Table des matières

Contributions

1 Introduction générale

2 Les réseaux de capteurs sans fil et Mobilité

3 La collecte de données dans un RCSF avec élément mobile

4 Quel élément mobile à Déplacer dans un RCSF

5 Quel modèle de mobilité pour le déplacement du Sink

6 Comment peut-on optimiser les Déplacements du Sink dans un RCSF mobile

2 Les réseaux de capteurs sans fil et Mobilité

3 La collecte de données dans un RCSF avec élément mobile

4 Quel élément mobile à déplacer dans un RCSF

5 Quel modèle de mobilité pour le déplacement du Sink

6 Comment peut-on optimiser les déplacements du Sink dans un RCSF Mobile

Glossaire

ACK	Acknowledgment packet (Acquittement)
AODV	Ad-hoc On-demand Distance Vector
B-MAC	Berkeley MAC
C	Capteur
CDMs	Collecteurs de Données Mobiles
CHs	Cluster-Heads
CM	Cluster Member
CSMA/CA	Carrier Sense Multiple Access with Collision Avoidance
EIOSM	Environnement Intégré d'Optimisation et de Simulation basé sur les Méta-heuristiques
EM	Elément Mobile
GPRS	Global Packet Radio Service
GPS	Global Positioning System)
ID	Identifiant
LEACH	Low Energy Adaptive Clustering Hierarchy
MAC	Medium Access Control
MP	Mobile Peers
OSI	Open System Interconnection
PDA	Personal Digital Assistant
QoS	Quality of Service
RAM	Random Access Memory
RWMM	Random Walk Mobility Model
RWpMM	RandomWaypoint Mobility Model
RCSF	Réseau de Capteurs Sans Fil
RCSF-EM	Réseau de Capteurs Sans Fil à Elément Mobile
RCSF-S	Réseau de Capteurs Sans Fil Stationnaire
RM	Relais Mobile
RN	Relocatable Nodes
RS	Recuit Simulé
RT	Recherche Tabou
SM	Sink Mobile
S-MAC	Sensor MAC
SSN	Special Support Node
TDMA	Time Division Multiple Access
T-MAC	Timeout MAC
WSN	Wireless Sensor Networks
X-MAC	Excessive-MAC

Chapitre 1

Introduction générale

Sommaire

1.1 Contexte de recherche et Problématique

Au cours de ces dernières décennies, une révolution scientifique considérable en technologies de pointe a vu le jour, notamment dans les domaines de l'informatique et de l'électronique. Ceci a renforcé l'émergence d'objets miniaturisés dotés d'une certaine intelligence, de processeurs et de moyens de communication mobiles, leur permettant non seulement de capturer les changements qui surviennent dans leur environnement mais également de traiter les informations et de les transmettre à travers des réseaux de communication. Cette évolution s'inscrit dans le cadre de ***l'informatique ubiquitaire (Ubiquitous computing*** en anglais***)*** et de ***l'internet des objets (Internet of things*** en anglais***)***, qui connait un essor considérable et qui s'impose progressivement dans notre vie quotidienne. Elle permet aussi de combler le fossé entre le monde réel et le monde virtuel en rendant les objets « Intelligents ». Les récentes avancées technologiques dans le domaine des communications sans fil ainsi que le couplage des trois technologies : la nanotechnologie, la micro-électro-mécanique « ***MEMS*** en anglais Micro-Electromechanical System »et la technologie sans fil ont favorisé le développement d'entités électroniques minuscules appelées « ***nœuds capteurs » (Senor nodes*** en anglais***)*** [1,2,3].Ces petites entités autonomes mesurent les conditions ambiantes (luminosité, température, humidité, vibration, pression barométrique, son, ondes sismiques, etc.) et les transforment en signaux électriques permettant aux équipements informatiques spécifiques de les traiter. Il existe une large variété de nœuds capteurs ayant une architecture matérielle typique. Celle-ci est généralement composée de :

- Une ou plusieurs unités de capture, chargées de collecter les informations sur l'environnement proche,
- Une unité de traitement composée d'un microcontrôleur et d'une mémoire de travail,
- Une source d'énergie (batteries),
- Un module de transmission sans fil (radio).

De par leur petite taille, les nœuds capteurs se caractérisent par un faible coût de fabrication, une faible consommation d'énergie et surtout une grande flexibilité. Individuellement, ils ne peuvent pas rivaliser avec les ordinateurs ou terminaux de poche actuels car ils sont dotés de ressources très limitées en termes de capacité de calcul, d'espace de stockage de données à cause de leurs batteries limitées, non rechargeables et non remplaçables dans la plupart des scénarios de déploiement[1,3].

