

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique



UNIVERSITE 20 AOUT 55 DE SKIKDA
FACULTE DES SCIENCES DE L'INGENIEUR
DEPARTEMENT d'INFORMATIQUE



Mémoire

Pour l'Obtention du Diplôme de Magistère
en Informatique de l'Ecole Doctorale de l'Est (EDIE)

OPTION: Intelligence Artificielle

Plateforme Basée Agents Pour l'Aide à la Conception et la Simulation des Réseaux de Capteurs Sans Fil

Présenté Par: **BOUDJAADAR Amina**

Devant le Jury:

Pr. BOUFAIDA Mahmoud. PRESIDENT

Dr. MAAMRI Ramdane, Rapporteur

Dr. REDJIMI Mohammed, Examineur

Dr. MAZOUZI Smaine, Examineur

Encadreur: Pr. **SAHNOUN Zaidi**

Rapporteur: Dr. **MAAMRI Ramdane**

Année Universitaire: 2009/2010



Dédicace

Je remercie Dieu de m'avoir donné le courage pour accomplir ce modeste travail que je dédie :

A mes très chers parents qui sont la bougie qui illumine ma vie

A mon mari Samir

A ma fille loudjaine

A mon frère Hamza

A mes sœurs Oumeïma, Nousseïba, Safa

A toute ma grande famille, à tous mes amis et à tous mes enseignants

Et à tout ceux que j'aime et qui m'aiment

BOUDJAADAR AMINA



Remerciements

Tout d'abord, je présente mes remerciements les plus sincères aux initiateurs de ce travail ; Le Professeur Sahnoun Zaidi et Le Professeur Maamri Ramdane.

Je tiens à remercier Le Professeur Sahnoun Zaidi pour son encadrement et pour l'encouragement et l'intérêt qu'il m'a apporté pour l'accomplissement de ce travail et surtout pour son grande aide et ses qualités humaines.

Je tiens à remercier vivement Le Docteur Maamri Ramdane pour son implication totale et son aide importante pendant la rédaction de cette thèse.

Mes remerciements s'adressent aussi au Professeur Boufaïda Mahmoud qui a accepté de présider le jury de soutenance.

Je remercie également messieurs Le docteur Maazouzi smail et Le docteur Redjimi Mohamed qui ont eu un impact majeur sur ma formation d'ingénieur et qui m'honorent aujourd'hui par leurs présences en qualité de membres du jury pour apprécier ce travail.

I. Introduction générale.....	1
II. Introduction sur les réseaux de capteurs sans fil	3
II.1 Introduction.....	3
II.2 Les domaines d'application des WSNs	3
II.2.1 Applications militaires.....	3
II.2.2 Applications environnementales.....	5
II.2.3 Applications de santé.....	6
II.2.4 Applications à la maison.....	7
II.2.5 D'autres applications commerciales.....	7
II.3 Taxonomie des réseaux sans fil.....	8
II.3.1 Selon la zone de couverture.....	8
II.3.1.1 Réseaux personnels sans fil (WPAN).....	8
II.3.1.2 Réseaux locaux sans fil (WLAN).....	13
II.3.1.3 Les réseaux métropolitains sans fil (WMAN).....	14
II.3.1.4 Les réseaux sans fil étendus (WWAN).....	14
II.3.2 Selon l'infrastructure.....	16
II.3.2.1 Réseaux cellulaire (avec infrastructure).....	16
II.3.2.2 Réseaux ad hoc (sans infrastructure).....	18
II.4 Les réseaux de capteurs sans fil.....	19
II.4.1 Définition de capteur.....	19
II.4.2 Architecture physique d'un capteur.....	19
II.4.3 Définition d'un réseau de capteur sans fil	22
II.4.4 Caractéristiques requises des WSNs.....	22
II.4.5 Mécanismes utilisés pour atteindre les caractéristiques requises des WSNs.....	23
II.4.6 Architecture d'un réseau de capteurs.....	24
II.4.6.1 Réseau étoile (point-à-multipoint).....	24
II.4.6.2 Réseau maillé.....	25
II.4.7 Architecture de communication.....	26
II.4.7.1. Couche physique.....	27
II.4.7.2. Couche liaison de données.....	28
II.4.7.3. Couche réseau.....	30
II.4.7.4. Couche de transport.....	30
II.4.7.5. Couche application.....	30
II.4.7.6. Gestion d'énergie.....	31
II.4.7.7. Gestion de mobilité.....	31
II.4.7.8. gestion des tâches.....	31
II.5 Réseaux de capteurs vs réseaux ad hoc.....	32
II.6 Conclusion	33
III. Les systèmes multi-agents.....	34
III.1 Introduction.....	34
III.2 Les systèmes complexes.....	34
III.2.1 Définition.....	34
III.2.2 Caractéristiques d'un système complexe ouvert.....	34
III.3 L'origine du paradigme multi-agent.....	35
III.4 Les agents.....	36
III.4.1 Définitions.....	36
III.4.2 Les caractéristiques multidimensionnelles d'un agent.....	37
III.4.3 Taxonomie des agents.....	38
III.4.3.1 Agents a capacités cognitives ou réactives.....	38
III.4.3.2 Agents communicants ou situés.....	39

III.4.3.3 Classification basée sur la structure de l'agent.....	39
III.5 Les systèmes multi-agents.....	40
III.5.1 Définition	40
III.5.2 Taxonomie des systèmes multi-agents	41
III.5.2.1 Système multi-agents ouvert ou fermé.....	41
III.5.2.2 Système multi-agents homogène/hétérogène.....	41
III.5.3 Les constituants d'un système multi-agents.....	41
III.5.3.1 Les agents (A).....	41
III.5.3.2 L'environnement (E).....	41
III.5.3.3 Les interactions (I).....	42
III.5.3.4 L'organisation (O).....	43
III.5.4 Plateformes pour les systèmes multi agents.....	43
III.5.4.1 AgentTool.....	43
III.5.4.2 AgentBuilder.....	43
III.5.4.3 Jack.....	43
III.5.4.4 DECAF.....	44
III.5.4.5 Jade.....	44
III.5.4.6 JAFMAS et JiVE.....	44
III.5.4.7 Zeus.....	44
III.5.4.8 Madkit.....	45
III.6 Conclusion.....	45
V. La simulation des réseaux de capteurs.....	46
V.1 introduction.....	46
V.2 Différentes approches de test.....	46
V.2.1 Le test en environnement réel.....	46
V.2.2 La simulation.....	46
V.2.3 L'émulation.....	47
V.3 Les simulateurs Actuellement disponibles.....	47
V.4 conclusion.....	50
VI. Les protocoles de communication utilisés dans les WSNs.....	51
VI.1 Introduction.....	51
VI.2 Les protocoles de la couche MAC.....	51
VI.2.1 Les raisons de la perte d'énergie.....	51
VI.2.2 Les modèles de communication dans un WSN.....	52
VI.2.3 Les attributs d'un bon protocole MAC.....	52
VI.3 Les protocoles de routage.....	58
VI.3.1 Les protocoles basés sur la structure du réseau.....	59
VI.3.1.1 Le routage plat.....	60
VI.3.1.2 Le routage hiérarchique.....	65
VI.3.1.3 Le routage basé sur la localisation.....	70
VI.3.2 Les protocoles de routage en fonction de leurs modes opératoires.....	74
VI.3.2.1 Protocoles de routage multichemins.....	74
VI.3.2.2 Le routage basé sur les requêtes.....	75
VI.3.2.3 Routage basé sur la négociation.....	76
VI.3.2.4 Routage basé sur la qualité de service.....	77
VI.3.2.5 Processus cohérents et non cohérents.....	78
VI.4 Les perspectives du routage dans les WSNs.....	79
VI.5 Conclusion.....	80
VII. Conception et réalisation.....	81
VII.1 Introduction.....	81

VII.2 Description générale de l'approche proposée.....	81
VII.3 Les services fournis par la plateforme de simulation.....	81
VII.4 Description détaillée de l'approche proposée.....	82
VII.5 Conception de la plateforme a base d'agent.....	87
VII.5.1 Description d'un agent.....	88
VII.5.2 Description des agents de la plateforme.....	89
VII.5.2.1 Agent objet cible.....	89
VII.5.2.2 Agent capteur.....	92
VII.5.2.3. Agent station de base.....	106
VII.6 Implémentation.....	107
VII.6.1 Environnement de développement.....	107
VII.6.2 Construction des agents de la plateforme de simulation.....	109
VII.6.2.1 Les classes implémenté.....	109
VII.6.2.2 Caractéristiques et méthodes de chaque classe.....	110
VII.6.2.3 Les méthodes principales qui constituent un agent dans la plateforme	
MadKit.....	114
VII.6.2.4.Les agents construits.....	114
VII.6.3 L'exécution de la simulation et l'affichage des statistiques.....	117
VII.7 Quelques interfaces de la plateforme.....	118
VII.8 Conclusion.....	120
VIII. Conclusion générale.....	121

Figure II.1 : Classification des réseaux sans fil.....	8
Figure II.2 : Topologie point à point de Bluetooth.....	9
Figure II.3 : Topologie Piconet de Bluetooth.....	10
Figure II.4 : Topologie Scatternet de Bluetooth.....	10
Figure II.5 : topologies étoile et peer to peer des réseaux IEEE 802.15.4.....	13
Figure II.6 : Le modèle des réseaux mobiles avec infrastructure.....	17
Figure II.7 : Le principe de réutilisation de fréquence.....	18
Figure II.8 : Architecture physique d'un capteur.....	20
Figure II.10 : la topologie étoile des réseaux.....	25
Figure II.11 : la topologie maillée des réseaux.....	26
Figure II.12 : architecture de communication.....	27
Figure II.13 : Energie consommée par les sous-systèmes d'un capteur.....	31
Figure III-1 : illustration graphique simplifiée des systèmes qui relèvent de laComplexité...35	35
Figure III-2 : Fonctionnement d'un agent.....	36
Figure III-3 : Les agents dans un SMA	40
Figure VI.1 : Scénario de transmission de messages pour S-MAC.....	54
Figure VI.2 : le concept du protocole WiseMAC.....	56
Figure VI.3 : Arbre de rassemblement de données et schéma de réveil.....	57
Figure VI.4 : Taxonomie des protocoles de routage.....	60
Figure VI.5 : la différence entre le routage AC et DC.....	61
Figure VI.6 : Exemple pour les zones fixe pour le routage VGAR	69
Figure VI.7 : Un exemple de découpage en zone dans un réseau de capteurs.....	72
Figure VI.8: Transmission géographique récursive dans GEAR.....	74
Figure VII.1 : architecture d'un réseau de capteur sans fil.....	83
Figure VII.2 : architecture d'un nœud capteur.....	84
Figure VII.3 : architecture d'un nœud puits ou station de base.....	85
Figure VII.4 : architecture d'un nœud cible.....	85
Figure VII.5 : Description d'un agent.....	88
Figure VII.6 : architecture de l'agent objet cible.....	90
Figure VII.7: architecture de l'agent capteur.....	93
Figure VII.8 : Etats de transition de l'unité de capture.....	94
Figure VII.9 : Exemple d'une demande.....	95
Figure VII.10 : Exemple d'une réponse par une source.....	95
Figure VII.11 : Propagation des intérêts.....	96
Figure VII.12 : Exemple montrant le backoff et le defering.....	101
Figure VII.13 : état de transition de l'unité de traitement (CPU).....	103
Figure VII.14 : état de transition de la radio.....	104
Figure VII.15 : synthèse des tâches et des données de l'agent capteur.....	106
Figure VII.16 : architecture de l'agent station de base.....	107
Figure VII.17 : Modèle Aalaadin.....	109
Figure VII.18 : shema général de l'agent capteur.....	115
Figure VI-19 : La fenêtre principale.....	118
Figure VI-20 : consommation d'énergie d'un nœud après la simulation.....	119
Figure VI-21 : Evolution de la réserve de batterie des quatre nœuds.....	119

Tableau II.1 : Réseaux de capteurs vs réseaux ad hoc.....	32
Tableau III.1 : Types d'interactions d'après.....	42
Tableau V.1 : plateformes de simulation des réseaux.....	49
Tableau V.2 : plateformes spécialisées dans la simulation des réseaux de capteurs sans fil.....	50
Tableau VII.1 : les Paramètres de simulation de MAC 802.11.....	101

Ce mémoire est dédié à la réalisation d'une plateforme d'aide à la conception et à la simulation des réseaux de capteurs sans fil à base d'agents. Il est constitué de cinq chapitres.

Le premier chapitre est une introduction aux réseaux de capteurs. Il met l'accent sur les domaines d'applications de ces réseaux et établit une typologie, notamment la sous-catégorie des réseaux sans fils. Nous décrivons, aussi, dans ce chapitre l'architecture détaillée d'un capteur, les caractéristiques requises d'un réseau sans fils ainsi que les mécanismes en vigueur pour réaliser ces caractéristiques. Ce chapitre se termine par une taxonomie des réseaux de capteurs, les différentes architectures utilisées et l'architecture de communication en usage.

Le second chapitre aborde les systèmes multi-agents. En premier lieu, nous présentons les spécifications inhérentes à un système complexe, pour mieux justifier dans notre travail l'approche par le paradigme agent, en illustrant, au passage, les caractéristiques multidimensionnelles de l'entité fondamentale d'un SMA et en procédant à une taxonomie multicritères et détaillée des agents et des SMA, ainsi que les constituants d'un SMA et les plateformes célèbres destinées à les concevoir et à les réaliser.

Le troisième chapitre traite les aspects relatifs à la simulation des réseaux de capteurs. Il met en évidence les différentes approches pour tester le fonctionnement d'un réseau. On montre, dans ce chapitre, l'importance cruciale de passer par une phase de simulation lors de la conception des réseaux sans fils et on y expose les différents simulateurs, leurs fonctionnalités et les tâches qu'ils peuvent réaliser.

Le quatrième chapitre s'intéresse aux différents protocoles utilisés, en vue de faire communiquer les capteurs dans un réseau sans fils, qu'il s'agisse de la couche MAC ou de la couche réseau. On y fait aussi une étude comparative entre protocoles.

Le cinquième chapitre décrit le travail de conception et de réalisation de la plateforme. L'évaluation des performances d'un réseau de capteurs sans fils et sa capacité à adopter le meilleur comportement pour répondre de façon idoine aux différentes situations, se fait sur la base d'une simulation, avant tout déploiement dans l'univers réel. L'approche, par le paradigme muti-agents, s'est révélé un choix incontournable pour modéliser les réseaux complexes, en l'occurrence, les réseaux sans fils et ceci grâce à leur caractère distribué et dynamique. Notre contribution consiste à implémenter une plate forme d'aide à la conception et à la simulation des WSN basée agents. Les objectifs assignés à cette plateforme sont la modélisation des différents nœuds du réseau, les architectures et les tests des différents protocoles de communication.

Dans notre approche, le réseau de capteurs sans fils est constitué de trois types de nœuds : Objet cible, capteur et station de base. Chacun de ces trois types possède ses propres caractéristiques et son propre comportement. Ces nœuds interagissent entre eux par le biais d'émissions de messages. L'agent objet cible émet des messages de type « stimulus ». A la réception de ces stimulus, les agents capteurs adopte une démarche collaborative par un routage multi-sauts pour réaliser un objectif commun, celui de la collecte des données pertinentes au niveau des agents stations de base, avec le souci de minimiser l'énergie consommée pour allonger au maximum la durée de vie du réseau. Cette plateforme a été écrite en langage Java en utilisant la plateforme MadKit pour modéliser les agents.

L'exécution de la simulation se fait par étapes :

- On initialise les paramètres des nœuds en faisant renseigner par l'utilisateur les champs relatifs au choix du modèle de propagation du signal capté : acoustique/sismique, le modèle de propagation sans fil, les positions des nœuds, etc..
- On fixe la durée de la simulation.
- On invoque la méthode « activate » qui opérationnalise les agents.

A l'invocation de cette méthode, l'interaction entre agents est déclenchée par l'envoi de messages. Au terme de la période de simulation, des vecteurs sont générés par chaque nœud. Ils contiennent les informations nécessaires à l'analyse des performances du réseau, entre autres, le nombre de paquets transmis/reçus par nœud, l'énergie consommée par chaque nœud et le nombre de paquets capturés. A titre d'exemple, si on constate une surcharge de certains nœuds (un grand nœud de paquets reçus/transmis), entraînant une grande consommation d'énergie, on procède à un changement d'architecture ou à un redéploiement des nœuds pour pallier à cette surcharge et maximiser de la sorte la durée de vie du réseau. Les vecteurs susmentionnés sont consignés dans des fichiers et représentés ensuite graphiquement sous forme d'histogrammes et de courbes pour mieux analyser les performances. Pour notre plateforme, on a choisi d'enregistrer deux paramètres pour permettre aux utilisateurs de mieux analyser les performances de leurs réseaux :

- La densité ou nombre moyen de voisins : Ce paramètre permet d'apprécier l'influence de la densité du réseau sur la consommation énergétique.
- Le nombre de nœuds du réseau : ce paramètre est cruciale pour tester l'évolution des performances en fonction du nombre de nœuds formant le réseau. C'est la propriété dite de scalabilité. Notons enfin, que pour notre plateforme, l'évaluation d'un protocole ou d'une architecture est basée sur l'estimation de l'énergie moyenne consommée par chaque nœud et qui permet alors de prédire la durée de vie du réseau.

II.1 Introduction

Les avancées récentes en technologie des systèmes micro-électro-mécanique (MEMS), des communications sans fil, et de l'électronique numérique ont permis le développement de nœuds capteurs peu coûteux, multifonctionnels et de basse puissance. Ces capteurs sont petits par la taille et communiquent sur des courtes distances. Ces nœuds capteurs minuscules, sont constitués de composants de capture, de traitement de données et de communications. Ils ont influencé l'idée des réseaux de capteurs basés sur la collaboration d'un grand nombre de nœuds. Les réseaux de capteurs représentent une amélioration significative par rapport aux capteurs traditionnels. L'étendue des applications des réseaux de capteurs est vaste, on les retrouve dans le domaine de la santé, de la sécurité et dans le secteur militaire. Les réseaux de capteurs permettent à l'utilisateur une meilleure compréhension de l'environnement. De nos jours, les réseaux de capteurs sans fils font partie intégrante de notre vie. La réalisation des applications mettant en œuvre les réseaux de capteurs sans fil nécessitent l'utilisation des techniques ad-hoc. Dans ce chapitre, on va présenter en premier lieu les domaines d'application des réseaux de capteurs, ce qu'est un réseau sans fils et les différents types de réseaux. Nous aborderons ensuite l'architecture physique d'un capteur ainsi que les couches utilisées dans l'architecture de communication d'un réseau de capteurs sans fil.

II.2 Les domaines d'application des WSNs

Les réseaux de capteurs peuvent inclure différents types de capteurs selon la nature des signaux, à titre d'exemples, taux sismique, thermique, visuel, infrarouge, acoustique et radar, qui peuvent surveiller une grande variété de conditions ambiantes incluant : la température, l'humidité, le mouvement véhiculaire, l'état de foudre, la pression, les niveaux de bruit, la présence ou l'absence de certains genres d'objets, les caractéristiques courantes d'un objet, telles que la vitesse, la direction et la taille.

Les nœuds capteurs peuvent être employés pour la capture continue, la détection d'événements, l'identification d'événements et la commande locale des déclencheurs. Le raccordement sans fils des nœuds de micro-capteur permet un large éventail d'applications, essentiellement dans le domaine militaire et environnemental.

II.2.1 Les applications militaires

Dans les armées modernes, les réseaux de capteurs sans fil constituent une partie intégrante des dispositifs de commande, de contrôle, de communication, de calcul, d'intelligence, de surveillance et de ciblage.

Le déploiement rapide, l'auto-organisation et la tolérance aux fautes sont les caractéristiques majeurs qui font des réseaux de capteurs sans fils une technique vitale pour les aspects militaires désignés par l'acronyme C4ISRT (Military Command, Control, Communications, Computing, Intelligence, Surveillance, Reconnaissance and Targeting systems). Puisque les réseaux de capteurs sont basés sur le déploiement dense et peu coûteux de nœuds de capteurs jetables, la destruction de quelques nœuds par des actions hostiles n'affectera en rien une opération militaire autant que la destruction d'un capteur traditionnel. De ce fait le concept de réseau de capteurs est la meilleure approche pour les champs de bataille. Certaines des applications militaires des réseaux de capteurs sont destinées à la surveillance des forces amicales, de leurs équipements et munitions ; d'autres à la surveillance du champ de bataille, à la reconnaissance des forces et du terrain d'opposition ; enfin d'autres sont destinées à l'évaluation des dommages, à la détection et à la reconnaissance des attaques par des armes de destruction massives.

❖ **Surveillance des forces amicales, des équipements et des munitions :** Les chefs surveillent constamment le statut des troupes amicales, l'état et la disponibilité de l'équipement et des munitions dans un champ de bataille par l'utilisation des réseaux de capteurs. Les équipements et munitions critiques véhiculés par une troupe peuvent être attachés à des capteurs qui rapportent leurs statuts. Ces rapports sont collectés dans des nœuds puits et envoyés aux chefs des troupes. Les données peuvent également être envoyées aux niveaux supérieurs de la hiérarchie de commande tout en étant agrégé avec les données à partir d'autres unités à chaque niveau.

❖ **Surveillance du champ de bataille :** les terrains, les itinéraires d'approche, les chemins et les détroits critiques peuvent être rapidement couverts par un réseau de capteurs et être étroitement observés quand aux activités des forces hostiles. Pendant que les opérations évoluent et que de nouveaux plans opérationnels sont préparés, de nouveaux réseaux de capteurs peuvent être déployés à tout moment pour la surveillance du champ de bataille.

❖ **Reconnaissance des forces et du théâtre d'opérations :** les réseaux de capteurs peuvent être déployés dans des terrains critiques, ce qui permettra la récolte des renseignements précieux et détaillés concernant les forces hostiles sur le terrain en si peu de temps.

❖ **Optimisation :** les réseaux de capteurs peuvent être incorporés aux systèmes de guidage des munitions intelligentes.

❖ **Évaluation de dommages de la bataille** : Juste avant ou après des attaques, des réseaux de capteurs peuvent être déployés dans la zone cible pour collecter les données d'évaluation des dommages de la bataille.

❖ **Détection et reconnaissance d'attaques nucléaires, biologiques et chimiques** : Dans la guerre chimique et biologique, être près du point d'impact est important pour la détection opportune et précise des agents chimiques. Les réseaux de capteurs déployés dans la région amicale sont utilisés comme un système d'alerte chimique ou biologique permettant de fournir aux forces amicales un temps de réaction critique, leurs évitant des dommages. On peut avoir également recours à l'emploi des réseaux de capteurs pour la reconnaissance détaillée après qu'une attaque NBC (Nucléaire, Biologique ou chimique) soit détectée. Par exemple, on peut avoir une reconnaissance nucléaire sans exposer une équipe à la radiation nucléaire.

II.2.2 Applications environnementales

Les applications environnementales des réseaux de capteurs incluent :

❖ l'observation des mouvements des oiseaux, des petits animaux et des insectes.

❖ la surveillance des conditions environnementales qui affectent les récoltes, le bétail et l'irrigation.

❖ Les macro-instruments pour la surveillance à grande échelle de la terre et l'exploration planétaire : comprenant la détection chimique/biologique, le contrôle biologique, la surveillance de l'environnement dans les contextes : marin, terrestre et atmosphérique.

❖ La détection des feux de forêts : en déployant un réseau avec la bonne densité, il devient possible de localiser l'origine exacte du foyer et préparer la riposte pendant que le feu est encore contrôlable.

❖ la recherche météorologique ou géophysique et l'étude de la pollution [1,2,3, 4,5,6,7,8,9,10,11, 12, 13, 14, 11,12,13,14,15].

❖ La détection et la prévention des inondations [16] : Un exemple de détection d'inondation est le système ALERT [17] déployé aux Etats-Unis. Plusieurs types de capteurs déployés dans le système ALERT mesurent les précipitations, le niveau d'eau dans les nappes phréatiques et fournit ces

informations au système centralisé de base de données permettant de réaliser des prédictions quand à l'éventualité d'une inondation.

❖ L'Agriculture de précision : L'avantage qu'apporte les réseaux de capteurs est lié à la capacité de surveiller, en temps réel, le niveau des pesticides, le niveau de l'eau potable, le niveau de l'érosion des sols et le niveau de la pollution atmosphérique en temps réel.

II.2.3 Applications dans le domaine santé :

Les applications dans le domaine de la santé sont nombreuses, elles concernent :

❖ Les interfaces pour les personnes handicapées, la surveillance intégrée des patients, l'automatisation du diagnostic et la surveillance des mouvements.[3,11, 18,19, 14].

❖ La télé-surveillance des données physiologiques humaines : Les données physiologiques rassemblées par les réseaux de capteurs peuvent être stockées pendant une longue période [20], et peuvent être employées pour l'exploration médicale [21]. Les réseaux de capteurs installés peuvent également surveiller et détecter le comportement des vieilles personnes. Ces petits noeuds capteurs permettent au sujet une plus grande liberté de circulation et permettent à des médecins d'identifier très tôt des symptômes prédéfinis [22]. Le "Health Smart Home » a été conçu à la faculté de médecine de Grenoble en France pour valider la faisabilité d'un tel système [18].

❖ L'Observation et la surveillance des médecins et des patients à l'intérieur d'un hôpital : Chaque patient a de petits et légers noeuds de capteurs qui lui sont attachés. Chaque noeud de capteur a sa tâche spécifique. Par exemple, un noeud de capteur peut détecter la fréquence cardiaque tandis qu'un autre détecte la tension artérielle. Les médecins peuvent également porter un noeud de capteur, qui permet à d'autres médecins de les localiser dans l'hôpital.

❖ Administration de drogue dans les hôpitaux : Si des noeuds capteurs peuvent être attachés aux médicaments, la chance d'obtenir et de prescrire le faux médicament aux patients peut être réduite au minimum. Puisque, les patients auront des noeuds capteurs qui identifient leurs allergies et les médicaments exigés. Les systèmes automatisés comme décrit dans [23] ont prouvé qu'ils peuvent aider à réduire au minimum des événements défavorables à l'utilisation des drogues.

II.2.4 Les applications ménagères :

❖ **Automatisation à la maison :** Les avancées technologiques ont permis d'incruster des nœuds capteurs intelligents et des déclencheurs dans des appareils électroménagers, tels les aspirateurs, les fours à micro-ondes et les réfrigérateurs [24]. Ces nœuds capteurs à l'intérieur des dispositifs domestiques peuvent agir l'un sur l'autre et l'un avec l'autre et avec le réseau externe par Internet ou satellite. Ils permettent à des utilisateurs de contrôler les dispositifs à la maison, localement et/ou à distance, plus facilement.

II.2.5 Les applications commerciales

Certaines des applications commerciales réalisent :

❖ La surveillance de la fatigue du matériel et le contrôle de la qualité des produits.

❖ La commande et le guidage des robots dans les environnements de fabrication automatique ainsi que les jouets interactifs, le contrôle des processus et l'automatisation des cycles d'usinage.

❖ La surveillance des secteurs où des désastres ont pu se produire, le diagnostic des machines

❖ Le contrôle de l'environnement dans des bâtiments de bureau : La climatisation et la chaleur de la plupart des bâtiments sont centralement commandées. Par conséquent, la température à l'intérieur d'une salle peut changer par peu de degrés ; une force latérale soit plus chaude que l'autre parce qu'il y a seulement une commande dans la chambre et la circulation d'air du système central n'est pas également distribuée. Un réseau réparti de capteurs sans fil, peut être installé pour commander la circulation d'air et la température dans différentes parties de la salle.

❖ Détection et surveillance des vols de voiture : Des nœuds de capteurs sont déployés pour détecter et identifier des menaces dans une région géographique et pour rapporter ces menaces aux utilisateurs à distance par l'Internet pour l'analyse [25].

❖ Vérification de l'inventaire de gestion : A chaque article dans un entrepôt peut être joint un nœud capteur. Les utilisateurs peuvent découvrir l'endroit exact de l'article et connaître le nombre d'articles dans la même catégorie.

II.3 Taxonomie des réseaux sans fil

Un réseau sans fil (wireless network) est un réseau où au moins deux nœuds peuvent avoir une communication sans liaison filaire. Ces réseaux de communications permettent aux utilisateurs de profiter de tous les services traditionnels des réseaux indépendamment de leurs positions géographiques. Les réseaux sans fil peuvent avoir une classification selon deux critères. Le premier est la zone de couverture du réseau. Au vu de ce critère il existe quatre catégories : les réseaux personnels, les réseaux locaux, les réseaux métropolitains et les réseaux étendus. Le second critère est l'infrastructure ainsi que le modèle adopté. Par rapport à ce critère on peut diviser les réseaux sans fils en : réseaux avec infrastructures et réseaux sans infrastructure, comme on le voit dans l'illustration de la figure II.1.

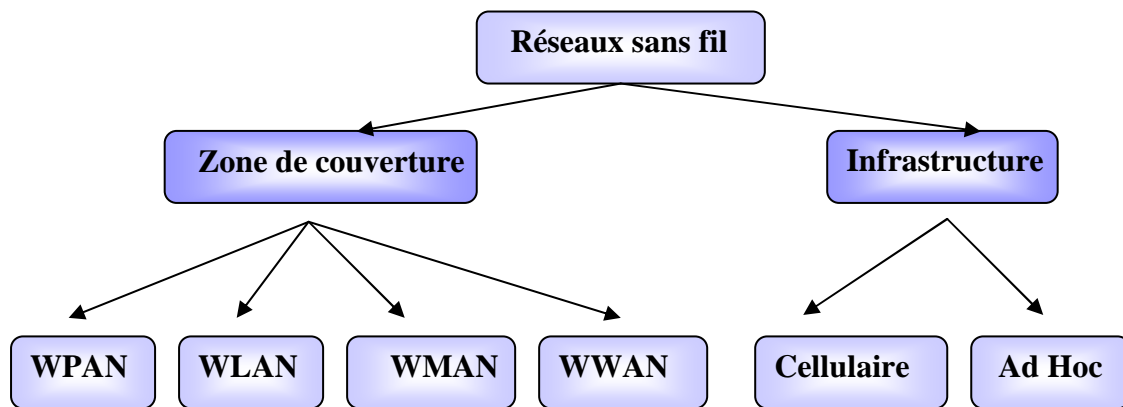


Figure II.1 : Classification des réseaux sans fil

II.3.1 Le critère de la zone de couverture

II.3.1.1 Réseaux personnels sans fil (WPAN)

Les réseaux personnels sans fil ou Wireless Personal Area Network (WPAN), sont des réseaux sans fil à très faible portée, de l'ordre d'une dizaine de mètres. Ils sont le plus souvent utilisés à faire communiquer entre eux des matériels présents sur une personne (par exemple une oreillette et un téléphone portable). Ils sont également utilisés pour relier des équipements informatiques entre eux sans liaison filaire : par exemple pour relier une imprimante ou un PDA (Personal Digital Assistant) à un ordinateur de bureau ou faire communiquer deux machines très peu distantes. Il existe plusieurs technologies permettant la mise en oeuvre de tels réseaux qui sont :

❖ Bluetooth

La principale technologie est IEEE 802.15.1 [26] ou Bluetooth. Elle fut proposée par Ericsson en 1994 et fournit un taux de transmission radio théorique de 1 Mbit/s, pour une portée de 10 mètres (jusqu'à 100 m avec amplificateur). Cette technologie utilise une communication radio, elle propose une pile protocolaire complète, de la couche physique à la couche application de la pile OSI. Bluetooth utilise la bande de fréquences ISM (industrial, scientific and medical) des 2,4 GHz (2,402 GHz à 2,480 GHz) en divisant cette bande en 79 canaux de largeur 1 MHz et met en oeuvre une modulation de type GFSK (Gaussian Frequency Shift Keying) en FHSS. Le débit en bande de base est de 1 Mbits/s et la fréquence nominale des sauts FHSS est de 1600 Hz, soit une période de 625 μ s. Du fait de l'utilisation des sauts de fréquence, Bluetooth impose un fonctionnement en maître/esclave pour le partage du médium radio. Les topologies existantes d'un réseau Bluetooth sont : le « point à point » qui relie un maître à un esclave, comme illustré dans la figure II.2, Piconet qui relie un maître à deux (jusqu'à sept) esclaves actifs, comme illustré dans la figure II.3 ou Scatternet qui relie jusqu'à 10 piconets comme illustré dans la figure II.4. Quelle soit la topologie choisie, toutes les communications de station esclave à station esclave passent par la station maître. Cette norme possède l'avantage d'être très peu gourmande en énergie, ce qui la rend particulièrement intéressante pour être intégrée dans des petits équipements autonomes comme les PDA.

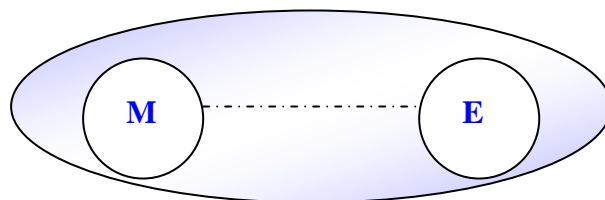


Figure II.2 : Topologie point à point de Bluetooth [60]

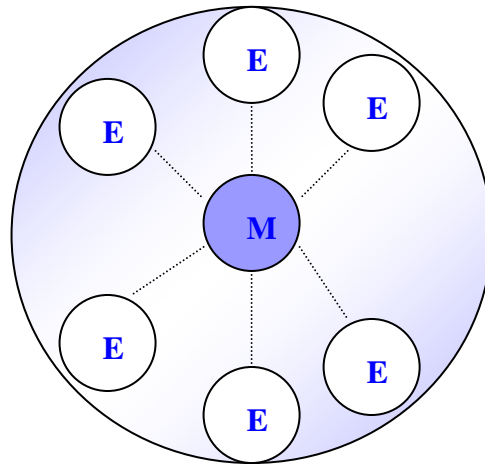


Figure II.3 : Topologie Piconet de Bluetooth [60]

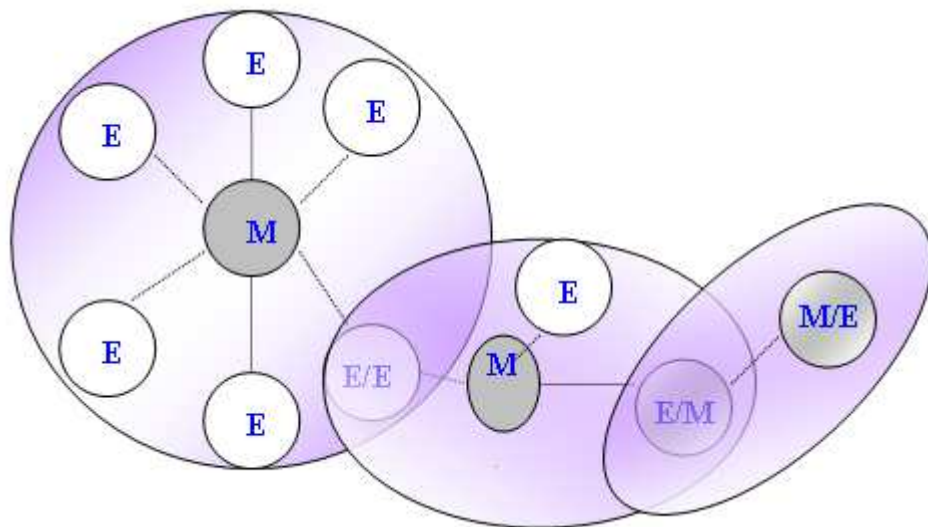


Figure II.4 : Topologie Scatternet de Bluetooth [60]

❖ **HomeRF**

Comme son nom l'indique, HomeRF [27] est une norme de réseau destinée à un usage domestique pour partager un accès à

Internet ou transporter des communications téléphoniques DECT3. Il a été lancé par le consortium industriel formé entre autres par HP, IBM, Siemens, Proxim, Compaq, Intel et Microsoft. HomeRF proposait une couche physique travaillant dans la bande des 2,4 GHz, en FHSS (à 50 sauts par secondes), sur une modulation de type 2-FSK ou 4-FSK. Le débit bande de base est de 1 Mbits/s ou 2 Mbits/s, suivant la modulation utilisée, avec une portée typique de l'ordre d'une cinquantaine de mètres. HomeRF n'a pas connue de succès et a été abandonnée.

❖ IEEE 802.15.4, ZigBee

Le TG15.4 de l'IEEE propose une norme pour les réseaux sans fil personnels à faible consommation énergétique (Low Power-Wireless Personal Area Network, LP-WPAN). Le standard IEEE 802.15.4 [28] propose une norme pour les couches physique et liaison de données, orientée très faible consommation énergétique, qui rend cette technologie bien adaptée à de petits appareils électroniques (appareils électroménagers, hifi, jouets, ...), et plus particulièrement aux réseaux de capteurs. La pile proposée par l'IEEE et la ZigBee Alliance suit les recommandations de l'ISO en terme de séparation des rôles attribués aux différentes couches. Cette pile reprend les couches 1 et 2 normalisées dans la norme 802.15.4 et ajoute ses propres couches supérieures. La couche une, la couche physique, décrit les caractéristiques de l'interface radio, à savoir les fréquences, la largeur de bande, le type de modulation, le débit binaire, etc. La deuxième couche, la couche liaison, décrit les caractéristiques de la sous-couche MAC, c'est-à-dire, la gestion des accès au médium et la sous-couche SSCS relative à la formation des trames et à la convergence des données.

