

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

Mémoire de fin d'études

pour l'obtention du diplôme de Master en Informatique
(Réseaux et Systèmes Distribués)

Thème

Routage géographique multi-chemins dans les réseaux de capteurs sans fil

Réalisé par :

- KADDOURI Ahlem
- BOUDAOUUD Chahrazed

Présenté le 22 Juin 2016 devant le jury composé de :

- BENMAMMAR Badr (Président)
- BEKARA Chakib (Examineur)
- LEHSAINI Mohamed (Encadrant)
- CHIKH Asma (Co-encadrante)

Année universitaire : 2015-2016

Remerciement

Au terme de ce travail, on tient à exprimer notre profonde gratitude et nos sincères remerciements à Mr Lehsaini pour son encadrement, son soutien, ainsi que pour ses conseils instructifs durant toute la période de ce travail,

On remercie particulièrement notre co-encadreur Mlle Chikh qui malgré ses multiples occupations n'a ménagé aucun effort pour nous écouter et nous orienter depuis le choix du thème jusqu'à la fin de la présente étude.

Nos plus vifs remerciements s'adressent aussi à tout le cadre professoral et administratif de la formation Master spécialisé Réseaux et Système Distribuer (RSD), pour leurs patiences et savoir qui nous a illuminés durant ces deux années de formation.

En fin, on tient aussi à remercier tous les membres du jury qui nous ont fait l'honneur d'accepter de juger notre travail.

Pour tous ceux que qu'on a oubliés, on vous dit Merci.

Dédicaces

Je dédie ce modeste travail à celle qui m'a donné la vie, le symbole de tendresse, qui s'est sacrifiée pour mon bonheur et ma réussite, à ma Mère ...

A mon père, écolé de mon enfance, qui a été mon ombre durant toutes les années des études, et qui a veillé tout au long de ma vie à m'encourager, à me donner l'aide et à me protéger.

Que dieu les gardes et les protège.

A mes frères REDA et RAYENE et ma sœur ASMA.

A ma très chère grand-mère

Et a tous les membres de ma grande famille :

tantes, oncles, cousins, cousines.

A mon binôme CHAHRAZED et toute sa famille.

A tous mes enseignants

A tous mes amis qui j'ai passé avec eux des bons moments durant ces 5 ans.

Kaddourí Ahlem

Dédicaces

A peine nous venons de terminer la rédaction du mémoire de fin de cycle de Master, je voudrais très vite le dédier avec une immense joie, un grand honneur et un cœur chaleureux :

❧ *A ma très chère maman en signe de ma profonde et affectueuse reconnaissance; pour son amour sans mesure, pour tous les sacrifices, les soutiens, les tolérances et les encouragements qu'elle a bien voulu consentir pour moi. Tous les mots restent faibles pour exprimer mes sentiments envers elle.*

❧ *A Ma sœur Ahlam et son mari Amine;*

❧ *A mes tantes, oncles et cousin(e)s ;*

❧ *Enfin à tous mes ami(e)s et connaissances.*

Boudaoud Chahrazed

Table des matières

Table des matières	1
Liste des figures.....	3
Liste des abréviations	5
Introduction générale.....	6
Chapitre 1 Généralités sur les réseaux de capteurs sans fil	8
1.1 Introduction.....	8
1.2 Nœud capteur	8
1.2.1 Définition d'un capteur	8
1.2.2 Capteur intelligent.....	9
1.2.3 Composants d'un nœud capteur.....	9
1.3 Réseaux de Capteurs Sans Fil (RCSF)	11
1.3.1 Définition d'un RCSF.....	11
1.4 Types des RCSF	12
1.4.1 Réseaux de capteurs terrestres.....	12
1.4.2 Réseaux de capteurs souterrains	12
1.4.3 Réseaux de capteurs sous-marins	13
1.4.4 Réseaux de capteurs multimédias.....	13
1.4.5 Réseaux de capteurs mobiles.....	13
1.5 Domaines d'applications des RCSF.....	14
1.5.1 Applications militaires	14
1.5.2 Applications médicales.....	15
1.5.3 Applications environnementales.....	16
1.5.4 Applications domestiques	16
1.5.5 Applications agricoles.....	17
1.5.6 Applications industrielles	17
1.6 Communications dans les RCSF.....	18
1.6.1 Pile protocolaire d'un réseau de capteurs	18
1.6.2 Technologie de la communication dans les RCSF.....	19
1.7 Contraintes de conception des RCSF.....	19
1.8 Conclusion	21
Chapitre 2 Routage dans les réseaux de capteurs sans fil	22
2.1 Introduction.....	22

2.2	Contraintes de routage dans les réseaux de capteurs sans fil	22
2.3	Classification des protocoles de routage dans les RCSF.....	23
2.3.1	Protocoles plats.....	23
2.3.2	Protocoles basés sur la localisation (géographiques)	24
2.3.3	Protocoles hiérarchiques.....	25
2.4	Techniques de routage dans les RCSF	26
2.4.1	Routage mono-chemin	27
2.4.2	Routage multi-chemins	27
2.5	Les principaux protocoles multi-chemins.....	29
2.5.1	TPGF (Two Phase geographical Greedy Forwarding)	29
2.5.2	MPMPS (Multi-Priority Multi-Path Selection)	29
2.5.3	DGR (Directional Geographical Routing)	29
2.6	Conclusion	30
Chapitre 3	Implémentation : Une version améliorée de GPSR	31
3.1	Introduction.....	31
3.2	Outils de réalisation.....	31
3.2.1	OMNeT++ (Objective Modular Network Testbed in C++)	31
3.2.2	Castalia	34
3.3	Description de protocole GPSR	37
3.3.1	Transmission gloutonne (Greedy Forwarding).....	38
3.3.2	Transmission de périmètre (Perimeter Forwarding).....	39
3.4	Implémentation.....	39
3.4.1	Implémentation de GPSR_monchemin et GPSR_multichemin	41
3.4.2	Evaluation des performances de GPSR et de GPSR_Modifié	43
a)	Contexte d'exécution	43
b)	Evaluation du nombre de paquets envoyés.....	44
c)	Evaluation du nombre de paquets reçus.....	45
d)	Evaluation du taux d'erreurs	47
e)	Evaluation de la consommation d'énergie.....	47
f)	Evaluation de la latence	49
3.5	Conclusion	50
	Conclusion générale	51
	Bibliographie.....	52

Liste des figures

Figure 1.1: Anatomie d'un capteur intelligent Telosb de Crossbow [10].....	9
Figure 1.2: Architecture d'un nœud capteur [12].....	11
Figure 1.3: Réseau de capteur sans fil [14]	12
Figure 1.4: Applications des réseaux de capteurs sans fil [20]	14
Figure 1.6: Un réseau de capteurs dans le domaine médical [15]	15
Figure 1.7: Application environnementale	16
Figure 1.8: Application domotique [22].....	17
Figure 1.9 : Pile protocolaire dans un réseau de capteurs sans fil [23]	18
Figure 2.1 : Protocoles de routage dans les RCSF [25].....	23
Figure 2.2 : Routage plat	24
Figure 2.3 : Routage hiérarchique	25
Figure 2.4 : Routage multi-chemins [32]	27
Figure 3.1 : Architecture OMNeT++ [6].....	32
Figure 3.2 : Modèle Castalia [8].....	35
Figure 3.3 : Composition d'un nœud dans Castalia [8].....	35
Figure 3.4 : Hiérarchie des dossiers dans Castalia	37
Figure 3.5 : Transmission gloutonne [4]	38
Figure 3.6 : Présence d'un trou [4].....	39
Figure 3.7 : Transmission de périmètre [4]	39
Figure 3.8 : Fichier .ini	41
Figure 3.9 : Un exemple d'exécution avec déploiement des nœuds	44
Figure 3.10 : Nombre de paquets transmis en mode mono-chemin	44
Figure 3.11 : Nombre de paquets transmis en mode multi-chemin	45
Figure 3.12 : Nombre de paquets transmis dans les deux modes	45
Figure 3.13 : Nombre de paquets reçus et perdus en mode mono-chemin	46
Figure 3.14 : Nombre de paquets reçus et perdus en mode multi-chemin	46
Figure 3.15 : Nombre de paquets reçus dans les deux modes.....	46
Figure 3.16 : Taux de paquets perdus dans les deux modes.....	47
Figure 3.17 : Consommation d'énergie en mode mono-chemin	48
Figure 3.17 : Consommation d'énergie en mode multi-chemin	48

Figure 3.19 : Consommation d'énergie dans les deux modes..... 49

Figure 3.20 : Calcul de la latence dans les deux modes 49

Liste des abréviations

ADC	<i>Analog to Digital Converter</i>
BAN	<i>Body Area Network</i>
CH	<i>Cluster Head</i>
DD	<i>Directed Diffusion</i>
DGR	<i>Directional Geographical Routing</i>
FEC	<i>Forward Error Correction</i>
GEAR	<i>Geographic and Energy Aware Routing</i>
GF	<i>Greedy Forwarding</i>
GNED	<i>Graphical Network Descriptor</i>
GPS	<i>Global Positioning System</i>
GPSR	<i>Greedy Perimeter Stateless Routing</i>
GUI	<i>Graphical User Interface</i>
LEACH	<i>Low-Energy Adaptive Clustering Hierarchy</i>
MAC	<i>Medium Access Control</i>
MPMPS	<i>Multi-Priority Multi-Path Selection</i>
OMNet++	<i>Objective Modular Network Testbed in C++</i>
PF	<i>Perimeter Forwarding</i>
RCSF	<i>Réseau de Capteurs Sans Fil</i>
TEEN	<i>Threshold sensitive Energy Efficient sensor Network protocol</i>
TPGF	<i>Two Phase geographical Greedy Forwarding</i>
WSN	<i>Wireless Sensor Network</i>

Introduction Générale

Introduction générale

Le paradigme de l'informatique ubiquitaire (ubiquitous computing) et le nouveau concept de l'internet des objets (internet of things) sont devenus désormais une réalité et ont connu un essor remarquable ces dernières années en s'imposant progressivement et sûrement dans notre quotidien. Aujourd'hui, le savoir acquis dans les technologies des microsystèmes électromécaniques, des communications sans fil et de l'électronique numérique a permis la conception et le développement de nœuds capteurs de petite taille à faible coût de fabrication et à faible consommation d'énergie. Ces nœuds capteurs sont devenus des éléments incontournables dans tous les systèmes où les informations issues de l'environnement extérieur sont nécessaires pour évaluer et agir. De ce fait, avoir une connaissance précise et complète sur ce milieu exige le déploiement de plusieurs nœuds capteurs formant ainsi un réseau appelé réseau de capteurs sans fil (RCSF).

Le champ d'applications des réseaux de capteurs est de plus en plus élargi grâce aux évolutions techniques et l'évolution des technologies des communications sans fil telles que Bluetooth [1] et Zigbee [2]. Cette évolution a permis aux réseaux de capteurs sans fil d'envahir une gamme variée de domaines d'applications telles que le domaine militaire, le domaine de la médecine, le domaine environnemental, le secteur industriel, etc.

Le routage des données dans un RCSF représente la fonctionnalité la plus importante du réseau. Il doit prendre en considération toutes les caractéristiques des capteurs afin d'assurer les meilleures performances du système : durée de vie, fiabilité, temps de réponse, ... etc.

Il existe deux techniques de routage dans les réseaux de capteurs sans fil qui sont le routage mono-chemin et le routage multi-chemins. Le routage mono-chemin permet seulement d'acheminer les paquets sur un seul chemin, ce qui ne répond pas aux exigences de qualité de service des applications multimédias en termes de débits et de délai de transmission par contre le routage multi-chemins sert à router les données sur plusieurs chemins. Le routage multi-chemins est un domaine de recherche actif sur les protocoles de routage avec gestion de la qualité de service pour les réseaux filaires et sans fil.

Les protocoles de routage les plus étudiés dans les réseaux de capteurs sans fil sont les protocoles de routage géographiques, ces protocoles se basent sur le fait que tous les nœuds connaissent leur position, par exemple, grâce à un équipement GPS (Global Positioning

System) ou encore par un système de positionnement distribué. Il existe plusieurs protocoles qui appartiennent à cette famille, parmi lesquels nous pouvons citer : GEAR [3], GPSR [4] et TPGF [5], etc.

Dans le cadre de ce projet de fin d'études, nous avons choisi de nous focaliser sur le protocole GPSR (Greedy Perimeter Stateless Routing) où une version modifiée de ce protocole géographique est proposée qui utilise une technique de routage multi-chemins contrairement au GPSR traditionnel qui utilise un schéma de routage mono-chemin.

Le manuscrit s'articule autour de trois chapitres :

Le premier chapitre de ce manuscrit présente des généralités sur les réseaux de capteurs sans fil, leurs caractéristiques et leurs domaines d'applications.

Le deuxième chapitre présente un état de l'art des principaux protocoles de routage dans les réseaux RCSF.

Le troisième chapitre constitue le cœur de notre travail, où nous présentons le simulateur OMNet++ (Objective Modular Network Testbed in C++) [6,7] et le simulateur Castalia [8] avec une brève description de leurs composants, leurs principaux caractéristiques et fonctionnalités ainsi leurs exploitations pour mettre en place l'application qui fait l'objet de notre projet de fin d'études.

Ce travail est terminé par une conclusion générale dans laquelle nous synthétisons l'objet de ce projet de fin d'études et nous présentons quelques perspectives.

Chapitre 1

Généralités sur les réseaux de capteurs sans fil

Chapitre 1

Généralités sur les réseaux de capteurs sans fil

1.1 Introduction

Les avancées technologiques dans le domaine de la micro-électronique, de la micromécanique et des communications sans fil ont donné naissance à une nouvelle génération de réseaux appelés réseaux de capteurs sans fil (RCSF) (Wireless Sensor Networks : WSNs en anglais). Ces derniers forment un ensemble de dispositifs électroniques miniaturisés autonomes, alimentés par une batterie et pouvant communiquer entre eux sans fil. Ces dispositifs appelés nœuds capteurs, ont pour mission de récolter des données et les transmettre à un nœud spécifique nommé « station de base ».

Les RCSF sont très utilisés dans plusieurs domaines allant du domaine militaire au domaine médical, en passant par l'industrie, l'écologie, la domotique, etc. Pour cela mener un travail de recherche dans ce domaine nécessite la connaissance d'un certain nombre de concepts généraux.

Dans ce chapitre, nous présentons les réseaux de capteurs sans fil avec un plan méthodologique que nous avons adopté. Nous commençons par une définition d'un capteur, son architecture et voir comment ces derniers sont déployés pour former un réseau de capteurs sans fil. Ensuite, les types des RCSF, les domaines d'application des réseaux de capteurs sans fil, ainsi que les différents facteurs de conception et les spécificités des RCSF seront étudiés, et nous terminons par une conclusion.

1.2 Nœud capteur

1.2.1 Définition d'un capteur

Un nœud capteur (dit "mote" en anglais) est un petit dispositif électronique doté d'un ou plusieurs capteurs qui sont capables de mesurer une valeur physique environnementale

(température, lumière, pression, etc.) ou physiologique (glycémie, tension, etc.), et de la communiquer à un centre de contrôle via une station de base. Plusieurs types de capteurs ont été développés dont les plus connus sont les suivants : les capteurs de distance, les capteurs tactiles, les capteurs de vision, les capteurs de température, les capteurs d'humidité, etc [9].

