

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE
UNIVERSITE MOULOUD MAMMERI DE TIZI OUZOU
FACULTE DE GENIE ELECTRIQUE ET INFORMATIQUE
DEPARTEMENT D'INFORMATIQUE



Mémoire

De fin d'études

En vue de l'obtention du diplôme de Master 2 en informatique

Option : Réseaux, mobilité et systèmes embarqués

Thème

Elaboration d'un protocole de routage efficace en
énergie pour les réseaux de capteurs sans fil

Proposé et dirigé par

M S.TALBI

Réalisé par

M^{lle} Sadia INGRACHEN

M^{lle} Sarah OUBABAS

Soutenu publiquement le 11 septembre 2013 devant le jury :

Mme AOUDJIT Rachida

président

Mme BELKADI Malika

Examineur

M^{lle} CHAMEK

Examineur

M TALBI Said

Promoteur

Promotion 2012/2013

Remerciements

En premier lieu, nous tenons à remercier notre promoteur M.TALBI pour sa disponibilité, ses idées, ses conseils et son encouragement depuis le début de ce travail jusqu'à la fin

Nous tenons également à remercier chaleureusement nos très chers parents pour leur présence, leur encouragement, leur compréhension et leur soutien indéfectible qu'ils ont su nous apporter à tout moment

Nos remerciements les plus sincères vont à nos amies qui étaient toujours là durant les moments critiques à nous écouter

Enfin, nos remerciements s'adressent aux membres du jury qui vont nous faire l'honneur de juger notre travail

Résumé

Le développement récent dans les domaines des technologies de communication sans fil et de la micro-électronique ont donné naissance à des capteurs de très petites tailles à un coût considérablement réduit. Les caractéristiques attractives de ces dispositifs électroniques et les différents types de capteurs commercialisés ont permis leur incorporation dans différents types d'applications. Toutefois, ces composants miniatures sont contraints en termes de ressources énergétiques vu qu'ils sont alimentés par de petites batteries interchangeables et non rechargeables. Pour cette raison, l'énergie est la ressource la plus critique pour un réseau de capteur qu'il convient de gérer de la manière la plus attentive. La tâche de communication consomme beaucoup plus d'énergie comparée aux autres, telle la sensation ou le traitement. Dès lors, plusieurs approches de routage ont été élaborées dans le but de conserver l'énergie des capteurs et maintenir leur fonctionnement aussi longtemps que possible.

EAR (Energy Aware Routing protocol) est un exemple de protocoles de routage pour réseaux de capteurs à topologie plate qui permet de minimiser la dissipation de l'énergie. Etant basé sur un routage multi-chemins, EAR conserve l'énergie des nœuds se trouvant sur la route vers la SB et évite la mort prématurée de ces derniers. Le but de notre travail est de développer le protocole existant EAR en profitant des avantages de l'approche de clustering qui permet de balancer les charges entre les nœuds du réseau. Les résultats de simulations obtenus ont montré que HEAR permet une réduction de la dissipation de l'énergie et prolonge la durée de vie du réseau par rapport au protocole de base EAR.

Mots clés : réseaux de capteurs sans fil, routage, conservation d'énergie, clustering, nœud gateway.

Table des matières

Résumé	II
Liste des figures	IX
Liste des tableaux	X
Introduction Générale.....	1
Chapitre 1 : Généralités sur les réseaux de capteurs sans fil.....	3
1.1. Introduction	4
1.2. Réseaux sans fil	4
1.2.1. Définitions et caractéristiques	4
1.2.2. Classification des réseaux sans fil	5
1.3. Réseaux de capteurs sans fil	8
1.3.1. Définition d'un réseau de capteurs (RCSF).....	8
1.3.2. Qu'est-ce qu'un capteur ?.....	9
1.3.3. Anatomie d'un capteur	9
1.3.4. Caractéristiques d'un capteur	13
1.3.5. Caractéristiques d'un réseau de capteurs.....	13
1.3.6. Communication dans les réseaux de capteurs	15
1.3.7. Domaines d'application.....	20
1.4. Axes de recherche actuels	22
1.4.1. Routage.....	22
1.4.2. Couche MAC.....	23
1.4.3. Dissémination de données	23
1.4.4. Sécurité	23
1.4.5. Qualité de service	23
1.4.6. Localisation	24
1.5. Conclusion.....	24
Chapitre 2 : Routage dans les réseaux de capteurs sans fil	25
2.1. Introduction	26
2.2. Routage dans les réseaux de capteurs.....	26
2.3. Facteurs influant sur la conception d'un protocole de routage.....	26
2.3.1. Mobilité du réseau	27
2.3.2. Déploiement des nœuds.....	27

2.3.3.	Consommation d'énergie.....	27
2.3.4.	Modèle de livraison de données	27
2.3.5.	Hétérogénéité Nœud/Lien	28
2.3.6.	Tolérance aux fautes.....	28
2.3.7.	Scalabilité	28
2.3.8.	Connectivité.....	28
2.4.	Métriques de routage	28
2.4.1.	Métriques pour la consommation énergétique.....	29
2.4.2.	Route à nombre de sauts minimum	29
2.4.3.	Perte de paquets.....	29
2.5.	Taxonomie des protocoles de routage	30
2.5.1.	Selon le paradigme de communication.....	30
2.5.2.	Selon la structure du réseau.....	31
2.5.3.	Selon le modèle de livraison de données.....	32
2.5.4.	Selon le mode d'établissement des chemins.....	34
2.5.5.	Selon le mode de fonctionnement du protocole	35
2.6.	Conclusion.....	36
Chapitre 3: Etat de l'art sur les protocoles de routage hiérarchiques dans les RCSFs		37
3.1.	Introduction	38
3.2.	Notions de base	38
3.2.1.	Définition.....	38
3.2.2.	Processus de clustering.....	39
3.2.3.	Avantages de l'approche de clustering.....	40
3.3.	Objectifs du clustering.....	41
3.3.1.	Equilibrage des charges.....	41
3.3.2.	Tolérance aux fautes.....	41
3.3.3.	Amélioration de la connectivité et réduction des délais de transmission	42
3.3.4.	Augmentation de la durée de vie du réseau.....	42
3.3.5.	Caractéristiques d'un protocole hiérarchique	42
3.4.	Exemples de protocoles de routage hiérarchiques pour les réseaux de capteurs.....	44
3.4.1.	LEACH (Low Energy Adaptive Clustering Hierarchy)	44
3.4.2.	DECSA (Distance-Energy Cluster Structure Algorithm).....	46
3.4.3.	EADUC (Energy Aware Distributed Unequal Clustering Protocol).....	48
3.4.4.	CW-EBCR (combination weight energy balance clustering and routing algorithm)	49

3.4.5.	UCRA (Unequal Clustering Routing Algorithm).....	51
3.4.6.	PEGASIS (Power-Efficient Gathering in Sensor Information System).....	54
3.4.7.	HEED (Hybrid Energy Efficient Distributed Approach)	54
3.4.8.	ADRP (Adaptive Decentralized Re-clustering Protocol).....	55
3.5.	Conclusion.....	57
Chapitre 4 : HEAR : un nouveau protocole de routage hiérarchique pour les RCSF		58
4.1.	Introduction	59
4.2.	Protocole EAR (Energy Aware Routing)	59
4.3.	Nouvelle approche de routage pour les RCSFs hiérarchiques	61
4.3.1.	Motivations.....	61
4.3.2.	Hypothèses	62
4.3.3.	Principe de fonctionnement.....	63
4.4.	Discussion sur l'efficacité du protocole	72
4.4.1.	Auto organisation	72
4.4.2.	Fonction pondérée d'élection des cluster-heads.....	73
4.4.3.	Routage multi-sauts au niveau des cluster-heads	73
4.4.4.	Agrégation des données.....	73
4.5.	Conclusion.....	73
Chapitre 5 : mise en œuvre et simulation		74
5.1.	Introduction	75
5.2.	Exemples de simulateurs de RCSFs	75
5.2.1.	NS2	75
5.2.2.	TOSSIM	76
5.2.3.	J-Sim.....	76
5.2.4.	OMNeT++	77
5.3.	Environnement de simulation.....	78
5.3.1.	Modèle de simulation implémenté	78
5.3.2.	Type de messages échangés	80
5.4.	Résultats de simulation.....	81
5.4.1.	Critères de performances.....	81
5.4.2.	Scénario de simulation	82
5.4.3.	Discussion des résultats	83
5.5.	Conclusion.....	86
Conclusion générale et perspectives.....		87

Bibliographie	89
---------------------	----

Liste des figures

Fig.1.1: principe d'un réseau cellulaire.....	7
Fig.1.2: Exemple réseau ad hoc	8
Fig.1.3: nœuds senseurs répartis dans un champ de captage.....	8
Fig.1.4: Exemples de nœuds capteurs (Berkeley Mote).....	12
Fig.1.5: Pile protocolaire dans les réseaux de capteurs.....	16
Fig.1.6: Vue d'ensemble d'un RCSF destiné à une application de monitoring médicale.....	20
Fig.2.1 : Topologie plate.....	32
Fig.2.2 : Topologie hiérarchique.....	32
Fig.2.3 : Acheminement des données dans un RCSF.....	34
Fig3.1: Structure hiérarchique d'un réseau de capteurs.....	49
Fig.3.2 : Architecture du réseau en clusters dans LEACH.....	54
Fig.3.3 : Procédure de routage de l'algorithme de routage multi-sauts MEC.....	52
Fig.4.1: Phase d'initialisation : Phase d'élection des cluster-heads.....	68
Fig.4.2: Phase d'initialisation : réception d'un message d'annonce.....	70
Fig.4.3 : Phase de construction des chemins.....	72
Fig4.4 : Phase de transmission de données.....	72
Fig.5.1: Architecture du réseau selon le modèle de simulation implémenté	81
Fig.5.2 : Moyenne d'énergie consommée.....	84
Fig.5.3 : Ecart type de l'énergie consommée.....	84
Fig.5.4 : Durée de vie du réseau.....	85

Liste des tableaux

Tab.1.1 : Classes de RSFs selon l'étendue et leurs caractéristiques.....	18
Tab.2.1 : Taxonomie des protocoles de routage.....	41
Tab.4.1 : Entête du paquet NewRoundMsg.....	73
Tab.4.2 : Entête du paquet HelloMsg.....	73
Tab.4.3 : Entête du paquet WeightUpdate.....	74
Tab.4.4 : Structure de la table de voisinage des nœuds.....	74
Tab.4.5 : Entête d'un message d'annonce (ChAdvMsg).....	77
Tab.4.6: Entête d'un paquet d'affiliation à un cluster.....	77
Tab.4.7: Entête d'un paquet SetUpMsg.....	81
Tab.4.8 : Entête d'un paquet de données DataMsg.....	81
Tab.5.1 : Messages échangés dans HEAR.....	82
Tab.5.2 : Scénario de simulation.....	84

Introduction Générale

Durant ces dernières années, le développement des technologies de communications sans fil et de la fabrication des micro-processeurs embarqués ont permis la production de différents types de composants électroniques miniatures à un coût réduit. Après les réseaux téléphoniques et les réseaux ad hoc, les chercheurs s'orientent actuellement vers les réseaux de capteurs : un nouveau type de réseau sans fil déployé dans des environnements généralement inaccessibles afin de surveiller un phénomène d'intérêt. Un réseau de capteurs est constitué d'un grand nombre de capteurs capables de prélever des grandeurs physiques à partir d'un environnement donné et de les transformer en données numériques, effectuer des traitements éventuels sur ces dernières afin de les transférer vers un utilisateur final. Les caractéristiques et la large gamme de capteurs commercialisés ont permis d'étendre les applications utilisant les réseaux de capteurs pour inclure les applications médicales, environnementales, commerciales, les applications domotiques et autres.

En effet, la taille réduite des capteurs impose des limites sur leurs capacités de traitement, de stockage et plus particulièrement, énergétiques. En outre, le déploiement des réseaux de capteurs dans des environnements difficiles d'accès où ils doivent opérer durant de longues périodes de l'ordre de l'année, pose le problème de remplacement et du rechargement des batteries des capteurs. Par conséquent, l'économie d'énergie représente un défi perpétuel et de première importance afin de prolonger la durée de vie du réseau autant qu'il faut pour atteindre l'objectif initial de l'application pour laquelle le réseau est dédié.

Parmi les tâches qu'effectue un capteur, la communication est de loin l'activité la plus consommatrice d'énergie. Dans le but de procurer un gain maximal en termes d'énergie lors des communications, plusieurs protocoles de routage ont été proposés. Les protocoles de routage dans les RCSFs sont indispensables afin de transférer les données apparues dans le champ de captage vers l'utilisateur final vu l'absence de toute infrastructure tout en considérant les limitations des capteurs.

Un des protocoles de routage proposé, nous citons EAR qui sera à la base de notre travail. Celui-ci est adapté aux topologies plates et se base sur un routage multi-chemins qui permet de maintenir la connectivité du réseau et de prolonger la durée de vie des capteurs en conservant leur énergie.

Toutefois, les réseaux plats présentent l'inconvénient de l'envoi de toutes les données captées vers la station de base même si ces dernières proviennent des capteurs voisins et sont identiques. Afin d'éliminer la redondance des données au niveau de la station de base, la hiérarchisation du réseau s'est présentée comme une approche de routage qui contribue de manière efficace à l'économie d'énergie, la réduction de messages échangés dans le réseau ainsi que l'amélioration de l'utilisation du canal de transmission. Cette approche consiste à diviser le réseau en groupes (cluster) de nœuds guidés, chacun, par un chef de groupe (cluster-head). Le cluster-head coordonne les tâches des nœuds d'un cluster, agrège leurs données et envoie le rapport de l'agrégation vers la station de base.

Notre travail consiste à améliorer le protocole plat EAR en bénéficiant des avantages de l'approche de clustering afin de prolonger la durée de vie du réseau et d'optimiser l'utilisation de ses ressources.

Ce mémoire est organisé en cinq chapitres comme suit :

Le premier chapitre présente des généralités sur les réseaux de capteurs sans fil avec une description de leurs architectures, leurs caractéristiques principales ainsi que le type de communication utilisé dans ces réseaux. Nous spécifions par la suite les axes de recherche actuels.

Le deuxième chapitre est consacré au routage dans les RCSFs. Les facteurs influant sur la conception des protocoles de routage ainsi que certains exemples de ces derniers qui ont été élaborés dans des travaux récents.

Le troisième chapitre étale l'approche de hiérarchisation des protocoles de routage en exposant quelques notions de base sur la méthode de clustering, les objectifs de cette dernière et ses avantages. Les caractéristiques des protocoles de routage hiérarchiques seront également présentées ainsi que des exemples d'algorithmes existants.

Le quatrième chapitre est consacré au nouveau protocole de routage. Il englobe une étude détaillée du fonctionnement du protocole EAR et la description de la nouvelle approche proposée.

Le dernier chapitre commence par la présentation de l'environnement de simulation choisi OMNeT++4.1. Il se termine par une évaluation des performances du nouveau protocole en le comparant au protocole EAR de base.

Chapitre 1 : Généralités sur les réseaux de capteurs sans fil

1.1. Introduction

Les avancées technologiques de ces dernières années ont permis l'amélioration et la miniaturisation de différents dispositifs utilisés pour les applications embarquées et l'apparition de la notion de « capteur sans fil » telle qu'elle est employée aujourd'hui. Les capteurs sont des composants très petits. Ils sont utilisés essentiellement pour surveiller un environnement. Plusieurs technologies ont contribué à produire des composants de type capteur plus petits et à un prix faible. En plus, récemment ces composants sont munis d'un système de communication leur permettant de communiquer avec d'autres capteurs, et ainsi les réseaux de capteurs sans fil sont nés. L'objectif de ce chapitre est de donner une vue d'ensemble sur les réseaux de capteurs sans fil (RCSF). Le point de départ est le capteur en lui-même qui sera ensuite intégré dans un réseau de manière à répondre à une application donnée. Nous allons discuter également de leurs domaines d'applications, leurs architectures de communication ainsi que les principaux facteurs et contraintes qui influencent la conception des réseaux de capteurs sans fil.

1.2. Réseaux sans fil

Actuellement, les communications sans fil sont omniprésentes et leur utilisation est en pleine extension. Dans cette section, nous allons définir ce type de réseaux, exposer leurs caractéristiques ainsi que les différents paramètres utilisés dans leur classification.

1.2.1. Définitions et caractéristiques

Un réseau sans fils (en anglais Wireless Network) est, comme son nom l'indique, un réseau dans lequel au moins deux terminaux peuvent communiquer sans liaison filaire. Grâce aux réseaux sans fils, un utilisateur a la possibilité de rester connecté tout en se déplaçant dans un périmètre géographique plus ou moins étendu.

La solution du sans fil est en pleine extension grâce aux avantages qu'elle offre aux utilisateurs. Ce type de réseau utilise comme support de transmission les ondes radio (radiofréquences), ceci évite aux administrateurs, ainsi que les utilisateurs, la gestion d'infrastructures filaires qui peuvent être trop complexes. Les communications entre les terminaux peuvent s'effectuer directement ou par le biais de la station de base (AP : Access Point). Toutefois, l'absence de l'infrastructure physique dans les réseaux sans fil cause l'interférence des phénomènes physiques externes avec les données transitant sur le canal radio. Les signaux radiofréquences s'atténuent. Cela signifie qu'ils perdent en intensité à mesure qu'ils s'éloignent de leur point d'origine. Cette atténuation du signal peut être un problème dans un réseau local sans fil lorsque les stations rivalisent pour l'accès au support. Le contrôle de l'accès est plus difficile. Afin de remédier à ces insuffisances, la méthode d'accès au support CSMA/CA basée sur l'écoute de porteuse/évitement de collisions a été proposée. Cette méthode est plus efficace en termes de réduction du trafic généré par les reprises sur erreurs connues dans les réseaux filaires.

1.2.2. Classification des réseaux sans fil

Les réseaux sans fil peuvent être classés selon divers critères. Les deux classifications les plus courantes sont la classification selon l'étendue géographique et selon le mode de fonctionnement.

1.2.2.1. Selon l'étendue

La classification selon l'étendue permet de distinguer quatre catégories de RSF : les WPAN (Wireless Personal Area Network), WLAN (Wireless Local Area Network), WMAN (Wireless Metropolitan Area Network), WWAN (Wireless Wide Area Network). Ils sont différenciés par la distance maximale séparant les points les plus éloignés du réseau.

- Réseaux personnels sans fil (WPAN)

Réseaux à faible portée de l'ordre de quelques mètres (15m environ). Permettent la communication entre des assistants personnels (PDAs), des téléphones...etc. Elle est standardisée par le groupe 802.15. Plusieurs sous groupes de ce dernier peuvent être cités comme le sous-groupe 802.15.1 connu sous le nom de Bluetooth, le 802.15.3 qui définit la norme UWB (Ultra Wide Band). Le débit de ces deux derniers est de 480 Mbit/s sur une portée de 3 m et décroît à environ 120 Mbit/s à une dizaine de mètres. Ils offrent des performances moyennes et un débit de 2Mbits/s [02]. Enfin, le 802.15.4 qui s'occupe de la norme ZigBee, dont le but est d'assurer des communications à courte distance comme le propose la norme Bluetooth, mais plus simple et moins cher.

- Réseaux locaux sans fil (WLAN)

Les réseaux locaux sans fil constituent une extension des infrastructures LAN Ethernet 802.3. standardisés par 802.11 ou WiFi (*Wireless fidelity*). Ils autorisent des débits allant de 2 à 54Mbits/s. Les débits de données des différentes normes de réseaux locaux sans fil sont affectés par la technique de modulation utilisée.

- Réseaux métropolitains sans fil (WMAN)

Les WMAN sont utilisés pour l'accès aux réseaux d'infrastructure (boucle locale). La norme principale est 802.16 qui offre des débits de plusieurs dizaines de Mbit/s avec une portée atteignant plus de dix kilomètres. Un exemple de WMAN, le WiMax (Worldwide Interoperability for Microwave Access). Il représente une classe de réseaux caractérisée par de grandes cellules, de la taille d'une métropole. WiMAX se présente en deux versions, une version fixe, qui a été finalisée sous le nom de WiMAX IEEE 802.16-2004 et la version mobile IEEE 802.16e-2005. L'utilisation de WiMAX est très semblable à celle d'un modem ADSL, si ce n'est qu'au lieu d'un câble téléphonique, on utilise la voie hertzienne. C'est pourquoi l'on parle de WDSL (Wireless DSL) pour décrire la solution WiMAX fixe. En ce qui concerne la version mobile, son utilisation est identique à celle d'un ADSL mobile. L'avantage évident de WiMAX mobile est sa capacité à supporter la mobilité des utilisateurs. Avec un équipement muni d'une connexion WiMAX, un utilisateur peut se connecter de partout, tout le temps.

- Réseaux étendus sans fil (WWAN)

Recouvrent essentiellement les réseaux de transmission de voix avec leurs extensions données (GSM, GPRS et UMTS), les débits sont relativement faibles, estimés à quelques dizaines de kbit/s (10 à 384 kbit/s). Standardisé par la norme 802.20 correspondant aux solutions cellulaires permettant de recouvrir des zones de l'ordre d'un pays.

Le tableau suivant résume les classes de RSFs citées précédemment ainsi que leurs principaux standards.

Catégorie	WPAN	WLAN	WMAN	WWAN
Standard	IEEE 802.15	IEEE 802.11	IEEE 802.16	IEEE 802.20
Technologies	Bluetooth HomeRF Zigbee IR(Infrarouge)	Wifi HyperLAN2	Wimax	GSM GPRS UMTS
Couverture	Quelques dizaines de mètres	Une centaine de mètres	Quelques dizaines de kilomètres	Une centaine de kilomètres
Débit	< 1 Mbps	2 à 54 Mbps	Jusqu'à 70 Mbps	10 à 384 Kbps
Applications	Point à point Équipement à équipement	Réseau d'entreprise	Fixe, accès au dernier Kilomètres	GSM PDA...

Tab.1.1 Classes de RSFs selon l'étendue et leurs caractéristiques

1.2.2.2. Selon le mode de fonctionnement

Les réseaux sans fil peuvent être soit indépendants de toute infrastructure filaire soit une continuation de celle-ci.

- Réseaux avec infrastructure (cellulaires)

Composés de cellules ou de zones géographiques qui peuvent être limitées à quelques kilomètres. Une cellule est centrée autour de la base radio (AP: Access Point). Une cellule est constituée de l'ensemble formé par le point d'accès et les stations situées dans sa zone de couverture appelé couramment *ensemble de service de base* (Basic Service Set noté BSS). Lorsqu'un mobile quitte une cellule (BSS, Basic Service Set), pour maintenir la communication, il doit être accueilli par une autre cellule. Les techniques de gestion de la mobilité sont différentes selon que le mobile soit un mobile voix (téléphone) ou un mobile données (station). Une communication téléphonique peut être interrompue quelques millisecondes sans nuire à l'intelligibilité de la conversation (temps de basculement d'une cellule vers une autre). Dans un service données, la moindre interruption provoque une erreur de transmission [01].

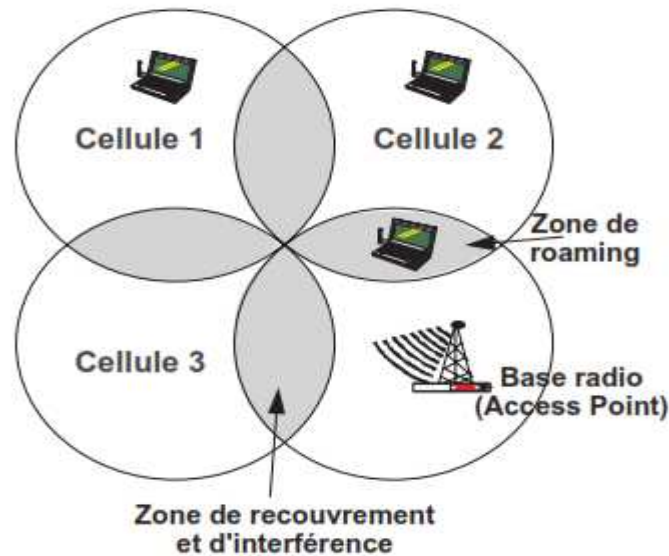


Fig.1.1: principe d'un réseau cellulaire.

- Réseaux sans infrastructure (ad hoc)

L'infrastructure dans un réseau ad hoc est réduite aux seules stations qui jouent, en même temps, le rôle de terminaux et de routeurs pour garantir le transfert de l'information entre les terminaux du réseau sans qu'ils soient directement connectés. La communication de deux stations trop éloignées l'une de l'autre passe par des nœuds intermédiaires qui jouent le rôle de relais. Ces terminaux peuvent être statiques (ordinateurs de bureau) ou mobiles (PDA, ordinateurs portables, téléphones, véhicules... etc.) et créent ainsi un réseau dont le nombre de terminaux et la topologie évoluent dans le temps. L'intérêt de ce mode de fonctionnement est en premier lieu la réduction du coût de mise en place d'une infrastructure. De plus, il permet des communications de proche en proche dans des endroits où il n'est pas possible d'installer une infrastructure complète pour des raisons techniques ou fonctionnelles [03]. Le routage des informations se fait soit directement c'est-à-dire que les nœuds se communiquent sans intermédiaire, ce qui garantit plus de sécurité pour la transmission, ou se fait par le biais de nœuds intermédiaires, dans ce cas, les informations ne sont pas routées directement vers leur destination.

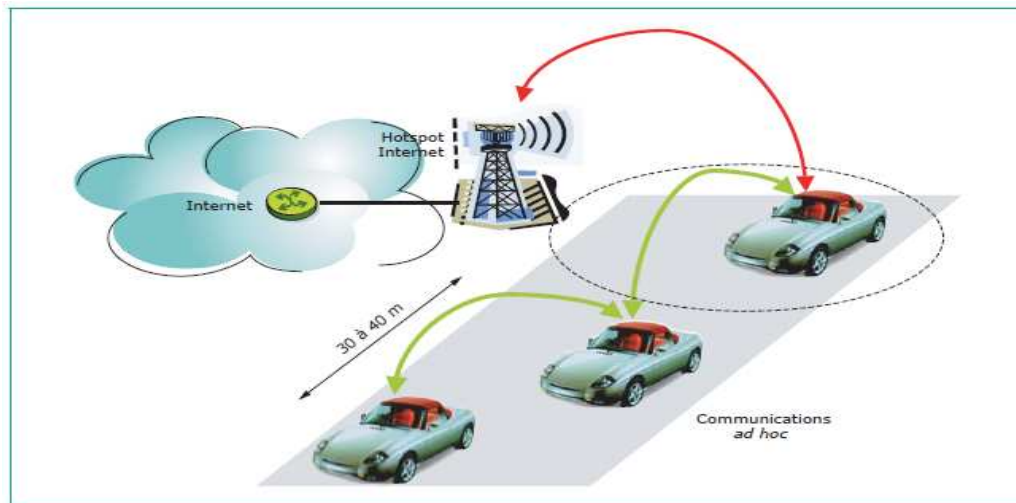


Fig.1.2 : Exemple de réseau ad hoc [3]

1.3. Réseaux de capteurs sans fil

Les progrès réalisés ces dernières années dans le domaine de la micro électronique ont donné naissance à un autre type de réseaux ad hoc sans fil qui sont les réseaux de capteurs. Ce type de réseaux profite du développement de la miniaturisation des circuits électroniques pour prendre sa place dans tous les domaines.

1.3.1. Définition d'un réseau de capteurs (RCSF)

Un réseau de capteurs sans fil (Wireless sensor Network : WSN) est un réseau ad hoc multi-saut à grande échelle (composé de centaines voire des milliers de capteurs) dédiés pour la surveillance de zones d'intérêt appelée zone de captage. Ces réseaux sont déployés de manière aléatoire les capteurs sont catapultés depuis un hélicoptère). Le déploiement se fait généralement dans des endroits hostiles où l'intervention humaine n'est pas facile, voire impossible. Les capteurs communiquent entre eux les données captées via un module sans fil utilisant des signaux radiofréquences pour le partage d'informations et le traitement coopératif. Contrairement aux réseaux ad hoc qui se base sur les communications point à point, les réseaux de capteurs utilisent le mode de communication broadcast (diffusion). Deux types de nœuds capteurs peuvent être distingués : capteurs (nœuds ordinaires) et nœud puits [38]. Un nœud capteur détecte les phénomènes physiques et surveille son environnement immédiat pour transmettre les données nécessaires au nœud puits appelé aussi Station de Base. Ce dernier collecte les données, les stocke et les analyse pour les utiliser dans une application spécifiée au préalable ou les communiquer à d'autres réseaux auxquels il est relié par Internet ou Satellite [38,39].

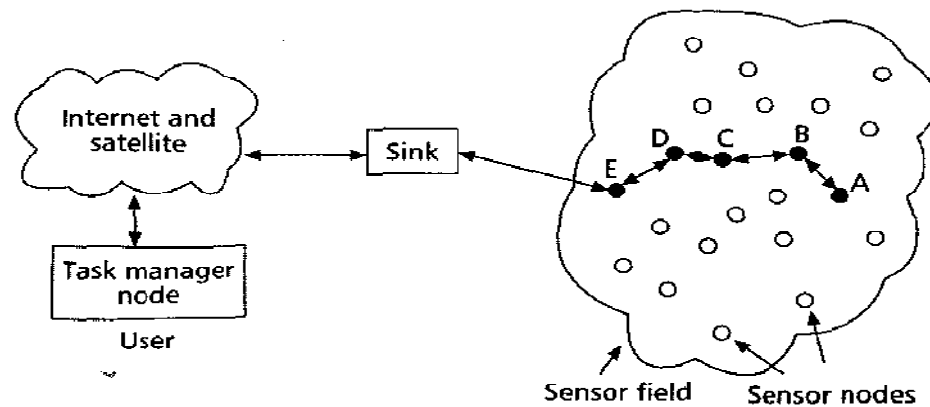


Fig.1.3 : nœuds senseurs répartis dans un champ de captage [38].

1.3.2. Qu'est-ce qu'un capteur ?

Un capteur est un système embarqué capable de récolter une valeur physique environnementale (température, pression,...) selon l'application pour laquelle il est conçu. Un capteur est composé de quatre unités qui lui permettent l'accomplissement de tâches de sensation, de calcul et de communication. Un capteur collecte des grandeurs physiques qu'il convertit en données numériques. Ces dernières seront par la suite envoyées directement à la SB dans leur état brut ou après avoir subi un traitement donné, et ce, pour les utiliser à des fins de commande.