Ces nœuds capteurs sont capables de détecter, de collecter des grandeurs physiques relatives à l'événement surveillé, parfois de les traiter puis de les transmettre à d'autres nœuds. De ce fait, avoir une connaissance précise et complète de l'environnement à surveiller exige le déploiement d'un ensemble de nœuds capteurs formant ainsi un réseau de capteurs sans fil (RCSF) ou *Wireless Sensor Network* (WSN) en anglais [4]

Un RCSF est un nouveau type de réseau Ad-Hoc[1] sans infrastructure fixe, il est constitué d'un grand nombre de nœuds capteurs déployés densément, de manière déterministe ou aléatoire, dans une région d'intérêt et peuvent fonctionner sans intervention humaine avec une

[1] Un réseau Ad-hoc est un réseau sans fil capable de s'organiser sans la présence d'infrastructure réseau.

grande tolérance aux défaillances [5]. Ces réseaux peuvent être déployés de façon rapide dans des zones sensibles et/ou difficilement accessibles.

Les RCSFs possèdent des atouts manifestes : leur mission est le plus souvent de surveiller et de couvrir de très grands espaces géographiques, de prendre régulièrement des mesures et de faire remonter des alarmes vers certains nœuds du réseau, appelés nœuds collecteurs, capables de relayer l'information à grande échelle [6]. Toutes ces tâches doivent être réalisées en consommant le moins d'énergie possible afin de prolonger la durée de vie du réseau.

Le facteur commun, et qui constitue une préoccupation principale et un critère de performance prédominant dans pratiquement la majorité des travaux sur les réseaux de capteurs, est celui de la réduction de la consommation énergétique, ou du moins sa rationalisation. Plusieurs travaux de recherche gravitent autour d'un objectif commun : l'identification et la caractérisation des activités les plus consommatrices en énergie et l'optimisation de la consommation énergétique des nœuds-capteurs. Il est couramment admis que l'émetteur radio est un des composants les plus gourmands en énergie [7,8, 9]. Par conséquent la plupart de l'énergie dissipée par un nœud-capteur concerne la transmission et la réception des données.

Les RCSF sont connu ces dernières années une utilisation très diversifiée dans différents domaines [1,4,7]: monitoring environnemental, santé, surveillance des frontières, industrie, applications militaires, sécurité, les maisons intelligentes, transport, désastres naturels …etc. Dans la plupart des études de recherche, menées sur ce type de réseaux, les nœuds capteurs sont considérés comme stationnaires et où les données collectées par les nœuds capteurs sont acheminées vers une station de base appelée nœud puits ou *Sink*[2] via une communication multi-sauts sans fil, connue sous le nom de « *Convergecast* » c'est à dire une communication de plusieurs-à-un (*many-to-one*).

Cette approche nécessite la mise en place d'un protocole de routage des données afin de gérer les multiples transmissions pour atteindre le destinataire du paquet de données. Cette gestion n'est pas aisée car la topologie d'un RCSF est dynamique : un nœud capteur peut ne plus être joignable s'il a épuisé toutes ses réserves énergétiques ou parce que les conditions environnementales ont fortement détérioré ses capacités de communication.

Les RCSFs sont conçus pour être déployés pour des périodes allant de quelques mois à plusieurs années et donc les protocoles de routage conçus pour ces réseaux doivent également être à économie d'énergie. De plus, les conditions de déploiement peuvent rendre l'accès aux capteurs difficiles. Il sera donc impossible de remplacer ou reconfigurer les nœuds une fois que ces derniers seront déployés [1].Cette configuration paraissait efficace pour la majorité des applications pendant longtemps ; mais récemment, elle s'est avérée inefficace [3,10,11].En effet, de nouvelles applications émergentes comme le suivi d'objets mobiles, l'exploration d'espaces sensibles et/ou difficilement accessibles, exigent la prise en considération de la mobilité de certains nœuds capteurs dans l'espace de déploiement. Cette mobilité concerne soit un nœud capteur, soit un actionneur dans le cas des réseaux de capteurs/actionneurs, soit le nœud *Sink.*

[2] Le Sink est un nœud puits (en anglais)

Un nœud *Sink* peut avoir plusieurs fonctions au sein d'un réseau de capteurs. Il peut être considéré comme un équipement collecteur et disposer d'une puissance de calcul plus élevée ainsi que d'une grande capacité de mémoire pour traiter directement les données reçues.

Dans ce cas, un utilisateur pourra donc récupérer manuellement les données à un endroit unique. Le nœud *Sink* peut également disposer d'une interface de communication supplémentaire permettant de le relier aux réseaux standards (réseau privé ou Internet). Il offre ainsi la possibilité aux utilisateurs d'accéder à distance aux données stockées à son niveau ou de transmettre directement les données des nœuds capteurs à une base de stockage distante [1]. Le but principal de l'utilisation d'un nœud *Sink* mobile est d'assurer et d'améliorer la collecte de données dans un RCSF.

