS ou Galileo. Une solution approximative peut être obtenue en équipant uniquement quelques nœuds capteurs appelés *landmarks* par des GPS et les autres nœuds infèrent leurs positions à partir de ces nœuds (*landmarks*). Cependant, cette solution est très coûteuse et ne peut pas être fiable puisque la réception de GPS peut être obstruée par des obstacles. Une solution moins coûteuse est de considérer le problème d'inférence des positions des nœuds dans lequel aucun nœud ne connaît sa position physique.

En se basant sur les positions des capteurs et celle de la station de base, les algorithmes de routage géographique sélectionnent le meilleur chemin pour acheminer les paquets. On peut distinguer deux approches des protocoles de routage géographique. Dans la première approche les protocoles de routage supposent que chaque nœud connaît sa position physique et en se basant sur ces positions physiques les décisions de routage sont prises. Dans la deuxième approche, les protocoles de routage supposent que les nœuds ne connaissent pas leurs positions physiques et les décisions de routage sont

prises en se basant sur un système de coordonnées virtuelles. Une approche hybride consiste à utiliser des nœuds spécifiques qui connaissent leurs positions physiques appelés ancres (*anchors*) et à partir de ces nœuds les autres nœuds calculent leurs positions virtuelles.

3.2 La localisation

La localisation est le processus par lequel un nœud détermine ses coordonnées spatiales dans un champ donné. Les différentes approches proposées pour résoudre le problème de la localisation diffèrent dans les hypothèses qu'ils mettent sur le réseau et les capacités des nœuds (modèle de propagation du signal, exigences énergétiques, structure du réseau, nature de l'environnement, synchronisation des nœuds, ...).

3.2.1 Les méthodes basées mesure

Dans les méthodes de localisation basées mesure, le processus de localisation peut être divisé en deux étapes. Dans la première étape, un nœud estime ses distances par rapport aux autres nœuds dans son voisinage en utilisant une des technologies de mesure. Dans la deuxième étape, le nœud utilise ces distances pour calculer sa position actuelle.

3.2.1.1 Technologies de mesure

Plusieurs technologies permettent à un capteur de mesurer la distance qui le sépare d'un capteur voisin (ToA, TDoA, RSSI) ou bien de mesurer l'angle qu'il forme avec celui-ci (AoA).

- **Temps d'arrivée (ToA)** : pour calculer la distance en utilisant la technologie ToA (Time of Arrival), le temps de propagation du signal de la source jusqu'à la destination est calculé et en se basant sur la vitesse de propagation du signal, la distance peut être déduite. Dans cette technique, les nœuds du réseau doivent être synchronisés. GPS [ea96] est l'exemple le plus connu qui utilise cette technologie.
- **Différence des temps d'arrivée (TDoA)** : La technologie TDoA (*Time Difference of Arrival*) [SHS01] se base sur la différence des dates d'arrivée d'un ou plusieurs signaux et suppose également que la vitesse de propagation des signaux est connue. Cette technologie s'applique dans les cas suivants :
 - Un émetteur envoie des signaux de natures différentes (par exemple, l'ultrason, l'onde radio, ...) à un récepteur.

- Un récepteur reçoit des signaux d'une même nature d'au moins trois émetteurs.
- Un émetteur envoie un signal reçu par au moins trois récepteurs (dans ce dernier cas une vue globale des signaux sera connue).

Dans chacun des cas, les récepteurs mettent en corrélation leurs informations et en déduisent les distances qui les séparent des émetteurs.

- **Puissance du signal (RSSI) :** La technologie RSSI (*Received Signal Strength Indicator*) [eBN03], utilise un modèle théorique pour transformer la puissance du signal en une distance. RADAR [BP00] est l'un des premiers qui utilise la technologie RSSI.
- **Angle d'arrivée (AoA) :** La technologie AoA (*Angle of Arrival*) [eBN03] calcule l'angle formé entre deux capteurs. Chaque capteur est doté d'antennes orientées de sorte à déduire l'angle qu'il forme avec un voisin lorsque ce dernier lui envoie un signal. Cet angle est reporté par rapport à un axe propre au capteur. Toutefois, un capteur peut être équipé d'une boussole et, dans ce cas, l'angle sera reporté sur un des axes nord, sud, est ou ouest.

3.2.1.2 Méthodes de localisation

La méthode utilisée dans la deuxième étape pour calculer la position actuelle dépend de la technologie de mesure utilisée dans la première étape et elle peut être divisée en trois classes comme suit :

- **Triangulation :** Dans cette classe, le nœud collecte les angles qu'il forme avec les différents voisins. Ensuite, en utilisant des relations géométriques et des propriétés simples, le nœud calcule sa position.
- **Trilateration :** Si le nœud possède les distances par rapport à ses voisins, alors la technique de trilateration peut être utilisée pour calculer sa position. Dans l'espace 2-D, trois voisins avec des positions connues, utilisés comme références, sont suffisants pour calculer la position d'un nœud en résolvant les équations des trois cercles ayant comme centres ces trois voisins et comme rayons la distance entre le nœud et chacun de ces voisins.
- **Multilateration :** la trilateration ne donne pas des positions exactes. Pour cela, la multilateration consiste à utiliser plusieurs voisins pour calculer la position du nœud. Dans l'espace 3-D, quatre voisins au moins sont nécessaires pour calculer la position du nœud.

3.2.2 Les méthodes libres de mesure

Dans les méthodes de localisation libres de mesure, la connectivité entre les nœuds est exploitée pour déterminer les contraintes dans les positions des nœuds. Par conséquent, ces méthodes ne peuvent fournir que des positions estimées des nœuds. Comme dans les méthodes basées mesure, des nœuds ancrés sont utilisés pour fournir une référence pour la localisation.

Il existe deux techniques courantes dans ce type de méthodes. La première consiste à définir des zones contenant les nœuds dont les centres de gravité correspondent à leurs positions estimées. Dans la seconde technique, chaque nœud estime les distances qui le séparent des ancrés et applique la multilatération pour calculer sa position estimée.

3.3 Protocoles basés sur la position physique

Dans cette classe les protocoles de routage utilisent l'information de la position physique des nœuds (coordonnées physiques) comme des adresses des nœuds. Donc pour envoyer un paquet à un nœud destinataire, le nœud source envoie le paquet vers la position où le nœud destinataire se trouve. L'information de la position d'un nœud est obtenue à partir d'un module GPS, Galelio,

3.3.1 Les algorithmes greedy

Les algorithmes de routage dits greedy (avides) basent leur raisonnement sur la trajectoire à vol d'oiseau entre une source et une destination. Le but est alors d'acheminer les paquets de sorte que leurs parcours soient les plus proches de cette trajectoire [Saa09] .

3.3.1.1 MFR (Most Forwarding with Radius)

La stratégie de routage définie par la méthode MFR (Most Forward within Radius), décrite dans [TK84] , [eVL86], suppose que chaque nœud connaît sa position, celles de ses voisins et enfin, celle de la destination. Lorsqu'un capteur doit envoyer un paquet à une destination non voisine, il sélectionne parmi ses voisins celui qui fera suivre le paquet selon le principe suivant : le nœud commence par tracer une droite entre lui et la destination. Ensuite, il trace les projetés orthogonaux des positions des capteurs voisins sur la droite. Celui dont le projeté orthogonal est le plus proche de la destination sera sélectionné pour faire suivre le paquet et appliquera à son tour cette sélection jusqu'à

atteindre la destination. Par exemple, dans la figure 3.1, le nœud source sélectionne le nœud M qui a le projeté orthogonal le plus proche de la destination.

3.3.1.2 Greedy Routing

Dans la méthode *greedy routing* [Fin87], les capteurs connaissent les positions de leurs voisins ainsi que celle de la destination. Elle se différencie de la méthode MFR dans sa stratégie de sélection du voisin devant faire suivre le paquet. Cette stratégie est la plus intuitive pour le routage géographique : elle consiste à sélectionner le voisin qui est le plus proche en distance euclidienne de la destination. Ce procédé est répété jusqu'à ce que la destination soit atteinte. Pour reprendre l'exemple en figure 3.1, le nœud source sélectionnera le capteur G puisque la distance euclidienne entre G et la destination est la plus petite.

3.3.1.3 Compass Routing

L'idée générale de la méthode *Compass routing* [UKS99] (appelée aussi DIR) diffère de celle des deux méthodes précédentes où le paquet est transféré au voisin qui se rapproche le plus de la destination. Dans la méthode Compass Routing, la stratégie de routage consiste à faire suivre le paquet au voisin qui minimise l'angle formé par lui même, l'expéditeur et la destination. En d'autres termes, il s'agit de sélectionner le voisin dont la direction est la plus proche de celle de la destination. Dans l'exemple de la figure 3.1 le nœud source choisira le voisin C pour faire suivre son paquet à la destination, puisque C est le plus proche de la ligne st qui relie la source et la destination. Ce processus se répète pour chaque nœud intermédiaire jusqu'à atteindre la destination.

3.3.1.4 NFP (Nearest Forward Progress)

La stratégie de routage définie par la méthode NFP (Nearest Forward within Progress) [Sto02, MWHE01], suppose que chaque nœud connaît sa position, celles de ses voisins et enfin, celle de la destination. Dans cette méthode, le nœud choisit, parmi les nœuds voisins les plus proches de la destination, son plus proche voisin comme prochain nœud. Dans la figure 3.1, le nœud source sélectionne le nœud N qui est son plus proche voisin.

3.3.1.5 GEDIR

La méthode GEDIR [SL01a] est une stratégie greedy avec des étapes de retour arrière. Lorsqu'un paquet arrive à un trou, le paquet est retourné arrière vers le nœud

précédant qui appliquera l'algorithme greedy une autre fois en excluant de la sélection ces derniers nœuds visités qui conduisent à des trous.

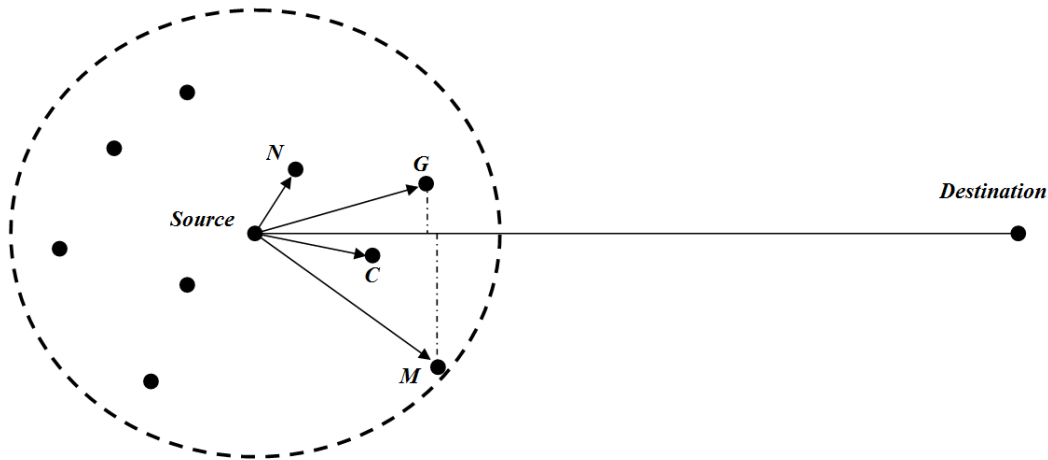


FIG. 3.1 – Les différentes méthodes de routage Greedy.

L'avantage de ces méthodes est leur simplicité et qu'elles calculent des routes qui sont généralement proches du plus court chemin (chemin optimal) avec peu d'overhead dans la maintenance des structures de routage auxiliaires [FGG⁺05].

Cependant, plusieurs remarques doivent être faites pour ces méthodes. La première concerne la configuration du réseau : l'efficacité de ces méthodes, en termes de taux d'acheminement des paquets (garantie de livraison), est liée à la densité en nœuds du réseau. Ainsi, en forte densité, les messages arriveront à destination en grande majorité. En revanche, dans des réseaux à faible densité, la présence de " trous " augmente et, de ce fait, entraîne une forte baisse du taux d'acheminement des paquets [Saa09]. Deuxièmement, dans ces méthodes le nœud voisin le plus proche de la destination est utilisé fréquemment, tandis que les autres nœuds sont rarement utilisés. Cette distribution de la charge inégale conduit à l'épuisement rapide des nœuds sélectionnés fréquemment ce qui engendre des trous dans le réseau [AAEI10].

3.3.2 Routage basé sur graphe planaire

Le routage basé sur le graphe planaire est une stratégie de routage géographique qui est capable de surmonter le problème du routage greedy (le problème des trous). On dit qu'un graphe est planaire si on peut le dessiner dans le plan de sorte que ses arêtes ne se croisent pas [Jun08]. Le graphe planaire forme un ensemble de faces comme illustré dans la figure 3.2 et le routage se fait d'une face à une autre en utilisant la règle de la main droite (ou de la main gauche) [eUM76].

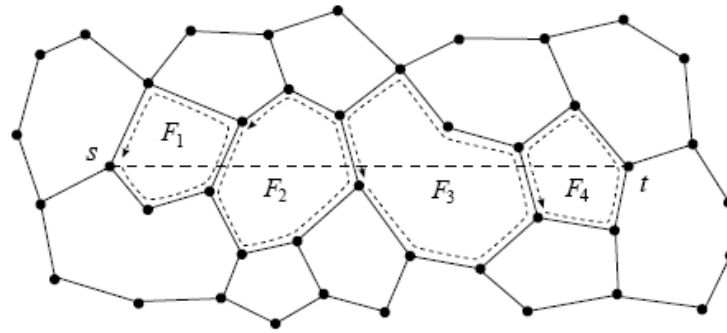


FIG. 3.2 – Graphe planaire.

3.3.2.1 Construction d'un graphe planaire

En général, un graphe d'un réseau de capteur sans fil arbitraire n'est pas un graphe planaire. Pour cela, avant que le routage basé sur le graphe planaire puisse être mis en œuvre, un mécanisme de construction du graphe planaire doit être appliqué d'avance. Dans ce qui suit, nous allons présenter les trois techniques les plus utilisées pour la construction d'un graphe planaire.

- **Le graphe de Gabriel (GG)** : Un arc (u,v) existe entre les deux nœuds u et v s'il n'y a aucun autre nœud à l'intérieur du cercle ayant le diamètre uv . De façon formelle :

$$\forall w \neq u, v : d^2(u, v) < [d^2(u, w) + d^2(v, w)]$$

Dans la figure 3.3 (a) pour que l'arc (u,v) soit inclus, le cercle ombré ne doit contenir aucun nœud w .

- **Le graphe des voisins relatifs (RNG)** : Un arc (u,v) existe entre les deux nœuds u et v si la distance entre eux $d(u,v)$ est inférieure ou égale à la distance entre chaque autre nœud w . De façon formelle :

$$\forall w \neq u, v : d(u, v) < \max[d(u, w) + d(v, w)]$$

Dans la figure 3.3 (b), pour que l'arc (u,v) soit inclus, la sphère ombrée ne doit contenir aucun nœud w .

- **Triangulation de Delaunay (DT)** : La triangulation de Delaunay contient en général tous les triangles satisfaisant la condition suivante : le cercle passant

par les points du triangle ne contient aucun autre nœud. Dans la figure 3.3 (c) le cercle circonscrit à ces nœuds (u,v,z) ne doit contenir aucun autre nœud w .

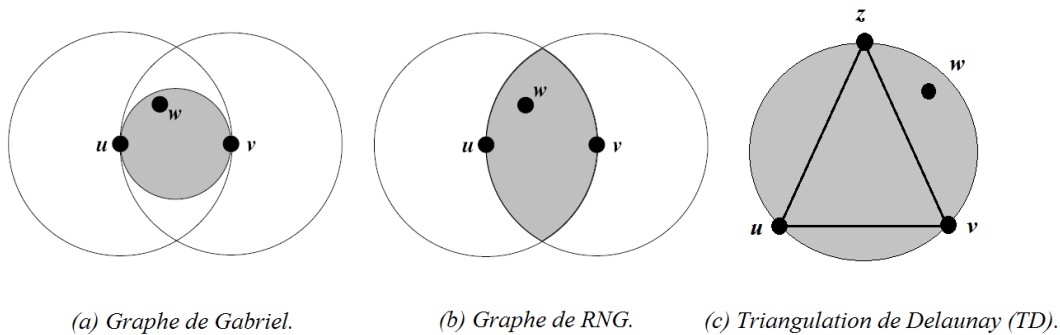


FIG. 3.3 – Construction du graphe planaire.

3.3.2.2 Routage par faces (Face routing)

Une fois le graphe planaire est construit, lorsqu'une source envoie un paquet à une destination, celui-ci sera transmis le long des faces et effectue le changement de face avant de traverser la droite passant par la source et la destination (la ligne st). Dans chaque parcours de la face, un nœud u vérifie si l'arc vers le nœud prochain (u, u') croise la droite passant par la source et la destination (la ligne st). Si c'est le cas, le nœud u change la face et continue à traverser la face suivante. Chaque face est traversée en utilisant la technique de la main droite [eUM76].

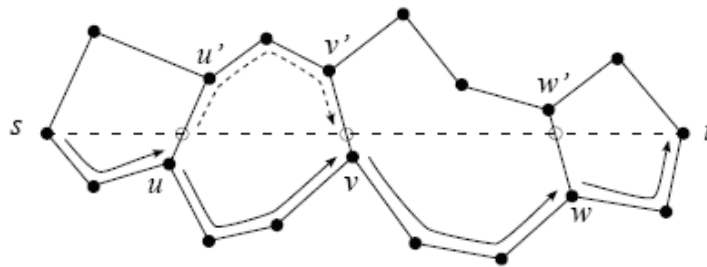


FIG. 3.4 – Routage par face, les changements de faces surviennent dans les nœuds u , v , w .