1.3.3. Anatomie d'un capteur

Un capteur est composé de deux parties : une partie matérielle(hardware), composée à son tour, de manière générale d'une unité de transmission ou transceiver , une unité de traitement, une unité de captage et une unité de stockage et une partie logiciel(software) qui constitue le système d'exploitation et les applications embarqués sur le capteur.

1.3.3.1. Partie hardware

Une des plateformes hardware des nœuds senseur la plus utilisée on cite Mica2 mote développée par Crossbow Tecnology [12]. Toutes les plateformes hardware des capteurs ont en commun les composants suivants [12,40] :

- Module transmission

Ou unité de communication pour l'échange de données avec d'autres nœuds du réseau. Dotée d'une interface radio qui émet à courtes distances, cette unité est la plus gourmande en énergie et sa consommation augmente lors de la transmission ou de la réception et dépend de la puissance d'émission.

Il existe trois technologies de communication entre les capteurs: La communication optique (laser), la communication infrarouge et la communication par fréquence radio [09] :

Infrarouge : ne nécessite aucune antenne mais il a une capacité limitée pour la diffusion. Les systèmes infrarouges ont besoin d'un champ de vision (lignes de mire) ce qui n'est pas toujours réalisable.

Laser : Les communications de type optique sont robustes vis-à-vis des interférences électriques. Elles consomment moins d'énergie par rapport à l'infrarouge. Néanmoins, elles présentent l'inconvénient d'exiger une ligne de vue permanente entre les entités communicantes. Par conséquent, elles ne peuvent pas établir de liaisons à travers des obstacles mais il nécessite une antenne et il est sensible aux changements atmosphériques.

Radio fréquence (RF) : Les unités de transmission de type radiofréquence sont faciles à utiliser mais leur fabrication est plus complexe. Elles comprennent des circuits de modulation, démodulation, filtrage et multiplexage; ce qui implique une augmentation de la complexité et du coût de production du micro-capteur. Concevoir des unités de transmission de type radiofréquence avec une faible consommation d'énergie est un véritable défi. En effet, pour qu'un nœud ait une portée de communication suffisamment grande, il est nécessaire d'utiliser un signal assez puissant. L'énergie consommée augmente très vite en fonction de la portée du signal radio [35]. La consommation est de l'ordre de 20mW pour une portée de quelques dizaines de mètres. L'autre alternative serait d'utiliser de longues antennes, mais ceci n'est pas possible à cause de la taille réduite des capteurs.

Afin de minimiser la consommation de l'énergie, les nœuds capteurs éteignent leurs modules radio durant les périodes d'inactivité. Typiquement, le module sans fil d'un nœud capteur a trois modes opérationnels. Avant toute utilisation, un capteur est initialisé, c'est l'état *Init*. Puis il fonctionne selon trois modes : le mode *Actif* où il exécute une fonction ou transmet un message, le mode *Idle* où il est seulement à l'écoute d'éventuels messages à recevoir et le mode *Sleep* où il est en veille. Enfin il se met à l'état *Stop* lorsque la batterie est vide ou s'il est mis hors tension. Lorsqu'il est actif, le capteur peut envoyer et recevoir des messages et capter des événements (et seulement recevoir s'il est dans l'état *Idle*). En ce qui concerne la transmission des messages il y a trois états. Avant toute transmission, le réseau doit être vide de message après quoi le capteur peut, soit envoyer, soit recevoir un message [10,37]. Un capteur ne peut pas envoyer et recevoir simultanément des messages, il fonctionne en half duplex [10].

- **Unité de capture**

Se compose de deux modules : un capteur qui prend des mesures physiques (pression, son, température, vibration, pollution, radiation,...etc.) et un convertisseur analogique-numérique noté par ADC (Analog to Digital Converter) qui transforme le signal analogique en données numériques et les transmet à l'unité de traitement [13].

- **Unité de traitement**

Elle est composée d'un processeur embarqué et d'une unité de stockage. Le rôle du processeur embarqué dans un nœud capteur est d'ordonnancer les tâches, traiter les données et contrôler d'autres composants du senseur. Les types de processeurs qui peuvent être utilisés dans un capteur incluent : le microcontrôleur, les DSP (Digital

Signal Processor), les FPGA (Field Programmable Gateway Array), et les ASIC (Application Specific Integrated Circuit). Parmi toutes ces alternatives, les microcontrôleurs sont les plus utilisés grâce à leur coût et leur faible consommation énergétique [12]. La consommation du microcontrôleur dépend de sa vitesse de traitement. L'unité de traitement fonctionne à l'aide d'un système d'exploitation spécialement conçu pour les réseaux de capteurs.

L'unité de stockage inclut la mémoire réservée aux programmes dont les instructions sont exécutées par le processeur et la mémoire réservée aux données (pour conserver les données fournies par l'unité de captage). La taille de cette mémoire est souvent limitée essentiellement par les considérations économiques.

- **Une batterie**

Un capteur est doté d'une batterie qui alimente le reste de ses composants. En conséquence de sa petite taille, ses ressources sont limitées. Elle est souvent interchangeable puisque les capteurs sont déployés dans des environnements inaccessibles. Pour cela, l'énergie constitue la ressource la plus critique pour un réseau de capteurs qu'il convient de gérer scrupuleusement pour optimiser la durée de vie du capteur et ainsi la durée de vie du réseau.

En plus, un capteur peut avoir d'autres unités selon l'application pour laquelle il a été conçu.

- **Unité de localisation**

Dans plusieurs applications de réseaux de capteurs, les nœuds et les données collectées doivent être localisés dans la zone de captage afin de parvenir à déterminer le lieu où l'événement est apparu [40]. Le positionnement est garanti par l'utilisation d'un système de localisation global (GPS). Ces systèmes sont très importants pour certaines applications de RCSFs telle la poursuite d'une cible car, ils ne participent pas seulement à la localisation mais aussi aux décisions de routage et au contrôle de la densité du réseau [08].

- **Générateur d'énergie**

Peut être sous forme de cellules solaires qui transforment l'énergie solaire en électricité afin de prolonger la durée de vie de la batterie.

- **Unité assurant la mobilité**

Certaines applications nécessitent la mobilité des nœuds afin d'assurer une couverture complète du champ de captage. Par exemple, une application de poursuite d'une cible, Le changement dynamique de la zone d'intérêt nécessite une mobilité des nœuds.

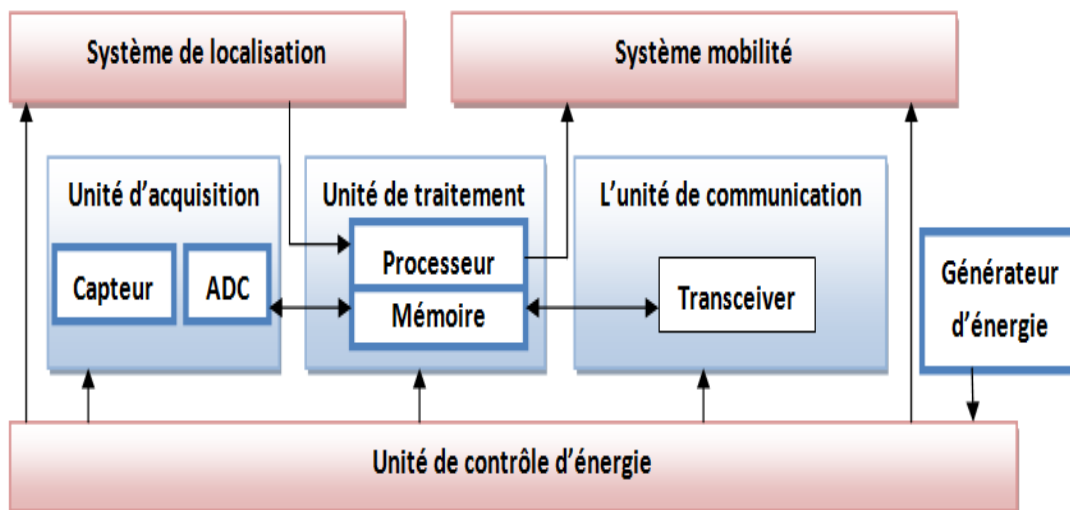


Fig.1.4 : anatomie d'un capteur

1.3.3.2. Partie software

Les systèmes d'exploitation pour les capteurs doivent présenter plus de performance en termes d'efficacité, temps d'exécution, d'occupation de la mémoire et surtout en termes de consommation de l'énergie. Le système d'exploitation le plus utilisé pour les réseaux de capteurs est TinyOs [12]. TinyOs est un système d'exploitation open source. Sa conception a été entièrement réalisée en NesC : un langage orienté composants, syntaxiquement proche du C [64]. TinyOS fournit un ensemble de composants essentiels comme les pilotes d'éléments hardware, un ordonnanceur et des protocoles de communication de base [18,41]. Ces composants peuvent être utilisés tels qu'ils sont, comme ils peuvent être aussi adaptés à des applications précises. En s'appuyant sur un fonctionnement événementiel, TinyOS propose une gestion très précise de la consommation d'énergie du capteur et permet une meilleure adaptation à la communication sans fil [42].

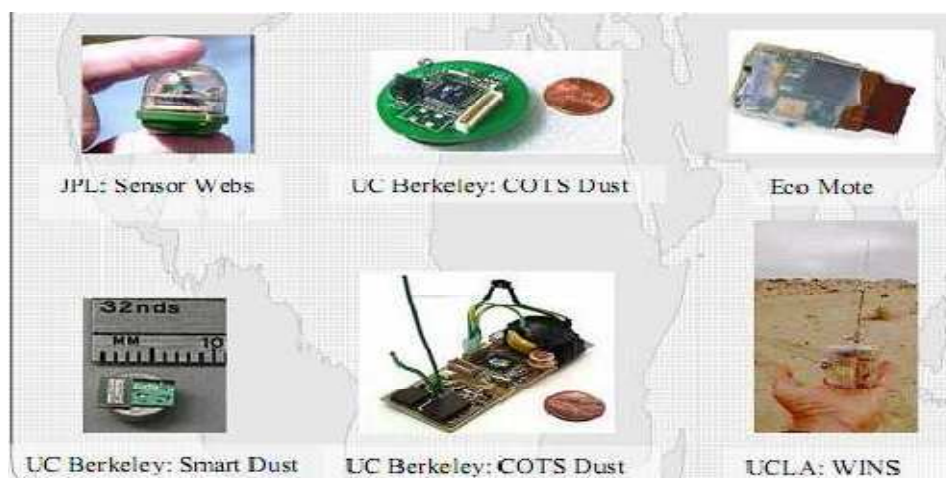


Fig.1.5 : Exemples de nœuds capteurs (Berkeley Mote)

1.3.4. Caractéristiques d'un capteur

Actuellement, les capteurs sont des composants incontournables dans plusieurs applications. Grâce à la grande variété de capteurs existante, ils sont considérés comme meilleur moyen d'interroger les phénomènes physiques. Le choix du type de capteurs dépend de l'application et de l'environnement dans lequel ils seront déployés. Les capteurs offrent différents services grâce à certaines caractéristiques dont les plus importantes sont :

1.3.4.1. Sensibilité

Elle est définie par le rapport entre le signal physique en entrée et le signal électrique de sortie. Caractérise ainsi l'aptitude du capteur à détecter la plus petite variation de la grandeur à mesurer. A titre d'exemple, une sensibilité élevée d'un capteur thermique se produira lors qu'un petit changement de température correspond à un grand changement de tension.

1.3.4.2. Précision

La précision d'un capteur s'exprime en fonction de la grandeur physique mesurée. Elle est généralement présentée comme étant la plus grande erreur prévue entre le réel perçu et les signaux électriques obtenus en sortie du capteur.

1.3.4.3. Rapidité

C'est le temps de réaction d'un capteur entre la variation de la grandeur physique qu'il mesure et l'instant où l'information sera prise en compte par la partie commande.

1.3.4.4. Fidélité

Pour une série de mesures de la même valeur de la grandeur d'entrée, le signal délivré par un capteur ne varie pas.

1.3.4.5. Bruit

Tous les capteurs produisent un certain bruit de rendement, en plus du signal de sortie. Dans beaucoup de cas, le bruit généré limite les performances d'exécution du système embarqué sur le capteur.

1.3.5. Caractéristiques d'un réseau de capteurs

Vues leur petites tailles, les capacités de traitement et de communication des capteurs sont limitées. La conception d'un RCSF est contrainte par plusieurs facteurs dus aux limitations des nœuds capteurs et de la nature des applications des RCSFs qui déterminent l'approche à adopter lors du choix d'algorithmes à implémenter sur les capteurs et en voici quelques uns :

1.3.5.1. Energie limitée

Les RCSFs étant constitués de micro senseurs tirant leur autonomie d'énergie à partir de petites batteries interchangeable et souvent déployés dans des environnements hostiles, par conséquent, leur maintenance se révèle difficile voire impossible. De ce fait, une des contraintes les plus importantes pour les nœuds capteurs, le besoin de consommer de basses

quantités d'énergie. Contrairement aux réseaux ad hoc qui visent à garantir une meilleure qualité de service (QoS), les protocoles conçus pour les réseaux de capteurs doivent s'intéresser essentiellement à la conservation de l'énergie. Ils doivent implémenter des mécanismes qui évitent les échanges inutiles entre les nœuds afin de prolonger la durée de vie du réseau [39]. Les expériences ont montré que la transmission de données est très coûteuse en termes de consommation d'énergie alors que leur traitement est significativement inférieur [11].

1.3.5.2. Taille importante

Le nombre de nœuds déployés pour un projet peut atteindre le million. Un nombre aussi important de nœuds engendre beaucoup de transmissions inter nodales et nécessite que le puits "sink " soit équipé de beaucoup de mémoire pour stocker les informations reçues.

1.3.5.3. Densité

Les RCSFs sont caractérisés par une grande densité des nœuds déployés dans la région à surveiller. La densité est reliée à des besoins de fiabilité pour assurer la disponibilité des données.

1.3.5.4. Sécurité limitée

La sécurité constitue l'un des problèmes majeurs des réseaux de capteurs sans fil. Ceci est dû au fait que ces derniers sont déployés dans des endroits hostiles (volcans, zones sinistrées,...) ce qui peut entraîner la destruction physique des nœuds. Les ressources mémoire, énergétiques et de traitement relativement limitées des capteurs ne permettent pas l'application des solutions de sécurité existantes sur eux telle la cryptographie à clé publiques, alors il convient de proposer des protocoles de sécurité non gourmands en termes de consommation énergétique et traitement.

1.3.5.5. Topologie dynamique

La vulnérabilité physique des capteurs est à l'origine des changements très fréquents de la topologie d'un réseau de capteurs. Les capteurs sont détruits lors de leur éjection d'un avion dans la zone de déploiement ou après avoir épuisé leurs batteries. Les nœuds qui disparaissent sont remplacés par d'autres pour ne pas isoler des parties du réseau.

1.3.5.6. Interférence

Les réseaux de senseurs utilisent comme support de transmission le canal radio. Vu que les ondes radiofréquences sont très sensibles aux perturbations électromagnétiques et électriques environnantes, les communications sont sujettes à des déformations.

1.3.5.7. Mode d'adressage

Les réseaux de capteurs sont déployés de manière très dense qui peut atteindre des milliers dans une zone de quelques mètres. Ce qui fait que l'utilisation de l'adressage IP utilisé dans les réseaux filaires est trop lourde.

1.3.5.8. Ressources réduites

Les nœuds qui composent un réseau de senseurs sont de petite taille et par conséquent ont des ressources de calcul, de stockage, de communication et d'énergie très limitées.

1.3.5.9. Agrégation de données

Les capteurs génèrent de grands volumes de données brutes. Alors, il est nécessaire de développer des techniques efficaces en énergie pour convertir ces données en des connaissances plus utiles.

Il a été montré dans plusieurs publications scientifiques que la transmission d'un bit est équivalente, en termes d'énergie, à l'exécution d'environ 1000 instructions. Cette valeur augmente avec la portée de la radio. Plus le capteur devra transmettre loin, et par conséquent augmenter sa puissance d'émission, plus il consommera de l'énergie, et donc, réduire sa durée de vie. Il convient donc de réduire en compressant ou en agrégeant les données lors de leur routage [07]. L'agrégation permet de réduire le trafic dans le réseau et d'améliorer la qualité des données reçues par l'utilisateur final.

1.3.5.10. Tolérance aux fautes

Certains nœuds peuvent générer des erreurs ou ne plus fonctionner à cause d'un manque d'énergie, un problème physique ou une interférence. La tolérance aux fautes est la capacité de maintenir les fonctionnalités du réseau sans interruptions dues à une erreur intervenue sur un ou plusieurs capteurs [38,39]. Les réseaux de capteurs étant très denses, ils entraînent une redondance de données. Cette redondance garantit le fait que la mort de certains nœuds n'affecte pas le fonctionnement du réseau et que le système peut continuer à fournir des informations d'une qualité acceptable [43].

1.3.5.11. Coût de production

Puisque les réseaux de senseurs consistent en un grand nombre de nœuds, le coût d'un seul nœud est très important pour justifier le coût global du réseau. Si le coût de production d'un réseau de capteurs est plus grand que celui du déploiement d'un réseau de capteurs traditionnel, alors, le réseau n'a pas un coût justifié. Pour cela, il est nécessaire de maintenir un coût réduit pour les capteurs.

1.3.6. Communication dans les réseaux de capteurs

Les nœuds capteurs sont usuellement dispersés dans le champ de captage. Chaque nœud déployé dans le champ de captage est capable de collecter les données et de les router vers le sink. Les nœuds doivent permettre une communication multi sauts pour les données qui circulent dans la zone de captage. Pour cela, un modèle de communication est proposé dont le rôle principal est la standardisation de la communication entre les participants afin que différents constructeurs puissent mettre au point des produits (logiciels ou matériels) compatibles. Ce modèle combine la gestion de l'énergie et le routage, intègre les données avec les protocoles réseaux et promeut une communication efficace entre les différents nœuds à travers un medium sans fil.

1.3.6.1. Pile protocolaire

La pile protocolaire utilisée par le nœud puits ainsi que tous les autres capteurs du réseau est illustrée par la figure 1.6.

Par analogie au modèle OSI des réseaux filaires, la pile protocolaire dans les réseaux de capteurs comprends 5 couches qui ont les mêmes fonctions que celles du modèle OSI. Cette pile prend en charge le problème de consommation d'énergie, intègre le traitement des données transmises dans les protocoles de routage, et facilite le travail coopératif entre les capteurs [54]. Elle est composée de la couche application, transport, réseau, liaison de données, physique, ainsi que de trois niveaux qui sont : le niveau de gestion d'énergie, de gestion de tâches et le niveau de gestion de mobilité.

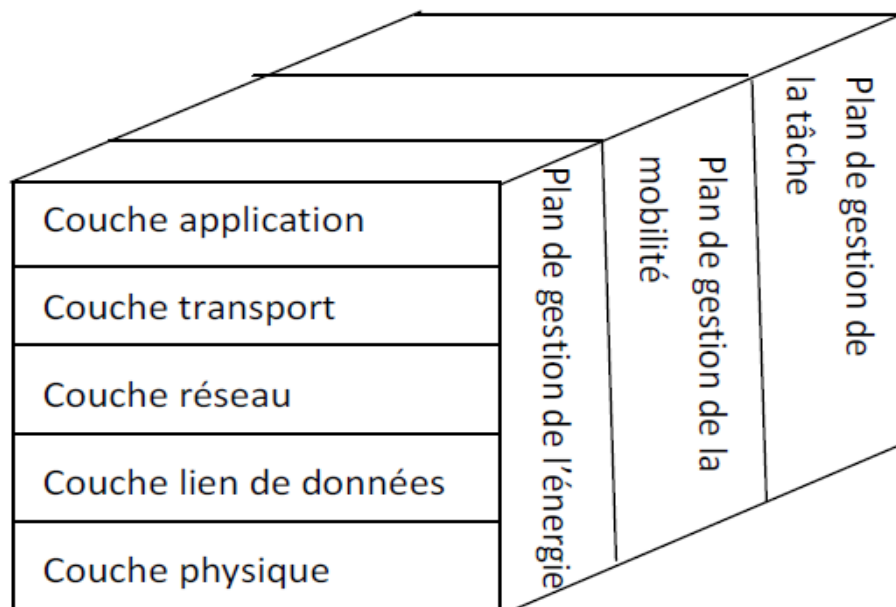


Fig.1.6 : Pile protocolaire dans les réseaux de capteurs [38]

- Couche application

Selon la tâche assignée aux capteurs, différentes applications peuvent être utilisées et implémentées sur cette couche.

La couche application doit fournir des mécanismes qui permettent à l'utilisateur d'interagir avec le réseau à travers des interfaces, éventuellement, par l'intermédiaire d'un réseau étendu (Internet à titre d'exemple), tout en restant transparent par rapport au matériel et aux mécanismes de communication dans les couches inférieures.

Quelques protocoles ont été proposés dans ce domaine parmi lesquels : SMP (Sensor Management Protocol), TADAP (Task Assignment and Data Advertisement Protocol) and SQDDP (Sensor Query and Data Dissemination Protocol) [19].

- Couche transport

Cette couche est surtout utile quand le système est conçu pour être accessible via Internet ou autre réseau externe [39]. Elle assure une liaison d'interface entre la couche application et la couche réseau. Ses principaux objectifs consistent à :

- ✓ Multiplexer et démultiplexer les messages entre les applications et la couche réseau.
- ✓ Remise fiable des données entre application.
- ✓ Réalisation de contrôle de haut niveau des données.
- ✓ Vérifie le bon acheminement des données et la qualité de transmission.

Le développement de la couche transport est un véritable challenge à cause des facteurs d'influence sur les nœuds senseurs, particulièrement les contraintes hardware telles que l'énergie et la mémoire limitées. Contrairement aux nœuds d'un réseau classique, les nœuds senseurs sont incapables de ranger de grandes quantités de données et les accusés de réception des données transmises. Il convient alors de proposer d'autres schémas pour la gestion des communications de bout en bout.

- Couche réseau

Le rôle de la couche réseau consiste essentiellement en deux fonctions qui sont l'adressage des nœuds et assurer l'acheminement des données fournies par la couche transport. Dans les RCSFs, les nœuds sont déployés d'une manière aléatoire et dense. Afin de communiquer dans ces réseaux, des protocoles de routage basés sur une communication multi-saut est la mieux adaptée car les protocoles de routages conçus pour les réseaux traditionnels deviennent inadéquats et ils ne répondent pas aux exigences des RCSFs. Pour cela, la couche réseau doit observer l'efficacité énergétique. Les communications multi-sauts ne réduisent pas seulement la consommation de l'énergie des nœuds, mais aussi la propagation du signal et sa déperdition sur le canal sans fil inhérente aux communications sans fil sur de longue-portée.

- Couche liaison de données

Elle est responsable de la détection de la trame de données, du contrôle d'accès au médium et du contrôle d'erreurs. Elle assure de manière fiable les connexions point-à-point et point-multipoint dans le réseau [12,17]. Les protocoles de la couche MAC dans les réseaux classiques ne conviennent pas aux réseaux de capteurs. La conservation d'énergie est atteinte par l'utilisation de modes d'opération de sauvegarde d'énergie en privilégiant les timeouts aux ACK là où c'est possible. Quelques protocoles MAC proposés dans la littérature:

- SMACS : Self-Organizing Medium Access Control for Sensor Networks
- EAR Algorithm : Eaves-drop And Register Algorithm
- CSMA-Based Medium Access : A Carrier Sens Multiple Access (CSMA) based MAC scheme for Sensor Networks
- Hybrid TDMA/FDMA-Based.

- Couche physique

La couche physique est responsable de la sélection de fréquence, de la génération de la fréquence de porteuse, de la détection du signal, de la modulation et du cryptage de la donnée [12,29]. Il est bien connu que les communications sans fil sur de longues portées sont coûteuses en termes de consommation d'énergie ainsi que la complexité d'implémentation. Lors de la conception de la couche physique, la minimisation de l'énergie dissipée est d'une grande importance pour un réseau de capteurs. En général, l'énergie nécessaire à la transmission d'un signal est proportionnelle à d^n , tel que $2 \leq n \leq 4$ [39,]. Une autre étape critique dans la conception de la couche physique est le choix d'un schéma de modulation de données convenable afin d'assurer la fiabilité des communications point à point dans les réseaux de capteurs.

La pile protocolaire dans les réseaux de capteurs est également caractérisée par trois niveaux de gestion :

- **Plan de gestion des tâches**

Son rôle consiste à répartir et ordonnancer les tâches de sensations dans le champ de captage, ainsi, seulement les nœuds nécessaires se verront assigner la tâche de sensation et le reste des nœuds vont se focaliser sur le routage et l'agrégation des données.

- **Plan de gestion de la mobilité**

Il détecte et enregistre le mouvement des nœuds senseurs ainsi le routage de retour à l'utilisateur est toujours maintenu et les nœuds senseurs peuvent garder la trace de leurs voisins senseurs. En connaissant leurs voisinages, les nœuds senseurs peuvent balancer leur énergie.

- **Plan de gestion de l'énergie**

Il gère la manière dont un nœud utilise son énergie. Par exemple, un nœud senseur met son émetteur/récepteur en mode Off après avoir reçu un message de son voisin. Cela permet d'éviter de recevoir des messages dupliqués. En plus, quand le niveau d'énergie d'un nœud senseur est bas, le senseur avise ses voisins par un broadcast que son niveau d'énergie est bas et donc il ne peut participer à l'opération de routage. Le reste de l'énergie est réservé à la détection.

Ces plans de gestion sont nécessaires pour que les nœuds puissent travailler ensemble pour une meilleure efficacité énergétique, pour router la donnée dans un réseau de senseurs et partager les ressources entre nœuds senseurs. Sans ces plans de gestion, chaque nœud senseur travaillera de manière individuelle. D'un point de vue d'un réseau de senseur, il est plus efficace aux nœuds de collaborer pour pouvoir prolonger la durée de vie d'un réseau.

1.3.6.2. Standards de communication dans les réseaux de capteurs

Pour assurer la communication sans fil entre les nœuds d'un réseau de capteurs, un certain nombre de modalités de communication peuvent être employées. Par exemple: la radio, la lumière diffusée, le laser...etc.

Parmi les standards les plus aptes à être exploités dans les réseaux de capteurs on retrouve : WIFI (IEEE 802.11), Bluetooth (IEEE 802.15.1 et 802.15.2), Zigbee (IEEE 802.15.4).

- **WI-FI (IEEE 802.11)**

Ce standard [78] soutenu par l'alliance WECA (Wireless Ethernet Compatibility Alliance), est très connu actuellement. Un certain nombre de normes dérivées ont été créées. La technologie WiFi (802.11 et ses variantes), bien qu'elle soit conçue pour les LANs sans fil qui se composent habituellement d'ordinateurs portables et de PDAs, est également supposée utilisable aux réseaux de capteurs. Cependant, la consommation d'énergie élevée et le débit excessif rendent les protocoles 802.11 non appropriés aux réseaux de capteurs. Ce fait a motivé les chercheurs à concevoir des protocoles MAC efficaces en énergie et spécifiques aux réseaux de capteurs.

- **Bluetooth (IEEE 802.15.1 et 802.15.2)**

La technologie Bluetooth, dont Ericsson a initié le projet en 1994 a pour but principal de remplacer les câbles sur de petites distances. Elle est utilisée dans les téléphones portables comme interface de connexion pour accéder à un PC. Malheureusement, le grand défaut de cette technologie est sa trop grande consommation d'énergie. Elle ne peut donc pas être utilisée par des capteurs qui sont alimentés par une batterie et qui, idéalement, devraient fonctionner durant plusieurs années [79]. En voici quelques caractéristiques de la technologie Bluetooth tirées de [80] :

- Bluetooth est une technologie sans fil, déposé comme standard à l'IEEE, c'est le 802.15. Elle fonctionne dans le spectre de fréquences 2,4 GHz.
- Bluetooth est destiné à un usage personnel et se classe dans la catégorie PAN (Personal Area Network).
- La technologie sans fil Bluetooth est orientée vers les applications données et voix.
- Elle peut fonctionner sur une distance de 10 ou de 100 mètres selon la classe de l'appareil Bluetooth. Le débit de données maximal avec des données rehaussées s'élève à 3 Mbits/s.
- Les ondes de la technologie sans fil Bluetooth peuvent traverser des objets massifs.
- Cette technologie est omnidirectionnelle et ne nécessite pas la visibilité directe des appareils connectés.
- Lors du développement de la spécification Bluetooth, la sécurité a toujours été et continue d'être une priorité.

- **Zigbee (IEEE 802.15.4)**

La technologie Zigbee, gérée par la Zigbee alliance et combiné avec IEEE 802.15.4, offre des caractéristiques qui répondent mieux aux besoins des réseaux de capteurs. Cette technologie qui est destinée aux environnements à faible consommation d'énergie et équipements électroniques portables permet un transfert stable de données, une installation facile, un coût réduit ainsi qu'une très basse consommation d'énergie nettement moins que Bluetooth (voir tableau 1.2) [79, 80]. Ces performances en termes de consommation d'énergie ont poussé plusieurs industriels de capteurs à l'intégrer dans leurs produits capteurs comme MicaZ, Telos, ...etc. Zigbee [81] fonctionne globalement

sur la bande de fréquences des 2,4 GHz mais également à 916 MHz en Amérique du nord et à 866 MHz bande libre en Europe. Ses principales caractéristiques sont :

- Débits compris entre 20 et 250 Kbits/s.
- Supporte différentes types de topologies réseau (étoile, mesh).
- Signal robuste et résistant aux interférences (mode d'accès CSMA/CA).
- Pénétration à travers les murs et plafonds.
- Sécurité: exige un contrôle d'accès, cryptage de données (128-bit AES, cryptage + authentification).
- permet de connecter jusqu'à 255 matériels par réseau sur une portée allant jusqu'à 100 mètres.
- Cycles d'émissions/réceptions et connexions au réseau très rapides ce qui permet une faible consommation d'énergie.

Ci-dessous un tableau comparatif entre les trois technologies de communications décrites précédemment :

Technologie	WiFi	ZigBee	Bluetooth
Norme	802.11a/b/g	802.15.4	802.15.1
Besoins mémoire	1 Mb+	4 - 32 Kb	250 Kb +
Autonomie avec pile	Heures	Années	Jours
Vitesse de transfert	11-54 et + Mb/s	250 KB/s	1 Mb/s
Portée	300 m	100 m	10-100 m

Tab.1.2 standards de communication dans les RCSFs

1.3.7. Domaines d'application

Au début de l'apparition des RCSFs, leurs domaines d'application étaient limités tout comme les types de capteurs qui existaient à cette époque. Aujourd'hui, une variété de capteurs est commercialisée. Des capteurs de température, des capteurs chimiques, sismiques, capteurs de pression, d'humidité et autre ont permis l'extension de la plage d'application utilisant les réseaux de capteurs pour atteindre tous les domaines.