Au niveau de la couche physique, la norme 802.15.4 offre deux options. Les deux couches physiques sont basées sur les techniques d'étalement de spectre par séquence directe (DSSS). La première couche physique opère dans la bande 2,4GHz et offre un débit de 250 kbit/s avec une modulation du type OQPSK (Offset Quadrature Phase Shift Keying). La seconde opère dans les bandes

868/915 MHz et offre un débit de 20 kbit/s pour la bande 868 MHz et 40 kbit/s pour la bande 915 MHz, la modulation utilisée ici est la BPSK (Binary Phase Shift Key). Selon les besoins de l'application, la norme IEEE 802.15.4 prévoit deux topologies : étoile ou point à point (peer to peer), qui sont illustrées dans la figure II.5. Dans la topologie « étoile », les entités RFD sont connectées à un noeud FFD central appelé coordinateur; tous les messages sont relayés par le coordinateur, comme dans un Piconet Bluetooth avec le maître. Les communications directes entre des entités RFD sont impossibles. Notons que le rôle central du coordinateur implique de plus fortes dépenses énergétiques ; un coordinateur devra donc généralement prévoir une source d'alimentation sans contrainte [29]. Dans la topologie point à point (peer-to-peer), un FFD peut communiquer directement avec tout autre FFD, si ils sont à portée radio l'un de l'autre. Dans cette topologie, on retrouve un coordinateur unique comme dans la topologie étoile. Son rôle est de tenir à jour une liste des participants au réseau et de distribuer les adresses courtes.

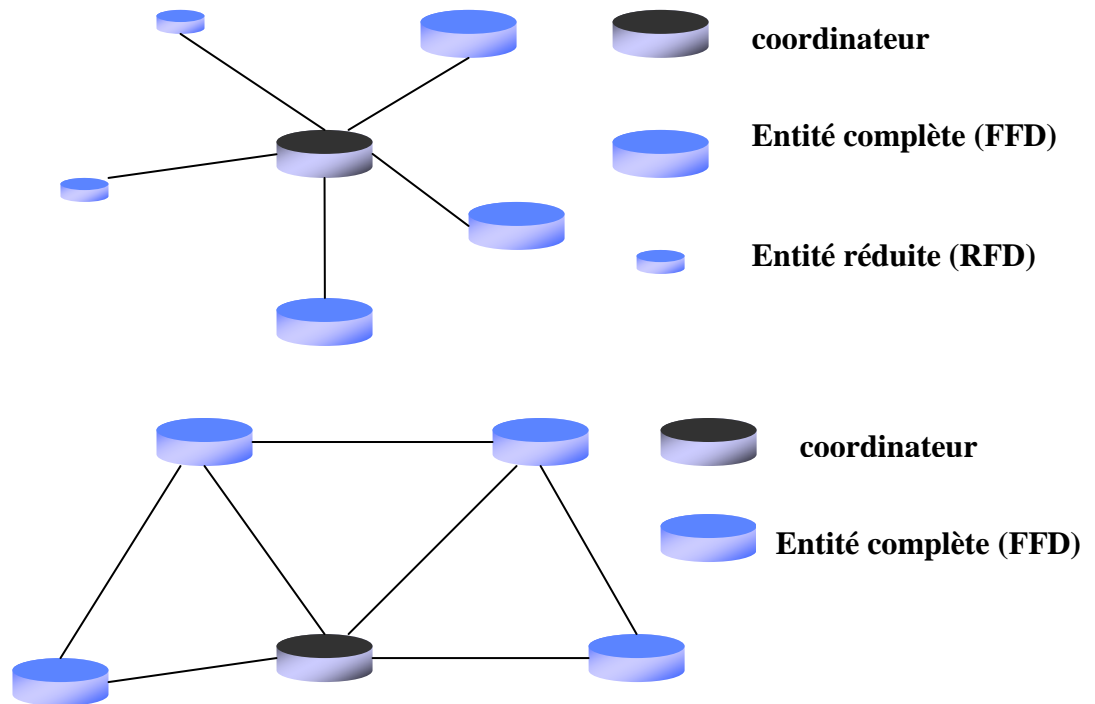


Figure II.5 : topologies étoile et peer to peer des réseaux IEEE 802.15.4 [60]

❖ Liaisons infrarouges

La technologie infrarouge ou IrDA est également utilisée dans ce type de réseaux. Cette technologie est cependant beaucoup plus sensible que Bluetooth aux perturbations lumineuses et nécessite une vision directe entre les éléments souhaitant communiquer, ce qui la limite bien souvent à un usage de type télécommande.

II.3.1.2 Réseaux locaux sans fil (WLAN)

Depuis le développement des normes qui offrent un haut débit, les réseaux locaux sans fil ou Wireless Local Area Network (WLAN) sont généralement utilisés à l'intérieur d'une entreprise, d'une université, mais également chez les particuliers. Ces réseaux sont principalement basés sur les technologies suivantes :

❖ IEEE 802.11, WiFi (Wireless Fidelity)

IEEE 802.11 [30] est un standard de réseau sans fil local proposé par l'organisme de standardisation Américain IEEE. La technologie 802.11 est généralement considérée comme la version sans fil de 802.3 (Ethernet). La technologie 802.11 a connu beaucoup d'évolutions, notamment la 802.11.a et la 802.11b qui proposent une amélioration de la norme initiale en introduisant la modulation CCK (Complementary Code Keying) dans la bande des 2,4 GHz. Deux nouveaux débits sont alors disponibles, 5,5 Mbits/s et 11 Mbits/s sur une portée de quelques dizaines de mètres environ. Le 802.11b est l'amendement de 802.11 qui a donné sa popularité au WiFi. Bien que 802.11b soit encore largement utilisé, il est maintenant supplanté par 802.11g. Ce dernier constitue une amélioration directe de 802.11b avec un débit bande de base de 54 Mbits/s sur la bande des 2,4 GHz.

❖ Hiperlan 1 & 2

Élaboré par l'ETSI (European Telecommunications Standards Institut), Hiperlan est exclusivement une norme européenne. La technologie de Hiperlan exploite la bande de fréquence de 5Ghz et les débits changent selon la version, ainsi : Hiperlan1 apporte un débit de 20 Mbit/s et Hiperlan2 offre un débit de 54 Mbit/s sur une portée d'action semblable dans celui de la Wi-Fi (100 mètres).

II.3.1.3 Les réseaux métropolitains sans fil (WMAN)

Les réseaux métropolitains sans fil ou Wireless Metropolitan Area Network (WMAN) sont aussi connus sous l'appellation de boucle locale radio (BLR). Les réseaux basés sur la technologie IEEE 802.16 ont une portée de l'ordre de quelques dizaines de kilomètres (50km de portée théorique annoncée) et un taux de transmission radio théorique pouvant atteindre 74 Mbit/s pour IEEE 802.16-2004 [31], plus connu sous le nom commercial de WiMAX.

II.3.1.4 Les réseaux sans fil étendus (WWAN)

Les réseaux sans fil (WWAN pour Wireless Wide Area Network) Cette catégorie possède assez peu de technologies à l'heure actuelle. Les seules technologies de WWAN disponibles sont des technologies utilisant les satellites géostationnaires ou en orbite basse

pour relayer l'information entre plusieurs points du globe. Les principales technologies sont les suivantes :

❖ GSM

GSM est l'abréviation de (Global System for Mobile Communication), c'est une norme établie en commun par les opérateurs européens depuis 1982, ayant pour objectif le développement d'un système de téléphonie mobile permettant des communications outre-mer. La communication a lieu par un paquet d'onde ayant deux fréquences : 900 MHz et 1800 MHz. Le GSM se distingue par plusieurs spécificités, la première est l'aspect numérique du réseau, qui offre une qualité supérieure grâce à sa résistance aux interférences. La deuxième spécificité du réseau de GSM réside dans sa configuration cellulaire. Le territoire est subdivisé en petites cellules attachées les unes aux autres. Chaque cellule se voit assigner un certain nombre de canaux permettant les communications.

❖ GPRS

Le GPRS (General Packet Radio Services) est une technologie de radiocommunication par commutation de paquets pour les réseaux de GSM. Les connexions des services de GPRS sont toujours ouvertes afin d'offrir aux utilisateurs des terminaux mobiles une disponibilité de réseau identique à celle qu'ils pourraient atteindre par des réseaux d'entreprise. Le GPRS offre une connectivité d'IP de bout en bout, du terminal GPRS jusqu'à n'importe quel réseau IP. Les terminaux peuvent être intégrés efficacement aux réseaux Internet. La vitesse "utile" sera d'environ 40 Kb/s (vitesse maximum : 171 Kb/s), l'un ou l'autre est quatre fois supérieure à celle du GSM.

❖ UMTS

UMTS L'abréviation de « Universal Mobile Telecommunications System », l'UMTS désigne une nouvelle norme de téléphonie mobile. Le principe de l'UMTS consiste à exploiter une

bande de fréquences plus grande pour faire transmettre plus des données et donc obtenir un débit plus important. En théorie, il peut atteindre 2 Mb/s.

La norme d'UMTS exploite de nouvelles bandes de fréquences situées entre 1900 et 2200 MHz. Cette technologie permet de faire passer des données simultanément et offre alors des débits nettement supérieurs à ceux atteints par le GSM et le GPRS.

II.3.2 Selon l'infrastructure

Les environnements mobiles sont des systèmes composés de sites mobiles et qui permettent à leurs utilisateurs d'accéder à l'information indépendamment de leurs positions géographiques. Les réseaux mobiles ou sans fil, peuvent être classés en deux classes : les réseaux avec infrastructure et les réseaux sans infrastructure.

II.3.2.1 Réseaux cellulaires (avec infrastructure)

Ce type de réseaux se compose des éléments suivants [61] :

- 1- Les "sites fixes" du réseau filaire..
- 2- Les "sites mobiles", réseaux sans fils.

Certains sites fixes, appelés stations de base (SB) sont munis d'une interface de communication sans fil pour la communication directe avec les sites mobiles localisés dans une zone géographique limitée, appelée cellule comme le montre la figure suivante:

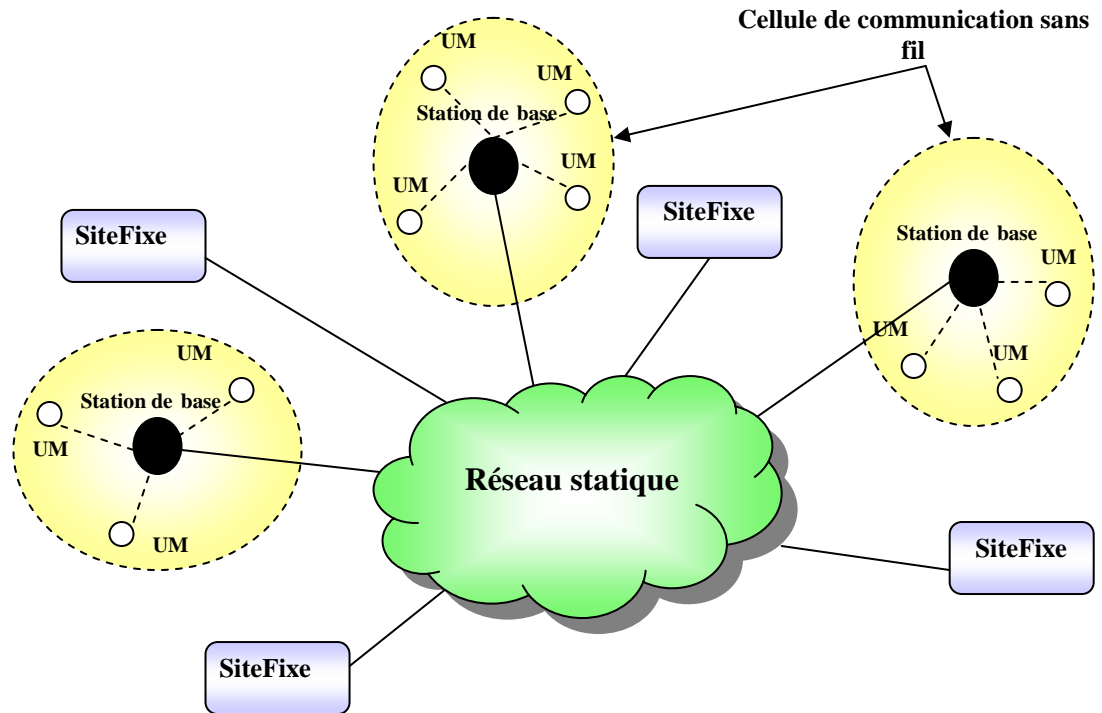


Figure II.6 : Le modèle des réseaux mobiles avec infrastructure [61].

A chaque station de base correspond une cellule à partir de laquelle des unités mobiles peuvent émettre et recevoir des messages. Alors que les sites fixes sont interconnectés entre eux à travers un réseau de communication filaire. Une unité mobile ne peut être, à un instant donné, directement connectée qu'à une seule station de base. Elle peut communiquer avec les autres sites à travers la station à laquelle elle est directement rattachée.

❖ Principe de fonctionnement

La configuration standard d'un système de communication cellulaire est un maillage de cellules hexagonales. Initialement, une région peut être couverte uniquement par une seule cellule, quand la compétition devient importante pour l'allocation des canaux, la cellule est généralement divisée en sept cellules plus petites comme c'est illustré dans la figure II.7. Les cellules adjacentes dans le maillage doivent utiliser des fréquences différentes, contrairement à celles qui sont situées sur les côtés opposés du maillage et qui peuvent utiliser la même fréquence sans risque d'interférence.

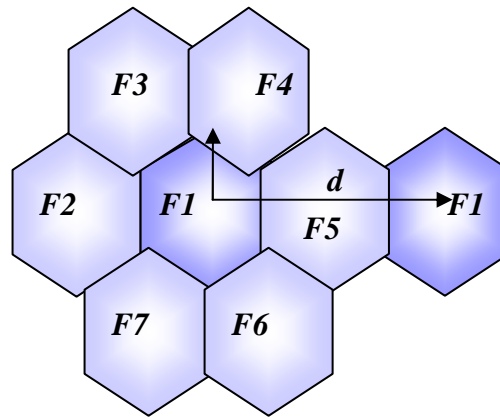


Figure II.7 : Le principe de réutilisation de fréquence [61].

II.3.2.2 Réseaux ad hoc (sans infrastructure)

Dans le modèle de réseau ad hoc ou MANET (Mobile Ad hoc NETwork), l'entité "site fixe" n'existe pas, tous les sites du réseau sont mobiles et communiquent d'une manière directe en utilisant leurs interfaces de communication sans fil. L'absence de l'infrastructure ou du réseau filaire composé des stations de base, oblige les unités mobiles à se comporter comme des routeurs qui participent à la découverte et la maintenance des chemins pour les autres hôtes du réseau. Alors un MANET consiste en une population, relativement dense, d'unités mobiles qui se déplacent dans un territoire quelconque et dont le seul moyen de communication est l'utilisation des interfaces sans fil, sans l'aide d'une infrastructure préexistante ou administration centralisée.

❖ Caractéristiques des réseaux ad hoc

On peut distinguer six grandes caractéristiques [32,33]:

- **La topologie est dynamique :** C'est une conséquence directe de la mobilité des unités qui composent le réseau. Le tracé des routes peut changer pendant l'émission d'un paquet.
- **Les liens sont asymétriques :** En effet, la liaison entre deux unités n'est pas forcément bidirectionnelle.

- **La bande passante est limitée** : car les communications par voies hertziennes imposent un partage du médium de communication entre les différents hôtes. On appelle ce phénomène réutilisation spatiale. Ce phénomène est dû à l'atténuation des signaux avec la distance qui fait que même si un médium peut être utilisé simultanément par plusieurs hôtes, il peut y avoir des collisions et donc la nécessité de procéder à des réémissions.
- **Fortes contraintes énergétiques** : Cela est dû au fait que chacune des unités doit bien souvent embarquer une alimentation autonome.
- **Absence d'infrastructure et de contrôle centralisé** : En effet, les hôtes sont mobiles et c'est à eux qu'il incombe de maintenir les connexions.
- **Les interférences** : Il y a de nombreuses interférences entre les hôtes du réseau ou encore d'une onde avec elle-même (cas d'une réflexion d'onde). Elles accroissent le nombre d'erreurs sur la transmission et imposent un amoindrissement des performances.

II.4 Les réseaux de capteurs sans fil

Les réseaux de capteurs sans fil ou WSN (Wireless Sensor Networks) constituent une catégorie particulière de réseaux ad hoc. Ces derniers sont conçus pour répondre à des problématiques de communications où l'homme est souvent un acteur principal (accès à un réseau global comme Internet, téléphonie, télécommande. . .). Les WSN offrent des moyens de communication très souvent spontanés entre objets autonomes, généralement sans aucune intervention humaine. Il existe des différences considérables entre les réseaux de capteurs et les réseaux ad hoc, donc dans la plus part des cas on ne pourra pas utiliser les mêmes protocoles. Un WSN est quasiment dédié à une unique application.

II.4.1 Définition d'un capteur

C'est un dispositif réagissant à un stimulus comme la chaleur, l'humidité ou la lumière, et générant un signal qui peut être mesuré ou interprété.

II.4.2 Architecture physique d'un capteur

L'implémentation et le développement du nœud capteur sans fil doivent avoir une conception, après une très bonne analyse de l'application. Ceci est obligatoire pour spécifier et définir le profil des éléments (hardware) aussi bien que les caractéristiques et les méthodes (logiciel et modèle de programmation). Ceci offre une meilleure flexibilité et efficacité pendant le fonctionnement. En outre, il doit prendre en compte le fait que les nœuds dans un WSN doivent être petits, énergétiquement efficaces, équipé de capteur(s)

ayant de bonnes performances de calculs et de stockages et des équipements de communication appropriés.

Pendant le choix des éléments matériels pour un nœud capteur sans fil, l'application finale du réseau définit la taille, le coût, et les besoins en énergie. Les éléments de base constituant un nœud capteur sont illustrés dans la figure II.8, selon [34, 35, 36, 37, 38]:

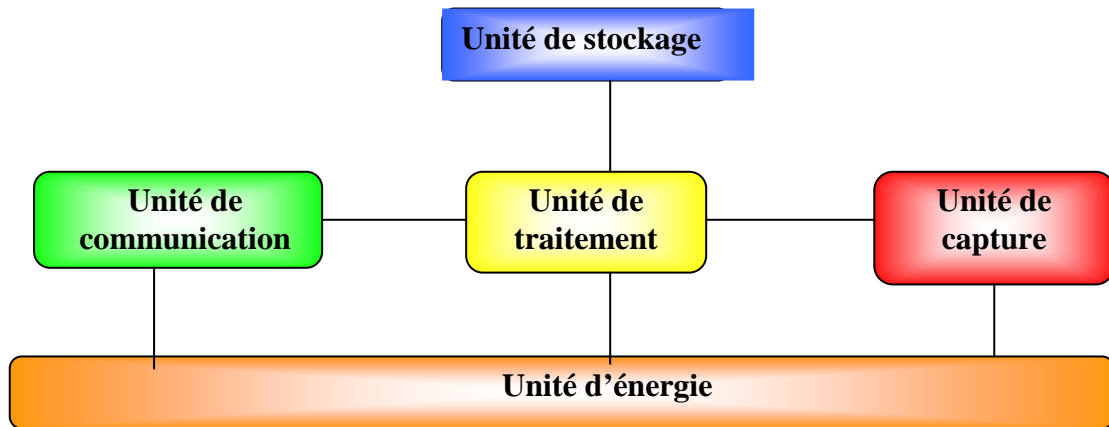


Figure II.8 : Architecture physique d'un capteur

- **Unité de traitement (processeur incorporé) :** Elle est responsable du traitement de toutes les données pertinentes et d'exécuter le code qui décrit le comportement du nœud capteur. Parmi les tâches communes assignées, il y a l'acquisition, le prétraitement et le traitement des informations entrantes et sortantes. Une autre tâche est l'implémentation du routage d'information selon différentes conditions de performance et de fiabilité. Des microcontrôleurs (se composant d'une unité de traitement, d'une mémoire, et de modules périphériques comme ADC, UART, temporisateurs, interfaces connectées via des bus) sont habituellement exploités pour exécuter ce rôle.
- **Unité de stockage (mémoire) :** Elle est responsable du stockage des programmes et des données. Généralement, une mémoire RAM est employée pour stocker les échantillons et/ou les paquets intermédiaires d'un capteur provenant d'autres nœuds. Des mémoires comme EEPROM ou flash sont employées pour stocker le code de programme. La consommation d'énergie par la mémoire

est un facteur important pour la conception appropriée du nœud capteur sans fil.

- **Unité de capture (capteurs) :** C'est l'interface au monde physique. Il y a une grande variété de capteurs en service, mais le choix doit être fait en considérant la taille et la consommation d'énergie exigée pour son fonctionnement. Une catégorisation commune des capteurs est illustrée comme ce suit :

- **Capteurs passifs, omni directionnels:** peut mesurer une quantité physique sans affecter réellement l'environnement. Les exemples de ce genre de capteurs sont les thermomètres, les accéléromètres, les microphones, les capteurs d'humidité et de pression.

- **Capteurs passifs, narrow-beam :** La mesure réalisée par ce type de capteurs dépend de la direction. Les exemples de ces capteurs sont les capteurs ultrasoniques ou les caméras. Ces capteurs exigent une orientation définie afin de mesurer une grandeur du milieu.

- **capteurs actifs :** prennent sans interruption des échantillons de l'environnement ; les exemples sont les sonars, les radars et les capteurs sismiques.

- **unité de communication (émission/réception) :** C'est la partie du matériel qui accomplit les fonctions de gestion du réseau de nœuds de capteurs. Quelques moyens habituels pour la communication sont les fréquences radio, la communication optique, les ultrasons et les ondes électromagnétiques (seulement pour des applications très spécifiques). Les fréquences radio (RF) sont généralement employées parce qu'elles fournissent un large intervalle de transmission et de réception à des débits élevés et avec un taux d'erreur acceptable pour l'énergie nécessaire.

- **unité d'énergie :** Il existe une variété d'options pour l'alimentation en énergie pour un nœud de capteur. L'option la plus commune est l'utilisation des batteries ; d'autres options sont l'énergie de balayage de l'environnement où le nœud de capteur est exposé. L'exemple le plus populaire est: les piles solaires.

- Certaines applications ont besoin de connaître l'emplacement du capteur. Pour cette raison, le capteur doit avoir un système de localisation tels qu'un GPS (Global Positioning System). Pour les réseaux de capteurs mobiles, des noeuds doivent se déplacer, donc un élément dédié à cette mobilité doit exister parmi les composants du capteur [39].

II.4.3 Définition d'un réseau de capteurs sans fil

Un réseau de capteurs sans fil ou WSN (Wireless Sensor Network) est un réseau composé d'un grand nombre de noeuds capteurs, avec une communication sans fil entre ces noeuds. Ces noeuds capteurs sont dispersés dans l'environnement (à surveiller) situé loin de l'utilisateur. Les entités principales qui constituent un réseau de capteurs sont :

- ❖ Les noeuds capteurs qui forment le réseau.
- ❖ La station de base qui communique avec l'utilisateur par l'intermédiaire d'Internet ou par une communication par satellite.
- ❖ Le phénomène faisant l'objet des mesures présentant de l'intérêt pour l'utilisateur.

II.4.4 Caractéristiques requises des WSNs

Un ensemble de caractéristiques sont importantes pour l'accomplissement des tâches assignées aux applications. Les plus importants sont :

- ❖ **Le Type de service :** on s'attend à ce que le WSN offre à l'utilisateur, des informations significatives sur l'objet d'intérêt.
- ❖ **La Qualité de service (QoS) :** C'est une métrique de la qualité de service qui va être offerte par un WSN à ses utilisateurs/applications. Le niveau de QoS est défini par un ensemble d'attributs comme le temps d'attente, la largeur de bande, et la perte de paquets qu'on relie directement avec le type de service du réseau. Dans QoS pour les WSNs, la quantité et la qualité d'information extraites à partir des puits deviennent appropriées.
- ❖ **Tolérance aux fautes :** il est important que le WSN soit capable de traiter l'échec des noeuds capteurs. Une manière reconnue de satisfaire cette contrainte est de réaliser un déploiement redondant des noeuds capteurs.
- ❖ **La durée de vie :** C'est la durée pendant laquelle le réseau reste opérationnel. On s'attend à ce que le WSN puisse fonctionner au moins pendant le temps requis pour accomplir la tâche donnée. Néanmoins, la définition de la durée

de vie dépend de l'application du WSN et elle est en relation directe avec le fonctionnement efficace du réseau.

❖ **Scalabilité :** Cette caractéristique traduit la capacité de maintenir la performance indépendamment de la taille du réseau. Comme un grand nombre de noeuds de capteur peuvent être employés dans les applications de WSN, les architectures et les protocoles doivent fournir le support approprié pour maintenir efficacement les services fournis par le réseau.

❖ **Maintenance :** les changements dans l'environnement du réseau, par exemple, l'apparition de noeuds de capteurs avec des batteries épuisées, exigent une solution permettant l'adaptation et le maintien des services du WSN.

❖ **Programmation flexible :** C'est la capacité des noeuds de capteur à modifier les options de traitement des données acquises et à effectuer des changements et des ajustements de leurs tâches.

II.4.5 Mécanismes utilisés pour atteindre les caractéristiques requises des WSNs :

Afin de réaliser les caractéristiques précédentes, des nouveaux mécanismes de communications, d'architectures et de protocoles doivent être développés. Les mécanismes typiques considérés dans les WSNs sont :

❖ **Communication multi-sauts :**

Pour les communications à longue distance, l'utilisation des noeuds intermédiaires pour transmettre les paquets, peut induire une réduction de l'énergie exigée.

❖ **Utilisation efficace d'énergie :** C'est là un mécanisme clé pour offrir et soutenir sur une durée longue l'opérationnalité du réseau.

❖ **Auto-configuration :** ce mécanisme est employé dans différents aspects des WSNs. Le noeud capteur devrait être capable d'adapter ses paramètres de service pour tenir compte des défaillances des autres noeuds, des obstacles et de l'ajout de noeuds au réseau.

❖ **Collaboration et traitement dans le réseau :**

Selon l'application, il est parfois exigé qu'un groupe de noeuds de capteurs interagissent afin de détecter un événement ou faire un traitement plus complet d'informations. La considération des approches des offres de traitement dans le

réseau comme l'agrégation de données, qui réduisent la quantité de données transmises et par conséquent améliore l'efficacité énergétique du réseau.

❖ **Data-centric** : dans des applications communes, les noeuds de capteurs sont déployés d'une manière redondante, pour protéger le réseau contre des défaillances de noeuds. Dans une approche « data-centric », l'identité d'un noeud particulier fournissant des données devient non pertinente. Le plus important, ce sont les réponses générales que le WSN offre à l'utilisateur.

II.4.6 Architecture d'un réseau de capteurs [59]

Les principales topologies de réseau qui s'appliquent aux réseaux de capteurs sans fil sont décrites dans cette section.

II.4.6.1 Réseau étoile (Point-à-Multipoint)

Un réseau en étoile, comme illustré dans la figure II.10 est une topologie de communications où une station de base peut envoyer et/ou recevoir un message à un certain nombre de noeuds distants. Les noeuds distants peuvent seulement envoyer ou recevoir un message de la station de base, ils ne sont pas autorisés à envoyer des messages entre eux. L'avantage de ce type de réseau pour les réseaux de capteurs sans fil est sa simplicité ainsi que la capacité des noeuds capteurs de maintenir une consommation minimale d'énergie. Cette topologie assure également une basse latence de communication entre le noeud capteur et la station de base. L'inconvénient d'un tel réseau est que la station de base doit être dans la portée de transmission par radio de tous les différents noeuds. Ce qui, malheureusement, diminue la robustesse en raison de la dépendance du réseau à un noeud simple pour contrôler l'ensemble.

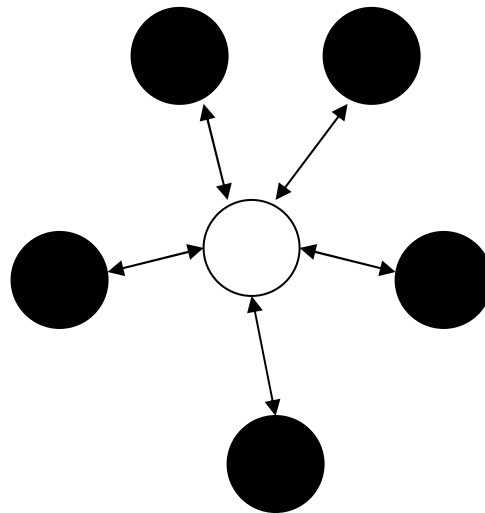


Figure II.10 : la topologie étoile des réseaux [59]

II.4.6.2 Le réseau maillé :

Un réseau maillé comme il est montré dans la figure II.11 est un réseau où n'importe quel nœud peut transmettre à n'importe quel autre à condition qu'il se situe dans sa portée de transmission par radio. Ceci nous conduit aux communications multi sauts, c'est-à-dire, que si un nœud veut envoyer un message à un autre hors de sa portée de communication par radio, il peut utiliser des nœuds intermédiaires pour expédier ce message au nœud désiré. Cette topologie de réseau a l'avantage de la redondance et de la scalabilité. Si un nœud est détruit, un nœud peut encore communiquer à n'importe quel autre nœud dans sa portée, ce dernier qui peut expédier le message à l'endroit désiré. L'inconvénient de ce type de réseau réside dans sa grande consommation d'énergie essentiellement par les nœuds qui implémentent la communication multi sauts par apport aux nœuds qui n'ont pas cette possibilité et qui limite souvent la durée de vie des batteries. Un autre inconvénient est l'augmentation du temps de réponse.

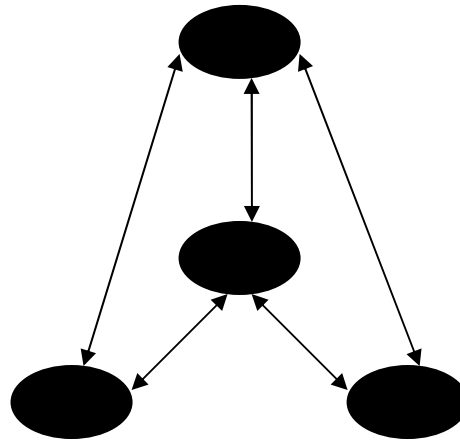


Figure II.11 : la topologie maillée des réseaux [59]

II.4.7 Architecture de communication

Comme tout les types de réseaux, les WSNs utilisent une architecture de communication en couches, ce sont les cinq premières couches du modèle OSI ; la couche physique, la couche liaison de données, la couche réseau, la couche transport et la couche application. Chaque couche a son propre rôle et ses propres protocoles pour atteindre son objectif. Nonobstant le fait que l'objectif d'un WSN n'est pas la communication elle-même, qu'il est soumis à de fortes contraintes énergétiques, par voie de conséquence, d'autres unités doivent lui être ajoutées afin de gérer la consommation d'énergie, la mobilité des nœuds et l'ordonnancement des tâches comme c'est illustré par la figure II.12.

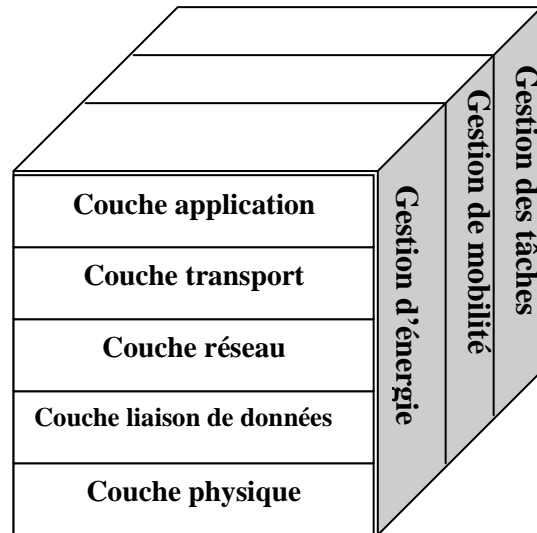


Figure II.12 : architecture de communication [39]

II.4.7.1. Couche physique :

La couche physique est responsable du choix de la fréquence, de la génération de la fréquence porteuse, de la détection du signal, de la modulation et du chiffage des données. Dans un réseau de capteurs multi-sauts, les noeuds communicants sont liés par un médium sans fil. Ces liens peuvent être constitués par les ondes radio ou des signaux infrarouges.

❖ Les signaux infrarouges :

La communication infrarouge est robuste et très peu sensible au problème d'interférence qu'on rencontre dans les dispositifs électriques. Les émetteurs récepteurs basés sur l'utilisation de l'infrarouge sont peu coûteux et plus faciles à construire. Plusieurs ordinateurs portables, PDAs et téléphones mobiles offrent une interface infrarouge d'échange de données. L'inconvénient principal cependant, est la condition d'une ligne de vue entre l'émetteur et le récepteur.

❖ Les radiofréquences :

Le principe est d'émettre des ondes électromagnétiques qui constituent la porteuse du signal à transmettre. Le spectre radio est découpé en bandes de fréquences divisées en canaux. Aujourd'hui, on peut utiliser les bandes ISM (Industrial, Scientific and Medical) pour les réseaux sans fil. Ces bandes sont gratuites mais avec des contraintes

sur le taux d'utilisation (rapport entre le temps d'émission et le temps total) et en puissance d'émission. Afin de limiter la puissance d'émission tout en ayant un débit efficace, des techniques d'étalement de spectre sont obligatoires dans certaines bandes.

II.4.7.2. Couche liaison de données

Beaucoup de recherches dans le domaine des réseaux de capteurs entrent dans le cadre de l'optimisation de la couche de liaison. Elle manipule toutes les issues de communication entre les noeuds voisins. Dans les réseaux sans fil, l'accès au médium commun (la fréquence) doit être contrôlé. Ceci est appelé le contrôle d'accès au Medium (MAC : Medium Access Control). La tâche principale de cette couche est d'interdire l'accès simultané au canal dans la même marge de fréquence radio. Si un récepteur reçoit deux signaux simultanément, c'est malheureusement une collision qui détruit toute l'information reçue par le récepteur. La plupart des protocoles MAC de la couche liaison tentent d'éliminer entièrement les collisions ou de réduire au minimum la capacité de canal qu'ils exigent. Il n'est cependant pas évident que cette stratégie soit la meilleure parce que la procédure responsable de l'élimination des collisions exige également un certain genre de négociation qui consomme une partie de la capacité du canal.

La différence principale entre les réseaux sans fil traditionnels et les réseaux de capteurs est le souci concernant la consommation d'énergie d'un noeud. À la différence de la couche physique, la couche de liaison empaquette souvent des bytes de données. Ces paquets contiennent une information sur le nombre de données qui permet au récepteur de détecter des erreurs de transmission.

Dans ce qui suit on présente les stratégies d'accès au médium pour les réseaux de capteurs, et quelques modes de contrôle d'erreur.

❖ **Sous couche MAC (Medium Access Control)**

On peut diviser les méthodes d'accès au médium de la couche MAC en deux catégories.

➤ **Accès centralisé** : il existe trois méthodes centralisées.

▪ ***Frequency Division Multiple Access (FDMA)***

Les différents noeuds ont différentes fréquences porteuses. Puisque les ressources en fréquences sont divisées, ceci aura comme effet de diminuer la largeur de bande disponible pour chaque noeud. FDMA exige

également du matériel additionnel et la surveillance au niveau de chaque noeud.

- ***Code Division Multiple Access (CDMA)***

Un code unique est employé par chaque noeud pour coder ses messages. Ceci augmente la complexité de l'émetteur et du récepteur.

- ***Time Division Multiple Access (TDMA)***

Le lien RF est divisé sur un axe de temps, A chaque nœud, est attribué un intervalle du temps prédéterminé qu'il peut employer pour la communication. Un avantage important est que TDMA peut être mis en œuvre au niveau du logiciel. Toutefois, TDMA exige pour tout les noeuds des horloges précises et synchronisées.

➤ **Accès distribué :** il existe deux méthodes dans cette classe.

- ***CSMA/CD (Carrier Sense Multiple Acces with Collision Detection).***

C'est une méthode d'accès partagé par écoute de la porteuse et détection des collisions. Son principe est simple: attendre que le support soit libre pour transmettre et détecter les collisions.

- **ALOHA**

Est une technique d'accès très simple, son principe est le suivant: les stations émettent de façon inconditionnelle des paquets dès qu'ils sont en leur possession. Il n'y a pas d'écoute du support avant la transmission. Dans le cas d'une collision, la station va retransmettre les paquets après un délai aléatoire.

❖ **Le contrôle d'erreurs :**

Une autre fonction importante de la couche liaison de données est le contrôle d'erreurs des données transmises. Il y a deux importants modes de contrôle d'erreur dans les réseaux de transmission, qui sont : FEC (Forward Error Correction) et ARQ (Automatic Repeat reQuest). L'utilité d'ARQ dans des applications de réseau de capteurs est limitée

par le coût additionnel de re-transmission. D'autre part, la complexité du décodage est plus grande dans FEC, car les possibilités de corrections d'erreurs doivent être intégrées. Des codes de contrôle d'erreurs simples avec moins de complexité de codage/décodage pourraient présenter de meilleures solutions pour des réseaux de capteurs.