Cette technique de routage garantit l'arrivée des paquets mais ces derniers peuvent emprunter des chemins très coûteux en énergie avec un délai de transmission de bout en bout plus grand.

3.3.2.3 GFG (Greedy-Face-Greedy)

La technique de routage GFG [BMSU01] consiste à combiner un algorithme *greedy* avec le routage par faces. Le principe est simple : initialement, le paquet est routé avec l'algorithme *greedy* mais, en présence d'un " trou ", la technique bascule en mode routage par faces pour le contourner puis revient dès que possible en mode *greedy*. Cette technique a le mérite de combiner les avantages des algorithmes *greedy* et ceux du routage par faces, mais elle ne prend pas en compte la consommation d'énergie.

3.3.2.4 Power Progress

La méthode Power Progress proposée dans [KNS06b] est une extension de la méthode GFG qui cette fois prend en compte la modélisation de la consommation d'énergie. En effet, la consommation d'énergie suit une loi de puissance par rapport à la portée d'émission : si d est la portée d'émission, la consommation d'énergie est supérieure à d^a avec $a \geq 2$ selon le modèle utilisé.

Dans cette méthode, les capteurs connaissent les positions de leurs voisins à deux sauts. Lorsqu'un capteur u veut transmettre un paquet à un capteur voisin v sélectionné grâce à un algorithme *greedy*, on vérifie s'il existe un capteur w voisin de u et voisin de v tel que la consommation d'énergie soit inférieure si le paquet suit le chemin uwv plutôt que le chemin direct uv . Si tel est le cas, u enverra le paquet à w qui de façon récursive agira selon le même procédé.

Cette méthode est la plus adaptée aux réseaux de capteurs dans le cas où ces derniers sont déployés de façon aléatoire sans connaissance préalable de l'environnement.

3.3.2.5 GPSR

Le protocole de routage GPSR (Greedy Perimeter Stateless Routing) proposé par Brad Karp et H. T. Kung dans [KK00], présente une autre solution au problème de trou dans le routage Greedy. Les auteurs dans [KK00] proposent d'utiliser le routage *greedy* et lorsque le paquet arrive à une région où le routage *greedy* est impossible le protocole recouvre par le routage autour du périmètre de la région. La figure 3.5 est un exemple de cette situation.

Dans la figure 3.5, traverser le cycle (x, w, v, D, z, y, x) par la règle de la main droite revient à naviguer autour du trou, spécifiquement, vers les nœuds proches de la destination que x (dans ce cas la destination elle-même est incluse). On appelle la séquence d'arcs traversés par la règle de la main droite un périmètre [KK00].

Dans GPSR l'entête du paquet contient un champ indiquant si le paquet est en

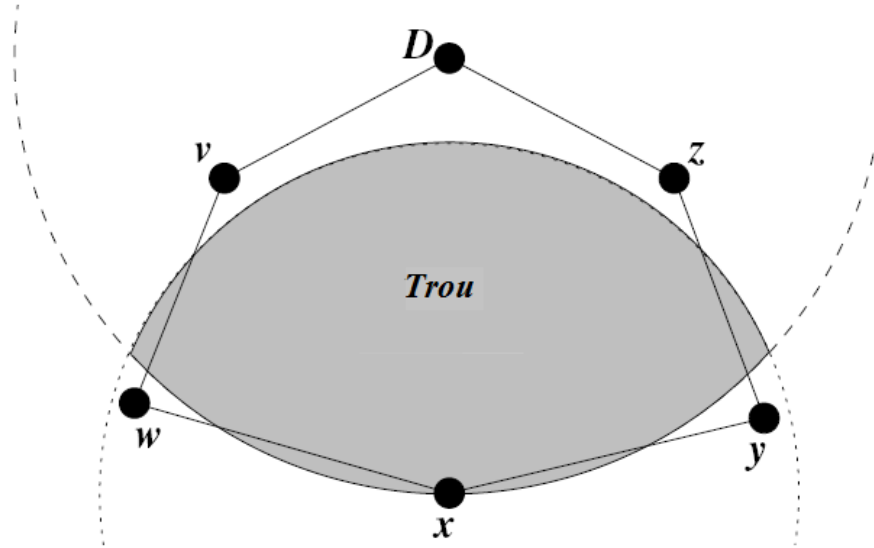


FIG. 3.5 – Exemple d'un trou.

mode *greedy* ou bien en mode *perimeter*. Tous les paquets sont marqués initialement à l'origine en mode *greedy*. Lorsqu'un nœud reçoit un paquet en mode *greedy*, il cherche dans sa table de routage un voisin plus proche de la destination. S'il n'existe aucun nœud plus proche de la destination, le nœud marque le paquet en mode *perimeter*. GPSR achemine les paquets en mode *perimeter* en traversant les faces d'un graphe planaire simple en utilisant la règle de la main droite. Lorsqu'un paquet entre en mode *perimeter*, GPSR enregistre dans le paquet la position L_p où le mode *greedy* a échoué. A la réception d'un paquet en mode *perimeter*, GPSR compare la position L_p avec les positions des prochains sauts, le paquet est retourné en mode *greedy* si la distance entre le prochain saut et la destination est inférieure à la distance entre L_p et la destination.

3.3.2.6 GOAFR

F. Kuhn, R. Wattenhofer, A. Zollinger ont proposé dans [KWZ03] une autre technique de routage qui utilise les deux techniques ; le routage *greedy* et le routage par face. GOAFR utilise le routage *greedy* le plus possible et lorsque le paquet arrive à un trou, l'algorithme recourt par le routage par face.

- **AFR (Adaptive Face Routing)** : Dans son chemin autour d'une face, l'algorithme garde la trace des points où il croise la ligne st qui connecte la source s et la destination t . Après le parcours de la face, l'algorithme retourne à une de ces intersections la plus proche de la destination où il commence à explorer la face suivante la plus proche de la destination.

- *OAFR (Other Adaptive Face Routing)* : Les auteurs dans [KWZ03] ont expliqué l'algorithme de OAFR en trois étapes : OFR, OBFR, et OAFR.

OFR : L'algorithme OFR diffère du routage par face dans le changement de la face. Au lieu de changer vers la face suivante dans la meilleure intersection de la frontière de la face avec la ligne st , OFR retourne au point le plus proche de la destination. Dans la figure 3.6 le routage par face commence dans s , explore la face F_1 , il trouve P_1 sur st , il explore F_2 , trouve P_2 , change à la face F_3 avant d'arriver à la destination t . Au contraire, OFR trouve P_3 , le point sur la frontière de F_1 le plus proche de la destination t , il continue à explorer F_4 où il trouve P_4 , et finalement atteint t via F_5 .

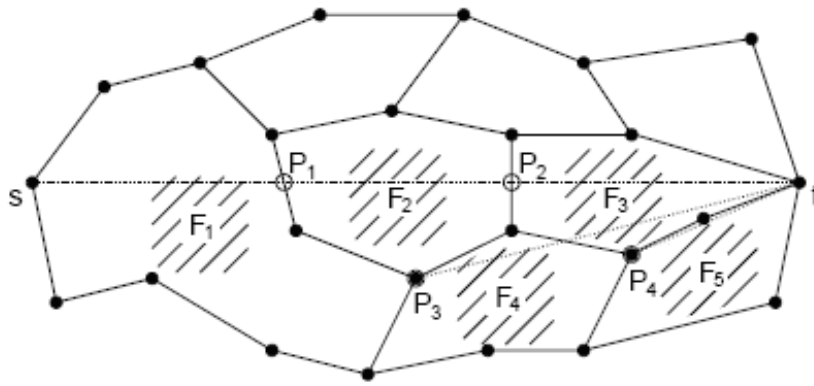


FIG. 3.6 – Other face routing.

L'algorithme OFR

1. Commencer dans s à explorer la face F qui contient la ligne st dans l'environnement immédiat de s .
2. explorer complètement la face F en utilisant la règle de la main droite.
3. après l'exploration de F , avancer au point p le plus proche de la destination t . Changer à la face qui contient la ligne pt et continuer avec l'étape 1. Répéter ces deux étapes jusqu'à atteindre la destination.

L'algorithme Other Bounded Face Routing (OBFR) :

Soit c la distance Euclidienne d'un chemin optimal et soit ε l'ellipse avec foci s et t .

1. Etape 1 de OFR
2. Etape 2 de OFR, mais ne quitter pas ε : lorsque ε est atteint, continuer l'exploration de la face courante F dans le sens opposé. L'exploration de F est terminée lorsque ε est atteint pour la deuxième fois.

3. Etape 3 de OFR.

L'algorithme Other Adaptive Face Routing (OAFR) :

1. Initialiser ε avec $\varepsilon(2|st|)$.
2. Démarrer OBFR avec ε .
3. Si la destination n'a pas été atteinte, doubler la longueur de ε et aller à l'étape 2.

GOAFR : Greedy OAFR

Le routage greedy est utilisé le plus possible et dans le cas d'un trou GOAFR utilise OAFR pour s'en sortir.

1. Initialiser ε avec $\varepsilon(2|st|)$ et commencer dans s .
2. Effectuer le routage greedy jusqu'à soit atteindre la destination soit arriver à un trou n_m . Si l'étape suivante conduit en dehors de ε , doubler la longueur de l'axe major de ε . Si un trou est atteint procéder avec l'étape 3 en commençant par n_m .
3. Exécuter OAFR sur la première face uniquement. Doubler la longueur de l'axe major de ε autant que nécessaire.
4. Terminer si OAFR atteint la destination. Si OAFR détecte une déconnexion, reporter ça à la source s . Sinon continuer avec l'étape 2 sur le nœud trouvé par OAFR le plus proche de la destination.

Dans la figure suivante, GOAFR procède dans le mode greedy jusqu'à arriver au trou n_1 . L'algorithme bascule au routage par face et explore la frontière de la face F pour trouver n_2 , le nœud le plus proche de la destination t sur la frontière de F. GOAFR retourne au mode greedy et finalement atteint t .

L'avantage major du routage basé sur graphe planaire est qu'il augmente le taux d'acheminement des paquets (c-à-d il garantit la livraison des paquets). Cependant, les protocoles de routage basés sur graphe planaire génèrent un trafic de contrôle très élevé, à cause du processus de construction du graphe planaire et de la maintenance des informations du graphe planaire sur les nœuds, ce qui augmente la consommation d'énergie (aucun des protocoles présentés ci-dessus ne prend en compte la consommation d'énergie pendant le routage). De plus, ces méthodes augmentent le délai de transmission de bout en bout à cause du mécanisme de recouvrement appliqué pour router au tour des trous.

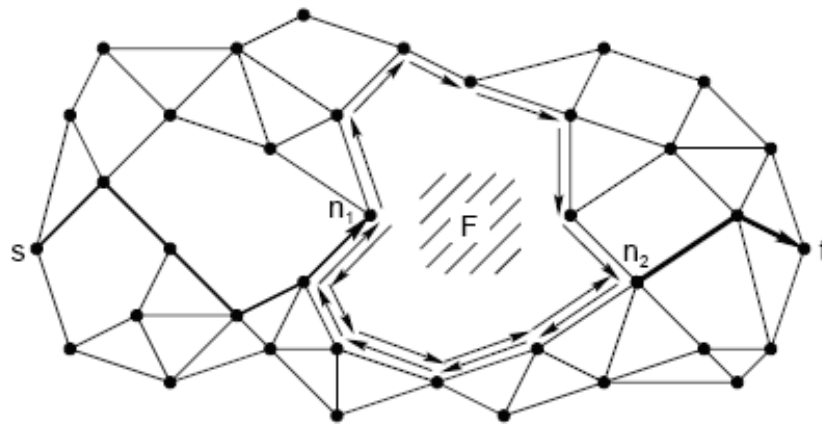


FIG. 3.7 – GOAFR.

3.3.3 Algorithmes hiérarchiques

Les protocoles basés sur l'utilisation d'un mécanisme hiérarchique sont proposés typiquement pour le routage des paquets dans les réseaux mobiles ad hoc à large échelle. Parmi les protocoles utilisant ce principe, on trouve Terminode [BGB01] et GRID [LTS01].

3.3.3.1 GRID

Le protocole GRID proposé dans [LTS01] est un protocole réactif. Dans GRID la surface géographique est partitionnée en grilles logiques de deux dimensions. Chaque grille est un carré de dimension $d \times d$ et est numérotée suivant les coordonnées (x, y) conventionnelles. Dans chaque grille un nœud est élu comme une passerelle et le routage s'effectue grille par grille à travers les nœuds passerelles des grilles. La responsabilité des nœuds passerelles consiste en :

- L'acheminement des paquets de découverte de route vers les grilles voisines.
- La propagation des paquets de données vers les grilles voisines.
- La maintenance des routes qui passent par la grille.

Les auteurs dans [LTS01] ont discuté six possibilités pour le choix de d étant donné r comme le rang de transmission des nœuds.

1. d est plus grand que r : le nœud passerelle ne peut pas atteindre les places à l'extérieur de la grille et donc la communication passerelle-à-passerelle ne peut pas être réalisée.

2. $d=r$: c'est la valeur maximale de d pour que les passerelles des grilles voisines puissent communiquer.
3. $d=2r\sqrt{2}$: c'est la valeur maximale de d pour qu'une passerelle située au centre d'une grille puisse communiquer avec ses 4 passerelles des grilles voisines.
4. $d = \sqrt{2}r/3$: c'est la valeur maximale de d pour qu'une passerelle située au centre d'une grille puisse communiquer avec ses 8 passerelles des grilles voisines.
5. $d= r/2\sqrt{2}$: c'est la valeur maximale de d tel qu'une passerelle située dans n'importe quelle position de la grille est capable de communiquer avec n'importe quelle passerelle de ses 8 grilles voisines.
6. d est plus petit : ça signifie qu'il y a peu de nœuds mobiles résidant dans une grille. Ce qui correspond à la chance pour un nœud mobile d'être une passerelle est grande.

Les auteurs de GRID ont adopté le protocole AODV pour développer GRID. Lorsqu'un nœud source S en position LS a besoin d'une route vers une destination D en position LD, il diffuse une requête RREQ ($S, s\text{-seq}, D, d\text{-seq}, id, rang$) pour chercher une route vers la destination. Lorsqu'un nœud passerelle reçoit cette requête, il vérifie s'il est dans la surface spécifiée dans le champ ' $rang$ ', si tel est le cas, il met un pointeur inverse vers la grille à partir de laquelle il a reçu la requête et rediffuse le paquet de la requête, sinon il ignore la requête. Lorsque la destination D reçoit RREQ, elle répond par un paquet RREP ($S, D, d\text{-seq}$) vers la source S . Ce paquet est un paquet unicast et suit les pointeurs inverses qui ont été établis pendant la propagation de RREQ. A la réception d'un paquet RREP, une passerelle ajoute une entrée à sa table de routage contenant une route vers D via la grille à partir de laquelle elle a reçu RREP.

Le mécanisme de diffusion utilisé par le protocole GRID augmente le taux de livraison des paquets. Cependant, ce mécanisme a un impact significatif sur la consommation d'énergie à cause du mécanisme de diffusion utilisé. De plus, GRID augmente le délai de transmission de bout en bout à cause du temps perdu dans la recherche d'un chemin entre la source et la destination.

3.3.3.2 Terminode

Le protocole de routage Terminode [BGB01] est une combinaison de deux protocoles TLR (*Terminode Local Routing*) et TRR (*Terminode Remote Routing*). TLR est un mécanisme qui permet d'atteindre une destination dans le voisinage d'un nœud et n'utilise pas l'information de position pour prendre les décisions de routage. TRR est utilisé pour envoyer les données aux destinations distantes et utilise les informations des positions géographiques. La méthode de routage de TRR par défaut est GPF (*Geodesic Packet Forwarding*). GPF est essentiellement une méthode greedy qui achemine

le paquet vers la position de la destination jusqu'à ce que la destination soit atteinte. GPF ne fonctionne pas bien si la source et la destination ne sont pas bien connectées. Si la source estime que GPF ne peut atteindre la destination il utilise les chemins ancrés (*Anchored Paths*). Un chemin ancré n'est pas une liste de nœuds à visiter pour atteindre la destination. Un chemin ancré est une liste de points géographiques fixés appelés ancres (*anchors*). Pour acheminer un paquet à travers un chemin ancré, TRR utilise la méthode appelée AGPF (*Anchored Geodesic Packet Forwarding*). Un nœud source ajoute au paquet un vecteur d'ancres. Lorsque un nœud relais reçoit un paquet avec un vecteur d'ancres, il vérifie s'il est proche de la première ancre dans le vecteur, si c'est le cas, il supprime la première ancre et envoie le paquet vers la prochaine ancre ou bien vers la destination finale en utilisant GPF.

Pour acheminer les paquets à travers les chemins ancrés, la source a besoin de les acquérir en utilisant des méthodes de découverte de chemin. Les auteurs dans [BGB01] ont présenté deux méthodes : FAPD (*Friend Assisted Path Discovery*), et GMPD (*Geographic Maps-based Path Discovery*). FAPD permet à la source d'apprendre le chemin ancré en utilisant les nœuds où la source connaît déjà comment router les paquets. GMPD suppose que tous les nœuds dans le réseau ont une connaissance complète ou partielle de la topologie du réseau.