La collecte de données est une tâche fondamentale dans les RCSFs. Elle a pour but de récupérer les données fournies par les nœuds capteurs selon un protocole de routage afin de les analyser et de les traiter pour les transmettre ensuite à un site distant. Le protocole de routage doit fournir des techniques rapides et fiables pour la dissémination des données.

La plupart des solutions de routage pour les réseaux de capteurs utilisent des nœuds *Sinks* statiques pour la collecte des données à partir des nœuds du réseau. Cette approche se traduit par une charge de trafic élevée aux alentours du *Sink*. Les nœuds situés près du *Sink* seront plus sollicités que ceux plus éloignés du réseau. Par conséquent, ces nœuds vont consommer plus d'énergie et risquent d'épuiser leurs batteries plus rapidement dans un réseau à grande échelle, ce qui peut ainsi causer une perte de connectivité et de couverture du réseau, voire même sa défaillance. Afin d'alléger la charge de trafic élevée et le goulot d'étranglement résultant dans le voisinage du *Sink*, causés par cette vision de collecte statique ; des techniques de déploiement de collecteurs de données mobiles(*CDMs*) seront introduites dans cette thèse [12].

Ceci est la première problématique qui se pose dans les RCSFs stationnaires. Pour remédier à ces problèmes, nous avons choisi, pour effectuer notre travail, « les réseaux de capteurs sans fil mobiles » et plus particulièrement, nous nous intéressons à la mobilité du nœud collecteur en considérant tous les autres nœuds du réseau comme statiques.

Cependant la mobilité soulève plusieurs défis à relever pour une conception et une mise en œuvre efficace d'un processus de collecte. Nous énumérons, dans ce qui suit, quelques défis spécifiques à l'application de *Sink* mobile dans les RCSFs [11] et nous mettons plus l'accent sur ceux traités dans cette thèse dans la section suivante.

- ***Contact avec le Sink***: dans le cas d'une mobilité du nœud *Sink*, la communication entre les capteurs stationnaires et le *Sink* mobile nécessite la présence de ce dernier dans leur portée radio. Cependant, la détection du *Sink* mobile dépend de sa vitesse de déplacement et des cycles d'activité (sommeil/réveil) des nœuds du réseau [13]. De plus, une vitesse de déplacement rapide du *Sink* conduit à de courtes liaisons entre ce dernier et son voisinage immédiat occasionnant potentiellement des pertes considérables.
- ***Localisation du Sink*** : le but d'un RCSF est de remonter les informations collectées par les nœuds vers le *Sink*, cet objectif ne peut alors être atteint sans une localisation préalable de ce point focal. Dans un contexte à *Sink* mobile, les nœuds n'ont à priori aucune connaissance de la position courante du *Sink* contrairement à un RCSF à puits statique.

1.2 Objectifs de la thèse

L'objectif de ce travail de thèse consiste principalement à faire avancer significativement l'état de l'art sur la mobilité des RCSFs, qui est un sujet relativement récent et pour lequel il existe peu de travaux concrets. Certains n'ont été réalisés que ces dernières années [1,2,11,12,13]. Notre travail consiste essentiellement à se focaliser sur l'étude de l'impact de la mobilité du *Sink* dans un RCSF en essayant de proposer des solutions aux problématiques soulevées dans cette thèse et qui sont présentées dans la figure 1.1.

- Quel est l'élément mobile d'un RCSF qu'il faut considérer pour améliorer la qualité de collecte et la durée de vie du réseau?
- Quel est le modèle de mobilité à choisir pour réaliser la mobilité parmi les modèles proposés dans la littérature pour atteindre des performances souhaitées ?
- Comment peut-on optimiser le déplacement du *Sink* pour réussir une collecte de données efficace et diminuer significativement la latence qui est un problème majeur à résoudre ?