1.3.7.1. Applications militaires

Ce domaine a été le moteur initial pour le développement des réseaux de capteurs sans fil. Grâce à leur faible coût et leur déploiement facile, l'usage des capteurs dans ce domaine est très efficace. Un réseau de capture peut être déployé dans un environnement hostile ou stratégique afin de surveiller les mouvements des forces ennemis. Ils permettent d'aider les unités militaires dans les champs de bataille en détectant la présence d'armes biologiques et chimiques. Une autre application importante est le ciblage qui consiste en l'incorporation des capteurs dans les systèmes de guidage des munitions intelligentes.

Le projet DSN (Distributed Sensor Network) lancé par DARPA (Defence Advanced Research Project Agency) au début des années 80, était l'un des premiers projets ayant utilisé les réseaux de capteurs pour rassembler des données distribuées et depuis, l'évolution de ces applications n'a pas cessé pour donner naissance à maints systèmes dont nous citons

l'exemple de VigilNet. Ce système est un réseau de capteurs auto organisant qui permet la détection de personnes, de véhicules et garantit une gestion équitable de l'énergie [18]. Il a été utilisé dans le but de récolter des données sur la position et les dispositions des forces ennemies [05].

1.3.7.2. Applications médicales

Les RCSFs occupent actuellement la plus grande partie des travaux de recherche dans le domaine de l'informatique et des applications destinées pour l'industrie médicale. Le but de ces applications est d'améliorer la prise en charge et la surveillance, à distance, des personnes âgées ou atteintes de maladies chroniques ainsi que les enfants [06].

Les progrès technologiques ont permis de fabriquer des senseurs avec un coût minimal dont les utilisateurs ordinaires peuvent bénéficier. Avec des capteurs économiques mais très évolués, le coût des systèmes de prise en charge médicale a baissé de manière remarquable. Les capteurs sont placés sur les objets qu'un malade utilise chez lui et les données récoltées sont envoyées vers une station de traitement pour prendre les mesures adéquates face à une situation anormale du malade. Actuellement, des micros caméras peuvent être ingérées dans le corps humain et sont capable, sans avoir recours à la chirurgie, de transmettre des images de l'intérieur de ce dernier. Un autre projet dans le domaine médical porte sur la création d'une rétine artificielle composée d'une centaine de micro capteurs pour améliorer la vision.

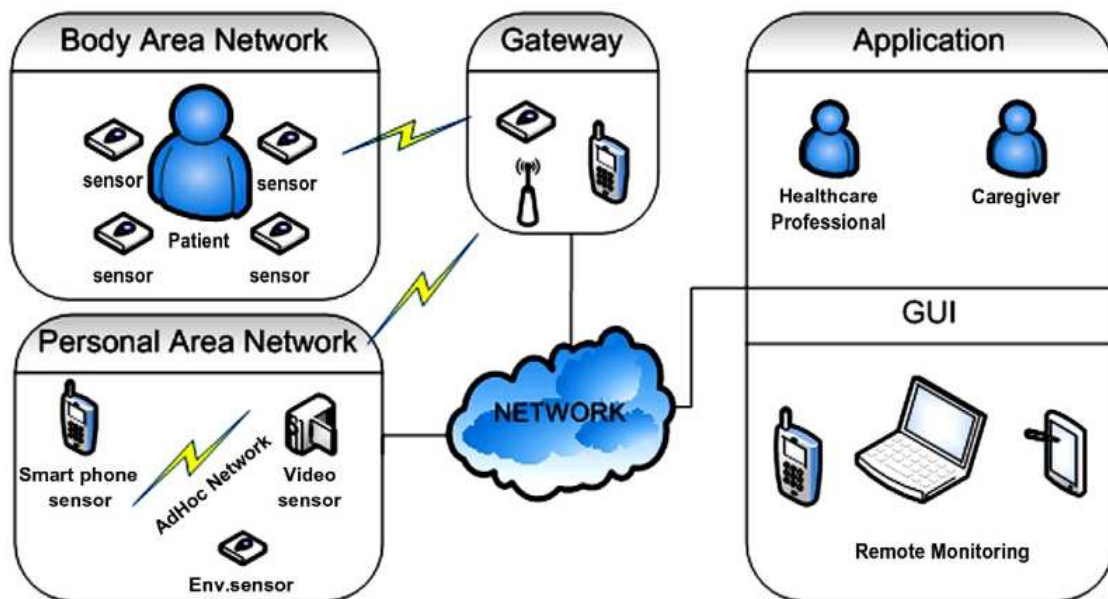


Fig.1.6: vue d'ensemble d'un RCSF destiné à une application de monitoring médical [15].

1.3.7.3. Application de surveillance de l'environnement (monitoring)

La diversité des technologies de fabrication des capteurs (capteurs thermiques, capteur de pression, ...) ont permis l'extension de leur utilisation aux applications de surveillance de l'environnement. Des capteurs chimiques, physiques, acoustique et des capteurs basés-images peuvent être utilisés pour l'étude de l'écosystème [32]. Ils peuvent être utilisés dans le but d'observer le mouvement d'insectes ou poursuivre les déplacements des animaux afin de

comprendre leur comportement ce qui peut aider à équilibrer l'écosystème. De plus, des capteurs thermiques déployés dans une forêt permettent de détecter tout changement de température et signaler un début d'incendie. Dans le domaine agricole, ils sont utilisés dans le processus de l'irrigation pour détecter les zones sèches dans le champ et réagir convenablement.

1.3.7.4. Contrôle des infrastructures critiques

Des capteurs de vibration peuvent être incorporés dans des infrastructures comme les ponts, les buildings ou autre afin de signaler toute détérioration.

1.3.7.5. Applications industrielles

Placer des capteurs dans des sites industriels comme les centrales nucléaires, les raffineries de pétrole afin de détecter les fuites de produits toxiques (gaz, produits chimiques, pétrole, radiations). Les nœuds senseurs sont aussi utilisés dans ce domaine pour le contrôle et l'automatisation des chaînes de montage [36].

1.3.7.6. Applications domotiques

Les capteurs peuvent être embarqués dans des appareils, tels que les aspirateurs, les fours à micro-ondes, les réfrigérateurs, les magnétoscopes, etc. Ces capteurs embarqués peuvent interagir entre eux et avec un réseau externe via Internet pour permettre à un utilisateur de contrôler les appareils domestiques localement ou à distance. Le déploiement des capteurs de mouvement et de température dans les futures maisons, dites intelligentes, permet d'automatiser plusieurs opérations domestiques telles que : la lumière s'éteint et la musique s'arrête quand la chambre est vide, la climatisation et le chauffage s'ajustent selon les points multiples de mesure afin de réduire la demande mondiale en énergie. Une autre application pour la sécurité des maisons est le déclenchement de l'alarme par le capteur anti-intrusion quand un étranger veut pénétrer dans la maison.

1.4. Axes de recherche actuels

Les travaux de recherche menés dans le domaine des réseaux de capteurs ont révélé plusieurs problématiques. Tous les challenges posés dans ce domaine sont dus principalement à la taille des capteurs. Autrement dit, on recherche une miniaturisation maximale et des performances optimales quant à la transmission, le débit et la consommation d'énergie qui est un facteur primordial dans ce type de réseaux. Parmi ces problématiques, nous citons :

1.4.1. Routage

Le routage dans les réseaux de capteurs constitue un défi majeur. Ceci est dû aux caractéristiques distinguant ce type de réseaux par rapport aux autres. D'abord, le déploiement d'un grand nombre de nœuds rend impossible l'établissement d'un schéma d'adressage global pour le réseau. Il n'est pas possible d'utiliser les protocoles de routage IP utilisé dans les réseaux classiques. En outre, les nœuds senseurs sont contraints en termes de capacité de traitement, de stockage et de ressources énergétiques ce qui nécessite une gestion attentive de ces ressources. Pour cette raison, il est recommandé de concevoir des protocoles de routage qui répondent aux exigences des réseaux de capteurs.

Le déploiement aléatoire des nœuds dans le champ de captage engendre des zones denses et d'autres moins denses. Les nœuds dans les zones denses peuvent générer des données redondantes dont le routage épuise l'énergie des capteurs inutilement, d'où la nécessité de filtrer ces données en utilisant des mécanismes d'agrégation lors du processus de routage qui optimise la consommation d'énergie et l'utilisation de la bande passante.

Certains protocoles de routage pour RCSFs sont centré-données, dans le sens où les données requises sont sous la forme Attribut-Valeur, seuls les nœuds qui captent des données satisfaisant les conditions appliquées à Valeurs envoient leurs données afin de diminuer le nombre de paquets circulant dans le réseau.

1.4.2. Couche MAC

La conception d'un protocole MAC pour les RCSFs doit répondre aux restrictions imposées par les caractéristiques des nœuds capteurs. Adopter un schéma d'accès au médium basé sur la contention s'avère inefficace pour les réseaux de capteurs. La reprise après des collisions est une tâche gourmande en énergie. Un protocole MAC efficace doit consommer un minimum d'énergie, éviter les collisions, être implémenté avec un minimum de code pour occuper un petit espace mémoire, minimiser les délais de délivrance des paquets et répondre aux exigences des applications des réseaux de capteurs.

1.4.3. Dissémination de données

Les protocoles de diffusion conçus pour les réseaux de capteurs doivent tenir compte de leurs spécificités ainsi que de leurs contraintes intrinsèques imposées. Souvent, dans les réseaux denses, les nœuds étant très proches produisent des données redondantes. Donc, pour la conception d'un protocole efficace pour ce type de réseaux, on doit assurer une connectivité maximale des capteurs composant le réseau, minimiser le nombre des réémissions et des réceptions redondantes et modérer la consommation d'énergie.

1.4.4. Sécurité

Certaines applications des réseaux de capteurs demandent une haute confidentialité et fiabilité des données telles les applications militaires et médicales. Pour garantir l'intégrité et la confidentialité de ces données, il est nécessaire de développer des méthodes de cryptographie qui soulève le problème de l'élaboration de protocoles de gestion de clés et qui doivent prendre en considération les limites des réseaux de capteurs [15]. Il est recommandé de protéger un réseau de capteurs des attaques externes telles le déni de service (DoS) qui peut être fatale pour les nœuds capteurs vu leurs batteries limitées [16].

1.4.5. Qualité de service

Des protocoles au niveau de la couche MAC devraient être capables d'établir des priorités entre les flux, limiter les pertes de paquets pour la gestion du réseau, ou du moins en restreindre l'impact. En outre, dans les réseaux de capteurs orientés événements où l'information devra être remontée au centre de contrôle dans un meilleur délai, il est nécessaire d'optimiser conjointement les critères latence et conservation d'énergie, car la faible latence peut avoir un impact négatif sur le nombre de communications et par la suite sur la consommation d'énergie dans ces réseaux. Pour cela, les protocoles de routage garantissant

la qualité de service doivent également respecter la contrainte énergétique. Cet axe de recherche motive davantage les chercheurs et attire de plus en plus d'attention.

1.4.6. Localisation

La localisation est le fait de déterminer la position géographique de chaque nœud dans le système. La localisation est l'un des problèmes difficiles et fondamentaux qu'il faut résoudre pour les RCSFs. La mobilité dans les RCSFs induit des changements fréquents de la topologie ce qui nécessite l'utilisation de protocole de routage basés localisation. La localisation se fait en utilisant des systèmes de positionnement globaux (GPS). Parfois l'utilisation du GPS par tous les nœuds du réseau se révèle onéreuse, alors une autre méthode basée sur la triangulation permet aux nœuds de connaître leurs positions en utilisant la puissance du signal (RSSI) reçu de quelques points dont les positions sont connues [20,22]. Cette approche est utilisée par certains protocoles de routage tel le protocole PEGASIS [21].

1.5. Conclusion

Les caractéristiques attrayantes des réseaux de capteurs telles la tolérance aux pannes, la mise à l'échelle, rapidité de déploiement et coût réduit créent, incessamment, de nouveaux domaines d'application. Ce large éventail d'applications fera des réseaux de capteurs une partie intégrante de notre vie. Toutefois, leur déploiement dans des environnements souvent hostiles rend l'intervention humaine difficile voire impensable ce qui pose le problème de disfonctionnement des nœuds dû à des dommages physiques ou à l'épuisement de leurs batteries. L'épuisement rapide des ressources énergétiques des nœuds est l'un des handicaps dont souffrent les réseaux de capteurs. Pour cela, les chercheurs consacrent une grande partie de leurs travaux aux mécanismes qui permettent de prolonger la durée de vie du réseau. Ceci peut être réalisé en minimisant l'énergie dissipée lors de la communication entre les nœuds puisque cette dernière est considérée comme étant la phase la plus gourmande en termes de consommation d'énergie. Par conséquent, des protocoles de routage approprié aux RCSFs ont été proposés.

Dans le prochain chapitre, nous allons aborder la notion de routage dans les RCSFs et exposer une taxonomie des protocoles de routage selon différents critères.

Chapitre 2 : Routage dans les réseaux de capteurs sans fil

2.1. Introduction

En général, le routage est une méthode d'acheminement des informations à la bonne destination à travers un réseau de connexion donné. Le problème de routage consiste à déterminer un acheminement optimal des paquets à travers le réseau, suivant certains critères de performances. Dans les RCSFs, l'utilisation des protocoles de routage conçus pour les réseaux Ad Hoc traditionnels est inappropriée. Ceci est en raison des caractéristiques par lesquelles se distinguent les deux types de réseaux, d'où la nécessité de les améliorer ou de développer de nouveaux protocoles de routage spécifiques aux RCSFs.

Dans ce chapitre nous commencerons par définir le routage dans les réseaux de capteurs, nous citerons quelques facteurs qui influencent sur la conception des protocoles de routage au sein des RCSF. En second lieu, nous étudierons les métriques qui permettent de tester l'efficacité du protocole une fois conçu. Par la suite, nous présenterons des critères de classification des protocoles de routage.

2.2. Routage dans les réseaux de capteurs

Le routage dans les réseaux de capteurs consiste à déterminer les routes reliant les nœuds capteurs et le nœud puits et à acheminer les données captées à travers ces routes. Les nœuds d'un réseau de capteurs sont équipés d'une interface de communication sans fil à portée limitée. La coopération des nœuds est donc nécessaire pour assurer le service de routage. Les nœuds doivent alors, en plus de leur rôle dans l'acquisition des données, jouer le rôle de routeurs, relayant de la sorte les paquets vers leur destination finale (le nœud puits). Ainsi, une donnée captée peut passer par plusieurs nœuds avant d'atteindre le nœud puits (routage multi-sauts).

Le routage doit être effectué par un algorithme distribué exécuté au niveau de tous les nœuds du réseau ou centralisé où l'exécution se fait au niveau d'un nœud ayant une vue globale sur le réseau. L'établissement des routes peut être réalisé soit de manière proactive, où les chemins sont construits au préalable, soit de manière réactive où les routes sont établies à la demande. Le choix de la route reliant une source de données au nœud puits se fait tout en optimisant une certaine métrique (consommation d'énergie, délai de transmission, nombre de sauts,...).

Plusieurs contraintes rendent le routage difficile. Par exemple, un protocole de routage doit maximiser la durée de vie du réseau, ce qui représente une contrainte primordiale. D'autres facteurs doivent être pris en compte et seront abordés dans la section suivante [62].

2.3. Facteurs influant sur la conception d'un protocole de routage

La conception des protocoles de routage dans les réseaux de capteurs nécessite la prise en compte de plusieurs facteurs. Ces derniers sont d'une grande importance, car ils servent de directives guidant les concepteurs pour qu'une communication efficace puisse être assurée. En outre, ils peuvent servir de métriques de comparaison entre les différents protocoles de routage proposés pour les réseaux de capteurs. Dans ce qui suit, nous exposerons les principaux facteurs qui influencent la conception des protocoles de routages dans les réseaux de capteurs :

2.3.1. Mobilité du réseau

Hormis les très peu installations qui utilisent des capteurs mobiles, la plupart des architectures de réseau supposent que les nœuds capteurs sont stationnaires et d'autre part supporte la mobilité du sink ou du cluster-head. L'évènement à capturer peut, aussi, être dynamique ou statique. Pour une application à événements statiques, le réseau travaille en mode réactif. Par contre, une application à événements dynamiques nécessite des signalisations périodiques [63].

2.3.2. Déploiement des nœuds

Le déploiement des nœuds est une considération importante. Les réseaux de capteurs peuvent être installés d'une manière déterministe ou auto-organisée. Quand le déploiement du réseau est déterministe, les nœuds sont placés manuellement d'une manière prédéterminée. Le routage pourrait suivre les chemins prédéterminés. Quand le déploiement du réseau est auto-organisé, les nœuds sont dispersés aléatoirement dans la région du phénomène à surveiller. Un autre aspect de déploiement consiste à déterminer la nature des nœuds puits ainsi que les nœuds têtes des cellules "cluster-heads" en choisissant des nœuds puissants pour assurer ces rôles [63].

2.3.3. Consommation d'énergie

Les nœuds capteurs peuvent utiliser leur approvisionnement en énergie pour calculer et transmettre l'information dans un environnement sans fil. Pour cela, les techniques de conservation d'énergie lors de la communication et le calcul sont essentielles. En effet, la durée de vie d'un nœud capteur a une forte dépendance de la durée de vie de la batterie. Dans un réseau de capteurs multi-sauts, chaque nœud joue un rôle dual comme un expéditeur et un routeur de données. Le mal-fonctionnement de quelques nœuds capteurs dû à la défaillance (à cause de la diminution totale d'énergie) peut causer les changements topologiques cruciaux [20].

2.3.4. Modèle de livraison de données

Le modèle de renvoi des données captées constitue un autre facteur important qui affecte les performances du protocole de routage utilisé. Le modèle de délivrance des données peut être classé dans l'une de ces catégories : continu (time-driven), événementiel (event-driven), orienté requête (query-driven) et hybride. Le modèle de la livraison continu convient aux applications qui nécessitent des données capturées périodiquement. Dans les modèles événementiels et orientés requêtes, les nœuds capteurs réagissent immédiatement et soudainement lors d'un changement rigoureux de valeur d'un attribut senti dû à l'occurrence de certains événements ou requêtes produites par la station de base. Ces modèles sont bien appropriés aux applications ayant des exigences temps réel. Une combinaison des modèles précédents est également possible. Le protocole de routage est fortement influencé par les modèles de données rapportées concernant la consommation d'énergie et la stabilité des routes.

2.3.5. Hétérogénéité Nœud/Lien

Plusieurs études ont supposé qu'un réseau de capteurs est constitué de nœuds homogènes ayant les mêmes capacités en termes de calcul, de transmission et d'énergie disponible. Cependant, selon l'application, un nœud capteur peut avoir des rôles délégués. L'existence d'un ensemble de capteurs hétérogènes soulève beaucoup de questions techniques liées au routage de données. Par exemple, quelques applications pourraient exiger divers types de capteurs pour surveiller la température, la pression et l'humidité de l'environnement et capturer l'image ou le cheminement visuel des objets mobiles. Ces capteurs spéciaux peuvent être déployés comme ils peuvent inclure les différentes fonctionnalités (un nœud peut être équipé de plusieurs unités de sensation différentes). Même le captage et la délivrance des données peuvent être produits par ces capteurs à différents taux. Par exemple, les protocoles hiérarchiques indiquent des nœuds cluster-heads qui sont différents des nœuds capteurs normaux. Ces cluster-heads peuvent être plus puissants que les autres nœuds capteurs en terme d'énergie, de bande passante et de mémoire. Par conséquent, ils sont chargés de la transmission des données à la station de base.

2.3.6. Tolérance aux fautes

Quelques nœuds capteurs peuvent être en panne ou être bloqués à cause du manque d'énergie, de dommage physique ou d'interférence environnementale. La défaillance des nœuds capteurs ne devrait pas affecter la tâche globale du réseau de capteurs. Si plusieurs nœuds sont en panne, les protocoles de routage doivent s'adapter à la formation de nouveaux liens et router les données collectées à la station de base. Ceci peut exiger l'activation d'ajustement de puissance et de vitesse de transmission sur les liens existants pour réduire la consommation d'énergie, ou re-router des paquets à travers des régions du réseau où plus d'énergie sera disponible.

2.3.7. Scalabilité

Les applications des RCSF nécessitent en général un déploiement dense des nœuds. Les protocoles de routage doivent donc être très scalables. Autrement dit, les protocoles de routage ne devraient pas souffrir d'une dégradation de performance dans le cas d'endommagement des nœuds aussi bien qu'avec un nombre plus élevé de nœuds [49].

2.3.8. Connectivité

Le nombre important de nœuds dans un RCSF, fait qu'ils sont généralement dispersés de façon aléatoire, et ne sont pas uniformément répartis sur le champ de captage. Ce qui implique que certaines régions du champ de déploiement puissent bénéficier d'une meilleure connectivité. Par conséquent, les protocoles de routage conçus pour les RCSF doivent avoir une capacité d'auto-organisation qui les adapte à la distribution aléatoire des nœuds et à la topologie dynamique du réseau [49].

2.4. Métriques de routage

Cette section étudie les métriques communes utilisées pour mesurer l'efficacité des protocoles de routage. Un calcul de métrique est un algorithme qui traite un coût associé à un chemin de routage. Les protocoles de routage permettent aux nœuds de comparer les

métriques calculées afin de déterminer les routes optimales à emprunter. Plus la métrique est optimale, plus le protocole de routage considère que la probabilité d'atteindre le nœud puits à travers ce nœud intermédiaire est grande [65].

Plusieurs métriques peuvent affecter le routage en termes d'énergie, délai, longueur du chemin, etc. De plus, elles peuvent être considérées seules ou combinées (hybrides).

2.4.1. Métriques pour la consommation énergétique

Les protocoles de routage utilisent cet ensemble de métriques pour minimiser la consommation d'énergie pendant le routage [54,66]. L'idée est de calculer l'énergie disponible pour chaque nœud du réseau et l'énergie nécessaire pour les transmissions des paquets entre une paire de nœuds.

Les routes entre les nœuds et le puits sont établies et chacune d'elles est caractérisée par la somme des énergies disponibles des nœuds qui la constituent et par la somme des énergies nécessaires des liaisons qui la construisent. La consommation d'énergie suit plusieurs approches dont nous pouvons citer :

2.4.1.1. Considération de puissance

La route choisie est celle caractérisée par la somme la plus élevée des énergies disponibles des nœuds.

2.4.1.2. Considération du coût

La route choisie est celle caractérisée par la plus petite somme des énergies nécessaires pour les transmissions.

2.4.1.3. Considération de puissance et du coût

Cette métrique est la combinaison des deux métriques précédentes. La route choisie est celle caractérisée par la plus petite somme des énergies nécessaires des liaisons et la plus grande somme des énergies disponibles des nœuds.

2.4.2. Route à nombre de sauts minimum

La route sélectionnée est celle qui traverse un nombre minimum de nœuds intermédiaires pour atteindre le nœud puits. Cette métrique est utilisée par les protocoles de routage pour minimiser le nombre de sauts pendant le routage.

2.4.3. Perte de paquets

Les protocoles de routage utilisent cette métrique dans le but de minimiser le nombre de paquets de données perdus lors du transfert depuis une source vers une destination [62].

L'idée est de calculer le ratio des paquets perdus et des paquets émis transitant dans le réseau. Autrement dit, on calcule le nombre de paquets perdus sur le nombre de paquets transmis lors d'une transmission. Dans le cas où le taux de perte de paquets est élevé, il est nécessaire de mettre en place des mécanismes qui permettent de minimiser les collisions.

2.5. Taxonomie des protocoles de routage

Récemment, les protocoles de routage conçus pour les RCSF ont été largement étudiés. Les méthodes employées peuvent être classifiées suivant plusieurs critères qui sont illustrés dans le tableau suivant :

Type de classification	Classes
Paradigme de communication	Centré-nœuds
	Centré-données
	Basé-localisation
Topologie de réseaux	Plate
	Hiérarchique
Livraison de données	Event driven
	Query driven
	Continu
Méthode d'établissement de routes	Protocoles proactifs
	Protocoles réactifs
	Protocoles Hybrides
Mode de fonctionnement du protocole	Routage multi-chemins
	Routage basé sur les requêtes
	Routage basé sur la négociation
	Routage basé sur la qualité de service

Tab.2.1 : taxonomie des protocoles de routage.

2.5.1. Selon le paradigme de communication

Le paradigme de communication détermine la manière dont les nœuds sont interrogés. On distingue trois paradigmes

2.5.1.1. Centré nœuds

Ce modèle est utilisé dans les réseaux où il est important de connaître les nœuds communicants. Cependant, ce paradigme ne reflète pas la vision des RCSF quant à leurs applications où la donnée transmise est plus importante que l'émetteur. Néanmoins, le paradigme centré nœuds n'est pas totalement écarté, car certaines applications nécessitent une interrogation individuelle des nœuds [66].

2.5.1.2. Centré-données

Le but principal dans ce routage est de minimiser les redondances de données pouvant devenir très pénalisantes surtout en termes de consommation d'énergie et surcharge du réseau.

Dans un tel routage, la station de base diffuse des requêtes vers des régions particulières ou vers tout le réseau et attend, ensuite, les données correspondant à ses requêtes depuis les nœuds situés dans la région cible ou d'intérêt [72,68].

2.5.1.3. Basé localisation

Ce paradigme est utilisé dans les applications où il est plus intéressant d'interroger le système en se basant sur la localisation des nœuds, et où on peut tirer profit des positions des nœuds pour prendre des décisions qui minimisent le nombre de messages transmis pendant le routage. Avant d'envoyer ses données à un nœud destination, le nœud source utilise un mécanisme pour déterminer sa localisation. Il est donc nécessaire de se pencher sur une solution de localisation géographique dont le degré de précision dépend de l'application visée [68].

2.5.2. Selon la structure du réseau

La topologie détermine l'organisation logique adoptée par les protocoles de routage afin d'exécuter les différentes opérations de découverte de routes et de transmission de données. Elle joue un rôle significatif dans le fonctionnement d'un protocole. La topologie peut être hiérarchique ou plate [20].

2.5.2.1. Protocole plat

La première catégorie des protocoles de routage est celle des protocoles de routage plats multi-saut (*multihop flat routing*), dont les protocoles sont basés sur le principe "orienté donnée" *data-centric*, où tous les nœuds ont le même rôle et ils collaborent entre eux pour accomplir la tâche de sensation. En raison du grand nombre de nœuds dans un RCSF, il n'est pas faisable d'affecter un identificateur global à chaque nœud. Pour pallier à cette lacune, il est usuel de propager des requêtes à tous les nœuds d'une région spécifique et attendre la réception des données des senseurs concernés dans cette région. Puisque, ces données sont demandées par des requêtes, un système de dénomination par attribut est nécessaire pour spécifier les propriétés des données [20].

Les topologies plates sont caractérisées par la simplicité des algorithmes exécutés par les protocoles de routage. Comme les RCSF souffrent des changements brusques de la topologie, une organisation plate permet la possibilité de construire différents chemins des nœuds sources vers puits. Cependant, les réseaux plats présentent des inconvénients comme celui défini par le problème de Hotspot. En effet, tous les nœuds sont homogènes et il n'y a que le nœud puits qui est chargé de la récolte d'informations, et, ces dernières passent forcément par les nœuds qui entourent le nœud puits et qui seront de ce fait épuisés. Par ailleurs, les nœuds doivent accomplir plusieurs tâches en même temps ce qui pourrait rapidement épuiser leurs ressources énergétiques et dégrader ainsi les performances du réseau [69].

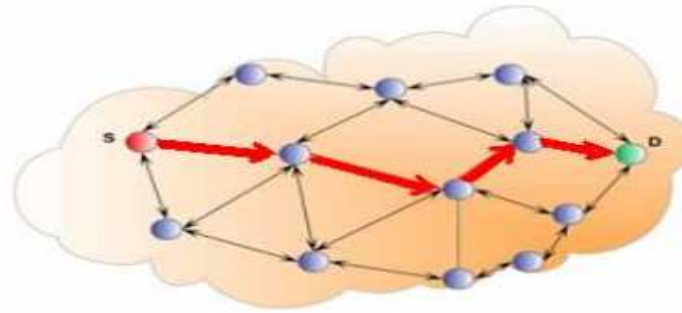


Fig.2.1 : Topologie plate

2.5.2.2. Protocole hiérarchique

Ce type de routage procède à un groupement logique des nœuds du réseau (clustering) qui consiste à élire un chef de groupe puis à déterminer les nœuds qui feront partie du groupe. Dans chaque groupe (cluster), les nœuds envoient leurs données au chef du groupe (voir figure 2.3) qui envoie, à son tour, la donnée au nœud puits (par l'intermédiaire des autres chefs de groupe, si nécessaire). Les chefs de groupe sont parfois supposés avoir un niveau d'énergie supérieur et une plus grande capacité de stockage et de calcul. Ils sont responsables du traitement, de l'agrégation et de la transmission des données. Alors que les nœuds plus contraints en ressources énergétiques se consacrent uniquement au captage. De cette façon, le mécanisme de clustering peut contribuer d'une façon considérable à l'économie d'énergie, et à une meilleure scalabilité du système. En effet, le routage hiérarchique facilite l'agrégation des données, ce qui conduit à diminuer les redondances et les transmissions inutiles vers le nœud puits. D'autre part, le clustering permet aux nœuds d'effectuer des transmissions sur de courtes distances avec leurs chefs de groupes, ce qui permet de minimiser la consommation d'énergie [62].

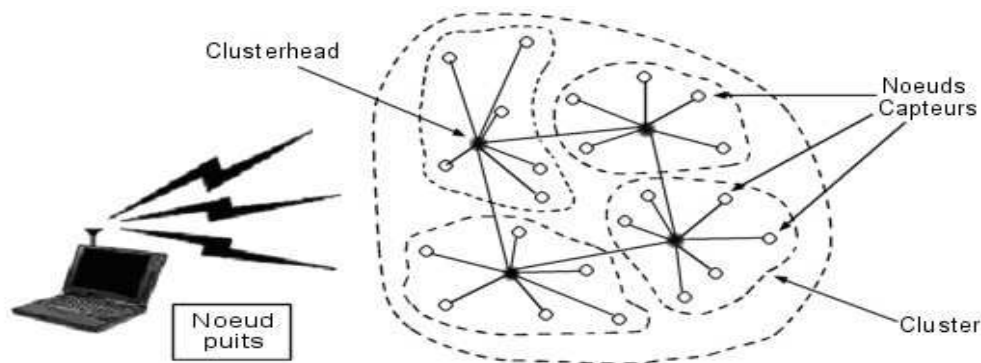


Fig.2.2 : Topologie hiérarchique

2.5.3. Selon le modèle de livraison de données

Il est possible de distinguer trois modèles de livraison de données : event driven, query driven, et le modèle continu.