L'utilisation d'une méthode de découverte de chemin engendre un trafic supplémentaire dans le réseau, ce qui augmente la consommation d'énergie. De plus, les chemins empruntés par les paquets peuvent ne pas être optimaux.

3.3.4 Algorithmes basés sur l'inondation dirigée

Les approches basées sur l'inondation dirigée ont été proposées dans le but de limiter la diffusion des requêtes de découverte d'itinéraire. Les requêtes sont diffusées seulement aux nœuds se trouvant dans la zone du nœud destinataire.

3.3.4.1 LAR

Pendant la découverte d'un itinéraire, le protocole LAR [KV98] ne définit pas un mécanisme de localisation des nœuds, mais il utilise des informations sur la position, considérée connue à l'avance, des nœuds. L'objectif est de limiter la diffusion des paquets de découverte de routes. Deux approches ont été proposées. Dans la première approche, deux types de zones ont été définis (voir figure 3.8 (a)) : *expected zone* (EZ) et *request zone* (RZ). L'EZ est définie comme la zone où devrait se trouver le nœud du point de vue de la source. Cette zone, à l'instant t_1 , est calculée sur la base de sa position antérieure connue par le nœud source à l'instant t_0 et de la vitesse moyenne v du mouvement

du nœud destinataire. En se basant sur ces informations, le nœud source définit cette zone comme un cercle de centre (X_d, Y_d) et de rayon $R = (t_1 - t_0)v$. La RZ est définie comme étant le plus petit rectangle comprenant la position actuelle du nœud source et de la zone EZ. La RZ contient généralement la zone EZ. Dans cette approche, lorsqu'un nœud S souhaite échanger des informations avec un nœud D, il calcule la zone EZ dans laquelle le nœud D peut être localisé. Si aucune information n'est disponible, LAR est réduit à de simples inondations. Si des informations de localisation sont disponibles, la zone RZ est définie. Durant la phase de découverte d'une route, les informations concernant la RZ sont jointes à la requête et seuls les nœuds appartenant à cette zone la diffuseront (ex. : les nœuds I et K dans la figure 3.8(a)). Cela permet de diminuer la charge des paquets de contrôle diffusés dans le réseau. A la réception du paquet, par le nœud destinataire, il génère une réponse en indiquant sa position actuelle, l'heure et sa vitesse moyenne.

Dans la deuxième approche, le nœud source calcule la distance $dist_S$ qui le sépare du nœud destinataire. Cette distance ainsi que les coordonnées (X_d, Y_d) de la destination D sont ensuite incluses dans le paquet de découverte d'itinéraires. Ce dernier est envoyé par la suite aux nœuds voisins (voir figure 3.8 (b)). Quand le nœud I reçoit ce paquet, il calcule la distance $dist_I$ qui le sépare de la destination D , et la compare avec la distance $\alpha \times dist_S + \beta$, où $dist_S$ est la distance contenue dans le paquet, les paramètres α et β sont des paramètres du système. Dans le cas où la distance calculée est inférieure ou égale à la distance $dist_I$, le nœud I rediffuse le paquet reçu. Lors de l'envoi, le nœud met à jour le champ de distance, $dist_I$, avec sa propre distance qui le sépare du nœud destinataire. Dans le cas contraire, le nœud I ne rediffuse pas le paquet. Quand un nœud J reçoit le paquet du nœud I pour la première fois, il procède de la même manière que le nœud I . Si aucune réponse de route n'est reçue après un certain temps (TTL par exemple), le nœud source rediffuse une nouvelle requête de découverte de route.

3.3.4.2 DREAM

Dans le protocole DREAM [BCSW98], le nœud source S diffuse le paquet de découverte à tous les voisins immédiats (se trouvant à un seul saut) qui se trouve dans la direction du nœud destinataire D. Pour déterminer cette direction, le nœud S calcule la région susceptible d'englober le nœud D, appelée EZ, expected zone (voir figure 3.9). L'EZ est délimitée par un cercle autour de la position du nœud D. Puisque l'information à propos de cette position peut changer, le rayon r de cette région est calculé de la manière suivante : $r = (t_1 - t_0)v_{max}$, où t_1 est le temps actuel ; t_0 est le temps associé à la dernière information que S possède à propos du nœud D, et v_{max} est la vitesse maximale d'un nœud dans le réseau. Rappelons que la position concernant le nœud D est déjà connue et stockée dans sa base de données à travers le processus de

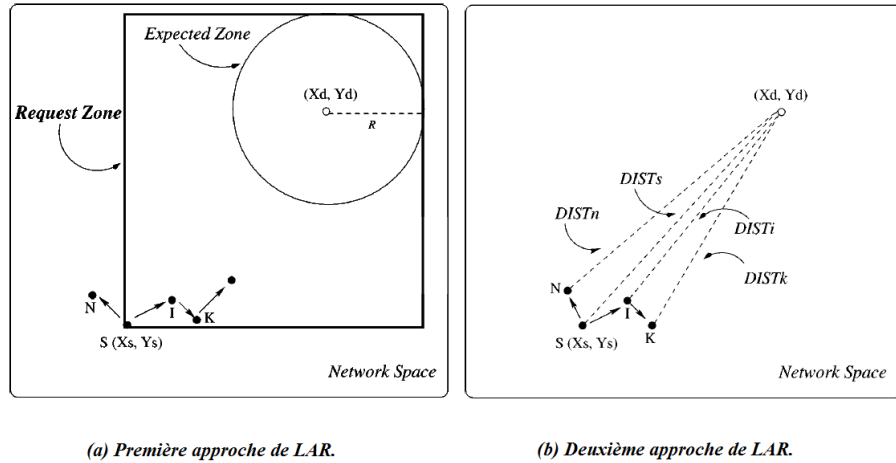


FIG. 3.8 – Comparaison des deux schémas de LAR.

localisation utilisé par DREAM. Etant donné la région EZ, la direction vers le nœud D est définie par la ligne entre S et D et de l'angle φ (voir figure 3.9). Les nœuds voisins répètent cette procédure en utilisant les informations sur la position de D. Si un nœud ne dispose pas d'un voisin situé à un seul saut dans la direction du nœud destinataire, une procédure de recouvrement doit être démarrée.

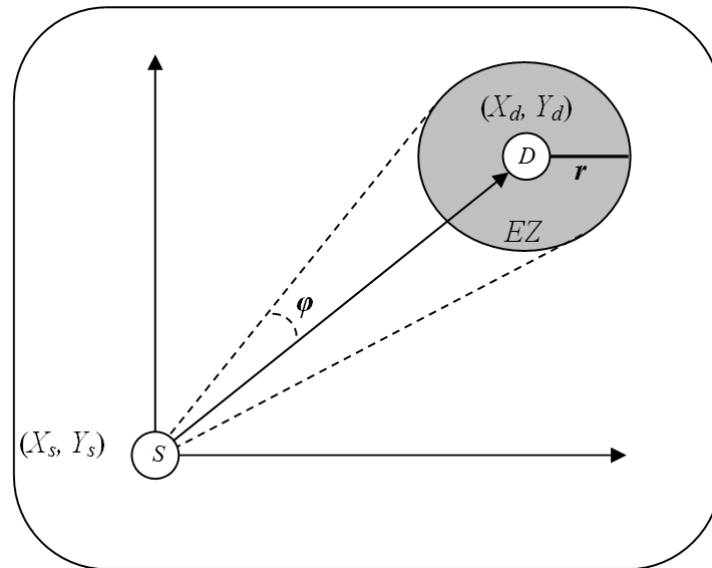


FIG. 3.9 – Exemple d'expected zone dans DREAM.

La technique de limiter la diffusion à une zone utilisée dans LAR et DREAM réduit la consommation d'énergie mais elle ne garantit pas la livraison des paquets car il

est possible qu'aucun chemin n'existe dans la zone de diffusion spécifiée. Dans ce cas, un mécanisme de recouvrement doit être utilisé ce qui augmente considérablement la consommation d'énergie .

3.4 Protocoles basés sur des coordonnées virtuelles

Les protocoles de routage géographiques basés sur la connaissance de la position physique des nœuds nécessitent que chaque nœud capteur connaisse sa position physique obtenue par un GPS ou un autre système de localisation qui sont généralement très coûteux. Pour cela, plusieurs solutions pour fournir un système de coordonnées virtuelles ont été proposées.

3.4.1 Routage greedy basé sur des coordonnées virtuelles

Dans cette section nous allons présenter des systèmes de construction des coordonnées virtuelles pour appliquer le routage greedy.

3.4.1.1 VCR (Virtual Coordinates based Routing)

Le protocole VCR proposé dans [CWLY06] adopte un nouveau critère basé sur des coordonnées virtuelles converties à partir des coordonnées absolues. L'intersection de la ligne qui connecte le nœud courant et la station de base et le cercle (O, R) , tel que O est le nœud courant, R est le rang de transmission, s'appelle la position stratégique. En se basant sur les coordonnées virtuelles, la distance d'un nœud voisin à la position stratégique est calculée. VCR essaye de sélectionner le nœud le plus proche de la position stratégique.

Obtention des coordonnées virtuelles

Dans la figure 3.10, o est l'origine, h est le nœud qui a initié la sélection du prochain nœud. Les coordonnées absolues du nœud $h(x_h^0, y_h^0)$ sont incluses dans le paquet transmis par h , de cette façon, le nœud i connaît la position de h , sa position, celle de la station de base. Dans [CWLY06], les coordonnées virtuelles d'un nœud sont définies comme les coordonnées dans le système de coordonnées virtuelles de deux-dimensions où le nœud source est l'origine, et l'axe X est la ligne entre le nœud source et la station de base. Dans la figure 3.10, les coordonnées virtuelles de i sont (x_i, y_i) et qui sont calculées comme suit :

$$x_i = \cos(\theta) \cdot (x_i^o - x_h^o) + \sin(\theta) \cdot (y_i^o - y_h^o),$$

$$y_i = \sin(\theta) \cdot (x_i^o - x_h^o) + \cos(\theta) \cdot (y_i^o - y_h^o),$$

$$\theta = \arctan\left(\frac{y_i^o - y_h^o}{x_i^o - x_h^o}\right)$$

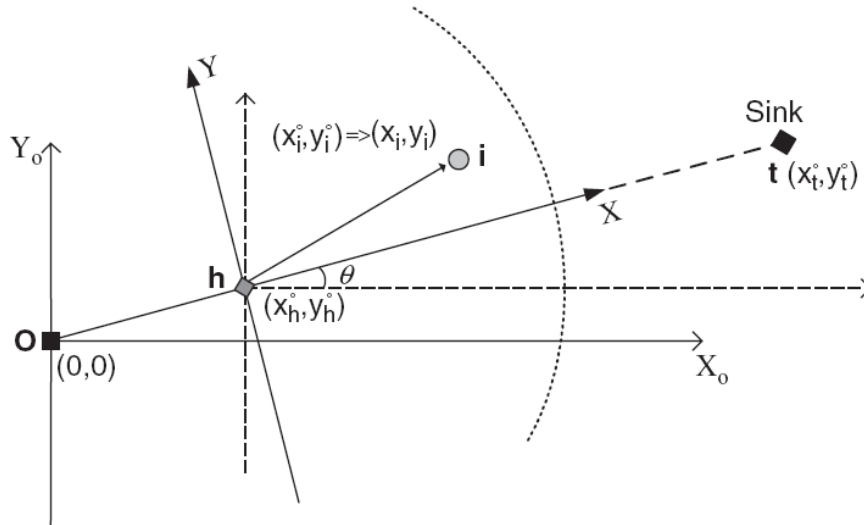


FIG. 3.10 – Obtention des coordonnées virtuelles.

Décisions de routage basées sur des CV

L'idée de base de routage dans VCR consiste à choisir le nœud le plus proche de la position stratégique. Un seuil DT ($0 < DT < R$) est mis pour limiter la surface de sélection. Soit ΔD la distance entre un nœud i et la position stratégique donc :

$$\Delta D = \sqrt{(x_i - r)^2 + (y_i)^2}$$

D est le critère de base de VCR. Pour ajuster l'impacte de y_i sur l'éligibilité de voisin à être sélectionné comme prochain nœud, un critère de poids est dérivé dans :

$$\Delta D_\alpha = \sqrt{(x_i - r)^2 + \alpha(y_i)^2}$$

Ce protocole présente deux inconvénients majeurs. Le premier est qu'il ne garantit pas la livraison des paquets, dans le cas où il y a un trou le paquet sera éliminé. Le deuxième est qu'il ne prend pas en considération la consommation d'énergie pendant le processus de routage.

3.4.1.2 ViP (Virtual Position)

Dans [YHL⁺10] les auteurs proposent d'utiliser les positions physiques des nœuds pour construire les coordonnées virtuelles. La position virtuelle d'un nœud est le point centre de tous les nœuds voisins directs. Supposons que A possède un ensemble de n voisins directs : $V_A = V_{A,1}(x_{A,1}, y_{A,1}), V_{A,2}(x_{A,2}, y_{A,2}), \dots, V_{A,n}(x_{A,n}, y_{A,n})$, la position virtuelle du nœud A est :

$$(X'_A, Y'_A) = (1/n \sum_{i=1}^n x_{A,i}, 1/n \sum_{i=1}^n y_{A,i})$$

Chaque nœud calcule sa position virtuelle suivant l'équation précédente et la diffuse à ses voisins directs.

La position virtuelle d'un nœud indique comment les nœuds voisins directs sont situés autour du nœud. Pour améliorer le taux de succès, les auteurs proposent de généraliser ViP à "*position virtuelle de haut niveau*" qui prend en considération d'autres nœuds (qui sont voisins à k -sauts, $k > 0$). La position virtuelle à k -sauts est calculée comme suit :

$$(X_A^k, Y_A^k) = ((X_{A,1}^{k-1} + X_{A,2}^{k-1} + \dots + X_{A,n}^{k-1})/n, (Y_{A,1}^{k-1} + Y_{A,2}^{k-1} + \dots + Y_{A,n}^{k-1})/n)$$

ViP(k) peut introduire des trous pendant le routage. Pour traiter ce problème, les auteurs proposent VHP (*Greedy Forwarding with Hierarchical Virtual Position*).

VHP

Lorsqu'un trou est rencontré en utilisant les positions virtuelles de K -niveau, VHP réduit le niveau de position virtuelle de K -niveau à $k-1$ niveau. Un flag est ajouté au paquet pour indiquer le niveau courant de position virtuelle.

Pour plus de garantie de livraison, les auteurs ont proposé RM-ViP. Dans RM-ViP(k), seulement un k -niveau et la position géographique sont utilisés. Si le prochain saut, en utilisant la position virtuelle de k -niveau, et la position géographique sont identiques, le nœud source envoie le paquet vers ce nœud. Sinon, deux copies sont envoyées vers les différents prochains sauts. Un nœud envoie au plus deux copies. Pour éviter la redondance, chaque nœud enregistre l'ID du paquet pour un certain temps T_{hold} . Pendant T_{hold} , chaque paquet arrivant avec cet ID sera éliminé.

L'utilisation des coordonnées virtuelles centrées et RM-ViP(k) permettent de prévoir les trous pendant l'acheminement et augmenter ainsi le taux de livraison. Cependant,

cette approche ne prend pas en considération la consommation d'énergie, de plus l'envoi de deux copies vers les différents prochains sauts engendre un trafic supplémentaire sur le réseau ce qui augmente de plus la consommation d'énergie.

3.4.1.3 VCP (Virtual Cord Protocol)

Les auteurs dans [ASGD08] ont proposé un nouveau protocole VCP basé sur DHT (*Distributed Hash Table*) qui offre en plus des fonctions DHT standards, un mécanisme de routage efficace [ASGD08]. Tous les items des données sont associés par des nombres dans un rang prédéterminé $[S, E]$ et les nœuds disponibles capturent ce rang. Donc, chaque nœud capture une part du rang entier.

Opération de jointure

En se basant sur des messages Hello, le nœud joignant le réseau obtient les informations concernant ses voisins physiques et leurs nœuds adjacents.

Le premier nœud est pré-programmé, avec la valeur la plus petite du rang (S , 'Start'). Le deuxième nœud joignant le réseau obtient le plus grand nombre de ce rang (E , 'End'). Chaque autre nœud joignant le réseau doit au moins recevoir un message Hello d'un nœud déjà existant dans le réseau pour avoir une position relative. Si un nœud peut communiquer avec un nœud de fond (S ou E), le nouveau nœud prend cette valeur (S ou E), l'ancien nœud prend une nouvelle position entre la valeur de fond (S ou E) et son successeur ou prédécesseur (selon son ancienne position). Le nouveau nœud devient prédécesseur de l'ancien nœud s'il reçoit la position S , sinon il devient son successeur.

Si un nœud peut communiquer avec deux nœuds adjacents, le nouveau nœud prend une position entre les valeurs des deux nœuds adjacents et le nouveau nœud devient le successeur de l'ancien nœud ayant la petite valeur et le prédécesseur de l'ancien nœud ayant la grande valeur.

Finalement, si le nouveau nœud peut communiquer uniquement avec un seul nœud, le nouveau nœud demande à ce nœud de créer un nœud virtuel. Ce nœud virtuel obtient une position entre la position du nœud réel et son successeur ou prédécesseur. Le nouveau nœud prend une position entre la position du nœud réel et la position du nœud virtuel.