II.4.7.3. Couche réseau

Les noeuds capteurs sont disséminés, avec une grande densité dans un domaine se situant au voisinage ou à l'intérieur de la région spatiale où se produit le phénomène à observer. Des protocoles spéciaux multi-sauts de routage entre les noeuds capteurs et le noeud puit sont nécessaires. Les techniques de routage ad-hoc ne se conforment pas habituellement aux contraintes des réseaux de capteurs. La couche réseau en ce qui concerne les réseaux de capteurs est habituellement conçue selon les principes suivants :

- L'efficacité d'énergie est toujours une considération importante.
- Les réseaux de capteurs sont la plupart du temps data-centric.
- L'agrégation de données (le clustering) est une fonctionnalité majeure.

II.4.7.4. La couche transport

La couche transport fournit un service de communication de bout en bout, fiable pour l'application. Elle manipule la segmentation des grands paquets. Elle effectue le contrôle des flots de données de bout en bout afin d'éviter la surcharge du récepteur ou du réseau. Dans les réseaux sans fil, le contrôle du flot ne fonctionne pas comme dans les réseaux câblés parce que les paquets retardés ne doivent en aucun cas congestionner le réseau. Aujourd'hui, la plupart des applications qui ont été suggérées pour des réseaux de capteurs essaient de maintenir des paquets très petits et de les envoyer peu abondamment. C'est pourquoi la couche transport n'a pas la même importance dans les réseaux de capteurs comparativement à l'Internet.

II.4.7.5. La couche application

Selon les tâches de capture, il existe différents types de logiciels qui peuvent être installés et employés pour la couche application.

II.4.7.6. Gestion d'énergie

La couche de gestion d'énergie contrôle la manière d'utiliser l'énergie par le nœud capteur, et gère la consommation d'énergie selon le mode de fonctionnement employé (capture, calcul, et communication par radio). Par exemple pour éviter de recevoir des messages redondants, le nœud capteur change son mode en « Off » après une réception d'un message d'un de ses voisins. En outre un nœud capteur annonce à ses voisins qu'il a atteint un bas niveau d'énergie, par conséquent il ne va pas participer au routage des données, et l'énergie restante va être utilisée pour capter et détecter des tâches.

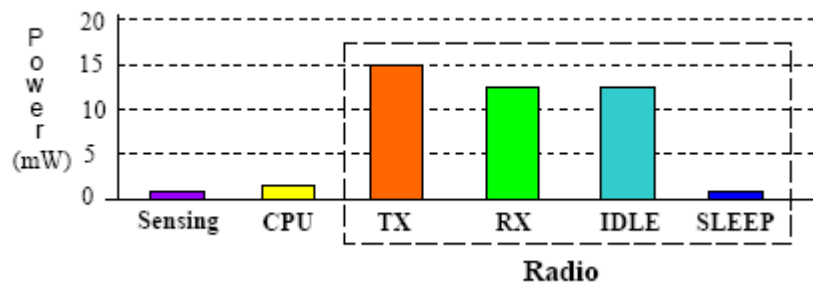


Figure II.13 : Energie consommée par les sous-systèmes d'un capteur [42]

II.4.7.7. Gestion de mobilité

La couche de gestion de mobilité détecte et enregistre le mouvement/mobilité des nœuds capteurs. En utilisant ces positions, les nœuds capteurs peuvent connaître qui sont leurs voisins. Parfois une auto-organisation des nœuds est nécessaire à cause de la destruction de quelques nœuds. Dans ce cas, la couche de gestion de mobilité doit être capable de faire changer la position des nœuds.

II.4.7.8. La gestion des tâches

La couche de gestion des tâches assure la coopération des efforts des nœuds capteurs, elle ordonnance les événements captés, et les tâches détectées dans une zone de capture spécifique. Par conséquent, les nœuds capteurs qui appartiennent à la même zone de capture ne sont pas obligés d'effectuer les tâches de capture en même temps. Selon leur niveau d'énergie, quelques nœuds capteurs peuvent accomplir des tâches de capture mieux que d'autres.

II.5 Réseaux de capteurs vs réseaux ad hoc

Les MANETs et les WSNs partagent quelques points communs mais quelques fois avec un poids différent. Le tableau suivant résume les similitudes et les différences entre les réseaux de capteurs sans fil et les réseaux ad hoc.

Réseau de capteurs sans fil	Réseau MANET
➤ Utilisation d'un médium sans fil.	➤ Utilisation d'un médium sans fil.
➤ Déploiement ad hoc.	➤ Déploiement ad hoc.
➤ Robuste aux pannes des nœuds (auto-organisation).	➤ Robuste aux pannes des nœuds (auto-organisation).
➤ Routage multi-saut.	➤ Routage multi-saut.
➤ La mobilité des nœuds est restreinte.	➤ La mobilité des nœuds.
➤ Grand nombre de nœuds (de l'ordre de mille).	➤ Nombre de nœuds moyen (de l'ordre de cents).
➤ Les nœuds ont une basse capacité de traitement et de stockage.	➤ Les nœuds ont une grande capacité de traitement et de stockage.
➤ Un objectif ciblé.	➤ Générique/communication.
➤ Les nœuds collaborent pour remplir un objectif.	➤ Chaque nœud a son propre objectif.
➤ Flot de données « many-to-one ».	➤ Flot « any-to-any ».
➤ L'énergie est un facteur déterminant.	➤ Le débit est majeur.
➤ Utilisation du broadcast.	➤ Communication point à point.

➤ Data-centric : souvent pas d'adresses uniques, les requêtes sont envoyés à tout les nœuds.	➤ Adress-centric : une adresse unique pour chaque nœud utilisée pour réaliser la communication entre les nœuds.
---	--

Tableau II.1 : Réseaux de capteurs vs réseaux ad hoc

II.6 Conclusion

Les caractéristiques de flexibilité, de tolérance aux fautes, de fidélité élevée de capture, les coûts bas et la rapidité de déploiement des réseaux de capteurs, créent beaucoup de nouveaux domaines d'application pour la télédétection. Actuellement, cette large étendue d'applications fait que les réseaux de capteurs sont une partie intégrante de notre vie. Cependant, la réalisation des réseaux de capteurs doit effectivement satisfaire des contraintes tels que la tolérance aux fautes, la scalabilité, le coût, le matériel, le changement de topologie, l'environnement et la consommation efficace d'énergie. Puisque ces contraintes sont impérieuses pour les réseaux de capteur, de nouvelles techniques ad-hoc de gestion de réseaux sans fil sont exigées. Beaucoup de chercheurs s'occupent actuellement à développer les technologies requises pour les différentes couches de la pile de protocoles comme nous allons le constater dans les chapitres suivants.

III.1 Introduction

Les systèmes multi-agents ont connu un grand essor et un grand nombre d'application dans différents domaines comme : la simulation de la vie artificielle, la robotique, la recherche d'information, le traitement d'image...etc. Grâce à son caractère décentralisé, le paradigme multi-agents présente un moyen incontournable pour la modélisation et l'implémentation des systèmes complexes.

III.2 Les systèmes complexes

III.2.1 Définition

Les systèmes complexes [43, 44] font l'objet d'études actives dans bien des domaines : la physique, la biologie, les sciences humaines et sociales, et enfin les sciences cognitives. Il s'agit donc d'un domaine d'études pluridisciplinaire. Les mathématiciens, les physiciens théoriciens mais aussi les informaticiens ont permis la création de méthodes, de formalismes et d'outils de modélisation. D'un point de vue terminologique, le terme "complexe" vient du latin *complexus* qui signifie embrasser et englober. Un problème est qualifié de complexe lorsqu'on éprouve de la difficulté à le rendre intelligible. Cette difficulté peut provenir d'un manque de connaissance, de méthode, de modèle ou de temps. A l'origine de la complexité serait donc l'interaction et la globalité. Le système serait donc complexe parla diversité et la multitude des interactions. La figure III.1 est une illustration graphique simplifiée des systèmes complexes.

III.2.2 Caractéristiques d'un système complexe ouvert

Un système complexe ouvert est un système [43, 44] :

- qui contient un grand nombre (variable) d'entités logicielles/matérielles.
- qui met en oeuvre de nombreuses interactions logiques/physiques.
- dont l'environnement est dynamique et ouvert.
- qui possède des objectifs globaux.
- dont le contrôle centralisé est impossible.
- et qui, pour toutes ces raisons, n'est pas entièrement spécifiable.

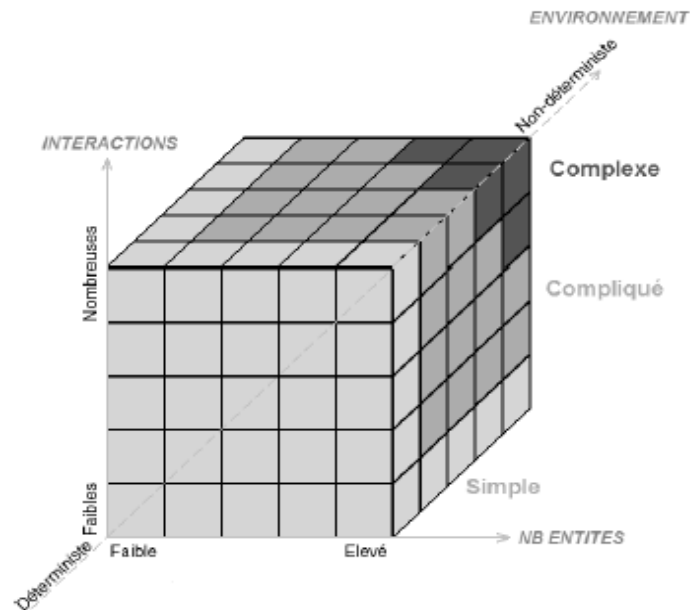


Figure III-1 : illustration graphique simplifiée des systèmes qui relèvent de la complexité.

III.3 L'origine du paradigme multi-agent

Les systèmes multi agents sont une approche de modélisation et d'implémentation qui est issue de la rencontre de divers domaines qui sont :

- l'intelligence artificielle pour les aspects prise de décision de l'agent.
- l'intelligence artificielle distribuée pour la distribution de l'exécution et la résolution des problèmes de nature distribuée.
- les systèmes distribués pour les interactions.
- le génie logiciel pour l'approche *agents* et l'évolution vers des composants logiciels de plus en plus autonomes.

III.4 Les agents

III.4.1 Définitions

Avant d'aborder les concepts fondamentaux des systèmes multi agents, il est essentiel de bien définir ce qu'est un agent, pour cette raison, on va citer quelques définitions parce qu'à l'heure actuelle il n'existe pas de définition standard de ce concept.

- ❖ **Jacques Ferber [45]** définit un agent comme étant une entité physique ou virtuelle évoluant dans un environnement dont il n'a qu'une représentation partielle et sur lequel il peut agir (voir figure III.2). Il est capable de communiquer avec d'autres agents et est doté d'un comportement autonome.

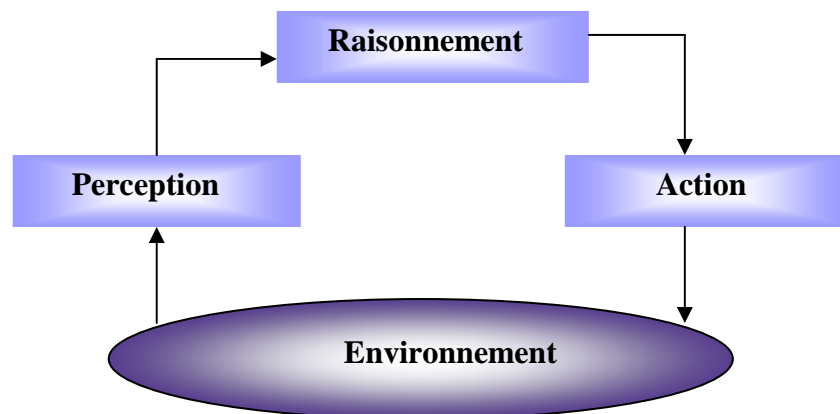


Figure III-2 : Fonctionnement d'un agent.

Cette définition met la lumière sur une caractéristique très importante des agents qui est l'autonomie, cette dernière est la capacité d'un agent d'avoir le contrôle de son comportement sans l'intervention d'un être humain. L'autre caractéristique aussi importante que la précédente est la capacité d'un agent à communiquer avec d'autres.

❖ **Selon Yves Demazeau [46]**, un agent est une entité réelle ou virtuelle dont le comportement est autonome, évoluant dans un environnement qu'il est capable de percevoir et sur lequel il est capable d'agir, et d'interagir avec les autres agents.

Cette définition introduit la notion d'interaction, qui indique la capacité d'un agent de communiquer avec les autres agents.

❖ **Pour Mickael Wooldridge [47]**, un agent est un système informatique capable d'agir de manière autonome et flexible dans un environnement.

Par flexibilité on entend :

- **Réactivité** : un système réactif maintient un lien constant avec son environnement et répond aux changements qui y surviennent.
- **Pro-activité** : un système pro-actif (aussi appelé téléonomique) génère et satisfait des buts. Son comportement n'est donc pas uniquement dirigé par des événements.
- **Capacités sociales** : un système social est capable d'interagir ou coopérer avec d'autres systèmes.

III.4.2 Les caractéristiques multidimensionnelles d'un agent

- **La nature** : agents physiques ou virtuels.
- **L'autonomie** : l'agent est plus ou moins indépendant de l'utilisateur, des autres agents, et des ressources (UC, mémoire, etc....)
- **L'environnement** : c'est l'espace dans lequel l'agent va agir ; celui-ci peut se réduire au réseau constitué par l'ensemble des agents.
- **La capacité représentationnelle** : l'agent peut avoir une vision très locale de son environnement, mais il peut aussi avoir une représentation plus large de cet environnement et notamment des agents qui l'entourent (accointances).
- **L'objectif (dimension téléonomique)** : l'agent peut poursuivre le but global du système, peut satisfaire des objectifs propres ou même se comporter dans la perspective de s'absoudre une fonction de survie.
- **La perception** : de l'environnement par l'agent.
- **La communication** : l'agent aura plus ou moins de capacités à communiquer avec les autres agents.
- **Le raisonnement** : l'agent peut être lié à un système expert ou à d'autres mécanismes de raisonnements plus ou moins complexes.

- **La quantité de ses congénères** : le système peut contenir de quelques-uns à plusieurs milliers d'agents.
- **Le contrôle** : il peut être totalement distribué entre les agents mais peut être voué à une certaine classe d'agents comme les agents « facilitateurs ».
- **L'anticipation** : l'agent peut plus ou moins avoir les capacités d'anticiper les événements futurs.
- **La granularité ou complexité** : l'agent peut être très simple comme un neurone mais aussi plus complexe.
- **La contribution** : l'agent participe plus ou moins à la résolution du problème ou à l'activité globale du système.
- **L'efficacité** : l'agent et sa rapidité d'exécution, d'intervention.
- **La bienveillance** : l'agent a plus ou moins le devoir d'aider ses congénères plutôt que de s'opposer à eux.
- **Intentionnalité** : un agent intentionnel est un agent guidé par ses buts.
- **Rationalité** : les agents rationnels disposent des critères d'évaluation de leurs actions, et sélectionnent selon ces critères les meilleures actions pour atteindre le but.
- **Adaptabilité** : un agent adaptable est un agent capable de contrôler ses aptitudes (communicationnelles, comportementales, etc.) selon l'environnement.
- **Engagement** : La notion d'engagement est l'une des qualités essentielles des agents coopératifs. Un agent coopératif planifie ses actions par coordination et négociation avec les autres agents. En construisant un plan pour atteindre un but, l'agent se donne les moyens d'y parvenir et donc s'engage à accomplir les actions qui satisfont ce but ; l'agent croit qu'il a élaboré, ce qui le conduit à agir en conséquence.
- **Intelligence** : On appelle agent intelligent un agent cognitif, rationnel, proactif et adaptatif.

III.4.3 Taxonomie des agents

III.4.3.1 Agents à capacités cognitives ou réactives

- ❖ **Agent cognitif.** Les agents à *capacités cognitives* proviennent d'une métaphore du modèle humain. Ces agents possèdent une représentation explicite de leur environnement, des autres agents et d'eux-mêmes. Ils sont aussi dotés de capacités de raisonnement et de planification ainsi que de communication. Ces agents sont structurés en société où il règne donc une véritable organisation sociale. Le travail le

plus représentatif de cette famille d'agent porte sur le modèle BDI (Believe Desir Intention) [48].

- ❖ **Agent réactif.** Les agents à *capacités réactives* ne possèdent pas de moyen de mémorisation et n'ont pas de représentation explicite de leur environnement : ils fonctionnent selon un modèle stimuli/réponse. En effet, dès qu'ils perçoivent une modification de leur environnement, ils répondent par une action programmée. L'exemple le plus célèbre est celui de la fourmilière étudié par Alexis Drogoul [49].
- ❖ **Agent hybride.** Les agents *hybrides* sont des agents ayant des capacités cognitives et réactives. Ils conjuguent en effet la rapidité de réponse des agents réactifs ainsi que les capacités de raisonnement des agents cognitifs. Cette famille regroupe donc des agents dont le modèle est un compromis autonomie/coopération et efficacité/complexité.

III.4.3.2 Agents communicants ou situés

- ❖ **Un agent purement communicant :** est un système ouvert qui détient ses propres ressources. Il est conduit par l'ensemble de ses propres objectifs, et il peut communiquer avec d'autres agents. Son comportement est lié aux communications qu'il reçoit [50].
- ❖ **Un agent purement situé :** ne communique pas directement avec d'autres agents. Les interactions se font via l'environnement. Sa position dans l'environnement est celle qui détermine son comportement. Il peut se déplacer et se reproduire.

III.4.3.3 Classification basée sur la structure de l'agent

D'après [51] chaque agent est décomposé en architecture et programme. Selon cette structure on peut distinguer les types suivants :

- ❖ **Agent à réflexe simple (simple reflex agent) :** caractérisé par l'utilisation des règles condition-action : *if (condition) then action*.
- ❖ **Agent à réflexe basé sur un modèle d'état (model-based reflex agent) :** l'agent combine entre les perceptions courantes et l'historique des états internes (dit modèle interne) afin de construire une description de l'état courant.
- ❖ **Agent basé sur le but (Goal-based agent) :** l'agent utilise les connaissances sur l'état courant et des informations décrivant la situation désirable afin d'agir sur l'environnement.

- ❖ **Agent basé sur l'utilité (utility-based agent)** : durant le chemin vers le but final on peut assigner à chaque état d'une configuration particulière de l'environnement une valeur d'utilité afin d'aboutir à un comportement de bonne qualité.
- ❖ **Agent d'apprentissage (learning agent)**: afin d'améliorer la décision de l'agent, on lui donne le pouvoir d'apprendre au fur et à mesure ce dont il a besoin.

III.5 Les systèmes multi-agents

III.5.1 Définition

« Un système multi-agent est un système distribué composé d'un ensemble d'agents ... les SMA sont conçus et implantés idéalement comme un ensemble d'agents interagissant, le plus souvent, selon des modes de coopération, de concurrence ou de coexistence » [52]

De manière générale, le système multi-agent est considéré comme un système distribué composé d'un certain nombre d'entités autonomes (les agents), qui travaillent selon des modes complexes d'interaction, pour réaliser leurs propres buts et par-là même atteindre l'objectif global désiré. Les agents peuvent interagir en communiquant directement entre eux ou par l'intermédiaire d'un autre agent ou en agissant sur leur environnement. La figure III.3 va illustrer des agent dans un SMA.

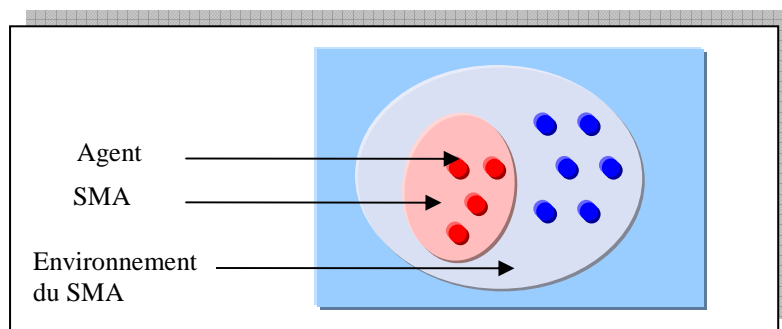


Figure III-3 : Les agents dans un SMA [53]

III.5.2 Taxonomie des systèmes multi-agents

III.5.2.1 Système multi-agents ouvert ou fermé

Un système multi-agents ouvert [54] partage les caractéristiques des systèmes ouverts. Pour ce qui concerne les entités élémentaires du système que sont les agents, ils n'ont pas la possibilité d'avoir une représentation complète de l'environnement. Pour ce qui est du système dans sa globalité, il doit être modulaire et extensible. La modularité concerne le fait que les systèmes multi-agents sont composés de plusieurs sous-systèmes mis en relation. Ces sous-systèmes ont chacun leur propre mode de fonctionnement. L'extensibilité se traduit par le fait que le système multi-agents supporte l'ajout et le retrait dynamique d'éléments.

A l'inverse, le qualificatif fermé signifie que l'ensemble des agents qui compose le système reste le même.

III.5.2.2 Système multi-agents homogène/hétérogène

Un système multi-agents homogène est composé d'agents homogènes. Deux agents ont cette particularité s'ils sont identiques du point de vue de leur modèle et de leur architecture. Le qualificatif hétérogène est utilisé pour préciser que le système multi-agents est composé d'agents différents du point de vue de leurs modèles et de leurs architectures.

III.5.3 Les constituants d'un système multi-agents

D'après Yves demazeau un SMA peut être décomposé en quatre axes : un axe **A**gent, un axe **E**nvironnement, des **I**nteractions et une structure **O**rganisationnelle explicite ou non.

$$SMA = Agent + Environnement + Interaction + Organisation$$

De plus l'intelligence d'un système multi-agents est supérieure à la somme des intelligences individuelles des agents qui le composent. Ce phénomène est appelé l'intelligence émergente.

$$Fonction (SMA) = \Sigma fonction(Agent) + fonction collective$$

III.5.3.1 Les agents (A)

Ce concept est déjà défini précédemment.

III.5.3.2 L'environnement (E)

Dans un système multi-agents, on appelle environnement l'espace commun aux agents du système. Un environnement peut être [55, 56]:

- ❖ **Accessible** : si un agent peut, à l'aide des primitives de perception, déterminer l'état de l'environnement et ainsi procéder, par exemple, à une action. Si l'environnement

est inaccessible alors il faut que l'agent soit doté de moyens de mémorisation afin d'enregistrer les modifications qui sont intervenues.

- ❖ **Déterministe ou non** : selon que l'état futur de l'environnement sera ou non fixé par son état courant et les actions de l'agent.
- ❖ **Épisodique** : si le prochain état de l'environnement ne dépend pas des actions réalisées par les agents.
- ❖ **Statique** : si l'état de l'environnement est stable (ne change pas) pendant que l'agent réfléchit. Dans le cas contraire, il sera qualifié de dynamique.
- ❖ **Discret** : si le nombre des actions faisables et des états de l'environnement est fini.

III.5.3.3 Les interactions (I)

Une des principales propriétés de l'agent dans un SMA est celle d'interagir avec les autres agents.

Selon *J.Ferber*, « Une interaction est la mise en relation dynamique de deux ou plusieurs agents par le biais d'un ensemble d'actions réciproques... Les interactions sont non seulement la conséquence d'actions effectuées par plusieurs agents en même temps, mais aussi l'élément nécessaire à la constitution d'organisations sociales. »

En effet, les interactions sont généralement définies comme toute forme d'action exécutée au sein du système d'agents et qui a pour effet de modifier le comportement d'un autre agent. Les agents peuvent interagir en communiquant directement entre eux, par l'intermédiaire d'un autre agent ou même en agissant sur leur environnement. Le tableau suivant illustre les différents mécanismes d'interaction.

Buts	ressources	Compétences	Type de situation	catégorie
Compatibles	suffisantes	suffisantes	indépendance	indifférence
		insuffisantes	Collaboration simple	
	insuffisantes	suffisantes	encombrement	Coopération
		insuffisantes	Collaboration coordonnée	
Incompatibles	suffisantes	suffisantes	Compétition individuelle pure	antagonisme
		insuffisantes	Compétition collective pure	
	insuffisantes	suffisantes	Conflits individuels pour des ressources	
		insuffisantes	Conflits collectifs pour des ressources	

Tableau III.1 : Types d'interactions d'après [45]

III.5.3.4 L'organisation (O) :

On peut la définir comme étant une structure décrivant les interactions et autres relations qui existent (dans le but d'attendrir un objectif commun) entre les membres de la dite organisation.

III.5.4 Plateformes pour les systèmes multi agents

Pour qu'un outil soit considéré comme un environnement de développement des systèmes multi agents, il doit répondre aux exigences suivantes :

- Accélérer le développement et diminuer l'effort de programmation.
- Abstraire les mécanismes de communication, d'interaction et de coordination.
- Permettre l'implémentation de systèmes relativement complexes.
- Permettre une bonne extensibilité du code.
- Fournir un support pour le déploiement (et l'exécution) des systèmes.

III.5.4.1 AgentTool

C'est un outil et une méthodologie qui mettent l'accent sur les premières phases du développement (analyse et développement). Il utilise la méthodologie MaSE (extension au modèle OO a 7 phases) : Trouver les buts, appliquer les cas d'utilisation, raffiner les rôles, créer les classes d'agents, construire les conversations, assembler les classes d'agents et l'implémentation.

C'est un outil qui permet la validation des conversations. Il génère le code des conversations en Java.

III.5.4.2 AgentBuilder

C'est un environnement de développement complet. Il utilise la méthodologie OMT à laquelle il ajoute une partie ontologie, les agents sont construits à partir d'un modèle BDI et de AGENT-0, le langage de communication entre les agents est KQML, l'exécution se fait à partir de l'engin d'AgentBuilder, qui a la possibilité de générer des fichiers « .class » et d'exécuter les systèmes sur une JVM standard.

III.5.4.3 Jack

C'est un gestionnaire de projet, qui a un éditeur de textes où se fait l'implémentation du système, la compilation et un passage de JAL à Java, puis l'exécution du système. Le langage JAL (Jack Agent Language) est une extension du langage Java. Il est développé par Agent Oriented Software Pty. Ltd., une société australienne. Les agents sont basés sur un modèle BDI (une évolution du modèle BDI de dMars). Jack Agent Kernel est la librairie de classes permettant l'exécution des agents.

III.5.4.4 DECAF

C'est un environnement de développement de plans. Cet outil fournit des utilitaires pour l'élaboration de plans et pour la coordination des tâches. Il est doté d'un planificateur qui applique des heuristiques pour trouver un ordonnancement aux tâches et d'un éditeur d'agent très utile pour le débogage.

III.5.4.5 Jade

C'est un outil répondant aux normes FIPA. Les trois modules principaux (nécessaires aux normes FIPA): le DF « directory facilitator » fourni un service de pages jaunes à la plate-forme, le ACC « agent communication channel » gère la communication entre les agents, le AMS « agent management system » supervise l'enregistrement des agents, leur authentification, leur accès et l'utilisation du système. Le langage de communication entre les agents est FIPA ACL. Jade possède aussi un éditeur pour l'enregistrement et la gestion des agents.

III.5.4.6 JAFMAS et JiVE

JAFMAS met l'accent sur les protocoles de communication, l'interaction entre les agents, la coordination et la cohérence à l'intérieur du système. Une méthodologie en cinq phases : identifier les agents, les conversations, les règles de conversation, analyser le modèle des conversations et l'implémentation. Il utilise les réseaux de Pétri. JiVE est le support graphique pour le développement avec JAFMAS.

III.5.4.7 Zeus

C'est un environnement de développement complet développé par l'Agent Research Program du British Telecom Intelligent System Research Laboratory, il utilise une approche « role modeling ».

Les agents possèdent trois couches :

- La définition : l'agent est perçu comme une entité autonome capable de raisonner grâce à ses croyances, ses ressources et ses préférences.
- L'organisation : relations entre les agents.
- La coordination : modes de communication entre les agents, protocoles, coordination et autres mécanismes d'interaction.

III.5.4.8 Madkit

MadKit (Multi-agents Development Kit) est un environnement écrit en Java basé sur la méthodologie Aalaadin ou AGR (agent/ groupe / rôle). Un agent joue des rôles à l'intérieur du groupe. MadKit a été développé en 1996 par Olivier GUTKNECHT et Michel FERBER au laboratoire d'Informatique, de Robotique et de Microélectronique de Montpellier (LIRMM) de l'université Montpellier II. L'outil fournit un éditeur permettant le déploiement et la gestion des SMA (G-box). Il offre aussi un utilitaire pour effectuer des simulations à grande échelle.

III.6 Conclusion

La complexité qui caractérise les systèmes informatiques d'aujourd'hui, et le contrôle centralisé extrêmement difficile de ces derniers, voire impossible, la plupart du temps, a conduit à introduire le paradigme multi-agents qui est issu de l'IAD (intelligence artificielle distribuée) et qui est basé sur des techniques de décentralisation des traitements et des contrôles pour améliorer les performances et assurer un fonctionnement cohérent des systèmes informatiques.

V.1 introduction

Puisque la simulation d'un réseau de capteurs avant son déploiement réel représente une alternative moins coûteuse et moins lourde que son expérimentation en environnement réel, On va présenter dans ce chapitre les raisons d'implémenter une plate-forme de simulation des réseaux de capteurs et on va citer quelques simulateurs de réseaux de capteurs qui sont actuellement disponibles.

V.2 Différentes approches de test

V.2.1 Le test en environnement réel

La première approche pour tester un protocole ou une application consiste à mettre réellement en oeuvre le réseau capteur sans fil qu'on doit tester. Cette solution est à priori la meilleure dans la mesure où l'on utilise vraiment le système opérationnel ainsi que l'implémentation finale du protocole. Cependant, cette solution peut être assez difficile à mettre en oeuvre et peut être très coûteuse dans certains cas. Un autre point négatif de cette solution est l'impossibilité d'évaluer le protocole aux conditions limites de propagation (pendant une tempête par exemple) ou de mobilité (par exemple avec des vitesses de déplacement très importantes). De même, il est très difficile de réussir à mettre en oeuvre des cas rares. Enfin, cette solution n'autorise pas au développeur l'exacte reproduction des conditions d'une expérience à l'autre, ce qui peut être très gênant pour réussir à mettre au point son protocole.

V.2.2 La simulation

La deuxième solution est la simulation. C'est souvent ce choix qui est fait par la communauté scientifique. Elle possède en effet l'avantage de reposer sur des modèles validés ou au moins reconnus. Il existe plusieurs simulateurs à événements discrets permettant de mettre en oeuvre un réseau sans fil tels que NS-2, Opnet, QualNet et GloMoSim. Un simulateur à événements discrets se caractérise par le fait que les changements d'états dans le réseau simulé (événements) se produisent à des instants (sans durée) répartis de manière discrète sur l'axe des temps. La simulation demande une modélisation complète, allant des applications au réseau physique, ce qui peut se révéler à la fois complexe et coûteux en terme de temps d'exécution et d'utilisation mémoire dans le cas des réseaux de tailles importantes. Ainsi, tous les simulateurs que nous avons cités précédemment simulent complètement les piles protocolaires ainsi que le trafic. Ainsi, le processus de simulation oblige l'utilisateur à faire un double développement de son système et ceci sans lui assurer la cohérence du modèle de simulation et de l'implémentation.

V.2.3 L'émulation

L'émulation peut être vue comme un compromis entre les deux solutions précédentes dans la mesure où elle va permettre de tester l'implémentation réelle du protocole/application sur un réseau simulé en temps réel. Le but de l'émulation sera donc de reproduire le comportement d'un réseau jusqu'au niveau d'une couche donnée en simulant en temps réel les effets des couches sous-jacentes du réseau cible. L'émulation permet d'avoir un contrôle équivalent à celui de la simulation sur les paramètres du réseau mais permet en plus d'utiliser des implémentations réelles à tester. Ainsi, elle permet de tester un protocole aux conditions limites, ce qui est très difficile à faire dans un réseau réel. On parlera alors d'évaluation d'un protocole sur des cas rares. De plus, comme l'environnement simulé est entièrement contrôlable, il est possible de comparer ce protocole avec d'autres protocoles existants en reproduisant plusieurs fois de suite la même expérience. Le fait de pouvoir reproduire précisément une expérience va également permettre, comme pour la simulation, de bien paramétrer son protocole.

Bien que l'émulation parait plus intéressante et porte plus d'avantages, elle impose cependant certaines contraintes. Comme l'émulation travaille en temps réel, les modèles utilisés pour simuler les couches basses ne peuvent pas être trop complexes ce qui implique un impact négatif sur le réalisme de l'émulation rendue. A cet effet, l'approche de simulation détient toujours sa place comme solution pour le test et la validation de protocoles et d'applications.

V.3 Les simulateurs Actuellement disponibles

❖ Ns-2

Ns-2 est un simulateur à événements discrets bien détaillé dans [62], il fournit un support étendu pour simuler les protocoles TCP/IP, de routage et multi sauts pour les réseaux câblés et sans fil [63]. Le modèle de propagation radio est basé sur la réflexion de deux rayons d'approximation et de média partagé modélisé dans la couche physique, le protocole MAC IEEE 802.11 dans la couche de liaison, et une implémentation du routage dynamique pour la couche réseau ont été développées dans le projet Monarch [64].

❖ SensorSim

SensorSim est construit sur Ns-2 pour inclure des modèles d'énergie et du canal de capture [65]. À chaque noeud, les consommateurs d'énergie opèrent dans de

multiples modes et consomment différentes quantités d'énergie en chaque mode. Le modèle du canal de capture est l'interaction dynamique entre l'environnement physique et les nœuds capteurs.

❖ **OPNET**

OPNET est une plateforme commerciale pour simuler les communications réseaux [66]. Conceptuellement, le modèle d'OPNET comporte les processus qui sont basés sur les machines à états finis, ces processus communiquent comme indiqué dans le modèle supérieur. Le modèle sans fil emploie une canalisation 13-stage pour déterminer la connectivité et la propagation entre les nœuds. Les utilisateurs peuvent indiquer la fréquence, la largeur de bande, et la puissance parmi autres caractéristiques comprenant des modèles de terrain.

❖ **J-Sim**

J-Sim [67] est une plateforme de simulation de réseaux écrite en Java, orientée objets, basée composant et à événements discrets. Des modules peuvent être ajoutés et supprimés par plug-and-play, J-Sim est utilisé pour la simulation et l'émulation de réseau en incorporant un ou plusieurs dispositifs réels de capteur. Cette plateforme fournit le support pour la modélisation de l'objet cible, les nœuds capteurs et les stations de base, les canaux de capture et les voies de transmission sans fil, les canaux sismiques, les modèles de puissance et les modèles d'énergie.

❖ **GlomoSim**

GlomoSim est une collection de modules de bibliothèque, dont chacun a simulé un protocole spécifique de transmission sans fil [68]. Il est employé pour simuler les réseaux sans fil ad-hoc et mobiles.

❖ **OMNeT++**

OMNeT++ « Objective Modular Network Test-bed in C++ » [69][70], est un environnement de simulation basé composant, modulaire et une architecture ouverte. Son domaine d'application primaire est la simulation des réseaux de transmission, mais grâce à son architecture générique et flexible, il a été employé avec succès dans d'autres domaines d'application comme la modélisation des multiprocesseurs et les systèmes hardware distribués et d'autres domaines.

	Langage/script	Modules disponibles	Support graphique	émulation
1DaSSF/TOSSF	C++/DML	C/W/A/WSN+	propriétaire	limité
JiST/SWANS	Java/Jython	W/A	-	oui
J-Sim	Java/Jacl	C/W/A/WSN+	Bonne édition et débugeage	oui
NCTUns	C	C/W/A	Excellente édition et débugeage	oui
1NS-2	C++/OTcl	C/W/A/WSN	-	limité
1OMNET++	C++/NED	C/W/A	Limité pour l'édition, excellent pour le débugeage et l'exécution est animée.	limité
Ptolemy II	Java	C/W/A/WSN+	Excellente édition et débugeage	Oui
<p>1 inclue un support pour une simulation parallèle.</p> <p>*C: modèles classiques (par exemple : TPC/IP, Ethernet)</p> <p>A: support Ad hoc (par exemple MANET protocols: AODV, DSR)</p> <p>W: support Wireless (par exemple, la propagation, mobilité, IEEE802.11)</p> <p>WSN: quelques protocoles communs des WSNs (par exemple, Directed Diffusion, S-MAC)</p> <p>WSN+: un support WSN riche avec un environnement et modèle de batterie.</p>				

Tableau V.1 : plateformes de simulation des réseaux [71].

	plateforme	Système d'exploitation	scalabilité	hétérogénéité
ATEMU	processeur AVR	Nimporte quel OS sur le processeur AVR	500 noeuds	Différents capteurs et différentes applications
EmStar EmTOS	MicroServers Berkeley Motes	Linux (FUSD) TinyOS	500 noeuds	Mica et Microservers Avec différentes applications
SNAP	processeur SNAP	—	100.000 noeuds	différents codes sur des processeurs SNAP
TOSSIM	Berkeley Motes	TinyOS	1000 noeuds	seulement TinyOS/Mica, la même application est exécutée

Tableau V.2 : plateformes spécialisées dans la simulation des réseaux de capteurs sans fil [71].

V.4 conclusion

La complexité des réseaux de communication, l'importance d'une phase de simulation avant le déploiement réel des réseaux en vue de choisir les protocoles de communication adéquats satisfaisant les métriques données par l'utilisateur ainsi que l'évaluation préalable des performances ont conduit à l'implémentation de plusieurs simulateurs.