2.5.3.1. Event driven

La génération et la transmission des paquets de données sont commandé par l'apparition d'un événement dans le réseau. La plupart des applications event-driven sont des applications intolérantes aux délais (temps réel), critiques et interactives [70]. Au lieu d'avoir un nœud émetteur et un autre récepteur de l'information, on trouve un nœud récepteur (le nœud de contrôle sink) et un groupe de nœuds capteurs, se trouvant proches de l'événement, qui sont tous des émetteurs de la même information. Pour ce modèle, la réussite de ces applications repose essentiellement sur la détection de l'événement et la rapidité des prises des réactions nécessaires pour assurer l'aspect temps réel des applications.

L'inconvénient majeur de ce modèle est la redondance des données. En effet, les nœuds excités par le même événement envoient la même information au nœud de contrôle (sink). Pour cela, un protocole de routage basé sur la négociation des données est recommandé.

2.5.3.2. Query driven

Le modèle query driven est semblable au modèle event-driven sauf que la collecte des informations sur l'état de l'environnement est initiée par des interrogations envoyées par le nœud sink tandis que dans le modèle event-driven, elle est déclenchée suite à un événement détecté. La plupart des applications query-driven sont des applications interactives, critiques, de non bout en bout et leur tolérance aux délais dépend de l'urgence de l'interrogation [71]. Notons que le modèle query-driven peut être utilisé pour contrôler et reconfigurer les nœuds. Par exemple, le sink peut envoyer des commandes au lieu des interrogations pour modifier le programme d'un nœud capteur, modifier son taux de trafic ou son rôle. Seul le nœud capteur jouant le rôle de sink est autorisé à émettre des demandes d'interrogations ou des commandes.

2.5.3.3. Le modèle continu

Dans le modèle continu, les nœuds capteurs envoient les informations d'une manière continue au sink suivant un volume de trafic prédéterminé.

Le processus d'acheminement des données à travers les nœuds du réseau peut prendre quatre (04) formes distinctes ; dans les architectures plates, les capteurs peuvent communiquer directement avec le puits en utilisant une forte puissance de transmission (figure.a) ou bien via un mode multi-sauts en utilisant une puissance faible (figure.b). Alors que dans les architectures hiérarchiques, les capteurs se structurent en groupes dits clusters où seuls les cluster-heads peuvent communiquer avec le puits directement (figure.c) ou bien en mode multi-sauts (figure.d); la communication intra-cluster, quant à elle, peut être en un seul-saut ou en multi-sauts.

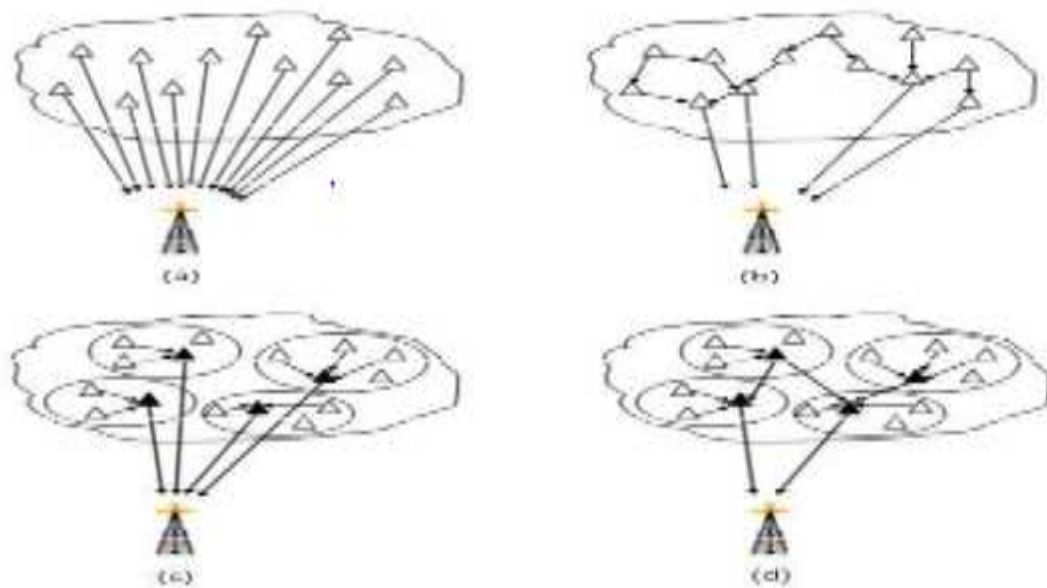


Fig.2.3 : acheminement des données dans un RSCF

2.5.4. Selon le mode d'établissement des chemins

Suivant la manière de création et de maintenance des routes lors de l'acheminement des données, les protocoles de routage peuvent être séparés en trois catégories : protocoles proactifs, réactifs ou hybrides [71].

2.5.4.1. Protocole proactif

Les protocoles proactifs sont utilisés principalement dans les réseaux de grande taille. Ils maintiennent en temps réel une vision de l'état du réseau suffisante pour que tous les sous-ensembles (nœuds) soient connectés à chaque instant, éventuellement, en passant par plusieurs liens consécutifs. Pour cela, chaque nœud relais envoie périodiquement à tous ses voisins, sa table de routage contenant l'état de tous ses liens. Les sous-ensembles du réseau possèdent alors assez d'informations pour trouver le meilleur chemin vers tout autre élément dans le réseau. Le trafic de contrôle ne contient que l'information sur les liens appartenant à ce nœud, au lieu de toute l'information sur tous les liens. Cette méthode constitue un moyen efficace pour économiser de l'énergie.

2.5.4.2. Protocole réactif

Contrairement aux protocoles proactifs, les tables de routages des nœuds ne sont plus envoyées périodiquement, mais seulement à la demande, lorsque du trafic utilisateur doit être acheminé vers une destination vers laquelle un chemin n'est pas connu à ce moment-là. Dans ce cas, une requête pour établir un tel chemin est diffusée dans tout le réseau. La destination (ou un nœud connaissant un chemin vers la destination) recevra alors cette requête diffusée, et pourra y répondre pour établir un chemin, envoyant une réponse prenant le chemin depuis la source de la requête. La connaissance de ce chemin sera maintenue dans le réseau tant que le trafic utilisateur l'empruntera, puis disparaîtra une fois les informations transmises.

Ce type de méthode réduit la quantité d'informations de routage à transmettre, mais les méthodes réactives sont susceptibles de ne pas gérer correctement un trafic utilisateur empruntant des routes trop différentes et trop changeantes. En effet, le fait que l'existence de certains liens ne soit pas connue peut créer des situations dans lesquelles les chemins les plus courts ne sont pas connus, et donc pas utilisés.

2.5.4.3. Protocole hybride

Ils exploitent les avantages des deux modes évoqués ci-dessus. Ils fonctionnent selon le mode proactif pour garder la connaissance locale de la topologie et le mode réactif pour les nœuds lointains. Nous avons donc un routage à deux niveaux avec l'utilisation de zones qui permet d'optimiser la diffusion des requêtes de demande de route [71].

2.5.5. Selon le mode de fonctionnement du protocole

Selon le fonctionnement et les avantages visés par les protocoles, quatre classes de routage peuvent être distinguées : routage basé sur la qualité de service (QoS pour Quality Of Service), routage multi-chemins, routage basé sur les requêtes et routage basé sur la négociation de données. Dans cette section, nous allons présenter les principales caractéristiques de ces différentes classes.

2.5.5.1. Routage multi-chemins

Le routage multi-chemins est utilisé pour pallier au problème de l'utilisation d'un seul chemin pour toutes les communications. Les nœuds constituant ce chemin épuisent rapidement leurs ressources et provoquent la déconnexion de certaines zones du réseau. Cette technique de routage garantit l'existence de plusieurs chemins fiables pour la transmission des données entre les capteurs et le sink quand le premier chemin est défaillant. Le routage multi-chemins augmente la tolérance aux fautes en fournissant des chemins alternatifs ce qui offre une reprise rapide du transfert de données suite à panne. Un exemple de protocole de routage multi-chemins, nous citons EAR (Energy Aware Routing) que nous allons étaler en détail dans le chapitre 4.

2.5.5.2. Routage basé sur les requêtes

Dans ce type de routage, le nœud puits génère une requête de données (portant sur une donnée particulière) propagée à travers le réseau afin d'interroger les nœuds capteurs. Le nœud qui détient les données requises doit répondre au nœud demandeur, en envoyant la donnée à travers le chemin inverse de la requête. Le protocole Directed Diffusion est l'un protocole basé sur les requêtes.

2.5.5.3. Routage basé sur la négociation

Cette famille de protocoles nommée Sensor Protocols for Information via Negotiation a été conçue afin de pallier aux inconvénients de l'inondation du réseau et adapter les protocoles de routage aux ressources des nœuds capteurs [38]. Au lieu de diffuser toutes les données, les protocoles utilisent des descripteurs de ces dernières. Le descripteur peut être le type de la donnée où la localisation de son origine. Un nœud ayant reçu ou généré une donnée envoie un message d'annonce ADV incluant le descripteur. Les nœuds qui reçoivent

ce message et qui sont intéressés par la donnée répondent par une requête REQ. Finalement, les données sont envoyées vers les nœuds intéressés avec un message DATA [51]. Ce mode garantit que seules les données utiles sont transmises et élimine la redondance de ces dernières.

2.5.5.4. Routage basé sur la qualité de service

Quelques applications des réseaux de capteurs peuvent avoir des exigences temps-réel, où les données captées doivent être délivrées à l'utilisateur dans un laps de temps limité, autrement elles deviendraient inutiles. Par exemple, dans une application de surveillance d'un local, un paquet de données envoyé par un nœud, signalant une intrusion, doit être acheminé au plus bref délai vers le nœud puits.

Les protocoles de routage basés-QoS (Quality of Service) dans les réseaux de capteurs essaient de répondre à quelques exigences de qualité de service tel le délai de transmission ou niveau de fiabilité) pendant le routage des données. Ceci, tout en maintenant un équilibre entre la qualité de service du routage et sa consommation énergétique [62].

2.6. Conclusion

Le routage dans les réseaux de capteurs est un problème complexe car nous devons assurer la fiabilité de livraison de données tout en consommant le minimum d'énergie. En outre, les protocoles de routage conçus pour les réseaux ad-hoc ne sont pas recommandés pour les RCSFs car ces derniers sont composés de nœuds à ressources limitées. Plusieurs protocoles de routage ont été proposés pour les réseaux de capteurs. Ils peuvent être classés en plusieurs catégories selon le mode d'établissement des chemins, l'architecture du réseau, le mode de fonctionnement du protocole et le paradigme de communication.

Nous avons également vu précédemment que la topologie plate du protocole de routage fournit de bonnes performances pour un réseau d'une taille petite ou moyenne. Cependant, lorsque la taille du réseau augmente, ses performances se dégradent vu le trafic de données qui augmente entre les nœuds. Afin de pallier à cet inconvénient, les chercheurs ont proposé l'architecture hiérarchique qui vise à augmenter la scalabilité du réseau et optimiser l'utilisation des ressources énergétiques. Parmi les méthodes de hiérarchisation, nous allons traiter le clustering qui se révèle une bonne méthode de routage qui permet la conservation de l'énergie dans les réseaux de capteurs.

Dans le chapitre suivant, nous allons décrire l'approche de clustering et exposer quelques protocoles de routage basé sur cette approche et discuter leur apport pour les RCSFs.

Chapitre 3: Etat de l'art sur les protocoles de routage hiérarchiques dans les RCSFs

3.1. Introduction

Les réseaux de capteurs sont fortement contraints en termes de ressources énergétiques. Actuellement, la plus grande partie des travaux de recherches se focalise sur le prolongement de la durée de vie du réseau en optimisant la consommation de l'énergie des capteurs. Le clustering a été proposé comme une approche de routage qui contribue, non seulement à la réduction de la consommation de l'énergie, mais aussi à diminuer le nombre de messages dans le réseau ainsi que l'optimisation de l'utilisation de la bande passante.

Ce chapitre est organisé comme suit. En premier lieu, nous définirons l'ensemble des concepts relatifs au mécanisme de clustering. En second lieu, nous présenterons le processus de clustering et nous passons en revue les principaux avantages d'un protocole de routage hiérarchique basé sur le mécanisme de clustering. Par la suite, nous énumérons quelques unes des caractéristiques d'un protocole de routage hiérarchique. Nous exposons, également un ensemble de solutions hiérarchiques proposées dans la littérature tout en expliquant leurs caractéristiques et les étapes de leurs constructions.

3.2. Notions de base

La manière de déployer les nœuds capteurs en réseau affecte profondément l'application en question. Le déploiement du réseau pour une application donnée peut être fait de manière aléatoire et non uniforme ou manuellement dans le champ de captage [20]. Dans le cas d'une topologie non uniforme, le clustering représente une méthode efficace qui permet de mieux gérer le trafic dans le réseau et diminuer le nombre de messages échangés entre les nœuds.

3.2.1. Définition

Généralement, les RCSFs sont déployés de manière très dense dans des environnements difficiles où l'organisation manuelle du réseau n'est pas envisageable d'où la nécessité de trouver un schéma d'organisation logique du réseau. Parmi les approches proposées à cet effet, nous citons le clustering. Cette méthode a été présentée comme un moyen performant pour augmenter la mise à l'échelle du réseau et améliorer sa durée de vie [45]. Les algorithmes de clustering divisent le réseau en groupes de nœuds nommés clusters afin de relayer les données captées de manière efficace vers la station de base [26, 45, 48, 49]. Chaque cluster est représenté par un chef de cluster ou cluster-head. Chaque nœud dans un cluster envoie ses données vers le cluster-head. Le rôle de ce dernier consiste à relayer les données reçues à partir des membres de son cluster directement vers la station de base ou à travers des CHs intermédiaires. Le rôle d'un CH ne consiste pas seulement à relayer les données, mais aussi, à effectuer certains traitements sur elles tels l'agrégation des paquets reçus à partir des nœuds du cluster. Cette opération permet de réduire considérablement le nombre de paquets transmis vers la station de base et diminue aussi les interactions entre les CHs. Ce mécanisme permet d'économiser l'énergie des nœuds et prolonger la durée de vie du réseau puisque les nœuds effectuent des transmissions sur de courtes distances vers leur CH et que le nombre de paquets redondants est abaissé ce qui optimise l'utilisation du medium de transmission.

3.2.1.1. Nœud membre

Le nœud capteur est le composant noyau d'un RCSF. Les nœuds senseurs peuvent prendre différents rôles dans un réseau, tel que la sensation, le stockage des données, le routage ou le traitement des données. Les tâches d'un capteur dépendent du rôle qui lui est affecté par le protocole de clustering.

3.2.1.2. Cluster

Le cluster constitue l'unité organisationnelle pour les RCSFs. Le déploiement très dense de ces réseaux nécessite leur division en groupes de nœuds afin de faciliter les tâches de communication et répondre aux différentes contraintes.

3.2.1.3. Cluster head

Les cluster-heads (CH) sont les représentants des clusters. Ils sont responsables de la coordination entre les nœuds à l'intérieur de son cluster et l'agrégation de leurs données ainsi que les communications avec d'autres cluster-heads [44]. Chaque cluster est guidé par un seul CH ou deux, où le deuxième est prévu pour remplacer le CH principale en cas de panne.

3.2.1.4. Station de base

La station de base est dotée de ressources énergétiques, de capacité de traitement et de stockages très importantes. Elle représente le premier niveau dans la hiérarchie d'un RCSF. Elle fournit une liaison entre le réseau de senseurs et l'utilisateur final. Les données sont traitées et sauvegardées au niveau de la SB selon le type de l'application et les besoins de l'utilisateur final.

3.2.1.5. Utilisateur final

Les données générées par un réseau de capteurs sont utilisées par une large gamme d'applications [39]. Ces données sont accessibles via internet en utilisant un PDA ou un ordinateur bureau. L'utilisateur envoie des requêtes qui seront diffusées dans le réseau afin de recueillir les données d'intérêt.

3.2.2. Processus de clustering

L'efficacité d'un algorithme de clustering est évaluée en termes du nombre de clusters formés et de leurs stabilités en fonction des changements de la topologie du réseau. Il existe plusieurs méthodes de formation de cluster mais elles sont, communément, exécutées en trois phases :

3.2.2.1. Phase d'élection des cluster-heads

La première étape d'un algorithme de clustering consiste en l'élection des CHs. le CH joue le rôle d'un coordinateur local ou sink dans son cluster. Il supporte plusieurs tâches de coordination entre les nœuds membres de son cluster, la collection des données à l'intérieur de son cluster, l'agrégation et la transmission des données agrégées vers le sink. La division des nœuds du réseau en clusters dépend du nombre de CH désiré et leurs localisations. En conséquence, la sélection du CH joue un rôle important dans le reste du processus de clustering. Cette phase détermine les performances, la durée de vie et l'efficacité énergétique du réseau [47]. Les CH sont élus selon une métrique spécifique telle, le plus grand/petit ID

dans son voisinage, le degré de connectivité, la puissance de transmission, l'énergie résiduelle,...etc. ou selon un poids qui représente une combinaison de celle-ci.

Plusieurs travaux de recherches se sont focalisés sur l'optimisation de la sélection des CHs.

3.2.2.2. Communication intra-cluster et inter-cluster

Chaque CH est chargé des communications à l'intérieur des son cluster et de maintenir les informations de routage lui permettant de joindre la station de base. Pour atteindre la station de base, les cluster-heads communiquent directement avec cette dernière sinon une communication multi-saut est adoptée en utilisant d'autres cluster-heads comme nœud relais.

3.2.2.3. Maintenance des clusters

Dans le but de s'adapter aux changements de la topologie du réseau, une mise à jour des clusters est dynamiquement réalisée pour reconnaître les nouveaux nœuds ajoutés ou ceux qui ont disparus (les nœuds qui ont épuisé leurs énergies). D'autre part, un CH consomme beaucoup plus d'énergie comparé aux nœuds ordinaires, du moment qu'il assure des tâches plus complexes. Si le CH garde son statut durant toute la vie du réseau, même s'il ne possède pas par exemple le poids maximum dans son propre cluster alors, il perdra son rôle une fois sa batterie sera épuisée et ceci conduira à la déconnexion de son cluster du reste du réseau [73]. Dans le but d'éviter de telles situations, le processus de clustering est généralement exécuté en round pour alterner le rôle de CH entre les nœuds et balancer la consommation de l'énergie dans le réseau.

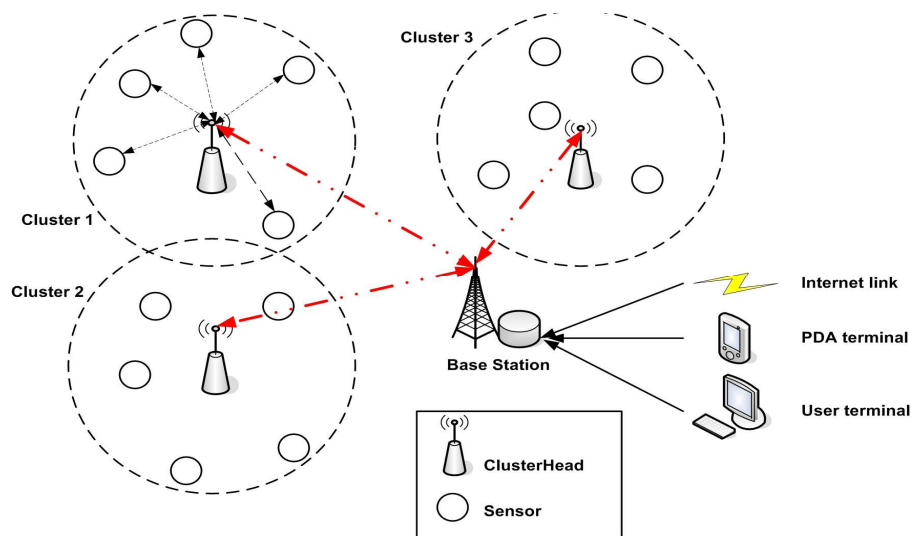


Fig 3.1 : Structure hiérarchique d'un réseau de capteurs [46]

3.2.3. Avantages de l'approche de clustering

Récemment, un grand nombre d'algorithmes de clustering ont été proposés dont l'objectif majeur est d'optimiser l'utilisation des ressources du réseau de capteurs. L'approche de clustering permet de [45,46]:

- Localiser les routes établies à l'intérieur des clusters et ainsi, réduire la taille des tables de routage au niveau de chaque nœud.
- Permet d'augmenter considérablement la scalabilité du réseau sans dégrader ses performances.
- En limitant la portée des communications inter-cluster et le nombre de messages redondant échangés entre les nœuds, le clustering optimise l'utilisation de la bande passante.
- Un CH peut ordonnancer les tâches des nœuds à l'intérieur de son cluster, ainsi, les nœuds peuvent passer en mode sleep durant les périodes d'inactivité ce qui permet de réduire la consommation de l'énergie. Les transmissions des nœuds vers le CH peuvent aussi être ordonnancées afin d'éviter les collisions sur le canal de transmission.
- En outre, les CH agrègent les données captées par les nœuds membres de son cluster ce qui permet de diminuer le nombre de paquets relayés vers la station de base.
- Stabiliser la topologie du réseau au niveau des nœuds de manière à ce que chaque nœud se soucie seulement du cluster à joindre et ne sera pas affecté par les changements au niveau supérieur. Ceci réduit la charge de la maintenance de la topologie.

3.3. Objectifs du clustering

Les objectifs des algorithmes de clustering varient selon les besoins des applications. Par exemple, si une application est sensible à la latence des données, à la durée de vie des nœuds ou autre, la connectivité intra cluster et inter cluster ainsi que la longueur des chemins sont des facteurs à considérer lors de l'élection des CHs et la formation des clusters []. Nous allons exposer, dans cette section, les objectifs souvent visés par le processus de clustering.

3.3.1. Equilibrage des charges

La répartition équilibrée des capteurs sur les clusters est généralement un objectif pour les configurations où les CHs sont sélectionnés par ces capteurs, afin d'effectuer des fonctions de traitement de données et de gestion intra-clusters. Cependant, la mise en clusters à taille égale devient cruciale pour l'allongement de la durée de vie du réseau. Ceci empêche l'épuisement de l'énergie d'un sous-ensemble de CHs à taux élevés de nœuds, ou de les rendre prématurément dysfonctionnelles. Il est impératif de disposer d'un nombre identique de nœuds dans les clusters quand les CHs effectuent l'agrégation de données de sorte que les rapports combinés deviennent prêts au même temps pour un traitement ultérieur au niveau de la station de base, ou au niveau supérieur dans la hiérarchie du réseau.

3.3.2. Tolérance aux fautes

Dans plusieurs applications, les s sont opérationnels dans des environnements hostiles où les nœuds sont exposés aux risques incessants de dysfonctionnement lié au dommage physique. Tolérer la défaillance des CH est nécessaire pour de telles applications afin d'éviter la perte de données importantes. La méthode la plus intuitive pour recouvrir la défaillance des CHs est le re-clustering du réseau. Toutefois, cette méthode est souvent source de perturbation des opérations courantes. Par conséquent, les nouvelles techniques de tolérance

aux fautes sont plus appropriées, telle l'assignation de CHs de secours (backup CH) et la rotation du rôle de CH entre les nœuds. Dans la première méthode, le rôle des backup-CHs peut varier durant le fonctionnement normal du réseau. La deuxième technique est une autre solution tolérante aux fautes en plus de ses avantages d'équilibrage des charges.

3.3.3. Amélioration de la connectivité et réduction des délais de transmission

La connectivité inter-cluster est impérative pour plusieurs applications. Quand un CH est un nœud ordinaire, l'objectif de la connectivité est limité à garantir la disponibilité des chemins entre chaque CH et la station de base ou réduite à limiter la longueur des chemins. Quand la latence des données est critique pour l'application, la connectivité intra cluster représente une recommandation à prendre en considération lors de la conception des clusters. Pour optimiser la latence des données, il est nécessaire de diminuer le nombre de sauts formant le chemin vers la destination.

3.3.4. Augmentation de la durée de vie du réseau

Comme les capteurs sont contraints en termes de ressources énergétiques, la durée de vie du réseau est un défi majeur principalement pour les applications des réseaux déployés dans des environnements difficiles. Quand les CHs sont plus riches en ressources il est impératif de minimiser l'énergie consommée lors des communications intra-cluster en les plaçant au centre du cluster près des nœuds membres de ce dernier. Si les CH sont des nœuds ordinaires, leur durée de vie peut être étendue en équilibrant les charges. Le clustering adaptatif et la configuration des routes peuvent aussi être considérées pour étendre la durée de vie du réseau [54,82].

3.3.5. Caractéristiques d'un protocole hiérarchique

Un protocole de routage hiérarchique doit spécifier plusieurs tâches que nous pouvons classer selon :

3.3.5.1. Algorithme de clustering utilisé

Du fait de la variété des objectifs qui sont visés, il existe plusieurs types d'algorithmes de clustering. Selon [55] Ces derniers peuvent être classés en 3 types : centralisé, distribué et géographique.

- Centralisé

Dans les solutions d'ordonnancement centralisées, il y a une entité centrale qui possède toutes les informations sur l'état du réseau. Cette entité est donc en mesure de calculer à chaque instant le chemin optimal entre deux nœuds. Ainsi tout nœud source désirant établir une connexion doit s'adresser à cette entité, ce qui augmente le temps pour calculer une route. De plus, il existe un problème de fiabilité important. En effet, si ce nœud de routage venait à être hors service, ou si un des liens le reliant au reste du réseau était coupé, il y aurait alors un impact sur le bon fonctionnement du réseau.

- Distribué

La technique du clustering réparti est une méthode locale dans laquelle les stations voisines échangent des messages sur l'état du trafic et du réseau afin d'obtenir la synchronisation des tâches de contrôle. On conçoit donc que ce système adaptatif puisse, avec un certain délai, minimiser la communication relative à la sauvegarde de la vue globale du réseau. C'est vers cette méthode que s'orientent les travaux, car avec l'apparition des réseaux hauts débit, il est important de réagir rapidement au changement de topologie du réseau. Cette méthode reste fiable, et le dysfonctionnement d'un nœud ne compromet pas la bonne marche du système [55, 56].

- Géographique

L'algorithme de la tâche de contrôle à exécuter par un nœud dépend de sa position géographique.

3.3.5.2. Réélection des cluster-heads

La rotation des CHs s'avère également un facteur important pour l'organisation des réseaux de capteurs. Puisque la BS (station de base ou sink) est généralement loin du champ des capteurs, les CHs diffusent une quantité plus importante d'énergie pour la transmission de données à la BS. Par conséquent, les CHs mourront rapidement si le même nœud fonctionne continuellement comme un CH. Ainsi, pour ne pas épuiser la puissance de batterie d'un capteur simple, les algorithmes de groupement (clustering) étudiés jusqu'ici présentent la rotation de CHs parmi les capteurs.

3.3.5.3. Nature des clusters générés

Les algorithmes de Clustering utilisés peuvent générer deux types de cluster: des clusters disjoints et des clusters interconnectés (Overlapped). Dans le premier type, un nœud ne peut appartenir qu'à un et un seul cluster à la fois (le cas le plus fréquent); sauf que pour des applications spécifiques telles que le routage inter-cluster, les clusters interconnectés sont utilisés. Ce type de Clustering permet aux nœuds d'appartenir à un ou plusieurs clusters à la fois

3.3.5.4. Communication intra-cluster

La communication entre le nœud ClusterHead et les autres nœuds membres du cluster peut se faire, soit en un seul saut soit, en plusieurs sauts. Dans le cas d'une communication direct (en un saut), les paquets de données sont envoyés directement au ClusterHead. Cela suppose que les nœuds membres sont capables d'atteindre le CH en utilisant une transmission assez puissante pour une bonne réception de données. Ce type de communication engendre une consommation importante d'énergie si la distance, entre le CH et les nœuds, est grande. Pour réduire la consommation de l'énergie, une communication en plusieurs sauts, de petites distances, est utilisée, dans ce cas chaque membre du cluster envoie ses données au plus proche membre de son cluster jusqu'à l'aboutissement au CH. Des communications de type Unicast sont souvent utilisées pour réduire le nombre de collisions. On trouve aussi des techniques employées au niveau MAC pour garantir un accès équitable et sans erreurs, à savoir, CDMA (Code Division Multiple Access) en utilisant un code pour chaque cluster, TDMA (Time Division Multiple Access) en allouant pour chaque nœud du cluster un temps

spécifique (dit Frame) pour envoyer ses données, ou bien, FDMA (Frequency Division Multiple Access) ou chaque nœud utilise une fréquence spécifique pour l'envoi de données.

3.3.5.5. Communication inter-cluster

Les CH communiquent avec la (les) station(s) de base soit directement, soit en deux ou plusieurs sauts via des nœuds appelés généralement des nœuds Gateway. Ces nœuds peuvent être des CH ou bien des nœuds membres d'un cluster. L'utilisation de la communication en multi-sauts permet de réduire la consommation d'énergie et d'augmenter la scalabilité du réseau.

3.3.5.6. Niveau d'agrégation de données

Puisque les nœuds capteurs peuvent produire des données significatives et superflues, les paquets des différents nœuds peuvent être agrégés de sorte que le nombre de transmissions soit réduit. L'agrégation de données est la combinaison des données de différentes sources selon une certaine fonction d'agrégation : suppression, minimum, maximum et moyenne, etc. Cette technique a été employée pour optimiser la consommation d'énergie lors de transfert des données dans un certain nombre de protocoles de routage [57].

3.4. Exemples de protocoles de routage hiérarchiques pour les réseaux de capteurs

Dans cette section, nous allons décrire le fonctionnement de quelques algorithmes de routage hiérarchiques.

3.4.1. LEACH (Low Energy Adaptive Clustering Hierarchy)

Proposé par Heinzelman et al, LEACH [23,24] est l'un des premiers protocoles de routage hiérarchiques conçu pour les réseaux de capteurs sans fil. Bénéficiant des avantages de l'approche de clustering, ce nouveau protocole a permis de pallier à différents problèmes rencontrés dans les protocoles de routages traditionnels basés sur une topologie plate du réseau.