Le routage

Pour le routage, chaque nœud doit savoir son successeur et son prédécesseur et aussi ses voisins physiques. Ensuite, un algorithme greedy est utilisé pour envoyer les paquets vers les nœuds les plus proches de la destination jusqu'à ce qu'il n'y aurait plus de progrès et la valeur (calculée dans l'opération de jointure) lie entre la positions du prédécesseur et la positions du successeur.

VCP s'appuie sur une corde pré-établie. Pour cela, le routage greedy va toujours conduire à un chemin vers la destination et donc garantit la livraison des paquets. L'inconvénient majeur de ce protocole et qu'il ne prend pas en considération la consommation d'énergie pendant le processus de routage.

3.4.1.4 Aligned VCS (Aligned Virtual Coordinates System)

Les coordonnées virtuelles de chaque nœud dans un système de coordonnées virtuelles est un vecteur de nombre de sauts à partir des différents nœuds ancrés. La valeur intégrale de la coordonnée virtuelle à la dimension i ne peut pas faire la différence entre deux nœuds ayant la même valeur de la coordonnée. Pour cela, l'alignement réfère au processus de calcul des coordonnées virtuelles d'un nœud données en fonction de ses propres coordonnées et des coordonnées de ses voisins.

Une fonction d'alignement possible, proposée dans [LAG06], qui produit un vecteur de coordonnées virtuelles alignées $AV(A)$ d'un nœud A est donnée comme suit :

$$AV(A)_i^d = \frac{(\sum_{j=1}^n AV(N_j)_i^{d-1})/n + AV(A)_i^{d-1}}{2}$$

Où i : est la i^{eme} dimension virtuelle.

N_j : j^{eme} voisins de A .

n : le nombre de voisins de A .

d : est la profondeur des coordonnées virtuelles alignées.

Une profondeur de 1 signifie faire la moyenne des coordonnées à partir des voisins à un saut. Une profondeur de n signifie prendre en compte les voisins à n sauts. Une équation alternative de coordonnées virtuelles alignées peut être :

$$AV(A)_i = \frac{((\sum_{j=1}^n AV(N_j)_i^{d-1}) + AV(A)_i^{d-1})}{n + 1}$$

Calcul de la distance sur les CV alignées

Pour un réseau où les nœuds sont statiques, la stabilité des nœuds garde les CV alignées stables. Donc, pour les décisions de routage, les coordonnées virtuelles alignées de la destination ne sont pas difficiles à obtenir. Cependant, si les nœuds sont mobiles, les coordonnées virtuelles alignées des nœuds seront rapidement primées. Pour cette considération, les coordonnées virtuelles alignées de chaque nœud destinataire ne sont pas utilisées pour le routage, ce qui signifie que les paquets de données comportent uniquement la valeur entière de la coordonnée virtuelle de la destination ($V(dest)$, ou $AV(dest)^0$).

Les coordonnées virtuelles sont utilisées uniquement pour le calcul de la distance localement. Donc, la distance Euclidienne entre un nœud X et une destination dst est calculée comme suit :

$$EDis(X, dst)^d = \sqrt{\sum_{i=1}^n (AV(X)_i^d - V(dst)_i)^2}$$

Ce protocole de routage augmente le taux de livraison mais, tout comme les algorithmes précédents, il ne prend pas en considération la consommation d'énergie pendant le processus de routage.

3.4.2 Algorithmes utilisant des ancrs pour construire des coordonnées virtuelles

Dans cette section, nous allons présenter la famille des protocoles qui essaient d'installer des nœuds spécifiques qui s'appellent ancrs (*anchors*) et à partir de ces ancrs les autres nœuds calculent leurs coordonnées virtuelles.

3.4.2.1 VCap

Dans [CCDU05], les auteurs introduisent un nouveau protocole d'assignement des coordonnées virtuelles, appelé VCap, pour supporter le routage géographique qui définit les coordonnées exclusivement en se basant sur les distances de sauts (*hop distances*). Donc, une topologie virtuelle du réseau est construite.

Le protocole VCap [CCDU05] exploite quatre ronds (basées sur la diffusion) pour identifier trois nœuds ancrs (nommés X, Y, et Z), et il assigne à chaque nœud un triplet

de coordonnées où chaque coordonnée représente la distance (en nombre de sauts) entre le nœud et un ancre.

- **Election de W** : une fois le réseau est déployé, la station de base (où n'importe quel autre nœud programmé) génère un message W_SET contenant un compteur de sauts initialisé à 1 et incrémenté à chaque nœud. Le compteur est appelé la coordonnée w . Si un nœud reçoit plusieurs messages, il prend en compte celui qui a la valeur de w la plus petite.
- **Election de X** : les nœuds ayant la valeur maximum de w décident qu'ils sont dans les frontières et ils génèrent un message X_ELECT contenant leurs ID, leurs w et un compteur de sauts initialisé à 1. Ce compteur est appelé la coordonnée x et sa valeur est incrémentée à chaque nœud. Si un nœud reçoit plusieurs messages, il prend en compte celui qui a la valeur de w la plus grande.
- **Election de Y** : cette phase est similaire à l'élection de X . la différence est dans la règle utilisée pour sélectionner le nœud éligible à être Y . la règle sélectionne les nœuds ayant le maximum x et qui satisfont $w > \gamma$. Les nœuds éligibles génèrent un message Y_ELECT contenant leurs ID, leurs x , et un compteur de sauts initialisé à 1. Le message Y_ELECT va installer une coordonnée y .
- **Election de Z** : cette phase est similaire à la précédente avec la différence que seulement les nœuds dont x et y satisfont une règle $\phi(x, y)$ et w est maximum, génèrent un message Z_ELECT contenant leurs ID et qui installent la coordonnée z .

Il est à noter que le nœud W est utilisé pour permettre aux X , Y , et Z à être proches de la frontière du réseau. Après que X , Y , et Z soient établis W n'est plus utilisé. Après que la dernière phase soit terminée, chaque nœud utilise le triplet assigné (x, y, z) comme coordonnées. Cependant, en général, les mêmes coordonnées peuvent être partagées par un ensemble de nœuds. Cet ensemble est appelé zone. La règle $\phi(x, y)$ utilisée dans la dernière phase pour choisir Z est une heuristique pour trouver un nœud satisfaisant cette contrainte.

Après que chaque nœud soit assigné par un triplet de coordonnées virtuelles, donné comme le nombre de sauts entre le nœud et chaque nœud ancre, les nœuds utilisent un routage greedy comme MFR avec la distance de Hamming (au lieu de la distance Euclidienne) calculée à partir de ces coordonnées virtuelles.

3.4.2.2 ABVCap

Dans le protocole ABVCap proposé dans [TYH07], chaque nœud u est assigné au moins par des coordonnées virtuelles de 5-tuplet $(u.lo, u.la, u.rp, u.up, u.dn)$ dans un processus à quatre phases. Où $u.lo, u.la, u.rp, u.up, u.dn$ représentent les coordonnées longitude, latitude, ripple, up, et down respectivement. Dans la première phase, quatre nœuds ancres X, Y, Z, et Z' sont sélectionnés par la même méthode utilisée dans VCap. Les axes, y compris une latitude parallèle et un nombre de meridans, sont établis dans la deuxième phase. Les coordonnées longitude, latitude, et ripple des nœuds situées dans les axes (appelés nœuds des axes) sont aussi assignés dans cette phase. Dans la troisième phase, les coordonnées longitude, latitude, et ripple des autres nœuds sont assignées. Finalement, les coordonnées up et down de tous les nœuds sont assignées dans la quatrième phase.

- *Latitude parallèle* : est l'ensemble des nœuds dans l'un des plus courts chemins entre les ancres X et Y.
- *La i^{eme} meridian* : est l'ensemble contenant le nœud u dans la latitude parallèle dont la coordonnée x est égale à i , les nœuds dans l'un des plus courts chemins entre u et Z, et les nœuds dans l'un des plus courts chemins entre u et Z'.

Le Routage

Lorsqu'un nœud u reçoit un paquet, il choisit une coordonnée virtuelle (s'il a plusieurs coordonnées virtuelles) en utilisant la stratégie suivante :

1. Si $s.lo < d.lo$: celle pour laquelle la différence entre la coordonnée longitude et $d.lo$ est la plus petite, parmi toutes les coordonnées virtuelles dont les coordonnées longitudes ne sont pas supérieurs à $d.lo$, est choisie.
2. Si $s.lo \geq d.lo$: celle pour laquelle la différence entre la coordonnée longitude et $d.lo$ est la plus petite, parmi toutes les coordonnées virtuelles dont les coordonnées longitudes ne sont pas inférieurs à $d.lo$, est choisie.

En utilisant cette stratégie, le nœud u choisit une coordonnée virtuelle pour chaque nœud voisin ayant plusieurs coordonnées virtuelles.

Sélection du prochain nœud

Si $u.lo \neq d.lo$, le nœud u envoie le paquet vers le nœud pour lequel la distance à la destination est la plus petite parmi toutes les coordonnées longitudes candidates (routage longitude). Sinon, si $u.la \neq d.la$, le nœud envoie le paquet vers le nœud

pour lequel la distance à la destination est la plus petite parmi toutes les coordonnées latitudes candidates (routage latitude). Sinon, le routage proactif dans les nœuds avec les mêmes coordonnées longitudes et latitudes comme destination est utilisé. Ici, la distance entre deux nœuds u, v est définie comme :

$$\sqrt{(u.lo - v.lo)^2 + (u.la - v.la)^2}$$

Un voisin v d'un nœud u est un candidat longitude s'il satisfait l'une des conditions suivantes :

1. $u.lo < v.lo \leq d.lo$
2. $u.lo > v.lo \geq d.lo$
3. $v.lo = u.lo < etv.up < u.up$ et
4. $v.lo = u.lo > d.lo etv.dn > u.dn$

Un voisin v d'un nœud u est un candidat latitude, s'il satisfait l'une des conditions suivantes :

1. $v.lo = u.lo$ et $|v.la - d.la| < |u.la - d.la|$ et
2. $v.lo = u.lo$ et $v.la = u.la$ et $v.rp < u.rp$.

3.4.2.3 VCost

Dans VCost [EMSR07] les auteurs supposent que les nœuds ne connaissent pas leurs positions et ils proposent une méthode basée sur des landmarks. Les nœuds calculent leurs distances vers ces landmarks pour obtenir des coordonnées virtuelles. Ensuite, VCost utilise ces coordonnées virtuelles pour acheminer les paquets de telle façon que le taux de coût pour envoyer un paquet vers la destination soit minimum.

Plusieurs nœuds L_1, \dots, L_k , avec $k \geq 3$, dans le réseau sont distingués comme des landmarks. Un nœud arbitraire x connaît son vecteur de distance $l(x) = (l_1, \dots, l_k)$ où l_i est la distance en sauts entre x et L_i . A partir du vecteur $l(x)$, le nœud génère les coordonnées virtuelles $c(x) = (x_1, \dots, x_m)$, $m \geq 2$. En général $m \leq k$. Cette fonction de calcul est notée Γ . On considère deux fonctions de Γ , l'identité noté par $\Gamma_{id}(x_i = l_i)$ et "*les coordonnées virtuelles centrées*" noté par :

$$\Gamma_{cvc}(x_i = l_i - \mu \text{ où } \mu = \frac{1}{k} \sum_{i=1}^k l_i^2)$$

Pour router un paquet vers une destination d , un nœud extrait les coordonnées virtuelles de d à partir du paquet et choisit un nœud prochain à partir de son voisinage. Les auteurs proposent d'utiliser "*Cost Over Progress*" présenté dans [KNS06a]. L'idée

est que le nœud courant x choisit le nœud $y \in N(x)$ qui minimise $\frac{cost(x,y)}{progress(x,y,d)}$, où $cost(x,y)$ représente le coût pour envoyer le paquet de x vers y , et $progress(x,y,d)$ est le progrès dans le processus de routage. Le progrès peut être exprimé comme la différence $dist(x,d) - dist(y,d)$ où $dist(u,v)$ est la distance entre les nœuds u et v .

Les auteurs dans [EMSR07] ont présenté deux fonctions de coût : $cost_1(x,y) = 1$ lorsque le nœud x ne peut pas adapter son rang de communication et $cost_e(x,y) = |xy|^\alpha + c$ sinon. La distance $|xy|$ est la distance géographique entre le nœud x et le nœud y . pour la fonction de distance $dist$ les auteurs considèrent trois fonctions différentes. La première est la distance de Hamming définie par $dist_h(x,y) = \sum_{i=1}^m |x_i - y_i|$. La deuxième est la distance Euclidienne, définie par :

$$dist_e(x,y) = \sqrt{\sum_{i=1}^m (x_i - y_i)^2}$$

La dernière fonction est la distance Euclidienne carrée et elle est définie simplement par :

$$dist_{se}(x,y) = dist_e(x,y)^2$$

Par la combinaison des fonctions Γ , les différentes fonctions de $dist$ et les fonctions de coût, on obtient une famille de protocoles. Par exemple, le triplet $(\Gamma_{id}, dist_h, cost_1)$ correspond au protocole VCap. Pour les autres protocoles, on considère la fonction de coût $cost_e$ et on utilise les abréviations suivantes pour chaque protocole : VCost pour $(\Gamma_{id}, dist_h)$, VeCost pour $(\Gamma_{id}, dist_e)$, VseCost pour $(\Gamma_{id}, dist_{se})$, CVCCost pour $(\Gamma_{cvc}, dist_h)$, CVCeCost pour $(\Gamma_{cvc}, dist_e)$, et CVCseCost pour $(\Gamma_{cvc}, dist_{se})$.

3.4.2.4 GLIDER

Les auteurs dans [FGG⁺05] ont proposé un schéma de routage appelé GLIDER qui dépend uniquement de la connectivité des nœuds et pas de la connaissance de positions des nœuds. Les auteurs proposent de partitionner le réseau en *tiles* et d'adresser chaque nœud par le nom de son *tile* et un ensemble de coordonnées locales. Le partitionnement particulier est défini par la sélection d'un ensemble de nœuds bien dispersés pour être des *landmarks*, et les *tiles* sont les cellules de Voronoi des landmarks, où une cellule Voronoi d'un landmark u est l'ensemble des nœuds dont le plus proche *landmark* est u (en terme de nombre de sauts). Un nœud peut appartenir à plus d'un *tile*. La cellule complexe associée au tel partitionnement est appelée LVC (*Landmark Voronoi Complex*). Le complexe dual de LVC a été appelé CDT (*Combinatorial Delaunay Triangulation*). CDT est utilisé pour enregistrer la relation d'adjacence entre les cellules de Voronoi.

Pour un graphe $G=(V,E)$ et un sous-ensemble de *landmarks* L inclus dans V , la cellule de Voronoi $T(v)$ d'un nœud $v \in L$ est définie comme suit :

$$T(v) = \{u \in V \mid \forall w \in L, \tau(u, v) \leq \tau(u, w)\}$$

Où $\tau(u, v)$ est le plus court chemin en terme de nombre de sauts entre u, v dans le graphe de communication.

Le nom de chaque nœud consiste en deux parties : le nom global du *tile* et les coordonnées *landmark* locales. Le nom global du *tile* est simplement l'identité de son plus proche *tile*. Les coordonnées *landmark* locales sont dérivées de l'ensemble des distances entre le nœud et ses *landmarks* qui sont proches de lui.

Les coordonnées landmark locales

Les auteurs de [FGG⁺05] ont présenté deux versions pour le calcul des coordonnées virtuelles ; la version continue et la version discrète

- *Version continue* : soit u_i un ensemble de k *landmarks*. Un vecteur de coordonnées virtuelles est assigné à chaque point p : $A(p) = (|p - u_1|, |p - u_2|, \dots, |p - u_n|)$. Où $|p - u_i|$ est la distance euclidienne entre p et u_i . La distance virtuelle dans ce système de coordonnées entre les points p et q est :

$$d(p, q) = |A(p) - A(q)|^2 = \sum_{i=1}^k (|p - u_i| - |q - u_i|)^2$$

- *Version discrete* : pour un ensemble de *landmarks* u_1, u_2, \dots, u_n et pour chaque nœud p , soit $\tau(p, u_i)$ dénote la distance (en nombre de sauts) entre p et u_i . Soit $\bar{\tau}(p) = \sum_{i=1}^k \tau(p - u_i)^2 / k$. Ensuite, on assigne à p le vecteur de coordonnées virtuelles centré :

$$C(p) = (\tau(p, u_1)^2 - \bar{\tau}(p), \dots, \tau(p, u_i)^2 - \bar{\tau}(p))$$

La distance virtuelle centrée entre deux points p, q est donc : $d(p, q) = |C(p) - C(q)|^2$.

Le routage

Supposons que le routage se fait du nœud u vers le nœud v . Le routage dans GLIDER consiste en deux étapes : routage global et le routage local.

- *Routage global* : ça revient à identifier le plus court chemin entre $h(u)$ et $h(v)$ dans CDT. Le chemin fournit une séquence de tiles ; T_1, T_2, \dots, T_k , où $T_i = T(u_i)$ pour les landmarks u_i , avec $u_1 = h(u)$, $u_k = h(v)$. $h(u)$ dénote le nom de *landmark* de u .
- *Routage local* : Le routage local consiste en le routage *inter-tile*, qui est responsable de découverte de chemin entre les *tiles* T_i, T_{i+1} , et routage *intra-tile*, qui est responsable de découverte de chemin vers v une fois T_k est atteint.