VI.1 Introduction

Nous nous intéressons dans ce chapitre aux protocoles de communications dans les réseaux WSN. Un grand nombre de protocoles ont vu le jour, durant les trente dernières années, marquées par une intense activité de recherche dans ce domaine. Ils diffèrent les uns des autres par les types de média utilisés et les exigences de performance pour lesquelles ils ont été conçus et optimisés.

Nous présenterons les aspects fondamentaux des protocoles MAC et nous expliquerons les exigences spécifiques et les problèmes auxquels ces protocoles doivent respectivement y répondre et y faire face. La plus importante des exigences est sans doute l'efficacité de l'énergie. A ce titre, il est impératif d'examiner les sources spécifiques de perte d'énergie, inhérentes aux protocoles MAC.

VI.2 Les protocoles de la couche MAC

VI.2.1 Les raisons de la perte d'énergie

Physiquement, un nœud émetteur-récepteur du WSN, possède quatre modes principaux de fonctionnement :

- Le mode émetteur.
- Le mode récepteur.
- Le mode sommeil.
- Le mode ralenti.

A chacun des quatre états est associé un niveau de consommation d'énergie. Celle-ci n'est nulle pour aucun des quatre régimes de fonctionnement.

Un ensemble d'autres raisons, intimement liées au protocole, vont être responsables de pertes supplémentaires d'énergie, nous citons :

- Les collisions : Lorsque plus d'un paquet convergent simultanément vers un nœud récepteur, qu'ils chevauchent partiellement ou totalement, ils sont en collisions et sont par conséquent tous rejetés et doivent faire l'objet de re-transmissions.
- La sur-écoute, « Overhearing », compte tenu du caractère « large diffusion » du médium utilisé dans les WSN, tous les nœuds récepteurs du voisinage de la source émettrice reçoivent le paquet de données avant de se rendre compte qu'il ne leur est pas destiné.
- L'« Overemitting » : Cette situation correspond à la transmission d'un paquet par un nœud émetteur alors que le destinataire n'est pas prêt à la réception.

- Les paquets de contrôle : ce sont les flots de données destinés au contrôle et qui doivent être réduits à leur expression minimale.

VI.2.2 Les modèles de communication dans un WSN

Il existe plusieurs typologies pour classer les modèles de communication dépendant des critères utilisés. Kulkarni [72] et al définissent trois types de modèles de communications dans les WSN :

- La diffusion large, «broadcast », c'est généralement le schéma utilisé par une station de base pour transmettre une information à tous les nœuds du réseau. L'information, objet de la transmission, peut inclure des requêtes destinées aux nœuds, des mises à jour de programmes ou des paquets de contrôle.
- La diffusion convergente, « convergecast », ce deuxième schéma correspond à la situation où un groupe de capteurs ciblent un nœud particulier par une information. Ce nœud particulier peut aussi bien être la tête d'une grappe « Clusterhead », un nœud central de fusion de données ou une station de base.
- La conversation locale, « local gossip », c'est le cas d'un capteur qui envoie à ses voisins (avec une définition adéquate de la profondeur du voisinage) une information particulière, qui peut être, par exemple, la détection d'un intrus.

Dans les protocoles incluant la mise en grappes, les clusterheads communiquent avec leurs membres, les récepteurs visés par la communication peuvent ne pas être tous les voisins du clusterhead, mais un ensemble de voisins. Ce genre de communication peut être considéré comme un quatrième modèle, désigné par diffusion multiple, « multicast ». C'est de la diffusion multiple sélective qui vise un ensemble spécifique, à ne pas confondre avec la large diffusion.

VI.2.3 Les attributs d'un bon protocole MAC

Au vu de ce que nous venons de mentionner, on peut dresser une liste d'attributs qu'un protocole MAC bien conçu devrait posséder. Traditionnellement, les exigences de performance les plus importantes pour les protocoles MAC sont l'efficacité, la stabilité, un temps mort d'accès minimal (c'est l'intervalle de temps qui s'écoule entre l'arrivée d'un paquet de données et la première tentative pour le transmettre) et un temps mort minimal pour la transmission (la durée séparant l'arrivée d'un paquet de données et sa livraison avec succès). A cela, on peut ajouter, l'adaptabilité aux changements et l'auto ajustement d'échelle « scalability ».

Pour des applications pour lesquelles le temps revêt un caractère critique, le protocole doit fournir des garanties déterministes ou stochastiques sur le temps de livraison ou sur le taux minimal de transmission de données. Quelquefois, le traitement préférentiel des paquets jugés importants doit primer sur les paquets de moindres importances.

Ne perdons pas de vue, cependant, que comme bien même on essaie de réunir toutes ces qualités dans le protocole MAC, celui-ci reste soumis aux contraintes incontournables de la couche physique. Rappelons, en effet, qu'en raison du médium utilisé pour les WSNs (communications par ondes électromagnétiques), un tas de problèmes risque d'affecter le fonctionnement de ces réseaux tels que : l'atténuation rapide du signal, perte sur le trajet, bruits thermiques, etc...

A défaut de pouvoir satisfaire toutes les exigences, on préfère mettre en avant, pour une application donnée dans un contexte spécifique, les attributs les plus pertinents pour le fonctionnement du WSN en cherchant le meilleur compromis possible.

Dans ce qui suit on va décrire le comportement des protocoles MAC les plus célèbres, utilisés dans les WSNs avec leurs avantages et leurs inconvénients.

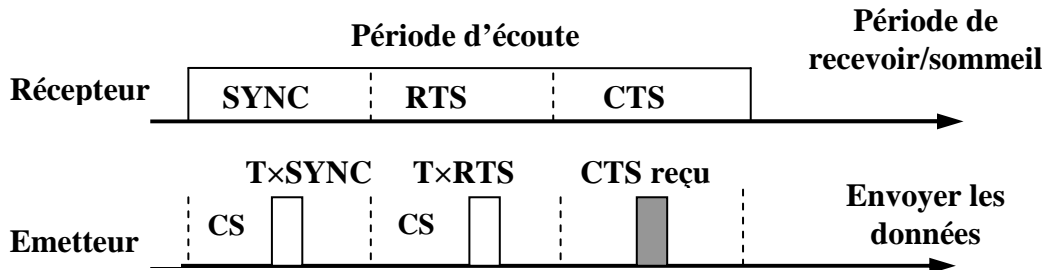
❖ *Sensor-MAC(S-MAC)*

La gestion locale des synchronisations et les ordonnancements périodiques des phases de sommeil - écoute basés sur ces synchronisations constitue l'idée fondamentale du protocole S-MAC.

Les noeuds voisins forment des clusters virtuels pour installer un ordonnancement commun du sommeil. Si deux noeuds voisins résident dans deux clusters virtuels différents, ils se réveillent pour l'écoute sur les périodes associées aux deux clusters.

Un inconvénient de S-MAC est la possibilité de suivre deux ordonnancements différents, qui augmente la consommation d'énergie à cause de l'écoute oisive et l'overhearing. Des échanges d'ordonnancement sont accomplis par les émissions périodiques de paquets SYNC aux voisins immédiats. La période de chaque noeud pour l'envoi d'un paquet SYNC s'appelle *période de synchronisation*. L'action permettant d'éviter une collision est réalisée par un *carrier sense*, qui est représenté comme CS. En outre, les paquets de RTS/CTS sont utilisés pour le type paquets de données d'unicast. Une technique importante de S-MAC est sa manière de passer le message où les messages volumineux en les divisant en frames. Grâce à cette technique, il devient possible de conserver l'énergie en réduisant la communication overhead. Le sommeil périodique peut avoir comme conséquence un temps d'attente

élevé. Le temps d'attente causée par le sommeil périodique s'appelle *le sleep delay* [73].



RTS: Request To Send **CTS:** Clear To Send

Figure VI.1 : Scénario de transmission de messages pour S-MAC [73]

- **Avantages**

- La perte d'énergie provoquée par l'écoute oisive est réduite par les ordonnancements des phases de sommeil.
- La simplicité d'implémentation et la synchronisation des temps.
- L'overhead peut être évité par les ordonnancements des phases de sommeil.

- **Inconvénients**

- La diffusion des paquets de données n'utilise pas RTS/CTS qui augmente la probabilité de collision.
- Peut causer l'overhead, si le paquet est non destiné au noeud qui écoute.
- Les périodes de sommeil/écoute sont prédéfinies et constantes, ce qui diminue l'efficacité de l'algorithme lorsque la charge de trafic est variable.

- ❖ **WiseMAC**

Le protocole TDMA et CSMA spatiaux avec l'échantillonnage de préambule est proposé dans [74] où tous les nœuds capteurs sont définis de façon à disposer de deux canaux de communication. L'accès au canal de données se fait par la technique TDMA, alors que l'accès au canal de contrôle est réalisé par la technique CSMA. Enz *et autres* [75] ont proposé le protocole WiseMAC qui est similaire au travail de Hoydi *et autres* [74] mais exige seulement un canal unique.

Le protocole WiseMAC utilise non-persistent CSMA (np-CSMA) avec l'échantillonnage de préambule comme dans [74] pour diminuer l'écoute oisive. Dans

la technique l'échantillonnage de préambule, un préambule précède chaque paquet de données pour alerter le nœud récepteur. Tous les nœuds dans le réseau échantillonnent le médium avec la même période, mais leurs déplacements d'ordonnement relatifs sont indépendants. Si le nœud trouve le médium occupé après son réveil et qu'il échantillonne le médium, il continue alors l'écoute jusqu'à ce qu'il reçoit un paquet de données ou jusqu'à ce que le médium devienne inoccupé. La taille du préambule est initialement fixée égale à la période d'échantillonnage.

Cependant, pour diverses raisons, le récepteur peut ne pas être prêt à la fin du préambule, comme par exemple l'interférence, qui peut causer la perte d'énergie par l'*overemitting*. D'ailleurs, l'*overemitting* augmente avec la longueur du préambule et le nombre de paquets de données. Pour réduire l'énergie consommée causée par la taille fixée et prédéfinie du préambule, WiseMAC offre une méthode pour définir dynamiquement la taille du préambule. Cette méthode utilise la connaissance de l'ordonnement du sommeil des nœuds voisins directs du nœud émetteur. Les nœuds apprennent et actualisent l'ordonnement du sommeil de leurs voisins pendant chaque échange de données par un message d'acquiescement. De cette manière, chaque nœud maintient une table d'ordonnement du sommeil de ses voisins. En se basant sur la table d'ordonnement de sommeil des voisins, WiseMAC ordonnance les transmissions de façon que le temps d'échantillonnage du nœud destinataire correspond à la moyenne du préambule du nœud émetteur. Pour réduire la probabilité de collision provoqué par l'heure de départ spécifique du réveil du préambule. Un réveil aléatoire du préambule est conseillé. Un autre paramètre affectant le choix de la taille du réveil du préambule est le décalage potentiel d'horloge entre la source et la destination. Une borne inférieure pour la taille du préambule est calculée comme le minimum de la période d'échantillonnage de destination, T_w , et le décalage potentiel des horloges, qui est alors un multiple du temps depuis le dernier paquet ACK arrivé. Cette borne inférieure, T_p , de la taille du préambule est choisie aléatoirement.

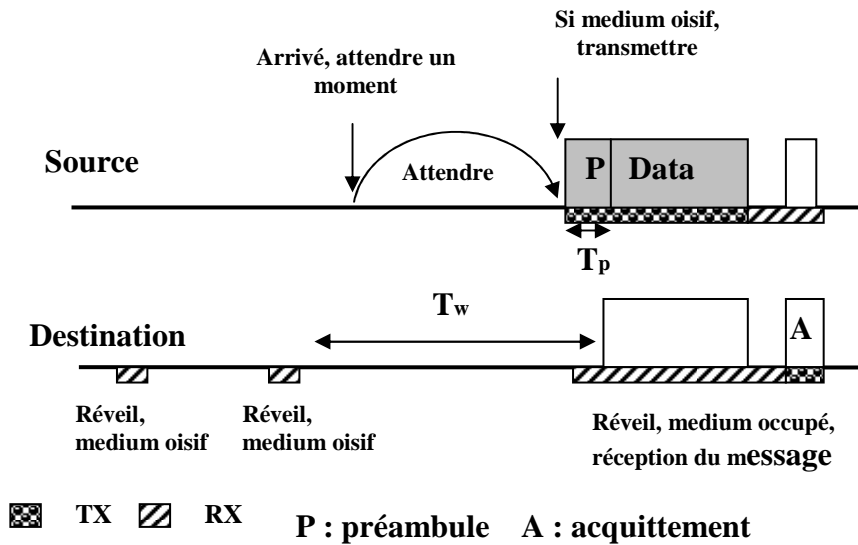


Figure VI.2 : le concept du protocole WiseMAC [75]

- **Avantages**

- Les résultats de simulation ont montré que WiseMAC est meilleur que les variantes de S-MAC [75].
- L'ajustement dynamique de la taille du préambule donne de bonnes performances dans des conditions de trafic variable.
- Le décalage d'horloge est traité par la définition du protocole qui répond au besoin de synchronisation du temps externe.

- **Inconvénients**

- L'inconvénient principal de WiseMAC est l'ordonnancement décentralisé d'écoute-sommeil, les résultats d'établissement du sommeil et d'écoute dans des temps différents pour chaque voisin du nœud. Ceci constitue un problème particulièrement important pour la diffusion, puisque le paquet envoyé sera protégé pour des voisins en mode de sommeil et livré beaucoup de fois à chaque voisin qui se réveille. Cette redondance de transmission va augmenter le temps d'attente et la consommation d'énergie.
- Un autre problème de WiseMAC, comme dans le modèle d'algorithme de TDMA et CSMA spatiaux avec l'échantillonnage de préambule, est lié au fait que WiseMAC est basé sur np-CSMA. Ce problème aura comme conséquence des collisions quand un nœud commence à transmettre le préambule à un nœud qui est déjà en situation de réception à partir d'un autre nœud.

❖ *DMAC*

« convergecast » est le type de modèle de communication le plus utilisé dans les réseaux de capteurs. Ses chemins unidirectionnels depuis les sources possibles jusqu'aux destinataires peuvent être représentés par un arbre de rassemblement de données (data gathering tree). Le but principal de DMAC [76] est d'atteindre un temps d'attente minimal, pour être efficace en terme d'énergie. DMAC peut être vu comme un algorithme du genre ALOHA amélioré où les créneaux horaires sont assignés à des ensembles de nœuds basé sur un arbre de rassemblement de données comme on le montre dans la figure IV-3. Par conséquent, Pendant la période de réception d'un noeud, tous ses noeuds enfants disposent de périodes pour transmettre et pour se contester le medium. Un temps d'attente bas est obtenu en assignant des créneaux subséquents aux nœuds successifs dans le chemin de transmission de données.

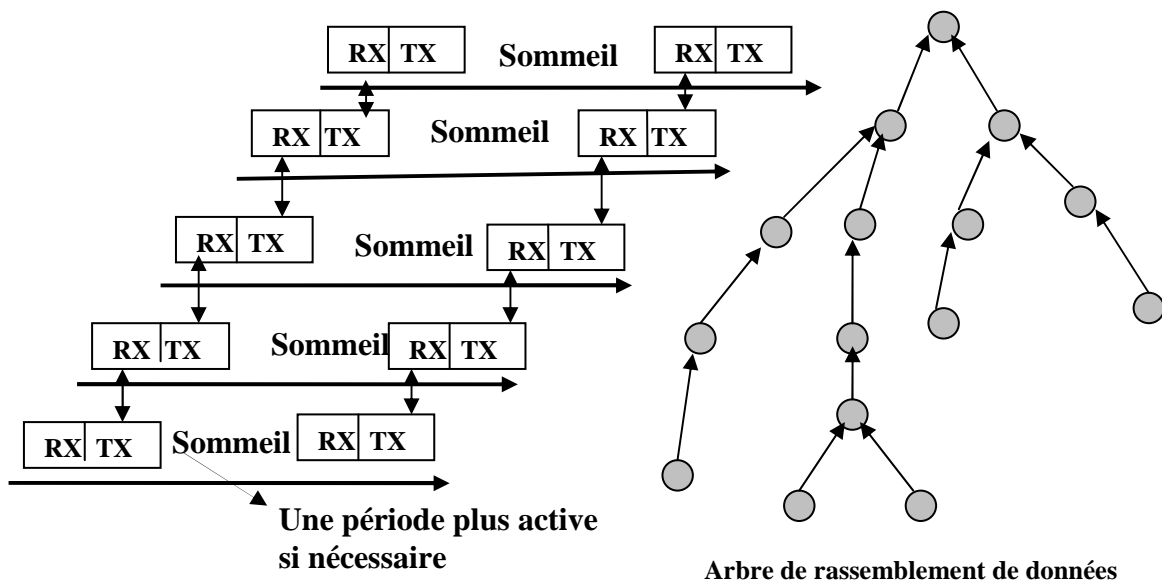


Figure VI.3 : Arbre de rassemblement de données et schéma de réveil [76]

• *Avantages*

- DMAC réalise un très bon temps d'attente comparativement à d'autres méthodes d'ordonnancement de période d'écoute/sommeil. Le temps d'attente du réseau est crucial pour certains scénarios pour lesquels DMAC pourrait être un candidat fort.

- **Inconvénients**

- Les méthodes pour éviter les collisions ne sont pas à l'oeuvre, par conséquent quand un nombre de nœud de même niveau dans l'arbre essayent de transmettre vers le même nœud, une collision aura fatalement lieu.
- En outre, les chemins de transmission de données peuvent ne pas être connus à l'avance, ce qui empêche la formation d'arbre de rassemblement de données.

VI.3 Les protocoles de routage

Une panoplie de protocoles de routage a été élaborée durant les trente dernières années. La diversité trouve son origine dans la recherche incessante et croissante de protocoles de plus en plus optimisés, répondant aux impératifs de performance et tenant compte de la multiplicité des contextes.

Nous allons décrire les protocoles célèbres, les plus utilisés, en mettant l'accent sur leurs avantages et leurs inconvénients, en rappelant que tout est affaire de compromis.

Signalons, en premier lieu, que les protocoles de routage, peuvent être classés, approximativement en trois catégories :

- Les protocoles à assignations fixées.
- Les protocoles à demande d'assignation.
- Les protocoles à accès aléatoires.

a- Les protocoles à assignations fixées

Pour cette classe de protocoles, les ressources disponibles sont réparties entre les nœuds à long terme. Chaque nœud peut alors utiliser exclusivement les ressources qui lui sont alloués sans risque de collisions. Le long terme, signifie ici, des durées importantes de plusieurs minutes, de plusieurs heures, voire plus, par opposition avec le court terme, pour lequel l'affectation des ressources a un horizon temporel de quelques centaines de millisecondes au plus. Pour tenir compte des changements de topologie, dus, par exemple, aux nœuds qui disparaissent ou aux nœuds nouvellement déployés ou tout simplement aux changements induits par la mobilité, ces protocoles doivent contenir des mécanismes de signalement pour renégocier l'allocation des ressources.

b- Les protocoles à demande d'assignation

L'affectation exclusive de ressources aux nœuds est faite, dans ce cas sur le court terme, typiquement, la durée nécessaire au traitement d'un paquet. Cette

deuxième catégorie de protocoles se subdivise en deux sous classes : Les protocoles centralisés et les protocoles distribués.

Dans les protocoles centralisés, un nœud est érigé en centre, il reçoit des requêtes de sollicitations de ressources et accepte ou refuse l'allocation au demandeur. Dans les protocoles distribués, cette mission d'allocation des ressources à la demande n'est pas le privilège unique d'un seul nœud, mais d'un ensemble.

c- Les protocoles à accès aléatoire

Pour cette catégorie de protocoles, les nœuds ne sont pas coordonnés. Les protocoles opèrent alors de manière pleinement distribuée. Ils incorporent, habituellement, un élément aléatoire comme par exemple l'exploitation de l'arrivée aléatoire d'un paquet pour fixer les temporisateurs à des valeurs aléatoires et ainsi de suite.

A côté de cette classification grossière, que nous venons de développer, une taxonomie plus fine a été tentée par des chercheurs qui travaillent dans le champ des WSNs. La figure VI.4 montre une taxonomie qui différencie les protocoles en fonction de leurs modes opératoires puis en fonction des structures des réseaux sur lesquelles ils vont exercer leurs actions.

En raison, de la vision panoramique qu'offre cette taxonomie, nous allons la développer en donnant des exemples de protocoles qui s'inscrivent dans les différentes catégories. Nous commencerons par passer en revue les protocoles basés sur la structure du réseau et les principes essentiels qui mettent en lumière leurs fonctionnements, ensuite nous allons explorer les classes de protocoles de routage en fonction de leurs modes opératoires.

VI.3.1 Les protocoles basés sur la structure du réseau

On distingue trois classes de protocoles basés sur la structure du réseau.

- **Le routage plat** : Dans ces protocoles, les mêmes rôles sont affectés à tous les nœuds du réseau.
- **Le routage hiérarchique** : Désigné aussi par routage basé sur les grappes « Cluster-Based-Routing ». Pour ces protocoles, les nœuds peuvent se voir affecter des rôles différenciés dans le réseau. Le protocole inclut, alors, la création des grappes, et additionnellement la désignation des tâches pour les nœuds capteurs est réalisée.

- **Le routage adaptatif** : Dans le cas des protocoles adaptatifs, les paramètres du système sont contrôlés pour s'adapter aux conditions actuelles du réseau en utilisant l'information acquise sur le réseau et la négociation entre nœuds.

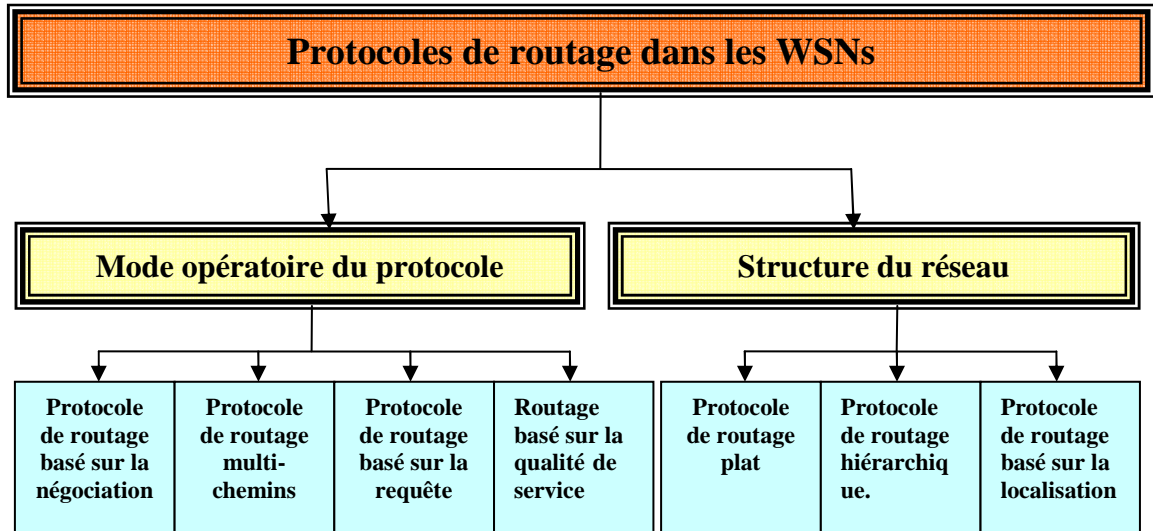


Figure VI.4 : Taxonomie des protocoles de routage

VI.3.1.1 Le routage plat

Plusieurs protocoles basés sur des algorithmes différents réalisent un routage plat. Les plus connus sont :

a- Routage par assignation séquentielle, « Sequential Assignment Routing »

Ce protocole [77] dépend de trois facteurs, la qualité de service (QoS), les ressources d'énergie et la priorité du paquet. Il utilise une approche multi-chemins. L'objectif de l'algorithme SAR est de minimiser le poids pondéré de la métrique associé au QoS sur la durée de vie du réseau. Une procédure d'établissement de liaison entre nœuds voisins est utilisée pour se remettre d'un échec de communication éventuel.

b- La diffusion guidée ou dirigée « Directed Diffusion »

La diffusion dirigée [78] ou Directed Diffusion est basé sur le Data-Centric (DC). L'idée principale du paradigme DC est de combiner les données venant de différentes sources, éliminer les redondances et minimiser le nombre de transmissions afin de conserver l'énergie du réseau ; Contrairement au traditionnel paradigme appelé Address-Centric (AC) qui tente de trouver le plus court chemin entre deux nœuds mobiles (end-to-end routing).

Pour ce type, les données générés par les nœuds capteurs sont nommés en utilisant des paires d'attributs-valeurs et utilisées dans les traitements dans le réseau. Il fonctionne selon le paradigme Publier/S'abonner. L'idée prévalente ici, bien qu'elle ne soit pas la seule, est de s'abonner auprès des sources qui publient les données à un rythme défini sur une durée déterminée. La diffusion guidée est considéré actuellement beaucoup plus une philosophie de conception qu'un protocole concret. Il existe, en réalité, un nombre de variantes de ce protocole, optimisé pour différentes situations.

Dans cette configuration, la distribution des données commence par l'annonce de certains nœuds de leurs intérêts pour un certain type de données nommées. Ils spécifient leurs intérêts par un ensemble de paires d'attributs-valeurs. Dans le langage du Publier/S'abonner ceci correspond parfaitement à un abonnement aux données. Les messages véhiculant les intérêts sont distribués dans le réseau. Grâce à ce flot, porteur des informations sur les intérêts des nœuds, il devient possible de mettre en place un arbre de diffusion convergente « convergecast », avec chaque nœud gardant en mémoire le nœud à partir duquel il a reçu le message porteur de l'intérêt.

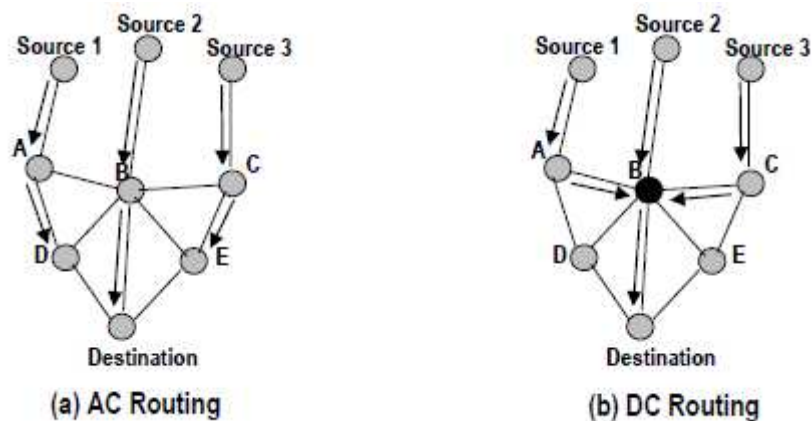


Figure VI.5 : la différence entre le routage AC et DC [95]

Ce protocole élimine la redondance des données et minimise le nombre de transmissions, il en résulte une sauvegarde et une économie d'énergie qui prolonge la durée de vie du réseau. Additionnellement, ce protocole vise l'adaptation de l'échelle du réseau.

c- Le routage par la rumeur « Rumor routing »

Braginsky et Estrin [79] ont considéré cette approche dans le contexte de la notification d'événement. Supposons que des capteurs sont intéressés par certains événements (exemple : la température excédant un certain seuil) et qu'un capteur puisse l'observer. L'approche classique consiste à inonder le réseau d'une requête sur

l'événement ou sur ses notifications sur le réseau entier. Le routage par la rumeur n'inonde pas le réseau par l'information portant sur l'occurrence d'un événement mais installe un petit nombre de chemins dans le réseau en y répandant un ou plusieurs agents. Chacun de ces agents va se propager d'un nœud à l'autre et chemin faisant il va installer l'information de routage concernant l'événement et ceci dans chaque nœud visité. Lorsque un nœud tente une requête sur l'événement, il dissémine lui-même un ou plusieurs agents. Un tel agent de recherche est envoyé à travers le réseau jusqu'à ce qu'il croise un chemin pré-installé de l'événement. Il saura alors comment trouver l'événement.

Comme, on peut s'en rendre compte, ce protocole de routage a un caractère probabiliste. La probabilité pour trouver un nœud disposant de l'information sur l'événement a été estimée à 69% pour deux chemins. Cette probabilité atteint 99.7% pour cinq chemins.

d. Le routage par gradient « (GRAB) »

L'algorithme de routage par gradient [80] attribue une hauteur à chaque nœud. Le nombre de sauts qui sépare le nœud du puits est un candidat, qui exprime cette grandeur, pour gérer de manière efficiente l'énergie ; Le couple formé par le nombre de sauts qui sépare le nœud du puits et l'énergie résiduelle du nœud peut se substituer au choix précédent pour exprimer la hauteur. L'attribution de la hauteur au nœud se fait lors d'une phase d'initialisation. Au cours d'une première étape, le puits envoie un message avec une hauteur courante nulle, dans une seconde phase chaque nœud destinataire de ce message, initialise sa hauteur à la valeur courante incrémentée de 1 et renvoie le message avec cette nouvelle valeur comme hauteur courante. Au terme de ce processus d'initialisation, tout nœud dispose de sa hauteur et de la hauteur du proche voisinage. L'information relative à la hauteur des voisins peut être explorée par l'envoi de messages périodiques.

Une fois les hauteurs attribuées, la différence entre la hauteur d'un nœud et la hauteur de son voisin forme le gradient sur ce lien. Des techniques auxiliaires, tel par exemple, l'agrégation des données, ont été élaborées et viennent en appoint pour améliorer l'efficacité de l'algorithme en étalant le trafic pour l'équilibrer uniformément sur le réseau. Sur ce dernier les nœuds font office de relais sur des chemins multiples ; Il est alors possible de combiner des entités pour agréger des données. D'un autre côté, trois techniques différentes d'étalement des données ont été proposées :

- Un schéma stochastique : qui consiste à choisir de façon aléatoire entre deux liens qui ont même gradient.
- Un schéma basé sur l'énergie : lorsque l'énergie d'un nœud chute au dessous d'une valeur seuil, ce nœud augmente sa hauteur pour augmenter le gradient.
- Un schéma basé sur le flot : l'idée est de dissuader les nouveaux flots pour les éloigner des nœuds qui font partie du chemin d'autres flots.

e. Le routage d'informations par la négociation

L'un des protocoles de routage opérant par la négociation est désigné par l'acronyme SPIN [81,82] (Sensors Protocols for Information via Negotiation). C'est l'un des premiers protocoles construits pour mettre en œuvre le mécanisme de routage centré-données. L'idée essentielle derrière le protocole SPIN est de nommer les données en utilisant des descripteurs de hauts niveaux ou des méta données. Avant la transmission, ces méta données sont échangées entre capteurs par un mécanisme d'annonce qui est la caractéristique clé du protocole SPIN.

A chaque réception d'une donnée nouvelle, le nœud l'annonce à ses voisins ou aux nœuds qui ont déjà exprimé leurs intérêts pour ce type de données. La figure(π) illustre le fonctionnement du protocole SPIN, on voit dans cette figure le mécanisme de propagation s'opérant par la négociation.

- Le nœud NA commence par annoncer les données à disposition au nœud NB.
- Le nœud NB émet une requête pour solliciter les données
- Le nœud NA livre alors les données à NB.
- Le nœud NB va alors réitérer l'action de NA, et alors de proche en proche les données vont se propager dans le réseau.

L'un des avantages du protocole SPIN tient au fait que les changements de topologie sont localisés, puisque chaque nœud a besoin de connaître uniquement les sauts de ces voisins. Le gain en terme de dissipation d'énergie est dans un rapport de 3.5 par rapport à un protocole qui inonde le réseau de façon non sélective. Le mécanisme de négociation parvient en général à réduire de presque de moitié les données redondantes. Toutefois, le mécanisme d'annonce du protocole SPIN ne peut pas garantir la livraison des données. Par exemple, si les nœuds qui ont de l'intérêt pour les données sont loin du nœud source et que les nœuds entre la source et la destination ne manifestent aucun intérêt pour les données. Dans cette situation défavorable, les

données ne seront jamais délivrées. Par conséquent, SPIN n'est pas considéré comme un bon choix, pour les applications intégrant la détection d'intrus, qui exigent une fiabilité en matière de livraison des paquets de données à intervalles réguliers.

f. Energy aware routing

Shah et Rabaey [83] ont proposé l'usage d'un ensemble de chemins sous-optimales de façon à augmenter occasionnellement la durée de vie du réseau. Ces chemins sont choisis en utilisant une fonction probabiliste qui dépend de la consommation d'énergie sur chaque chemin. Le critère de survie du réseau est considéré alors comme la métrique principale ciblée par ce type de protocoles.

Cette approche est basée sur l'argument de taille suivant:

Si on utilise toujours le chemin de faible consommation d'énergie, ceci aboutira fatalement à une diminution drastique des ressources d'énergie des nœuds appartenant à ce chemin. Il risque alors de devenir impraticable.

Pour éviter cette situation et accroître la durée de vie du réseau, il est alors préférable de router les données sur plusieurs chemins avec une certaine probabilité.

Ce protocole suppose que chaque nœud puisse faire l'objet d'un adressage basé sur une classe d'adresses qui contient les localisations et les types de nœuds. Il se déroule en 3 phases :

- La phase de préparation : elle consiste à localiser les chemins et à créer des tables de routage. Lors de cette phase, le coût total en énergie est calculé pour chaque nœud. Ainsi, si la requête est envoyé du nœud N_i vers le nœud N_j , ce dernier calculera le coût induit sur ce chemin par :

$$C_{N_i N_j} = \text{Coût}(N_i) + \text{Metric}(N_j, N_i)$$

Dans cette expression, la métrique de l'énergie utilisée comptabilisera la consommation à la transmission, à la réception et bien sûr l'énergie résiduelle des nœuds.

Au terme de ce calcul, les chemins présentant des coûts élevés seront bien évidemment écartés. La sélection du nœud est faite en fonction de la proximité à la destination.

Le nœud affecte une probabilité à chacun de ces voisins dans la table de routage correspondant à un chemin formé. La probabilité est inversement proportionnelle au coût.

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

Le nœud N_j calcule, alors le coût moyen, pour atteindre la destination, en utilisant les voisins dans la table d'envoi, et ceci en utilisant la formule :

$$Coût(N_j) = \sum_{i \in FT} P_{N_j, Ni} C_{N_j, Ni}$$

Ce coût moyen est fixé dans le champ coût de la requête

- **La phase de communication des données :**

Au cours de cette phase, chaque nœud procède à l'envoi par un choix aléatoire du nœud voisin, à partir de la table de routage en utilisant les probabilités. C'est ce choix aléatoire qui garantit que le chemin emprunté n'est pas le même à chaque fois et qui permettra de maintenir les énergies des nœuds à des niveaux acceptables permettant de prolonger la durée de vie du réseau.

- **La phase de maintenance des chemins :**

Pour s'adapter aux changements dans le réseau, les chemins sont ré explorés de temps à autre pour s'assurer que l'on garde toujours les meilleurs.

Des simulations ont été menées pour mesurer le gain en énergie et en allongement de la durée de vie. Ainsi, une comparaison avec l'algorithme DD montre un gain d'énergie de 21.5% et un allongement de la durée de vie de 44%.

- g. L'algorithme du coût minimal d'envoi « Minimum cost forwarding Algorithm », (MCFA)**

Cet algorithme [84] considère que la direction du routage est toujours connue (puits externe fixe), chaque nœud doit connaître le chemin de moindre coût qui le relie au puits. Il maintient le moindre coût estimé le reliant au puits. Chaque message envoyé par le nœud fait l'objet d'une large diffusion à ses voisins.

VI.3.1.2 Le routage hiérarchique

L'adaptation à l'augmentation de la taille du réseau et les capacités d'ajustement (scalability) sont des attributs fondamentaux visés en phase de conception du réseau. Un réseau à étage unique peut causer la surcharge des points d'entrée du réseau lorsque la densité des capteurs augmente. Ce type d'architecture ne permet pas l'adaptation à l'échelle pour un large ensemble de capteurs couvrant une grande aire d'intérêt, puisque typiquement les capteurs sont incapables de communication sur de longs parcours. Pour permettre au système de se débrouiller avec la charge additionnelle et d'être capable de couvrir un large domaine sans dégradation de service, la mise en grappes dans les réseaux (clustering) a été mise en œuvre dans les protocoles de routage. L'objectif principal du routage hiérarchique est de

maintenir de façon efficace la consommation d'énergie des nœuds en les impliquant dans des communications dans un cluster particulier et en procédant à l'agrégation et la fusion des données dans le but de diminuer la masse des messages transmis au puits. La formation du cluster est basée sur la réserve d'énergie des capteurs et de leurs proximités de la tête du cluster.

Des protocoles basés sur le routage hiérarchique ont vu le jour. L'un des tout premiers est LEACH, mais il y en a bien d'autres sur lesquelles nous allons nous arrêter.

a. LEACH « Low energy adaptation clustering hierarchy »

LEACH [85] est l'un des algorithmes de routage hiérarchique les plus populaires. L'idée qui gouverne le fonctionnement de LEACH est de former des grappes de nœuds de capteurs basés sur l'intensité du signal reçu et d'utiliser ensuite les têtes du cluster local comme des routeurs vers les puits. Ceci devrait évidemment économiser de l'énergie puisque les transmissions seront effectuées uniquement par les têtes des clusters plutôt que par tous les nœuds. Le nombre optimal de têtes de clusters est estimé à 5% du nombre total de nœuds.

Tous les traitements de données tel que la fusion et l'agrégation se déroulent localement au cluster.

Les têtes de cluster sont dynamiques, elles changent aléatoirement avec le temps pour équilibrer la dissipation d'énergie par les nœuds. Cette décision est prise par le nœud en choisissant un nombre aléatoire compris entre 0 et 1.