1.2.2 Capteur intelligent

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



Figure 1.1: Anatomie d'un capteur intelligent Telosb de Crossbow [10]

1.2.3 Composants d'un nœud capteur

Un nœud capteur est composé de quatre unités de base [9] (voir la Figure 1.2) :

- **Unité d'acquisition:** appelé aussi unité de captage, est généralement composée de deux sous-unités: les capteurs et les convertisseurs analogique-numériques (ADCs¹). Les capteurs obtiennent des mesures numériques sur les paramètres environnementaux et les transforment en signaux analogiques. Les ADCs convertissent ces signaux analogiques en signaux numériques pour être envoyé à l'unité de traitement.
- **Unité de traitement :** est composée de deux interfaces: une interface pour l'unité d'acquisition et une interface pour l'unité de transmission. Cette unité est également composée d'un processeur qui supporte un système d'exploitation spécifique. Elle acquiert les informations en provenance de l'unité d'acquisition et les envoie à l'unité de transmission. Son rôle est de contrôler le bon fonctionnement des autres unités sur certains capteurs. Elle peut embarquer un système d'exploitation pour faire fonctionner le nœud capteur. Elle peut aussi être couplée à une unité de stockage.
- **Unité de communication:** elle est responsable de la transmission et la réception des données via un support de communication sans fil et une antenne. On note que l'unité de transmission consomme beaucoup d'énergie par rapport à l'unité de traitement.
- **Unité d'énergie:** un des composants les plus importants d'un nœud sans fil est la source d'énergie. Habituellement l'alimentation par batterie est utilisée. Les batteries utilisées peuvent d'être rechargeables par l'énergie solaire ou non dans le cas où les capteurs sont déployés à l'abri du soleil comme dans les deux pôles (sud et nord). Souvent, dans les environnements hostiles il est impossible de recharger ou changer une batterie. L'unité d'énergie est responsable de l'alimentation des différentes unités et elle réduit les dépenses, par exemple en mettant en veille les composants inactifs.

Un nœud capteur peut contenir également, suivant son domaine d'application, des modules supplémentaires tels qu'un système de localisation (GPS²), ou bien un système générateur d'énergie (cellule solaire) [11]. On peut même trouver des micro-capteurs, un peu plus volumineux, dotés d'un système mobilisateur chargé de déplacer le micro-capteur en cas de nécessité comme les micro-capteurs déplacés par des drones.

¹ ADC : Analog to Digital Converter (convertisseur analogique numérique)

² GPS : *Global Positioning System*

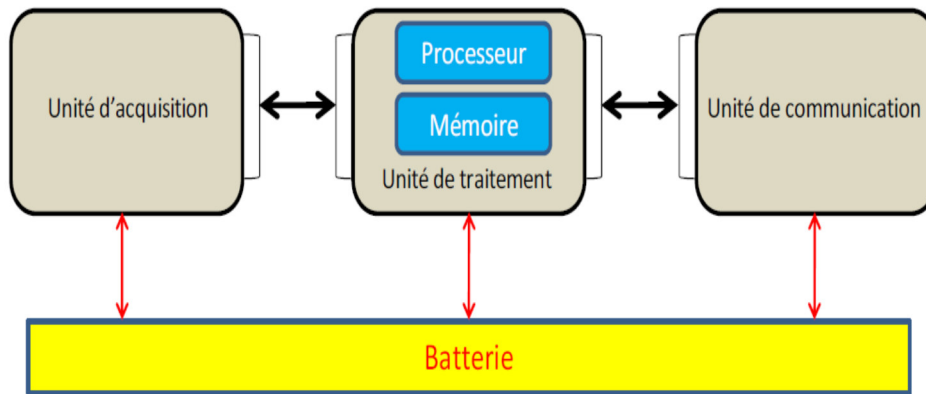


Figure 1.2: Architecture d'un nœud capteur [12]

1.3 Réseaux de Capteurs Sans Fil (RCSF)

1.3.1 Définition d'un RCSF

Un réseau de capteur sans fil (RCSF) [13], consiste en un ensemble de nœuds capteurs variant de quelques dizaines d'éléments à plusieurs milliers, placés de manière plus ou moins aléatoire, dans une zone géographique appelée zone de captage, ou zone d'intérêt, afin de surveiller un phénomène physique et de récolter leurs données d'une manière autonome. Les nœuds capteurs utilisent une communication sans fil pour acheminer les données captées vers un nœud collecteur appelé nœud puits (Sink en anglais), ou station de base. Le puits transmet ensuite ces données par Internet ou par satellite à l'ordinateur central pour analyser ces données et prendre des décisions. Ainsi, l'utilisateur peut adresser des requêtes aux autres nœuds du réseau, précisant le type de données requises, puis récolter les données environnementales captées par le biais du nœud puits. Dans le cas le plus simple, les nœuds capteurs seront dans le voisinage direct du puits (communication à un saut). Cependant, dans le cas d'un réseau à grande échelle, ils ne sont pas tous dans le voisinage du puits et les données seront acheminées du nœud source vers le puits en transitant par plusieurs nœuds selon un mode de communication multi-sauts comme l'illustre la figure 1.3.

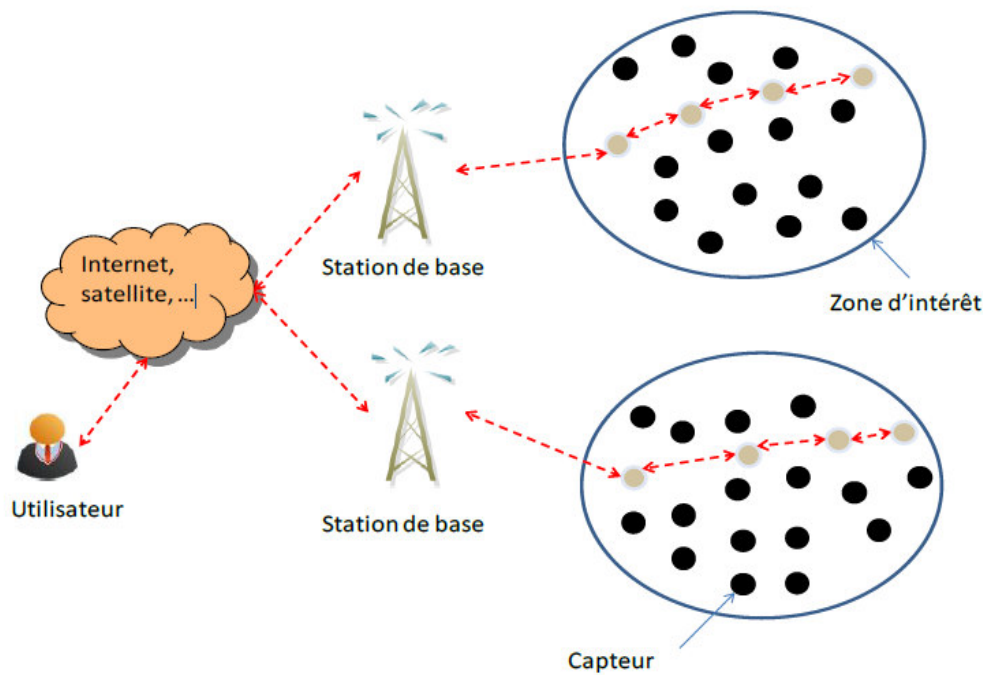


Figure 1.2: Réseau de capteur sans fil [14]

1.4 Types des RCSF

Les réseaux de capteurs actuels sont déployés sur terre, sous terre et sous l'eau. Selon l'environnement, un réseau de capteurs est confronté à différents défis et contraintes. Il existe cinq types de réseaux de capteurs [15]: terrestre, souterrain, sous-marin, multimédia et mobile.

1.4.1 Réseaux de capteurs terrestres

Dans un réseau de capteurs terrestre, la communication fiable dans un environnement dense est très importante. Les nœuds capteurs terrestres doivent être capables de communiquer efficacement les données vers la station de base. Alors que l'énergie de la batterie est limitée et ne peut être rechargeable, les nœuds capteurs terrestres peuvent être cependant équipés d'une source d'alimentation secondaire telle que les cellules solaires [11].

1.4.2 Réseaux de capteurs souterrains

Les réseaux de capteurs souterrains [16,17] se composent d'un nombre de nœuds capteurs enfouis sous terre ou dans une grotte ou une mine utilisés pour surveiller les conditions souterraines. Les nœuds stations de base additionnels sont situés au-dessus du sol pour relayer l'information à partir des nœuds capteurs vers la station de base. Un réseau de capteurs souterrain est plus cher qu'un réseau de capteurs terrestre en termes d'équipement, de déploiement et de maintenance. Les nœuds capteurs souterrains sont chers parce que les pièces

d'équipements appropriés doivent être choisies pour assurer une communication fiable à travers le sol, les roches, l'eau et autres contenus minéraux.

1.4.3 Réseaux de capteurs sous-marins

Les réseaux de capteurs sous-marins [18] se composent d'un nombre de nœuds capteurs et des véhicules déployés sous l'eau. A la différence des réseaux de capteurs terrestres, les nœuds capteurs sous-marins sont plus chers et sont moins déployés. Les véhicules sous-marins autonomes sont utilisés pour l'exploration ou la collecte des données de nœuds capteurs. Comparé à un déploiement dense des nœuds capteurs dans un réseau de capteurs terrestre, un déploiement clairsemé de nœuds capteurs est placé sous l'eau. Les communications sans fil sous-marines sont établies par transmission d'ondes acoustiques. Un défi en communication acoustique sous-marine est la bande passante limitée, le long temps de propagation, et le fading du signal issu. Un autre défi est la défaillance rapide des nœuds capteurs en raison des conditions environnementales.

1.4.4 Réseaux de capteurs multimédias

Les réseaux de capteurs multimédias [19] ont été proposés pour permettre la surveillance et le suivi des événements dans la forme de multimédia comme la vidéo, l'audio et l'image. Les réseaux de capteurs multimédia se composent d'un certain nombre de nœuds capteurs à faible coût équipés de caméras et des microphones. Ces nœuds capteurs interconnectent les uns avec les autres via une connexion sans fil pour la restitution, le traitement, la corrélation et la compression de données. Les nœuds capteurs multimédias sont déployés de manière pré-planifiée dans l'environnement pour garantir une couverture. Les défis dans les réseaux de capteurs multimédias comprennent une forte demande de bande passante, une forte consommation d'énergie, une qualité de service (QoS), des techniques de traitement et de compression de données et une conception inter-couche (cross-layer design).

1.4.5 Réseaux de capteurs mobiles

Les réseaux de capteurs mobiles sont constitués d'un ensemble de nœuds capteurs qui peuvent se déplacer par leurs propres moyens et d'interagir avec l'environnement physique. Comme les nœuds statiques, les nœuds mobiles ont la capacité de collecter, calculer et communiquer. Une différence clé est que les nœuds mobiles ont la possibilité de repositionner et de s'organiser en réseau.

Un réseau de capteurs mobiles peut commencer avec un déploiement initial et les nœuds peuvent alors s'étaler pour recueillir des informations. L'information recueillie par un nœud mobile peut être communiquée à un autre nœud mobile quand l'un se trouve à la même portée que l'autre. Une autre différence essentielle est la distribution des données [15].

1.5 Domaines d'applications des RCSF

La miniaturisation des capteurs, le coût de plus en plus faible, la large gamme des types de capteurs disponibles ainsi que le support de communication sans fil utilisé, permettent aux réseaux de capteurs de se développer dans plusieurs domaines d'applications. La figure 1.4 montre quelques domaines d'applications des RCSF. Parmi ces domaines, nous citons les domaines: militaire, l'environnemental, domestique, médical et agricole.

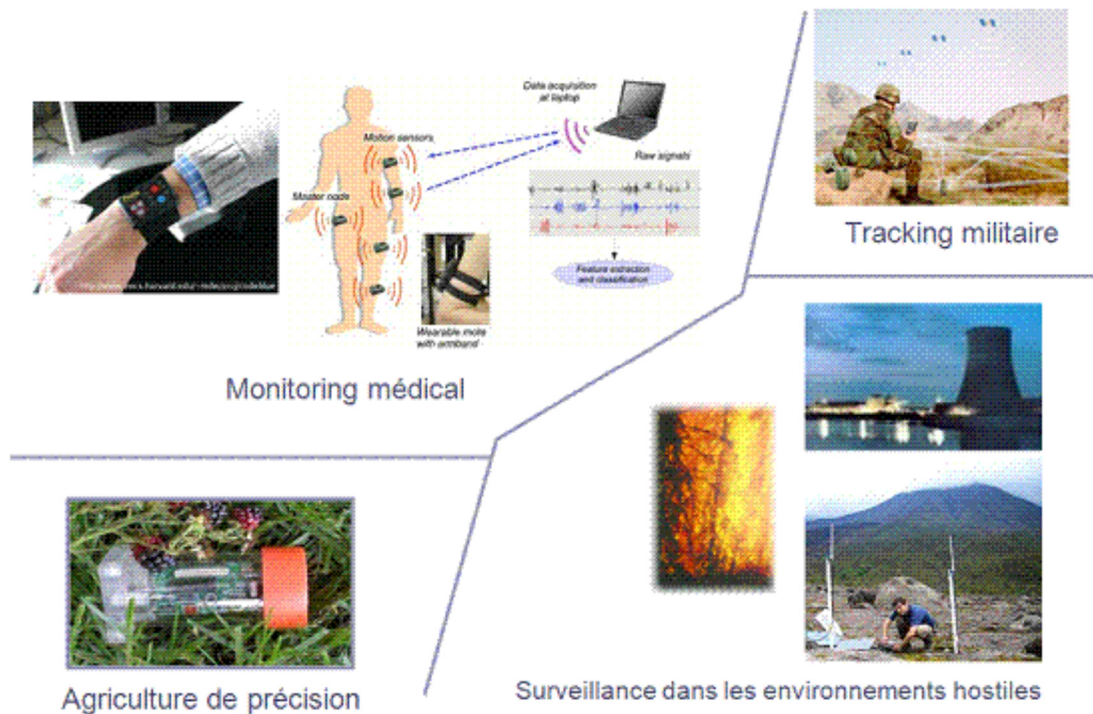


Figure 1.4: Applications des réseaux de capteurs sans fil [20]

1.5.1 Applications militaires

Les RCSF deviennent une partie intégrante de commandement militaire, de contrôle, de la communication et des systèmes de renseignement. Les capteurs sans fil peuvent être rapidement déployés dans un champ de bataille ou dans une région hostile sans aucune infrastructure. En raison de sa facilité de déploiement, d'auto-configurabilité et de sa tolérance aux pannes, les réseaux de capteurs joueront un rôle plus important dans les futurs systèmes

militaires et rendront les futures guerres plus "intelligentes" avec moins d'implication humaine [21].



Figure 1.5: Les réseaux de capteurs: Applications militaires [21]

1.5.2 Applications médicales

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

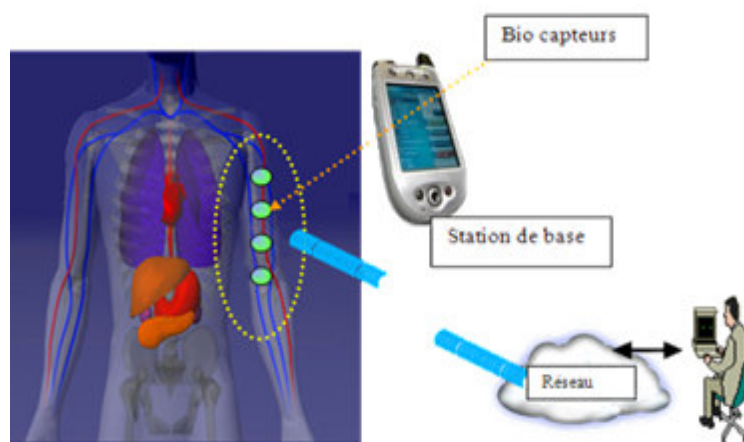


Figure 1.6: Un réseau de capteurs dans le domaine médical [15]

1.5.3 Applications environnementales

Les applications d'environnement qui peuvent bénéficier de la technologie des réseaux de capteurs sans fil, on peut citer par exemple, le cheminement des mouvements d'oiseaux; les macro-instruments utilisés pour la surveillance des terrains à grande échelle et les explorations planétaires; la détection chimique et biologique.