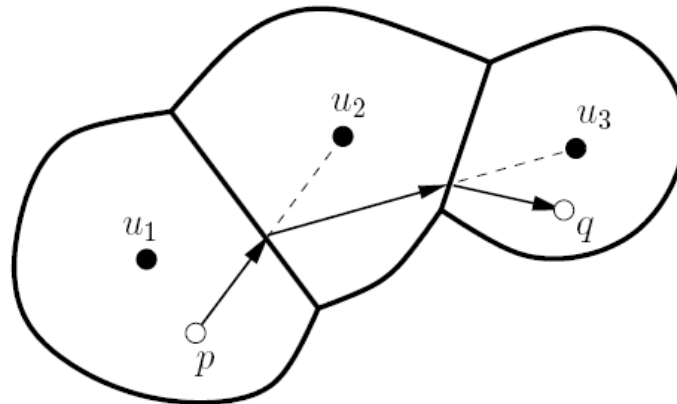


FIG. 3.11 – Routage a travers les tiles.

Le routage *intra-tile* est effectué en utilisant les coordonnées landmark locales. Plus spécifiquement, lorsque le paquet atteint un nœud $w \in T(h(v))$, ce dernier est relié à un voisin qui est plus proche du landmark v dans la distance euclidienne sur les coordonnées landmark locales.

Il est évident que les algorithmes qui utilisent des nœuds ancrés (*des landmarks*) pour calculer les coordonnées virtuelles ne garantissent pas la livraison des paquets car cette technique ne garantit pas l'unicité des coordonnées des nœuds (plusieurs nœuds peuvent avoir les mêmes coordonnées virtuelles) et donc n'identifient pas un nœud unique. De plus, VCap, ABVCap, et GLIDER ne prennent pas en considération le problème de la consommation d'énergie pendant le processus de routage.

3.5 Comparaison des protocoles de routage géographique dans les RCSFs

Dans cette section nous allons présenter une comparaison entre les différents protocoles de routage géographique dans les réseaux de capteurs sans fil étudiés précédemment dans ce chapitre, en se basant sur le type de coordonnées utilisées pour prendre les décisions de routage, la structure de réseau, la garantie de livraison, les informations sur la topologie nécessaire, la conservation d'énergie, et la stratégie du chemin.

Protocole	Type de coordonnées	Structure du réseau	Garantie de livraison	Localisé (Localized)	Energy aware	Chemin
MFR	Physique	plat	Non	Oui	Non	Chemin simple
Greedy routing	Physique	plat	Non	Oui	Non	Chemin simple
Compass routing	Physique	plat	Non	Oui	Non	Chemin simple
NFP	Physique	plat	Non	Oui	Non	Chemin simple
GEDIR	Physique	plat	Non	Oui	Non	Chemin simple
Face routing	Physique	planaire	Oui	Oui	Non	Chemin simple
GFG	Physique	planaire	Oui	Oui	Non	Chemin simple
Power progress	Physique	planaire	Oui	Oui	Oui	Chemin simple
GPSR	Physique	planaire	Oui	Oui	Non	Chemin simple
GOAFR	Physique	planaire	Oui	Oui	Non	Chemin simple
GRID	Physique	Grille carée	Oui	Non	Non	Chemin simple
Terminode	Physique	zones	Non	Non	Non	Chemin simple
LAR	Physique	plat	Non	Non	Non	Diffusion
DREAM	Physique	plat	Non	Non	Non	Diffusion
VCR	virtuelle	plat	Non	Oui	Non	Chemin simple
ViP	virtuelle	plat	Non	Oui	Non	Chemin simple
VCP	virtuelle	plat	Oui	Oui	Non	Chemin simple
Aligned VCS	virtuelle	plat	Non	Oui	Non	Chemin simple
VCap	virtuelle	plat	Non	Oui	Non	Chemin simple
Aligned VCS	virtuelle	plat	Non	Oui	Non	Chemin simple
VCap	virtuelle	plat	Non	Oui	Non	Chemin simple
ABVCap	virtuelle	plat	Non	Oui	Non	Chemin simple
VCost	virtuelle	plat	Non	Oui	Oui	Chemin simple
GLIDER	virtuelle	plat	Non	Oui	Non	Chemin simple

TAB. 3.1 – Comparaison entre les différents protocoles étudiés dans ce chapitre.

D'après le tableau comparatif ci-dessus établi à partir des protocoles de routage géographique dans les RCSFs étudiés dans ce chapitre, on peut dire qu'il est difficile de concevoir un protocole qui supporte tous les critères. Cependant, on peut en tirer les remarques suivantes :

- L'utilisation des coordonnées virtuelles influence la garantie de livraison des paquets (tous les protocoles basés sur des coordonnées virtuelles ne garantissent pas la livraison des paquets à l'exception de VCP).
- Tous les protocoles basés sur des coordonnées virtuelles sont localisés (*localized*), utilisent une structure de réseau à plat, et établissent un chemin simple.
- Les protocoles basés sur des coordonnées physiques qui utilisent une structure de réseau à plat ne garantissent pas la livraison des paquets.
- La majorité de ces protocoles ne prennent pas en considération la consommation d'énergie pendant l'acheminement des données, et peu de protocoles sont conçus pour conserver l'énergie des nœuds.
- Les protocoles basés sur des coordonnées physiques qui utilisent une structure de réseau planaire garantissent la livraison des paquets.

3.6 Conclusion

Dans ce chapitre, nous avons réalisé une brève étude sur les différents protocoles de routage géographique dans les réseaux de capteurs sans fil. Le routage basé sur des coordonnées physiques obtenues à partir des systèmes de localisation, et le routage basé sur des coordonnées virtuelles calculées à partir des coordonnées absolues ou bien à partir des nœuds ancrés (en termes de nombre de sauts). Cette étude nous a permis de mettre en évidence le fonctionnement, les points forts et les limites de ces protocoles. Une comparaison entre ces différents protocoles a été également présentée. Bien que plusieurs de ces stratégies paraissent prometteuses, il existe toujours certains défis qui persistent et nécessitent leur prise en considération par les protocoles de routage dans les réseaux de capteurs sans fil.

Nous nous intéresserons dans la suite à la conception d'un nouveau protocole de routage géographique localisé (*localized*) pour les RCSFs qui minimise la consommation d'énergie et qui garantit la livraison des paquets entre les nœuds sources et la destination.

Chapitre 4

Nouveau protocole de routage géographique

4.1 Introduction

LE routage géographique est l'une des techniques les plus utilisées pour router les données dans un réseau de capteurs sans fil. Il est basé sur une stratégie d'acheminement dite greedy dans laquelle le nœud capteur sélectionne parmi les nœuds voisins les plus proches de la destination, un nœud selon la métrique de routage utilisée. Les décisions de routage sont basées sur la position du nœud en cours, les positions des voisins, et la position de la destination. Ils sont généralement appelés "*localisés*" parce qu'ils se basent uniquement sur les informations locales fournies depuis les nœuds voisins.

Sachant que les nœuds capteurs sont alimentés par des batteries ayant une capacité limitée, l'efficacité énergétique dans le processus de routage est un facteur très important et un nouveau protocole de routage doit prendre en compte ce facteur.

Si les nœuds possèdent les informations des positions de tous les autres nœuds dans le réseau, alors le chemin optimal qui minimise la consommation d'énergie peut être calculé en utilisant l'algorithme du plus court chemin de Dijkstra, où le coût de transmission entre chaque deux nœud représente le coût de l'arc entre ces deux nœuds. Or, les nœuds ne connaissent que leurs positions, les positions de leurs voisins, et la position de la destination. Dans notre proposition, nous allons essayer d'exploiter ces connaissances pour développer un nouveau protocole de routage géographique localisé qui garantit la livraison des données tout en minimisant la consommation d'énergie.

4.2 Objectifs

L'objectif de notre travail est de développer un nouveau protocole de routage géographique localisé qui répond aux exigences suivantes :

1. **Consommation d'énergie** : la consommation d'énergie dans un protocole de routage est la métrique la plus pertinente. Dans notre protocole, nous avons essayé de minimiser la consommation d'énergie le plus possible.
2. **Localisé (localized)** : la connaissance globale de la topologie du réseau par chaque nœud est impossible. Pour cela, l'objectif de notre travail est de développer un protocole qui utilise uniquement les connaissances locales fournies depuis les nœuds voisins.
3. **Robustesse** : la panne des nœuds, l'apparition, et la disparition des nœuds dans le réseau sont des événements qui apparaissent fréquemment, donc le but est d'introduire un mécanisme de recouvrement pour que le protocole de routage puisse continuer à fonctionner dans de telles situations.

4.3 Modèle du réseau

Dans notre protocole on suppose que les nœuds sont statiques et chaque nœud connaît sa position (en utilisant un protocole de localisation par exemple SPA (*Self-Positioning Algorithm*) [CHH01]), la position de ses voisins, l'énergie résiduelle de ses voisins, et la position de la station de base.

Nous allons modéliser le réseau par un graphe $G = (E, V)$,
Où E représente l'ensemble des nœuds capteurs dans le réseau.
et V représente l'ensemble des arcs entre ces nœuds.

On suppose que chaque nœud $u \in E$ a un rang de transmission maximal r , et que tous les nœuds ont le même rang de transmission. un arc uv existe entre les deux nœuds $u, v \in E$ si et seulement si la distance euclidienne entre eux est inférieure ou égale au rang de transmission r (i.e. les deux nœuds peuvent communiquer entre eux). $uv \in V$ ssi $dist(u, v) \leq r$.

Dans notre système nous allons adopter le modèle énergétique de Ivan Stojmenovic [SL01b] dans lequel il a généralisé le modèle de Rodoplu et Meng [RM99] par l'ajout d'un facteur linéaire a :

$$u(d) = ad^\alpha + c; \tag{4.1}$$

Où d : est la distance entre les deux nœuds.

α : est un facteur d'atténuation du media entre 2 et 6.

c : est une constante.

a : Le facteur linéaire a peut être ajusté à l'environnement physique, unité de longueur considéré, longueur d'un signal, ...

4.4 Fonctionnement du protocole

L'idée de base de notre protocole *EGGR* (*Energy aware and delivery Guarantee Geographic Routing*) est d'exploiter les positions de tous les voisins situés dans la direction de la station de base pour trouver un chemin optimal menant vers la station de base. Ce chemin doit consommer le minimum d'énergie et garantir la livraison des données. EGGR est basé sur le routage greedy toute en essayant d'éviter les trous (un nœud qui ne possède aucun voisin plus proche de la destination).

Le protocole EGGR s'exécute sur deux phases : la phase de communication de données et la phase de maintenance du réseau. Dans la première phase, chaque nœud calcule son prochain nœud, ce dernier est utilisé pour l'acheminement des données vers la station de base. Tandis que la deuxième phase sert à mettre à jour les informations sur les nœuds voisins.

4.4.1 Phase de communication de données

L'objectif de cette phase est l'acheminement des données depuis les nœuds source vers la station de base. Dans cette phase, chaque nœud calcul son prochain nœud. Le calcul du prochain nœud commence par la construction d'un sous graphe G' à partir des voisins qui sont dans la direction de la station de base (voir la section 4.4.1.1). Le prochain nœud est le premier nœud dans le plus court chemin (voir la section 4.4.1.2) entre le nœud courant et la station de base.

Pour éviter les trous, lorsqu'un nœud détecte qu'il n'a pas de voisin plus proche de la station de base que lui, il informe tous ses voisins pour qu'ils ne le considèrent pas pendant le calcul du prochain nœud. Donc, ce nœud est considéré comme un nœud négatif. Un nœud négatif peut participer dans le calcul du plus court chemin.

Le nœud courant utilise le prochain nœud pour l'acheminement des données. Si le prochain nœud est négatif alors le plus court chemin est intégré dans l'entête du paquet et le prochain nœud achemine le paquet selon le chemin indiqué dans l'entête (routage par source SR). Si le nœud courant est négatif alors il choisit son plus proche voisin comme prochain nœud. Cette phase est résumé dans l'algorithme1 (voir la page78).

Le calcul du prochain nœud s'effectue à l'installation du réseau et lorsqu'un changement apparaît dans la table des voisins (changement dans l'énergie résiduelle d'un voisin, panne d'un voisin, apparition ou disparition d'un voisin, ...).

4.4.1.1 Construction du sous graphe G'

Le sous graphe G' est construit à partir des nœuds voisins les plus proches de la destination que le nœud courant. Chaque nœud maintient une table de voisins contenant :

- L'identifiant du voisin : pour distinguer entre les différents voisins.
- Adresse du voisin.
- Energie résiduelle : l'énergie restante du voisin.
- Etat (positif ou négatif) : sert pour savoir si le nœud présente un trou ou non.

Identifiant	Adresse	Energie résiduelle	Etat

TAB. 4.1 – Format de la table des voisins.

Soit $dist(u, v)$ la distance entre les deux nœud u et v .

Soit r le rang de transmission des nœuds. Le sous graphe $G' = (E', V')$ tel que :

E' est l'ensemble des nœuds voisins qui sont dans la direction de la station de base.

V' l'ensemble des arcs.

Un nœud $u \in E'$ si et seulement si $dist(u, p) \leq dist(s, p)$ et $dist(s, u) \leq r$. Un arc $uv \in V'$ si et seulement si $dist(s, u) \leq r$ et $dist(s, v) \leq r$ et $dist(u, v) \leq r$ où s est le nœud courant et p est la station de base. Plus un arc su entre le nœud courant s et chaque nœud $u \in E'$ et un arc ud entre chaque nœud $u \in E'$ et la station de base p .

Les coûts des arcs

Le coût de chaque arc représente l'énergie nécessaire pour l'émission et la réception entre les deux nœuds en question. Le coût de l'émission/réception entre chaque deux nœuds appartiennent à E' et le coût de l'émission/réception entre le nœud courant s et chaque nœud de E' [RM99] est :

$$u(d) = ad^\alpha + c. \quad (4.2)$$

Où d est la distance entre les deux nœuds.

Ivan Stojmenovic et Xu Lin ont démontré dans [SL01b] que si la distance entre la source et la destination d est supérieur strictement à $(c/(a(1 - 2^{1-\alpha})))^{1/\alpha}$ alors $n - 1$ nœuds intermédiaires entre la source et la destination avec des distances égales peuvent être sélectionnés pour la retransmission des paquets, où $n = d(a(\alpha - 1)/c)^{1/\alpha}$ (ajusté à l'entier le plus proche), produisant une consommation énergétique minimale égale à [SL01b] :

$$v(d) = dc(a(\alpha(\alpha - 1)/c)^{1/\alpha} + da(a(-1)/c)^{(1-\alpha)/\alpha}. \quad (4.3)$$

A partir de ce théorème, on suppose que le coût de retransmission entre chaque nœud $u \in E'$ et la destination d est optimal et égale à [SL01b] :

$$v(d) = dc(a(\alpha(\alpha - 1)/c)^{1/\alpha} + da(a(-1)/c)^{(1-\alpha)/\alpha}. \quad (4.4)$$

Où d est la distance vers la destination.

a est un facteur linéaire.

α est le facteur de l'atténuation du média entre 2 et 6.

c est un constant.

Si $\alpha = 2$ alors

$$v(d) = 2d(ac)^{1/2}. \quad (4.5)$$

4.4.1.2 Calcul du plus court chemin

Pour calculer le plus court chemin, nous utilisons l'algorithme de dijkstra [Dij65]. Le calcul du plus court chemin s'effectue sur le sous graphe G' . En plus des coûts de transmission sur les arcs et pour interdire de choisir toujours le même chemin et ainsi faire un équilibrage dans la consommation d'énergie, nous proposons d'ajouter une autre métrique qui est l'énergie résiduelle des nœuds ER_i . Donc, la fonction de calcul du plus court chemin est basée sur deux paramètres ; le coût de l'émission/réception et l'énergie résiduelle des nœuds :

$$f = u(d)/ER_i. \quad (4.6)$$

Algorithme1 : Calcul du prochain nœud

Soit E l'ensemble des voisins du nœud courant u , et soit $N(u)$ l'ensemble des voisins positifs du nœud courant u qui sont plus proches de la station de base que u .

$Cout_{SP}(u, v) + dc(a(\alpha(\alpha - 1)/c)^{1/\alpha} + da(a(-1)/c)^{(1-\alpha)/\alpha}$ représente le plus court chemin entre u et v plus le coût de retransmission entre v et la station de base SB que l'on suppose optimal.

Coût_Minimum = ∞ ;

Coût=0 ;

Construire le sous graphe G' ;

Tant que $u \neq$ de la destination **Faire**

Si $N(u) = 0$ **alors**

Si u est la source du paquet **alors**

Prochain_Nœud = plus proche voisin dans E ;

Sinon

Enlever la première adresse incluse dans l'entête du paquet ;

Prochain_Nœud = le nœud dont l'adresse est incluse dans l'entête du paquet ;

Finsi

Sinon

Pour chaque voisin $v \in N(u)$ et $v.etat \neq$ négatif **Faire**

CoûtSP(u,v) = plus court chemin entre u et v ;

$Coût = CoûtSP(u,v) + dc(a(\alpha(\alpha-1)/c)^{1/\alpha} + da(a(-1)/c)^{(1-\alpha)/\alpha}$;

Si $Coût < Coût_Minimum$ **Alors**

Coût_Minimum = *Coût* ;

Prochain_Nœud = Premier nœud dans $SP(u,v)$;

Si l'état du premier nœud dans $SP(u,v)$ est négatif **Alors**

Inclure le $SP(u,v)$ dans l'entête du paquet ; { utiliser Source Routing}

FinSi

FinSi

FinPour

Envoyer le paquet au *Prochain_Nœud* ;

FinSi

Fin Tant que

4.4.2 Phase de maintenance

L'objectif de cette phase est de garder la fraîcheur de la table des voisins. Des messages de mises à jour sont diffusés périodiquement, ils contiennent l'identificateur, l'adresse, l'énergie résiduelle, et l'état du nœud (positif ou négatif). De plus un message de changement d'état est diffusé dès que l'état est changé par la disparition (batterie épuisée, panne, ...) ou l'apparition d'un nœud dans le réseau pour informer les voisins du changement d'état.