Le nœud devient alors tête de cluster pendant cette période si le nombre tiré au hasard est inférieur au seuil suivant :

$$T(n) = \begin{cases} p & \text{si } n \in G \text{ et } 0 < p < 1 \\ 0 & \text{autrement} \end{cases}$$

Où p est le pourcentage de clusterhead souhaité, r le tour courant, G l'ensemble des nœuds qui n'ont pas été tête de cluster au cours des $1/p$ tours précédents.

Sur le registre des avantages de LEACH, nous citons une réduction d'un facteur 7 de la dissipation d'énergie comparativement avec la communication directe et un facteur de 4 à 8 avec le protocole de routage à minimum de transmission. Parmi les autres caractéristiques de LEACH citons aussi le fait qu'il soit complètement distribué et qu'il n'exige pas la connaissance du réseau. A sa charge, cependant, son fonctionnement à saut unique, c'est-à-dire que chaque nœud peut transmettre

directement à la tête du cluster ou au puit. Par conséquent LEACH n'est pas applicable aux réseaux déployés sur de vastes régions. En outre, l'idée du clustering dynamique amène des frais supplémentaires; le changement de tête du cluster, les annonces, etc...vont diminuer l'économie d'énergie

b. PEGASIS et Hierarchical PEGASIS, « power efficient gathering sensor information systems »

PEGASIS [86] est une amélioration du protocole LEACH. Au lieu de former des clusters multiples, PEGASIS forme des chaînes à partir des nœuds de capteurs de sorte que chaque nœud transmet et reçoit de son voisin, et un seul nœud de la chaîne est sélectionné pour assurer la transmission vers la station de base. Les données rassemblées se déplacent d'un nœud à l'autre, elles sont agrégées et éventuellement envoyées vers la station de base. La construction de la chaîne se fait selon une approche gloutonne.

A la différence de LEACH, PEGASIS réalise un routage multi-sauts en formant des chaînes et en sélectionnant un nœud et un seul pour transmettre vers la station de base au lieu que cela se fasse avec plusieurs nœuds. Une comparaison avec LEACH montre que PEGASIS dépasse en performance LEACH de 100 à 300 % pour des réseaux de différentes tailles et de différentes topologies. Ce gain en performance est attribué à l'élimination des frais supplémentaires générés par le clustering dynamique et la réduction du nombre de transmissions et de réceptions en utilisant l'agrégation des données.

A coté de ces aspects positifs, mentionnons toutefois des retards excessifs pour des nœuds très distants sur la chaîne, ainsi que la possibilité d'apparition de goulots d'étranglements.

A coté de PEGASIS, Hierarchical PEGASIS [87] a vu le jour comme une extension du premier pour tenter de palier aux retards encourus durant la transmission vers la station de base et propose d'apporter une solution à l'accumulation des données en considérant la métrique énergie x retard. Afin de réduire les retards invoqués pour PEGASIS, des transmissions simultanées de messages sont exécutés.

Pour éviter les collisions et les interférences entre les signaux des capteurs deux approches ont été tentées. La première incorpore le codage du signal (CDMA) (code division multiple access). La seconde approche consiste à autoriser uniquement les nœuds spatialement séparés à transmettre en même temps.

Le protocole basé sur des chaînes avec le codage de signaux CDMA autorise la construction de chaînes de nœuds qui forme une structure hiérarchique en arbre. Chaque nœud sélectionné dans un niveau particulier, transmet les données au nœud se trouvant dans un niveau supérieur de la hiérarchie. Cette méthode assure la transmission parallèle et réduit significativement le retard. Puisque l'arbre est équilibré, le retard sera en $O(\log N)$ où N est le nombre de nœuds.

La seconde approche, n'utilisant pas le CDMA, crée une hiérarchie à trois niveaux entre les nœuds. Les effets d'interférence se trouvent réduits si on planifie avec attention les transmissions simultanées. Une amélioration d'un facteur 60 a été invoquée pour un tel protocole comparé à la forme de base de PEGASIS.

c. Les protocoles TEEN et APTEEN « Threshold sensitive Energy Efficient sensor Network protocole »,

TEEN [88] est un protocole hiérarchique conçu pour être sensible à des variations soudaines des attributs des grandeurs observées (la température par exemple). La sensibilité est fondamentale pour les applications où le temps revêt un caractère critique. Pour ces applications, on attend du réseau une réaction prompte. TEEN vise une approche hiérarchique avec un mécanisme centré-données. L'architecture du réseau de capteurs est réalisée en groupant hiérarchiquement les nœuds de sorte que les nœuds les plus proches forment des clusters et ce procédé va vers le niveau secondaire jusqu'à atteindre le puits. Une fois les clusters formés, la tête du cluster diffuse deux seuils pour les nœuds, l'un dit « hard » et l'autre « soft » pour les grandeurs observées. Le seuil « hard » est la valeur minimale possible d'un attribut qui fait déclencher l'émetteur du nœud capteur et lui fait transmettre un message à la tête du cluster. Ainsi, le seuil « hard » permet aux nœuds de transmettre seulement lorsque les grandeurs observées ont des valeurs dans un domaine précis, réduisant ainsi de façon significative le nombre de transmissions. Lorsqu'un nœud détecte une valeur supérieure ou égale au seuil « hard », il transmet les données uniquement quand la valeur de l'attribut change d'une valeur supérieure ou égale du seuil « soft ». Et comme conséquence, le seuil « soft » va lui aussi réduire le nombre de transmissions. On peut même ajuster les deux seuils de façon telle qu'on peut contrôler le nombre de paquets transmis. Cependant TEEN n'est pas un bon protocole pour les applications nécessitant des rapports périodiques puisque l'utilisateur ne reçoit aucune donnée si les seuils ne sont pas atteints.

L' « adaptative Threshold sensitive Energy efficient sensor Network protocole (APTEEN) [89] Est une mouture remodelée de TEEN, avec en plus la possibilité de disposer de collections périodiques de données en plus d'une réactivité accrue aux événements temporellement critiques. L'architecture est la même que pour TEEN. Lorsque la station de base forme les clusters, les têtes de cluster diffuse les attributs et les valeurs seuils et le planning des transmissions à tous les nœuds. Les têtes de cluster réalise en plus l'agrégation des données dans le but de sauvegarder l'énergie. APTEEN supporte trois types de requêtes différentes :

- Historique : pour analyser les données du passé
- Temps réel : pour avoir une vue instantanée du réseau
- Persistante : pour surveiller un évènement sur une période de temps

Des simulations sur TEEN est APTEEN ont montré qu'il surpasse en performance LEACH. Les expériences ont montré que les performances d'APTEEN se situent entre LEACH et TEEN en matière de dissipation d'énergie et de durée de vie du réseau. TEEN réalise de meilleures performances en raison du nombre réduit de transmissions.

d. Routage par architecture en grille virtuelle :

Un paradigme de routage ciblant l'efficacité de l'énergie a été proposé. Il utilise l'agrégation des données et des traitements intra-réseau pour optimiser sa durée de vie. A cause de la stationnarité des nœuds et d'un niveau de mobilité extrêmement faible dans plusieurs applications utilisant les WSN.

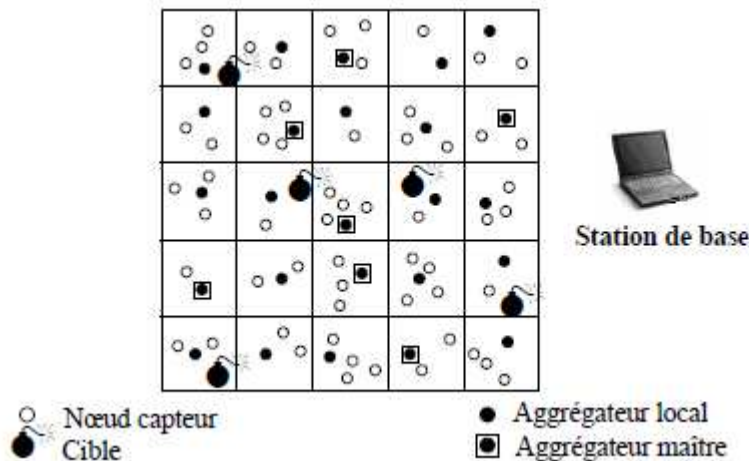


Figure VI.6 : Exemple pour les zones fixe pour le routage VGAR [96]

A l'intérieur de chaque zone, un nœud est sélectionné de façon optimale pour agir comme tête de cluster avec pour mission l'agrégation et la transmission des

données. En raison de l'architecture figée il est possible d'implémenter des stratégies spécifiques pour l'agrégation des données.

VI.3.1.3 Le routage basé sur la localisation

Il y a quelques protocoles de routage basé essentiellement sur la localisation :

- Le réseau de communication à petit minimum d'énergie.(small minimum Energy communication Network SMECN)
- Le routage par fidélité géographique adaptative (Geografic Adaptative fidelity)
- Routage géographique énergie (geografic and energy aware routing)

a. Le routage MECN

c'est un protocole qui a été proposé [90] en vue de calculer un sous réseau pour lequel l'énergie est efficiente, ce protocole (MECN) a été appliqué pour trouver le minimum d'énergie pour un certain réseau de capteurs utilisant un GPS à basse puissance. Le MECN identifie alors une région relais pour chaque nœud. La région « relais » est constituée de l'ensemble des nœuds d'un environnement où la transmission à travers ces nœuds est plus efficiente énergétiquement que la transmission directe. L'enclos pour un nœud i est créé en faisant la réunion de toutes les régions relais que le nœud i peut atteindre.

L'idée principale du MECN est de trouver un sous réseau possédant un nombre limité de nœuds et exigeant moins de puissance pour assurer la transmission entre n'importe quelle paire de nœuds. De cette manière, des chemins ayant un minimum global de puissance sont trouvés, sans qu'il y ait une nécessité de considérer tous les nœuds du réseau. Ceci est réalisé en procédant à une recherche locale pour chaque nœud en explorant sa région relais.

MECN est un protocole qui s'auto-reconfigure, il peut donc s'adapter dynamiquement en cas d'échec d'un nœud ou de déploiement de nouveaux nœuds. SMECN [91] est une extension de MECN. Alors que dans MECN, il est supposé que tout nœud est en mesure de transmettre aux autres, ce qui n'est pas toujours possible ; dans SMECN les obstacles possibles entre une paire de nœuds quelconque sont considérés. Toutefois, comme dans le cas de MECN, le réseau est supposé pleinement connecté. Le sous-réseau construit par SMECN pour le minimum d'énergie est plus petit en terme de contours que celui obtenu par MECN. Aussi, faut il noter, le fait que le sous-réseau obtenu par SMECN est plus petit que celui trouvé par MECN si la région de diffusion est circulaire autour du nœud émetteur pour une puissance donnée. En utilisant, le langage de la théorie des graphes, on dira que le sous graphe G' du

graphe G représentant le réseau de capteurs minimise l'utilisation de l'énergie dans les conditions suivantes :

- Le nombre de contours dans G' est inférieur à celui dans G .
- L'énergie requise pour transmettre des données d'un nœud à tous ses voisins dans le sous graphe G' est plus petite que l'énergie nécessaire pour transmettre à tous ses voisins dans G . supposons que $r = (u, u_1, \dots, v)$ soit un chemin entre u et v Traversant les $(k-1)$ nœuds intermédiaires u_1, u_2, \dots, u_{k-1} .

La puissance totale consommé sur un chemin comme r est donnée par

$$C(r) = \sum_{i=0}^{k-1} (p(u_i, u_{i+1}) + c)$$

où $u = u_0$ et $v = u_k$, la puissance requise pour la transmission dans ce protocole est :

$$p(u, v) = td(u, v)^n$$

Avec une constante appropriée t , n est un exposant exprimant les pertes lors de la propagation des ondes radio en plein air. Cet exposant est en général supérieur ou égal à 2, $d(u, v)$ est la distance entre u et v et c est la puissance consommée lors de la réception.

A coté des avantages de cet algorithme que nous avons mentionné de façon implicite, on peut lui reprocher sa localité dans la mesure où il ne trouve pas le chemin du minimum d'énergie. C'est, tout juste s'il construit un sous réseau dans lequel il existe un chemin de minimum d'énergie. En plus, la construction d'un sous réseau avec un petit nombre de contours n'est pas une opération sans frais, elle génère inévitablement une consommation supplémentaire.

b. Le protocole GAF

C'est un protocole [92] conçu initialement pour des réseaux ad hoc de mobiles, mais qui peut très bien s'appliquer aux réseaux de capteurs. La zone du réseau est divisée, en premier lieu, en zones fixes, formant ainsi une grille virtuelle. A l'intérieur de chaque zone, les nœuds vont collaborer pour jouer différents rôles. Par exemple, ils peuvent élire un capteur pour rester éveillé une certaine période de temps au cours de laquelle ils se mettent en état de sommeil. Le nœud élu est alors responsable au nom des noeuds de la zone de la veille et des rapports de données transmis à la station de base. Ainsi donc, le GAF conserve l'énergie en mettant en congé les nœuds qui ne sont pas nécessaires au réseau pendant un certain temps, sans affecter le niveau de fidélité du routage. Chaque nœud utilise ses coordonnées GPS pour se constituer comme un point de la grille virtuelle. Les nœuds associés au même point sont

considérés comme équivalents en matière de coût de routage des paquets. Cette équivalence est alors exploitée pour mettre en sommeil quelques nœuds localisés dans un secteur particulier de la grille avec évidemment le souci de sauvegarder l'énergie et la prolongation de la durée du réseau. Il y a trois états différents dans GAF, découverte, activité (reflétant la participation au routage) et sommeil. Afin de gérer la mobilité, chaque nœud de la grille estime le temps permis pour quitter la grille et envoie cette estimation à ses voisins. Les voisins ajustent la durée du sommeil avec l'impératif de maintenir la fidélité du routage. La relève entre équipes de capteurs se fait avec le souci de l'efficacité et du maintien de la qualité du service. Ainsi avant que les nœuds actifs ne quittent (se mettent en sommeil), les nœuds en sommeil se réveillent et l'un d'eux devient immédiatement actif.

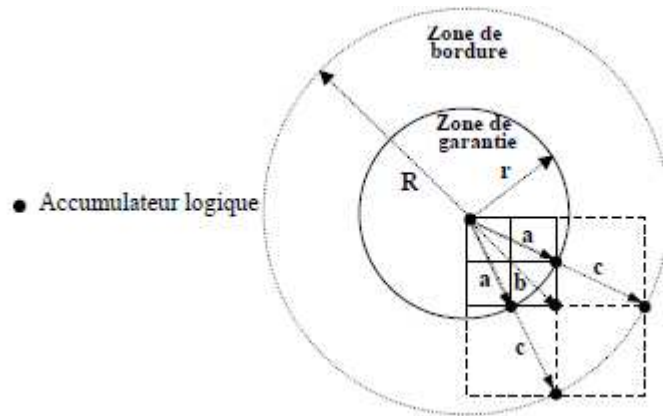


Figure VI.7 : Un exemple de découpage en zone dans un réseau de capteurs [95]

GAF a été implémenté aussi bien pour un contexte de non mobilité (GAF_basic) et en situation de mobilité (GAF-mobility adaptation) des nœuds. Les résultats des simulations ont montré que GAF est au moins aussi performant que les protocoles de routage ad hoc en terme de latence et de perte de paquets et qu'à qualité de service égale il permet de prolonger la durée de vie.

Par rapport à la taxonomie que nous avons évoqué, GAF peut être vu, aussi bien comme un protocole basé sur la localisation que comme un protocole à caractère hiérarchique, Entendu dans ce cas que le « clustering » se fait sur la base de la localisation géographique. Pour chaque secteur particulier de la grille un nœud représentatif agit comme leader pour garantir la transmission aux autres nœuds, mais il ne fait ni agrégation ni fusion, comme dans le cas des protocoles hiérarchiques que nous avons préalablement invoqués.

c. Le « geographic and Energy Aware Routing »(GEAR)

Yu et Al [93] ont discuté et argumenté l'utilisation de l'information géographique pendant la dissémination des requêtes dans les régions appropriées puisque les requêtes de données incluent les attributs géographiques.

Le protocole GEAR utilise l'énergie et informe géographiquement les voisins sélectionnés sur une base heuristique pour router le paquet vers la région de destination. L'idée de base est de restreindre le nombre d'intérêts, que nous avons évoqué avec le protocole de diffusion guidé, en ne considérant qu'une certaine région au lieu d'envoyer à l'intégralité du réseau. En précédent, de la sorte, GEAR réussit à conserver l'énergie beaucoup plus que ne le fera la diffusion dirigée.

Chaque nœud dans GEAR garde un coût estimé et un coût appris (mesuré par apprentissage) pour atteindre la destination à travers les voisins. Le coût estimé est une combinaison de l'énergie résiduelle et de la distance le séparant de la destination. Le coût appris est un raffinement du coût estimé comptabilité pour le routage autour des trous dans le réseau. Par trou, nous entendons le schéma qui se produit lorsque un nœud n'a de voisin proche dans la région cible autre que lui-même. En cas d'absence de trous, le coût estimé s'identifie alors au coût appris. Ce dernier est propagé d'un saut en arrière à chaque fois qu'un paquet atteint la destination de sorte que le routage du prochain paquet y sera ajusté. Cet algorithme GEAR se déroule en deux phases :

Expédition des paquets à travers la région cible

À la réception d'un paquet, un nœud vérifie ses voisins pour voir s'il n'en existe pas un qui soit plus proche que lui de la région cible. S'il en existe plus qu'un, le voisin le plus proche de la région cible est sélectionné comme étant le prochain saut. Si, tous s'avèrent plus loin que lui, cela signifie la présence d'un trou. Dans ce cas l'un d'eux est trié sur le volet pour se charger de la transmission en se basant sur le coût appris.

Ce choix, peut être, mis à jour conformément à la convergence du coût appris au cours de la livraison des paquets.

L'envoi des paquets à l'intérieur de la région

Si le paquet a atteint la région, il peut faire l'objet de diffusion dans cette région soit par un envoi géographique récursif ou par inondation restreinte. Cette dernière, est bien indiquée, lorsque les capteurs ne sont pas disséminés avec une forte densité. Dans les réseaux à forte densité, la

transmission géographique récursive est plus efficace en énergie que l'inondation restreinte.

Au cours de comparaisons entre protocoles par simulation, GEAR a été confronté aux performances de l'un des protocoles de routage géographique fonctionnant sur un paradigme non orienté energy_aware GPRS utilisant des graphes plans pour résoudre le problème des trous. Dans GPRS, les paquets suivent le périmètre d'un graphe plan pour trouver leurs itinéraires. Bien que GPRS réduit le nombre d'états qu'un nœud peut prendre, il a cependant été conçu pour les mobiles en général et les réseaux ad hoc et il nécessite un service de localisation pour représenter les localisations par des identifiants de nœuds. GEAR réduit non seulement la consommation d'énergie, mais aussi réalise de meilleures performances que GPRS à la livraison. Les simulations montrent que pour des distributions irrégulières du trafic GEAR livre 70 à 80% plus de paquets et pour des trafics uniformes 25 à 35 % de plus.

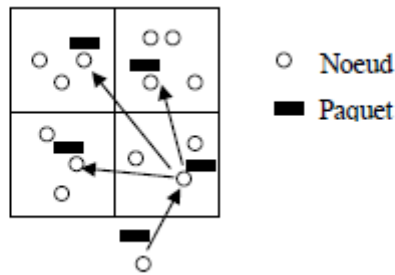


Figure VI.8: Transmission géographique récursive dans GEAR

Nous terminons avec les protocoles basés sur la localisation en mentionnant l'existence d'un protocole nommé SPAN basé lui aussi sur les positions. SPAN opère en sélectionnant des nœuds qui sont érigés en coordinateurs. Ce réseau de coordinateurs constituera la dorsale pour la transmission des messages.

VI.3.2 Les protocoles de routage en fonction de leurs modes opératoires

Nous allons maintenant examiner les protocoles de routage sous l'angle de leurs modes opératoires. L'opération du protocole inclut, à proprement parler, plusieurs fonctionnalités relatives au routage.

VI.3.2.1 Protocoles de routage multichemins

C'est l'ensemble des protocoles qui utilisent des chemins multiples, au lieu d'un seul chemin, pour améliorer la performance du réseau. La tolérance aux fautes d'un protocole est mesurée par la probabilité d'existence d'un chemin alternatif entre la source et la destination lorsque le chemin primaire a échoué. Cette tolérance peut être accrue en maintenant une

multiplicité de chemins entre la source et la destination au dépend de l'augmentation de la consommation d'énergie et de l'augmentation du trafic. Ces chemins alternatifs sont entretenus par des envois de messages et procurent une fiabilité supplémentaire au dépend d'une rallonge des frais. Parmi les approches qui ont été proposées pour améliorer le routage multi-chemins celle qui consiste à router les données à travers les nœuds du chemin possédant l'énergie résiduelle la plus large et de basculer vers un autre chemin dès que ce dernier se révèle meilleur. En adoptant cette stratégie, les nœuds du chemin primaire ne vont pas épuiser continûment leurs ressources en énergie en utilisant toujours le même chemin. Une approche alternative consiste à utiliser occasionnellement un ensemble de chemins sous optimaux pour augmenter la durée de vie. Ces chemins sont choisis sur une base probabiliste qui identifie parmi les chemins sous optimaux celui qui offre la consommation minimale. Le chemin avec le maximum d'énergie résiduelle peut s'avérer très coûteux lorsqu'il est utilisé pour router les données, il s'agit alors de trouver un compromis entre la minimisation de l'énergie totale consommée et l'énergie résiduelle du réseau. Certains auteurs ont proposé des types de routage multi-chemins en vue d'augmenter la fiabilité des WSN. Le schéma proposé est utile pour délivrer les données dans des environnements peu sûrs voire même hostiles. Nous avons déjà mentionné le fait que la fiabilité peut être renforcée de façon notable en multipliant le nombre de chemins entre la source et la destination et en envoyant le même paquet de données sur ces différents chemins. Cependant, utiliser cette technique génère inévitablement du trafic supplémentaire. On doit alors réfléchir à un compromis entre la quantité de trafic et la fiabilité ciblée du réseau. Ce compromis a été étudié avec l'aide d'une fonction de redondance qui dépend de la multiplicité des chemins et des probabilités de défaillance des chemins disponibles. L'idée est de partitionner le paquet original de données en sous paquets et de les envoyer chacun sur les chemins disponibles. La diffusion dirigée est un bon candidat pour le routage robuste et la livraison multi-chemins.

VI.3.2.2 Le routage basé sur les requêtes

Dans ce type de routage, les nœuds de destination propagent une requête pour des données à partir d'un nœud à travers le réseau. Un nœud disposant de ces données va les livrer au nœud initiateur de la requête. Habituellement ces requêtes sont décrites en langage naturel ou en langage de requêtes évolué, à titre d'exemple on peut imaginer que sur le théâtre d'une bataille, un client C1 soumette une requête au nœud N1 pour l'interroger sur l'existence ou non de véhicules mobiles dans une région du théâtre des opérations. Tous les nœuds désignés pour détecter des véhicules mobiles répondront à la requête dès sa réception. La diffusion guidée que nous avons déjà expliquée est un exemple de ce type de protocoles. En

effet la station de base va envoyer des messages exprimant des intérêts spécifiques aux différents nœuds capteurs, et alors que ces intérêts sont propagés à travers les nœuds de capteurs, les gradients de la source vers la station de base sont établis. Lorsque la source dispose des données ciblées elle procédera à leurs livraisons le long du chemin du gradient. Pour faire baisser la consommation d'énergie, l'agrégation des données est opérée au moment du routage.

Le protocole de routage par la rumeur utilise un ensemble d'agents de grande longévité pour créer des chemins dirigés vers les événements qu'ils rencontrent. Chaque fois qu'un agent traverse un chemin menant à un événement qu'il n'a pas encore rencontré, il crée un état indiquant comment aboutir à cet événement. Lorsque les agents traversent des chemins plus courts ou plus efficaces, ils optimisent en conséquence la table de routage.

Chaque nœud conserve une liste de ses voisins et une table d'évènements qui est mise à jour chaque fois que de nouveaux événements sont rencontrés. Chaque nœud peut aussi générer un agent de façon probabiliste contenant une table d'évènements synchronisés à chaque nœud visité. L'agent a une durée de vie de quelques sauts avant son extinction (sa mort).

Un nœud ne génère une requête que s'il connaît un itinéraire vers l'évènement en question. S'il n'y a aucun itinéraire disponible, le nœud transmet une requête dans une direction aléatoire. Ensuite le nœud se met en attente pour savoir si la requête a atteint la destination selon un certain délai; après quoi il procède à l'inondation du réseau si aucune réponse ne parvient de la destination.

VI.3.2.3 Routage basé sur la négociation

Ces protocoles utilisent des descripteurs de données de haut niveau afin d'éliminer la redondance des transmissions des données par le biais de la négociation.

Les décisions de communication sont prises conformément aux ressources qui leurs sont destinées. La famille de protocoles désignés par SPIN que nous avons évoqués précédemment sont des exemples de protocoles basés sur la négociation. La motivation est que le recours à l'inondation pour disséminer les données produira l'implosion et le chevauchement entre les données transmises, si bien que les nœuds vont recevoir des copies répliquées des mêmes données. Cette opération s'avère évidemment coûteuse en énergie. Les protocoles SPIN sont conçus pour disséminer les données d'un capteur à tous les autres en supposant que ces capteurs sont potentiellement des stations de base. Donc, l'idée principale du routage basé sur la négociation est de supprimer les informations répliquées et prévenir la redondance des données lors de l'envoi au prochain capteur ou à la station de base en

conduisant une série de messages de négociations avant le prélude de la transmission réelle des données.

VI.3.2.4 Routage basé sur la qualité de service

Dans les protocoles basés sur la qualité de service, un équilibre entre la consommation d'énergie et la qualité des données doit être trouvé. En particulier, le réseau doit satisfaire certaines métriques qui quantifient la qualité de service (retard, énergie, largeur de bande, etc...) lorsqu'il délivre les données à la station de base.

Le routage par affectation séquentielle est l'un des premiers protocoles dédiés aux WSN qui a introduit la notion de qualité de service dans les décisions de routage.

Une décision de routage dans le protocole de routage par affectation séquentielle dépend de trois facteurs:

- Les ressources d'énergie.
- La qualité de service sur chaque chemin.
- Le niveau de priorité du paquet.

Pour éviter l'échec qui serait dû au chemin unique, une approche multi-chemins et localisation de chemins restaurés est utilisée. Pour créer des chemins multiples d'un nœud source aux nœuds de destination, un arbre qui tire sa racine du nœud source et qui s'étend aux nœuds de destination est construit. Les chemins sur cet arbre sont construits pendant que les nœuds avec une basse énergie sont évités avec garantie de la qualité de service. Au terme de ce processus, chaque nœud de capteur devient partie intégrante des multi-chemins sur l'arbre.

A ce titre, le protocole de routage par affectation séquentielle est un protocole à chemin multiple entraîné par une table qui vise une consommation efficace d'énergie et une tolérance aux fautes. Ce protocole calcule une métrique de qualité de service pondérée formée par le produit de QOS additive et d'un coefficient de pondération associé au niveau de priorité du paquet. L'objectif au final est de minimiser la métrique que nous venons d'évoquer pour augmenter la longévité du réseau. Si la topologie change à cause de la défaillance de certains nœuds, le recalcul des chemins est nécessaire. Comme mesure préventive, un recalcul périodique des chemins est déclenché par la station de base pour tenir compte de tout changement de topologie. Une procédure basée sur un schéma de restauration locale entre nœuds voisins est utilisée pour se remettre d'un échec. La reprise après échec est réalisée en forçant la consistance de la table de routage en amont et en aval des nœuds sur chaque chemin.

Les résultats de simulation montrent que le protocole d'affectation séquentielle offre moins d'énergie consommée que l'algorithme basé sur la métrique du minimum d'énergie qui

est focalisé exclusivement sur la consommation d'énergie des paquets sans considération de priorité.

VI.3.2.5 Processus cohérents et non cohérents

Le traitement des données est un composant majeur dans les opérations menées par les réseaux de capteurs. Les techniques de routage emploient différentes techniques de traitement des données. En général, les nœuds de capteurs collaborent les uns avec les autres au cours du traitement des données qui inondent le réseau. Deux exemples de techniques de traitements de données pour les WSN ont été proposés. La première est qualifiée de cohérente, alors que la seconde est dite non cohérente. Pour la technique non cohérente, les nœuds traitent localement les lignes de données avant leur envoi à d'autres nœuds pour un traitement supplémentaire. Les nœuds qui réalisent des traitements supplémentaires sont appelés des agrégateurs. Pour la technique dite cohérente, les données sont envoyées aux agrégateurs après un traitement minimal.

L'approche non cohérente manifeste un trafic moins important. Elle se déroule en trois étapes :

- Détection de la cible, collecte des données et prétraitement.
- Déclaration de membre.
- Election du nœud central.

Au cours de la première phase, une cible est détectée, ses données sont collectées et prétraitées. Lorsque un nœud décide de participer à une entreprise collaborative, il entre en phase 2 et annonce cette intention à tous ses voisins. Ceci doit se faire le plus tôt possible, de sorte que chaque capteur ait une compréhension locale de la topologie du réseau. Il s'ensuit alors la phase 3 qui consiste à élire le nœud central. Ce dernier, à la vue des missions qui lui sont confiées doit disposer d'information plus sophistiquées sur les traitements, des performances de calcul plus importantes et des réserves étendus d'énergie.

Sohrabi et Pottie [94] ont proposé deux algorithmes, le premier dit du gagnant unique et le second des multiples gagnants pour simuler respectivement le traitement non cohérent. Au sujet du premier, un seul agrégateur est dédié aux traitements complexes. L'élection d'un nœud est basée sur ses réserves énergétiques et ses capacités de calcul. Au terme du processus du gagnant unique un minimum de sauts s'étendant sur l'arbre couvrira complètement le réseau. Le processus des « multiples gagnants » est une extension de la première technique. Lorsque tous les nœuds sont des sources et qu'ils envoient leurs données au nœud faisant office d'agrégateur central, une grande quantité d'énergie est consommée, ce qui rend ce

processus à haut coût. Une manière de réduire le coût en énergie est de limiter par exemple le nombre de sources autorisées à transmettre les données vers l'agrégateur central.

VI.4 Les perspectives du routage dans les WSN

En regardant les choses de plus près, lorsque nous avons procédé à la classification des protocoles de routage, nous constatons que certains protocoles ont un caractère hybride et peuvent aussi bien cadrer avec les spécificités de plusieurs catégories. La raison est que leurs concepteurs ont tenté de bénéficier simultanément des avantages de plusieurs catégories.

Devant cette panoplie de protocoles, on pourrait se demander si tout est résolu, qu'il n'y a nul besoin pour améliorer les choses. En réalité, des défis restent posés, des optimisations sont encore possibles et des algorithmes plus performants doivent impérativement voir le jour. Voici quelques pistes évoquées par les chercheurs et qui doivent faire l'objet d'une prospection plus profonde.

- **Exploitation de la redondance** : Les nœuds dans un réseau sont enclin à subir des échecs lors de l'exercice de leurs fonctions. Les techniques de routage qui emploient la tolérance aux fautes de manière efficiente restent dans le domaine de l'investigation.
- **Architectures hiérarchisées** : Le routage hiérarchique est une ancienne technique qui améliore l'adaptabilité à l'échelle et l'efficacité du protocole de routage. Toutefois, les nouvelles techniques employant l'agrégation du réseau pour maximiser la durée de vie du réseau sont des domaines chauds de recherche.
- **Exploitation de la diversité spatiale et de la densité des nœuds** : Les nœuds devraient embrasser l'aire d'action définie pour le réseau pour permettre la communication entre capteurs. Réaliser des communications énergétiquement efficaces dans un environnement densément peuplé est un objectif qui mérite qu'on s'y penche. Le déploiement dense de nœuds de capteurs devrait permettre au réseau de s'adapter à un environnement imprévisible.
- **Comportement global** : La réalisation du comportement désiré avec des algorithmes localement adaptatifs dans un environnement dynamique est sans doute difficilement modélisable.
- **Auto-configuration et reconfiguration** : sont des nécessités pour la durée de vie de réseau non soumis à la surveillance dans un environnement dynamique subissant la contrainte énergétique. A partir de l'instant où des nœuds meurent et quittent le réseau, les mécanismes de reconfigurations doivent entrer en action.

- **La sécurité du routage** : Les protocoles que nous avons évoqués ont été conçus avec le souci majeur de l'efficacité mais pas spécialement de la sécurité, qui n'a pas été prise sérieusement en compte. Il serait intéressant de les examiner et de les analyser sous l'angle de la sécurité.

VI.5 Conclusion

Le routage dans les réseaux de capteurs est un domaine de recherche d'une grande fertilité. Nous avons tenté dans ce chapitre de faire une description globale des techniques en usage dans ce domaine, ayant comme objectif commun l'augmentation de la durée de vie du réseau, sans compromettre la livraison des données.

Globalement, la classification des techniques de routage basée sur la structure du réseau distingue trois catégories : plat, hiérarchique et orientée localisation. En outre, ces protocoles se subdivisent en multi-chemins, basé sur la requête, basé sur la négociation et enfin sur la qualité de service lorsqu'ils sont examinés sous l'angle du mode opératoire. Enfin, bien que tous ces protocoles se révèlent prometteurs, ils ne répondent cependant pas à tous les défis que rencontrent ceux qui les déploient.

VII.1 Introduction

L'évaluation des performances d'un réseau de capteur sans fil nécessite une phase de simulation avant leur déploiement dans l'environnement réel. Le paradigme multi- agents constitue un choix incontournable pour la modélisation des réseaux de capteurs sans fil, ceci grâce à son caractère distribué et dynamique pour la réalisation des systèmes complexes, en l'occurrence dans le cas des WSNs. Ces réseaux possèdent un grand nombre d'entités (nœuds) mettant en œuvre de nombreuses interactions (communication entre les nœuds, transferts de données...etc) dans un environnement dynamique et ouvert, et collaborant à la réalisation d'un objectif commun ; Ils ne peuvent de ce fait, faire l'objet d'un contrôle centralisé. Notre travail consiste à implémenter une plateforme d'aide à la conception et à la simulation des réseaux de capteurs sans fil basée agents. Cette plateforme a comme caractéristique la possibilité de modéliser les différents types de noeuds du réseau et les diverses architectures de réseaux ainsi que la possibilité de tester les différents protocoles de communication.

VII.2 Description générale de l'approche proposée

Dans notre approche le réseau de capteurs sans fil est constitué par trois types de nœuds : objet cible, capteur et station de base. Chaque type est représenté par un agent qui a ses propres caractéristiques et son propre comportement. Ces nœuds interagissent entre eux par le moyen d'envoi de messages. L'agent objet cible envoie des messages de type « stimulus », lorsque les agents capteurs reçoivent ces stimulus, ils collaborent ensemble par un routage multi sauts pour atteindre un objectif commun, qui est la collecte des données pertinentes au niveau des agents stations de base.

VII.3 Les services fournis par la plateforme de simulation

L'interface utilisateur de la plateforme donne la possibilité aux utilisateurs de faire plusieurs activités :

➤ **La description des nœuds et de leurs composants :**

Elle offre à l'utilisateur la possibilité de définir les caractéristiques des composants physiques qui constituent le nœud et la simulation du comportement de ces composants.

➤ **La description des protocoles :**

Elle donne à l'utilisateur la possibilité de définir un protocole de communication soit de la couche de routage soit de la couche MAC par la description de la structure du paquet et la définition des opérations d'encapsulation/décapsulation de chaque couche.

➤ **La description du réseau :**

Elle permet aux utilisateurs de construire leurs réseaux, par l'indication du nombre des nœuds, leurs positions, la localisation des objets cibles...etc.

➤ **La définition des agents :**

Elle consiste à former la liste des agents par la définition de leurs caractéristiques et de leurs comportements. Les caractéristiques en question sont les paramètres physiques du nœud (les composants qui le constituent et leurs caractéristiques). On attribue aussi le type de comportement souhaité en associant la définition des protocoles de communication que chaque nœud doit posséder suivant le rôle qu'il doit jouer dans le réseau (une station de base, un objet cible, un nœud capteur).

➤ **Exécution et statistiques :**

La plateforme doit faire des exécutions par la simulation du comportement de chaque nœud et par voie de conséquence de tout le réseau, et finalement aboutir à l'affichage des statistiques de performance (énergie consommée, nombre de paquets échangés, densité moyenne, nombre de voisins d'un nœud...etc.).

VII.4 Description détaillée de l'approche proposée

L'objectif principal des réseaux de capteurs sans fil est de surveiller et capter des événements d'intérêts dans un environnement spécifique. Quand un événement d'intérêt est détecté (par exemple, changement du bruit acoustique, séismique, ou de la température), les noeuds capteurs font un rapport aux noeuds puits (station de base /utilisateur) périodiquement ou sur demande. Les événements (nommés stimulus) sont produits par des noeuds cibles.