Dans LEACH les nœuds s'auto organisent en clusters guidés par des chefs de cluster (cluster heads). Les nœuds non-CHs envoient leurs données vers les CHs. Quand un CH reçoit des données de la part des nœuds membres de son cluster, il effectue un traitement adéquat selon l'application avant de les envoyer vers la SB. Les CHs consomment plus d'énergie que les nœuds ordinaires. Si les CHs sont fixés a priori durant la durée de vie du réseau, ces nœuds épuisent rapidement leurs ressources limitées. Ainsi, LEACH incorpore la sélection randomisée des CHs de manière à ce que chaque nœud soit élu cluster head une fois durant un cycle et tous les nœuds doivent passer par le rôle CH afin d'équilibrer la consommation d'énergie entre tous les nœuds du réseau. L'élection du cluster head repose sur le choix d'une valeur aléatoirement entre 0 et 1 par tous les nœuds du réseau. Si cette valeur est inférieure à un seuil $T(n)$, alors le nœud est élu CH. Le seuil $T(n)$ est calculé selon la formule :

$$T(n) = \begin{cases} \frac{P}{1 - P \times \left(r \bmod \frac{1}{P}\right)} & \text{si } (n \in G) \\ 0 & \text{sinon} \end{cases}$$

Avec P : le pourcentage de CHs désiré, r : round actuel, G : ensemble des n'ayant pas été élus comme CH durant les r-1 rounds précédents.

LEACH s'exécute en round, chaque round est divisé en deux phases :

Construction des clusters

Après qu'un nœud est choisi CH, il informe tous les autres nœuds non CH de son nouveau rang dans ce cycle et ceci, en diffusant un message contenant son identificateur. La diffusion est faite en utilisant le protocole MAC CSMA (Medium Access Control Carrier Sense Multiple Access) pour éviter les collisions entre les CHs. Chaque nœud choisi son cluster-head selon la puissance du signal reçu, puis il l'informe en lui envoyant un message d'affiliation. Chaque cluster-head ayant reçu les messages des nœuds désirant appartenir à son cluster, diffuse un ordonnancement TDMA (Time Division Multiple Acces) aux membres de son cluster en attribuant à chaque membre un intervalle de temps dans une table appelée table TDMA durant lequel il pourra communiquer ses données ainsi qu'un code MAC CDMA (Code Division Multiple Access) afin d'éviter les collisions entre les clusters.

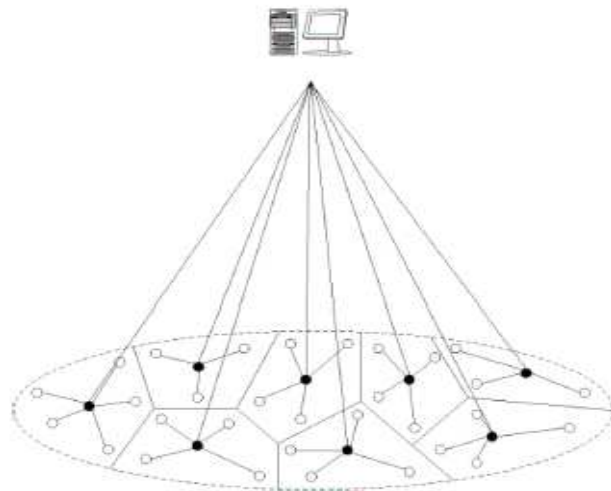


Fig.3.2 : architecture du réseau en clusters dans LEACH

Transmission de données

Les données dans LEACH sont délivrées selon le paradigme de communication time-driven. A la fin de chaque cycle, les nœuds transmettent les données aux cluster-heads en un seul saut pendant leurs slots appropriés. Cela leur permet d'éteindre leurs interfaces en dehors du slot. Les CHs font l'agrégation et la compression de ces données pour les envoyer à la station de base.

Ce processus est réitéré jusqu'à ce que tous les nœuds soient élus CHs, une seule fois au long des rounds précédents. Une fois fait, le round est réinitialisé à 0.

Discussion

Les résultats ont montré que LEACH améliore la consommation énergétique par rapport aux protocoles de routage plats multi sauts. L'adoption de l'approche de clustering basé sur la division du réseau en groupes guidés par des chefs de clusters permet le traitement local des données provenant des nœuds appartenant à un même cluster réduisant ainsi le nombre de paquets devant être transmis à la SB.

LEACH est un protocole décentralisé, l'auto élection des CHs minimise le trafic qui pourrait être induit par une élection basée sur des informations échangées entre les nœuds.

L'utilisation de la méthode d'accès TDMA permet aux nœuds d'éteindre leurs interfaces en dehors de leurs slots réservés et donc économiser de l'énergie. L'ordonnanceur TDMA évite aussi les collisions sur le canal de transmission. Toutefois, vu le manque de connaissances sur les positions des nœuds dans le réseau, la répartition des CHs ne peut être gérée ce qui conduit à une distribution non uniforme des CHs. L'élection des CHs en se basant sur une valeur stochastique permet à un nœud ayant un niveau d'énergie critique d'être élu comme CH. la négligence de l'énergie résiduelle des nœuds lors de ce processus conduit à leur mort prématuré du CH. La panne de ce dernier au milieu du round cause la déconnexion du cluster et la perte des données captée dans cette partie du réseau. Un autre inconvénient est la formation de cluster de tailles inégales ce qui déséquilibre la consommation de l'énergie dans le réseau.

LEACH est un protocole simple et ne nécessite pas une surcharge des communications. Mais ses performances dans des réseaux hétérogènes se dégradent puisque il ne prend pas en considération le niveau d'énergie des nœuds. D'autre part, il est supposé dans LEACH que tous les nœuds ont assez d'énergie pour transmettre vers la SB s'il le faut, c.à.d. les nœuds non compris dans les clusters peuvent communiquer directement avec la SB. Afin de communiquer avec la SB, ces nœuds augmentent leurs puissance d'émission ce qui épuise leurs batteries.

3.4.2. DECSA (Distance-Energy Cluster Structure Algorithm)

DECSA [34] est une approche de routage hiérarchique récemment proposée. Il est basé sur une méthode de clustering inégal distribuée où les nœuds entrent en compétition pour devenir CH. DECSA améliore le processus d'élection de CHs en prenant en considération la distance et l'énergie résiduelle des nœuds. L'élection d'un CH très éloigné de la station de base peut épuiser ce nœud et produire des endroits aveugles (blind spot) dans le réseau. Afin d'éviter cette situation, une structure hiérarchique à trois niveaux du réseau est utilisée. Dans cette structure, les nœuds sont divisés en 4 catégories : Base Station(BS), Base Station Cluster head(BCH), CH ordinaire et des nœuds senseurs ordinaires(SN).

Similaire au protocole LEACH, DECSA s'exécute en round. Chaque round est divisé en deux phases :

Phase d'initialisation

Durant cette phase, les CHs sont élus et des slots TDMA sont affectés aux nœuds par le CH. Le processus d'élection des CHs est divisé en deux parties : élection des CH ordinaires et

l'élection des BCH. Pour l'élection des CHs, DECSA utilise l'énergie résiduelle et la distance. Initialement chaque nœud génère un nombre aléatoire compris entre 0 et 1. si le nombre aléatoire généré par un nœud donné est inférieur à un seuil T prédéterminé, ce nœud devient un faux-CH durant le round courant. Les membres du cluster calculent leurs $k(i)$ et les compare à celle de leur faux-CH. Si le K d'un nœud est supérieur à celui de son faux-CH, il s'annonce CH, sinon, le faux-CH se déclare CH. le $K(i)$ est ainsi calculé :

$$k(i) = E_n(i) \div d_0(i)$$

Où $k(i)$ est le seuil d'élection du CH, $E_n(i)$ est l'énergie résiduelle du nœud i , d_0 est la moyenne de la distance entre le nœuds i et les autres nœuds du cluster.

Après l'élection des CH et dans la partie d'élection des nœuds BCH, on utilise un seuil TBCH pour élire les CH qui seront des BCH. Les CH sélectionnés sont ceux ayant un $TBCH(i)$ supérieur à $TBCH_0$. $TBCH(i)$ est défini comme suit :

$$TBCH(i) = (E_n(i) \div E_0) + (E_n(i) \div d(i))$$

$E_n(i)$ étant l'énergie résiduelle du nœud i , E_0 l'énergie initiale des nœuds dans le réseau, $d(i)$ la distance entre le nœud i et la SB.

Phase de transmission de données

La SB diffuse un message de début de la phase de transmission. Après avoir reçu ce message, les BCH sélectionnent, selon la valeur de TBCH, le CH ayant un TBCH maximal pour relayer le paquet de données. Les prochains sauts sont sélectionnés de la même manière jusqu'à ce que tous les CHs soient connectés. Ainsi, un chemin de communication complet est formé. Les nœuds ordinaires envoient leurs paquets de données vers leur chef de cluster qui fusionne les données et les envoie, à son tour au BCH au lieu de les envoyer directement vers la SB. La valeur de $TBCH_0$ doit être changée dynamiquement selon les changements de l'état du réseau afin de garantir que le BCH choisi est le plus approprié. La valeur de $TBCH_0$ doit être comprise entre la moyenne de TBCH et la valeur TBCH maximale dans le réseau.

Discussion

Les résultats de la simulation ont montré que DECSA augmente la durée de vie du réseau de 31% et diminue la consommation l'énergie de 40% par rapport à LEACH en améliorant le processus d'élection des CHs et de formation des clusters. L'élection dans ce protocole se fait en se basant sur l'énergie résiduelle des nœuds et leur distance par rapport à la SB ce qui attribue la plus grande probabilité aux nœuds avec un haut niveau d'énergie d'être élu comme CH. Cette méthode permet de prolonger la durée de vie du réseau. Cependant, le choix de la valeur $TBCH_0$ affecte directement les performances du réseau. Le changement de $TBCH_0$ selon l'état du réseau qui doit être perçu en temps réel s'avère difficile dû aux délais de communication dans le réseau et les changements rapide de l'état des nœuds. Le choix du seuil doit être fait précocement afin d'éviter de fausser les résultats.

3.4.3. EADUC (Energy Aware Distributed Unequal Clustering Protocol)

EADUC [28] est un autre exemple de protocole de routage hiérarchique basé sur une approche de clustering distribué qui permet d'organiser les nœuds d'un réseau hétérogène en clusters de tailles inégales. L'élection des CHs se fait en fonction d'un rapport entre la moyenne des énergies résiduelles des voisins d'un nœud et l'énergie résiduelle du nœud lui-même. Tout comme LEACH, EADUC est divisé en deux phases. Une phase d'initialisation et une phase de transmission de données. Pour former la topologie du clustering, la première phase est divisée en trois étapes : une étape de collection des informations des voisins, une phase de compétition des CHs et la phase de formation des clusters.

Phase d'initialisation

La SB envoie un signal avec une puissance donnée. Tous les nœuds recevant ce message calculent leurs distances par rapport à la SB à partir de la puissance du signal reçu.

Afin de découvrir leur voisinage, les nœuds échangent entre eux des informations en diffusant un message Node_Msg sur un rayon r . ce message contient son identifiant ainsi que son niveau d'énergie. A la réception des messages Node_Msg de la part de ses voisins, chaque nœud calcule la moyenne de leurs énergies selon la formule suivante :

$$E_a = \frac{1}{d} \sum_{i=1}^d S_i \cdot E_r$$

Où : S_i représente un nœud, E_r son énergie résiduelle et d le nombre de ses voisins.

Par la suite, chaque nœud calcule un délai d'attente t après lequel il devra diffuser un message d'annonce pour informer ses voisins de son nouveau statut. Le message d'annonce est diffusé dans une zone d'un rayon R_c .

Le délai d'attente est calculé selon la formule

$$t = \begin{cases} \frac{E_a}{E_r} T_2 V_r, & E_r \geq E_a \\ T_2 V_r, & E_r < E_a \end{cases}$$

En considérant l'énergie résiduelle des nœuds et leurs distances par rapport à la SB, le rayon R_c est calculé comme suit :

$$R_c = [1 - \alpha \frac{d_{max} - d(s_i, BS)}{d_{max} - d_{min}} - \beta(1 - \frac{E_r}{E_{max}})] R_{max}$$

Une grande distance par rapport à la station de base et un haut niveau d'énergie augmente le rayon R_c ce qui veut dire, que les CH loin de la SB et riche en énergie contrôlent des clusters sur de large surface tandis que les plus proches de la SB observent des clusters de petites tailles afin d'éviter leur mort prématuré, donc la déconnexion du réseau puisque ces CH sont les relais entre la SB et le reste du réseau.

Phase de transmission de données

Dans la phase de transmission des données, les CMs collectent des données de leurs environnements, les envoient vers leurs CHs directement. Pour les communications inter-cluster afin qu'ils soient agrégés et relayés vers le prochain sot. Dans cette phase, EADUC introduit un seuil DIST_TH. Si la distance entre le cluster head S_i et la SB est inférieure à DIST_TH, alors le CH transmet ses données directement vers la SB, sinon S_i sélectionne parmi ses voisins un ensemble de nœud candidat à être le prochain relais, puis sélectionne, dans cet ensemble le relais final selon le paramètre E_{relay} pour que S_i calcule E_{relay} pour choisir S_j comme relais, il utilise la formule suivante :

$$E_{relay} = d^2(S_i, S_j) + d^2(S_i, S_j)$$

Alors le CH sélectionne le CH S_j avec un maximum d'énergie résiduelle parmi les deux nœuds ayant le minimum d'énergie résiduelle dans l'ensemble des nœuds candidats à être relais. Après cette phase, les CHs construisent, avec la SB, un arbre contenant les chemins vers cette dernière. Les données seront transmises via les arcs de cet arbre vers la SB.

Discussion

Sachant que les nœuds dans un réseau hétérogène ont, initialement, des niveaux d'énergie différents. Dans ce cas, si tous les nœuds consomment leur énergie de la même manière, les nœuds ayant une faible énergie meurt prématurément réduisant ainsi la durée de vie du réseau. Afin de tirer profit de l'existence de nœuds plus puissant, ces derniers doivent assurer plus de tâches que les autres.

Dans EADUC, les auteurs ont supposé que les CHs générés peuvent communiquer directement avec la station de base. Les CHs les plus éloignés de la SB sont supposés être de tailles plus grandes ce qui augmente leur consommation d'énergie lors de la réception de grand nombre paquets envoyés par les CMs et lors de leur agrégation. La communication de ces CHs sur de grandes distances avec la SB engendre une surconsommation de l'énergie ce qui épuisent rapidement leurs ressources et cause la déconnexion de leurs clusters du reste du réseau. En outre, l'énergie résiduelle des voisins d'un nœud est une information utilisée durant toutes les étapes de formation des clusters. Ceci nécessite la mise à jour continuelle de cette information afin d'obtenir des résultats fiables. Le trafic de contrôle engendré par les mises à jour dégrade les performances du réseau et diminue sa durée de vie. L'existence de nœuds puissants dans le réseau augmente son coût et surtout dans le cas d'un réseau d'une grande envergure.

3.4.4. CW-EBCR (combination weight energy balance clustering and routing algorithm)

Le protocole CW-EBCR [31] prend en considération plusieurs paramètres pour l'élection des CHs. Ces paramètres sont : le degré d'un nœud, son niveau d'énergie, la distance moyenne entre un nœud et ses voisins, le total du temps pendant lequel un nœud était CH, la distance par rapport au sink et le nombre de clusters généré.

Dans cet algorithme, le réseau est considéré comme un graphe de connexion $G = (V, E, W)$ où V étant l'ensemble des nœuds, E l'ensemble des liens entre chaque deux nœuds

capables de communiquer entre eux en un seul saut et W l'ensemble des poids. Au début de l'algorithme, les nœuds calculent un poids selon la formule suivante :

$$W(V_i) = \alpha \frac{dg_{max}(V_j)}{dg(V_i)} + \beta lg \frac{T(V_i)}{T_{thr}} + \gamma \bar{d}(V_i) + \lambda \frac{1}{\eta(V_i)}$$

Où :

Dg_{max} est le degré maximal d'un nœud dans le réseau, $dg(V_i)$ est le degré du nœud i , $T(V_i)$: le temps durant lequel le nœud i est élu CH, T_{thr} : le temps seuil, $\bar{d}(V_i)$: la distance moyenne entre le nœud V_i et ses voisins, η : le rapport entre l'énergie résiduelle du nœud V_i et la moyenne des énergies résiduelles de tous les nœuds dans le réseau. $\alpha, \beta, \gamma, \lambda$ des coefficients de pondération tels que : $\alpha + \beta + \gamma + \lambda = 1$.

Processus de clustering initial

Si tous les nœuds ont la même quantité d'énergie initialement, alors le processus de clustering de LEACH est utilisé au premier round et le seuil T_0 est calculé ainsi :

$$T_0 = C_{head} / (N - C_{head})$$

Où C_{head} est le nombre optimal de CH calculé par la fonction suivante :

$$C_{head} = \frac{\sqrt{N}}{2\pi} \sqrt{\frac{\xi_{fs}}{\xi_{mp}}} \frac{A}{d_{toBS}^2}$$

Où N est le nombre de nœud et A la taille de la zone couverte.

Si l'énergie initiale n'est pas la même pour tous les nœuds, ce processus ne peut pas garantir que le nœud ayant le maximum d'énergie soit élu CH. Pour cette raison, un processus initial de clustering est utilisé. Ce processus est divisé en trois étapes :

- Etape1: étape de découverte de voisinage lancée par la SB en envoyant un message de Hello (ID, W, P). Ce message est relayé par les nœuds récepteurs en utilisant le même rayon de transmission. Les messages reçus sont stockés dans les tables de voisinage.
- Etape2 : étape d'ajustement des poids. Les nœuds envoient à leurs voisins un message Notice. une fois le message reçu, ces derniers mettent à jour leurs poids selon la formule suivante :

$$W(V_i) = \alpha_1 \frac{dg_{max}(V_j)}{dg(V_i)} + \alpha_2 \bar{d}(V_i) + \alpha_3 \frac{1}{W(V_i)_{old}}$$

Après l'ajustement des poids, les nœuds notifient leurs voisins.

- Etape3 : Le nœud avec le plus petit poids entre ses voisins se déclare candidat pour devenir CH et informe ses voisins en diffusant un message Leader sur le plus grand rayon de communication. Les nœuds se trouvant éloignés de la SB se déclarent candidat pour devenir CH.

Détection des routes entre clusters

La SB envoie un message pour initialiser cette phase. Des messages de découverte de routes sont envoyés entre les CHs. les CH qui garantissent une consommation minimale de l'énergie sont choisis comme relais vers la station de base.

Discussion

Les résultats de la simulation montrent que le nouveau protocole CW-EBCR diminue et équilibre la consommation de l'énergie dans le réseau et augmente ainsi sa durée de vie de 20% comparé avec ACO-EBR. Avant de router les données, les chemins sont évalués selon leurs couts et ceci en utilisant des messages de découverte de voisinage pour trouver des relais optimaux. Toutefois, le grand nombre de messages engendré durant la phase de clustering et de routage dans le but de contrôler les paramètres de ce protocole augmente la congestion dans le réseau et augmente la consommation de l'énergie.

3.4.5. UCRA (Unequal Clustering Routing Algorithm)

UCRA a été proposé par Ruihua Zhang, Lei Ju, Zhiping Jia et Xin Li dans [58]. UCRA est un protocole à topologie hiérarchique, son approche permet d'utiliser des communications multi-sauts (inter-clusters) pour l'acheminement des données collectées par les cluster-heads vers la station de base en se basant sur le minimum des énergies à consommer. L'idée de base d'UCRA est de construire des clusters inégaux en considérant que la taille d'un cluster est proportionnelle à sa distance par rapport à la station de base. L'objectif de cette approche de clustering est que les CH les plus proches de la SB devraient soutenir de plus petites tailles de groupe en raison de la plus haute consommation d'énergie pendant la communication inter-cluster.

Algorithme de clustering

UCRA, considère un champ de détection carré, et un total de nœud capteurs N est uniformément distribué dans le champ. Selon la supposition, dans l'étape de déploiement du réseau, le nœud puits calcule sa distance à chaque nœud, puis émet un message en diffusion à travers le réseau. Ce message inclut le paramètre dMax, dMin et l'emplacement du nœud puits. Chaque nœud reçoit le message du nœud puits, et calcule son rayon de concurrence selon la fonction suivante :

$$R_i = \left(1 - c \times \frac{d_{max} - d(i, SB)}{d_{max} - d_{min}}\right) R_{max}$$

Avec :

d(i,SB) : est la distance entre i et SB.

c: est un coefficient constant entre 0 et 1.

d_{max} et d_{min} : dénotent la distance maximum et minimum entre les nœuds capteurs et la station de base.

R_{max} : est le rayon maximum de concurrence qui est prédéfini.

Algorithme de routage des données

- Transmission des données intra-cluster

Les transmissions de données intra-cluster commencent après que les clusters aient été formés. Chaque CH reçoit et agrège les données de ses membres de groupe.

- Transmission des données inter-cluster

Les CH transmettent leurs données à la station de base avec une communication multi-sauts. Dans UCRA, les CH relais n'agrègent pas les paquets entrant. Chaque capteur connaît un seuil TD_MAX qui est utilisé dans le modèle de transfert par multi-sauts, si la distance d'un CH à la station de base est plus petite que TD_MAX il transmet ses données directement à la SB, et autrement il devrait trouver un nœud de relais qui peut transmettre ses données à la SB.

Afin de minimiser l'énergie consommée lors des communications inter-clusters, UCRA considère les deux corollaires suivant :

Corolaire 1

Etant donné la distance D et le nombre de saut k, la consommation totale d'énergie est minimisée quand toutes les distances des sauts sont rendues égales à D/k.

Corolaire 2

Le nombre optimal de sauts pour une communication est donné par la formule suivante :

$$k_{opt} = \left\lfloor \frac{D}{d_{opt1hop}} \right\rfloor$$

$d_{opt1hop}$: Représente la distance idéale pour un saut (distance caractéristique), elle est donnée par la formule suivante :

$$d_{opt1hop} = \sqrt[\alpha]{\frac{E_{elecTx} + E_{elecRx}}{(1-\alpha)E_{amp}}}$$

$d_{opt1hop}$, ne dépend pas de la distance D de transmission, elle dépend seulement des caractéristiques physiques du circuit de transmission et de la constante de la déperdition.

- L'algorithme de routage multi-sauts peut être expliqué par l'exemple de la figure Fig.3.3:

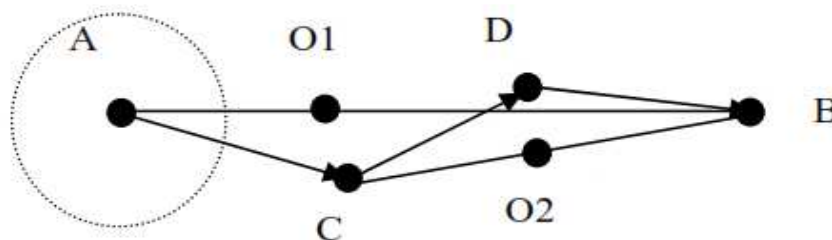


Fig.3.3 : procédure de routage de l'algorithme de routage multi-sauts MEC

Dans cet exemple un CH A doit délivrer un paquet à la station de base B. Dans cette phase, les CH changent leur puissance de transmission qui dépend de la distance vers la SB. Les opérations dans chaque CH sur le chemin peuvent être récapitulées comme suit :

Etape1

D'abord, le nœud A calcule la distance optimale d'un saut $d_{opt1hop}$ par la formule (2) et la distance $d(AB)$ entre A et B. Selon l'emplacement du nœud A(x_A, y_A) et B(x_B, y_B), le nœud A calcule l'emplacement optimal idéal prochain de relais par saut $O(x_{optHop}, y_{optHop})$ sur le segment de ligne AB, O est seulement une position idéale, et non pas un CH pratique. x_{optHop}, y_{optHop} sont calculées par les formules suivantes :

$$x_{optHop} = x_A + \frac{d_{opt1hop}}{d(AB)}(x_B - x_A)$$

$$y_{optHop} = y_A + \frac{d_{opt1hop}}{d(AB)}(y_B - y_A)$$

En second lieu, le nœud A émet un message *QUERY_MSG* (ID, $d(AB)$) à ses CHs voisins. Le CH voisin qui reçoit ce message calcule la distance $d(i,B)$ et la compare à $d(AB)$. Si $d(iB) < d(AB)$, il émet un message *answer_Msg*(ID, (x_i, y_i)) au nœud A.

Finalement le nœud A rassemble tous les messages *ANSWER_Msg*, les sauvegarde, et calcule leurs distance d_i, O de i à O et compare les d_i, O et il détermine le CH le plus proche de O , Ainsi le CH A sélectionne le nœud C comme prochain saut et lui délivre le paquet (comme mentionné dans Fig 3.3)

Etape2

Quand le paquet est acheminé de A à C, un procédé semblable à l'étape 1 est répété u nœud C sur le segment CB. Ainsi, il trouve le prochain emplacement optimal O2 et le prochain Ch relais D, alors C délivre son paquet au Ch D.

Etape3

Quand le paquet est acheminé de C à D, le CH D calcule la distance $d(DB)$ et la compare au seuil pré-sauvegardé. TD_MAX . Si $d(DB) < TD_MAX$ alors D délivre directement le paquet à la station de base.

Discussion

Les résultats ont montrés que UCRA produit une plus longue durée de vie comparé au protocole HEED. Puisque HEED génère des clusters équilibrés en taille (equal clustering), il ne prend pas en considération le problème de la consommation d'énergie non équilibré entre CH. Dans HEED, les CH transmettent leurs données au sink, avec une transmission multi-saut, et choisissent aléatoirement leurs CH relais en se basant uniquement sur la distance. Par conséquent des CH avec une basse énergie résiduelle peuvent toujours servir de nœuds relais ce qui engendre la mort de certains nœuds relai plus tôt que les autres.

3.4.6. PEGASIS (Power-Efficient Gathering in Sensor Information System)

Quand les nœuds ont à transmettre leurs données directement vers la station de base comme est le cas pour les nœuds non couverts par des CHs dans le protocole LEACH, ces derniers augmentent leur énergie de transmission et épuisent rapidement leurs batteries. Afin de pallier à ce problème, la nouvelle approche PEGASIS basée sur le protocole LEACH a été proposée par Lindsay et Raghavendra dans [25, 26]. L'idée de base de ce protocole est de construire des chaînes entre les nœuds senseurs de manière à ce que les communications s'effectuent avec les nœuds voisins. Le voisinage est déterminé selon la puissance du signal reçu. Les données reçues sont agrégées et transmises d'un nœud à un autre jusqu'à ce qu'elles atteignent un nœud désigné comme « leader » qui les transmet à la SB. Les nœuds qui transmettent les données vers la SB sont désignés parmi les membres de la chaîne tour à tour durant chaque round afin d'équilibrer la consommation de l'énergie entre les nœuds du réseau.

Discussion

Les résultats de simulation ont montré que PEGASIS peut prolonger de deux à trois fois la durée de vie d'un réseau de capteurs relativement à LEACH. Un tel gain de performance est réalisé par : l'élimination de l'overhead causé par le processus de formation de clusters dans LEACH (ce protocole n'utilise qu'un seul nœud dans une chaîne afin de transmettre à la station de base au lieu d'en utiliser plusieurs), ainsi que par l'utilisation de l'agrégation de données qui réduit le nombre de transmissions, de réceptions, et la largeur de la bande consommée dans la communication.

Bien que l'overhead du clustering soit évité, PEGASIS exige toujours un ajustement dynamique de la topologie puisqu'un nœud devrait connaître le niveau d'énergie de ses voisins avant de relayer ses données. Cependant, un tel ajustement de la topologie pourrait causer un overhead important. En outre, PEGASIS suppose que tout nœud est capable de communiquer directement avec la station de base. Or, cette supposition est loin de la réalité car les capteurs communiquent généralement en mode multi-sauts pour atteindre la station de base. En résumé, PEGASIS est adapté pour des réseaux où les nœuds senseurs sont immobiles. Son évaluation dans un environnement mobile peut dégrader considérablement ses performances.

3.4.7. HEED (Hybrid Energy Efficient Distributed Approach)

Dans [27], une nouvelle approche de clustering distribué a été proposée afin de prolonger la durée de vie d'un réseau de capteurs ad hoc sans fil. Younis et Fahmy ont développé le protocole HEED qui sélectionne les CH selon un critère hybride regroupant l'énergie résiduelle d'un nœud et un second paramètre tel que la proximité d'un nœud à ses voisins ou son degré. Il vise à réaliser une distribution uniforme des CHs et générer des clusters équilibrés en taille. Le premier paramètre est utilisé afin de choisir les CHs en calculant une probabilité P_{CH} les nœuds:

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

Où E_u est l'énergie restante du nœud u , E_{total} l'énergie globale dans le réseau et C_{prob} est le nombre optimal de clusters. Tandis que le deuxième paramètre est utilisé pour résoudre les conflits c.à.d. lorsqu'un nœud se trouve à la portée de plusieurs CHs. Dans ce cas, si le niveau de puissance utilisé pour la communication intra-cluster est fixe pour tous les nœuds, alors le coût de communication est proportionnel au degré du nœud. Cela signifie qu'un nœud joint le CH avec le degré minimum pour distribuer la charge ou bien il joint celui avec le degré max pour créer des clusters denses. Une fois les clusters sont formés, les CHs affectent aux membres de leurs groupes des tranches de temps différentes pendant lesquelles ils peuvent transmettre leurs données.

Après la collecte des données, les CHs acheminent l'information agrégée directement vers la station de base via un seul saut. Ces CHs peuvent alternativement communiquer avec l'un ou l'autre, pour agréger leur informations par des multiples sauts vue la connectivité assurée par le protocole HEED.

Discussion

HEED présente plusieurs arguments en sa faveur. C'est un protocole non centralisé convergent ; le clustering se termine dans $O(1)$ itérations. Il prolonge la durée de vie du réseau en distribuant l'énergie de la communication et le nombre de CHs d'une façon uniforme et donc produire des clusters compact tout en minimisant la charge des messages de contrôle. Son processus d'itérations est indépendant de la topologie ou de la taille du réseau. En outre, il n'indique aucune supposition sur la distribution ou la densité des nœuds, ainsi que leurs capacités.

L'évaluation d' E_{total} présente une certaine difficulté, à cause de l'absence de toute commande centrale. Un autre problème réside dans la détermination du nombre optimal de clusters. De plus, HEED ne précise pas de protocole particulier à utiliser pour la communication entre les CHs et le sink. A l'intérieur du cluster, le problème ne se pose pas car la communication entre les membres du cluster et le cluster CH est directe (à un saut). D'autre part, avec HEED, la topologie en clusters ne réalise pas de consommation minimale d'énergie dans les communications intra-cluster et les clusters générés ne sont pas équilibrés en taille.