Algorithme2 : Mise à jour

Lors de la réception d'un message de mise à jour :

Mettre à jour les informations du nœud correspondant ;

Si il y a un changement dans la table de voisins **alors**

Si tous les voisins dans la direction de la SB sont négatifs **alors**

 Etat = négatif ;

 Diffuser_message_de_mise_à_jour;

FinSi

 Calculer_prochain_nœud;

Finsi

Pour mieux comprendre notre protocole EGGR nous présentons dans ce qui suit un exemple illustrant son fonctionnement :

Dans la figure 4.1, soit S le nœud qui dispose d'un paquet à acheminer vers la destination D , il commence la construction du sous graphe $G' = \{E', V'\}$.

$$E' = (A, B, C, E, F, G) \cup (S, D). \quad V' = (SA, SB, SC, SE, SF, SG) \\ \cup (AB, AC, AE, BC, CE, EF, EG, FG) \cup (AD, BD, CD, ED, FD, GD).$$

Après la construction du sous graphe G' , le nœud source S calculera le plus court chemin en terme de consommation d'énergie entre lui est la destination D selon la fonction illustrée dans la section 4.4.1.2.

On suppose, dans cet exemple, que le plus court chemin en terme de consommation d'énergie entre S et D est $SAECD$. Donc, le prochain nœud de S est A (le premier nœud dans le plus court chemin). Le nœud source S utilise ce prochain nœud A pour

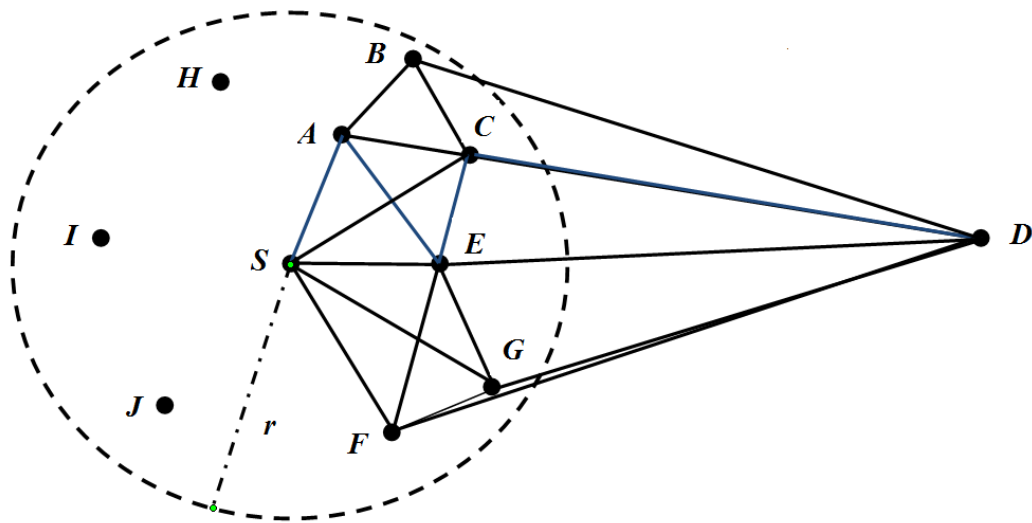


FIG. 4.1 – Exemple de Fonctionnement de EGGR.

router les données vers la destination jusqu'à ce qu'un changement apparaisse dans la table des voisins.

4.5 Evaluation des performances de EGGR

Dans son sens le plus général, la simulation est une technique de modélisation du monde réel. Elle consiste à modéliser un système en représentant toutes ses entités, leurs comportements et leurs interactions.

Dans le domaine des réseaux de capteurs sans fil, la simulation est une étape incontournable lorsqu'on veut tester et évaluer des modèles d'application ou des protocoles de communication. L'expérimentation réelle s'avère quelques fois très coûteuse. De plus, la simulation offre un gain considérable en temps, une flexibilité en permettant la variation des paramètres et une meilleure visualisation des résultats sous forme de graphes faciles à analyser et interpréter.

Plusieurs environnements de simulation sont utilisés pour évaluer les performances des protocoles. Certains sont libres et parfois Open Source, par contre, il y en a d'autres qui sont commercialisés. Aucun de ces simulateurs n'est parfait ni ne répond à tous les besoins. Chacun d'eux présente des avantages et des inconvénients.

Les simulateurs les plus utilisés sont NS2, OPNET, OMNET++, TOSSIM, GlomoSim, et J-Sim.

NS2 : C'est un simulateur développé à Lawrence Berkeley National Laboratory (LBNL) [2.]. Il est conçu principalement pour le monde de l'Internet. Il permet de simuler le comportement des protocoles standard de l'Internet tels que TCP, IP et permet d'étendre le simulateur aux nouveaux protocoles de l'Internet (routage, transport, application) et aux nouvelles architectures.

OPNET : OPNET (Optimised Network Engineering Tool) [OT] est un simulateur à événements discrets. C'est un outil très puissant dédié à l'étude des réseaux de télécommunications. Au niveau architectural, OPNET se décompose en trois parties : la partie modélisation, la partie tests et la partie évaluation chargée de l'analyse des résultats

OMNeT++ : OMNeT++ (Objective Modular Network Test-bet in C++) [Sys] est un environnement de simulation des événements discrets. Il est essentiellement utilisé pour la simulation des communications réseaux, mais grâce à son architecture générique et flexible, il est utilisé avec succès dans plusieurs autres champs tels que : la simulation de systèmes complexes de traitement et de communications de données, les réseaux de files d'attente, la modélisation des multiprocesseurs et d'autres systèmes hardware distribués, la validation des architectures hardware, etc.

GloMoSim : GloMoSim (Global Mobile Simulator) [Lib] est un environnement de simulation à grande échelle pour les réseaux sans fil et filaires qui à été développé à UCLA (University of California, Los Angeles). Il a été conçu en utilisant la capacité de la simulation parallèle fournie par PARSEC (PARallel Simulation Environment for Complex systems) qui est une extension du langage C.

TOSSIM : TOSSIM [LL] est un simulateur d'évènements discrets pour les réseaux de capteurs utilisant le système TinyOS (l'OS le plus utilisé pour les RCSFs). Au lieu de compiler l'application directement dans le capteur, l'utilisateur peut le faire dans la framework TOSSIM qui s'exécute sur un PC.

JSim : J-Sim [pag, SCH⁺06, Tya02] est un logiciel utilisé pour simuler le comportement des processus pseudo-parallèles, programmé en Java. J-Sim repose sur une structure logicielle basée sur des composants, appelée " Autonomous Component Architecture" (ACA). Le code source est organisé en paquetages relatifs à un type de composants. Un composant est une entité indépendante représentant un objet physique (une batterie, un module radio, une couche logicielle, etc.) ou logique (un protocole de routage, un modèle de mobilité, etc.). Ces composants seront ensuite connectés à l'aide de ports afin de générer un réseau simulé.

La simulation du fonctionnement d'un réseau de capteurs, qui exige la définition des composants et leur mise en relation, est réalisée grâce à un langage spécifique, TCL

[201]. Il s'agit d'un langage de script dans lequel on spécifie l'architecture du réseau ainsi que les paramètres de simulation et d'analyse. Les commandes de script peuvent également être fournies en ligne de commande, instruction par instruction. A l'aide de TCL, on définit les composants puis on les connecte. Tous les composants sont hébergés dans un conteneur, qui est à son tour un composant. La définition des composants est en fait la création des objets.

Cette répartition en composants, rend J-Sim une plate-forme véritablement extensible et un environnement réutilisable. Il offre également une analyse des résultats plus aisés que le simulateur NS-2 par exemple. Plus important encore, il permet de simuler des réseaux de l'ordre de 1000 nœuds via l'architecture suffisamment bien structurée de ses composants. En outre, J-Sim gère correctement l'aspect consommation d'énergie grâce au module batterie et les modèles de consommation d'énergie offerts, sans oublier le modèle de mobilité, canaux de communication sans fil et les supports physiques disponibles aussi dans ce simulateur.

4.5.1 Choix de l'environnement de simulation

Nous avons utilisé J-Sim comme environnement de simulation pour évaluer les performances de notre nouveau protocole EGGR. Notre choix se justifie par le fait que ce simulateur libre et ouvert est très puissant dans la gestion de l'aspect consommation d'énergie grâce aux modèles de consommation d'énergie offerts par ce simulateur le point le plus crucial dans les réseaux de capteurs sans fil. De plus, ce simulateur est développé entièrement en Java, langage orienté objet satisfaisant aux besoins de l'immense majorité des développeurs, caractérisé par sa fiabilité, réutilisabilité, architecture neutre et indépendante, performances élevées et multithread. En outre, J-Sim est très scalable, il peut simuler jusqu'à 1000 nœuds avec moins deux ordres de grandeur inférieure de mémoire allouée par rapport à NS-2. Cela fournit une très grande performance pour effectuer des simulations à grande échelle.

4.5.2 Paramètres de simulation

Les nœuds capteurs utilisés dans la simulation sont considérés comme homogènes : possédant la même quantité d'énergie initiale, les mêmes capacités de calcul et mémoire, les mêmes portée de transmission et équipés par les mêmes interfaces de communication IEEE 802.11. De plus, l'énergie de la station de base est considérée comme illimitée. Ces capteurs sont déployés à l'intérieur d'un carré de taille 100 par 100 créant ainsi des topologies contenant 20 à 160 capteurs. La position des nœuds dans les topologies générées est réalisée d'une manière aléatoire. Le tableau ci-dessous résume les paramètres

utilisés :

Paramètre	Valeur
Nombre de nœuds	De 20 à 160
Nombre de stations de base	1
Nombre de nœuds cibles	5
Surface de simulation	100X100
Portée de transmission	30
Energie initiale	0.5 Joules
Puissance de transmission radio (Watt)	0.028
Puissance de réception radio (Watt)	0.036
La sous couche MAC	IEEE 802.11
α	2
C	1000
a	1

TAB. 4.2 – Paramètres de simulation.

4.5.3 Les métriques d'évaluation

Pour évaluer l'efficacité énergétique et la garantie de livraison de notre protocole EGGR, nous avons choisi le protocole GPSR comme protocole de comparaison. Ce choix est justifié par le fait que le protocole GPSR est considéré comme protocole de référence pour les protocoles de routage géographiques dans les réseaux de capteurs sans fil qui garantissent la livraison des paquets entre la source et la destination. La

comparaison de notre protocole avec ce protocole portera sur les métriques suivantes :

4.5.3.1 Consommation d'énergie

Le but principal de notre protocole EGGR est de minimiser la consommation d'énergie des nœuds, car celle-ci influe directement sur la durée de vie du réseau. Pour cela, nous allons comparer notre protocole avec le protocole GPSR en prenant comme critère, l'énergie moyenne EM consommée par tous les nœuds dans le réseau.

$$EM = \frac{\sum_{i=1}^n (E_i - ER_i)}{n}$$

Où : E_i est l'énergie initiale du nœud i .

ER_i est l'énergie résiduelle du nœud i .

n est le nombre de nœuds dans le réseau.

4.5.3.2 Taux de succès

Le taux de succès représente le rapport entre le nombre de paquets reçus par la station de base et le nombre total de paquets envoyés par tous les nœuds dans le réseau.

$$\text{Taux_succès} = (\text{nbr_paquets_reçus_SB} / \text{nbr_total_paquets_envoyés}).$$

Un protocole de routage pour les réseaux de capteurs doit minimiser le nombre de paquets de données perdus lors du transfert depuis un nœud capteur vers la station de base, car l'information transportée peut être très importante.

4.5.3.3 Durée de vie du réseau

Pour mesurer la durée de vie du réseau, nous avons, dans un premier temps, défini la durée de vie d'un RCSF comme étant la durée écoulée depuis le déploiement du réseau jusqu'à ce que le dernier nœud meurt. Ensuite, nous avons défini la durée de vie d'un RCSF comme étant la durée écoulée depuis le déploiement du réseau jusqu'à ce que le premier nœud meurt. Enfin, nous avons défini la durée de vie d'un RCSF comme étant la durée de vie du quatre vingtième nœud qui meurt. Cette métrique est très importante car elle nous permet d'évaluer directement l'efficacité énergétique du protocole.

4.5.3.4 La latence moyenne

La latence est le temps nécessaire à un paquet pour qu'il soit acheminé à partir du nœud source jusqu'à la station de base. Donc, la latence moyenne est la somme des

latences de tous les paquets divisée par le nombre total des paquets reçus.

$$La_latence_moyenne = \left(\sum_{i=1}^n (temps_reception_i - temps_emission_i) \right) / n$$

Où : n est le nombre de paquets reçus par la station de base.

4.5.4 Discussion des résultats

Dans ce qui suit, nous allons présenter et analyser les résultats de la simulation obtenus suivant les métriques de performance discutées précédemment. Le résultat de chaque scénario est la moyenne de l'exécution de 30 simulations indépendantes.

Consommation d'énergie

Les résultats de la simulation présentés sur les figures 4.2 et 4.3 montrent que le protocole EGGR améliore considérablement la consommation moyenne d'énergie dans le réseau par rapport au protocole GPSR. Cette amélioration peut être justifiée par le mécanisme appliqué par EGGR qui essaye d'acheminer le paquet via le chemin optimal en termes de consommation d'énergie. Par contre, le protocole GPSR achemine les paquets à travers un chemin qui peut être coûteux en termes de consommation d'énergie et surtout lorsque il y'en a un trou (mécanisme de recouvrement).

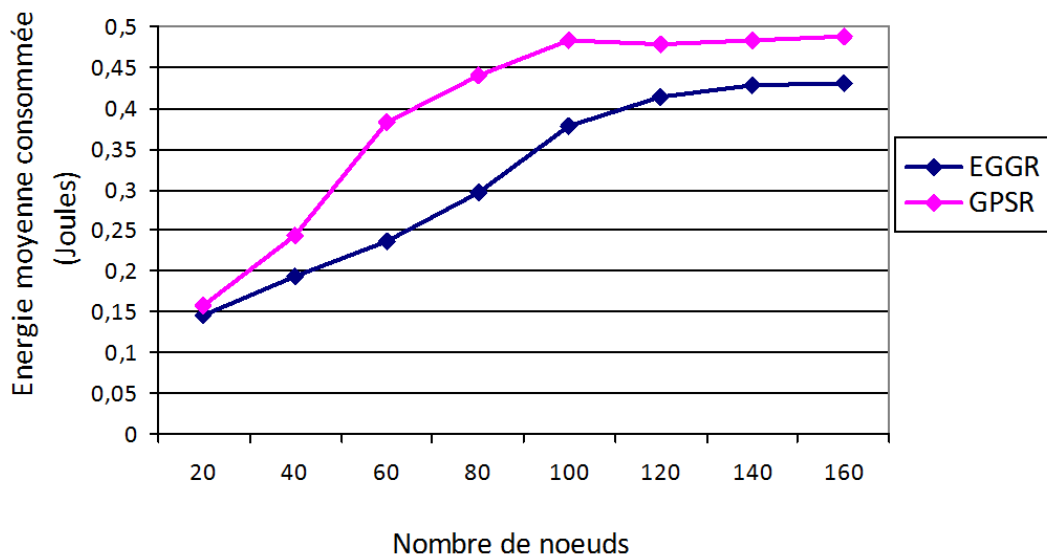


FIG. 4.2 – Moyenne d'énergie consommée par rapport au nombre de noeuds dans le réseau.

De même, on peut remarquer, d'après la figure 4.2, que le nombre de nœuds dans le réseau augmente l'énergie moyenne consommée pour les deux protocoles. Ceci se justifie par le nombre de messages de contrôle et de maintenance (pour garder la fraîcheur de la table de voisins au niveau de chaque nœud) qui circulent dans le réseau, ce qui augmente l'énergie moyenne consommée.

D'après la figure 4.3, on peut constater qu'avec GPSR les nœuds épuisent leurs énergies plus rapides qu'avec notre protocole EGGR. Et ca démontre l'efficacité du mécanisme de balance, appliqué par notre protocole pendant le calcul du plus court chemin, qui essaye de distribuer la consommation d'énergie sur tous les nœuds de manière plus efficace ce qui permet aux nœuds de conserver leurs énergies pour une longue période de temps.

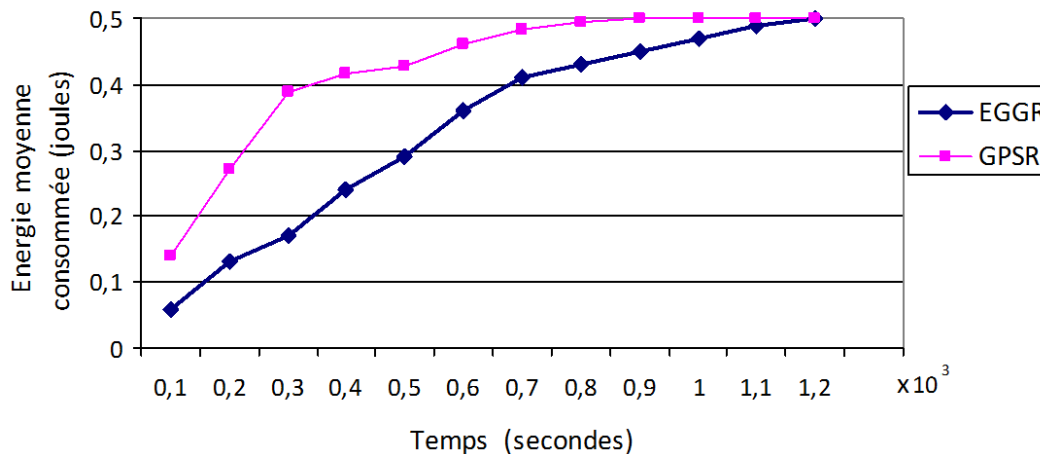


FIG. 4.3 – Moyenne d'énergie consommée dans le temps.