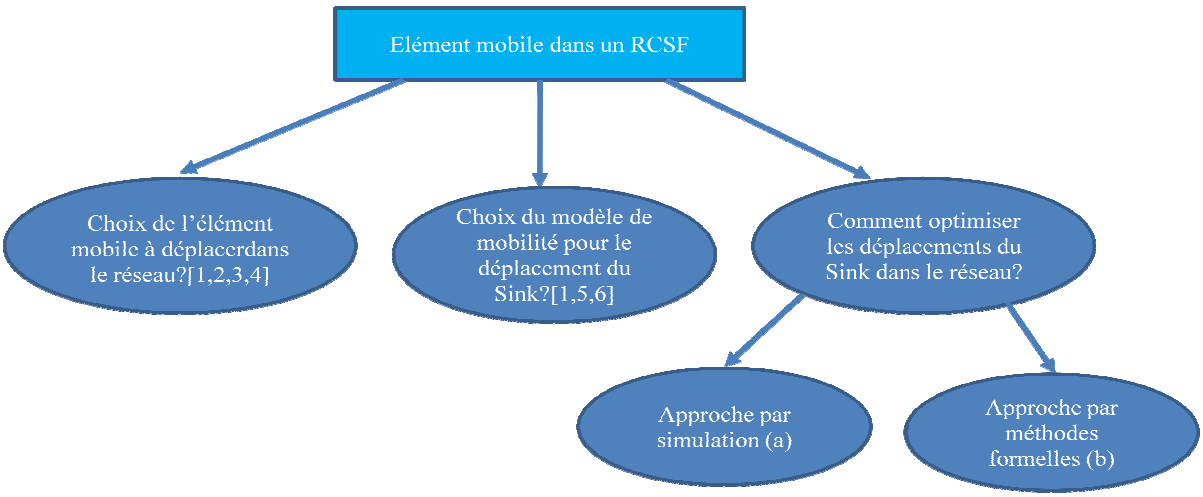


Figure1.1 – Problématiques étudiées

1.3 Contributions de la thèse

Comme nous l'avons signalé plus-haut, le domaine des RCSFs mobiles est une technologie relativement récente. Lorsque nous parlons de RCSF mobiles, nous supposons que ce sont soit les nœuds capteurs qui sont mobiles, soit le nœud *Sink* ou bien les deux en même temps, ce qui représente une problématique beaucoup plus complexe. Dans ce travail, nous nous intéressons précisément à la mobilité du nœud *Sink* c'est-à-dire la mobilité du collecteur de données.

Nos contributions dans cette thèse visent à proposer des éléments de réponse aux questions ci-dessus et à mener dans un premier temps des études par simulation pour leur validation. Elles peuvent être résumées comme suit :

- Le choix du meilleur élément mobile à déplacer dans le réseau afin de collecter les données de manière fiable et efficace tout en économisant de l'énergie et en respectant

la contrainte de délai. Cette contribution a été présentée et défendue dans les conférences [3,14]. Notre travail consiste à simuler le processus de collecte avec deux variantes afin de déterminer l'approche la plus rentable : les collecteurs de données mobiles (*CDMs) de type Sink*, ou bien les *CDMs de type nœud relais* et ceci en menant des simulations d'un certain nombre de scénarios représentant les deux approches de collecte de données. Ensuite, nous comparons les résultats obtenus dans chacune d'elles afin de retenir celle qui garantit le plus la performance du réseau. Ce travail a été réalisé en utilisant le simulateur OMNET++ couplé à deux frameworks de mobilité « MiXiM » et « INET» en utilisant « MiXiM-INET-Buddle» (version de Mars 2013).

- Le choix du meilleur modèle de mobilité parmi les trois modèles étudiés dans cette thèse (*RandomWayPoint, Gauss Markov* et *Random Walk*), selon lequel nous devons déplacer notre élément mobile. Il s'agit surtout de voir son impact sur le processus de collecte des données. Plusieurs scénarios ont été testés. Dans chacun, nous avons fait varier le type du modèle de mobilité, la vitesse de déplacement du *Sink*, le temps de simulation ainsi que le nombre de capteurs. Après la comparaison des différents scénarios réalisés avec le simulateur OMNET, et la plateforme INET, les résultats obtenus constituent notre deuxième contribution. Celle ci a fait l'objet d'une publication internationale [15].
- Un aspect important de cette thèse, novateur par rapport aux contributions déjà effectuées dans le domaine, consiste à trouver un moyen efficace pour gérer et optimiser les déplacements du *Sink* dans un RCSF en se basant sur les deux précédentes problématiques. Nous avons réalisé cette contribution en se basant sur deux approches : *Approche par simulation* et *Approche par méthodes formelles*. Dans la première approche, nous avons implémenté notre propre modèle de mobilité, ensuite nous avons simulé un certain nombre de scénarios afin d'évaluer ses performances et de le comparer avec d'autres modèles de mobilité en utilisant le simulateur CASTALIA. Ce modèle permet au *Sink* d'emprunter une trajectoire ou un parcours optimal en utilisant les paramètres adéquats de déplacement (vitesse et temps de pause) et le principe d'organiser le réseau en Clusters dans la perspective de limiter la distance entre le *Sink* et les capteurs pour effectuer une bonne collecte de données. A la fin, nous avons comparé les résultats obtenus et nous avons déduit la solution qui offre les meilleurs résultats.