Figure 1.7: Application environnementale

1.5.4 Applications domestiques

Les RCSF peuvent être utilisés pour surveiller des habitations et contribuer au confort domestique, en transformant les logements en environnements intelligents dont les paramètres (température, pression, humidité, luminosité, etc..) s'adaptent automatiquement au comportement des individus.

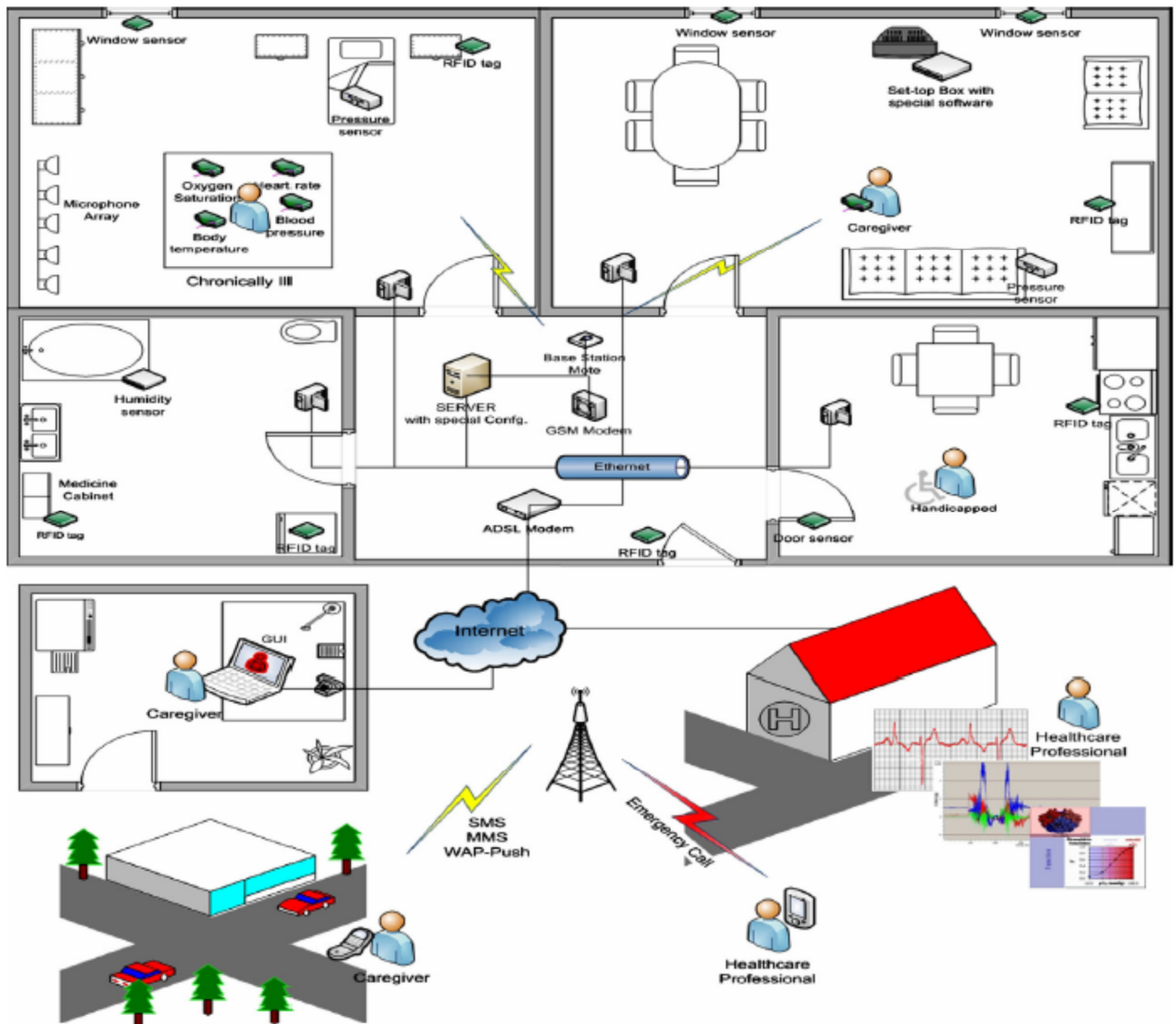


Figure 1.8: Application domotique [22]

1.5.5 Applications agricoles

Dans les champs agricoles, les capteurs peuvent être semés avec les graines. Ainsi, les zones sèches seront facilement identifiées et l'irrigation sera donc plus efficace.

1.5.6 Applications industrielles

Les capteurs peuvent être utilisés pour observer des lignes de production pour repérer des pannes éventuelles de machines ou même de prédire des pannes en surveillant l'état des machines.

1.6 Communications dans les RCSF

1.6.1 Pile protocolaire d'un réseau de capteurs

La pile protocolaire utilisée par les RCSF est illustrée dans la figure 1.9. Cette pile comprend cinq couches, une couche d'application, une couche de transport, une couche réseau, une couche de liaison de données et une couche physique. Ainsi que trois niveaux qui sont: le niveau de gestion d'énergie, le niveau de gestion de la mobilité et le niveau de gestion des tâches [23].

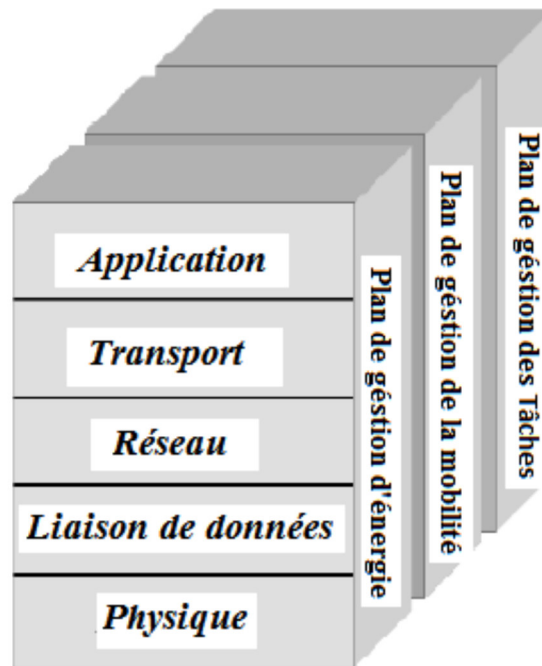


Figure 1.9 : Pile protocolaire dans un réseau de capteurs sans fil [23]

Le rôle de chacune de ces couches est résumé comme suit :

- **Couche physique** : matériels pour envoyer et recevoir les données.
- **Couche liaison de données** : gestion des liaisons entre les nœuds et la station de base, contrôle d'erreurs.
- **Couche réseau** : routage et transmission de données.
- **Couche transport** : transport de données, contrôle de flux.
- **Couche application** : interface pour les applications au haut niveau.
- **Plan de gestion d'énergie** : contrôle la consommation d'énergie d'un nœud.
- **Plan de gestion de mobilité** : surveille la mobilité des nœuds de capteurs.
- **Plan de gestion de tâche** : assure la distribution des tâches pour les nœuds de capteurs.

1.6.2 Technologie de la communication dans les RCSF

- **Bluetooth / IEEE**

Bluetooth [1] est un standard de communication sans fil qui fournit une configuration spéciale de pico-réseau maître/esclave avec le maximum de huit unités actives. Il supporte les connexions spontanées entre les périphériques sans leur exigeant des connaissances détaillées sur l'autre. Malheureusement, un grand défaut de cette technologie est sa trop grande consommation d'énergie. En plus, elle ne peut pas être utilisée pour les RCSF denses.

- **ZigBee / IEEE**

ZigBee [2] est une norme de transmission de données sans fil permettant la communication de machine à machine. ZigBee offre des débits de données moindres, mais sa très faible consommation d'énergie et son coût de production très bas en font d'elle une candidate idéale pour la domotique et le secteur industriel.

1.7 Contraintes de conception des RCSF

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

- **Tolérance aux pannes :** Certains nœuds peuvent générer des erreurs ou ne plus fonctionner à cause d'un manque d'énergie, un problème physique ou une interférence. Ces anomalies ne doivent pas affecter le reste du réseau, c'est le principe de la tolérance aux pannes. Cette contrainte mesure la capacité de maintenir les fonctionnalités du réseau sans interruption dues à une panne intervenue sur un ou plusieurs capteurs.
- **Passage à l'échelle :** La plupart des protocoles conçus pour RCSF ont été établis pour des réseaux dont le nombre de nœuds est moyen. Cependant, les performances de ces protocoles risquent de se dégrader lorsque le nombre de nœuds augmente. Ce principe est appelé le passage à l'échelle. D'où on dit qu'un protocole s'adapte au passage à l'échelle si et seulement si quand le nombre de nœuds augmente dans le réseau, les performances de ce dernier ne doivent pas se dégrader.
- **Coût de production :** Souvent, les réseaux de capteurs sont composés d'un très grand nombre de nœuds. Le prix d'un est critique afin de pouvoir construire un réseau de

surveillance traditionnel. Actuellement, un nœud ne coûte souvent pas beaucoup plus que 1\$.

- **Environnement :** Les nœuds capteurs doivent être conçus d'une manière à résister aux sévères conditions de l'environnement: forte chaleur, pluie, humidité ...
- **Medias de transmission :** Dans un réseau de capteurs, les nœuds sont reliés par une architecture sans fil. Pour permettre des opérations sur ces réseaux dans le monde entier, le média de transmission doit être normé. On utilise le plus souvent l'infrarouge, le Bluetooth et les communications radio Zigbee.
- **Consommation d'énergie:** Les capteurs sont généralement alimentés par des batteries et dans certaines applications ils sont déployés dans des zones hostiles où l'intervention humaine est presque impossible pour toute opération de maintenance. De ce fait, le remplacement de la batterie est impossible. Par ailleurs, la durée de vie d'un capteur dépend grandement de la durée de vie de la batterie. Cette énergie est consommée par les différentes unités du capteur afin de réaliser les tâches de captage, traitement de données et de communication. Cette dernière est l'opération qui consomme le plus d'énergie, c'est pour cette raison que les recherches actuelles se concentrent principalement sur les moyens de réduire cette consommation pour garantir une longue durée de vie aux réseaux déployés.
- **Contraintes matérielles:** La principale contrainte matérielle est la taille du capteur. La petite taille des capteurs ne permet de les doter par d'autres unités telles qu'un grand nombre de batteries et un mobilisateur.
- **Topologie de réseau:** Le déploiement d'un grand nombre de nœuds nécessite une maintenance de la topologie. Cette maintenance consiste en trois phrases:
 - Déploiement.
 - Post-déploiement.
 - Redéploiement de nœuds additionnels.

1.8 Conclusion

Dans ce chapitre, nous avons présenté en premier lieu quelques généralités sur les réseaux de capteur sans fil, à savoir la description d'un réseau de capteurs sans fil, leur son architecture et ses types. En outre, nous avons présenté leurs principaux domaines d'applications et les défis de conception matérielle et logicielle auxquels ils sont confrontés.

Chapitre 2

Routage dans les réseaux des capteurs sans fil

Chapitre 2

Routage dans les réseaux de capteurs sans fil

2.1 Introduction

Une des principales fonctionnalités dans la conception de RCSF est le routage. Le problème de routage consiste à déterminer et à maintenir le chemin le plus adapté pour faire transiter les paquets de données d'une source vers une destination. Cette fonctionnalité doit ainsi prendre en considération les propriétés inhérentes des capteurs (contraintes énergétiques et celles de calcul et de capacité mémoire) afin d'assurer les meilleures performances du système : durée de vie, fiabilité, temps de réponse, etc.

Cependant, diverses contraintes liées aux réseaux de capteurs impliquent la conception de nouveaux protocoles de routage qui diffèrent de ceux dédiés aux réseaux ad hoc.

Nous présenterons dans ce chapitre un état de l'art des principaux protocoles de routage dans les réseaux RCSF car la présentation de ces protocoles nous permettra de mieux analyser leur fonctionnement.

2.2 Contraintes de routage dans les réseaux de capteurs sans fil

Le routage dans les réseaux de capteurs diffère de celui des réseaux ad hoc dans les points suivants:

- Il n'est pas possible d'établir un système d'adressage global pour le grand nombre de nœuds.
- Les applications des réseaux de capteurs exigent l'écoulement de données mesurées depuis des sources multiples vers la destination finale « station de base ».
- Les différents capteurs peuvent générer les mêmes données à proximité d'un phénomène (problème de la redondance des données).

- Les nœuds capteurs exigent ainsi une gestion soignée des ressources en particulier la ressource d'énergie.

En effet, les protocoles de routage dans les réseaux de capteurs doivent [24]:

- Etre basés sur un échange réduit de messages vus les contraintes énergétiques et celles de calcul et de capacité mémoire.
- Supporter un nombre élevé de capteurs dont les adresses peuvent être inconnues lors du déploiement.
- Etre robustes puisque tous les nœuds capteurs peuvent jouer le rôle de routeurs.
- Opérer dans des topologies qui soient même aléatoires.

2.3 Classification des protocoles de routage dans les RCSF

Selon la structure de réseau, les protocoles de routage dans les réseaux de capteurs sans fil peuvent être découpés en trois familles: les algorithmes de routage plats, hiérarchiques ou géographiques (voir la figure 2.1).

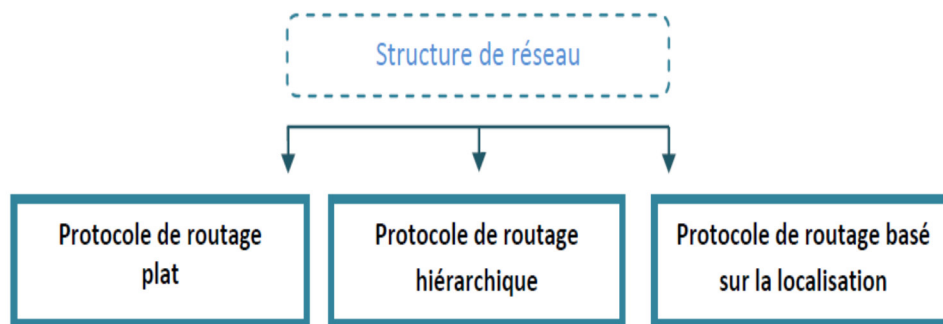


Figure 2.1 : Protocoles de routage dans les RCSF [25]

2.3.1 Protocoles plats

Le routage plat est le modèle le plus simple où tous les capteurs sont égaux et jouent les mêmes rôles, c'est à dire ont les mêmes fonctions à exécuter sauf la station de base qui est chargée de collecter toutes les informations issues des différents nœuds capteurs pour les transmettre vers l'utilisateur final. La décision d'un nœud de router des paquets vers un autre dépendra de sa position et pourra être remise en cause au cours du temps.