Après la collecte des données des membres de la grappe, le chef achemine l'information agrégée directement vers la station de base. Cela limite l'étendue du réseau et pose un problème pour l'adéquation de cet algorithme pour les grands réseaux.

3.4.8. ADRP (Adaptive Decentralized Re-clustering Protocol)

ADRP [50] est un protocole de clustering conçu pour les réseaux de capteurs. Il est utilisé pour élire des cluster-heads et des prochain-head (next-heads). La sélection des CHs et des next-heads se fait selon un poids calculé qui dépend de l'énergie restante de chaque nœud senseur et de l'énergie moyenne de chaque cluster. ADRP est un protocole adaptatif où le rôle de CHs est alterné entre les nœuds afin de balancer la dissipation de l'énergie dans le réseau.

Dans ADRP l'activité du réseau est organisée en round. Chaque round est divisé en deux(02) phases. Une phase initiale est une phase de cycle. La première phase consiste à organiser le réseau et elle est plus courte que la deuxième afin de minimiser la charge. Durant la phase d'initialisation, la station de base reçoit des informations sur les positions et le

niveau d'énergie de chaque nœud. Chaque nœud récupère sa position grâce à un système de localisation global(GPS) activé au début de chaque phase d'initialisation. Une fois les données reçues, la SB calcul le niveau d'énergie de tous les nœuds et élit les CHs en minimisant la somme totale des distances entre les CHs et les nœuds senseurs. La SB garantie que seulement les nœuds ayant assez d'énergie participent à l'élection des CHs. ADRP permet de distribuer la consommation de l'énergie entre les nœuds senseurs en positionnant le CH au centre du cluster. En plus des nœuds CHs et membres, ADRP implémente un nouveau type de nœud qui consiste en prochain-head (next-head).

Le CH est responsable de la collecte des données captées au niveau de son cluster et de leur agrégation. si ce rôle était fixé, les CHs auraient épuisé leur énergie rapidement et causé la déconnection du réseau. Afin d'éviter de tels scénario, le rôle de cluster head est alterné parmi les nœuds.

Une fois les CHs élus, ADRP sélectionne un ensemble de nœuds comme next-heads. L'élection se fait en calculant un seuil (énergie moyenne) de chaque cluster. Les nœuds ayant un niveau d'énergie inférieur au seuil calculé ne peuvent pas devenir next-heads durant le round courant. Le seuil est calculé selon la formule suivante :

$$T_j = \frac{1}{m} \sum_{i=1}^m E_i(t)$$

Où m est le nombre de nœuds dans le cluster j et $E_i(t)$ est le niveau d'énergie courant du nœud i . La fonction privilégie les nœuds ayant le plus haut niveau d'énergie à devenir next-head. Si $E_i(t) \geq T_j$, alors le nœud i est membre de l'ensemble NH_j où NH_j est l'ensemble des nœuds pouvant être élus next-head dans le cluster j . après la construction des ensembles NH_j , ADRP élit, parmi les nœuds de cet ensemble les next-heads et leurs nœuds membres. Comme prochaine étape, la SB envoie un message contenant l'ID du cluster-head et du next-head à la destination de tous les nœuds.

Durant la phase re-clustering, les nœuds transitent vers les next-head sans l'assistance de la SB. Si l'ensemble des next-head est vide alors une nouvelle phase d'initialisation s'exécute une autre fois.

Durant la phase de transmission, les CHs affectent des stots TDMA aux membres de leurs clusters durant lesquels ils émettent leurs données captées. Le clusters-head agrège les données reçues et envoie le rapport résultant vers la SB en utilisant le protocole MAC CSMA

Discussion

Durant la phase de re-clustering, les nœuds transitent directement vers les next-heads. Ce mécanisme implémenté par ADRP permet la minimisation de la surcharge due aux communications avec la SB dans le réseau. En plus de privilégier les nœuds ayant le plus d'énergie à devenir CHs ou next-heads, ADRP garantie le positionnement des CHs au centre des clusters, ce qui balance la consommation de l'énergie entre les nœuds de chaque cluster. Ainsi, ADRP permet aux nœuds de consommer la moindre quantité d'énergie et prolonge la durée de vie du réseau. Toutefois, l'utilisation du GPS afin de se localiser les nœuds a un

coût élevé. De plus, l'envoi des informations vers la SB au début de chaque phase d'initialisation augmente la consommation de l'énergie des capteurs vu que la SB est éloigné du champ de captage.

3.5. Conclusion

Nous avons traité, le long de ce chapitre, l'approche de clustering dans les réseaux de capteurs sans fil. Nous avons commencé par une définition de cette dernière puis nous sommes passées à la description du processus de clustering et énumérer ses objectifs, ses avantages ainsi que certaines des caractéristiques d'un protocole de routage hiérarchique. Plusieurs méthodes d'élection de cluster-heads ont été proposées dans la littérature dont nous avons présenté quelques exemples.

Bien que ces techniques de routage semblent prometteuses, ils restent encore de nombreux défis à relever en proposant de nouvelles approches de routage efficaces en énergie.

Dans le chapitre suivant, nous allons décrire le nouveau protocole de routage que nous avons développé. Nous allons également détailler son principe de fonctionnement ainsi que son avantage par rapport au protocole EAR sur lequel nous nous sommes basé pour accomplir notre travail.

Chapitre 4 : HEAR : un nouveau protocole de routage hiérarchique pour les RCSF

4.1. Introduction

Nous avons vu dans le chapitre précédent l'apport des protocoles de routage hiérarchiques aux réseaux de capteurs en termes de consommation d'énergie et de prolongement de la durée de vie du réseau. Pour cette raison, nous avons opté pour le développement du protocole plat EAR basé sur un routage multi-chemins dont nous avons étalé les avantages, et ceci, en introduisant la notion de clustering qui minimise considérablement le trafic dans le réseau grâce au mécanisme d'agrégation de données.

Dans le reste du chapitre, nous allons présenter le protocole EAR sur lequel se base notre travail. Par la suite, nous passerons à la description du nouveau protocole que nous avons développé et que nous avons nommé HEAR (*Hierarchical Energy Aware Routing Protocol*).

4.2. Protocole EAR (Energy Aware Routing)

L'utilisation des chemins avec un minimum d'énergie fréquemment peut ne pas être optimale du point de vue de la durée de vie du réseau et de la connectivité du réseau. Afin d'optimiser ces mesures, un nouveau schéma de routage a été proposé dans [29].

EAR est un protocole réactif tout comme AODV conçu pour les réseaux ad hoc et le protocole Directed Diffusion conçu pour les RCSFs. Cependant, EAR ne trouve pas un unique chemin optimal qu'il utilise pour toutes les communications, mais garde un ensemble de chemins optimaux et choisit des chemins différents pour différentes communications. Le choix se fait en se basant sur une méthode probabiliste. Afin d'optimiser ces mesures (la durée de vie et la connectivité du réseau), EAR utilise occasionnellement des chemins moins optimaux pour procurer des gains en énergie. Au lieu d'utiliser un seul chemin pour toutes les communications, EAR choisit des chemins différents à différents instants de manière à ce que les nœuds d'un chemin n'épuisent pas leurs énergies et ainsi maintenir la connectivité dans le réseau. EAR est un protocole plat qui s'exécute en trois phases comme suit :

Phase d'initialisation

Appelée aussi phase de propagation des intérêts. Des diffusions locales sont faites afin de trouver des chemins entre la source et la destination ainsi que leurs coûts en énergie. C'est la phase de construction des tables de routage.

La station de base initie la connexion en diffusant un message SetUp dont le champ coût est à 0 à la destination de ses voisins.

$$Cost(N_d) = 0$$

Chaque nœud recevant cette requête, la retransmet seulement à ses voisins proches. La requête est envoyée par le nœud N_i vers le nœud N_j qui satisfait la condition suivante :

$$d(N_i, N_s) \geq d(N_j, N_s)$$

$$d(N_i, N_D) \leq d(N_j, N_D)$$

$d(N_i, N_S)$: Distance entre le nœud N_i et le nœud source qui est la station de base.

$d(N_j, N_D)$: Distance entre les nœuds N_j et N_D qui est le nœud destinataire.

Après la réception de la requête, la métrique de l'énergie du nœud voisin est calculée et ajoutée au cout total du chemin.

$$C_{N_j, N_i} = cost(N_i) + metric(N_j, N_i)$$

Les chemins les plus optimaux en termes de consommation d'énergie sont ajoutés à la table de routage FT_j du nœud N_j :

$$FT_j = \{i | C_{N_i N_j} \leq \alpha(\min C_{N_j N_k})\}$$

Le nœud N_j attribut à chaque voisin dans sa table de routage une probabilité inversement proportionnelle au cout :

$$P_{N_j, N_i} = \frac{1/C_{N_j, N_i}}{\sum_{k \in FT_j} 1/C_{N_j, N_k}}$$

Ainsi, chaque nœud N_j possède un ensemble de voisins qui lui servent de relais vers la destination. N_j calcule le coût moyen pour atteindre la destination :

$$Cost(N_j) = \sum P_{N_j, N_i} C_{N_j, N_i}$$

Le coût moyen calculé est inséré dans le champ cost du message SetUp et envoyé vers le nœud source en suivant le procédé de l'étape 2.

Phase de communication de données

Les données sont envoyées de la source vers la destination en utilisant les informations de la phase précédente concernant les routes. Chaque nœud choisit dans sa table de routage le relais vers la SB ayant la plus grande probabilité jusqu'à ce que les données atteignent leur destination.

Phase de maintenance des routes

Cette phase nécessite un trafic minimal qui consiste en des diffusions périodiques du message setUp de la source vers la destination afin de maintenir les chemins en reprenant les étapes de la phase d'initialisation.

Métrique de l'énergie

La métrique de l'énergie utilisée pour évaluer les chemins est

$$metrique = e_{ij}^{\alpha} R_i^{\beta}$$

Où :

e_{ij} : L'énergie nécessaire pour une transmission ou une réception sur le canal de transmission.

R_i : L'énergie résiduelle du nœud i .

α, β : Paramètres de pondération.

Avantages et inconvénients de *Energy Aware Routing*

Contrairement à certains protocoles tels que Directed diffusion où les données sont envoyées par des routes multiples et l'une d'entre elle est renforcée pour envoyer des données à des débits plus élevés, Energy-Aware Routing (EAR) sélectionne un chemin aléatoirement pour économiser de l'énergie et augmente, ainsi, la survivabilité du réseau.

Cependant, cette approche a besoin de collecter des informations sur les localisations et d'établir un mécanisme d'adressage pour chaque nœud ce qui complique sa structure.

Le protocole EAR hérite des insuffisances du routage plat tel le grand nombre de messages échangés entre les nœuds ce qui impose des limites sur la scalabilité du réseau. Ce trafic important conduit à la saturation rapide du réseau et la mort précoce des nœuds voisins de la station de base. Ces nœuds sont connus sous le nom de hot spots (points chauds) parce qu'ils constituent le dernier saut que tous les paquets empruntent pour atteindre la SB. D'autre part, la construction de chemins alternatifs nécessite plus de ressources en termes d'énergie pour contrôler les chemins et d'espace mémoire pour stocker les tables de routage. En outre, EAR est très sensible à la mobilité des nœuds puisque le déplacement d'un nœud implique le changement des coûts des chemins. Ceci nécessite la reconstruction des tables de routage.

4.3. Nouvelle approche de routage pour les RCSFs hiérarchiques

Dans cette section, nous proposons un nouveau protocole de routage nommé HEAR (Hierarchical Energy Aware Routing Protocole) qui a pour but de minimiser la consommation de l'énergie afin d'augmenter la durée de vie du réseau.

4.3.1. Motivations

Le protocole EAR présenté précédemment permet de prolonger la durée de vie des batteries des nœuds en utilisant un routage multi chemins. Selon cette approche, la route la moins couteuse en termes de consommation d'énergie est sélectionnée pour chaque communication au lieu de choisir une seule route pour toutes les communications. Toutefois, nous avons remarqué que le réseau se sature rapidement par le grand nombre de messages échangés entre les nœuds. Ceci est dû à la topologie plate du réseau où tous les nœuds échangent entre eux les informations concernant les coûts des chemins. Le nombre de messages augmente rapidement avec l'augmentation de la taille du réseau. Sachant que la majorité des applications des RCSFs nécessite le déploiement d'un grand nombre de nœuds,

la structure plate est complètement délaissée de nos jours. Son utilisation est restreinte aux petits réseaux déployés dans des zones de tailles limitées telles les bâtiments. Afin de remédier aux déficiences de l'approche plate, nous avons procédé à la hiérarchisation du protocole.

Le contrôle de la topologie du réseau s'avère une bonne méthode pour faciliter la gestion du réseau et augmenter sa durée de vie et le clustering en est une des plus performantes. L'organisation du réseau en cluster permet d'effectuer des communications locales au niveau de chaque cluster. Chaque nœud envoie ses données vers son CH. Ceci diminue considérablement le trafic dans le réseau. Pour l'élaboration de notre protocole nous sommes orientées vers un clustering distribué où la décision d'élection des CH est faite localement au niveau des nœuds. Au lieu d'utiliser une méthode probabiliste pour l'élection des CH comme est le cas dans les protocoles LEACH [23], HEED [26],...etc. nous avons adopté une méthode basée sur une fonction de poids pondéré selon l'importance des paramètres où chaque nœud s'élit CH si son poids est le plus grand parmi ses voisins.

Les CHs consomment beaucoup plus d'énergie par rapport aux autres nœuds puisqu'ils sont responsables de l'agrégation des données provenant des membres de leurs clusters et de la retransmission des données agrégées vers la SB. Par conséquent, il convient de sélectionner les nœuds ayant plus de ressources pour assurer le rôle de CH. Pour garantir ceci, nous avons utilisé une fonction de poids qui dépend de l'énergie résiduelle et du nombre de voisin d'un nœud. Cette approche nous permet d'éviter la charge causée par les messages de contrôle échangés entre la station de base et les nœuds. Les métriques utilisées pour l'élection des CHs permettent d'avoir des cluster-head distribué de manière équilibrée dans le champ de captage.

L'objectif sous-jacent de n'importe quel protocole de routage est de rendre un réseau utilisable et efficace. Un protocole de routage coordonne les activités des nœuds dans le réseau afin d'atteindre le but global de l'application. Les approches de routage utilisant un seul saut pour communiquer les paquets consomment beaucoup d'énergie pour atteindre la station de base surtout si les nœuds devant lui router les paquets sont très éloignés. Afin d'optimiser la consommation de l'énergie durant cette étape, nous avons repris le principe de routage multi-sauts d'EAR.

Pour diminuer la charge des CH, nous avons, également, introduit la notion de nœuds gateway (passerelles) situés entre les CH. Ceci contribue à la conservation de l'énergie des CH en diminuant la portée des communications.

4.3.2. Hypothèses

- Notre protocole est conçu pour les topologies hiérarchiques à un seul niveau.
- Les capteurs (nœuds) sont supposés être statiques, ce qui est le cas dans de nombreux réseaux RCSFs.
- Le protocole est basé sur le mode réactif pour l'établissement des chemins où les routes sont construites dynamiquement à chaque transmission d'un paquet.

- Les nœuds du réseau envoient leurs données vers une station de base unique et fixe. Cette station de base est caractérisée par une capacité illimitée (calcul, stockage et énergie).
- Le réseau est constitué d'un ensemble de nœuds homogènes et à capacités égales (même énergie initiale, même capacité de stockage et de traitement de données). Tous les nœuds du réseau sont fixes. Cependant ils peuvent avoir des rôles différents :
 - Nœud membre* : il a pour mission de surveiller son environnement et de transmettre les données captées vers son cluster-head.
 - Nœud cluster-head* : il est responsable de l'agrégation des données collectées par ses membres et de leur acheminement vers la station de base. Il participe également au routage des données des autres clusters.
 - Nœud gateway* : sa fonction consiste à assurer la communication entre les CHs.
- Notre solution n'exige aucun système de géo-localisation, donc les capteurs n'ont aucun système de localisation, et ignorent leur emplacement géographique.

4.3.3. Principe de fonctionnement

L'algorithme que nous proposons procède de manière itérative. A chaque itération, une opération de clustering est effectuée de manière distribuée. En premier lieu, la station de base initie le protocole par la diffusion d'un paquet *NEW_ROUND* vers tous les nœuds du réseau. En parallèle, les nœuds envoient entre eux le message *HELLO_MSG* pour la découverte de leur voisinage. A la réception du message *NEW_ROUND*, chaque nœud calcule son poids qui est une combinaison de métriques que nous détaillerons dans le paragraphe suivant. Ce poids est diffusé par la suite vers ses voisins afin de procéder à l'élection des cluster-heads. Une fois les CHs désignés, ils annoncent leur statut pour que les nœuds non cluster-heads puissent les rejoindre. Ces derniers choisissent un cluster-head et l'informent de leur intention de le rejoindre. Les nœuds ayant reçu plusieurs messages d'annonces de la part des nœuds élus CH, changent leurs statuts et se déclarent Gateway pour jouer le rôle de passerelle entre les CHs.

Après le prélèvement des données depuis l'environnement surveillé, les nœuds envoient leurs données vers leurs CH. Chaque CH achemine les informations agrégées de son cluster vers la station de base. Vu que les CHs ont pour mission d'assurer les fonctions les plus coûteuses en énergie, à savoir la communication et les traitements de données, ces derniers relaient les paquets de données vers leurs CHs voisins ou les nœuds gateway au lieu de les envoyer directement vers la SB, et ceci, afin de réduire la consommation d'énergie.

Chaque round de l'algorithme de routage s'exécute en deux phases : une phase d'initialisation et une phase de transmission de données

4.3.3.1. Phase d'initialisation

A l'initialisation, la station de base envoie un message *NEW_ROUND_MSG*. Ces paquets sont diffusés périodiquement par la station de base afin de lancer le round suivant. Tandis que les nœuds membres diffusent un message *HELLO_MSG* pour découvrir leurs voisins, ces derniers paquets sont diffusés une seule fois dans la vie du réseau.

Les paquets échangés à l'initialisation ont la structure suivante :

Champ	Taille (bits)
SinkId	16
Round	16

Tab.4.1 : entête du paquet *NEW_ROUND_MSG*

Champ	Taille
NodeId	16

Tab.4.2 : entête du paquet *HELLO_MSG*

Dès qu'un nœud membre reçoit un message *HELLO_MSG*, il vérifie si le voisin émetteur ne figure pas dans sa table de voisinage. Si c'est le cas, il l'ajoute à la liste de ses voisins.

Après la construction des tables de voisinage, chaque nœud calcule son degré en sommant les entrées de sa table de voisinage. A la réception du message *NEW_ROUND_MSG* de la SB, les nœuds calculent leurs poids qu'ils diffusent à leurs voisins.

Au départ, tous les nœuds sont des nœuds membres, la création des clusters se fait en trois étapes :

Election des cluster-heads

La prise en compte du facteur énergétique lors de la sélection des cluster-heads est d'une importance majeure. D'autres facteurs peuvent être pris en compte afin de procéder à un bon choix des cluster-heads. Pour cela, nous proposons une fonction qui permet de déterminer le poids de chaque nœud. Ce poids est basé sur la combinaison des métriques suivantes :

- Le niveau d'énergie du nœud

Le cluster-head choisi doit avoir un niveau d'énergie élevé afin qu'il puisse exécuter les tâches dont il est responsable. Ainsi, les nœuds dont la réserve d'énergie est élevée auront plus de chances d'être élus cluster-heads.

- Le degré de connectivité du nœud

Le nœud choisi comme cluster-head doit être connecté au plus grand nombre de nœuds possible.

La fonction utilisée pour le calcul du poids *Weight (i)* est définie par :

$$Weight(i) = a \cdot \frac{E_i}{E_{max}} + b \cdot \frac{D_i}{D_{max}}$$

Avec :

E_{max} : Quantité d'énergie maximale du nœud.

E_i : Quantité d'énergie restante du nœud i .

D_{max} : Degré de connectivité maximum du nœud.

D_i : Degré de connectivité du nœud i .

a, b : Constantes de pondération.

Phase d'enchère

Après avoir calculé son poids, le nœud le sauvegarde dans une variable (**myWeight**) et le transmet à ses voisins en diffusant un message *WEIGHT_UPDATE* dont l'entête est la suivante :

Champ	Taille
Id	16
Weight	64

Tab.4.3 : entête du paquet *WEIGHT_UPDATE*

A la réception de ce message, les nœuds mettent à jour leurs tables de voisinage en éditant le poids. La table de voisinage (construite et maintenue durant la création des clusters) contient les champs suivants :

ID	Weight

Tab.4.4 : Structure de la table de voisinage des nœuds

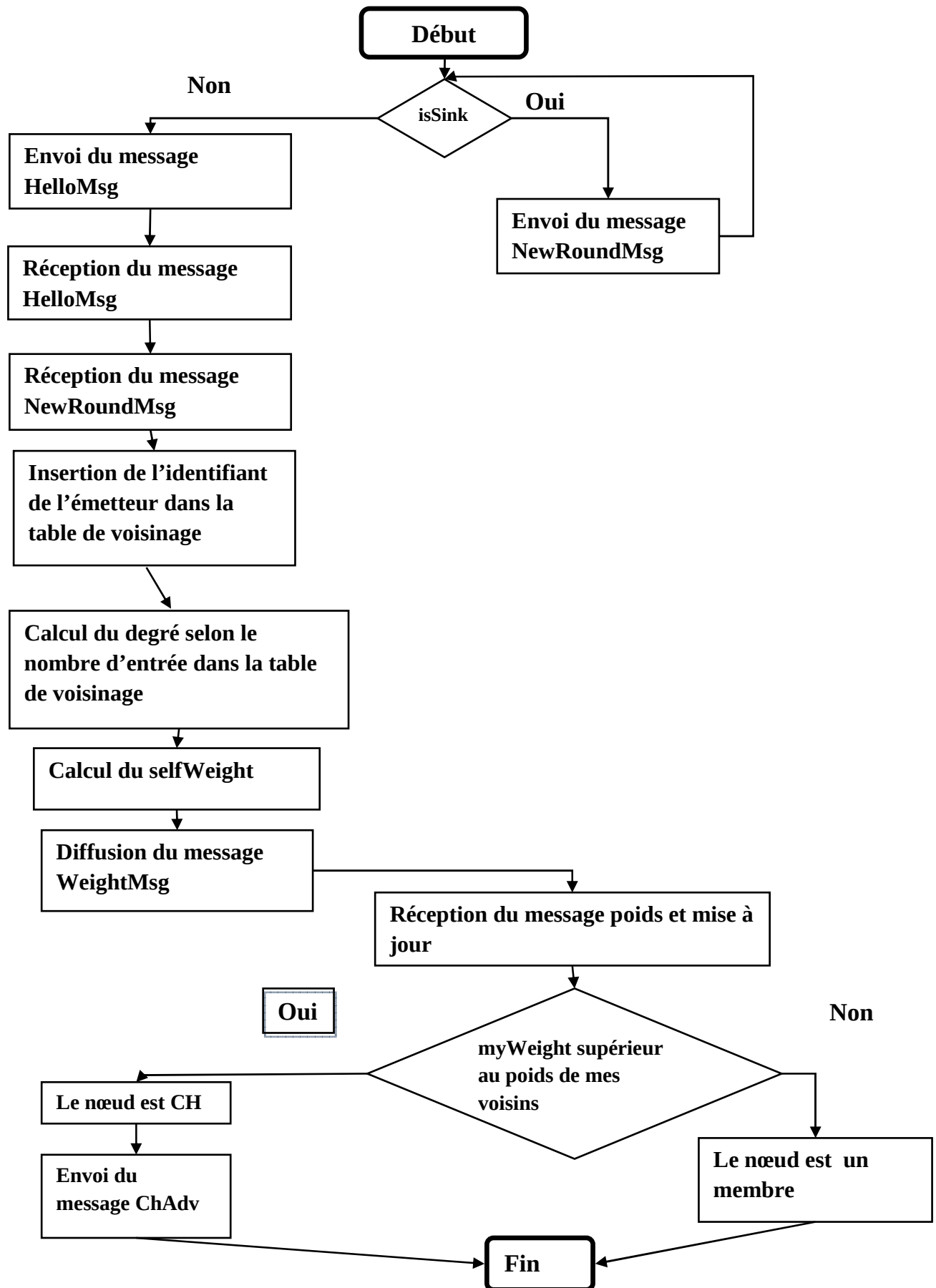


Fig.4.1: organigramme de la phase d'initialisation : élection des cluster-heads

Formation des clusters

Au cours de cette étape, les cluster-heads annoncent leur statut, et les nœuds membres décident du cluster qu'ils vont rejoindre.

- Annonce

Les cluster-heads invitent leurs voisins à rejoindre leur cluster en leur diffusant un message d'annonce (ChAdvMsg) dont l'entête est la suivante :

Champ	Taille
chId	16

Tab.4.5 : Entête d'un message d'annonce (ChAdvMsg)

A la réception d'un message d'annonce, deux cas de figures peuvent se présenter :

- Si le nœud récepteur est un cluster-head, le récepteur ignore ce message.
 - Sinon, le récepteur sauvegarde l'identifiant du cluster-head émetteur ainsi que la puissance (RSSI) avec laquelle il a reçu le paquet. Si un nœud reçoit plusieurs messages d'annonce, les identifiants des cluster-heads seront sauvegardés tout en choisissant le cluster-head dont la puissance est la plus élevée et lui envoie un message d'adhésion.
- Si un nœud reçoit un nombre de messages d'annonce supérieur à un certain seuil, il change son statut en se déclarant *gateway*.

Il se peut qu'un nœud ne reçoive aucun paquet d'annonce, dans ce cas le nœud sera isolé du reste du réseau et il ne pourra ni communiquer ses données ni participer au routage durant le round courant.

- Affiliation

Dans cette étape, chaque nœud membre décide du cluster-head auquel il va appartenir en se basant sur la puissance du signal reçu. Ainsi, le choix se portera sur le cluster-head dont le RSSI est le plus élevé (i.e. le CH le plus proche).

Chaque membre doit donc informer le cluster-head choisi de son intention de le rejoindre en lui transmettant un paquet (JoinClusterMsg) ayant la structure suivante :

Champ	Taille
NodeId	16
chId	16

Tab.4.6: Entête d'un paquet d'affiliation à un cluster.

Les différents champs du paquet d'affiliation à un cluster sont initialisés de la façon suivante :

- Le champ NodeId est initialisé avec l'identifiant du nœud émetteur.
- Le champ chId contient l'identifiant du cluster-head auquel le nœud a décidé de se joindre

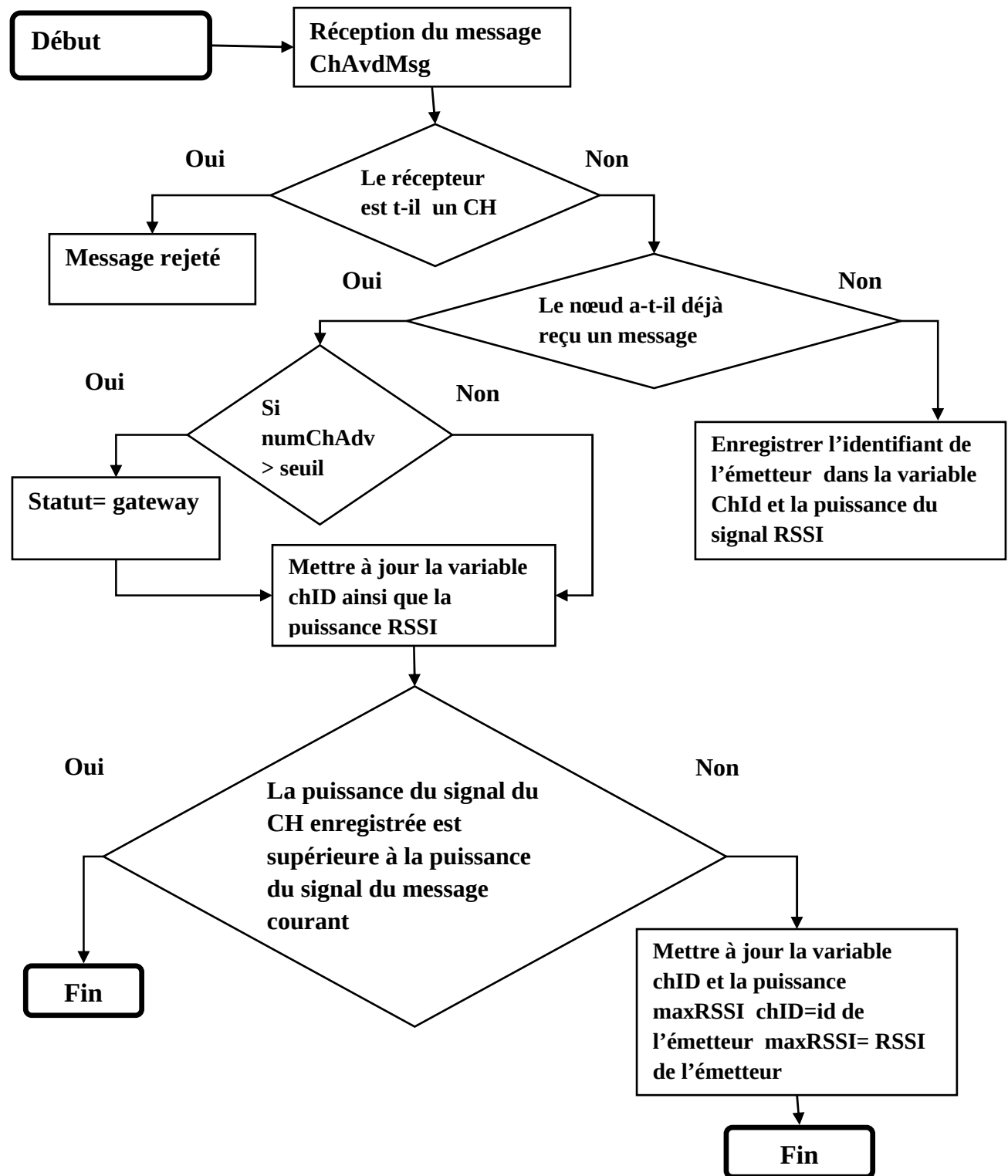


Fig.4.2: Phase d'initialisation : réception d'un message d'annonce.