  La seconde approche consiste à utiliser les méthodes approchées (incomplètes). Parmi ces méthodes, nous avons choisi les méta-heuristiques à solution unique « ***Recherche tabou*** *et* ***Recuit simulé*** » qui sont plus appropriées à notre problématique. Notre objectif est de trouver l'optimum qui représente dans notre étude la plus courte trajectoire de déplacement du *Sink* en parcourant tous les nœuds capteurs du réseau. Pour cela, nous avons implémenté notre propre outil d'optimisation et de simulation EIOSM (*Environnement Intégré d'Optimisation et de Simulation basé sur les Méta-heuristiques*) que nous avons étudiés en détail. Quelques métriques de performances ont été considérées dans cette approche telles que la consommation d'énergie, le taux d'erreur, la distance parcourue, la latence….

Ces deux premières contributions assemblées, ont fait l'objet d'une publication internationale [15], et nous ont permis d'économiser l'énergie des nœuds capteurs, de prolonger la durée de vie du réseau, de minimiser la latence, d'empêcher une perte considérable de données et surtout de résoudre le problème du goulot d'étranglement caractérisant les RSCFs statiques.

1.4 Structure de la thèse

La suite du manuscrit est organisée comme suit : deux parties principales. La première partie est consacrée à un état de l'art sur les travaux de recherche concernant notre travail et se compose de deux chapitres.

La deuxième partie est composée de trois chapitres représentant nos trois contributions, comme nous pouvons le voir sur la figure 1.2.

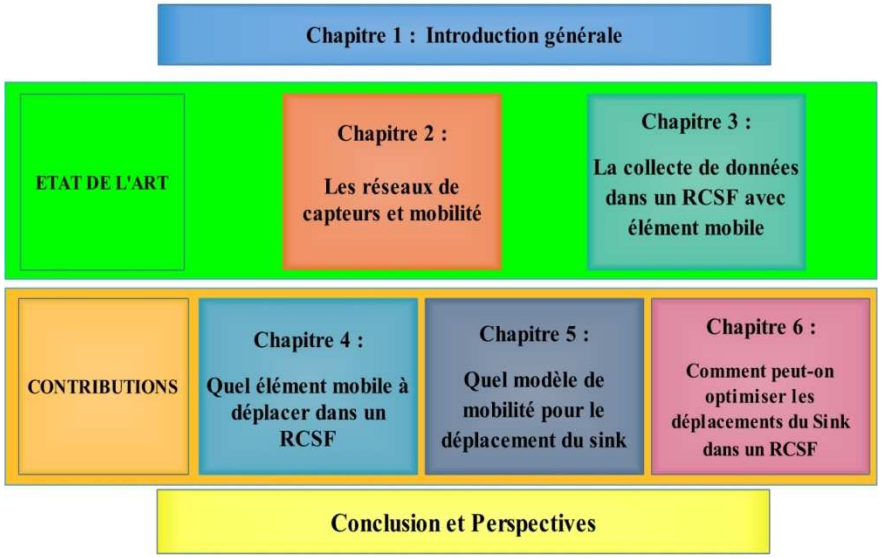


Figure 1.2 – Structure de la thèse

Dans le Chapitre 2, nous allons faire un survol des RCSFs stationnaires en donnant quelques définitions de base (leurs contraintes, les composants et les différents types de nœuds capteurs, quelques domaines d'applications, protocoles Mac ainsi que les protocoles de routage leurs correspondant etc). Ensuite nous consacrons le reste du chapitre aux RCSFs à élément mobile (*RCSFs-EM*) que nous avons utilisés dans notre travail. Nous aborderons la mobilité dans les RCSFs en général en discutant ses avantages, ses raisons, les types de mobilité : aléatoire, déterministe et contrôlée ainsi que les défis à relever pour sa mise en œuvre. Nous mettons ensuite l'accent sur la mobilité du collecteur de données pour réaliser la collecte de données et qui sera détaillée dans le chapitre suivant). A la fin, une classification des objectifs de la mobilité sera donnée.

Le Chapitre 3, se focalise sur l'une des tâches les plus importantes dans les RCSFs, à savoir la collecte de données. Nous allons faire un tour d'horizon des différents travaux utilisant le processus de collecte de données ainsi que les types de données à collecter, ensuite une classification des types de collecte sera donnée.

Nous étudierons, par la suite et dans le détail, les architectures de collecte de données avec trois types d'éléments mobiles (nœuds délocalisables, collecteurs de données :CDMs et noeuds pairs) en se concentrant plus sur la collecte de données avec CDMs de type *Sink* mobile. Les différentes phases et approches de collecte seront abordées vers la fin du chapitre pour une meilleure compréhension du processus.
La Deuxième partie regroupe toutes les contributions apportées dans ce travail, ce qui constitue le cœur de cette thèse.

Le Chapitre 4, traite la problématique du choix du meilleur élément mobile à déplacer dans le réseau. Pour cela, nous commencerons par nous intéresser aux différents travaux concernant cette problématique. Ensuite nous proposerons les scénarios de collecte de données réalisés avec deux types de CDMs (*Sink* et nœud relais) sous l'environnement de simulation MIXIM. Nous analyserons et interpréterons les résultats de simulation obtenus en déduisant l'approche la plus rentable entre les deux, en termes d'économie d'énergie, de latence et de maximisation de la durée de vie du réseau.