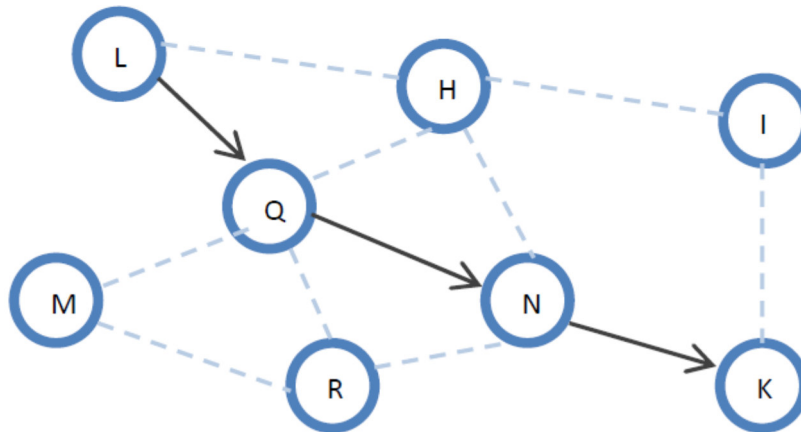


Figure 2.2 : Routage plat

2.3.2 Protocoles basés sur la localisation (géographiques)

Le routage géographique met l'accent sur le fait que chaque nœud connaît ses coordonnées géographiques et utilise celles de la destination finale d'un paquet pour les décisions de routage. Dans ce qui suit on va présenter quelques protocoles de routage géographiques dans les RCSF :

a) GEAR (Geographic and Energy Aware Routing)

Yu et al. [3] ont suggéré l'utilisation de l'information géographique tout en diffusant les requêtes pour s'approprier les régions puisque les requêtes de données comprennent souvent des attributs géographiques. Le protocole, nommé GEAR, utilise les heuristiques de l'énergie et de la sélection du voisin informé géographiquement pour acheminer un paquet vers la région cible. L'idée est de limiter le nombre des intérêts dans le protocole DD (Directed Diffusion) [26] en ne considérant une région plutôt que d'envoyer les intérêts à l'ensemble du réseau. De cette manière, GEAR améliore le protocole DD et donc économise plus d'énergie.

Dans GEAR, chaque nœud maintient par le biais de ses voisins un coût estimé et un coût d'apprentissage de leur destination. Le coût estimé est une combinaison de l'énergie résiduelle et de la distance à la destination. Le coût d'apprentissage est une amélioration du coût estimé qui compte pour le routage autour des trous dans le réseau.

b) GPSR (Greedy Perimeter Stateless Routing)

Originellement conçu pour les réseaux ad hoc mobiles (Mobile Ad Hoc Networks ou MANETs), le protocole de routage GPSR [4] s'est très rapidement vu s'adapter aux réseaux de capteurs. GPSR dispose d'une technique de transmission gloutonne (Greedy Forwarding) pour l'acheminement des paquets vers la destination. Basé sur la métrique de proximité par calcul

de distance, ce mode glouton fait référence à la sélection du voisin le plus proche de la destination comme prochain saut pour le transfert d'un paquet. En fait, par une découverte de voisinage préalable grâce l'échange de messages beacons, chaque nœud dispose des coordonnées géographiques de ses voisins à un saut. Le paquet est ainsi transmis et relayé saut par saut en procédant à chaque étape à la sélection du voisin le plus proche de la destination. Occasionnellement, la technique "Greedy Forwarding" peut rencontrer quelques désagréments dans la mesure où le nœud courant peut se trouver plus proche de la destination finale que tous ses voisins, et que la destination finale reste inaccessible en un seul saut i.e. présence d'un trou de routage. GPSR recourt à la technique "Perimeter Forwarding" qui est basée sur la règle de la main droite. Ce mode est évoqué quand la technique "Greedy Forwarding" est prise à l'échec, de ce fait la technique "Perimeter Forwarding" consiste à router le paquet autour du trou dans le sens contraire des aiguilles d'une montre jusqu'à atteindre un nœud capable d'effectuer la progression gloutonne.

2.3.3 Protocoles hiérarchiques

L'objectif principal de routage hiérarchique est de maintenir efficacement la consommation d'énergie des nœuds capteurs en les impliquant dans la communication multi-sauts dans un groupe de nœuds particuliers, appelé cluster et en effectuant l'agrégation et la fusion de données afin de diminuer le nombre de messages redondants transmis à la station de base (voir Figure 2.3). La formation des clusters est généralement basée sur la réserve d'énergie de capteurs et de la proximité du capteur au chef du groupe, appelé clusterhead (CH) [27,28].

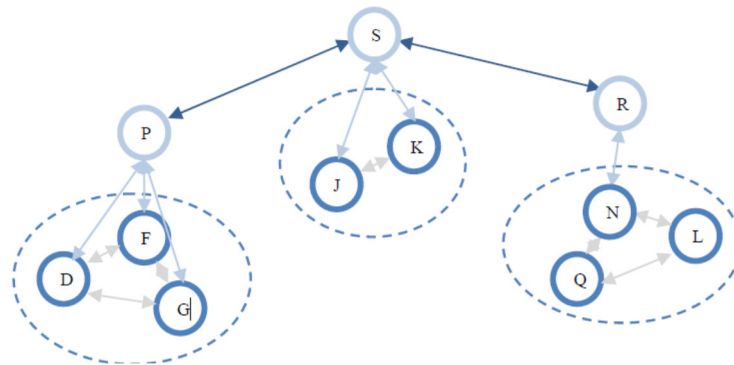


Figure 2.3 : Routage hiérarchique

Dans ce qui suit on va présenter quelques protocoles de routage hiérarchiques dans les RSCF :

a) **LEACH (Low-Energy Adaptive Clustering Hierarchy)**

LEACH [29] est l'une des premières approches de routage hiérarchiques pour les réseaux de capteurs sans fil. Son principe est de former des groupes de nœuds capteurs en se basant sur la force du signal reçu et utiliser les clusterheads locaux comme routeurs vers la station de base. Cela permettra d'économiser de l'énergie puisque les transmissions ne seront faites que par des CHs plutôt que tous les nœuds de réseaux. Tous les traitements de données telles que la fusion et l'agrégation de données sont faits par les CHs. Les nœuds prennent le rôle de CH aléatoirement et d'une manière périodique afin d'équilibrer la dissipation d'énergie des nœuds.

b) **Energy-aware routing for cluster-based sensor networks**

Younis et al. [30] ont proposé un algorithme de routage hiérarchique basé sur une architecture à trois niveaux. Les capteurs sont regroupés en groupes (clusters) avant l'opération du routage. L'algorithme utilise des clusterheads, nommés aussi passerelles, qui sont moins contraintes en énergie que les autres capteurs de réseaux et supposés qu'ils connaissent l'emplacement des autres nœuds. Les passerelles maintiennent les états des capteurs et mettent en place des itinéraires multi-sauts pour la collecte des données. La technique TDMA est utilisée par les nœuds pour envoyer les données à la passerelle. Celle-ci informe chaque nœud sur les slots dans lesquels il devrait écouter la transmission aux autres nœuds et les slots que le nœud peut utiliser pour sa propre transmission. La station de base communique seulement avec les passerelles.

c) **TEEN (Threshold sensitive Energy Efficient sensor Network protocol)**

TEEN [31] est un protocole hiérarchique conçu pour répondre à des changements soudains dans les attributs détectés telle que la température. La réactivité est importante pour les applications où le temps est un facteur critique, dans lesquelles le réseau fonctionne en mode réactif. Ce protocole utilise un mécanisme centré sur les données. L'architecture de réseau de capteurs est basée sur un regroupement hiérarchique où les nœuds plus étroits forment des clusters et ce processus est récursif pour le deuxième niveau jusqu'à ce que la station de base soit atteinte.

2.4 Techniques de routage dans les RCSF

Il existe deux techniques de routage dans les RCSF :

2.4.1 Routage mono-chemin

La plupart des protocoles de routage mono-chemin ont été conçus sans tenir compte des effets de différents débits de données injectés dans le réseau. Dans cette technique de routage, chaque nœud source sélectionne un seul chemin, qui peut satisfaire les exigences de performance de l'application prévue pour la transmission de son trafic vers la station de base. Par conséquent, en raison des ressources limitées des nœuds capteurs et le manque de fiabilité des liaisons sans fil, les approches de routage mono-chemin ne peuvent pas être considérées comme des techniques efficaces pour répondre aux exigences de diverses applications.

2.4.2 Routage multi-chemins

Afin de faire face aux limites de routage mono-chemin, une autre stratégie de routage, appelée routage multi-chemin est utilisée dans les réseaux de capteurs sans fil. Dans le cadre de ce PFE, on met l'accent sur le routage multi-chemins.

a) Définition de routage multi-chemins

Le routage multi-chemin permet d'envoyer les données sur un ensemble de chemins menant d'une source à une destination. Ce schéma de routage semble être une solution efficace dans les réseaux de capteurs sans fil en permettant de se prémunir contre le problème de rupture de routes et de distribuer le trafic à écouler entre une source et une destination sur plusieurs chemins. Ceci permet d'améliorer les performances des communications et d'agréger les ressources disponibles dans le réseau.

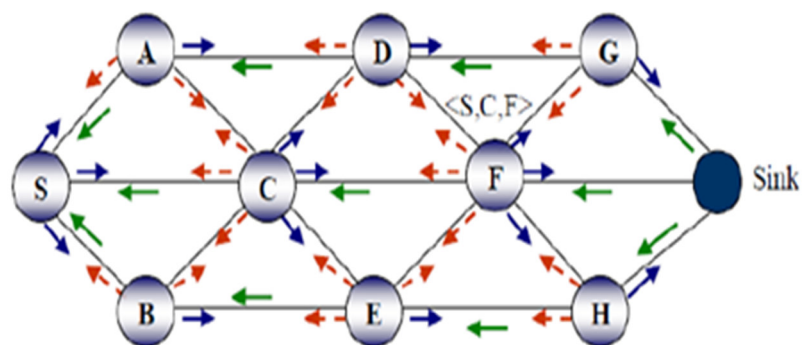


Figure 2.4 : Routage multi-chemins [32]

b) Phases de routage multi-chemins

Le routage multi-chemins est décomposé en trois phases: la découverte de chemins, répartition du trafic, et la maintenance de chemins.

- **Découverte de chemins:** cette phase sert à déterminer un ensemble de nœuds intermédiaires qui doivent être sélectionnés pour construire plusieurs chemins entre un nœud source et un nœud station de base. La construction de multi-chemins est basée sur le degré de disjonction, et selon ce degré deux types de chemins peuvent être distingués : chemins non-disjoints (chemins qui peuvent avoir des nœuds et des liens en commun) et chemins disjoints.
- **Répartition de trafic:** il existe diverses stratégies d'allocation du trafic sur les chemins disponibles. Un algorithme de sélection de chemin est utilisé pour sélectionner un sous-ensemble de chemins disponibles en fonction de certaines caractéristiques.
- **Maintenance de chemins:** le but de la maintenance de chemins est de valider les chemins existants et de trouver des chemins remplaçants appropriés lorsque l'un des chemins existants devient défaillant.

c) Avantages

Il existe plusieurs avantages du routage multi-chemin :

- L'amélioration de la performance ou la tolérance aux pannes (liens et nœuds),
- Agrégat des ressources disponibles,
- Améliorer la sécurité du réseau,
- L'équilibrage de charge (amélioration de la durée de vie du réseau),
- Fiabilité,
- Alléger la congestion du réseau,
- Réduire la fréquence des découvertes de route.

d) Inconvénients

Le routage multi-chemins améliore la livraison des paquets par l'envoi des copies multiples d'un même paquet sur des chemins différents. Cependant, cette approche n'est pas très efficace en termes de bande passante et de consommations d'énergie en raison de la grande quantité de messages inutiles transmis, de même la possibilité d'interférences entre chemins.

2.5 Les principaux protocoles multi-chemins

Dans la suite on présente quelques protocoles de routage multi-chemins dédiés aux réseaux de capteurs :

2.5.1 TPGF (Two Phase geographical Greedy Forwarding)

TPGF est un protocole de routage multi chemin reposant sur les informations géographiques afin de faciliter l'établissement de chemins disjoints à la demande [5]. Cet algorithme détermine le plus court chemin vers la destination avec évitement des vides par marquage. Par une exécution répétée ce protocole est capable de définir, à la demande, les meilleurs chemins disjoints vers la station de base. TPGF opère en deux phases : une première pour l'exploration des chemins potentiels vers le puits, et la seconde pour l'optimisation de ces chemins.

2.5.2 MPMPS (Multi-Priority Multi-Path Selection)

MPMPS [33] définit des niveaux de priorité pour chacun des flux audio et vidéo en fonction de l'application considérée : par exemple une plus haute priorité du flux vidéo dans un contexte de détection d'intrusion, par contre le flux audio le remporte dans le cadre d'une surveillance sous-marine. Les chemins sont ensuite établis en fonction de la priorité de chaque flux. Ces approches orientées QoS assurent un approvisionnement suffisant en bande passante des applications multimédia.

2.5.3 DGR (Directional Geographical Routing)

DGR sert à aborder le problème du streaming vidéo H.26L en temps réel sur un WMSN contraint en ressources (bande passante, énergie, . . .) consistant en quelques capteurs vidéo et plusieurs capteurs scalaires [34]. Par la combinaison d'un schéma de routage multi chemin avec la technique de codage d'erreur FEC (Forward Error Correction), le protocole DGR construit en fonction de l'application un nombre proportionnel de chemins disjoints pour un nœud vidéo afin de transmettre des flux vidéo protégés en parallèle dans un environnement peu fiable restreint en bande passante. Afin de répondre aux caractéristiques de la transmission vidéo, les chemins multiples dans DGR contribuent à l'équilibrage de la charge, l'agrégation de bande passante et la réduction de délais de transmission de paquets.

2.6 Conclusion

Ce chapitre est consacré aux protocoles de routage dans les RCSF, ces derniers sont classés en trois catégories : les protocoles plats, les protocoles hiérarchiques et les protocoles basés sur la localisation. Ainsi ces protocoles utilisent soit un routage mono-chemin ou bien un routage multi-chemins.

Dans le chapitre qui suit, nous améliorons GPSR de telle sorte qu'il soit un protocole multi-chemins pour bénéficier des avantages des schémas de routage multi-chemins.

Chapitre 3

Implémentation : Une version améliorée de GPSR

Chapitre 3

Implémentation : Une version améliorée de GPSR

3.1 Introduction

Ce chapitre présente la description de notre solution proposée dans le cadre de ce projet de fin d'études. Nous commençons la première partie par préciser l'environnement et les outils utilisés dans le développement d'une version améliorée de GPSR. Cette nouvelle version utilise un schéma de routage multi-chemins et oriente le processus de découverte de chemins vers la station de base en utilisant une vue angulaire. Ensuite, nous présentons le fonctionnement de l'application développée.

3.2 Outils de réalisation

Dans cette partie, nous présentons les outils logiciels nécessaires pour le développement de notre application. Pour cela, nous avons utilisé Ubuntu 14.04 LTS, le simulateur Omnetpp-4.6 et le simulateur Castalia-3.2.

3.2.1 OMNeT++ (Objective Modular Network Testbed in C++)

a) Définition

OMNeT++ [6] est un environnement de simulation open source possédant une interface graphique ergonomique intégrant un noyau de simulation. C'est un outil de simulation à événements discrets, orienté objets, basé sur C++, gratuit pour les utilisations académiques. Il a été conçu pour simuler :

- Les systèmes réseaux de communication.
- Les systèmes multiprocesseurs.
- Les systèmes distribués.
- Actuellement, il est utilisé dans les universités pour la validation de nouveaux matériels et logiciels, ainsi que pour l'analyse et l'évaluation de protocoles de communication.

- Les réseaux de file d'attente.

OMNET++ est connu par la facilité d'apprentissage, d'intégration de nouveaux modules et la modification de ceux déjà implémentés.

b) Architecture

L'architecture du modèle OMNeT++ se compose de plusieurs modules hiérarchiquement (visualisé dans la figure 3.1 ci-dessous) imbriqués qui sont :

- Module système.
- Modules simples (les feuilles) : programmés en C++ encapsulant le comportement réel d'un système. Pour chaque module simple correspond un fichier .cc et un fichier .h.
- Modules composés : constitués d'un ou de plusieurs modules simples ou des modules composés reliés entre eux. Les paramètres, les ports et les sous modules de chaque module sont spécifiés dans un fichier ".ned".