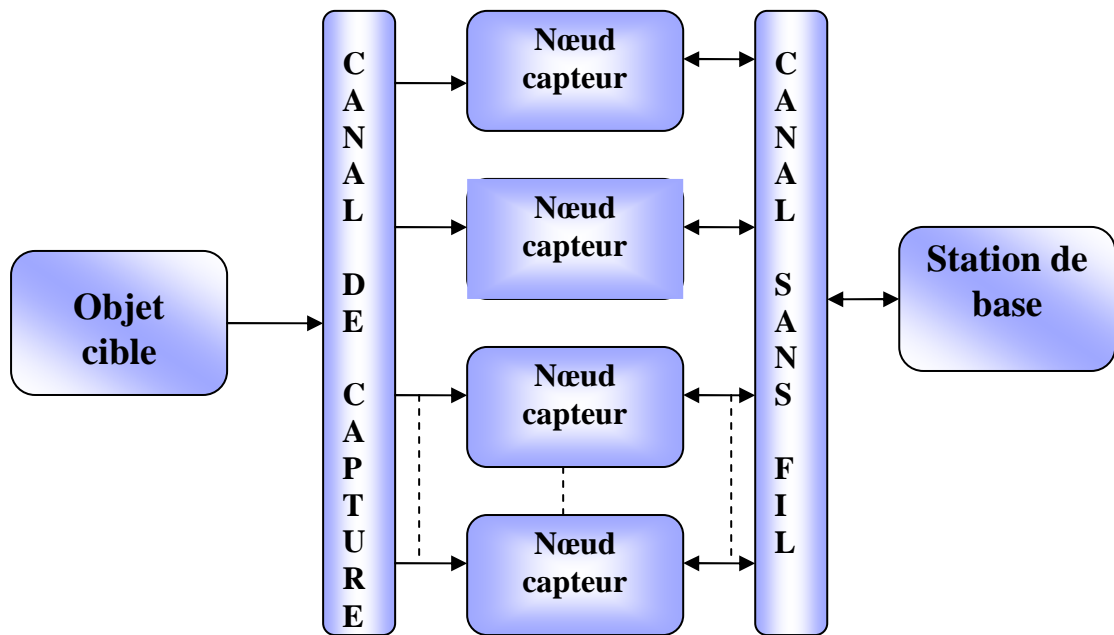


Figure VII.1 : architecture d'un réseau de capteur sans fil.

Par exemple, un véhicule mobile peut produire de petites vibrations en surface qui peuvent être détectées par les capteurs sismiques. Dans la perspective de la simulation, un réseau de capteurs sans fil est composé typiquement de trois types de nœuds ; Les nœuds capteurs qui détectent les événements d'intérêts, les nœuds cibles ou objets cibles qui produisent des événements d'intérêts/stimuli, et enfin, les nœuds puits ou station de base qui utilisent et manipulent l'information captée.

Dans notre plateforme de simulation, les nœuds capteurs détectent les signaux relatifs aux stimulus qui sont produits par les nœuds cibles à travers le canal de capture et font aboutir l'information détectée aux nœuds puits à travers un canal sans fil. Il est important de signaler que la nature de la propagation du signal entre les nœuds cibles et les nœuds capteurs à travers le canal de capture est différente de celle entre les nœuds capteurs et les nœuds puits à travers le canal sans fil. Deux modèles différents pour la propagation du signal sont donc inclus : un modèle de propagation de capteur et un modèle de propagation sans fil. Un nœud capteur est doté de :

- ❖ une pile de protocole de capteur, qui lui permet de détecter des signaux produits par des nœuds cibles à travers le canal de capture.

- ❖ une pile de protocole sans fil, qui lui permet d'envoyer des rapports aux autres noeuds capteurs (et par la suite aux stations de base) à travers le canal sans fil.

Un noeud capteur a également *un modèle d'énergie* qui représente les composants de production d'énergie (par exemple la batterie) et les composants consommateurs d'énergie (par exemple radio et unité centrale de traitement).

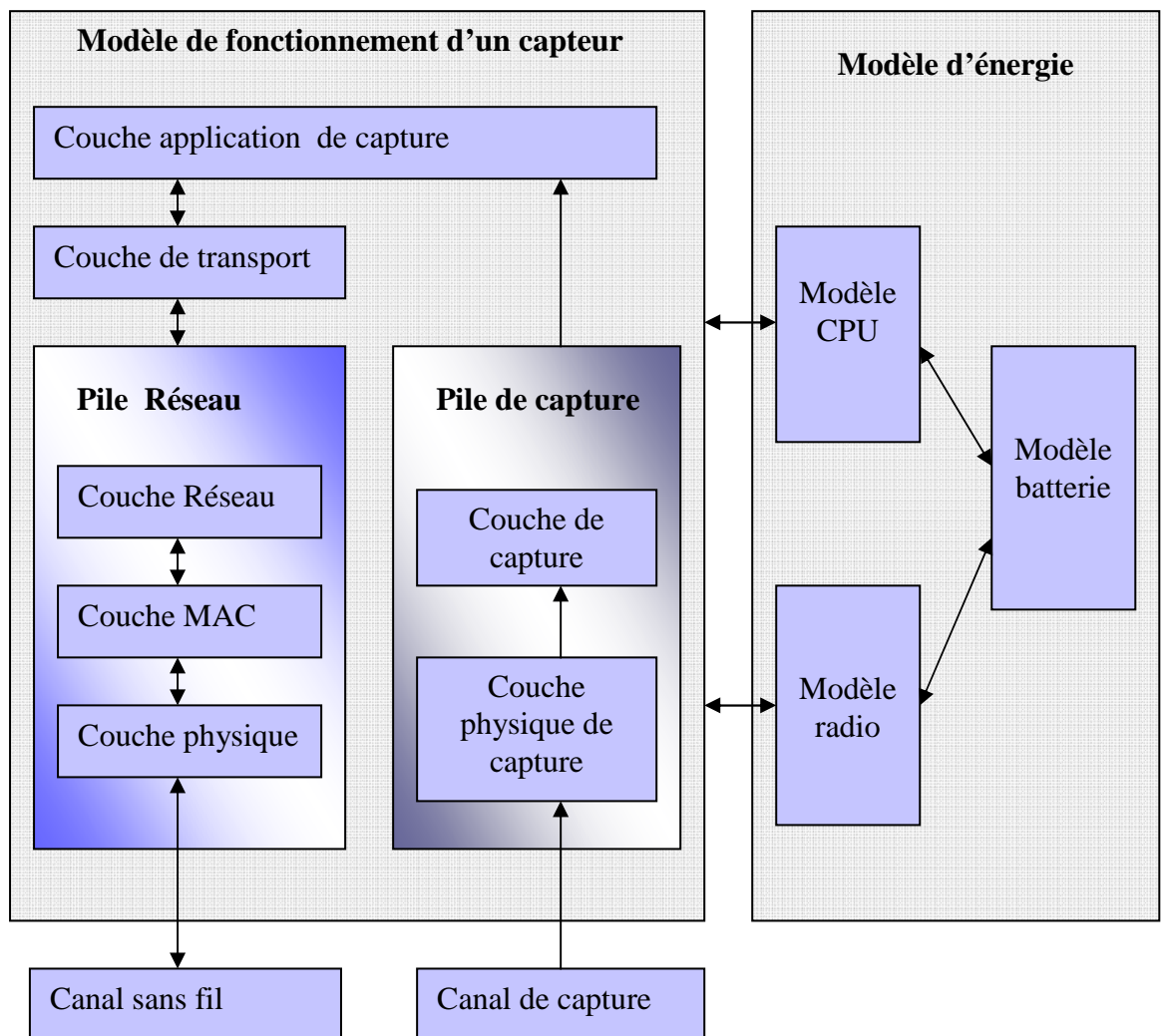


Figure VII.2 : architecture d'un nœud capteur

Un noeud cible a seulement une pile de protocole de capture, et un noeud puits a seulement une pile de protocole sans fil. En conclusion, afin de permettre la simulation des noeuds mobiles, un *modèle de mobilité* est inclus.

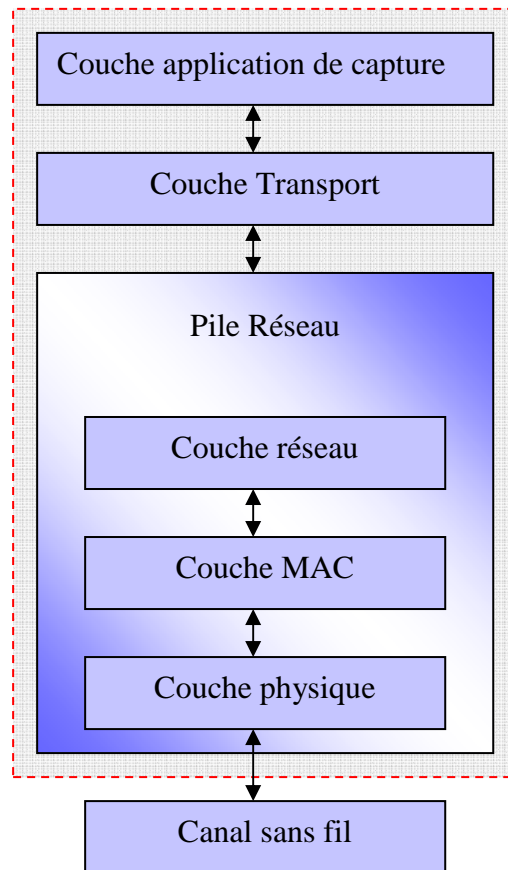


Figure VII.3 : architecture d'un nœud puits ou station de base

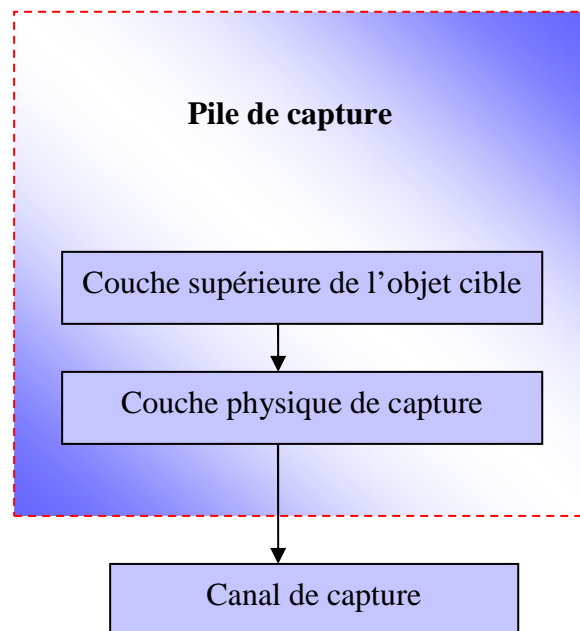


Figure VII.4 : architecture d'un nœud cible

Le fonctionnement de l'environnement de simulation de réseaux de capteurs sans fil proposé peut être illustrée en considérant un protocole de transport de l'événement au puits (station de base), un stimulus est périodiquement produit par un objet cible et propagé à travers le canal de capture. Un noeud cible peut seulement envoyer (mais ne peut pas recevoir) des paquets de données à travers le canal de capture. Les nœuds capteurs voisins (par exemple, les noeuds capteurs qui sont dans la même zone de capture du noeud cible) reçoivent le stimulus à travers le canal de capture. Un noeud capteur peut seulement recevoir (mais ne peut pas envoyer) des stimulus à travers le canal de capture. Cependant, étant donné que le signal peut être atténué au cours de la propagation à travers le canal de capture, un noeud capteur reçoit et détecte un stimulus seulement si la puissance du signal reçu est au moins égale à un seuil de réception prédéterminé. Le calcul de la puissance du signal reçu est déterminé par le modèle de propagation de signal capté utilisé.

Chaque noeud capteur qui reçoit et détecte le stimulus à travers le canal de capture doit envoyer le résultat capté à un ou plusieurs noeuds puits en utilisant le canal sans fil. À l'intérieur d'un noeud capteur, la coordination entre la pile de protocole de capture et la pile de protocole sans fil est faite par un *coeur agent*. Le réseau de capteur fonctionne selon les spécifications de l'application, un nœud capteur peut envoyer les paquets de données d'un noeud à l'autre dès qu'il détecte des événements, ou d'abord les traite (par exemple, calcule la température moyenne mesurée sur quelques minutes) puis envoie les données traitées (par exemple, la température moyenne) au noeud puits / station de base. En général, n'importe quel mécanisme de traitement dans le réseau peut être implémenté dans la couche application du capteur. Puisque le noeud puits peut ne pas être proche du noeud capteur, la communication à travers le canal sans fil est souvent *multi-sauts*. Plus exactement, afin d'envoyer un paquet d'un noeud capteur (*ci*) à un noeud puits (*pj*), les nœuds capteurs intermédiaires entre (*ci*) et (*pj*) sont sollicités afin de jouer le rôle de routeurs pour acheminer le paquet tout au long du chemin depuis la source (*ci*) jusqu'au la station finale (*pj*). Ceci implique que les noeuds capteurs doivent pouvoir envoyer et recevoir des paquets de données à travers le canal sans fil. Les noeuds capteurs peuvent être détruits en raison de l'épuisement de leurs réserves d'énergie ; La topologie du réseau de capteurs sans fil ou WSN peut alors s'ajuster dynamiquement, Il s'ensuit pour le protocole de routage multi-sauts la nécessité de s'adapter au changement de topologie. Pour signaler la propagation sur le canal de

capture, un noeud capteur ou puits reçoit et traite un paquet de données du canal sans fil seulement si la puissance du signal reçu excède un certain seuil de réception prédéterminé. Le calcul de l'énergie du signal reçu est déterminé par le modèle de propagation sans fil utilisé dans le modèle du canal sans fil. L'information reçue au noeud puits à travers le canal sans fil peut être encore analysée par un serveur et/ou un opérateur humain. Selon le contexte de l'information, le noeud puits peut devoir envoyer des requêtes aux noeuds capteurs. Ceci explique pourquoi, comme il est montré dans la figure VII.3, les noeuds puits doivent pouvoir être en mesure d'envoyer et de recevoir des paquets de données à travers le canal sans fil. Comme on le montre dans la figure, le modèle d'énergie dans un noeud capteur inclut des composants de fourniture d'énergie (par exemple, batterie) et des composants consommateurs d'énergie (par exemple, l'unité centrale de traitement, l'unité de capture et le générateur du signal radio). Le modèle de fonctionnement du capteur, c'est-à-dire, la combinaison de la pile de protocole de capture, de la pile de protocole de réseau, et de la couche application a une influence sur le modèle d'énergie. Par exemple, l'énergie consommée en manipulant un paquet de données est influencée par le modèle d'unité centrale de traitement ; Elle est aussi influencée par le modèle radio. L'unité centrale de traitement, les modèles radio ainsi que l'unité de capture peuvent se trouver, selon le cas, parmi plusieurs modes opératoires. Par exemple, le modèle radio peut être dans un des modes opératoires suivants : Arrêt «Off», Sommeil «Sleep», Oisif «Idle», Réception «Receive» et Transmission «Transmit». La quantité d'énergie consommée par un composant dépend du mode dans lequel se trouve ce composant. Le mode d'opération de l'unité centrale de traitement et les modèles radio peuvent influencer le modèle de fonctionnement du capteur, et le modèle de fonctionnement du capteur peut également changer le mode d'opération de l'unité centrale de traitement et des modèles radio.

VII.5 Conception de la plateforme à base d'agent :

Dans le système proposé, chaque nœud du réseau est modélisé par un agent qui possède deux parties, une partie « caractéristiques » et une autre décrivant le comportement de ce dernier.

VII.5.1 Description d'un agent

Un agent, dans notre situation, est une instance de la classe de base qui modélise les caractéristiques et le comportement de l'agent. Il est constitué de trois parties comme on le montre dans la figure suivante :

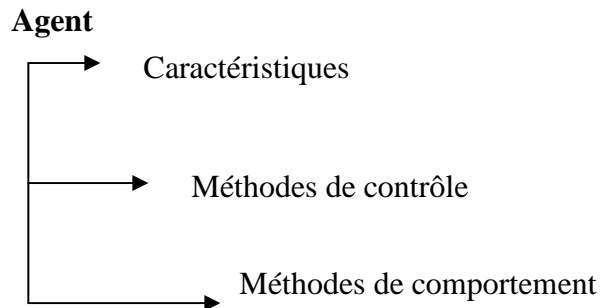


Figure VII.5 : Description d'un agent

❖ Caractéristiques d'un agent

Cette partie définit toutes les données qui décrivent l'agent, elle contient l'identifiant de l'agent, les composants qui le constituent (cpu, radio, batterie...etc.). Ces derniers sont des instances des classes qui les modélisent, par conséquent, toutes les caractéristiques de ces composants sont des caractéristiques de l'agent : le rôle de l'agent ou son type (agent objet cible, agent capteur ou agent station de base), sa position, la table des voisins, la table de routage, son mode (sleep, off, idle...etc.), son niveau d'énergie...etc.

❖ Méthodes de contrôle

Cette partie contient la collection des méthodes qui permettent de manipuler les caractéristiques de l'agent, par exemple la lecture du niveau d'énergie ou la modification de ce dernier, le changement de position du nœud, l'initialisation d'un timer, la lecture de la table des voisins...etc.

❖ Méthodes de comportement

Cette partie contient les méthodes qui décrivent les tâches à accomplir par un agent selon les événements reçus. On peut distinguer deux types d'événements :

➤ Les événements internes

Ce sont les événements qui se produisent lors du changement de l'état interne du nœud, on cite par exemple : l'épuisement de l'énergie, le changement de position, la réception d'un message à partir d'une couche de communication inférieure / supérieure du même nœud...etc.

➤ **Les événements externes**

Ce sont les événements qui sont déclenchés par un autre agent (nœud) ou par l'environnement, on cite par exemple : la réception d'un message provenant d'un autre nœud, une requête de mise à jour de la table des voisins, un changement de la table de routage...etc.

On peut illustrer quelques comportements d'un agent :

- A chaque période, l'objet cible va générer un message qui représente l'événement à détecter.

Exp : *Si timer=0 alors envoyer(mesure,date,@voisin)*

- Lorsque le niveau d'énergie atteint un certain seuil le composant va changer son mode.

Exp : *Si batterie.energie<=seuil alors radio.mode := Off*

- Lorsque un nœud est détruit, la table de routage va être changée

Exp : *Si nœud.etat=détruit alors MAJ(table_de_routage)*

VII.5.2 Description des agents de la plateforme

Dans cette section, nous décrivons les détails d'architecture et d'exécution de la plateforme de simulation de WSN, par la description détaillée de chaque type d'agent qui forment cette plateforme. On va implémenter trois types d'agents : agent objet cible, agent capteur et agent station de base. Chacun d'entre eux, possédera ses propres caractéristiques, ses propres méthodes de contrôle et son propre comportement.

VII.5.2.1 Agent objet cible

Afin de réaliser un agent objet cible figure VII.6, nous avons implémenté les unités suivantes :

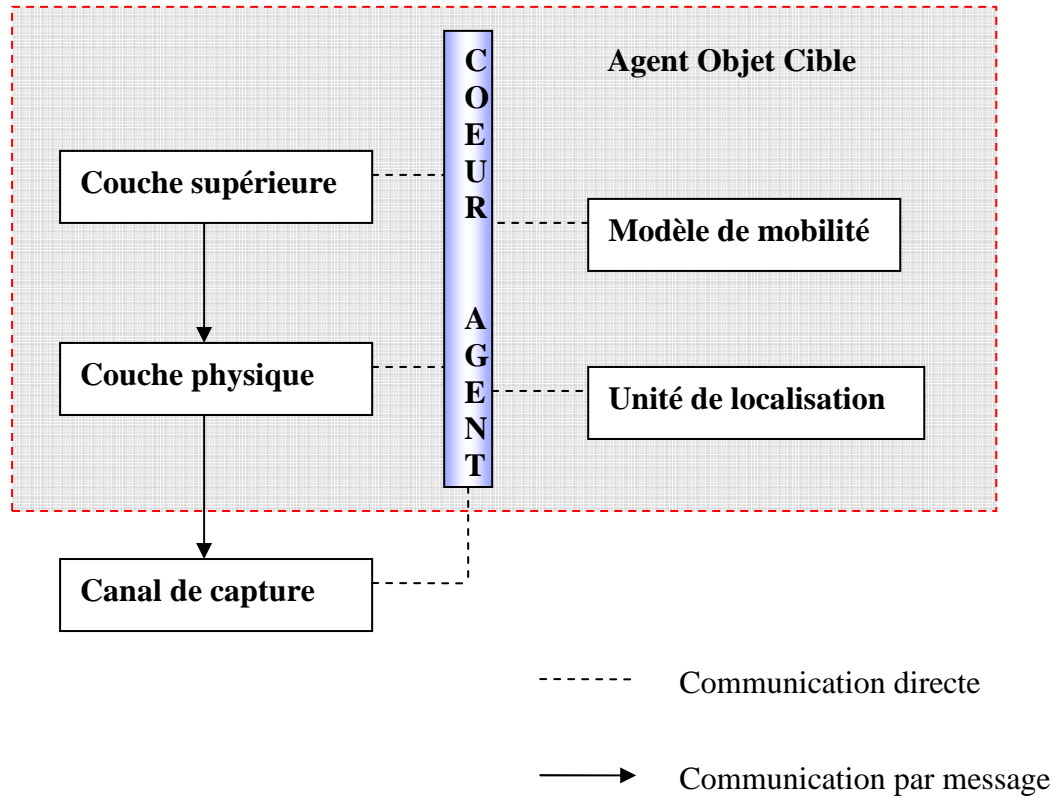


Figure VII.6 : architecture de l'agent objet cible.

➤ « *couche supérieure* »

Fournit la fonctionnalité de base d'un noeud objet cible. La « couche supérieure » produit périodiquement des stimulus (signaux) et les passe à la couche inférieure afin qu'ils soient transmis à travers le canal de capture. Cette unité a principalement deux méthodes : la méthode « *send* » employé pour passer des paquets de données vers la couche inférieure (la couche physique), et une méthode « *set_timer()* » employée pour installer des temporisateurs. Quand le temporisateur expire, une fonction de rappel « *timeout ()* » est appelée pour produire un nouveau stimulus. Ce stimulus est un paquet produit par la couche supérieure.

➤ « *modèle de mobilité* »

Il maintient le modèle de localisation, de vitesse et de mobilité de chaque objet cible et noeud capteur. Cette unité simule le mouvement des noeuds mobiles. « *modèle de mobilité* » implémente deux modèles différents de mobilité : modèle trajectoire et modèle aléatoire. Dans le modèle trajectoire, un tableau de position est donné par l'utilisateur et employé pour indiquer comment les noeuds

mobiles se déplacent. Un noeud mobile se déplace d'un point sur la trajectoire au prochain point à une vitesse constante.

➤ **« Couche physique »**

Son rôle est de recevoir un stimulus produit par l'unité « couche supérieure », interroge l'unité « modèle de mobilité » pour obtenir la localisation courante du noeud cible, et envoie le stimulus produit avec la position du noeud cible au canal de capture, dans notre cas à l'unité « canal de capture ». cette unité a deux méthodes : « *send* » et « *receive* » la première est appelée pour envoyer le paquet stimulus au canal de capture, la deuxième est appelée quand un paquet stimulus est reçu a partir de la couche supérieure,

➤ **Canal de capture**

Pour simuler le fonctionnement du canal de capture, on a implémenté les unités suivantes :

1) **« unité de localisation »**

Cette unité maintient les localisations de tous les noeuds cibles et capteurs dans le réseau de capteurs sans fil. Cette information relative à la position est rendue par l'unité « modèle de mobilité » de chaque noeud cible/capteur. Le service principal que procure l'unité de localisation est de déterminer quels noeuds capteurs sont dans la zone de capture du noeud cible et par conséquent, devraient recevoir le stimulus produit par ce noeud cible.

2) **« canal de capture »**

Simule le fonctionnement d'un canal de capture, cette unité reçoit le stimulus depuis la couche physique, interroge l'unité de localisation pour obtenir la liste des noeuds capteurs qui sont dans la zone de capture de ce noeud cible, puis envoie le stimulus produit à chaque noeud capteur qui est sur la liste.

➤ **Modèle de propagation du signal capté**

Afin de simuler le modèle de propagation du signal capté, on a réalisé l'unité « *modèle de propagation* »

C'est une unité qui simule deux alternatives pour le modèle de propagation du signal capté sur le canal de capture.

❖ **La propagation sismique :**

Le modèle de propagation sismique calcule la puissance du signal reçu comme une fonction de la distance entre l'émetteur et le récepteur. La puissance du signal reçu P_r est donnée par:

$$P_r = \frac{P_t}{\max(d, d_0)^{f_a}}$$

Où

P_t : la puissance du signal transmis.

d : la distance entre l'émetteur et le récepteur.

d_0, f_a : facteurs d'atténuation du signal.

❖ **La propagation acoustique**

Dans la propagation acoustique, la puissance du signal reçu P_r est calculée par l'équation suivante:

$$P_r = N(P \times \mu_g, \sigma_g^2) \text{ Où}$$

$$P = \frac{P_t}{\max(d, d_0)^{f_a}}, \mu_g = U(\min_g, \max_g)$$

P_t : la puissance du signal transmis

d : la distance entre l'émetteur et le récepteur

➤ **Cœur d'agent**

Le rôle de l'unité cœur d'agent est la coordination entre les différentes unités précédemment décrites.

VII.5.2.2 Agent capteur

Afin de simuler le noeud capteur, comme il est illustré dans la figure VII.7, on a réalisé les unités suivantes :

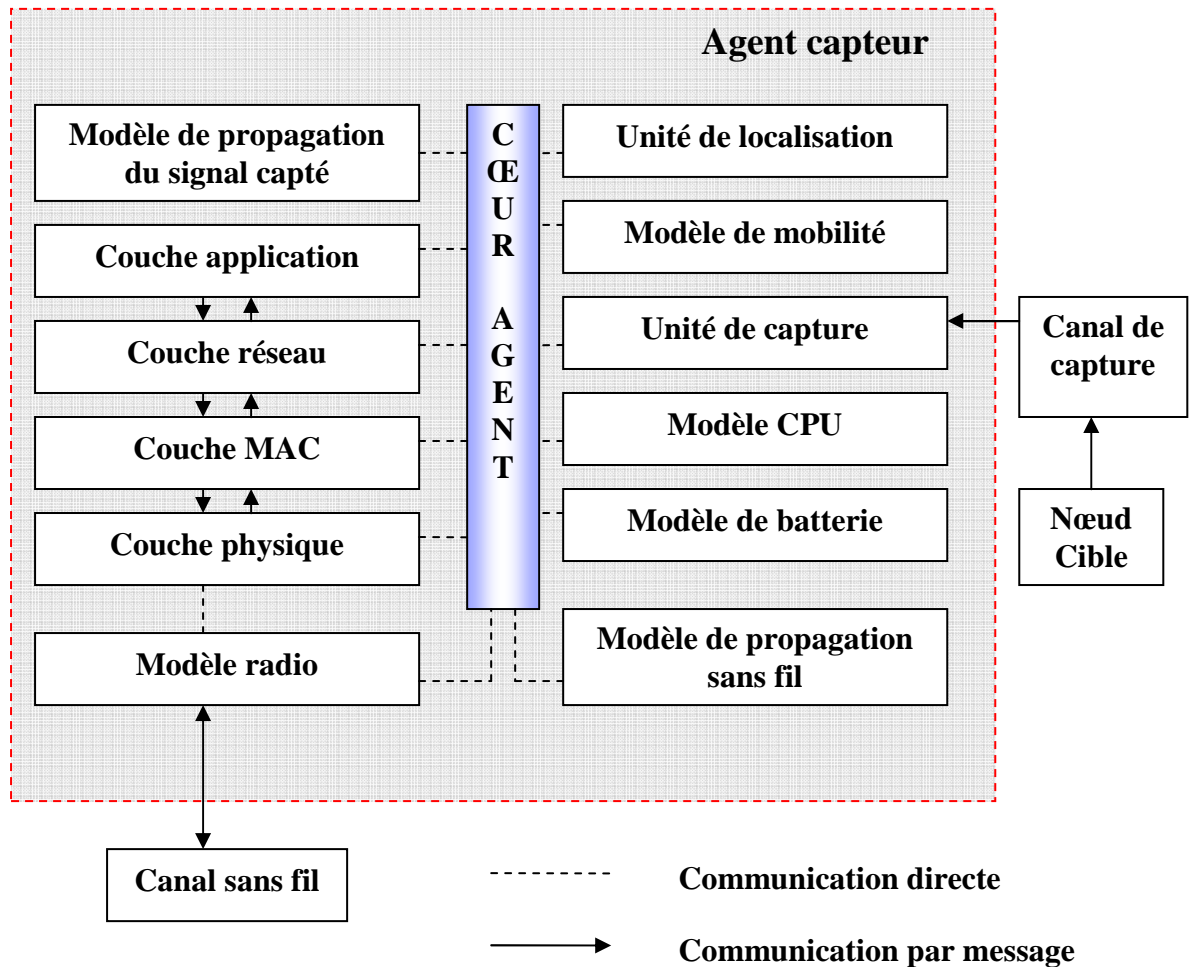


Figure VII.7: architecture de l'agent capteur

➤ **L'unité de capture**

Cette unité est destinée à simuler la fonctionnalité de capture dans un nœud. Elle est divisée en réalité en deux sous unités :

a) **La couche inférieure**

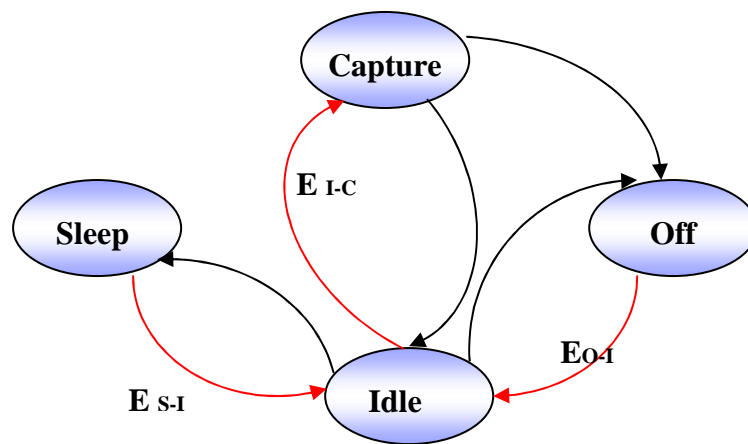
Elle simule le fonctionnement de la couche physique d'un capteur. Son rôle est de recevoir un stimulus produit par un agent objet cible par l'intermédiaire du canal de capture. En tenant compte de la position de l'objet cible à l'instant de production du stimulus, de la position courante du nœud capteur fournie par l'unité de localisation, de la puissance avec laquelle le stimulus a été produit ainsi que du modèle de propagation, la couche inférieure va calculer la puissance du signal reçu (P_r). Si la puissance du signal reçu est

inférieure à un certain seuil le stimulus est ignoré. Sinon, il est envoyé à la couche supérieure de l'unité de capture.

b) La couche supérieure

Son rôle est de recevoir le stimulus envoyé par la couche inférieure de l'unité de capture et d'en extraire les données pertinentes. Ces données sont spécifiques à la couche application dédiée à la capture. Les paramètres qui rentrent en jeu sont la durée du stimulus produit et la position de l'objet cible.

L'unité de capture a trois états qui sont : Capture, Off, Idle, Sleep. La figure montre les états de transition (ou les transitions d'états) de cette unité.



➤ Eétat1-état2 : quantité d'énergie consommée

Figure VII.8 : Etats de transition de l'unité de capture.

➤ **La couche application**

Cette couche reçoit les données spécifiques à l'application de capture à partir de l'unité de capture, elle fait aussi des tâches de traitement, en particulier la construction du paquet qui contient des données qui doivent faire l'objet d'un envoi à la station de base. Ce paquet sera passé vers les couches basses de la pile de protocole de communication sans fil (couche réseau, couche MAC, couche physique) et puis vers le canal sans fil pour arriver à la station de base.

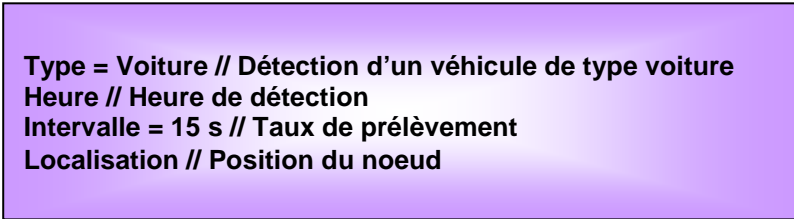
➤ **La couche réseau**

Cette couche est responsable de la mise à jour de la table de routage conformément à la politique du protocole de routage. A chaque période le nœud

capteur envoie des requêtes pour savoir qui sont ses voisins pour acheminer les données. Elle implémente le protocole de routage pour le réseau de capteurs. La couche réseau reçoit le message de la couche application puis elle transforme le message en un paquet MAC et l'envoie à la couche MAC. Le paquet au niveau de la couche réseau est destiné soit à être diffusé ou bien à être envoyé à un nœud spécifique (station de base). Pour notre plateforme on a implémenté la simulation du protocole « Directed Diffusion ».

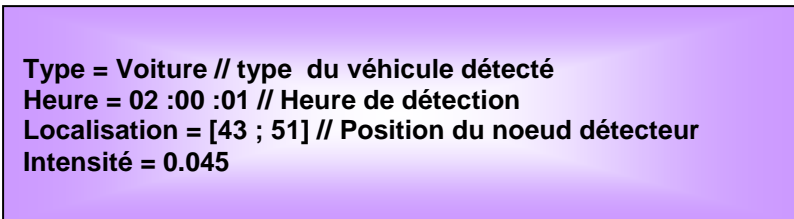
➤ Le protocole Directed Diffusion

Directed Diffusion se compose de plusieurs éléments: intérêts, données, gradients, et renforts. Un intérêt est une demande ou une interrogation qui indique ce que veut l'utilisateur. Chaque intérêt contient une description d'une tâche de capture. Typiquement, les données dans les réseaux de capteurs sont les informations rassemblées ou traitées concernant un phénomène physique. Dans ce protocole, les données sont décrites en utilisant des paires attribut-valeurs, comme il est modélisé dans la figure VII.9. Cette diffusion établit des gradients ou des chemins dans le réseau pour véhiculer les événements.



```
Type = Voiture // Détection d'un véhicule de type voiture
Heure // Heure de détection
Intervalle = 15 s // Taux de prélèvement
Localisation // Position du nœud
```

Figure VII. 9 : Exemple d'une demande



```
Type = Voiture // type du véhicule détecté
Heure = 02 :00 :01 // Heure de détection
Localisation = [43 ; 51] // Position du nœud détecteur
Intensité = 0.045
```

Figure VII.10 : Exemple d'une réponse par une source

Un gradient est une direction créée dans chaque nœud recevant un intérêt. Le gradient est dirigé du nœud récepteur vers le nœud émetteur de l'intérêt. Les événements commencent à couler vers les sources d'intérêts à travers plusieurs chemins. Le puit renforce un ou plusieurs chemins.

Directed Diffusion se déroule en trois phases principales avant la transmission de données depuis les noeuds sources vers les noeuds puits: dissémination des intérêts et établissement des gradients, propagation des données exploratoires et renforcement des chemins.

a) Propagation de l'intérêt

Le noeud puit défini, en utilisant une liste de paires attribut-valeurs, un message d'intérêts qui sera injecté dans le réseau avec un faible débit. Chaque noeud du réseau maintient à son niveau un cache qui a pour but de garder traces des intérêts reçus pour orienter les données provenant des sources vers les noeuds puits.

Le premier message d'intérêt envoyé peut être considéré comme exploration, il essaye de déterminer si il y a, en effet, des noeuds qui détectent les informations demandées. L'intérêt est rediffusé périodiquement par le noeud puit vers ses voisins. La retransmission des intérêts est nécessaire parce qu'il ne sont pas sûrement répandus dans tout le réseau a cause de la perte ou l'expiration de la durée de vie. Chaque noeud maintient dans sa mémoire une table d'intérêt. Pour chaque intérêt, les attributs correspondant sont mémorisés afin de les utiliser pour la comparaison des données reçues. Les attributs de l'intérêt mémorisé ne contiennent pas les informations sur le noeud puit mais juste du saut immédiatement précédent, c'est-à-dire le noeuds duquel on a reçu l'intérêt.

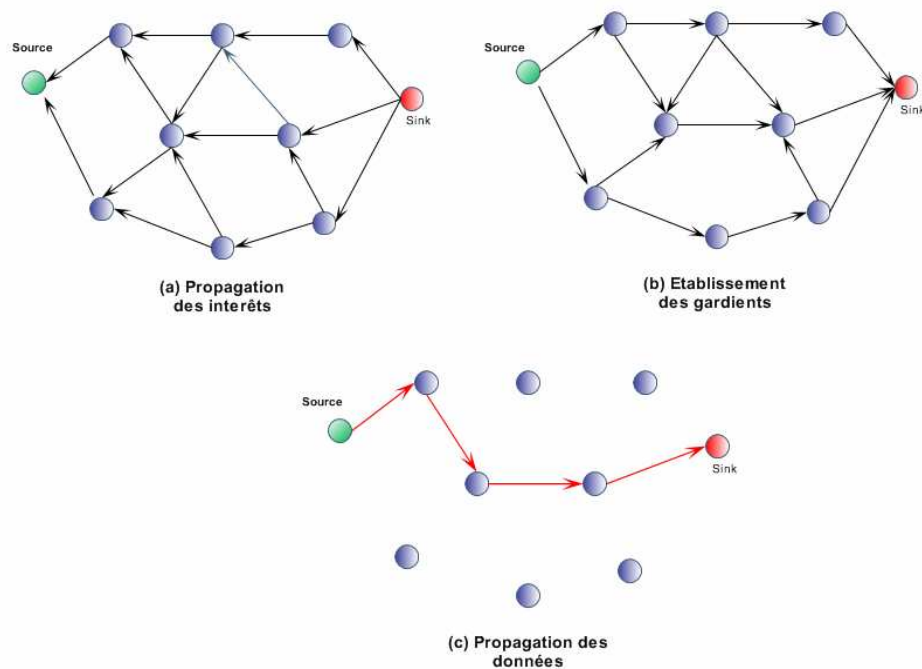


Figure VII.11 : Propagation des intérêts

Chaque gradient possède une durée de vie qui représente la durée de vie de l'intérêt associé. Quand un gradient expire, il est retiré de l'entrée d'intérêt. Quand tous les gradients d'une entrée d'intérêt expirent, l'entrée elle-même est supprimée du cache.