Le Chapitre 5, aborde la deuxième problématique, qui est le choix du meilleur modèle de mobilité parmi les trois modèles étudiés (*RandomWayPoint, Gauss Markov* et *RandomWalk*) selon lequel nous devons déplacer cet élément mobile. Nous présenterons un état de l'art concernant les modèles de mobilité pour le déplacement du *Sink* dans un RCSF. Pour répondre à cette question, nous avons simulé un certain nombre de scénarios dans lesquels nous avons varié le modèle de mobilité afin de voir son impact sur la collecte des données dans le but d'améliorer les performances du réseau et répondre à nos exigences. Ces simulations ont été faites avec le simulateur INET et les résultats obtenus seront discutés et comparés afin de désigner le meilleur modèle de mobilité qui convient à notre travail.

Le Chapitre 6, illustre la troisième problématique, celle de trouver la façon la plus optimale pour assurer une collecte de données fiable et efficace. Ceci bien sûr en optimisant la trajectoire de déplacement du *Sink* dans le réseau et en se basant sur deux approches : La simulation et les méthodes formelles (méta-heuristiques). Dans la première approche, nous avons implémenté notre propre modèle de mobilité en utilisant le principe de Clustering (protocole LEACH modifié) sous l'environnement de simulation Castalia dans le but d'optimiser les déplacements du *Sink*, Nous avons également développé notre propre outil de visualisation « CastaliaViz» afin de pouvoir visualiser notre réseau. Dans la deuxième approche, nous avons utilisé les méta-heuristiques. Parmi ces méthodes, nous avons choisi la *Recherche Tabou* et le *Recuit Simulé* afin de trouver l'optimum et confirmer que la trajectoire trouvée par simulation dans la première approche est bien la trajectoire optimale. Pour cela, nous avons implémenté notre propre outil d'optimisation et de simulation EIOSM et ceci dans le but d'optimiser les déplacements du *Sink* mobile dans le RCSF, ensuite nous avons simulé ces résultats obtenus par optimisation avec le simulateur Castalia.

Le Chapitre 7 clôture cette thèse par une conclusion générale dans laquelle nous présenterons les contributions apportées, et suggérons quelques perspectives de travaux de recherche futurs en guise de suite aux résultats présentés dans cette thèse.

Chapitre 2

Les réseaux de capteurs sans fil et Mobilité

Sommaire

2.1 Introduction

A l'heure actuelle, la mobilité des réseaux de capteurs n'est pas largement traitée. La majorité des travaux de recherche existants considèrent les réseaux de capteurs statiques où les paquets, provenant des nœuds, suivent des chemins multi-sauts vers un point névralgique appelé (*puits ou Sink*).Cette configuration s'avère l'unique solution qui permet de remonter les données capturées par les nœuds capteurs. Ainsi, certains chemins peuvent être chargés (plus sollicités que d'autres), et les nœuds proches de la station de base relayent plus de paquets [16] et sont sujets à l'épuisement prématuré de leurs batteries. Ceci peut provoquer une dégradation des performances du réseau [17].
De ce fait, de nouvelles applications émergentes telles que le suivi d'objets mobiles et l'exploration d'espace inaccessible et dangereux…, imposent la prise en considération de la mobilité dans un RCSF [11].

Dans notre travail, nous nous focalisons sur la mobilité du *Sink* car c'est un nœud qui demeure l'unique entité du réseau immunisé contre les contraintes de ressources (capacités en mémoire, traitement, bande passante, énergie, etc.). Cette mobilité consiste à déplacer le *Sink* périodiquement dans une zone d'intérêt pour la collecte de données.
Le *Sink* peut être embarqué à une entité mobile telle qu'un être humain, un animal, un véhicule, un drone ou un robot [18].
L'objectif des RCSFs mobiles est de collecter plus d'informations sur un environnement en utilisant moins de nœuds capteurs, et de permettre au réseau de s'auto-organiser. En outre, il devient capable de déplacer ses capteurs dynamiquement en cas de changements environnementaux, ce qui le rend adaptatif à l'évolution de son environnement [19].
Dans ce chapitre, nous allons commencer par présenter les RCSFs stationnaires à travers quelques définitions de base (les RCSFs et leurs contraintes, les composants et les différents types de nœuds capteurs, quelques domaines d'applications, etc). Ensuite nous aborderons la classe des RCSFs à élément mobile (*RCSFs-EM*) que nous avons utilisés dans notre travail en décrivant le concept de mobilité du *Sink* à travers ses avantages et les défis à relever pour la mise en œuvre de la mobilité. Nous présenterons ensuite les types de mobilité : aléatoire, déterministe et contrôlée. Nous terminerons ce chapitre en montrant comment la mobilité peut être exploitée pour résoudre le problème de collecte de données, et plus précisément en utilisant des nœuds collecteurs appelés collecteurs de données mobiles (*CDMs*) qui seront détaillés dans le chapitre suivant).