La communication entre les différents modules se fait à travers les échanges de messages. Les messages peuvent représenter des paquets, des trames d'un réseau informatique...etc. Les différents paramètres de chaque module sont spécifiés dans un fichier ".ini". OMNET++ génère à la fin de chaque simulation deux nouveaux fichiers "omnet.vec" et "omnet.sca" qui permettent de tracer les courbes et calculer des statistiques.

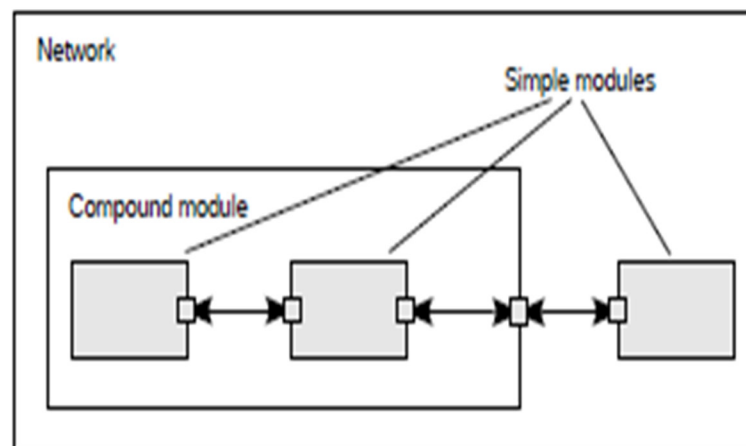


Figure 3.1 : Architecture OMNeT++ [6]

c) Composants

OMNET++ comporte plusieurs composants qui sont :

- Librairie de simulation (noyau ou kernel),
- Compilateur pour le langage de description de topologie (NED pour Network Descriptor),
- Editeur graphique de réseau (GNED pour Graphical Network Descriptor),
- Interface Graphique (GUI pour Graphical User Interface) pour l'exécution de simulation (appelée Tkenv),
- Interface ligne de commande pour l'exécution de simulation (appelée Cmdenv),
- Outil de traçage de vecteur graphique (Plove),
- Divers outils (outil de génération de nombres aléatoires, outil de création de fichier makefile, etc...).

d) Installation

Pour installer OMNET++ sous Ubuntu, nous procédons comme suit:

- Installez tous les paquets nécessaires pour compiler et exécuter OMNet++ : Vous pouvez exécuter la commande suivante pour installer tous les paquets pour de multiples plateformes:

```
sudo apt-get install build-essential gcc g++ bison flex perl \ tcl-dev tk-dev libxml2-dev zlib1g-dev default-jre \ doxygen graphviz libwebkitgtk-1.0-0
```
- Télécharger omnetpp-4.6-src.tgz à partir de : <http://omnetpp.org>,
- Décompresser le fichier : `$ tar xvfz omnetpp-4.6-src.tgz`,
- Entrer dans le dossier omnetpp-4.6 : `$ cd omnetpp-4.6`,
- Définir les variables d'environnement de façon non permanente à travers la commande suivante : `$. Setenv` ou de façon permanente à travers la modification de fichier `.bashrc` (`$ gedit ~/.bashrc`) dans le répertoire home en ajoutant la ligne suivante : `export PATH=$PATH:$HOME/omnetpp-4.6/bin` puis enregistrer-le,
- Dans le répertoire omnetpp-4.6 taper la commande suivante : `$./configure`,
- Lorsque `./configure` se termine, vous pouvez compiler OMNeT ++ à travers la commande : `$ make`,

- Vous pouvez lancer l'IDE OMNeT ++ Simulation en tapant la commande suivante dans le terminal: omnetpp.

3.2.2 Castalia

a) Définition

Castalia [8] qui est un simulateur de réseaux de capteurs (WSN), de réseaux corporels (BAN : Body Area Network) et plus généralement des réseaux de matériels embarqués de faible puissance. Il est basé sur la plateforme OMNET++ et qui offre un modèle radio intéressant basé sur les travaux de Zuniga et al. [35]. Cela permet aux utilisateurs de se rapprocher le plus possible de la réalité avec les aléas d'un médium de communication, avec un comportement de nœud réaliste en particulier en matière d'accès à la radio.

Castalia peut être utilisé pour évaluer des applications spécifiques pour des différentes plateformes, car il est très paramétrique, et peut simuler une grande variété de plateformes. Castalia gère également la consommation d'énergie des capteurs et de processeur, l'utilisation de la mémoire et du temps CPU.

b) Modèle de Castalia

Castalia est une mise en œuvre d'un modèle composé appelé Sensor Network qui est lui-même composé des modules simples : Physical Process, Wireless Channel et Node (voir figure 3.2).

- Physical Process : représentant les processus physiques et/ou les objets cibles à contrôler.
- Wireless Channel : représentant le canal de transmission sans fil.
- Node : représentant les nœuds capteurs et la station de base.

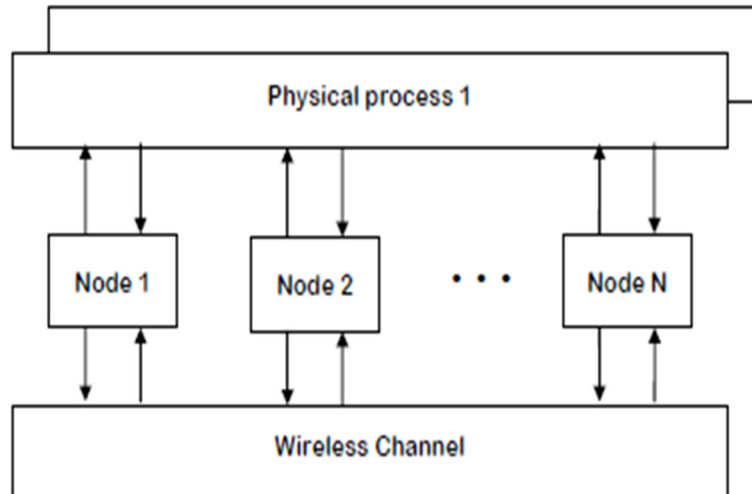


Figure 3.2 : Modèle Castalia [8]

c) Structure d'un nœud dans Castalia

Castalia définit le nœud de capteur comme décrit par la figure ci-dessous. Le nœud y est défini comme une série de blocs interconnectés définissant tous les processus qu'un nœud de capteur utilise. En effet, on retrouve le modèle OSI (couches radio, MAC, routage et application) ainsi que d'autres blocs liés à la gestion des ressources, la mobilité et la gestion du capteur.

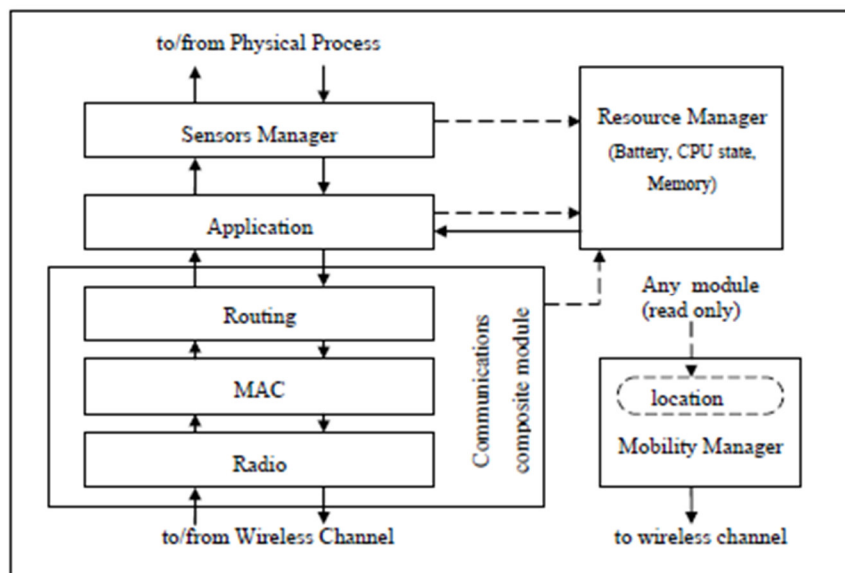
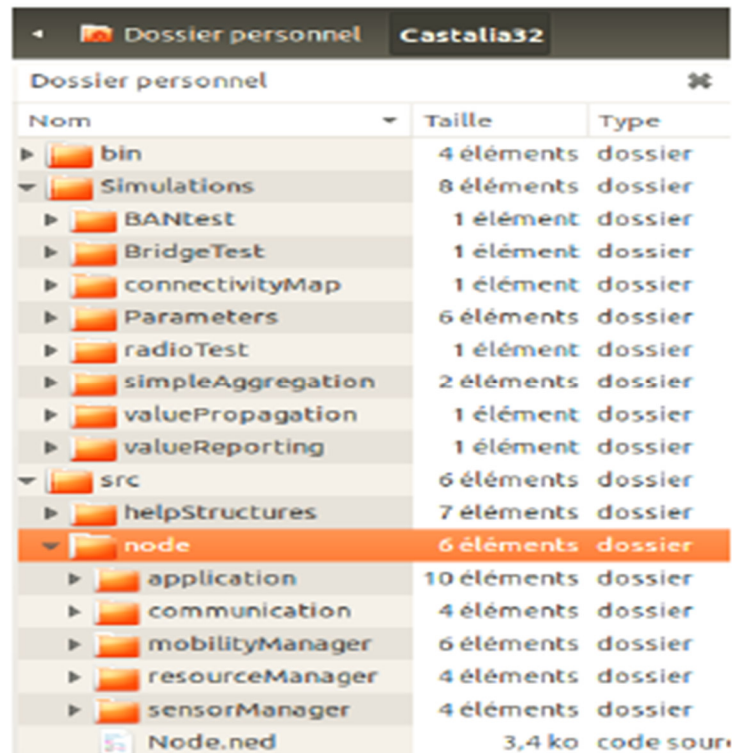


Figure 3.3 : Composition d'un nœud dans Castalia [8]

- **Module radio:** supporte plusieurs états : en transmission, en réception/attente, en veille. Il est possible de faire varier les puissances et les délais de transmission pour les différents états.
- **Module MAC (Medium Access Control) :** est une partie importante dans le comportement du nœud. La première motivation dans le développement de Castalia a été de tester le réglage du protocole MAC dans des conditions réalistes du canal radio.
- **Module réseau:** à partir de la version 1.2, le simulateur introduit le module réseau (routage). Au début, ils n'ont pas donné une importance à cet élément, donc il n'y avait pas de module pour le routage. L'utilisateur doit alors traiter le routage des paquets dans le module application s'il en a besoin.
- **Module du gestionnaire du dispositif de perception :** module intermédiaire entre le module application et le module processus physique (mesure captée). L'application envoie les requêtes de lecture des valeurs ambiantes vers ce module, et ce dernier accède au processus physique pour lire la valeur. Ce module permet de simuler la réception de données (événements).
- **Module application :** dans lequel, l'utilisateur effectue normalement beaucoup de changements pour implémenter de nouveaux algorithmes.
- **Module mobilité :** sert à spécifier comment les nœuds se déplacent dans l'espace. Les autres modules peuvent y accéder à n'importe quel moment. Des notifications périodiques sont envoyées au canal sans fil pour donner la position des nœuds.

Castalia utilise les caractéristiques d'OMNET++ pour définir l'architecture d'un nœud capteur. Toutes les définitions sont décrites et implémentées dans le répertoire *Castalia-3.2/src*. Pour mieux visualiser l'architecture de Castalia, on se réfère à la capture d'écran suivante montrant la disposition des différents éléments qui le composent.



Nom	Taille	Type
bin	4 éléments	dossier
Simulations	8 éléments	dossier
BANtest	1 élément	dossier
BridgeTest	1 élément	dossier
connectivityMap	1 élément	dossier
Parameters	6 éléments	dossier
radioTest	1 élément	dossier
simpleAggregation	2 éléments	dossier
valuePropagation	1 élément	dossier
valueReporting	1 élément	dossier
src	6 éléments	dossier
helpStructures	7 éléments	dossier
node	6 éléments	dossier
application	10 éléments	dossier
communication	4 éléments	dossier
mobilityManager	6 éléments	dossier
resourceManager	4 éléments	dossier
sensorManager	4 éléments	dossier
Node.ned	3,4 ko	code sourc

Figure 3.4 : Hiérarchie des dossiers dans Castalia

La version Castalia 3.2 est basée sur le simulateur de réseaux OMNeT++ [6], version 4.0, 4.1 et 4.6. Le système d'exploitation Linux est recommandé pour l'utilisation de Castalia. Après avoir téléchargé Castalia du site : <http://castalia.npc.nicta.com.au> faire entrer les commandes suivantes:

- Extraire le dossier Castalia : `$ tar -xvzf Castalia-3.2.tar.gz.`
- Compiler Castalia : `$ cd Castalia-3.2/`
`$./makemake`
`$make`

3.3 Description de protocole GPSR

Dans cette section, nous présentons avec plus de détails le protocole GPSR [4] pour montrer l'amélioration pourra porter sur quels éléments pour que GPSR soit très efficace.

GPSR est un protocole de routage réactif et efficace qui a été conçu et adapté pour les réseaux ad hoc mobiles et les réseaux de capteurs. Son modèle de fonctionnement suppose que tous les nœuds se trouvent au niveau d'un même plan. Il utilise la position géographique des nœuds pour l'acheminement des paquets de données ou de contrôle. Alternativement, le protocole GPSR permet au nœud d'encapsuler sur quelques bits leur position dans les paquets

de données qu'il envoie. Dans ce cas, toutes les interfaces des nœuds doivent être en mode promiscuité afin de recevoir les paquets s'ils se trouvent dans la zone de couverture de l'émetteur.

L'acheminement des paquets dans le protocole GPSR se fait selon deux modes: transmission gloutonne "Greedy Forwarding" pour l'acheminement des paquets vers la destination et la transmission de périmètre "Perimeter Forwarding" basée sur la règle de la main droite.

3.3.1 Transmission gloutonne (Greedy Forwarding)

Basée sur la métrique de proximité par calcul de distance, ce mode glouton fait référence à la sélection du voisin le plus proche de la destination comme prochain saut pour le transfert d'un paquet. En fait, par une découverte de voisinage préalable grâce l'échange de messages beacons, chaque nœud dispose des coordonnées géographiques de ses voisins à un seul saut. La figure 3.5 illustre un scénario où une source S, positionnée au point de coordonnées (x_s, y_s) , désire transmettre un paquet de données au nœud D (x_d, y_d) . Le paquet est ainsi transmis et relayé saut par saut en procédant à chaque étape à la sélection du voisin le plus proche de la destination.

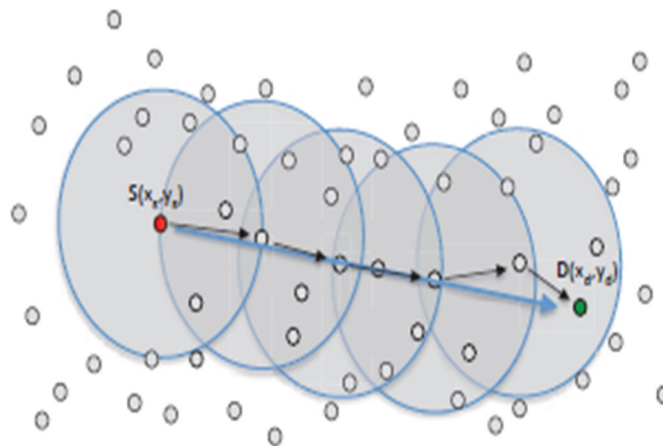


Figure 3.5 : Transmission gloutonne [4]

Ce mode d'acheminement offre un taux de réussite assez proche des réseaux filaires dans le cas où la mobilité de la destination n'est très forte. Lorsqu'un paquet de données atteint une région où la transmission gloutonne échoue (présence d'un trou comme montre la figure 3.6), alors la transmission de périmètre est utilisée.