Construction des tables de routage :

Après l'achèvement du processus de clustering, la station de base envoie un message (SetUpMsg) de cout nul (champ cost = 0). Ces paquets sont diffusés périodiquement par la

SB, à la réception de ce message par les clusters-heads ou les gateway, le champ cost est mis à jour par la fonction suivante :

$$Cost(N_j) = \sum P_{N_j, N_i} C_{N_j, N_i}$$

Où :

P_{N_j, N_i} : Les probabilités calculées pour les CHs voisins et les gateways selon la fonction suivante

$$P_{N_j, N_i} = \frac{1/C_{N_j, N_i}}{\sum_{k \in FT_j} 1/C_{N_j, N_k}}$$

C_{N_j, N_i} : Coût moyen du voisin vers la destination calculé selon la fonction suivante :

$$C_{N_j, N_i} = cost(N_i) + metric(N_j, N_i)$$

La figure suivant résume l'étape de construction des tables de routage :

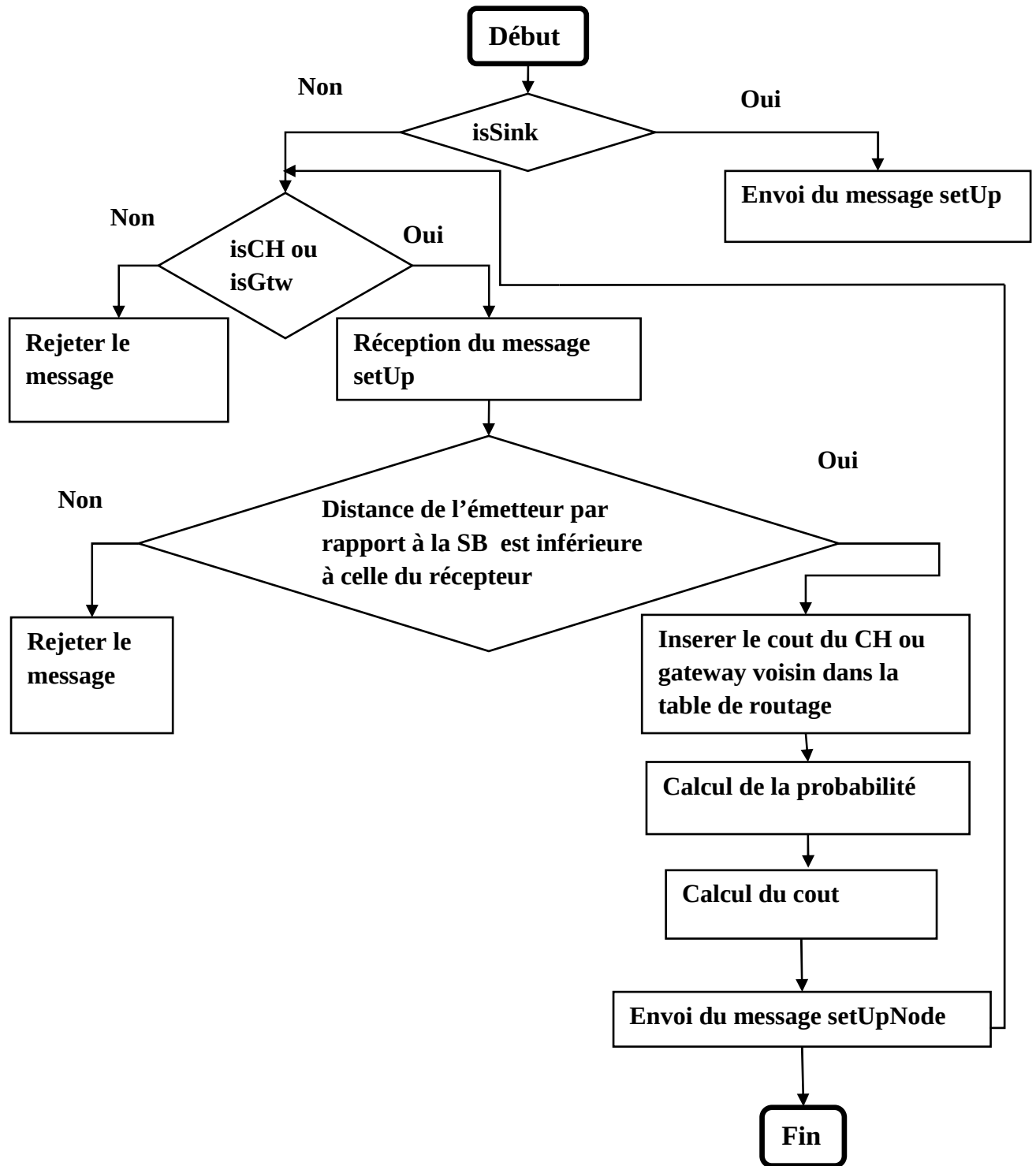


Fig.4.3 : phase de construction des chemins

Le CH qui reçoit ce paquet, le retransmet à ses voisins en incluant son coût. Un CH qui reçoit le paquet (SetUpMsg) relayé, vérifie si le CH émetteur figure dans sa table de routage alors il met à jour son coût, sinon il l'insère et calcule son coût et sa probabilité selon les fonctions précédemment citées. Le tableau suivant représente l'entête du paquet SetUp :

Champ	Taille
NodeId	16
Cost	64
numSeq	16
chDestination	16
distanceToBs	64

Tab.4.7: Entête d'un paquet SetUpMsg.

4.3.3.2. Phase de transmission de données

Le protocole HEAR est basé sur le modèle de livraison de données Time Driven

Nous pouvons distinguer deux types de communications, Intra-clusters et Inter-clusters.

- Communications Intra-clusters

Durant cette phase, les nœuds membre transmettent leurs données vers leurs cluster-heads respectifs.

La structure de l'entête du paquet de données envoyé par le nœud membre vers son cluster-head est la suivante:

Champ	Taille
Source	16
lastHop	16
currentHop	16
nextHop	16
Data	64

Tab.4.8 : Entête d'un paquet de données DataMsg

Après avoir reçu les paquets de données, les CH les sauvegardent puis les agrègent en calculant leur moyenne.

- Communication Inter-clusters

Durant cette phase, les paquets agrégés sont transmis vers la station de base en utilisant un routage multi-sauts. Les CHs créent un nouveau paquet de données et insèrent la donnée agrégée. Le routage s'effectue selon les règles de routage du protocole EAR : chaque CH ou gateway participant à cette opération choisit le prochain saut en prenant le voisin ayant la probabilité maximale dans sa table de routage. L'acquittement est implicite et se fait lorsqu'un CH relai ou gateway retransmet le message.

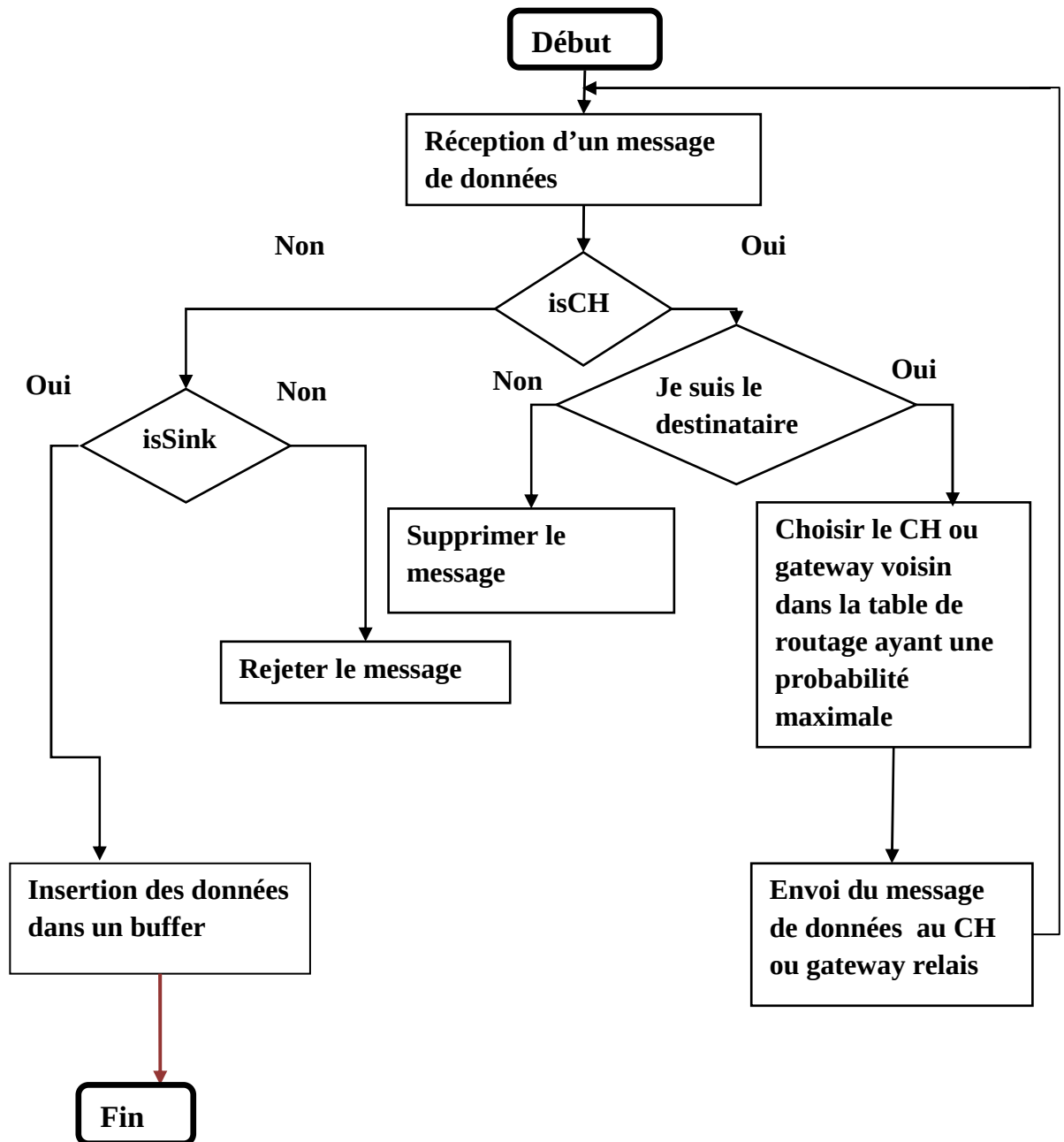


Fig4.4 : Phase de transmission de données

4.4. Discussion sur l'efficacité du protocole

Nous avons constaté qu'en outre les avantages apportés par le protocole EAR, HEAR apporte les améliorations suivantes aux insuffisances d'EAR :

4.4.1. Auto organisation

HEAR, opère d'une manière complètement distribué pour la formation des clusters. Contrairement à la plupart des approches distribuées, notre protocole ne fixe pas le nombre de cluster-head désiré et n'utilise pas les probabilités pour leur élection. Les nœuds décident d'une manière autonome s'ils peuvent devenir cluster-heads ou pas et ceci selon les paramètres de la fonction de coût proposée.

4.4.2. Fonction pondérée d'élection des cluster-heads

Notre protocole prend en compte le niveau d'énergie des nœuds et leur degré de connectivité. En effet, la fonction proposée pour la formation des clusters évite d'avoir recours à des probabilités. La fonction est pondérée selon l'importance des métriques.

4.4.3. Routage multi-sauts au niveau des cluster-heads

Dans les protocoles hiérarchiques qui se basent sur un routage à un seul saut entre les cluster-heads et la station de base, les CHs s'épuisent rapidement car ils peuvent être éloignés de la station de base. Dans notre approche, un schéma de routage multi-sauts est adopté, où les chemins optimaux pour l'acheminement des paquets de données sont choisis selon une probabilité minimisant la dissipation d'énergie. Nous avons également introduit des passerelles entre les CHs. Ces passerelles assurent le rôle de relais afin de diminuer la portée des communications entre les CHs quand ces derniers sont séparés par de grandes distances ce qui permet de conserver plus d'énergie.

4.4.4. Agrégation des données

Les cluster-heads compressent et agrègent les paquets de données envoyés par leurs membres, puis acheminent les paquets résultant de l'agrégation vers la station de base. Ceci permet de réduire le nombre de messages de données transmis, d'optimiser la consommation d'énergie et de rendre le réseau plus scalable.

4.5. Conclusion

Pour un protocole de routage plat, nous avons constaté que les performances du réseau se dégradent à mesure que le réseau est de plus en plus ample. Ceci est dû aux messages de contrôle échangés entre tous les nœuds du réseau afin d'établir des chemins vers la SB, et aussi au fait que toutes les données prélevées dans le champ de captage sont envoyées vers l'utilisateur final, même si ces dernières proviennent de nœuds voisins. Ceci épuise l'énergie des nœuds inutilement et réduit la durée de vie du réseau.

Dans le protocole proposé dans ce chapitre, nous avons adopté une méthode de hiérarchisation du réseau qui permet d'organiser le réseau en cluster de manière distribuée en se basant sur les métriques de l'énergie et du degré de connectivité d'un nœud afin de garantir que les CHs élus sont capables d'accomplir les tâches qui leur seront affectées. Dans le but de procurer un gain important en termes d'énergie et prolonger la durée de vie du réseau, le principe de routage multi-chemins d'EAR a été appliqué sur les CHs. Les CH ayant la plus grande probabilité sont choisis comme prochain relais du paquet de données vers la SB. Afin de mieux balancer la consommation de l'énergie dans le réseau, nous avons introduit la notion de nœuds Gateway.

Le chapitre suivant traite de l'implémentation de notre protocole dans l'environnement de simulation OMNeT++4.1.

Chapitre 5 : mise en œuvre et simulation

5.1. Introduction

Les réseaux de capteurs trouvent leur utilité sur des terrains difficiles d'accès, ce qui fait de leur mise en place une tâche complexe et onéreuse. Il n'est donc pas envisageable, ou du moins extrêmement délicat d'effectuer des modifications sur le réseau d'où la nécessité d'une phase de test avant sa mise en place, et ce afin de minimiser les bogues et les problèmes et assurer la cohérence du réseau tant au niveau fonctionnel que matériel après son déploiement. Pour cela, une solution peu coûteuse qui consiste en la simulation a été proposée.

La simulation de réseaux a été initialement développée pour aider à l'étude, la validation et la création de protocole réseaux. Elle offre un gain en temps et en flexibilité en faisant varier les paramètres de simulation. La simulation des réseaux de capteurs consiste principalement en la reproduction du comportement, du fonctionnement des protocoles de communication des nœuds capteurs dans un environnement informatique. Dans ce chapitre, nous allons commencer par la présentation de quelques exemples de simulateurs ainsi que l'environnement choisi pour l'évaluation de notre protocole et qui est OMNeT++ 4.1 ainsi que le module de simulation implémenté. Par la suite, nous allons présenter les résultats de la simulation du protocole élaboré HEAR ainsi que le protocole de base EAR.

5.2. Exemples de simulateurs de RCSFs

Dans cette section, nous présenterons quelques environnements de simulation utilisés pour évaluer les performances des protocoles et des architectures proposés pour les RCSFs.

5.2.1. NS2

Le simulateur NS [52] désigne la deuxième version du NS (Network Simulator). NS2 est un simulateur à événements discrets qui permet d'exécuter tout type de scénarios sur des topologies définies par l'utilisateur. Il constitue un très bon support pour la description et la simulation des réseaux IP (sans fil ou filaires) avec des protocoles des couches réseau, transport et application. Le réseau est représenté (modélisé) par ses sources de trafic (applications), ses protocoles (UDP, TCP), ses routeurs (avec leurs files d'attente) et les liens qui les relient.

Le simulateur NS2 suit une approche orientée objet et bénéficie de toutes les possibilités qu'offrent les techniques objets comme l'héritage, le polymorphisme, la surcharge, etc. Il est écrit en C++ avec une interface textuelle (ou shell) utilisant le langage OTcl (Object Tool Command Language) qui représente l'extension objet au langage de commande Tcl. Le langage C++ sert à décrire le fonctionnement interne des composants de la simulation (définir les classes). Quant au langage OTcl, il fournit un moyen flexible et puissant de contrôle de la simulation comme le déclenchement d'événements, la configuration du réseau, les caractéristiques des liens physiques, la collecte de statistiques, etc. L'outil NAM (Network Animator) associé au simulateur NS permet de visualiser des animations de la simulation (transfert des paquets d'un nœud à un autre, taille des paquets, remplissage des files d'attentes ...).

L'approche orientée objet sur laquelle est basé le simulateur NS2 constitue son avantage principal et qui permet l'ajout de module à la demande. Toutefois, la dépendance entre les modèles rend difficiles l'ajout de nouveaux modèles.

5.2.2. TOSSIM

TOSSIM [53] est un simulateur discret basé sur la programmation par événements. Il a été conçu afin de simuler les réseaux de capteurs qui utilisent la plateforme TinyOS. Il offre un comportement très proche de ce qui se passe dans le monde réel avec ce type de réseau.

TOSSIM fournit plusieurs modèles de communication. Au niveau radio, deux modèles peuvent être utilisés :

- Le modèle 'simple' propose une communication idéale sans erreurs de transmission ;
- Par contre, le modèle 'lossy' prend en considération les erreurs de transmission et la perte de paquets.

TOSSIM est souvent utilisé avec une interface graphique (TinyViz) pour une meilleure compréhension et visualisation de l'état du réseau. Cette application graphique, réalisée en JAVA, donne un aperçu du réseau de capteurs à tout instant, ainsi que des divers messages émis. Elle permet de déterminer un délai entre les itérations des capteurs afin de permettre une analyse pas à pas du déroulement des actions.

5.2.3. J-Sim

Ce simulateur est codé entièrement en Java. J-Sim repose sur une structure logicielle basée sur les composants appelée *Autonomous Component Architecture* (ACA). Ces composants communiquent les uns les autres via l'envoi/réception de données. Chaque composant est représenté par une classe. Le composant est une entité indépendante qui représente un objet physique (une batterie, un module radio, une couche logicielle,...) ou logique (un protocole de routage, un module de mobilité, un module d'énergie,...etc.). Le code source de ce simulateur est organisé en paquetages relatifs à un type de composant [59]. Ces composants sont connectés à l'aide de ports et la communication des données entre les composants est régie par des contrats. Quand les données arrivent au port du composant, elles sont immédiatement traitées par le composant dans un contexte d'exécution indépendant [61].

La simulation du fonctionnement du réseau exige la définition des composants. Leur mise en relation est réalisée par le TCL. Il s'agit d'un langage de script (complètement supporté par le simulateur J-Sim) qui permet de spécifier l'architecture du réseau et les paramètres de simulation. TCL permet une manipulation plus aisée des classes Java contenues dans les paquetages du J-Sim comme la création d'un objet Java à partir d'une classe Java [60].

Le simulateur j-Sim est relativement récent par rapport aux autres simulateurs, c'est pour cela que tous les protocoles présents dans le simulateur n'ont pas encore été suffisamment validés et testés. Il reste encore quelques erreurs d'implantation. L'autre point faible de ce simulateur est qu'il ne possède pas encore une bibliothèque de modèles très fournie, bien qu'il soit l'un des seuls à disposer d'un modèle très pointu pour la modélisation

des réseaux de capteurs. C'est donc un logiciel qui n'est pas encore mature, mais, dont il faut suivre l'évolution car il importe plusieurs fonctionnalités qui ne sont pas présentes dans les autres outils.

5.2.4. OMNeT++

OMNeT++ [77] (Objective Modular Network Test-bed in C++) est un simulateur open source qui peut être utilisé librement dans un cadre éducatif et académique. Il présente une architecture modulaire et flexible. Grâce à cette structure flexible, il est utilisé pour la modélisation de processus dans plusieurs domaines qui sont essentiellement :

- Modélisation des communications dans les réseaux filaires et sans fil.
- Modélisation de réseaux de file d'attente.
- Modélisation de protocoles.
- Modélisation des systèmes multiprocesseurs.

Sa flexibilité permet à l'utilisateur de l'étendre avec de nouveaux modèles spécifiques aux réseaux de capteurs sans fil. Nous pouvons citer les modèles CASTALIA [75] et SenSim (*Sensor Simulator*) [75] qui sont deux plateformes implémentées sous OMNeT++ simulant le comportement spécifique des réseaux de capteurs.

Le noyau de simulation OMNET ++ utilise le langage C++ qui facilite son intégration avec d'autres applications et environnements de développement. Et pour construire les modèles de simulation: OMNET++ possède une hiérarchie de modules séparés en deux catégories simple et composée. Les modules simples sont programmés en C++, tandis que les modules composés constitués d'un ou de plusieurs modules simples, sont programmés dans le langage de haut niveau NED, sans l'utilisation de scripts, ce qui rend sa configuration pour les différentes simulations très facile. Ces modules peuvent être combinés de différentes manières.

Pour la communication entre les nœuds, les modules implémentés échangent entre eux des messages. OMNET++ donne aux utilisateurs la possibilité d'étudier l'effet du passage à l'échelle, l'architecture du nœud, l'efficacité énergétique, l'architecture de communications, l'architecture du système, les protocoles, etc.

Les simulations par OMNET ++ peuvent être exécutées sous différentes interfaces utilisateurs: interfaces graphiques, interfaces d'animation et les interfaces utilisateurs de commande en ligne qui sont les meilleurs. De plus, il est disponible pour les systèmes d'exploitation les plus courants (Linux, Mac OS/X, Windows).

Le véritable avantage d'OMNeT++ est, sans conteste, son architecture modulaire flexible. Ainsi, un modèle OMNeT++ consiste en un ensemble de modules hiérarchiquement emboîtés communiquant entre eux par échange de messages. Le niveau hiérarchique des modules n'étant pas limité, l'utilisateur est capable de représenter la structure réelle de son système grâce aux modèles de structure d'OMNeT++. Le module de plus haut niveau est appelé le module système ou le module réseau. Un module simple est l'unité élémentaire d'exécution. Un module composé peut contenir un ou plusieurs modules simples et/ou composés.

La vue modulaire nous donne la possibilité d'implémenter des modules indépendants et de les intégrer par la suite dans la même architecture en définissant les interfaces de communication entre les eux.

OMNeT++ présente plusieurs interfaces utilisateur à des fins différentes : démonstration, débogage et exécution par lots. L'utilisateur peut également intégrer du code source, paramétrer, compiler et exécuter des modèles de simulation.

Vu les avantages que présente le simulateur OMNeT++, nous l'avons choisi comme environnement de simulation pour tester et évaluer les performances du protocole proposé HEAR.

5.3. Environnement de simulation

Dans cette section, nous allons présenter l'environnement de simulation.

5.3.1. Modèle de simulation implémenté

OMNet++ est basé sur une architecture modulaire. Afin d'exploiter cette architecture, le réseau est considéré comme un module composé de plusieurs autres modules (*fig5 .1*).

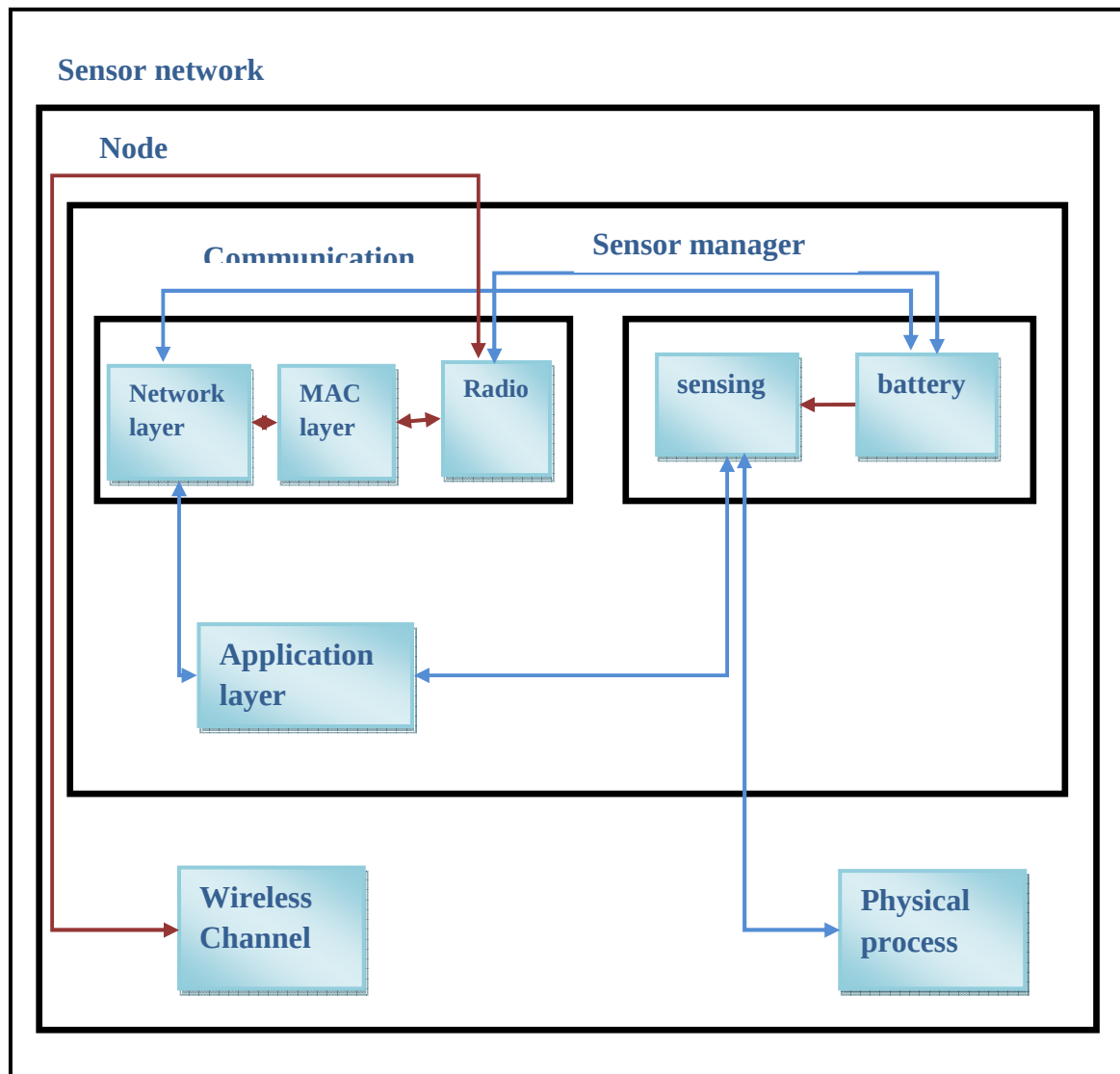


Fig.5.1 Architecture du réseau selon le modèle de simulation implémenté

La plateforme est modélisée par un module composé *SensorNetwork* qui comprend les modules simples *Physical Process* et *Wireless Channel* en plus du module composé *Node*. Le module *physicalProcess* représente les processus physiques et/ou les objets cibles à contrôler (dans notre cas, nous avons considéré un seul phénomène physique) et "Wireless Channel" représentant le canal de transmission sans fil. Le module composé *Node* représente les capteurs et la station de base.

Dans cette architecture, les nœuds ne sont pas directement connectés entre eux mais ils se communiquent via le module simple *WirelessChannel* qui décide du nœud vers lequel le message doit être acheminé selon des informations contenues dans l'entête du message et avec quel puissance ce dernier doit être acheminé. Afin de simuler la capture, tous les nœuds sont connectés au module simple *physicalProcess*.

Les sous modules composant le module *Node* sont les suivants :

- Module Application : Modélise les applications utilisateur exécutées au niveau du capteur. Il permet de traiter l'information capturée avant de la transmettre à la pile de communication.

Ce module a des liens bidirectionnels avec les modules Network et Sensing Device Manager.

- **Module Communication:** Ce module représente les couches de communication situées au dessous de la couche application dans la pile protocolaire et qui sont la couche réseau, la couche Mac et la couche physique.

MacLayer : simule la couche mac dans un capteur. Dans notre application son rôle est résumé au transfert du message reçu de la couche physique vers la couche réseau et le transfert du message à envoyer dans le sens inverse. Nous avons considéré cette couche comme étant parfaite

NetLayer : simule les protocoles de routage implémentés au niveau de la couche réseau d'un capteur. C'est au niveau de cette couche que sera implémenté notre protocole.

Radio : modélise le dispositif caractérisant les propriétés sans fil. Elle représente à la fois la couche physique et l'unité de transmission/réception. Les différents états de la radio (disable et activité) sont également pris en considération. Dans notre application, nous avons considéré les caractéristiques du module radio pour le capteur Mica de Berkeley.

Le module radio du capteur Mica est caractérisé par une portée out-door estimée à 152.4 mètres et un débit de 38.4 bauds. La sensibilité du capteur à la réception est égale à -98dBm

- **Module SensorManager** : Module composé qui représente le plan de gestion de tâches de sensation qui est le module sensing et le module battery manager qui représente le plan de gestion.

batteryManager : il modélise la source d'énergie du capteur. Son rôle consiste à mettre à jour l'énergie disponible au niveau du nœud. Il permet également de déconnecter le nœud du réseau si l'énergie de ce dernier est complètement dissipée.

Sensing : représente le module de capture au niveau d'un nœud. Il est l'intermédiaire entre le module PhysicalProcess et le module Application. Le module sensing génère une donnée chaque seconde et l'envoie vers le module Application.

5.3.2. Type de messages échangés

Ce tableau résume l'ensemble des interactions entre les nœuds du réseau utilisées par le protocole HEAR.

Type de paquet	Taille de l'entête du paquet	Fonction
NewRoundMsg	32	Envoyé par la station de base pour lancer le nouveau round.
HelloMsg	80	Paquet envoyé par tous les nœuds pour découvrir leurs voisins.
WeightUpdate	80	Paquet envoyé par tous les nœuds pour diffuser leur poids.
ChAdv	16	Message d'annonce envoyé par les CH pour informer les autres nœuds de son statut et former son cluster.
JoinClusterMsg	32	Message envoyé par les nœuds vers le CH pour l'informer qu'ils désirent joindre son cluster.
SetUpMsg	176	Initié par la SB et relayé par les CHs pour échanger le coût des chemins vers la SB et construire les tables de routage.
DataMsg	128	Paquet de données pour les transmissions entre les nœuds membre et leurs CHs, inter-cluster et entre les CHs et la SB.

Tab.5.1 : messages échangé dans HEAR

5.4. Résultats de simulation

Afin d'évaluer les performances de notre protocole, nous allons effectuer des tests sur les deux protocoles : EAR et HEAR. Nous comparerons ensuite les résultats obtenus.