2.2 Réseau de Capteurs sans fil (RCSF)

2.2.1 Définition

Un RCSF est constitué de milliers de nœuds appelés nœuds capteurs sans fil ou *Motes*. Ces noeuds sont des dispositifs équipés d'un processeur, d'une interface radio, d'un convertisseur analogique-numérique (*ADC* en anglais *Analog to Digital Converter*), de capteurs, de mémoire et d'une alimentation. Le processeur assure la gestion des fonctions des nœuds et effectue le traitement des données. Les capteurs attachés aux nœuds sont capables de mesurer des grandeurs physiques telles que la température, l'humidité, la lumière, etc. En raison de la

bande passante et des contraintes énergétiques, les nœuds soutiennent principalement les faibles unités de données avec une puissance de calcul et un taux de capture limités. La mémoire est utilisée pour stocker des programmes (instructions exécutées par le processeur) et des données (des mesures brutes traitées par les capteurs). Les nœuds capteurs sont équipés d'un faible taux (10-100 kbps) et d'une petite portée radio sans fil (moins de 100 m), par exemple, la radio IEEE 802.15.4 [4]pour communiquer entre eux. Depuis la communication radio consomme plus de puissance. Ainsi la radio doit contenir des techniques de communication à économie d'énergie. La source d'alimentation qui est utilisée est la pile rechargeable [20].

Leur principale fonction est de capter des informations collectées dans différents environnements et de les transmettre. Ces nœuds peuvent avoir des positions fixes ou bien peuvent être déployés aléatoirement dans des environnements hostiles ou difficiles d'accès.

Un réseau de capteurs fonctionne de la manière suivante : les nœuds sont déployés dans une zone appelée zone d'intérêt pour pouvoir la surveiller. Ces nœuds sont souvent déployés par voie aérienne à l'aide d'avions ou hélicoptères). Ce déploiement aléatoire des capteurs nécessite que le protocole, utilisé pour les réseaux de capteurs, possède des algorithmes d'auto organisation. Afin de résister aux déploiements, ces capteurs doivent être très solides et de plus, ils doivent aussi pouvoir survivre dans les conditions les plus extrêmes dictées par leur environnement d'utilisation (feu ou eau par exemple).

Lorsqu'un nœud détecte un évènement, il a la possibilité de le traiter localement ou de l'acheminer vers des nœuds spéciaux appelés nœuds-puits « *Sinks*» via une communication multi-sauts. Ces nœuds-*Sink* possèdent plus de ressources matérielles que les nœuds capteurs. Ces données sont ensuite envoyées vers un centre de traitement des données via Internet ou par satellite pour l'analyse des informations remontées par les différents capteurs et la prise de décision ultérieure [19]. Une passerelle est utilisée pour adapter le type des données au canal. Ce processus est illustré dans la figure 2.1.

Un RCSF est constitué essentiellement de plusieurs nœuds capteurs, un nœud *Sink* et un centre de traitement des données.

- ***Nœuds*** : ceux sont des nœuds capteurs, leur type, leur architecture et leur disposition géographique dépendent de l'exigence de l'application en question. Leur énergie est limitée puisqu'ils sont alimentés par des batteries.