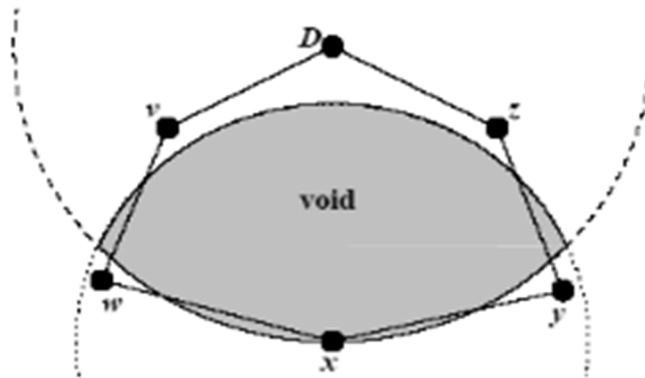


Figure 3.6 : Présence d'un trou [4]

3.3.2 Transmission de périmètre (Perimeter Forwarding)

Basée sur la règle de la main droite, ce mode est évoqué quand le mode à base de la transmission gloutonne échoue à router le paquet autour du trou dans le sens contraire des aiguilles d'une montre jusqu'à atteindre un nœud capable d'effectuer la progression gloutonne. la figure 3.7 montre un exemple plus précis de ce mode.

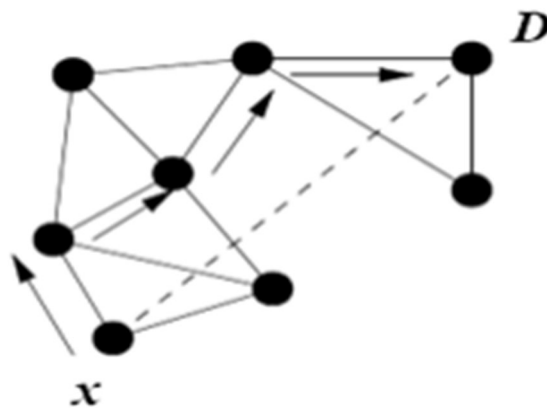


Figure 3.7 : Transmission de périmètre [4]

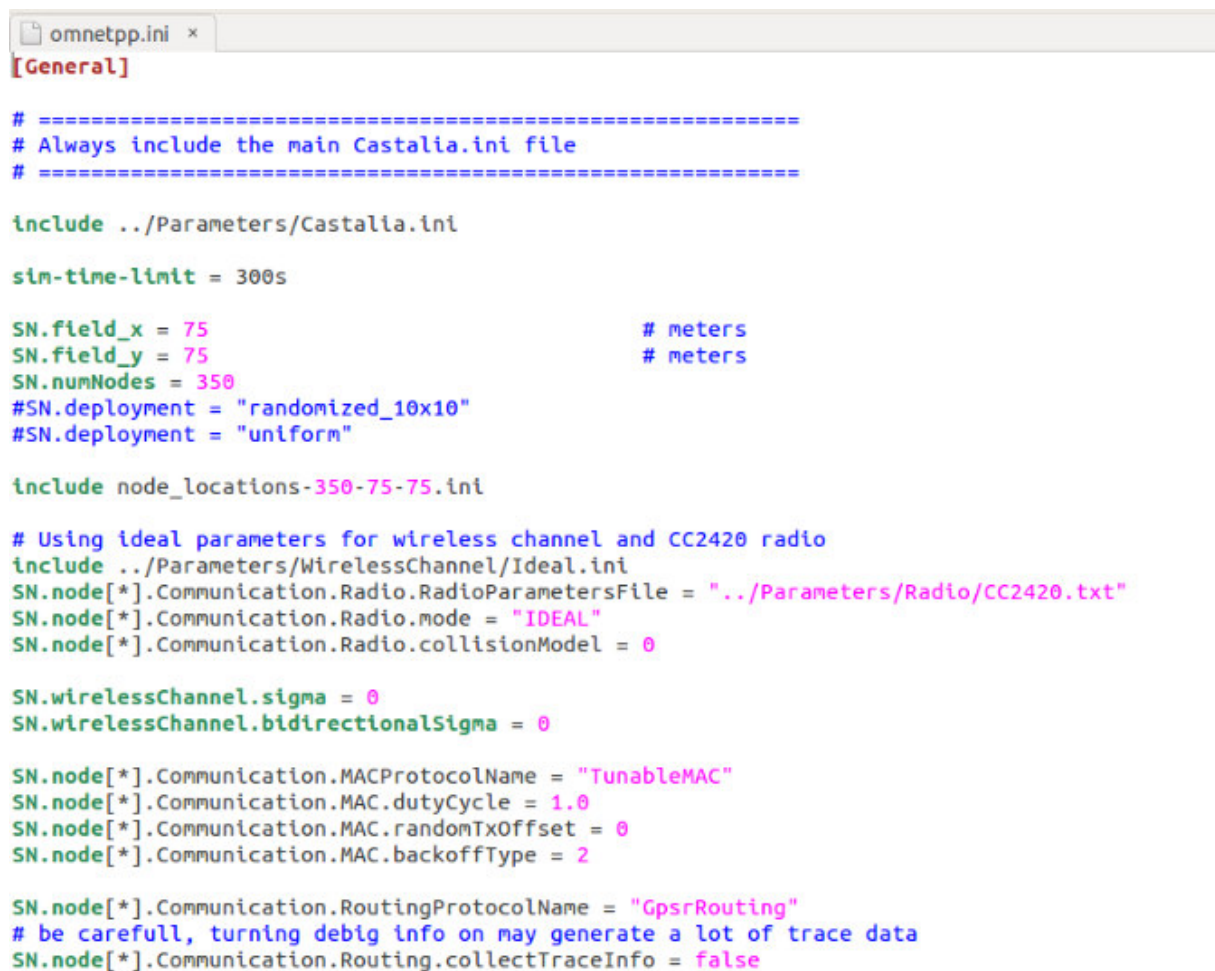
3.4 Implémentation

Notre objectif dans ce projet est de proposer une version améliorée de GPSR afin de palier de ses limites. Dans ce contexte, nous avons proposé un schéma de routage multi-chemins pour que GPSR puisse d'une part être un bon support pour les applications qui visent à envoyer

une grande quantité de données telles que les applications multimédias et d'autre part pour faire de GPSR un protocole tolérant aux pannes.

Pour atteindre cet objectif, nous avons tout d'abord évalué la version originale de GPSR pour illustrer ses limites en termes de latence et de fiabilité. En outre, nous avons testé GPSR pour des réseaux plus ou moins denses pour mettre en valeur ses performances quand on passe à l'échelle. Pour ce faire nous avons conduit plusieurs simulations en utilisant le simulateur Castalia et nous avons mis le point en particulier sur les métriques suivantes : le nombre de messages envoyés et reçus, la consommation d'énergie et le délai de transmission.

Pour cela on a utilisé le script *Castalia* qui génère un fichier qui contient les résultats, ensuite le script *CastaliaResults* pour afficher les diverses statistiques, et le script *CastaliaPlot* pour afficher ces statistiques en mode graphique. La figure 3.8 illustre l'environnement de simulation pour évaluer *GPSR* et *GPSR_Modifié*



```
omnetpp.ini x
[General]

# =====
# Always include the main Castalia.ini file
# =====

include ../Parameters/Castalia.ini

sim-time-limit = 300s

SN.field_x = 75           # meters
SN.field_y = 75           # meters
SN.numNodes = 350
#SN.deployment = "randomized_10x10"
#SN.deployment = "uniform"

include node_locations-350-75-75.ini

# Using ideal parameters for wireless channel and CC2420 radio
include ../Parameters/WirelessChannel/Ideal.ini
SN.node[*].Communication.Radio.RadioParametersFile = "../Parameters/Radio/CC2420.txt"
SN.node[*].Communication.Radio.mode = "IDEAL"
SN.node[*].Communication.Radio.collisionModel = 0

SN.wirelessChannel.sigma = 0
SN.wirelessChannel.bidirectionalSigma = 0

SN.node[*].Communication.MACProtocolName = "TunableMAC"
SN.node[*].Communication.MAC.dutyCycle = 1.0
SN.node[*].Communication.MAC.randomTxOffset = 0
SN.node[*].Communication.MAC.backoffType = 2

SN.node[*].Communication.RoutingProtocolName = "GpsrRouting"
# be carefull, turning debug info on may generate a lot of trace data
SN.node[*].Communication.Routing.collectTraceInfo = false
```



```
SN.node[*].ApplicationName = "ThroughputTest"
SN.node[*].Application.collectTraceInfo = true

SN.node[14].Application.packet_rate = 0.2
SN.node[14].Application.maxData = 100
SN.node[14].Application.startupDelay = 10
SN.node[14].Application.nextRecipient = "6"

SN.node[10].Application.packet_rate = 0.2
SN.node[10].Application.maxData = 100
SN.node[10].Application.startupDelay = 20
SN.node[10].Application.nextRecipient = "6"

SN.node[1].Application.packet_rate = 0.2
SN.node[1].Application.maxData = 100
SN.node[1].Application.startupDelay = 15
SN.node[1].Application.nextRecipient = "11"

#SN.wirelessChannel.printDebugInfo = true
```

Figure 3.8 : Fichier .ini

3.4.1 Implémentation de GPSR_monochemin et GPSR_multichemin

Le routage dans GPSR_monochemin se fait selon l'algorithme 1 alors que le routage dans GPSR_multichemin se fait selon l'algorithme 2.

Algorithme1: le routage mono-chemin

Inputs :

nexH_GF : l'@ du prochain saut le plus proche calculé depuis une fonction qui utilise les positionnements des nœuds et la distance entre eux.

nexH_PF : l'@ du prochain saut le plus proche calculé depuis une fonction qui utilise la main droite

Outputs :

GPSR_mono-chemin

- 1: **if** (nexH_GF!= -1) **then** ,
- 2: **GPSR_mono-chemin**=nexH_GF
- 3: acheminer les paquets vers un seul chemin le plus proche **GPSR_mono-chemin** en mode greedy
- 4: **else if** (nexH_PF!= -1) **then**
- 5: **GPSR_mono-chemin**=nexH_PF
- 6: acheminer les paquets vers un seul chemin le plus proche **GPSR_mono-chemin** en mode
- 7: perimeter
- 8: **else** existence d'un trou
- 9: **end**

Algorithme2: le routage multi-chemin

Inputs :

long : la longueur du paquet

Tmax : la taille maximum d'un paquet

nbr_frg : le nombre de fragments obtenu depuis la division de **long** sur **Tmax**

nbr_bit : le reste de la division de **long** sur **Tmax**

tab: la table des indices de voisinages triée de la plus petite distance entre le nœud courant et la destination a la plus grande en fonction de positionnement des nœuds.

T_tab : la taille du **tab**

Outputs :

GPSR_mono-chemin: l'@ du prochain saut le plus proche

GPSR_multi-chemin : les @ des prochains sauts les plus proches

```
1  i=0, j=0, k=0
2:  if (long<Tmax) then
3:    GPSR_mono-chemin=tab[0] puisque la table des voisinages est trié de plus petit au plus grand
4:    acheminer les paquets ver un seul chemin le plus proche GPSR_mono-chemin
6:  else if (nbr_bit<T_tab)
8:      while ( (i<= nbr_frg) && (j<=nbr_frg) ) do
9:          if (i!=nbr_frg -1)
10:             affecter au paquet fragmenté une taille égale a Tmax bites
11:             else affecter au paquet fragmenté une taille égale à nbr_bit
12:             envoyer le paquet au nœud qui a l@ égale GPSR_multi-chemin [j]
14:             i=i+1
15:             j=j+1
16:         end while
17:         else
18:             while (i<=nbr_frg) do
19:                 while (j<=T_tab)
20:                     affecter au paquet une taille égale a Tmax bites
21:                     envoyer le paquet au nœud qui a l@ égale GPSR_multi-chemin [j]
22:                     i=i+1
23:                     j=j+1
24:                 end while
25:                 k=0
26:                 if (i!=nbr_frg -1)
27:                     affecter au paquet fragmenté une taille égale a Tmax bites
29:                 else affecter au paquet fragmenté une taille égale a nbr_bit
30:                 envoyer le paquet au nœud qui a l@ égale GPSR_multi-chemin [k]
31:                 i=i+1
32:                 k=k+1
33:             end while
34:  end
```

Algorithme3: localisation des nœuds via la vue angulaire

Inputs :

self_p : la position du nœud courant

Dest_p: la position destinataire

Dest_@ : l'@ destinataire

Source_p: la position source

Intersection: une fonction qui vérifie l'existence d'intersection entre **self_p**, **Dest_p**, **Source_p** et le **prochain saut**

Outputs:

GPSR_voiAngulaire : le calcul de l'angle minimal qui se trouve entre l'axe qui relie un nœud avec **Source_p** et le deuxième axe qui relie ce nœud avec **Dest_p** par rapport aux nœuds existant dans la planaire

GPSR_voiAngulaire@: l'@ du nœud retourner depuis la fonction **GPSR_voiAngulaire**

nexH : l'@ du prochain saut

```
1: HasIntersection =faux
2:   if HasIntersection retourne une valeur
3:     then
4:       HasIntersection =vrai
5:       nexH= Dest_@
6:     else
7:       nexH= GPSR_voiAngulaire@
8:     end
```

3.4.2 Evaluation des performances de GPSR et de GPSR_Modifié

Dans la version originale de GPSR, GPSR dispose d'un schéma de routage mono-chemin c'est-à-dire à partir d'un nœud source tous les paquets empruntent le même chemin c'est comme on dispose d'un mode connecté alors que dans la version modifiée de GPSR, un schéma de routage multi-chemins est adopté où les paquets n'empruntent pas forcément le même chemin.

a) Contexte d'exécution

Nous avons fait des simulations en fonction de la taille des réseaux : 25, 50, 100, 175 et 350 nœuds pour illustrer la robustesse de notre solution lors du passage à l'échelle. La figure 3.9 représente un exemple d'exécution d'un scénario.