Quand un noeud reçoit un intérêt, il vérifie dans son cache d'intérêts s'il existe une entrée correspondante à l'intérêt reçu.

- Si oui, alors il vérifie l'existence d'un gradient pour l'émetteur de l'intérêt.
 - Si oui, alors mettre à jour le gradient.
 - Sinon, ajouter un gradient vers le voisin émetteur (pour établir plusieurs gradients pour le même intérêt, ce qui permet le changement rapide du chemin en cas de défaillance).
- Sinon, alors créer une nouvelle entrée dans le cache, et créer dans cette entrée un gradient vers le voisin à partir duquel l'intérêt a été reçu.

b) Propagation des Données

Si un noeud détecte un événement, il le compare aux différents intérêts mémorisés dans son cache. S'il trouve une correspondance, il envoie immédiatement le message contenant l'événement vers les voisins.

Le voisin, à son tour, lorsqu'il reçoit un message de l'un de ses voisins, consulte sa mémoire pour vérifier s'il ne l'a pas reçu précédemment et s'il y a une correspondance avec l'un des intérêts stockés.

- Si oui, le noeud vérifie alors son cache de données (associé à l'entrée de l'intérêt) qui garde la trace des données récemment envoyées afin d'empêcher les boucles.
 - Si le message reçu correspondant à une entrée du cache de données, alors il est abandonné car il a déjà été envoyé.
 - Sinon, le message est inséré dans le cache puis renvoyé aux voisins pour lesquels un gradient a été établi.
- Si non, il supprime le message, et ainsi de suite jusqu'à atteindre le noeud puit.

c) Renforcement des chemins

Dans le schéma que nous venons de décrire jusqu'à maintenant, le noeud puit diffuse, au début et à plusieurs reprises, un intérêt à faible débit. On appelle cet intérêt *explorateur*, puisqu'il est prévu pour l'établissement et la préparation des chemins. On appelle les gradients établis suite à ces intérêts des gradients

explorateurs. Le but étant d'obtenir les données à haut débit appelées *données renforcées* en appliquant le renforcement positif et supprimer les chemins redondants et les boucles par le renforcement négatif.

- **Renforcement positif**

Quand les données exploratoires atteignent le noeud puits, ce dernier choisit un voisin et lui envoie un message de renforcement positif avec un débit plus élevé. Un nœud intermédiaire qui reçoit ce message de renforcement, renforce son gradient vers l'émetteur (en incrémentant sa valeur par exemple) et l'envoie à l'un de ses voisins. De cette manière, l'un des chemins explorés est récursivement renforcé.

Le choix du voisin par le noeud puits ou intermédiaire dépend de la règle de renforcement appliqué. Par exemple quand un noeud choisit le premier voisin à partir duquel il a reçu une donnée exploratoire. Par conséquent, un chemin ayant la plus faible latence est établi. Ou bien, on peut choisir le voisin à partir duquel le noeud a reçu le plus grand nombre de messages de données. Ainsi, le chemin le plus fiable est renforcé.

Même en présence d'un chemin renforcé, le noeud source rediffuse périodiquement des données exploratoires dans le but de renforcer ultérieurement un meilleur chemin vers le noeud puits.

- **Renforcement négatif**

Le renforcement négatif est utilisé pour supprimer les chemins inutiles. La règle locale appliquée au niveau de chaque noeud dépend de la règle choisie pour le renforcement positif.

- ❖ *Supprimer les chemins redondants*

Quand un noeud puits renforce un nouveau chemin grâce à une nouvelle donnée exploratoire, il envoie un message de renforcement négatif au noeud voisin appartenant à l'ancien chemin. Quand ce noeud voisin reçoit ce message, il dégrade son gradient vers le puit et envoie le renforcement négatif aux voisins qui lui ont transmis des données. Par cet ordre, l'ancien chemin renforcé positivement est supprimé rapidement.

- ❖ *Suppression des boucles*

Bien que les messages tournants en boucle seront immédiatement supprimés grâce au cache de données, il est préférable de tronquer les

boucles pour l'économie de ressources avec la règle locale pour le renforcement négatif.

❖ *Réparation locale de chemins défaillants*

Les communications entre chaque paire de noeuds voisins du même chemin renforcé peuvent subir quelques dégradations dues à l'épuisement d'énergie d'un noeud, ou à des facteurs environnementaux (les obstacles par exemple). On peut appliquer les règles de renforcements pour permettre la réparation locale des chemins défaillants ou dégradés.

➤ **La couche MAC**

Le message MAC reçu de la couche réseau est envoyé au canal sans fil par l'intermédiaire de la couche physique qui utilise le modèle de radio pour transformer l'état de la radio avant d'envoyer le message au canal sans fil. Dans notre cas on a implémenté une classe qui simule le fonctionnement du protocole MAC_802_11.

➤ **Le protocole MAC 802.11**

En utilisant ce protocole, chaque paquet envoyé par la couche réseau à la couche MAC est encapsulé dans un frame MAC avec une en-tête. Le paquet envoyé par la couche réseau est muni d'un indicateur qui spécifie si le paquet doit être diffusé ou envoyé à un noeud spécifique.

Si le paquet est destiné à la diffusion, alors il est encapsulé dans un frame MAC à diffuser avec l'en-tête MAC approprié et mis dans la file d'attente de la couche MAC. Si le paquet est destiné à un noeud spécifique, alors un frame RTS est créé puis inséré dans la file d'attente de la couche MAC. Si la taille du paquet réseau est supérieure à celle du frame MAC, il sera divisé en fragments portant des en-têtes MAC et insérés dans la file d'attente qui leur est consacrée. La couche MAC attend alors que le canal soit libre pour envoyer le frame qui se trouve dans la file d'attente. La couche MAC possède une durée dans le NAV (Network Allocation Vector), qui spécifie pour combien de temps le médium sera utilisé par un autre noeud. Si la durée spécifiée dans le NAV a expiré, alors la couche MAC attend que le canal soit libre pendant le temps de DIFS (D : Distributed Coordination Function, IFS : Inter Frame Space), si le canal est resté libre après la période DIFS, elle choisit un BackOff, celui ci correspond à une période exprimée par un nombre aléatoire de ST (Slot Time). Le premier noeud qui constate que son

BackOff a expiré, a le droit de transmettre son prochain frame. Tous les noeuds intermédiaires qui reçoivent ce frame, procéderont à la mise à jour de leurs NAVTimer à la valeur obtenue à partir du champ d'en-tête du frame reçu ; La décrémentation du temporisateur BackOff des noeuds intermédiaires est alors arrêté. Une fois que le canal devient libre (c'est-à-dire lorsque le NAVTimer expire), les noeuds commencent à décrémenter leurs temporisateurs BackOff. Le premier noeud, dont le BackOff a expiré, s'accapare le canal et envoie le premier message dans la file d'attente. Si le message est à diffuser, alors tous les noeuds qui appartiennent à sa région reçoivent ce message. Si c'est un frame RTS, le noeud destinataire vérifie si son temporisateur NAV permet la réception (sa région de transmission est occupée ou pas) et lui répond alors en envoyant un frame CTS. Les temporisateur NAV de tous les noeuds intermédiaires qui reçoivent le RTS sont mis à jour par la durée de CTS+DATA+ACK, qui signifie que le canal est occupé pendant cette durée. Le noeud source qui envoie le RTS a un temporisateur RTSExpired qui est déclenché quand le RTS est envoyé. Ce temporisateur expire après la durée de RTS+CTS. Si le noeud source ne reçoit pas le CTS dans cette durée, et que le temporisateur RTSExpired a expiré alors le compteur de nouvelle tentative du frame RTS est incrémenté. Si ce compteur de tentatives est inférieure à ShortRetryLimit (selon les spécifications de l'application), alors le CW (Contention Window) est doublée et la durée aléatoire donnée par le BackOff est choisie entre un et CW. Si le noeud destinataire répond par l'envoi de CTS, les noeuds intermédiaires qui reçoivent le CTS mettront à jour leur NAVTimer obtenu à partir du champ d'en-tête du frame CTS (durée DATA + ACK). Une fois que le noeud source obtient le CTS, il envoie le fragment correspondant du paquet réseau à la destination. Quand le noeud destinataire reçoit le fragment, il envoie l'acquittement au noeud source. Le temps de pause qui sépare l'envoi d'un paquet de données et la réception de son acquittement est appelé SIFS (Short Inter Frame Space). Une fois que le noeud source obtient l'acquittement, il vérifie s'il y a d'autres fragments à envoyer à ce noeud sans nécessités d'autres frames RTS. Le tableau suivant illustre les valeurs des paramètres de MAC 802.11 de notre implémentation. Et la figure VII.12 représente le fonctionnement de ce protocole.

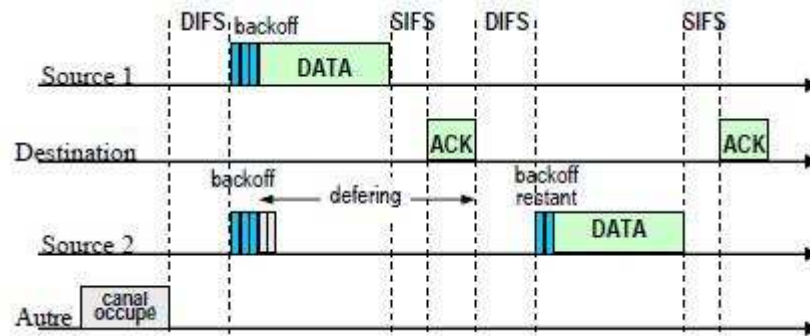


Figure VII.12 : Exemple montrant le backoff et le defering [97]

Propriété	Valeur
SIFS	10 μ sec
DIFS	28 μ sec
Slot Time	20 μ sec
Data Rate	1 Mbps
RTS Length	44 bytes
CTS Length	38 bytes
ACK Length	38 bytes

Tableau VII.1 : les Paramètres de simulation de MAC 802.11

➤ **Couche physique**

Cette couche simule des fonctionnalités de la couche physique d'une carte sans fil. Elle envoie des frames MAC à travers le canal sans fil. Différentes puissances de transmission peuvent être définies dans cette couche. Quand un frame MAC est reçue par le biais du canal sans fil, la couche physique utilise le modèle de propagation pour déterminer si ce frame MAC peut être décodé. S'il peut être décodé correctement, il va être passé à la couche MAC, puis aux couches supérieures de la pile de protocole de communication sans fil.

➤ **Modèle de propagation sans fil**

Il simule le modèle de propagation par radio à travers le canal sans fil. On a implémenté deux modèles de propagation radio :

❖ **Modèle de propagation à espace libre (Free space model):**

Il suppose une propagation idéale à condition qu'il n'y ait pas d'obstacles entre l'émetteur et le récepteur. H.T. Friis a présenté l'équation qui calcule la puissance du signal reçu qui est la suivante.

$$P_r = \frac{P_t * G_t * G_r * \lambda^2}{(4\pi)^2 * d^2 * L}$$

- P_t est la puissance du signal transmis.
- P_r est la puissance du signal reçu.
- G_t, G_r sont respectivement les gains d'antenne de la transmission et de la réception
- L est la perte et d est la longueur d'onde.

❖ **Modèle Two-ray ground reflection:**

Il est plus exact que le précédent puisque il prend en considération les réflexions. L'équation de la puissance du signal reçu est la suivante.

$$P_r = \frac{P_t * G_t * G_r * h_t^2 * h_r^2}{d^2 * L}$$

h_t et h_r sont les hauteurs des antennes d'émission et de réception respectivement.

➤ **Unité de localisation**

Cette unité maintient les localisations de tous les noeuds cibles et capteurs dans le réseau de capteurs sans fil. Cette information de localisation est rendue par l'unité « modèle de mobilité » de chaque nœud cible/capteur. Les informations de positions données par cette unité sont utilisées pour calculer la puissance du signal reçu par le canal sans fil / canal de capture.

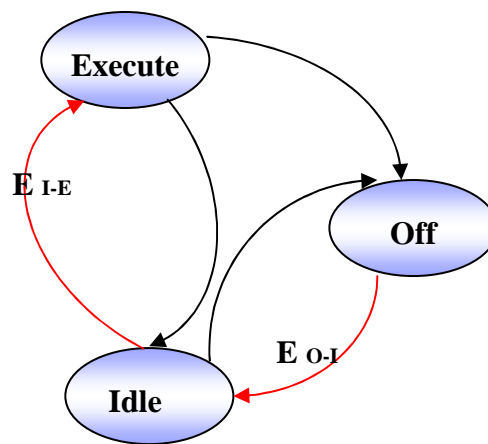
➤ **Modèle de mobilité**

Cette unité simule le mouvement des noeuds mobiles. (Comme illustré précédemment).

➤ **Modèle CPU (unité de traitement)**

Cette unité est conçue pour simuler le rôle de l'unité de traitement (CPU) qui est l'exécution de l'ensemble des instructions qui assure le fonctionnement du nœud capteur. Le code exécuté est formé de l'ensemble des instructions qui manipulent un paquet passant dans la pile de protocoles de communication sans fil et qui assure le traitement de chaque couche (couche physique, couche MAC,

couche réseau et couche application) ainsi que tous les traitements faits lorsque un stimulus est détecté, par exemple : le calcul de la puissance du signal reçu. Pour des raisons de simplification et en raison du fait que la quantité d'énergie consommée par la CPU est peut être négligeable, on va pas simuler l'exécution de chaque instruction (ADD, SHIFT, MOV...etc.), mais on va prendre en compte l'énergie consommée à cause du changement de l'état de la CPU qui peut être : oisif (Idle), inactif (Off), en exécution (Execute). Comme on le montre dans la figure VII.13.

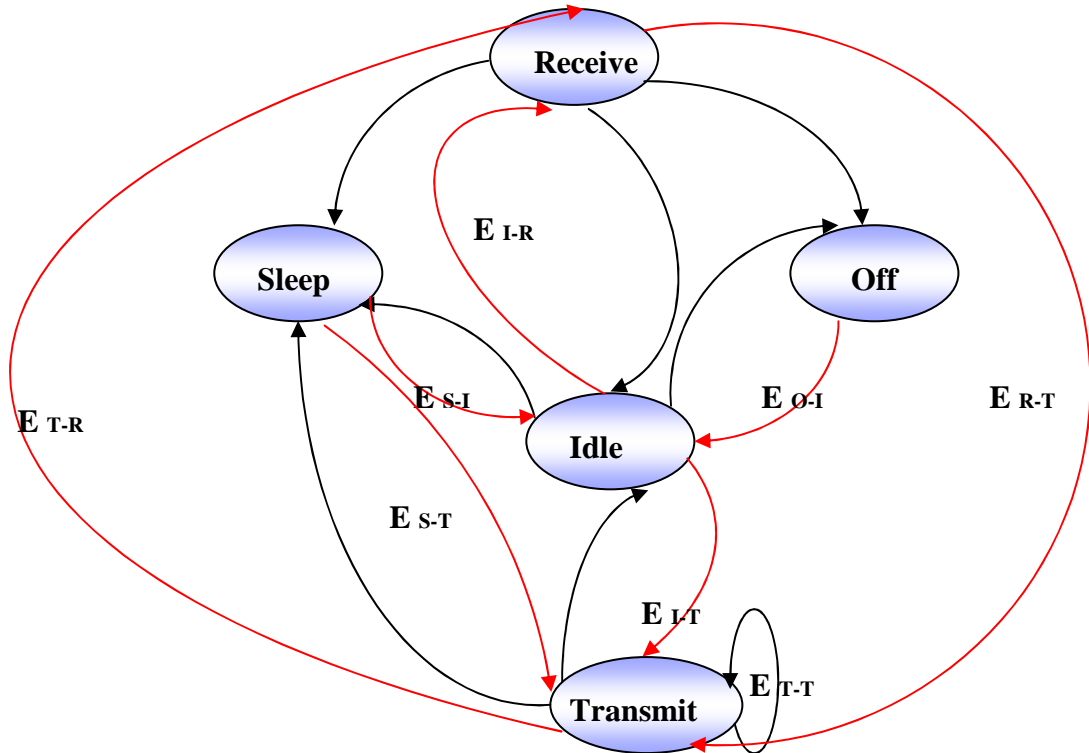


➤ **E_{état1-état2} : quantité d'énergie consommée**

Figure VII.13 : état de transition de l'unité de traitement (CPU).

➤ **Modèle radio**

Cette unité est conçue pour simuler le rôle de la radio dans un capteur, les états de ce composant sont utilisés pour calculer l'énergie consommée lorsque un message est transmis ou reçu. L'utilisateur va configurer ce composant par l'indication de la puissance du signal transmis. Les différents modes associés à la radio sont : Arrêt «Off », Sommeil « Sleep », Oisif « Idle », Réception « Receive » et Transmission « Transmit ». Comme il est illustré dans la figure VII.14.



➤ $E_{\text{état1-état2}}$: quantité d'énergie consommée

Figure VII.14 : état de transition de la radio.

➤ **Modèle de batterie**

Cette unité a pour rôle de simuler le fonctionnement d'un générateur d'énergie, c'est la ressource qui fait fonctionner tous les autres composants hardware (unité de traitement, radio, unité de capture). Cette unité a besoin d'une fonction qui estime la durée de vie de la batterie. Cette estimation est nécessaire pour que l'agent capteur puisse prendre des décisions qui minimise la consommation d'énergie afin de maximiser la durée de vie. Ces décisions consistent à changer du mode opérationnel d'un composant, par exemple mettre l'unité de capture en mode « sleep », ou décider de ne pas entrer dans l'opération de routage pendant un certain temps. La durée de vie de la batterie peut être exprimé par la formule suivante:

$$T = C/I,$$

Où :

- C est la capacité restante de la batterie donnée en Ampères/Heure.
- I est l'énergie consommée couramment par le noeud donné en Ampères.

Plusieurs modèles peuvent être utilisés pour estimer la capacité restante de la batterie. Le plus simple est de prendre le modèle linéaire défini par la formule suivante:

$$C = C_m - \int_{\Delta t} I(t) \Delta t$$

- C_{in} est la capacité initiale de la batterie.
- $I(t)$ est l'énergie consommée pendant une durée Δt

On peut dire qu'il y a une relation de dépendance entre le modèle de batterie et les modes de fonctionnement des composants hardware (unité de traitement, radio, unité de capture). En fonction de la durée de vie estimée de la batterie, les modes de fonctionnement des composants seront changés (par exemple : mettre la radio en mode sleep à chaque période). D'un autre côté le mode de fonctionnement des composants hardware influence la capacité de la batterie puisque à chaque mode correspond sa propre consommation d'énergie. La batterie est une ressource critique et l'accès à cette unité est protégé ; Par conséquent, c'est à l'agent qu'incombe les tâches de gestion des demandes de consommation d'énergie et la mise à jour de la batterie.

➤ **Cœur agent**

Cette unité est responsable de la coordination entre les différentes unités hardware (unité de traitement, radio, unité de capture), et software (pile de protocoles de communication sans fil, pile de protocole de capture). C'est elle qui a le droit d'accéder à la capacité de la batterie, et toutes les demandes d'énergie sont traitées par elle. Après satisfaction des demandes, elle procède à la mise à jour de la batterie. Elle décide des changements de mode en se basant sur la quantité d'énergie restante et de la participation ou pas à une opération de routage...etc. La figure VII.15 donne une synthèse des tâches et des données de l'agent capteur.

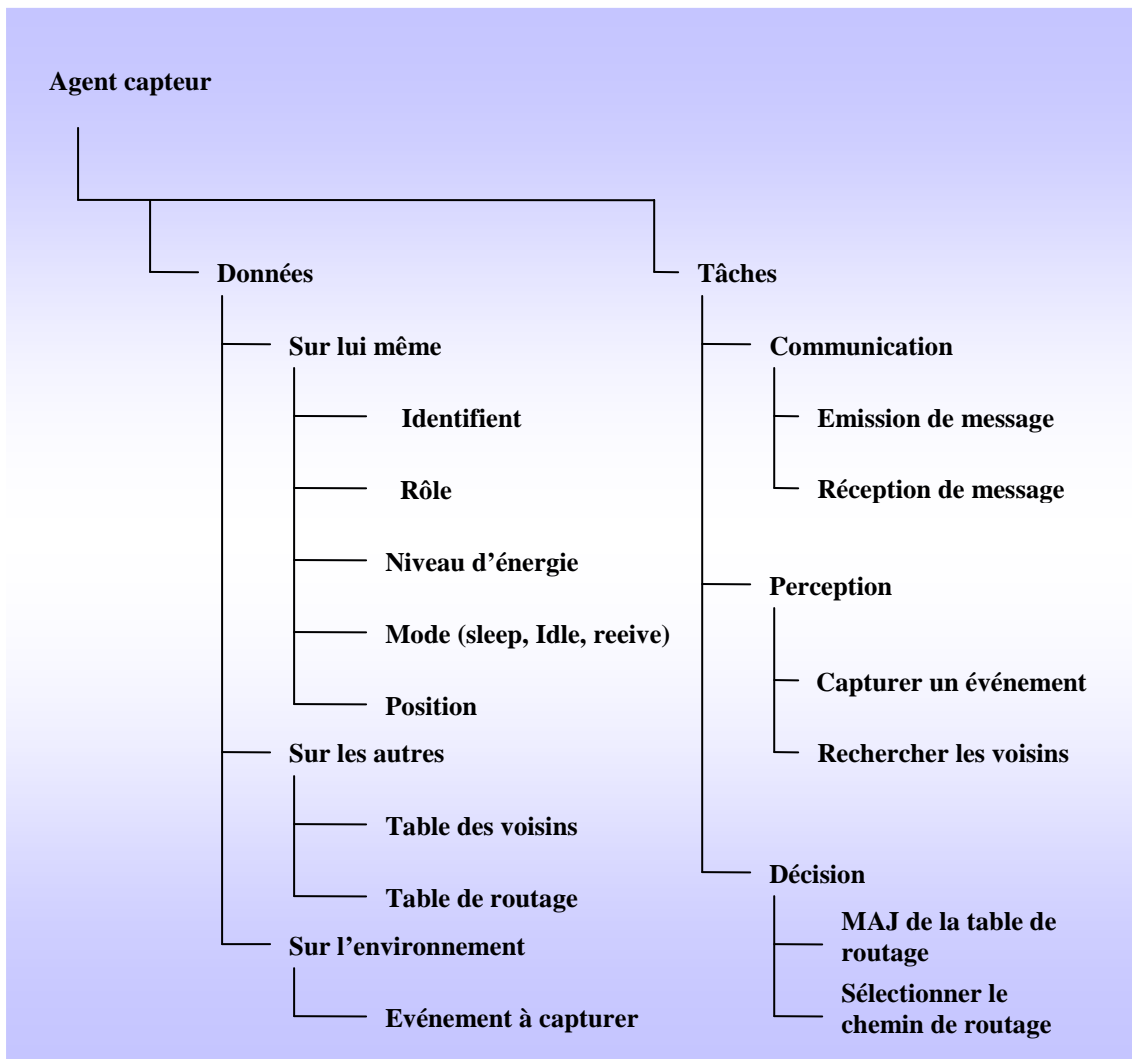


Figure VII.15 : synthèse des tâches et des données de l'agent capteur.

VII.5.2.3. Agent station de base

Pour simuler le fonctionnement d'une station de base, on a implémenté les unités qui sont illustrées dans la figure VII.16. Toutes les unités qui constituent cet agent sont décrites précédemment.

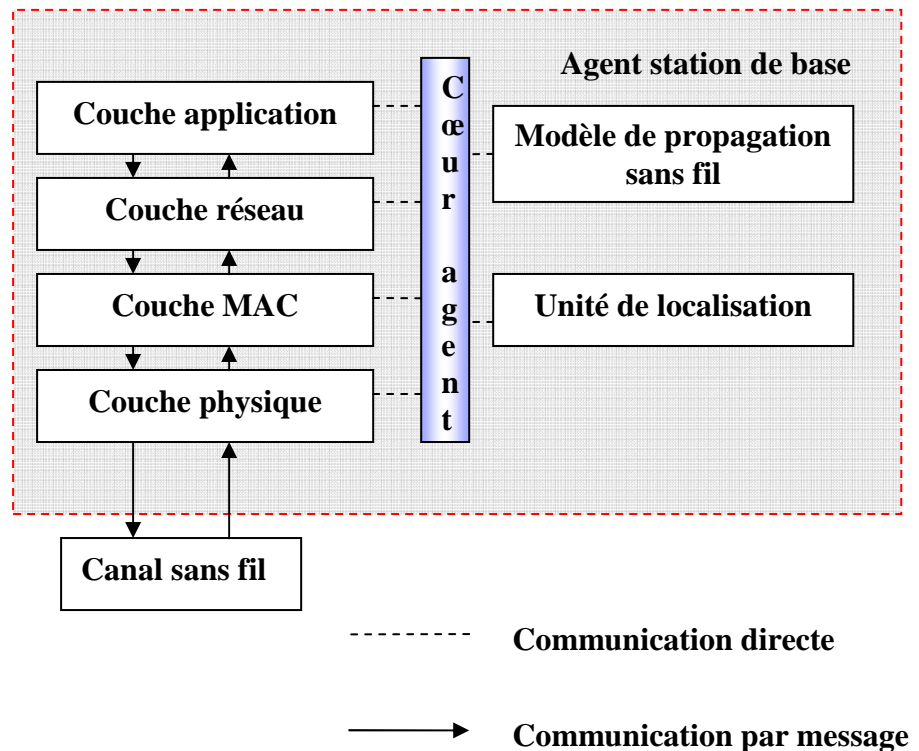


Figure VII.16 : architecture de l'agent station de base.

VII.6 Implémentation

Dans cette section on va détailler l'implémentation des agents et les classes qui les constituent destinés à la simulation des réseaux de capteurs sans fil. Pour cette objectif on a choisi la plateforme d'implémentation des systèmes multi agents « MadKit ».

VII.6.1 Environnement de développement

« MadKit » est une plateforme de développement de systèmes multi agents destinée au développement et à l'exécution de systèmes et particulièrement des systèmes multi agents fondés sur des critères organisationnels (groupes et rôles). MadKit n'impose aucune architecture particulière aux agents. Il est ainsi possible de développer aussi bien des applications avec des agents réactifs que des agents cognitifs et communicationnels, et même de faire interagir aisément tous ces types d'agents.

Cela permet ainsi aux développeurs d'implémenter l'architecture de leur choix. MadKit est écrit en Java et fonctionne en mode distribué de manière transparente à partir d'une architecture "peer to peer" sans nécessiter de serveur dédié. Il est ainsi

possible de faire communiquer des agents à distance sans avoir à se préoccuper des problèmes de communication qui sont gérés par la plateforme.

MadKit est construit autour du concept de “micro-noyau” et “d’agentification de services”. Le micro-noyau de MadKit est très petit (moins de 100Ko de code), car il ne gère que les organisations (groupes et rôles) et les communications à l’intérieur de la plateforme. De ce fait, MadKit peut être utilisé aussi bien comme outil de développement d’applications que comme un noyau d’exécution de système multi agents qui peut être embarqué dans des applications quelconques.

MadKit est un logiciel libre de type “Open Source” avec une licence mixte GPL/LGPL. LGPL pour le micro-noyau et les outils de communication, et GPL pour les outils de développement.

En général, MadKit a les qualités suivantes:

- Intégration facile de la distribution au sein d’un réseau.
- L’aspect pratique et efficace des concepts organisationnels pour créer différents types d’applications.
- Hétérogénéité des applications et des types d’agents utilisables: on peut faire tourner sur MadKit aussi bien des applications utilisant des agents réactifs simples de type fournis, que des applications disposant d’agents cognitifs sophistiqués.

MadKit considère que les agents sont des entités qui vivent au sein d’organisation. Elle repose sur le modèle AGR (Agent, Groupe, Rôle) présenté par Olivier Gutknecht et Jacques Ferber, en 1998, La particularité d’AGR est de ne rien supposer quant à l’architecture ou au mode de fonctionnement d’un agent. Voici les principales définitions d’AGR :

- Un **agent** est une entité informatique qui envoie et reçoit des messages, qui peut entrer (et sortir) de groupes et qui joue des rôles dans ces groupes. Cette entité est autonome et communicante.
- Un **groupe** est constitué de rôles et regroupe des agents qui ont soit un aspect en commun (par ex. des agents qui possèdent le même langage de communication), soit un ensemble d’agents qui travaillent ensemble, qui collaborent à une même activité (par ex. des agents qui essayent de résoudre un problème).
- Un **rôle** correspond soit au statut d’un agent, soit à une fonction particulière qu’un agent joue dans le groupe. Les rôles sont locaux aux groupes et n’ont de sens qu’à l’intérieur d’un groupe. Un rôle ne peut pas être à cheval sur deux groupes.

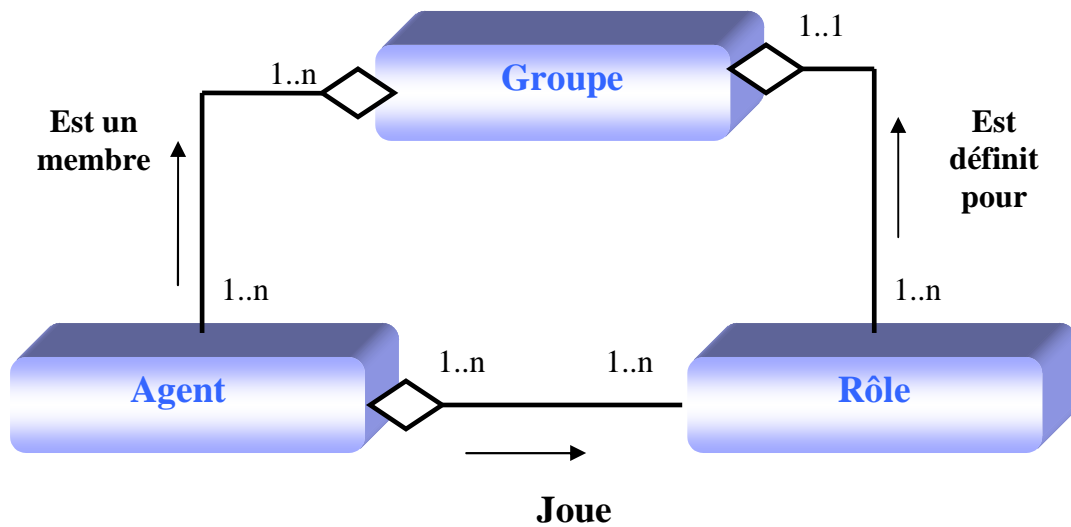


Figure VII.17 : Modèle Aalaadin

VII.6.2 Construction des agents de la plateforme de simulation

Dans cette section on va décrire la manière de construction des différents agents de notre plateforme de simulation des réseaux de capteurs sans fil.

VII.6.2.1 Les classes implémentées

Pour que les agents puissent accomplir leurs tâches, on a implémenté les classes suivantes :

- Classes pour simuler le fonctionnement des composants hardware :
 - **CpuUnit** : pour simuler le fonctionnement de l'unité de traitement.
 - **SensorUnit** : pour simuler le fonctionnement de l'unité de capture.
 - **BatteryModel** : pour simuler le fonctionnement de la batterie.
 - **RadioUnit** : pour simuler le fonctionnement de la radio.
 - **MobilityModel** : pour simuler le mouvement des noeuds mobiles.
 - **LocalisationUnit** : pour simuler le fonctionnement de l'unité de localisation.
 - **WirelessChannel** : pour simuler le fonctionnement d'un médium sans fil.
 - **SensorChannel** : pour simuler le fonctionnement d'un canal de capture.
 - **SensorPropagationModel** : pour décrire le modèle de propagation du signal capté.
 - **WirelessPropagationModel** : pour décrire le modèle de propagation sans fil.

- **StimulusGeneration** : pour simuler la génération de stimulus pour un objet cible.
- Classes pour simuler le fonctionnement de la pile de protocoles de communication sans fil :
 - **ApplicationLayer** : pour simuler les fonctions de base de la couche application.
 - **NetworkLayer** : pour simuler les fonctions de base de la couche réseau.
 - **MacLayer** : pour simuler les fonctions de base de la couche MAC.
 - **PhysicalLayer** : pour simuler les fonctions de base de la couche physique.
 - **Mac_802_11** : pour simuler le fonctionnement du protocole MAC 802.11.
 - **DirectedDiffusion** : pour simuler le fonctionnement du protocole Directed Diffusion

VII.6.2.2 Caractéristiques et méthodes de chaque classe

➤ CpuUnit

- **Caractéristiques**

id : identificateur de l'unité de traitement.

name : nom de l'unité de traitement.

execute_power_cons : énergie consommée en exécution.

idle_power_cons : énergie consommée en idle.

state : le mode de fonctionnement.

- **Méthodes de contrôle**

getState : cette méthode donne l'état courant du CPU (Idle, Execute, Off).

setState : cette méthode est utilisé par l'agent pour changer l'état de CPU.

➤ SensorUnit

- **Caractéristiques**

id : identificateur de l'unité de capture.

name : nom de l'unité de capture.

sensor_power_cons : énergie consommée en capture.

idle_power_cons : énergie consommée en idle.

sleep_power_con : énergie consommée en mode sleep.

State : état de l'unité de capture.

- **Méthodes de contrôle**

getState : cette méthode donne l'état courant de l'unité de capture.

setState : cette méthode est utilisé par l'agent pour changer l'état de l'unité.

receiveFromSensorChannel : cette méthode est déclenché lorsqu'un stimulus est reçu, elle calcule la puissance du signal reçu par l'utilisation du modèle de propagation, et décide de son acceptation ou pas.

sendToApplicationLayer : cette méthode est utilisé après l'extraction des données significative du stimulus et les envoie à la couche application.

- **BatteryModel**

- **Caractéristiques**

id : identificateur de la batterie.

name : le nom de la batterie.

Initial_energy : l'énergie initial de la batterie.

Energy : l'énergie restante.

- **Méthodes de contrôle**

setEnergy : cette méthode est utilisé par l'agent pour changer l'énergie restante après une opération.

getEnergy : cette méthode retourne la capacité restante de la batterie.

- **RadioUnit**

- **Caractéristiques**

id : identificateur de la radio.

name : nom de la radio.

tx_power_cons : l'énergie consommée en transmission.

rx_power_cons : l'énergie consommée en réception.

idle_power_cons : l'énergie consommée en idle.

sleep_power_cons : l'énergie consommée en sleep.

state : état de la radio.

- **Méthodes de contrôle**

receiveFromWirelessChannel : méthode déclenchée lorsqu'un message est reçu depuis le canal sans fil.

receiveFromPhyLayer : méthode déclenchée lorsqu'un message est reçu depuis la couche physique.

setState : cette méthode est utilisée par l'agent pour changer l'état de la radio.

sendToPhyLayer : pour envoyer un message à la couche physique.

sendToWirelessChannel : pour envoyer un message au canal sans fil.

getState : cette méthode donne l'état courant de l'unité de capture.

➤ **WirelessChannel**

- **Caractéristiques**

delay : délai de transmission.

Data_rate : débit de transmission.

- **Méthodes de contrôle**

receiveFromRadio : méthode déclenchée lorsqu'un message est reçu depuis la radio.

sendToRadio : pour envoyer un message à la radio.

➤ **SensorChannel**

- **Caractéristiques**

delay : délai de transmission.

data_rate : débit de transmission.

- **Méthodes de contrôle**

sendToSensorUnit : pour envoyer un message à l'unité de capture.

receiveFromTarget : méthode déclenchée lorsqu'un message est reçu depuis un objet cible.

➤ **ApplicationLayer**

- **Méthodes de contrôle**

receiveFromSensorUnit : méthode déclenchée lorsqu'un message est reçu depuis l'unité de capture.

receiveFromNetLayer : méthode déclenchée lorsqu'un message est reçu depuis la couche réseau.

sendToNetLayer : pour envoyer un message à la couche réseau.

➤ **NetworkLayer**

- **Méthodes de contrôle**

receiveFromMacLayer : méthode déclenchée lorsqu'un message est reçu depuis la couche MAC.

receiveFromAppLayer : méthode déclenchée lorsqu'un message est reçu depuis la couche application.

sendToMacLayer : pour envoyer un message a la couche MAC.

sendToAppLayer : pour envoyer un message a la couche application.

➤ **DirectedDiffusion**

- **Méthodes de contrôle**

Cette classe est une sous classe de la classe mère « NetworkLayer » alors elle a les mêmes méthodes de cette dernière avec une spécification du fonctionnement du protocole DirectedDiffusion.

➤ **MacLayer**

- **Méthodes de contrôle**

receiveFromNetLayer : méthode déclenchée lorsqu'un message est reçu depuis la couche réseau.

receiveFromPhyLayer : méthode déclenchée lorsqu'un message est reçu depuis la couche physique.

sendToNetLayer : pour envoyer un message a la couche réseau.

sendToPhyLayer : pour envoyer un message a la couche physique.

➤ **Mac_802_11**

- **Caractéristiques**

SIFS : Short Inter Frame Space.

DIFS : DCF Inter Frame Space.

slot_time : unité du backoff.