5.4.1. Critères de performances

Pour la comparaison entre les trois protocoles, nous prenons en compte les métriques suivantes :

- Moyenne d'énergie consommée (MEC)

Le protocole HEAR vise à optimiser la consommation d'énergie afin de prolonger la durée de vie du réseau. De ce fait, la consommation d'énergie est un paramètre primordial lors de la phase de tests. Pour la comparaison des trois protocoles, nous nous sommes intéressées à l'énergie moyenne consommée au niveau des nœuds. La moyenne de l'énergie consommée est calculée selon la fonction suivante :

$$MEC = \frac{\sum_{i=1}^n E_i}{n}$$

Où :

E_i : l'énergie consommée pour un capteur i . Elle présente la différence entre l'énergie initiale et son énergie résiduelle. Et n : le nombre de capteurs dans le réseau.

- Durée de vie du réseau

L'objectif du développement du protocole HEAR est de prolonger la durée de vie du réseau. Cette dernière est définie par le temps écoulé avant que tous les nœuds du réseau épuisent leur énergie.

- Ecart type de l'énergie consommée

L'écart-type sert à mesurer la dispersion, ou l'étalement, d'un ensemble de valeurs autour de leur moyenne. Dans notre cas, nous avons calculé l'écart type de la moyenne de l'énergie consommée pour évaluer l'équilibrage de la consommation de l'énergie dans le réseau. L'écart type est calculé selon la fonction suivante :

$$EcartType = \sqrt{\frac{\sum_{i=1}^n (MEC - ES)^2}{n}}$$

Où :

MEC : La moyenne de l'énergie consommée.

ES : L'énergie consommée par les nœuds du réseau.

n : Le nombre de nœuds du réseau.

5.4.2. Scénario de simulation

Paramètre	Valeur
Nombre de nœuds	25, 50, 75, 100, 125, 150, 175
Durée de simulation	200s
Taux de paquets généré par seconde	1/s
Nombre de station de base	1
E _{elect}	0.00005 mj
E _{Amp}	0.0000001 mj
E _{Ag}	0.0005mj
Energie initiale	1000 mj
Seuil de réception du signal	-95 dBm
Taux de transmission	19.66 kbps

Tab.5.2 scénario de simulation

5.4.3. Discussion des résultats

Dans cette section, nous allons présenter et analyser les résultats des simulations réalisées suivant les critères de performance décrits précédemment.

Moyenne d'énergie consommée

Le graphe de la figure5.2 représente les résultats de la moyenne d'énergie consommée en fonction du nombre de nœuds du réseau. Nous pouvons remarquer sur le graphe que le protocole hiérarchisé HEAR consomme moins d'énergie en le comparant avec le protocole plat EAR. Ceci est dû au grand nombre de messages échangés par tous les nœuds dans EAR contrairement à HEAR où le nombre de message est considérablement réduit grâce à l'agrégation de données qui s'effectue au niveau des cluster- head. On remarque aussi que l'augmentation des nœuds ne signifie pas forcément une diminution de la consommation d'énergie cela est dû aux différentes puissances utilisés pour les différentes topologies où les nœuds sont déployés de façon aléatoire.

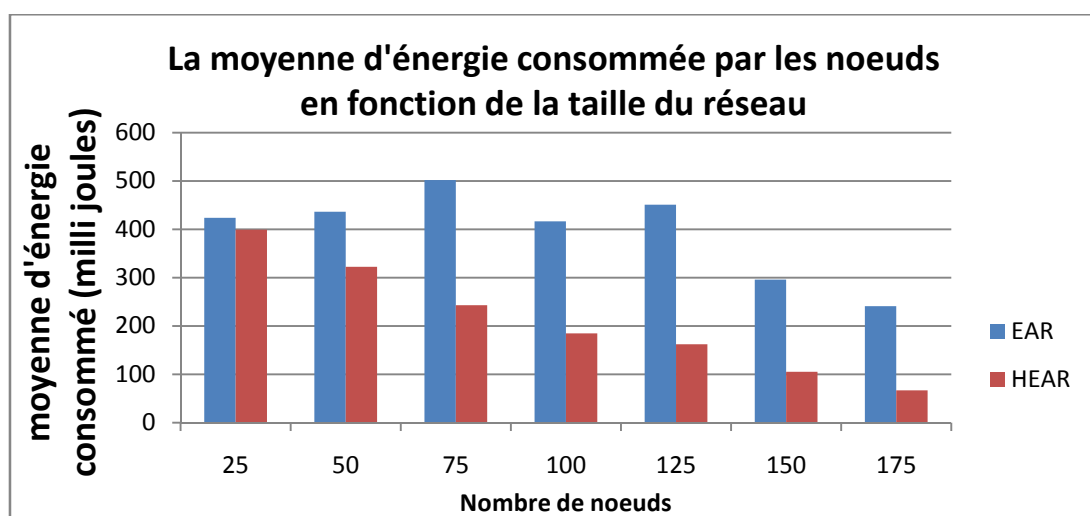


Fig.5.2 moyenne d'énergie consommée

Ecart type de l'énergie consommée

Le graphe de la figure 5.3 représente les résultats de l'écart type en fonction du nombre de nœuds du réseau. Nous constatons à partir de la figure que l'écart type de l'énergie consommée est plus important dans le protocole EAR comparé à HEAR. Ceci revient, d'un côté, aux nœuds proches de la station de base qui consomment plus d'énergie par rapport aux autres nœuds du fait qu'ils servent de relais entre la SB et le reste du réseau, d'un autre côté, à la présence des nœuds isolés qui ne participent pas au routage donc ne consomment pas d'énergie, tandis que dans HEAR, l'énergie est de plus en plus équilibrée en fonction de la densité des nœuds, ceci s'explique par la bonne répartition des tâches sur les différents nœuds (CH, nœuds membres, passerelles).

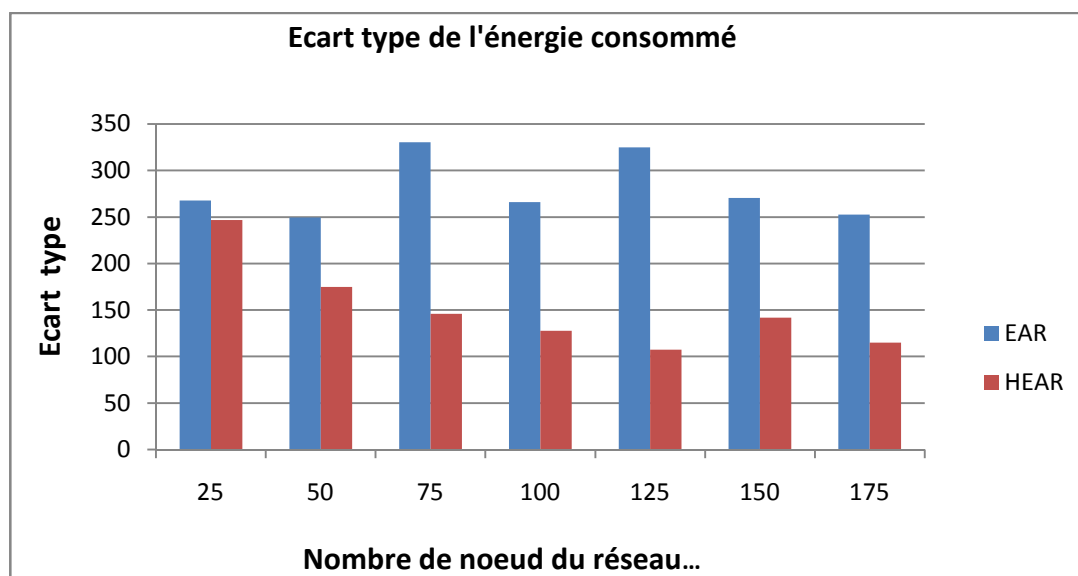


Fig.5.3 Ecart type de l'énergie consommée

Durée de vie du réseau

Pour pouvoir étudier la durée de vie du réseau en fonction du nombre de nœuds déployés nous avons suivi l'évolution de 100 nœuds dans le temps.

La fig.5.4 illustre la durée de vie du réseau au niveau des deux protocoles. Par rapport au protocole EAR, le protocole HEAR présente une amélioration considérable de la durée de vie du réseau. Ceci est dû aux mêmes raisons citées précédemment.

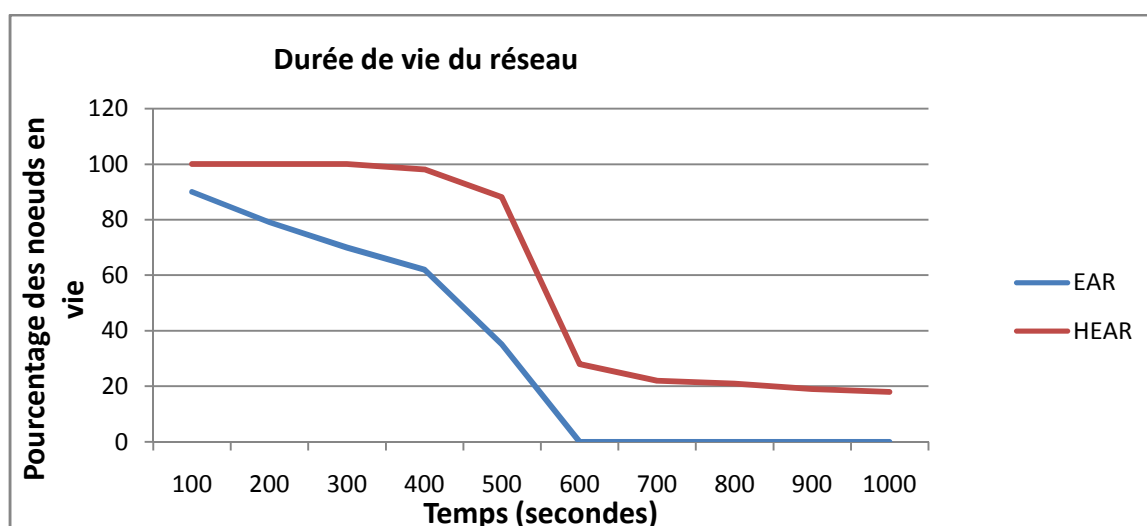


Fig.5.4 durée de vie du réseau

5.5. Conclusion

Dans ce dernier chapitre, nous avons présenté l'environnement de simulation OMNET++4.1, son principe de fonctionnement ainsi que la plateforme de simulation utilisée. Après avoir implémenté notre protocole HEAR et le protocole de base EAR, nous avons effectué une étude comparative entre les deux protocoles suivant différentes métriques. Les résultats des tests effectués sur HEAR et EAR ont montré que le nouveau protocole prolonge la durée de vie du réseau. En effet, grâce au mécanisme de clustering, le nombre de messages échangé dans le réseau a considérablement baissé. Les communications locales au niveau des clusters et l'agrégation de données ont participé à la réduction de l'énergie consommée dans tous le réseau, par conséquent, éliminé le problème des hot spots dans le protocole EAR. L'approche d'élection de CHs adoptée ainsi que l'ajout des nœuds gateway a prouvé son efficacité et a permis de distribuer la consommation de l'énergie dans le réseau ce qui prolonge sa durée de vie.

Conclusion générale et perspectives

La conception des réseaux de capteurs est fortement contrainte par la limitation de la ressource énergétique des nœuds capteurs. Actuellement, la plupart des travaux de recherche sur ce type de réseaux, sont consacrés à la conception des protocoles de routage visant à minimiser l'énergie inhérente aux communications qui sont la source principale de consommation d'énergie afin d'optimiser la durée de fonctionnement du réseau.

Dans cette optique, le routage hiérarchique s'est présenté comme étant une solution prometteuse pour conserver l'énergie des nœuds, et faciliter la transmission des données captées dans le réseau vers la station de base. Selon ce type de routage, les nœuds du réseau sont organisés en clusters gérés par un seul nœud (cluster-head). Ce cluster-head est non seulement responsable de la gestion des nœuds de son cluster, mais aussi l'assurance de la communication des données collectées localement à partir de ses nœuds membres vers la station de base. Cette structure hiérarchique permet d'alléger le trafic du réseau en effectuant des traitements au sein de chaque cluster, avant de faire parvenir les données à la station de base.

La minimisation de la consommation de l'énergie des capteurs et le prolongement de la durée de vie du réseau étaient les objectifs principaux de ce travail. En vu de réaliser ceci, nous avons développé un nouveau protocole de routage hiérarchique qui prend en considération les limitations du protocole de routage EAR sur lequel nous nous sommes basé.

Le nouveau protocole nommé HEAR prend en considération les contraintes imposées par les capteurs tels la basse capacité de stockage et durée de vie limitée, en permettant une gestion plus efficace de la ressource énergétique lors de la communication des données dans le réseau. Par ailleurs, notre protocole assure un équilibrage dans la charge des cluster-heads en termes du nombre de paquets à router dans le réseau. La prise en compte des capacités énergétiques des nœuds cluster-heads et de leur degré de connectivité permet d'optimiser leurs ressources énergétiques, et de mieux distribuer les CHs dans le champ de captage. L'introduction de notion de passerelles a participé à la réduction des charges des nœuds CHs et donc, au prolongement de leurs durées de vie.

L'évaluation des performances de notre protocole a été réalisé par simulation, en utilisant l'environnement OMNeT++4.1. ce dernier représente un outil modulaire d'une grande flexibilité qui facilite la modélisation de n'importe quel système. Les tests effectués sur notre protocole selon différentes métriques ont prouvé son efficacité en termes de conservation de l'énergie des capteurs ce qui a permis d'étendre la durée de vie du réseau. Par ailleurs, les résultats de simulation selon différentes topologies ont montré que le nouveau protocole supporte la scalabilité du réseau sans dégrader ses performances.

Comme perspectives pour notre travail, nous envisageons d'apporter une amélioration sur le mécanisme d'élection des nœuds cluster-heads, en permettant une sélection de nœuds backups (cluster-heads remplaçants ou secondaire). Ces derniers prendront le relais lors de l'épuisement de l'énergie des nœuds choisis comme cluster-heads primaires. Cette technique

permettra de contrôler le fonctionnement cyclique en diminuant le déclenchement périodique du processus de clustering pour l'élection des cluster-heads et la formation des clusters.

Après avoir testé le réseau dans un environnement dense et homogène, nous examinerons les résultats du protocole proposé, dans le cas d'un réseau avec des nœuds capteurs hétérogènes.

Nous avons remarqué lors de l'élaboration de ce travail que le fonctionnement et les performances des protocoles EAR et HEAR seront fortement affectés par la mobilité des nœuds donc inadaptés aux environnements mobiles. Pour ceci, nous avons projeté une autre perspective qui consiste à étudier notre protocole dans un environnement mobile. En effet, pour pouvoir prendre en charge la mobilité de la station de base ou des nœuds capteurs, nous devons modifier le fonctionnement de notre protocole de façon à l'adapter à cet environnement mobile.

Bibliographie

- [01] Claude Servin, « *réseaux et télécommunication* », éditions DUNOD, 2003.
- [02] Guy Pujolle, « *les réseaux* », édition EYROLLES, 2008.
- [03] Nazim AGOULMINE, Jean-Michel CORNU, « *Réseaux ad hoc : du concept aux applications* », Techniques de l'ingénieur, mai 2005.
- [04] John A. Stankovic, Anthony D. Wood, Tian He, “*Realistic Applications for Wireless Sensor Networks*”, Theoretical Aspects of Distributed Computing in Sensor Networks, Monographs in Theoretical Computer Science. An EATCS Series, Springer-Verlag Berlin Heidelberg, 2011.
- [05] T. He, S. Krishnamurthy, L. Luo, T. Yan, L. Gu, R. Stoleru, G. Zhou, Q. Cao, P. Vicaire, J. Stankovic, T. Abdelzaher, J. Hui, and B. Krogh, “*Vigilnet: An integrated sensor network system for energy efficient surveillance*”. ACM Transactions on Sensor Networks, 2(1):1–38, February 2006.
- [06] Hand Alemdar, Cem Ersoy, “*Wireless Sensor Networks for healthcare: A survey*”, Computer Networks, s54, 2688-2710, 2010.
- [07] Claude Castelluccia, Aurelien Francillon, « *protéger les réseaux de capteurs sans fil* », INRIA, 2009.
- [08] Zoran BOJKOVIC, Bojan BAKMAZ, “*A survey on wireless sensor network deployment*”, faculty of transport and traffic engineering, université de Belgrad, , Issue12 volume7, Decembre 2008.
- [09] Sayad Maya, Energy Efficient Protocol (EEP) : « *un protocole de routage efficace en énergie pour réseaux de capteurs sans fil* », mémoire de magister, ESI, 2009.
- [10] Nelson Lee, Philip Levis, Jason Hill, « *Mica High Speed Radio Stack* », 11 Septembre 2002.
- [11] V.Raghunathan, C.Schurghers, S.Park, M.Srivastava, “*Energy-Aware wireless microsensor networks*”, IEEE Signal Processing Magazine (2002) 40-50.
- [12] Qinghua Wang, Ilangko Balasingham, « *Wireless sensor network-An introduction* », departement d’électronique et de télécommunications, université des sciences et technologies, d’Oslo, Norvège, Décembre 2010.
- [13] Yacine Younis, « *minimisation d’énergie dans un réseau de capteurs* », mémoire de magister, UMMTO, 2012.
- [14] Pratyay Kuila, Prasanta K.Jana, “*Energy Efficient Load-balanced clustering algorithm for wireless sensor networks*”, 2nd international conference on communication Security [ICCCS-2012]

- [15] M. Philipose, S. Consolvo, I. Smith, D. Fox, H. Kautz, D. Patterson, “ *Fast, detailed inference of diverse daily human activities*”, Sixth International Conference on Ubiquitous Computing, 2004.
- [16] A. Ouadjaout , A. Bachir , Y. Challal , M. Bagaa , N. Lasla and L. Khelladi : “*Security Essentials for Wireless Sensor Networks*”, IEEE Security & Privacy.
- [17] Sophia Kaprantzis, “*Security Models for Wireless Sensor Networks*”, March 20th, 2006.
- [18] John A. Stankovic, “*wireless Sensor Networks*”, Department of Computer Science, University of Virginia, June 19, 2006.
- [19] C. Karlof and D. Wagner, “*Secure routing in wireless sensor networks: Attacks and countermeasures*”, Ad Hoc Networks, vol. 1, no. 2-3, pp. 293–315, 2003.
- [20] Jamal N. Al-Karaki Ahmed E. Kamal, “*Routing Techniques in Wireless Sensor Networks: A Survey*”, Departement of Electrical and Computer Engineering, Iowa State University.
- [21] S. Lindsey, C. Raghavendra, “*PEGASIS: Power-Efficient Gathering in Sensor Information Systems*”, IEEE Aerospace Conference Proceedings, 2002, Vol. 3, 9-16 pp. 1125-1130.
- [22] N. Bulusu, J. Heidemann, D. Estrin : “*GPS-less low cost outdoor localization for very small devices*”, Technical report 00-729, Computer science department, University of Southern California, Apr. 2000.
- [23] Wendi Rabiner Heinzelman, Anantha Chandrakasan, Hari Balakrishnan: “*Energy-Efficient Communication Protocol for Wireless Microsensor Networks*”, Massachusetts Institute of Technology, Proceedings of the Hawaii International Conference on System Sciences, January 4-7, 2000.
- [24] Wendi B. Heinzelman, Anantha P. Chandrakasan, Hari Balakrishnan: “*An Application-Specific Protocol Architecture for Wireless Microsensor Networks*”, IEEE Transactions on Wireless Communications, vol. 1 no. 4, octobre 2002.
- [25] S.Lindsey et C.S.Raghavendra, “*PEGASIS: Power Efficient Gathering in Sensor Information Systems*”, IEEE Transactions on Parallel and Distributed Systems, vol. 13, n°19, pp. 924-935, septembre 2002.
- [26] Ossama Younis and Sonia Fahmy: “*HEED: A Hybrid, Energy-Efficient, Distributed Clustering Approach for Ad-hoc Sensor Networks*”, Department of Computer Sciences, Purdue University.
- [27] S. Lindsey, S. Raghavendra, “*Data Gathering Algorithms in Sensor Networks Using Energy Metrics*”, IEEE Transactions on parallel and distributed systems, Vol.13, No.9, 2002.

- [28] Jiguo Yu, Yingying Qi, Gang Wang, Qiang Guo, Xin Gu, “*An Energy-Aware Distributed Unequal Clustering Protocol for Wireless Sensor Networks*”, International Journal of Distributed Sensor Networks Volume 2011, Article ID 202145, 8 pages.
- [29] Rahul C. Shah and Jan M. Rabaey: “*Energy Aware Routing for Low Energy Ad Hoc Sensor Networks*”, Wireless Communications and Networking Conference, 2002.
- [30] Pratyay Kuila, Prasanta K. Jana, “*An Energy Balanced Distributed Clustering and Routing Algorithm for Wireless Sensor Networks*”, 2nd IEEE International Conference on Parallel, Distributed and Grid Computing, 2012.
- [31] Zhang Jin-rong, Wang Yue, Wang Dong, Han Jing: “*A clustering and routing algorithm with combination weight for energy consumption balance of WSNs*”, International Conference on Multimedia Information Networking and Security, 2009
- [32] Kazem Sohraby, Daniel Minoli, Taieb Znati, “*Wireless Sensor Networks: technology, Protocols and applications*”, John Wiley & Sons, Inc Publications, Hoboken, New Jersey, USA, 2007.
- [33] P. Rentala, R. Musunuri, G. Shashidha, U. Saxena, “*Survey on Sensor Networks*”, The international Conference on mobile computing and networking, 2001.
- [34] Zhu Yong, Qing Pei, “*An Energy-Efficient Clustering Routing Algorithm Based on Distance and Residual Energy for Wireless Sensor Networks*”, 2012 International Workshop on Information and Electronics Engineering (IWIEE), Elsevier 2012.
- [35] Boubiche Djallel Eddine, « *Protocoles de routage pour les réseaux de capteurs sans fil* », mémoire de magister, université de Batna.
- [36] Alessandra Flammini, Paolo Ferrari, Daniele Marioli, Emiliano Sisinni, Andrea Taroni , “*Wired and wireless sensor networks for industrial applications*”, Microelectronics Journal 40 (2009) 1322–1336.
- [37] Christine Louberry, Philippe Roose, Marc Dalmu, « *Modèle unifié de composants pour l'intégration de capteurs dans les applications multimédias* », IUT de Bayonne pays Basque, 2007.
- [38] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cayirci “*A survey on sensor networks*”, IEEE Communications Magazine, Vol. 40, pp. 102-116, August 2002.
- [39] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. I. Cayirci. “*Wireless sensor networks: A survey*”. IEEE Computer Networks, Vol. 38, No. 4, pp. 393-422, March 2002.
- [40] Bhaskar Krishnamachari, « *Networking wireless sensors* », Cambridge University Press, 2005.

- [41] Jason Hill, Robert Szewczyk, Alec Woo, Seth Hollar, David Culler, Kristofer Pister, “*System Architecture Directions for Networked*”, ACM 0-89791-88-6/97/05, 2000.
- [42] Jason Hill, Robert Szewczyk, Alec Woo, Seth Hollar, David Culler, Kristofer Pister, “*System Architecture Directions for Networked*”, ACM 0-89791-88-6/97/05, 2000.
- [43] Wendi Beth Heinzelman, “*Application specific protocol Architectures for Wireless Networks*”, these de doctorat, institut des technologies de Massachusetts, juin 2000.
- [44] Jun Zheng, Abbas Jamalipour “*Wireless sensor networks: a networking perspective*”, Institute of Electrical and Electronics Engineers, 2009
- [45] Ameer Ahmed Abbasi, Mohamed Younis, “*A survey on clustering algorithms for wireless sensor networks*”, Computer Communications. vol 30, pp 2828-2541, 2007.
- [46] D. J. Dechene, A. El Jardali, M. uccini, A. Sauer, “*A Survey of Clustering Algorithms for Wireless Sensor Networks*”, Project Report, Department of Electrical and Computer Engineering, The University of Western Ontario, Canada, December 2006-2007.
- [47] Sorouh Naeimi, Hamidreza Ghafghazi, Chee-Onn Chow, Hiroshi Ishii, «*A Survey on the Taxonomy of Cluster-Based Routing Protocols for Homogeneous Wireless Sensor Networks* », Journal List Sensor, V 12(6), pp 7350–7409, 2012.
- [48] Chunjuan Wei, Junjie Yang, Yanjie Gao, Zhimei Zhang, «*Cluster-based Routing Protocols in Wireless sensor networks* », International Conference on Computer Science and Network Technology, 2011.
- [49] Jamil Ibriq, Imad Mahgoub, «*Cluster-based Routing in Wireless Sensor Networks: issues and challenges* », Departement of computer science and engineering, Florida Atlantic University, 2004.
- [50] Fuad Bajaber, irfan Awan, «*Adaptive decentalized re-clustering prptocol for wireless sensor networks* », Journal of Computer and System Sciences vol 77, pp282-292, 2011.
- [51] Mark A. Perillo, Wendi B. Heinzelman, «*Wireless sensor network protocols* », Department of Electrical and Computer Engineering University of Rochester, NY, 2005.
- [52] The Network Simulator - ns-2: <http://www.isi.edu/nsnam/ns/>.
- [53] Philip Levis and Nelson Lee, «*TOSSIM: a simulator for TinyOs networks* », university of Berkeley, septembre 2003
- [54] Lyes KHELLADI, Nadjib BADACHE, «*les réseaux de capteurs : état de l’art* », rapport de recherche, laboratoire des logiciels de base, CERIST, Algérie, février 2004.

- [55] Luis Javier García Villalba, Ana Lucila Sandoval Orozco, Alicia Triviño Cabrera, Cláudia Jacy, Barenco Abbas, “*Routing Protocols in Wireless Sensor Networks*”,
- [56] khaled BOUCHAKOUR, « *Routage hiérarchique sur les réseaux de capteurs sans fil: Protocole KhLCH (K-hop Layered Clustering Hierarchy)* », mémoire de magister, ESI, 2012.
- [57] E. Yoneki and J. Bacon, "A survey of wireless sensor network technologies: Research trends and middleware's role", Technical Report, Computer Laboratory, Septembre 2005.
- [58] Ruihua Zhang, Lei Ju, Zhiping Jia, Xin Li, «*Energy Efficient Routing Algorithm for WSNs Via Unequal Clustering*», IEEE 14th International Conference on High Performance Computing and Communications, 2012.
- [59] K. Beydoun, « *Conception d'un protocole de routage hiérarchique pour les réseaux de capteurs* », thèse, université de Franche-Comté, 2009.
- [60] site officiel : <http://sites.google.com/site/jsimofficial/>
- [61] Ahmed Sobeih, Mahesh Viswanathan, Darko Marinov, and Jennifer C. Hou, « *J-Sim: An Integrated Environment for Simulation and Model Checking of Network Protocols* », IEEE Wireless Communications magazine, 2007.
- [62] Naouel Doufene, Hani Hadjammar, « *Routage dans les réseaux de capteurs : optimisation du protocole Directed Diffusion*», Projet de fin d'étude, Institut National de formation en Informatique INI, Algérie, 2006.
- [63] Kemal Akkaya, Mohamed Younis, “*A survey on routing protocols for wireless sensor networks*”, Department of Computer Science and Electrical Engineering, University of Maryland, 2003.
- [64] Philip Levis, David Gay, David Culler, “*Programming Sensor Networks with Application Specific Virtual Machines*”, OSDI, 2004.
- [65] Richard Draves Jitendra Padhye Brian Zill, “*Comparison of Routing Metrics for Static Multi-Hop Wireless Networks*”, Microsoft Research, SIGCOMM, September 2004.
- [66] Soumya Roy and J.J.Garcia-Luna-Aceves, “*Node-Centric Hybrid Routing for Ad Hoc Networks*”, INFOCOM 2002.
- [67] Naourez Mejri, Farouk Kamoun, « *Algorithme de Routage Hiérarchique MHEED à Plusieurs Sauts pour Les Grands Réseaux de Capteurs* », IEEE 4th International Conference: Sciences of Electronic, Technologies of Information and Telecommunications, TUNISIA, March 2007

- [68] Jacques M. Bahi, Abdallah Makhoul, Ahmed Mostefaoui , “*A Mobile Beacon Based Approach for Sensor Network Localization*”, 3rd IEEE International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob 2007)
- [69] Abdelhakim Hamzi, « *Plateforme basée agents pour l'aide à la conception et la simulation des réseaux de capteurs sans fil* », Mémoire de magister, Institut National de formation en Informatique (I.N.I), Algérie, 2007.
- [70] Dazhi Chen and Pramod K. Varshney, “*QoS Support in Wireless Sensor Networks: A Survey*”, Department of EECS, Syracuse University Syracuse, NY, 2005
- [71] Nathalie Mitton, « *Réseaux de capteurs, un état de l'art des algorithmes de routage, Cours* », INRIA, université Lyon1, 2009.
- [72] C. INTANAGONWIWAT, R. GOVINDAN, and D. ESTRIN,” *Directed diffusion : a scalable and robust communication paradigm for sensor networks*”, Proceedings of the ACM/IEEE international conference on Mobile computing and networking, pages 55–67, Boston, USA, Août 2000.
- [73] Chunjuan Wei, Junjie Yang, Yanjie Gao, Zhimei Zhang “*Cluster-based Routing Protocols in Wireless Sensor Networks: A Survey*”,*IEEE* International Conference on Computer Science and Network Technology, 2011.
- [74] C. Mallanda, A. Suri, V. Kunchakarra, S.S. Iyengar, R. Kannan and A. Durresi , « *Simulating Wireless Sensor Networks with OMNeT++* », Sensor Network Research Group, Department of Computer Science, Louisiana State University, 2005.
- [75] A. BOULIS, « *Castalia: A simulator for Wireless Sensor Networks* », Version 1.3, User's Manual, National ICT Australia, January 2008.
- [76] Ian. F. AKYILDIZ, Mehmet Can VURAN, “*Wireless Sensor Networks*”, John Wiley & Sons Ltd, 2010.
- [77] András Varga and openSim ltd, « *OMNeT++, User Manual Version 4.1* », 2010.
- [78] Ananthram Swami, Qing Zhao ,Yao-Win Hong ,Tsing Hua Lang Tong, ”*Wireless Sensor Networks Signal Processing and Communications Perspectives*” ,Willy edition, 2007.
- [79] K. Baumgartner, “*Réseaux de capteurs sans fil*”, IBCOM, Décembre 2005.
- [80] [En ligne] <http://french.bluetooth.com/> dernière consultation : le 28 janvier 2009.
- [81] [En ligne] <http://www.zigbee.org/> dernière consultation : le 28 janvier 2009
- [82] M. Younis, M. Youssef, K. Arisha, “*Energy-aware management in cluster-based sensor networks*”, Computer Networks 43 (5) (2003) 649–668.