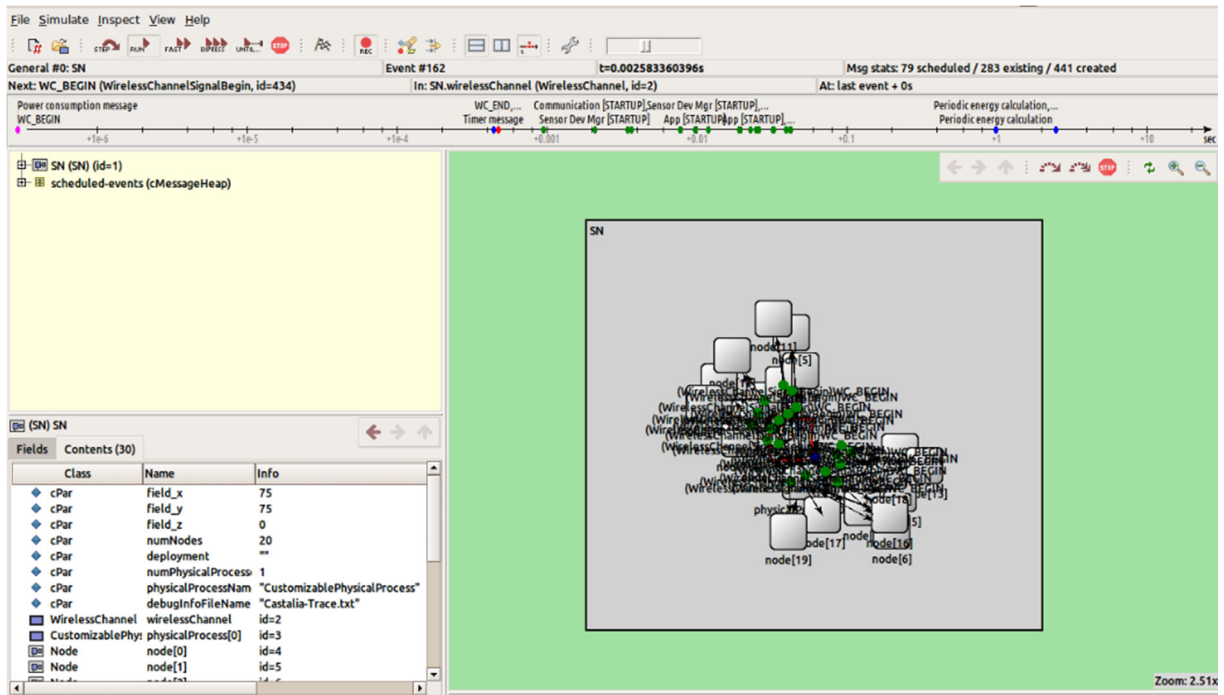


Figure 3.9 : Un exemple d'exécution avec déploiement des nœuds

b) Evaluation du nombre de paquets envoyés

Nous avons évalué le nombre de paquets transmis durant une période de 300 secondes pour différentes tailles de réseaux. Les figures 3.10 et 3.11 illustrent le nombre de paquets transmis dans un réseau dont la taille est 100 nœuds respectivement en mode mono-chemin et multi-chemin.

```

ahlem@ahlem-Aspire-E1-570:~/Bureau/a/Castalia-3.2-GPSR/Simulations/gpsrTest$ ../../bin/CastaliaResults -i 160615-164652.txt -s Tx pkts
Communication.Radio:TXed pkts - TX pkts
+-----+
|         |
+-----+
| 122.3   |
+-----+

ahlem@ahlem-Aspire-E1-570:~/Bureau/a/Castalia-3.2-GPSR/Simulations/gpsrTest$ ../../bin/CastaliaResults -i 160615-164652.txt -s Tx pkts --sum
Communication.Radio:TXed pkts - TX pkts
+-----+
|         |
+-----+
| 12230   |
+-----+

ahlem@ahlem-Aspire-E1-570:~/Bureau/a/Castalia-3.2-GPSR/Simulations/gpsrTest$
    
```

Figure 3.10 : Nombre de paquets transmis en mode mono-chemin

```
ahlem-Aspire-E1-570: ~/Bureau/a/GPSR-mintab/Simulations/gpsrTest
ahlem@ahlem-Aspire-E1-570:~/Bureau/a/GPSR-mintab/Simulations/gpsrTest$ ../../bin/CastaliaResults -i 160615-180835.txt -s Tx pkts

Communication.Radio:TXed pkts - TX pkts
+-----+
|         |
+-----+
| 121.7   |
+-----+

ahlem@ahlem-Aspire-E1-570:~/Bureau/a/GPSR-mintab/Simulations/gpsrTest$ ../../bin/CastaliaResults -i 160615-180835.txt -s Tx pkts --sum

Communication.Radio:TXed pkts - TX pkts
+-----+
|         |
+-----+
| 12170   |
+-----+

ahlem@ahlem-Aspire-E1-570:~/Bureau/a/GPSR-mintab/Simulations/gpsrTest$
```

Figure 3.11 : Nombre de paquets transmis en mode multi-chemin

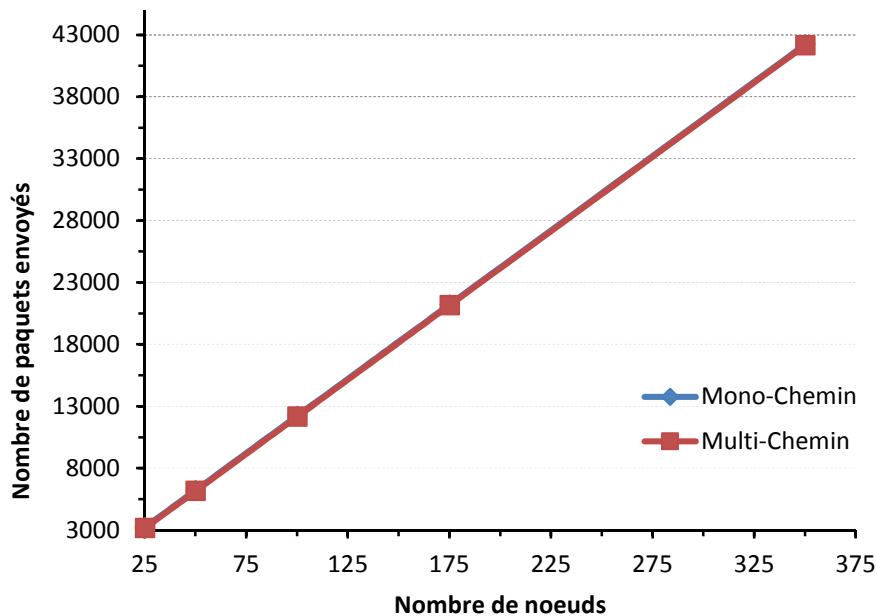


Figure 3.12 : Nombre de paquets transmis dans les deux modes

Nous remarquons que le nombre de messages transmis dans les deux modes est presque le même pendant une durée de simulation de 300 secondes.

c) Evaluation du nombre de paquets reçus

Les figures 3.13 et 3.14 illustrent le nombre de paquets transmis dans un réseau dont la taille est 100 nœuds respectivement en mode mono-chemin et multi-chemin.

```

ahlem-Aspire-E1-570: ~/Bureau/a/Castalia-3.2-GPSR/Simulations/gpsrTest
ahlem@ahlem-Aspire-E1-570:~/Bureau/a/Castalia-3.2-GPSR/Simulations/gpsrTest$ ../../bin/CastaliaResults -i 160615-164652.txt -s Rx pkts

Communication.Radio:RX pkt breakdown
+-----+-----+-----+
| Failed, below sensitivity | Failed, non RX state | Received with NO interference |
+-----+-----+-----+
| 3635.32                    | 39.87                 | 8205.96                       |
+-----+-----+-----+

ahlem@ahlem-Aspire-E1-570:~/Bureau/a/Castalia-3.2-GPSR/Simulations/gpsrTest$ ../../bin/CastaliaResults -i 160615-164652.txt -s Rx pkts --sum

Communication.Radio:RX pkt breakdown
+-----+-----+-----+
| Failed, below sensitivity | Failed, non RX state | Received with NO interference |
+-----+-----+-----+
| 363532                    | 3987                  | 820596                         |
+-----+-----+-----+
    
```

Figure 3.13 : Nombre de paquets reçus et perdus en mode mono-chemin

```

ahlem-Aspire-E1-570: ~/Bureau/a/GPSR-mintab/Simulations/gpsrTest
ahlem@ahlem-Aspire-E1-570:~/Bureau/a/GPSR-mintab/Simulations/gpsrTest$ cd ../../
ahlem@ahlem-Aspire-E1-570:~/Bureau/a/GPSR-mintab$ cd Simulations/
ahlem@ahlem-Aspire-E1-570:~/Bureau/a/GPSR-mintab/Simulations$ cd gpsrTest/
ahlem@ahlem-Aspire-E1-570:~/Bureau/a/GPSR-mintab/Simulations/gpsrTest$ ../../bin/CastaliaResults -i 160615-180835.txt -s Rx pckts

Communication.Radio:RX pkt breakdown
+-----+-----+
| Failed, non RX state | Received with NO interference |
+-----+-----+
| 26.53                | 11795.2                       |
+-----+-----+

ahlem@ahlem-Aspire-E1-570:~/Bureau/a/GPSR-mintab/Simulations/gpsrTest$ ../../bin/CastaliaResults -i 160615-180835.txt -s Rx pckts --sum

Communication.Radio:RX pkt breakdown
+-----+-----+
| Failed, non RX state | Received with NO interference |
+-----+-----+
| 2653                 | 1179520                        |
+-----+-----+
    
```

Figure 3.14 : Nombre de paquets reçus et perdus en mode multi-chemin

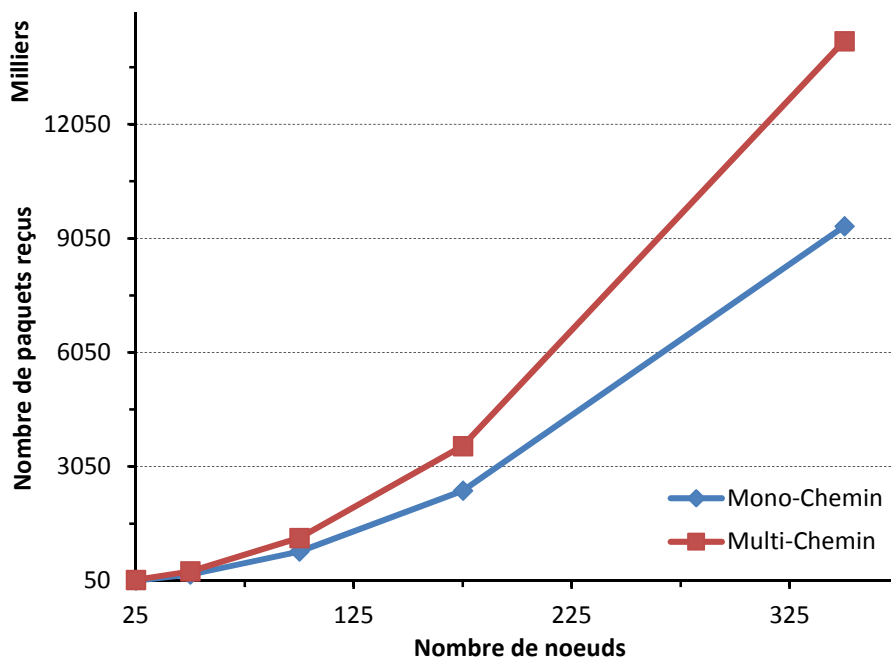


Figure 3.15 : Nombre de paquets reçus dans les deux modes

Notre solution augmente le nombre de paquets reçus au niveau de la station de base ceci est dû aux chemins alternatifs pris par les paquets transmis par les nœuds sources

d) Evaluation du taux d'erreurs

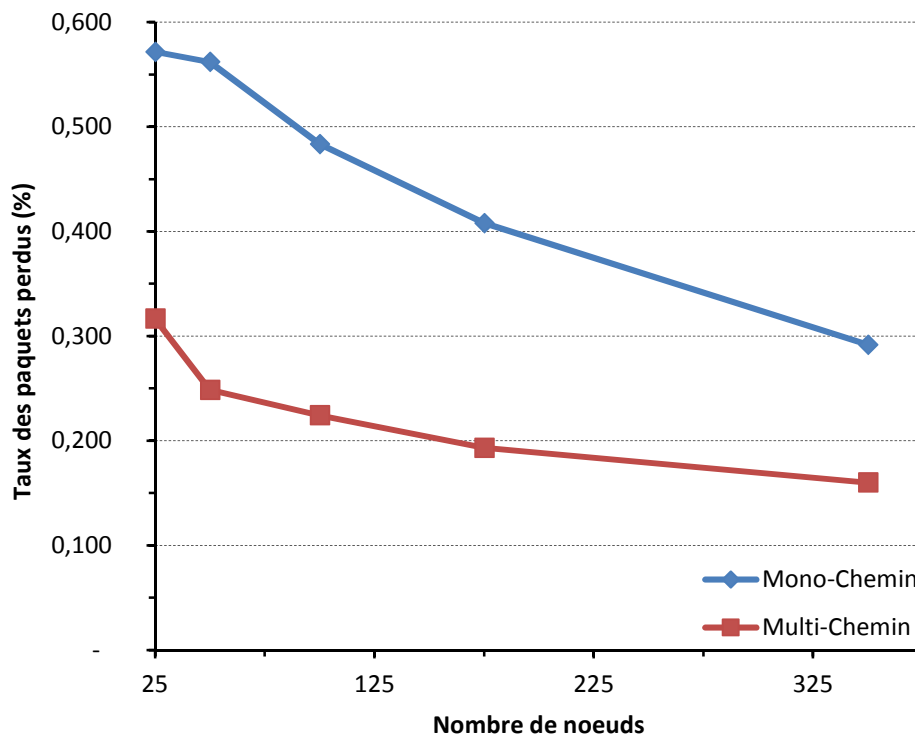


Figure 3.16 : Taux de paquets perdus dans les deux modes

Nous remarquons que notre solution réduit considérablement le taux de paquets perdus pourtant nous n'avons pas comptabilisé un type de paquets perdus appelés "Sensitivity failed packets" ceci est dû la présence de plusieurs chemins dans notre solution.

e) Evaluation de la consommation d'énergie

Les figures 3.17 et 3.18 illustrent la consommation d'énergie dans un réseau dont la taille est 350 nœuds respectivement en mode mono-chemin et multi-chemin.

```
ResourceManager:Consumed Energy
+-----+
|         |
+-----+
| 20.398 |
+-----+

ahlem@ahlem-Aspire-E1-570:~/Bureau/a/Castalia-3.2-GPSR/Simulations/gpsrTest$ ../../bin/CastaliaResults -i 160608-202458.txt -s energy --sum

ResourceManager:Consumed Energy
+-----+
|         |
+-----+
| 7139.308 |
+-----+

ahlem@ahlem-Aspire-E1-570:~/Bureau/a/Castalia-3.2-GPSR/Simulations/gpsrTest$ █
```

Figure 3.17 : Consommation d'énergie en mode mono-chemin

```
ResourceManager:Consumed Energy
+-----+
|         |
+-----+
| 8.504 |
+-----+

ahlem@ahlem-Aspire-E1-570:~/Bureau/a/GPSR-mintab/Simulations/gpsrTest$ ../../bin/CastaliaResults -i 160608-201713.txt -s energy --sum

ResourceManager:Consumed Energy
+-----+
|         |
+-----+
| 2976.493 |
+-----+

ahlem@ahlem-Aspire-E1-570:~/Bureau/a/GPSR-mintab/Simulations/gpsrTest$ █
```

Figure 3.17 : Consommation d'énergie en mode multi-chemin

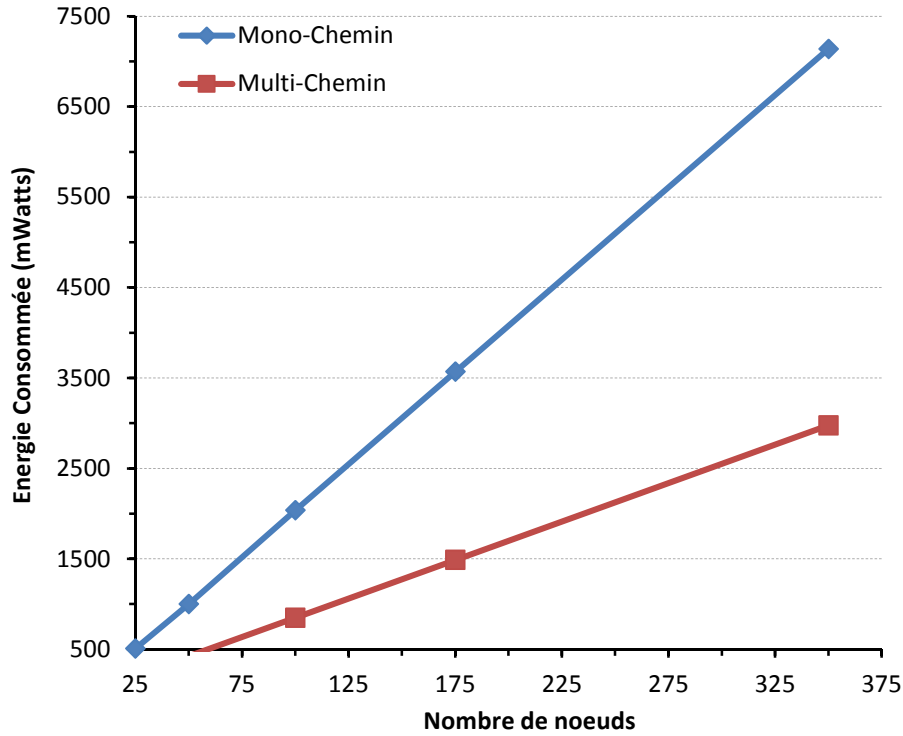


Figure 3.19 : Consommation d'énergie dans les deux modes

Nous constatons que notre solution est économe en termes d'énergie comparativement à la version originale de GPSR ceci est dû au nombre réduit de paquets perdus et de contrôle.