Taux de succès

Les résultats de la simulation présentés sur la figure 4.4 montrent que le taux de succès des deux protocoles varie entre 80 % et 99 % et cela montre l'efficacité du mécanisme de recouvrement qui garantit la livraison des paquets vers la station de base. On peut aussi remarquer que le nombre de nœuds dans le réseau influe sur le taux de succès ; en effet, plus le nombre de nœuds dans le réseau est élevé plus le taux de succès se rapproche de 100 % pour les deux protocoles, i.e, plus le nombre de nœuds dans le réseau est élevé plus la probabilité d'arriver à un trou est petite et plus le nombre de nœuds dans le réseau est petit plus la probabilité d'arriver à un trou est grande ce qui influe sur le taux de succès.

On peut aussi remarquer que notre protocole présente un avantage par rapport au

protocole GPSR avec lequel les paquets arrivent facilement à des trous, ce qui nécessite l'application du mécanisme de recouvrement.

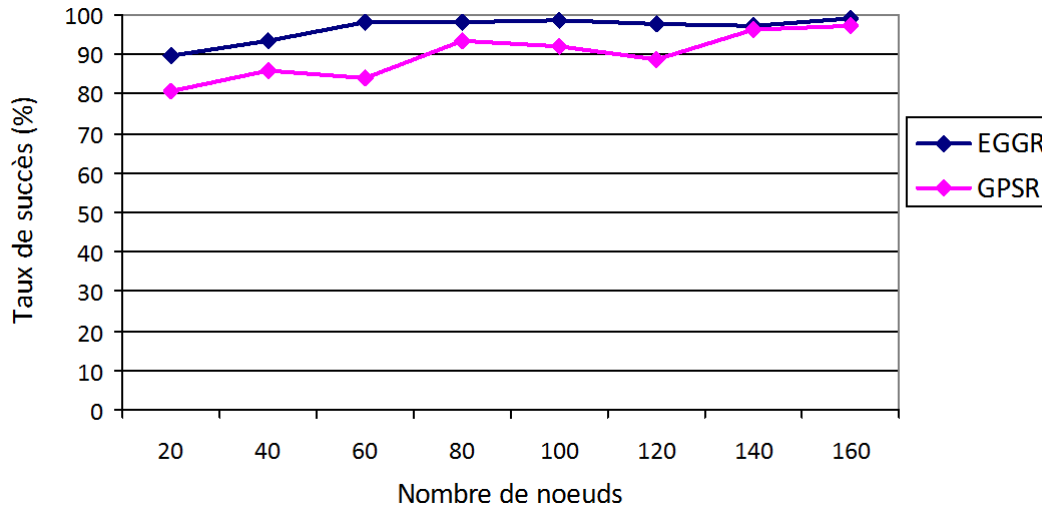


FIG. 4.4 – Taux de succès.

Durée de vie du réseau

A partir des figures 4.5 et 4.6 on peut constater que le nombre de nœuds dans le réseau influe sur la durée de vie du réseau entier. Et ça revient au nombre de messages de contrôle et de maintenance qui sont émis périodiquement par chaque nœud dans le réseau pour la mise à jour des tables de voisins. Plus il y a de nœuds dans le réseau plus il y a de messages de contrôle circulant dans le réseau, ce qui augmente la consommation d'énergie dans le réseau entier et ainsi réduit la durée de vie du réseau. Cependant, on peut remarquer dans la figure 4.5 (si on prend la durée de vie du réseau par rapport au dernier nœud qui meurt) que notre nouveau protocole surpasse le protocole GPSR et augmente la durée de vie du réseau à près de 25% et ceci revient au mécanisme de sélection du prochain nœud et au mécanisme de balance qui essaye de favoriser le nœud qui a une grande énergie résiduelle lors du calcul du plus court chemin. Par contre, si on prend la durée de vie du réseau par rapport au premier nœud qui meurt (la figure 4.6), notre protocole ne marque pas une grande différence avec le protocole GPSR.

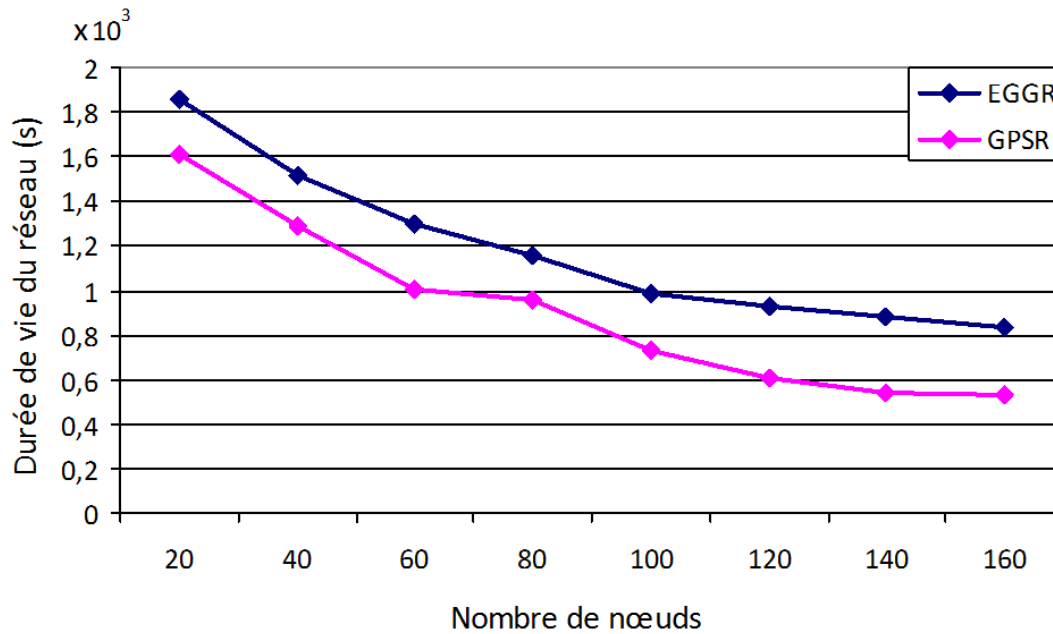


FIG. 4.5 – Durée de vie du réseau : durée de vie du dernier nœud qui meurt.

Dans la figure 4.7, on considère la durée de vie du réseau en fonction du nombre de nœuds en vie dans le réseau. Et à partir de ces résultats, il s'avère que notre protocole EGGR permet une distribution de la charge du trafic circulant dans le réseau sur tous les nœuds capteurs de manière très efficace par rapport au protocole GPSR ; ce qui permet de prolonger significativement la durée de vie des nœuds capteurs et ainsi prolonger la durée de vie du réseau entier.

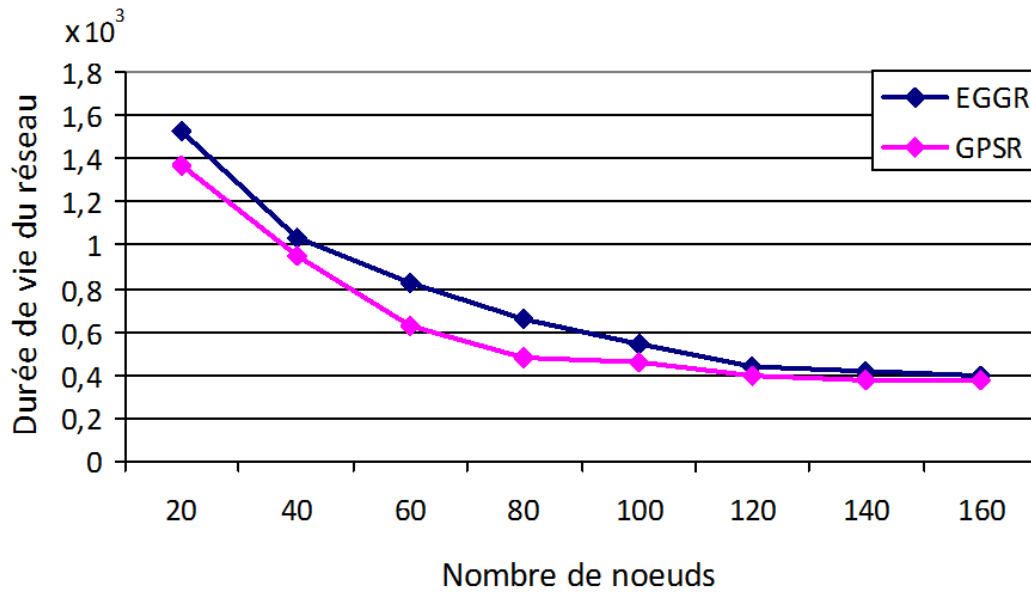


FIG. 4.6 – Durée de vie du réseau : durée de vie du premier nœud qui meurt.

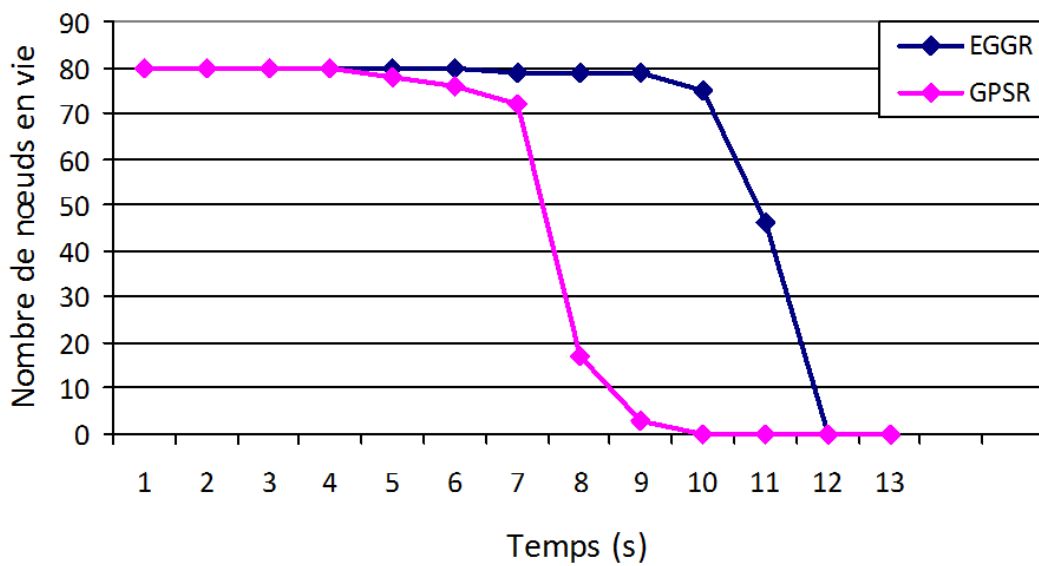


FIG. 4.7 – Durée de vie du réseau : Nombre de nœuds en vie.

La latence moyenne

On remarque d'après la figure 4.8 que la latence moyenne des paquets entre la source et la station de base varie entre 5,5 m-secondes et 7,5 m-secondes pour les deux protocoles. On remarque aussi que lorsque le nombre de nœuds est inférieur ou égale à 80, la latence moyenne obtenue avec notre protocole est meilleure que celle obtenue avec le protocole GPSR. Cependant, lorsque le nombre de nœuds est supérieur à 80, la latence moyenne obtenue avec le protocole GPSR est meilleure que celle obtenue avec notre protocole. Et cela revient au mécanisme de recouvrement appliqué par le protocole GPSR qui oblige le paquet de visiter plusieurs nœuds intermédiaires pour sortir d'un trou. On remarque aussi que la latence moyenne marquée par notre protocole augmente régulièrement avec le nombre de nœuds. Et cela se justifie, par le nombre de nœuds visités par le paquet le long de son chemin de bout en bout ; plus il y a de nœuds dans le réseau, plus il y a de nœuds dans le chemin de bout en bout. D'après ces résultats, on peut constater que notre protocole EGGR est meilleur lorsque le nombre de nœuds est inférieur à 80, par contre GPSR est meilleur lorsque le nombre de nœuds est supérieur à 80.

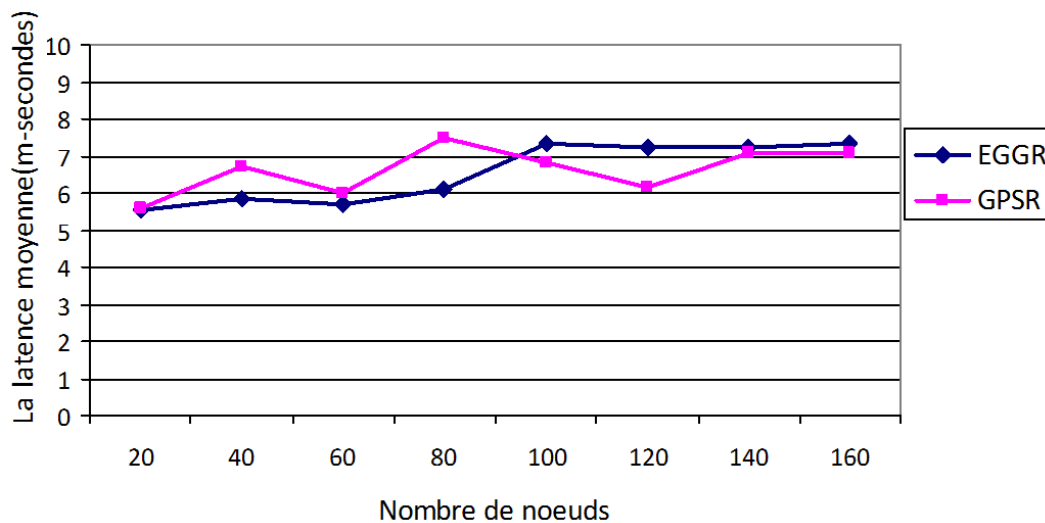


FIG. 4.8 – La latence moyenne.

4.6 Conclusion

Dans ce chapitre nous avons présenté une description détaillée d'un nouveau protocole de routage géographique pour les réseaux de capteurs sans fil qui minimise la

consommation d'énergie et qui garantit la livraison des paquets entre la source et la destination tout en exposant son principe de fonctionnement pour sélectionner le prochain nœud et son mécanisme pour éviter les trous. D'après les résultats de la simulation, on conclut que la forte densité du réseau permet d'augmenter le taux de livraison des paquets vers la station de base ; par contre, elle influe négativement sur la consommation d'énergie, ce qui réduit la durée de vie du réseau. En revanche, la faible densité du réseau permet de réduire la consommation d'énergie ; ce qui prolonge la durée de vie du réseau ; cependant, elle réduit le taux de livraison. Par ailleurs, les résultats de simulation ont prouvé que notre protocole réduit l'énergie consommée au niveau des nœuds capteurs conduisant ainsi à une prolongation de la durée de vie du réseau entier. De même, EGGR augmente le taux de livraison des données entre les nœuds et la station de base et surtout dans les réseaux denses. Cependant, notre nouveau protocole fournit un délai de transmission de bout en bout semblable à celui de GPSR.

Conclusion générale et Perspectives

La conception des réseaux de capteurs est fortement influencée par la limitation de la ressource énergétique disponible au niveau de chaque nœud. Pour cela, l'objectif primordial des concepteurs des réseaux de capteurs sans fil est l'augmentation de la durée de vie de ce type de réseaux. La plus grande quantité de la réserve d'énergie interne des nœuds est consommée lors de la communication. C'est pour cela que la communauté scientifique active beaucoup dans l'amélioration des protocoles de communication pour les réseaux de capteurs sans fil afin de prolonger leur durée de vie.

Dans cette optique, le routage géographique s'est présenté comme étant une solution prometteuse pour conserver l'énergie des nœuds, et faciliter la transmission des données capturées dans le réseau vers la station de base. Selon ce type de routage, les nœuds du réseau utilisent les informations de la position de chaque nœud voisin pour faire avancer les données vers la station de base.

Dans ce mémoire, nous avons proposé un nouveau protocole de routage géographique qui conserve l'énergie des nœuds et qui garantit la livraison des paquets dans les réseaux de capteurs sans fil en se basant uniquement sur les connaissances locales des nœuds capteurs (le voisinage). Ce protocole, nommé *EGGR (Energy-aware and delivery Guarantee Geographic Routing)* prend en considération les contraintes imposées par les nœuds capteurs, en permettant une gestion plus efficace de la ressource énergétique lors de la communication des données dans le réseau, à travers l'introduction d'un mécanisme pour calculer le plus court chemin en terme de consommation d'énergie pendant le calcul du prochain nœud. Par ailleurs, la garantie de livraison est assurée par l'introduction d'un mécanisme permettant d'éviter les trous.