- **Méthodes de contrôle**

Cette classe est une sous classe de la classe mère « MacLayer » alors elle a les mêmes méthodes de cette dernière avec une spécification du fonctionnement du protocole MAC 802.11.

➤ **PhysicalLayer**

- **Méthodes de contrôle**

receiveFromMacLayer : méthode déclenchée lorsqu'un message est reçu depuis la couche MAC.

receiveFromRadio : méthode déclenchée lorsqu'un message est reçu depuis la radio.

sendToRadio : pour envoyer un message a la radio.

sendToMacLayer : pour envoyer un message a la couche MAC.

VII.6.2.3 Les méthodes principales associées à un agent dans la plateforme MadKit

- **activate** : elle crée l'agent et l'active.
- **live** : fonction qui fait vivre l'agent.
- **handleMessage** : comportement de l'agent à la réception de message.
- **end** : action de l'agent lorsque « meurt ».

VII.6.2.4. Les agents construits

On a construit trois types d'agent qui sont les suivants :

- **TargetAgent** : est l'agent qui simule le comportement d'un objet cible dans un réseau de capteurs sans fil. C'est un agent qui a un comportement simple, il importe les classes « StimulusGeneration », « MobilityModel », « LocalisationUnit », « SensorChannel ». dans ce qui suit on va illustrer le rôle de chaque méthode dans cet agent.
 - ❖ **activate** : dans cette méthode on initialise les paramètres de l'agent objet cible. Ces paramètres sont la période pendant laquelle un timer sera initialisé, l'initialisation de la table des voisins, initialisation de la position du nœud...etc.
 - ❖ **live** : dans cette méthode, à chaque fois que le timer a expiré, la génération d'un paquet comportant un stimulus est faite par l'utilisation des méthodes de la classe « StimulusGeneration ». Et l'agent va envoyer ce paquet à l'ensemble des voisins par la méthode « broadcastMessage », cette méthode est utilisée pour diffuser un message à un groupe d'agent, et enfin la réinitialisation du timer.
 - ❖ **end** : cette méthode est appelée lorsque un agent est détruit.

- **SensorAgent** : est l'agent qui simule le comportement d'un nœud capteur dans le réseau. Cet agent importe les classes suivantes : « CpuUnit », « SensorUnit », « BatteryModel », « RadioUnit », « MobilityModel », « LocalisationUnit », « WirelessChannel », « SensorChannel », « SensorPropagationModel », « WirelessPropagationModel », « ApplicationLayer », « NetworkLayer », « MacLayer », « PhysicalLayer », « Mac_802_11 », « DirectedDiffusion ». dans ce qui suit on va illustrer le rôle de chaque méthode dans cet agent.

```

public class SensorAgent extends Agent
{
    public void activate()
    {
        // Initialisation de tous les paramètres de l'agent
        .....
    }
    public void live()
    {
        .....
        while(cond)
        {
            // la condition est la présence de l'énergie et le temps //de
            // simulation n'est pas écoulé
            Message m = waitNextMessage();
            handleMessage(m);

        }
    }
    void handleMessage(Message m)
    {
        .....
    }
    public void end()
    { // liberation des resources utilisées
        .....
    }
}

```

Figure VII.18 : schéma général de l'agent capteur

- ❖ **activate** : cette méthode va initialiser tous les paramètres de cet agent, parmi lesquels on cite : la table des voisins, la table de routage, la position du nœud, l'instanciation des classes utilisées et l'initialisation des état des

composants physiques (unité de traitement, la radio, unité de capture) pour être prêt à fonctionner...etc.

- ❖ **live** : cette méthode décrit le comportement de l'agent capteur, dans cette méthode on a utilisé la méthode «waitNextMessage» qui détecte l'arrivée d'un message. Lorsque un message est arrivé alors la méthode «handleMessage » va être déclenchée. On a implémenté dans cette méthode deux alternatives possibles. Premièrement on teste le rôle de l'agent émetteur de ce message ; S'il s'agit d'un agent objet cible alors on calcule la puissance du signal reçu en utilisant le modèle de propagation du signal capté, et si ce signal est accepté alors il va être passé à l'unité de capture ensuite à la couche application par l'utilisation des méthodes des classes implémentées des unités spécifiques. A la fin de ce processus une mise à jour de la batterie est nécessaire après une opération de capture. Dans l'autre situation, si le message est envoyé par un agent capteur ou un agent station de base alors on calcule la puissance du signal reçu en utilisant le modèle de propagation sans fil ; Et si ce signal est accepté alors il va être passé par la pile de protocole de communication sans fil en utilisant les différentes méthodes des classes qui implémentent ces couches. Et en fin une mise à jour de la batterie par le calcul de l'énergie consommée par la radio, et la période du temps pour laquelle l'unité de traitement est en exécution.
- ❖ **end** : dans cette méthode on enregistre dans des fichiers toutes les données nécessaires pour l'évaluation de la performance de réseau. Ces données sont : l'énergie consommée, le nombre de paquets transmis/reçu/capturés...etc.
- **BaseStationAgent** : est l'agent qui simule le comportement d'une station de base dans le réseau. Cet agent importe les classes suivantes : « LocalisationUnit », « WirelessChannel », « WirelessPropagationModel », « ApplicationLayer », « NetworkLayer », « MacLayer », « PhysicalLayer », « Mac_802_11 », «DirectedDiffusion ». dans ce qui suit on va illustrer le rôle de chaque méthode dans cet agent.
- ❖ **activate** : cette méthode va initialiser tous les paramètres de cet agent, parmi lesquels on cite : la table des voisins, la table de routage, la position du nœud et l'instanciation des classes utilisées.

- ❖ **live** : pour décrire le comportement de l'agent station de base. Dans cette méthode on a utilisé la méthode «waitNextMessage» qui détecte l'arrivée d'un message. Lorsque un message est arrivé alors la méthode «handleMessage » va être déclenchée. Dans ce cas on est sûr que le message est envoyé par un agent capteur alors on calcule la puissance du signal reçu en utilisant le modèle de propagation sans fil ; Si ce signal est accepté alors il va être passé par la pile de protocole de communication sans fil en utilisant les différentes méthodes des classes qui implémentent ces couches.
- ❖ **end** cette méthode est appelée lorsque un agent est détruit.

VII.6.3 L'exécution de la simulation et l'affichage des statistiques

Pour déclencher l'exécution de la simulation il faut initialiser tous les paramètres des noeud à partir des champs remplis par l'utilisateur (choix du modèle de propagation du signal capté : acoustique/séismique., le modèle de propagation sans fil, la position de noeud...ect), ainsi que le choix de la période pendant laquelle la simulation doit être exécutée. Après cette phase d'initialisation, on doit appeler la méthode « activate » pour activer les agents. Quand cette méthode est appelée les agents commencent d'interagir entre eux par l'envoi de messages. Quand cette période est écoulée, des vecteurs sont créés par chaque agent capteur. Ces vecteurs contiennent des informations qui nous aident à l'analyse des performances du réseau. Ces informations peuvent être : le nombre de paquet transmis/reçu pour chaque noeud, l'énergie consommée par chaque noeud, le nombre de paquet capturés. Par exemple si on constate qu'il existe des noeuds qui ont une grande surcharge (un grand nombre de paquet reçu/transmis) donc une grande consommation d'énergie, on doit changer l'architecture ou procéder au redéploiement des noeuds du réseau pour pallier à cette surcharge, et par voie conséquence maximiser la durée de vie du réseau. Ces vecteurs seront enregistrés dans des fichiers qui vont être utilisés pour afficher des histogrammes et des courbes, pour mieux analyser les performances.

Pour notre plateforme on a choisit l'enregistrement des paramètres suivants pour permettre aux utilisateurs d'analyser les performances de leurs réseaux.

❖ **La densité ou le nombre moyen de voisins**

Ce paramètre permet d'avoir une idée sur l'influence de la densité du réseau (le nombre moyen des voisins d'un nœud) sur la consommation d'énergie de ce dernier.

❖ **Le nombre de nœud dans le réseau**

Ce paramètre nous permet de connaître l'évolution des performances d'un réseau en fonction du nombre de nœuds qui le constituent. C'est donc un très bon indicateur en ce qui concerne la propriété de scalabilité.

Pour notre plateforme l'évaluation d'un protocole ou d'une architecture d'un réseau est basée sur l'évaluation de l'énergie moyenne consommée par chaque nœud. Cette dernière nous permet d'estimer la durée de vie de réseau.

VII.7 Quelques interfaces de l'application

Dans cette partie nous allons visualiser quelques aspects relatifs à l'interface de l'application.

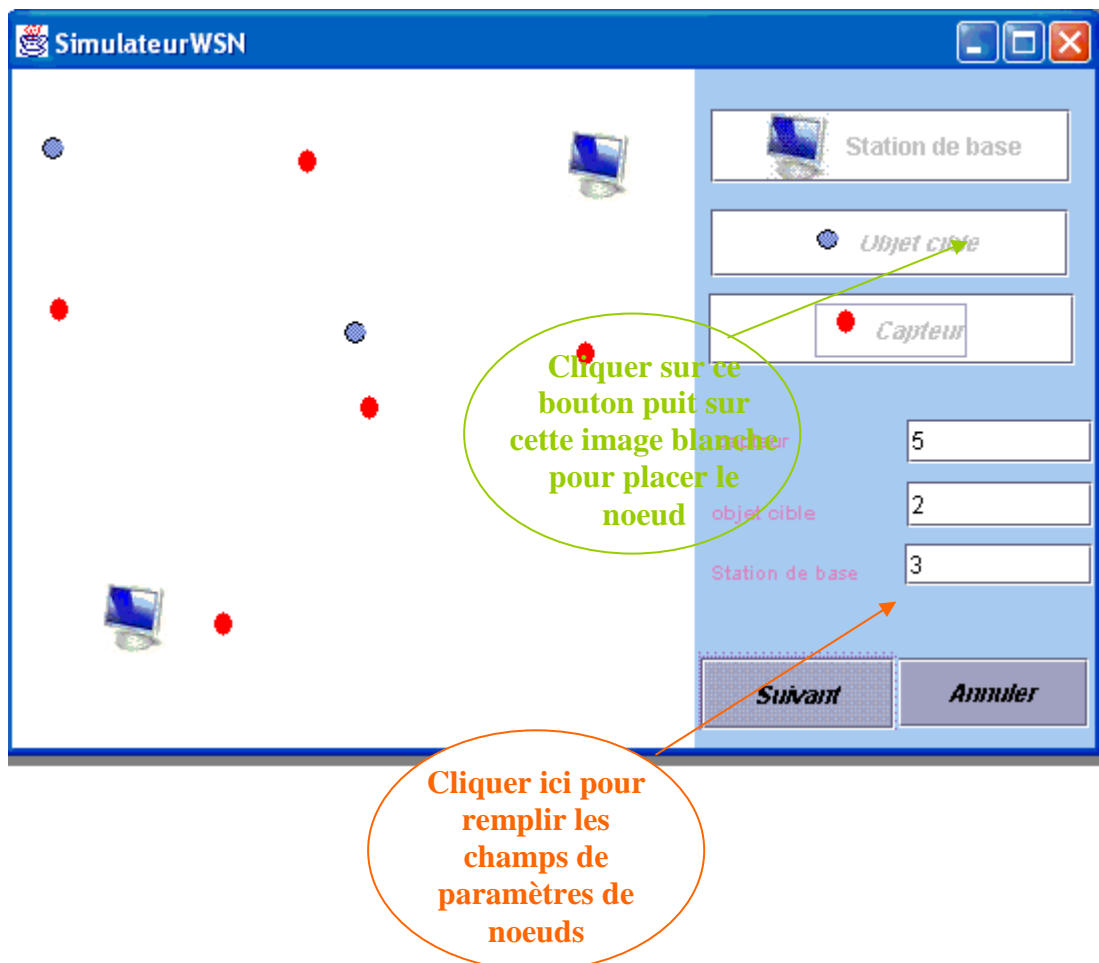


Figure VI-19 : La fenêtre principale.

Pour insérer des noeuds (station de base, noeud, objet cible) on va cliquer sur l'objet à insérer ensuite sur l'emplacement désiré sur l'image.

A la fin de la simulation, on va utiliser les différentes variables pour faire des statistiques, et pour visualiser des graphes comme il est illustré dans les figures.

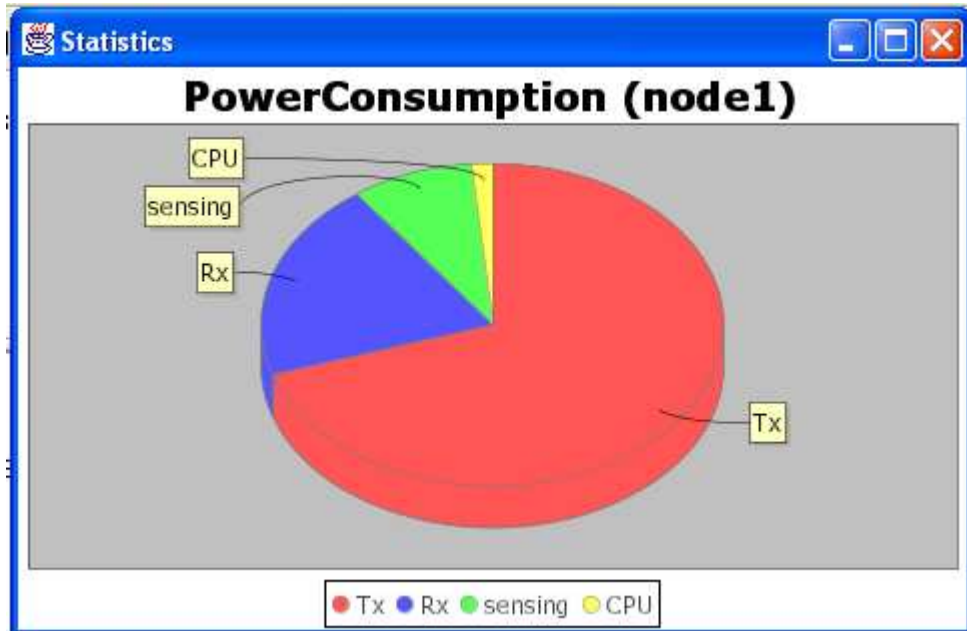


Figure VI-20 : consommation d'énergie d'un nœud après la simulation

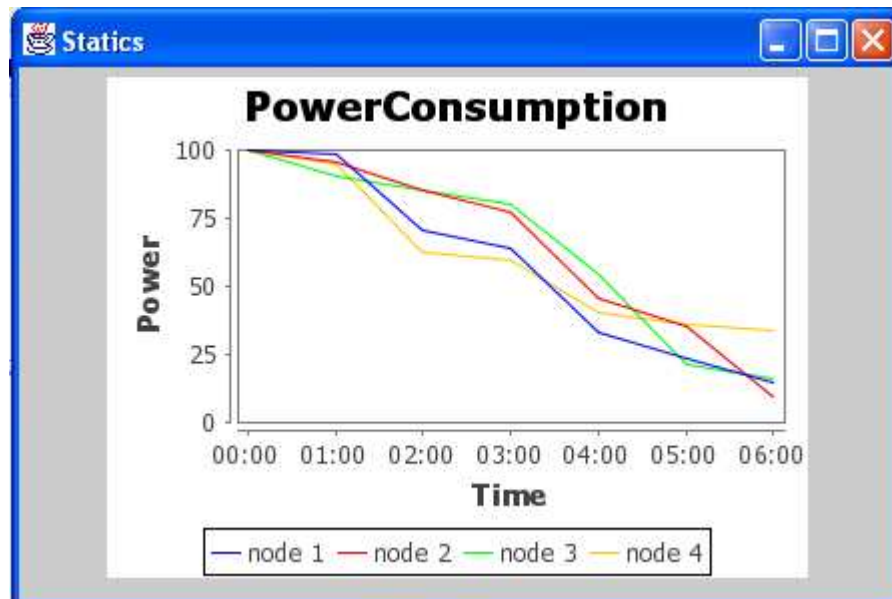


Figure VI-21 : Evolution de la réserve de batterie des quatre nœuds

VII.8 Conclusion

Dans ce chapitre, nous avons tenté de mettre en lumière la manière de concevoir et de réaliser une plateforme d'aide à la conception et à la simulation des réseaux de capteur sans fil. Cette plateforme comporte trois types d'agent : agent objet cible, agent capteur et agent station de base. Chacun de ces agents a son propre comportement suivant le rôle qu'il doit jouer dans cette plateforme, ce rôle est réalisé par la simulation du comportement de chaque composant physique du nœud et de chaque protocole de communication. Afin d'assurer une durée de vie maximale du réseau, il faut minimiser la consommation d'énergie de ses nœuds, notre plateforme donne une idée sur la consommation d'énergie pour pouvoir valoriser les protocoles et l'architecture de réseau.

Les réseaux de capteurs représentent une amélioration significative par rapport aux capteurs traditionnels. L'étendue des applications des réseaux de capteurs est vaste, on les retrouve dans le domaine de la santé, de la sécurité et dans le secteur militaire. Les réseaux de capteurs permettent à l'utilisateur une meilleure compréhension de l'environnement. De nos jours, les réseaux de capteurs sans fils font partie intégrante de notre vie.

Au terme de ce mémoire de magistère, je peux résumer les choses en disant que le travail que j'ai mené sur l'application du paradigme multi-agents aux réseaux de capteurs sans fil, m'a permis dans une première phase d'appréhender l'étendue et la complexité des concepts et des méthodes afférents à ce domaine. Je me suis familiarisée avec ces concepts et ces méthodes et pris connaissance de la palette assez large des applications en univers réel. Au cours d'une seconde phase, j'ai tenté de modéliser une plateforme basée agent dédiée à la simulation des réseaux de capteurs sans fil. La simulation est fondamentale avant le déploiement en environnement réel, en raison des avantages qu'elle offre et plus particulièrement en matière de coût et de tests qui peuvent être réalisés dans les conditions extrêmes. Fort heureusement pour l'élaboration de la plate forme, je ne suis pas partir de zéro puisqu'il existe déjà des outils de simulation qui m'ont inspiré et que j'ai avantageusement utilisé. En matière de fonctionnement des réseaux de capteurs, l'énergie est une ressource critique qui conditionne la durée de vie du réseau, c'est pour cette raison que nous lui avons consacré une attention particulière dans la plateforme que nous proposons ; D'autant plus que les plateformes existantes ne lui accorde pas l'intérêt qu'elle mérite.

Dans ce mémoire, nous avons proposé une nouvelle approche de conception de protocoles et d'applications pour les réseaux de capteurs qui est basée sur le paradigme multi-agents. Cette approche consiste à abstraire les caractéristiques physiques du réseau et des noeuds ainsi que leurs comportements en agents autonomes. La modélisation des agents s'est faite par le langage Java en utilisant la plateforme de modélisation MadKit.

Toutefois, cet effort ne peut être considéré que comme un début pour un travail plus profond qui aurait comme objectif une optimisation plus poussée pour augmenter la durée de vie des réseaux et en fin de compte leurs efficacités et leurs rentabilités.

- [1] J. Agre, L. Clare, An integrated architecture for cooperative sensing networks, *IEEE computer Magazine* (May 2000) 106–108.
- [2] M. Bhardwaj, T. Garnett, A.P. Chandrakasan, Upper bounds on the lifetime of sensor networks, *IEEE International Conference on Communications ICC'01*, Helsinki, Finland, June 2001.
- [3] N. Bulusu, D. Estrin, L. Girod, J. Heidemann, Scalable coordination for wireless sensor networks: self-configuring localization systems, *International Symposium on Communication Theory and Applications (ISCTA 2001)*, Ambleside, UK, July 2001.
- [4] A. Cerpa, D. Estrin, ASCENT: adaptive self-configuring sensor networks topologies, *UCLA Computer Science Department Technical Report UCLA/CSDTR-01-0009*, May 2001.
- [5] A. Cerpa, J. Elson, M. Hamilton, J. Zhao, Habitat monitoring: application driver for wireless communications technology, *ACM SIGCOMM'2000*, Costa Rica, April 2001.
- [6] S. Cho, A. Chandrakasan, Energy-efficient protocols for low duty cycle wireless microsensor, *Proceedings of the 33rd Annual Hawaii International Conference on System Sciences*, Maui, HI Vol. 2 (2000), p. 10.
- [7] B. Halweil, Study finds modern farming is costly, *World Watch* 14 (1) (2001) 9–10.
- [8] W.R. Heinzelman, J. Kulik, H. Balakrishnan, Adaptive protocols for information dissemination in wireless sensor networks, *Proceedings of the ACM MobiCom'99*, Seattle, Washington, 1999, pp. 174–185.
- [9] C. Intanagonwiwat, R. Govindan, D. Estrin, Directed diffusion: a scalable and robust communication paradigm for sensor networks, *Proceedings of the ACM MobiCom'00*, Boston, MA, 2000, pp. 56–67.
- [10] C. Jaikaeo, C. Srisathapornphat, C. Shen, Diagnosis of sensor networks, *IEEE international Conference on Communications ICC'01*, Helsinki, Finland, June 2001.
- [11] J.M. Kahn, R.H. Katz, K.S.J. Pister, Next century challenges: mobile networking for smart dust, *Proceedings of the ACM MobiCom'99*, Washington, USA, 1999, pp. 271–278.
- [12] With Glacier Park in Its Path, Fire Spreads to 40,000 Acres, *New York Times*, Vol. 150, Issue 51864, p. 24, 0p, 1 map, 4c, 9/2/2001.
- [13] S. Slijepcevic, M. Potkonjak, Power efficient organization of wireless sensor networks, *IEEE International Conference on Communications ICC'01*, Helsinki, Finland, June 2001.

- [14] B. Warneke, B. Liebowitz, K.S.J. Pister, Smart dust: communicating with a cubic-millimeter computer, *IEEE Computer* (January 2001) 2–9.
- [15] <http://www.fao.org/sd/EIdirect/EIre0074.htm>.
- [16] P. Bonnet, J. Gehrke, P. Seshadri, Querying the physical world, *IEEE Personal Communications* (October 2000) 10–15.
- [17] <http://www.alertsystems.org>
- [18] N. Noury, T. Herve, V. Rialle, G. Virone, E. Mercier, G. Morey, A. Moro, T. Porcheron, Monitoring behavior in home using a smart fall sensor, *IEEE-EMBS Special Topic Conference on Microtechnologies in Medicine and Biology*, October 2000, pp. 607–610.
- [19] J.M. Rabaey, M.J. Ammer, J.L. da Silva Jr., D. Patel, S. Roundy, PicoRadio supports ad hoc ultra-low power I.F. Akyildiz et al. / *Computer Networks* 38 (2002) 393–422 421 wireless networking, *IEEE Computer Magazine* (2000) 42–48.
- [20] P. Johnson et al., Remote continuous physiological monitoring in the home, *Journal of Telemed Telecare* 2 (2) (1996) 107–113.
- [21] M. Ogawa et al., Fully automated biosignal acquisition in daily routine through 1 month, *International Conference on IEEE-EMBS*, Hong Kong, 1998, pp. 1947–1950.
- [22] Y.H. Nam et al., Development of remote diagnosis system integrating digital telemetry for medicine, *International Conference IEEE-EMBS*, Hong Kong, 1998, pp. 1170–1173.
- [23] B. Sibbald, Use computerized systems to cut adverse drug events: report, *CMAJ: Canadian Medical Association Journal* 164 (13) (2001) 1878, 1/2p, 1c.
- [24] E.M. Petriu, N.D. Georganas, D.C. Petriu, D. Makrakis, V.Z. Groza, Sensor-based information appliances, *IEEE Instrumentation and Measurement Magazine* (December 2000) 31–35.
- [25] G.J. Pottie, W.J. Kaiser, Wireless integrated network sensors, *Communications of the ACM* 43 (5) (2000) 551–558.
- [26] Standard for Information technology_ Telecommunications and information exchange between systems_ Local and metropolitan area networks_ Specific requirements Part 15.1 : Wireless medium access control (MAC) and physical layer (PHY) specifications for wireless personal area networks (WPANs). New York, 2002.
- [27] J. Lansford – HomeRF : Bringing Wireless Connectivity Home – Intel HomeRF technology Tutorial (Avril 1999)

- [28] LAN-MAN Standards Committee of the IEEE Computer Society– 802.15.4 IEEE Standard for Information technology, Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (LR-WPANs) –IEEEStd 802.15.4-2003 (2003)
- [29] J.F. Llibre, P. Pinel et E. Campo – Quel choix de source d'énergie pour rendre un système communicant autonome ? – XIIIeme Colloque National de la Recherche dans les IUT, Brest, France (2006)
- [30] LAN-MAN Standards Committee of the IEEE Computer Society – 802.11 IEEE Standard for Information technology, Specific Requirements – Part 11 : Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications – IEEE standard 802-11 (1999)
- [31] IEEE Standard 802.16-2004 : IEEE Standard for Local and Metropolitan Area Networks - Part 16 : Air Interface for Fixed Broadband Wireless Access Systems. New York, 2004.
- [32] Agha, K. A., Pujolle, G., and Vivier, G. (2002). Réseaux de mobiles et réseaux sansfil. Eyrolles.
- [33] Lemlouma, T. (2000). Le routage dans les réseaux ad-hoc, Mémoire de maîtrise, Université des Sciences et de la Technologie Houari Boumèdiene.
- [34] K. Akkaya et M. younis. A survey on routing protocols for wireless sensor networks. Elsevier Ad Hoc Network, vol. 3/3:pages 325-349, 2005.
- [35] J.N. Al-Karak and A.E. Kamal. Routing techniques in wireless sensor networks: a survey. IEEE wireless communications, volume 11: pages 6-28, December 2004.
- [36] C.Intanagonviwat, R.Govindan, and D.estrin. directed diffusion : a scalable and robust communication paradigm for sensor networks. In proceedings of the 6 th annual international conference on mobile computing and networking, page 56-67, august 2000.
- [37] C.Intanagonviwat, R.Govindan, and D.estrin., J.Heidemann, and F.silva. directed diffusion for wireless sensor networking. IEEE/ACM transactions on networking IEEE/ACM transactions on networking, 11, February 2003.
- [38] Holger Karl and Andreas willig. Protocols and architectures for wireless sensor networks. Wiley,2005 . ISBN : 0470095105.
- [39] Ian F. Akyildiz, Weilian Su, Yogesh Sankarasubramaniam, and Erdal Cayirci. A Survey on Sensor Networks.Georgia Institute of Technology. Pages 102-114, IEEE Communications Magazine. August 2002.

- [40] Mohammad Ilyas and Imad Mahgoub. Handbook of sensor networks: compact wireless and wired sensing systems. CRC Press 2004.
- [41] Chonggang Wang and Kazem Sohraby. A Survey of Transport Protocols for Wireless Sensor Networks. University of Arkansas, Bo Li, The Hong Kong University of Science and Technology, Mahmoud Daneshmand, AT&T Labs Research, Yueming Hu, South China Agricultural University. June 2006.
- [42] Carlos Pomalaza-Rez. Wireless Ad Hoc & Sensor Networks. University of Oulu, Finland, 2004.
- [43] Simon, H. A. (1991). Sciences des systèmes, science de l'artificiel. Dunod.
- [44] Durand, D. (1998). La systémique. Que sais-je?, 2nd édition, PUF.
- [45] Ferber, J. (1995). Les Systèmes Multi-Agents : vers une intelligence collective. InterEditions.
- [46] Demazeau, Y. and Costa, A. R. (1996). Populations and organisations in open multi-agent systems. In 1st Symposium on Parallel and Distributed AI, Hyderabad, India.
- [47] Wooldridge, M. (1999). Intelligent agents. In Weiss, G., editor, Multiagent systems : A modern approach to Distributed Artificial Intelligence. MIT Press.
- [48] Rao, A. and Georgeff, M. (1995). Bdi agents : from theory to practice. In conference of 1st International Conference on Multi-Agent Systems ICMAS, pages 312–319. AAAI Press.
- [49] Drogoul, A. (1993). De la Simulation Multi-agents à la Résolution Collective de Problèmes, Thèse de doctorat, Université Paris VI.
- [50] Joël Quinqueton. Cours: Fondements des Systèmes Multi-Agents. LIRMM 2003. www.lirmm.fr/~jq/Cours/3cycle/smaEMA03c1.pdf
- [51] Stuart J. Russell and Peter Norvig. Artificial Intelligence. A Modern Approach. Prentice-Hall, Inc. A Simon & Schuster Company Englewood Cliffs, New Jersey 07632. 1995.
- [52] Aušra RAMONAITE « Contribution à la modélisation à base de systèmes multi-agents d'une architecture communicante pour l'accélération de flux applicatifs au travers de l'Internet »
- [53] E.ADAM « La programmation orientée agent »
- [54] Vercouter, L. (2000). Conception et mise en oeuvre de systèmes multi-agents ouverts et distribués, Thèse de doctorat, Université de Jean Monnet et Ecole des Mines de Saint-Etienne.

- [55] Russell, S. and Norvig, P. (1995). *Artificial Intelligence : a Modern Approach*. Prentice-Hall.
- [56] Wooldridge, M., Jennings, N., and Kinny, D. (2000). The gaia methodology for agent-oriented analysis and design. In *Autonomous Agents and Multi-Agent Systems*, volume 3, pages 285–312. Kluwer Academic Publishers.
- [57] Hairong Qi, Yingyue Xu, Xiaoling Wang. *Mobile-Agent-Based Collaborative Signal and Information Processing in Sensor Networks*. Department of Electrical and Computer Engineering, University of Tennessee, Knoxville, TN 37996-2100 USA. IEEE 2003.
- [58] Mohammed Ketel Numan S. Dogan Abdollah Homaifar. *Distributed Sensor Networks based on Mobile Agents Paradigm*. Dept. of Electrical & Computer Eng. North Carolina A&T State University Greensboro, NC 2741 1. IEEE 2005.
- [59] Farid Esmaili Motlagh (GS22752), *Wireless Sensor Networks: Challenges in Biomedical Applications*. Institute of Advanced Technology, University Putra Malaysia (2008)
- [60] Adrien VAN DEN BOSSCHE, Proposition d'une nouvelle méthode d'accès déterministe pour un réseau personnel sans fil à fortes contraintes temporelles, MEMOIRE de THESE DOCTORAT de L'UNIVERSITÉ de TOULOUSE II (2007)
- [61] Tayeb Lemlouma. *Le Routage dans les Réseaux Mobiles Ad Hoc*. Mini projet proposé par Dr. Nadjib Badache septembre 2000.
- [62] J. Misra. Distributed discrete-event simulation. *ACM Computing Surveys*, Vol. 18, No.1, pp. 39-65, March 1986.
- [63] NS -2. <http://www.isi.edu/nsnam/ns>
- [64] D.B. Johnson. The Rice University Monarch Project. Technical Report, Rice University, 2004.
- [65] S. Park, A. Savvides, and M. Srivastava. SensorSim: A simulation framework for Sensor networks. In *Proc. of the ACM international Workshop on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, 2000.
- [66] OPNET Technologies, Inc. OPNET Modeler
- [67] A. Sobeih and J. C. Hou. A simulation framework for sensor networks in J-Sim. Technical Report UIUCDCS-R-2003-2386, Department of Computer Science, University of Illinois at Urbana-Champaign, November 2003.

- [68] X. Zeng, R. Bagrodia, and M. Gerla. Glomosim: A library for parallel simulation of largescale wireless networks. In Workshop on Parallel and Distributed Simulation, pages 154–161, 1998.
- [69] OMNeT++, <http://www.omnetpp.org/>
- [70] A. Vargan. OMNET++ Discrete Event Simulation System, version 2.3 edition, 2003.
- [71] E. Egea-Lopez *et al.*, “Simulation Tools for Wireless Sensor Networks,” *Proc. Int’l. Symp. Perf. Eval. of Comp. And Telecommun. Sys.*, Philadelphia, PA, July 2005, pp. 559–66.
- [72] S.S Kulkarni, « TDMA services for Sensor Networks » Proceedings of 24th International Conference on Distributed Computing Systems Workshops, Pages 604-609 , 23-24 mars 2004
- [73] W. Ye, J. Heidemann, D.Estrin, “Medium Access Control With Coordinated Adaptive Sleeping for Wireless Sensor Networks” , IEEE/ACM Transactions on Networking, Volume: 12, Issue: 3,pages: 493-506,June 2004.
- [74] A. El-Hoiydi, “Spatial TDMA and CSMA with preamble sampling for low power ad hoc wireless sensor networks”, proceedings of ISCC 2002, Seventh International Symposium on Computers and Communications, pages:685-692, 1-4 July 2002.
- [75] C.C, Enz, El-Hoiydi, J-D. Decotignie, V. Peiris, “WiseNET: An Ultralow-power Wireless Sensor Network Solution”, IEEE Computer, Volume: 37, Issue: 8, August 2004.
- [76] G.Lu, B.Krishnamachari, C.S. Raghvendra, “An adaptive energy efficient and low-latency MAC for data gathering in wireless sensor networks”, proceedings of 18th International Parallel and Distributed Processing Symposium, Pages: 224,26-30 April 2004.
- [77] K. Sohrabi, J. Pottie, “ Protocols for self organisation of a wireless sensor network”, IEEE Personal communications, Volume 7, Issue 5, pp 16-27, 2000
- [78] C. Intanagonwiwat, R. Govindan, and D. Estrin, “ Direct diffusion: a scalable and robust communication paradigm for sensor networks” Proceedings of ACM MobiCom, Boston, 2000, pp 56-67
- [79] D. Braginsky and D. Estrin, “Rumor Routing Algorithm for Sensor Networks”, in the Proceedings oh the first Workshop on Sensor Networks and Applications , Atlanta, October 2003

- [80] C. Schurgers and M.B. Srivastava “ Energy efficient routing in wireless sensor networks” , in the MILCOM Proceedings on Communications for Network Centric Operations: Creating the Information Force, Mc Lean, VA, 2001
- [81] W. Heinzelman, J. Kulik, and H. Balakrishnan, « Adaptative Protocols for Information Dissemination in Wireless Sensor Networks » , Proc. 5Th ACM/IEEE Mobicom Conference Seattle, August 1999, pp. 174-185.
- [82] J. Kulik ,W. Heinzelman, and H. Balakrishnan « Negotiation based protocols for disseminating information in Wireless Sensor Networks », Wireless Networks, Volume8, pp 169-185, 2002
- [83] R.C. Shah and J. Rabaey « Energy Aware Routing for Low Energy Ad Hoc Sensor Networks », IEEE Wireless communications and Networking Conference, March 17-21, 2002, Orlando
- [84] F. Ye, A. Chen, S. Liu, L. Zhang, « A scalable solution to minimum cost forwarding in large sensor networks », Proceedings of the tenth International Conference on Computer Communications and Networks, pp 304-309, 2001
- [85] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, « Energy-Efficient Communication Protocol for Wireless Microsensor Networks » Proceedings of the 33th Hawaii International Conference on System Sciences, January 2002
- [86] S. Lindsey, C. Raghavendra, « PEGASIS : Power Efficient Gathering in Sensor Information Systems », IEEE Aerospace Conference Proceedings, 2002, Volume 3, pp 9-16
- [87] A. Savvides, C-C Han, M. Srivastava, « Dynamic fine grained localization in Ad-Hoc Networks of sensors », Proceedings of the seventh ACM Annual International Conference on Mobile Computing and Networking (MobiCom) , July 2001, pp 166-179
- [88] A. Manjeshwar and D. P. Agarwal, « TEEN : a routing protocol for enhanced efficiency in wireless sensor networks », In 1st Intenational Workshop on Parallel and Distributed Computing Issues in Wireless Networks and Mobile Computing, April 2001
- [89] A. Manjeshwar and D. P. Agarwal, « APTEEN : a hybrid protocol for efficient routing and comprehensive information retrieval in wireless sensor networks », Parallel and Distributed Processing Symposium, Proceedings International, 2002, pp 195-202
- [90] V. Rodoplu, and T. H. Meng, « Minimum Energy Mobile Wireless Network », IEEE Journal Selected Areas in Communications, vol, 17, no 8, Aug 1999

- [91] L. Li, and J. Y. Halpern, « Minimum Energy Mobile Wireless Networks Revisited », IEEE International Conference on communications, 2001, Vol 1, pp 278-283
- [92] Y. Xu, J. Heidemann, D. Estrin, « Geography-informed Energy Conservation for ad-hoc Routing » In proceedings of the Seventh Annual ACM/IEEE International Conference on Mobile Computing and Networking, 2001, pp 70-84
- [93] Y. Yu, D. Estrin, R. Govindan, « Geographical and Energy-aware Routing : A recursive Data Dissemination Protocol for Wireless Sensor Networks », UCLA computer Science Departement Technical Report, May 2001
- [94] K. Sohrabi, J. Pottie , « Protocols for self-organization of a wireless sensor network », IEEE Personal Communications, Volume 7, Issue 5, pp 16-27, 2000
- [95] Edgar H. Callaway. *Wireless Sensor Networks: Architectures and Protocols*. CRC Press 2004.
- [96] Jamal N. Al-Karaki, Raza Ul-Mustafa, Ahmed E. Kamal. *Data Aggregation in Wireless Sensor Networks - Exact and Approximate Algorithms*. Proceedings of IEEE Workshop on High Performance Switching and Routing (HPSR) 2004,,Phoenix, Arizona, USA. April 18-21, 2004.
- [97] Dominique Dhoutaut. *Etude du standard IEEE 802.11 dans le cadre des réseaux Ad hoc: de la simulation à l'expérimentation*. Thèse pour obtenir le grade de Doctorat en Informatique. 2003.