f) Evaluation de la latence

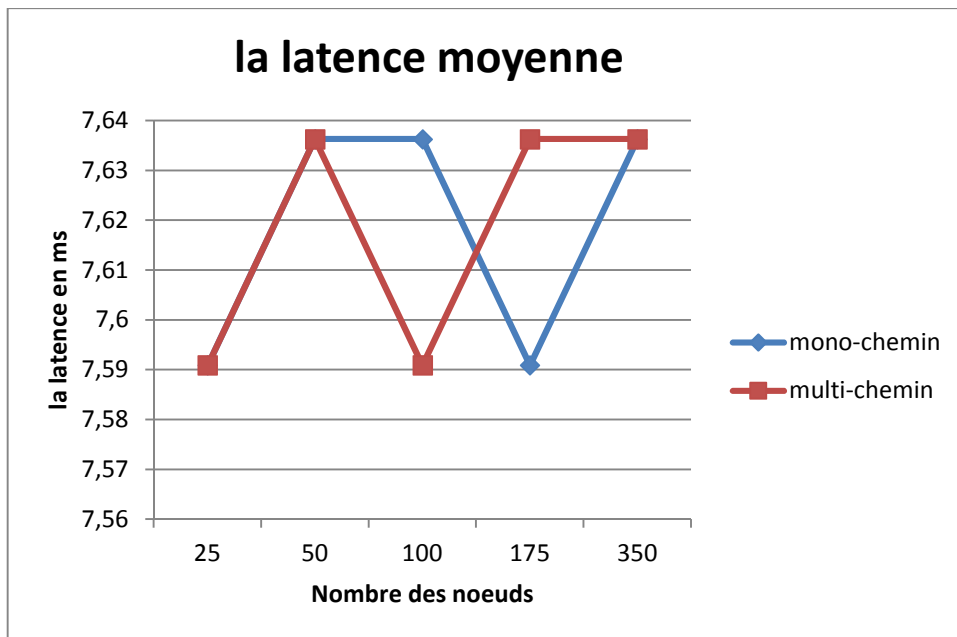


Figure 3.20 : Calcul de la latence dans les deux modes

Dans les deux modes nous n'avons pas constaté une amélioration dans la latence puisque les chemins une fois établis, les paquets sont acheminés pas à pas jusqu'à l'arrivée à la station de base.

Nous avons utilisé le mécanisme Castalia pour acheminer les paquets de contrôle de la couche d'application à la couche inférieure comme MAC, Radio en passant par la couche réseau "GPSR_Routing". De ce fait, nous avons amélioré le protocole GPSR de telle sorte que les paquets vont être fragmentés en plusieurs sous paquets puis acheminer via plusieurs chemins entre la source et la destination en utilisant la table des voisinages de chaque nœud. Toutes les démarches de ce mode de routage sont montrées au niveau de l'algorithme2 (routage multi-chemin). Par suite si cette méthode échoue, nous utiliserons une deuxième technique pour la localisation du prochain saut qui situe dans la vue angulaire. L'algorithme3 illustré au-dessous montre que Cette technique exploite un chemin en ligne de vision directe entre l'émetteur et le récepteur et 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. Encore si cette méthode ça n'a pas marché alors nous utilisons la technique basée sur "Perimeter Forwarding" sinon on constate qu'il existe un trou entre les nœuds.

3.5 Conclusion

Dans ce chapitre, nous avons évalué GPSR et nous avons proposé une version améliorée pour ce protocole. Cette version proposée implique d'une part plusieurs chemins lors d'un processus de routage et d'autre part elle utilise le mécanisme de la vue angulaire pour améliorer les performances au mieux.

Les résultats de la simulation ont montré que les performances de la version modifiée sont meilleures en termes de taux de paquets perdus, de l'énergie consommée et du nombre de paquets reçus.

Conclusion générale

Les réseaux de capteurs sans fil sont une nouvelle technologie qui a surgi après les grands progrès technologiques concernant le développement des capteurs intelligents, des processeurs puissants et des protocoles de communication sans fil capables de localiser des événements et acheminer les données vers un point de collecte appelé station de base. Ce type de réseaux se caractérise par l'aspect miniaturisé et les ressources limitées en termes calcul, stockage. Par ailleurs, la consommation d'énergie est un facteur critique car les nœuds sont généralement déployés dans des zones hostiles.

Dans ce projet nous avons étudié le protocole GPSR dans le but de faire une amélioration de ce dernier à l'aide des deux simulateurs OMNET++ et Castalia. Après avoir effectué plusieurs simulations sur la version originale et analysé les résultats obtenus, nous avons constaté que plus la surcharge des données par nœud augmente plus l'énergie augmente. En outre, nous avons constaté aussi que plus le nombre des nœuds dans le réseau est élevé plus la consommation d'énergie est élevée, par conséquent ce lourd surcoût de communication aura un impact négatif sur la durée de vie du réseau. Pour alléger cette limite, nous avons proposé une version améliorée qui permet d'impliquer plusieurs chemins pour router les données des nœuds sources à la station de base et utiliser une voie angulaire dans la direction de la station de base.

Les résultats de simulations étaient vraiment très prometteurs ce qui nous a permis de penser à adapter GPSR aux applications multimédias. Ces applications qui ont plus d'exigences en termes de qualité de service.

Bibliographie

- [1] Y. Zhang and H. Xiao, “Bluetooth-based sensor networks for remotely monitoring the physiological signals of a patient”, IEEE Transactions on Information Technology in Biomedicine, vol.13, no.6, pp.1040–1048, November 2009.
- [2] D.-M. Han and J.-H. Lim, “Smart home energy management system using IEEE 802.15.4 and ZigBee”, IEEE Transactions on Consumer Electronics, vol.56, no.3, pp.1403–1410, August 2010.
- [3] T. R. Sheltami, E. M. Shakshuki and I. K. Maarouf, “Performance evaluation for Geographical and Energy Aware Routing protocol”, In Proceedings of the 5th IEEE International Conference on *GCC Conference & Exhibition*, pp.1-5, 2009, Kuwait City.
- [4] B. Karp and H. T. Kung, “GPSR: greedy perimeter stateless routing for wireless networks”, In Proceedings of the 6th ACM annual International Conference on Mobile computing and networking (MobiCom '00), pp.243-254, New York, NY, USA, 2000.
- [5] L. Shu, Y. Zhang, L. T. Yang, M. Hauswirth, and N. Xiong, “TPGF : Geographic Routing in Wireless Multimedia Sensor Networks”, Telecommunication Systems, vol. 44, no. 1-2, pp.79–95, 2010.
- [6] OMNET++ Home Page. <http://www.omnetpp.org> Modified on May, 2016.
- [7] A. Varga, “The OMNET++ Discrete Event Simulation System”, In the Proceedings of the European Simulation Multiconference (ESM'2011), June 2001, Prague, Czech Republic.
- [8] Castalia “Wireless Sensor Simulator”, <https://castalia.forge.nicta.com.au/index.php/en/>, Modified on June, 2014.
- [9] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. “Wireless sensor networks: a survey”, Computer Networks (Elsevier), vol.38, no.4, pp.393-422, March 2002.
- [10] Telosb Sensors: <https://telosbsensors.wordpress.com/>, Modified on January 2014.
- [11] K. Lin, J. Yu, J. Hsu, S. Zahedi, D. Lee, J. Friedman, A. Kansal, V. Raghunathan, and M. Srivastava, “Helimote : Enabling long-lived sensor networks through solar energy harvesting,” in Proceedings of the 3rd ACM International Conference on Embedded Networked Sensor Systems (SenSys'05), pp.309–309, 2005.

- [12] M. LEHSAINI, "Diffusion et Couverture basées sur le clustering dans les réseaux de capteurs sans fil", Thèse de doctorat, Université de Tlemcen, Université de Franche-Comté, 2009.
- [13] Y. Younes, "Minimisation d'énergie dans un réseau de capteurs", Mémoire de Magister en Informatique, Université Mouloud Mammeri Tizi-ouzo, Algérie, Septembre 2012.
- [14] K. Beydoun, "Conception d'un protocole de routage hiérarchique pour les réseaux de capteurs", Thèse de Doctorat en Informatique, Université de Franche-Comté, France, Décembre 2009.
- [15] M. Hadjila, "Protocoles de routage économes en énergie pour les réseaux de capteurs sans fil", Thèse de Doctorat en Informatique, Université de Tlemcen, Algérie, Décembre 2009.
- [16] I. F. Akyildiz, Erich P. Stuntebeck, "Wireless underground sensor networks: Research challenges", *Ad Hoc Networks*, vol.4, no. 6, pp.669-686, November 2006.
- [17] M. Li and Y. Liu, "Underground Structure Monitoring with Wireless Sensor Networks", In the Proceedings of the 6th IEEE International Symposium on Information Processing in Sensor Networks, pp. 69-78, Cambridge, MA, 2007.
- [18] J. Heidemann, W. Ye, J. Wills, A. Syed and Y. Li, "Research challenges and applications for underwater sensor networking", In the Proceedings of the IEEE International Conference on Wireless Communications and Networking (WCNC'2006), pp. 228-235, Las Vegas, NV, 2006.
- [19] I. F. Akyildiz, T. Melodia, and K. R. Chowdhury, "A survey on wireless multimedia sensor networks", *Computer Networks*, vol. 51, no. 4, pp. 921-960, March 2007.
- [20] Y. CHALLAL, "Réseaux de capteurs sans fil", Support de cours, Novembre 2008.
- [21] I. F. Akyildiz, Weilian Su, Y. Sankarasubramaniam and E. Cayirci, "A survey on sensor networks", in *IEEE Communications Magazine*, vol. 40, no. 8, pp. 102-114, Aug 2002.
- [22] H. Alemdar and C. Ersoy, "Wireless Sensor Networks for Healthcare: A Survey", *Computer Networks*, vol. 54, no. 15, pp.2688-2710, October 2010.
- [23] J. Lester Hill, "System Architecture for Wireless Sensor Networks", PhD Thesis, University of California, Berkeley, 2003.
- [24] I. F. Akyildiz, M. C. Vuran, "Wireless Sensor Networks", Edition, Wiley, ISBN 978-0-470-03601-3(H/B), 2010.
- [25] K. Akkaya, M. Younis, "A survey on routing protocols for wireless sensor networks", *Ad Hoc Networks*, vol. 3, no. 3, pp.325-349, May 2005.

- [26] C. Intanagonwiwat, R. Govindan, D. Estrin, J. Heidemann, and F. Silva, "Directed diffusion for wireless sensor networking". *IEEE/ACM Transaction Networks*, vol. 11, no. 1, pp.2-16, February 2003.
- [27] C.R. Lin, M. Gerla, Adaptive clustering for mobile wireless networks, *IEEE Journal on Selected Areas in Communications*, vol. 15, no. 7, pp. 1265–1275, 1997.
- [28] A. Manjeshwar, D.P. Agrawal, APTEEN: a hybrid protocol for efficient routing and comprehensive information retrieval in wireless sensor networks, in: *Proceedings of the 2nd International Workshop on Parallel and Distributed Computing Issues in Wireless Networks and Mobile computing*, Ft. Lauderdale, FL, April 2002.
- [29] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," in *Proceedings of the 33rd Annual International Conference on System Sciences*, pp. 1–10, Hawaii, January 2000.
- [30] M. Younis, M. Youssef and K. Arisha, "Energy-aware routing in cluster-based sensor networks", In the *Proceedings of the 10th IEEE International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunications Systems (MASCOTS'2002)*, pp.129-136, 2002.
- [31] A. Manjeshwar, D.P., Agrawal, "TEEN: A Protocol for Enhanced Efficiency in Wireless Sensor Networks". In: *Proceedings of the 1st International Workshop on Parallel and Distributed Computing Issues in Wireless Networks and Mobile Computing*, San Francisco, April 2001.
- [32] M. Radi, B. Dezfouli, K. Abu Bakar and M. Lee, Multipath Routing in Wireless Sensor Networks: Survey and Research Challenges, *Sensors*, vol. 12, no. 1, pp.650-685, 2012
- [33] L. Zhang, M. Hauswirth, L. Shu, Z. Zhou, V. Reynolds, and G. Han, "Multi-priority Multi-path Selection for Video Streaming in Wireless Multimedia Sensor Networks", In *Proceedings of the 5th International Conference on Ubiquitous Intelligence and Computing (UIC '08)*, pp.439-452, Frode Eika Sandnes, Yan Zhang, Chunming Rong, Laurence T. Yang, and Jianhua Ma (Eds.). Springer-Verlag, Berlin, Heidelberg.
- [34] M. Chen, V. Leung, S. Mao, and Y. Yuan, "Directional geographical routing for real-time video communications in wireless sensor networks", *Computer Communications*, vol. 30, no. 17, pp. 3368–3383, November 2007.
- [35] M. Zuniga and B. Krishnamachari, "Analyzing the transitional region in low power wireless links", In the *Proceedings of the 1st Annual IEEE Conference on Sensor and Ad Hoc Communications and Networks*, pp.517-526, 2004.

Résumé

Le routage de données est l'une des plus importantes problématiques rencontrées dans les réseaux de capteurs sans fil. Dans ce contexte nous avons évalué le protocole de routage GPSR puis nous avons proposé une amélioration de ce protocole. La version améliorée proposée permet un routage multi-chemins dans GPSR ce qui permet d'une part de le rendre tolérant aux pannes et d'autre part un protocole ayant un délai réduit pour la livraison des paquets à la station de base. En outre, nous avons utilisé la technique de la voie angulaire pour minimiser la consommation d'énergie. Les résultats de simulation obtenus ont fait preuve de ces améliorations.

Mots clés :

RCSF, GPSR, GPSR multi-chemin, localisation géographique, vu-angulaire, consommation d'énergie

Abstract

The data routing is one of the most important problems encountered in wireless sensor networks.

In this context we evaluated the GPSR routing protocol then we proposed an improvement of this protocol. The enhanced version allows a multi-path routing scheme in GPSR to make it a fault-tolerant protocol and also a routing protocol with a reduced delay for the delivery of packets to the base station. In addition, we used the technique of the angular path to minimize energy consumption. The simulation results showed improvements.

Keywords:

WSN, GPSR, GPSR multi-path, geographical location , angular saw , energy consumption.

ملخص

في اطار مشروع التخرج قمنا بدراسة بروتوكول GPSR و تقييمه و من النتائج المتحصل عليها اقترحنا نسخة مطورة لهذا الاخير الذي يتخذ مسارات متعددة تكون قادرة على تجاوز الاعطاب و الاخطاء و من الناحية التالية استعمال البعث الموجه للمعطيات , يسمح بالمحافظة على الطاقة و قد أظهرت نتائج المحاكات على ان اقتراحنا كان جد مجدي .

كلمات المفتاحية

GPSR, GPSR, RCSF متعدد المسارات, الموقع الجيوغرافي, وجهة نظر الزاوية, استهلاك الطاقة.