La comparaison des performances de notre protocole EGGR et le protocole GPSR s'est faite moyennant le simulateur réseau Open source J-Sim. Ce dernier contient un module spécifique pour les réseaux de capteurs sans fil. Après avoir effectué plusieurs simulations et analysé les résultats obtenus, nous avons constaté que notre protocole EGGR réduit l'énergie consommée au niveau des nœuds capteurs conduisant ainsi à une prolongation de la durée de vie du réseau entier. La série de simulation a aussi

montré que notre protocole EGGR augmente le taux de livraison des données entre les nœuds et la station de base et notamment dans les réseaux denses. Néanmoins, notre protocole EGGR fournit un délai de transmission de bout en bout semblable à celui de GPSR.

Comme perspectives de notre travail, nous envisageons l'introduction de la notion d'erreur de positionnement pendant le calcul du prochain nœud. Dans notre travail, nous avons mis l'hypothèse que tous les nœuds capteurs connaissent leurs positions exactes. Or, la précision des mesures réalisées par les technologies de mesure (voir section 3.2.1.1) varie selon plusieurs paramètres liés à l'environnement du réseau : obstacles, degré d'humidité, vitesse du vent, ... Par exemple, les technologies RSSI ou AoA sont fortement déconseillées dans des milieux fermés. En effet, trouver une méthode qui garantit des positions précises quelque soit l'environnement du réseau semble difficile voire impossible sans un minimum de connaissances relatives à cet environnement. Pour cela, nous proposons d'introduire la notion de taux d'erreur de positionnement lors du calcul du plus court chemin.

Bibliographie

- [2.] The Network Simulator ns 2. "<http://www.isi.edu/nsnam/ns/>".
- [201] Tcl Developer Site. [En ligne] 2011. "<http://www.tcl.tk/>".
- [AAEI10] A. G. A.Elrahim, H. A.Elsayed, S. Elramly, and M. M. Ibrahim. "an energy aware wsn geographic routing protocol". *Universal Journal of Computer Science and Engineering Technology*, 1(4) :105–111, November 2010.
- [AKK04] J.N. Al-Karaki and A.E. Kamal. "routing techniques in wireless sensor networks : a survey". *IEEE Wireless Communications*, 11(6) :6–28, 2004.
- [and99] W. Heizelman, J. Kulik and H. Balakrishnan. "adaptive protocols for information dissemination in wireless sensor networks". *Proceedings of the 5th Annual ACM/IEEE International Conference on Mobile Computing and Networking*, pages 174–185, 1999.
- [and00a] W. Heizelman, A. Chandrakasan and H. Balakrishnan. "energy-efficient communication protocol for wireless mi-crosensor networks". *Proceedings of the 33rd Hawaii International Conference on System Sciences (HICSS 00)*, 2 :10, January 2000.
- [and00b] C. Intanagonwiwat, R. Govindan and D. Estrin. "dierected diffusion : a scalable and robust communication paradigm for sensor networks". *Proceedings of the 6th International Conference in Mobile Computing and Networking (Mobicom)*, pages 6–11, Aug. 2000.
- [ASGD08] A. Awad, C. Sommer, R. German, and F. Dressler. "virtual cord protocol (vcp) : A flexible dht-like routing service for sensor networks". *Preceedings of the 5th IEEE International Conference on Mobile Ad Hoc and Sensor Systems*, pages 133–142, 2008.
- [ASSC02] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. "a survey on sensor networks". *IEEE Communications Magazine*, 40(8) :102–114, 2002.
- [AV10] I. F. Akyildiz and M. C. Vuran. "*Wireless Sensor Networks*". John Wiley and Sons Ltd, 2010.

- [AY05] K. Akkaya and M. Younis. "a survey on routing protocols for wireless sensor networks". *Adhoc Networks*, 3(3) :325–349, 2005.
- [BCSW98] S. Basagni, I. Chlamtac, V. Syrotiuk, and B. A. Woodward. "a distance routing effect algorithm for mobility (dream)". *ACM/IEEE International Conference Mobile Computing and Networking, MOBICOM'98*, pages 76–84, 1998.
- [BGB01] L. Blazevic, S. Giordano, and J. Y. L. Boudec. "self organized terminode routing". *Technical Report DSC, Swiss Federal Institute of Technology*, 5 :2814–2818, 2001.
- [BMSU01] P. Bose, P. Morin, I. Stojmenovic, and J. Urrutia. "routing with guaranteed delivery in ad hoc wireless networks". *Wireless Networks*, 7 :609–616, 2001.
- [BP00] P. Bahl and V. N. Padmanabhan. "radar : An in-building rf-based user location and tracking system". *Proceedings of the IEEE INFOCOM 00*, 2 :775–784, March 2000.
- [CA06] Q. Cao and T. Abdelzaher. "liteos : A lightweight operating system for c++ software development in sensor networks". *Proceedings of the 4th International Conference on Embedded Networked Sensor Systems. ACM, New York, USA*, pages 361–362, 2006.
- [CASH08] Q. Cao, T. Abdelzaher, J. Stankovic, and T. He. "the liteos operating system : Towards unixlike abstractions for wireless sensor networks.". *Proceedings of the 7th International Conference on Information Processing in Sensor Networks. IEEE Computer Society, Washington, DC, USA*, pages 233–244, 2008.
- [CCDU05] A. Caruso, S. Chessa, S. De, and A. Urpi. "gps free coordinate assignment and routing in wireless sensor networks". *Proceedings of the 24th Annual Joint Conference of the IEEE Computer and Communications Societies*, 1 :150–160, 2005.
- [CGMT04] M. Conti, S. Giordano, G. Maselli, and G. Turi. "cross-layering in mobile ad hoc network design". *IEEE Computer*, 37(2) :48–51, 2004.
- [CHH01] S. Capkun, M. Hamdi, and J.-P. Hubaux. "gps-free positioning in mobile ad-hoc networks". *Proceedings of the 34th Hawaii International Conference on System Sciences*, page 10, 2001.
- [CWLY06] M. Chen, X. Wang, V. C. M. Leung, and Y. Yuan. "virtual coordinates based routing in wireless sensor networks". *Sensor Letters*, 4(3) :1–6, 2006.
- [D. 02] D. Braginsky and D. Estrin. "rumor routing algorithm for sensor network ". *Proceedings of the first Workshop on Sensor Networks and Application, Atlanta, USA*, pages 22–31, 2002.

- [DGV04] A. Dunkels, B. Gronvall, and T. Voigt. "contiki : A lightweight and flexible operating system for tiny networked sensors.". *Proceedings of the 29th Annual IEEE International Conference on Local Computer Networks. IEEE Computer Society, Washington, DC, USA*, pages 455–462, 2004.
- [Dij65] E. Dijkstra. "solution of a problem in concurrent programming control". *Communications of the ACM*, 8(9) :569, 1965.
- [DP10] W. Dargie and C. Poellabauer. *"Fundamentals of wireless sensor networks : theory and practice"*. John Wiley and Sons Ltd, 2010.
- [ea96] B. Parkinson et al. *"Global positioning system : Theory and application"*. American Institute of Aeronautics and Astronomy (AIAA), 1996.
- [eBN03] D. Niculescu et B. Nath. "ad hoc positioning system (aps) using aoa". *22nd Annual Joint Conference of the IEEE Computer and Communications Societies*, 3 :1734–1743, 2003.
- [EMSR07] E.H. Elhafsi, N. Mitton, and D. Simplot-Ryl. "cost over progress based energy efficient routing over virtual coordinates in wireless sensor networks". *IEEE International Workshop : From Theory to Practice in Wireless Sensor Networks (t2pWSN'07), Helsinki, Finland*, pages 1–6, 2007.
- [eUM76] J. Bondy et U. Murty. *"Graph theory with applications"*. Elsevier North-Holland, 1976.
- [eVL86] T. Hou et V. Li. "transmission range control in multihop packet radio networks". *IEEE Transactions on Communications*, (1) :38–44, 1986.
- [FGG⁺05] Q. Fang, J. Gao, L. J. Guibas, V. D. Silva, and L. Zhang. "glider : gradient landmark-based distributed routing for sensor networks". *24th Annual Joint Conference of the IEEE Computer and Communications Societies*, 1 :339–350, 2005.
- [Fin87] G. Finn. "routing and addressing problems in large metropolitanscale internetworks". *Technical Repport ISI/RR-87-180, University of Southern California, Marina del Rey : Information Sciences Institute*, pages xx–xx, 1987.
- [GLC07] D. Gay, Levis, and D. Culler. "software design patterns for tinyos". *ACM Transaction Embeded Computer Systems*, 6(4) :22, 2007.
- [HKS⁺05] C. C. Han, R. Kumar, R. Shea, E. Kohler, and M. Srivastava. "a dynamic operating system for sensor nodes". *Proceedings of the 3rd International Conference on Mobile Systems, Applications, and Services (MobiSys 05 :)*. ACM, New York, NY, USA, pages 163–176, 2005.
- [Jun08] D. Jungnickel. *"Graphs, Networks and Algorithms"*. Springer-Verlag Berlin Heidelberg, 2008.

- [K. 00] K. Sohrabi et al. "protocols for self-organization of a wireless sensor network". *IEEE Personal Communications*, 7(5) :16–27, 2000.
- [KB04] I. Khelladi and N. Badache. "les réseaux de capteurs : état de l'art, rapport de recherche ". *laboratoire LSI USTHB*, 2004.
- [KK00] B. Karp and H. T. Kung. "greedy perimeter stateless routing (gpsr) for wireless networks". *IEEE/MOBICOM International Conference on Mobile Computing and Networking*, pages 243–254, 2000.
- [KNS06a] J. Kuruvila, A. Nayak, and I. Stojmenovic. "progress and location based localized power aware routing for ad hoc sensor wireless networks". *International Journal of Distributed Systems. Sensor Networks*, 2 :147–159, 2006.
- [KNS06b] J. Kuruvila, A. Nayak, and I. Stojmenovic. "progress based localized power and cost aware routing algorithms for ad hoc and sensor wireless networks". *International Journal of Distributed Sensor Networks*, pages 147–159, 2006.
- [KV98] Y. B. Ko and N. H. Vaidya. "location-aided routing (lar) in mobile ad hoc networks". *ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom'98)*, 6(4) :307–321, October 1998.
- [KWZ03] F. Kuhn, R. Wattenhofer, and A. Zollinger. "worst-case optimal and average-case efficient geometric ad-hoc routing". in *4th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, pages 267–278, 2003.
- [LAG06] K. Liu and N. Abu-Ghazaleh. "aligned virtual coordinates for greedy routing in wsns". *IEEE International Conference on Mobile Adhoc and Sensor Systems (MASS)*, pages 377–386, 2006.
- [Lib] GloMoSim : Global Mobile Information Systems Simulation Library. "<http://pcl.cs.ucla.edu/projects/glomosim/>".
- [LL] P. Levis and N. Lee. "tossim : A simulator for tinyos networks".
- [LTS01] W.H. Liao, Y.C. Tseng, and J.P. Sheu. "grid : A fully location-aware routing protocols for mobile ad hoc networks". *Proceeding IEEE HICSS*, 1(3) :37–60, 2001.
- [MWHE01] M. Mauve, J. Widmer, H. Hartenstein, and N. Europs. "a survey on positionbased routing in mobile ad hoc networks". *IEEE Network*, 15(6) :30–39, 2001.
- [OT] Inc. OPNET Technologies. "<http://www.opnet.com/>".
- [pag] J-Sim Home page. "<http://sites.google.com/site/jsimofficial/>".

- [PR99] C. E. Perkins and E. M. Royer. "ad-hoc on-demand distance vector routing". *Proceedings of the 2nd IEEE Workshop on Mobile Computer Systems and Applications*, pages 90–100, 1999.
- [RM99] V. Rodoplu and T. Meng. "minimizing energy mobile wireless networks". *IEEE Journal on Selected Areas*, 17 :1333–1347, 1999.
- [Saa09] C. Saad. "quelques contributions dans les réseaux de capteurs sans fil : Localisation et routage". *Thèse de doctorat en informatique, Université d'Avignon et des Pays de Vaucluse, France*, 2009.
- [SCH⁺06] A. Sobeih, W. Chen, J. C. Hou, L. Kung, N. Li, H. Lim, H. Tyan, and H. Zhang. "j-sim : A simulation and emulation environment for wireless sensor networks". *IEEE Wireless Communications Magazine*, 13(4) :104–119, 2006.
- [SHS01] A. Savvides, C. C. Han, and M. B. Strivastava. "dynamic fine-grained localization in ad-hoc networks of sensors". *ACM SIGMOBILE*, pages 166–180, 2001.
- [SL01a] I. Stojmenovic and X. Lin. "loop-free hybrid single-path/flooding routing algorithms with guaranteed delivery for wireless networks". *IEEE Transaction on Parallel and Distributed Systems*, 12(10) :1023–1032, 2001.
- [SL01b] I. Stojmenovic and X. Lin. "power-aware localized routing in wireless networks". *IEEE Transactions On Parallel and Distributed Systems*, 12(11) :1122–1133, 2001.
- [SP00] K. Sohrabi and J. Pottie. "protocols for self-organization of a wireless sensor network". *IEEE Personal Communications*, 7(5) :16–27, 2000.
- [Sto02] I. Stojmenovic. "position-based routing in ad hoc networks". *IEEE Communications Magazine*, 40(7) :128–134, 2002.
- [Sys] OMNetpp Discrete Event Simulation System.
"http://www.omnetpp.org/".
- [TK84] H. Takagi and L. Kleinrock. "optimal transmission ranges for randomly distributed packet radio terminals". *IEEE Transactions on Communications*, 32(3) :246–257, 1984.
- [Tya02] H. Tyan. "design, realization and evaluation of a component-based compositional software architecture for network simulation". *Thesis report, The Ohio State University, USA*, 2002.
- [TYH07] M. J. Tsai, H. Y. Yang, and W. Q. Huang. "axis based virtual coordinate assignment protocol and delivery guaranteed routing protocol in wireless sensor networks". *26th IEEE International Conference on Computer Communications*, pages 2234–2242, 2007.

- [UKS99] J. Urrutia, E. Kranakis, and H. Singh. "compass routing on geometric networks". *Proceeding 11 th Canadian Conference on Computational Geometry, Vancouver*, pages 51–54, 1999.
- [xWG05] 2.x Working Group. "tt tinyos 2.0. sensys 05". *Proceedings of the 3rd International Conference on Embedded Networked Sensor Systems. ACM, New York, NY, USA*, pages xx–xx, 2005.
- [YHL⁺10] J. You, Q. Han, D. Lieckfeldt, J. Salzmann, and D. Timmermann. "virtual position based geographic routing for wireless sensor networks". *Computer Communications*, 33(3) :1255–1265, 2010.

Résumé

LES avancées récentes dans la technologie micro-électro-mécanique, les communications sans-fil et la micro-électronique, ont rendu possible le développement de capteurs multifonctionnels avec des coûts réduits capables de détecter, mesurer et rapporter des données physiques liées à l'environnement. Les réseaux de capteurs sont tout simplement un ensemble de ces capteurs, formant un réseau sans infrastructure. La petite taille de ces capteurs impose des limites sur leur capacité en termes d'énergie embarquée car ils sont généralement alimentés par des batteries non rechargeables et difficilement remplaçables. Et parce que la grande partie de l'énergie est consommée par le sous système radio (la communication), plusieurs solutions qui adressent tous les niveaux de la hiérarchie du réseau ont été proposées pour réduire la consommation d'énergie et étendre la durée de vie du réseau.

Dans ce travail, nous avons proposé un nouveau protocole de routage géographique qui conserve l'énergie des nœuds et qui garantit la livraison des paquets dans les réseaux de capteurs sans fil tout en se basant uniquement sur les connaissances locales des nœuds capteurs (le voisinage). Le protocole proposé *EGGR* (*Energy-aware and delivery Guarantee Geographic Routing*) permet une gestion plus efficace de la ressource énergétique lors de la communication des données dans le réseau, à travers l'introduction d'un mécanisme pour calculer le plus court chemin en termes de consommation d'énergie pendant le calcul du prochain nœud. La garantie de livraison est assurée par l'introduction d'un mécanisme permettant d'éviter les trous.

L'analyse de performances du protocole proposé EGGR montre qu'il réduit la consommation d'énergie et augmente le taux de livraison des paquets entre les nœuds capteurs et la station de base par rapport au protocole GPSR.

Mots clés : Réseaux de capteurs sans fil, Routage géographique, Localisation, Consommation d'énergie.

Abstract

RECENT advances in micro-electro-mechanic technology, wireless communication technologies, and in micro-electronic have enabled the development of multifunctional sensors at low cost capable of detecting, measuring, and reporting information about the environment. Wireless sensor networks are simply a set of these sensors, shaping a wireless network. The small size of the sensors confines the embedded energy because they are powered by non-rechargeable batteries, and even not easily replaceable. And because the most significant amount of energy is consumed by the sub-system radio (communication), several solutions, that address all network hierarchy layers, were proposed to reduce energy consumption and extend network lifetime.

The aim of this work is to propose a new geographic routing protocol that conserves nodes energy and that guarantees packets delivery in the wireless sensor network based only on local knowledge of sensor nodes (neighborhood). The proposed protocol EGGR (Energy-aware and delivery Guarantee Geographic Routing) manage energy resources efficiently during data communication in the network, by introducing a mechanism that calculates the shortest path in terms of energy consumption when setting up the next node. Delivery guaranty is assured by introducing a mechanism that avoids holes.

Our experimental results show that the proposed protocol EGGR decreases energy consumption and increases delivery rate between sensor nodes and the base station compared with GPSR protocol.

Keywords : Wireless sensor network, Geographic routing, Localization, Energy consumption.