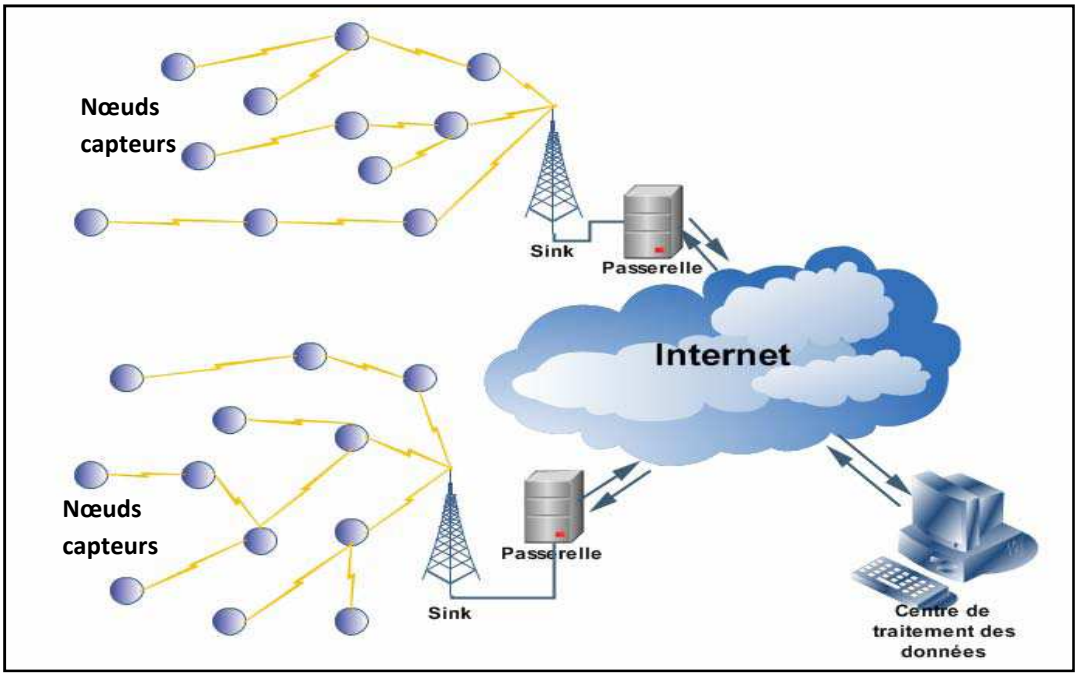


Figure 2.1– Architecture générale d'un RCSF

- ***Sink***: c'est un nœud particulier du réseau. Il est chargé de la collecte des données issues des différents nœuds du réseau. Il doit être toujours actif puisque l'arrivée des informations est aléatoire. C'est pourquoi son énergie doit être illimitée. Dans un RCSF plus ou moins large et à charge peu élevée, nous pouvons trouver deux *Sink* ou plus pour alléger la charge [5],Il existe principalement trois types de *Sink* que nous pouvons voir sur la figure 2.2.

  - Un nœud appartenant au réseau comme n'importe quel autre nœud.
  - Une entité extérieure au réseau. Par exemple, un ordinateur portatif ou un PDA (en anglais Personal Digital Assistant) [21] interagissant avec le réseau.
  - Une passerelle vers un autre réseau (Internet par exemple), où la demande d'information vient d'un centre de traitement lointain.

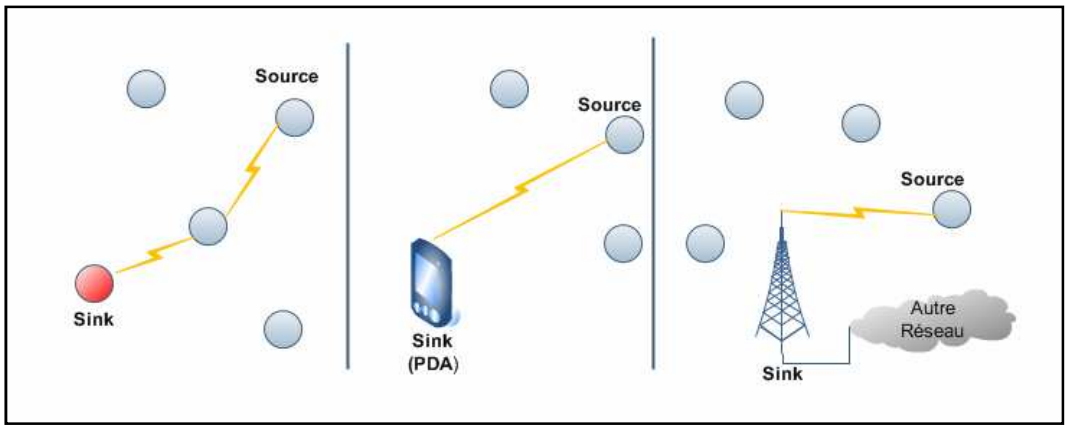


Figure 2.2 – Différents types de Sink

- ***Centre de traitement des données***: c'est le centre vers lequel les données collectées par le *Sink* sont envoyées. Ce centre a pour rôle de regrouper les données issues des nœuds et les traiter de façon à en extraire de l'information utile et exploitable. Le centre de traitement peut être éloigné du *Sink*, alors les données doivent être transférées à travers un autre réseau, c'est pourquoi nous introduisons une passerelle entre le *Sink* et le réseau de transport afin d'adapter le type de données au type de canal comme illustré dans la figure 2.1 [5].

2.2.2 Caractéristiques des RCSFs: défis et exigences

La nature collaborative des RCSFs apporte plusieurs avantages par rapport aux réseaux ad-hoc sans fil classiques, par exemple l'auto-organisation, le déploiement rapide, la souplesse et une capacité de traitement intelligent inhérent. Toutefois, les caractéristiques uniques des RCSFs représentent de nouveaux défis dans la conception matérielle, les protocoles de communication ainsi que les applications de conception. Une technologie de RCSFs doit relever ces défis pour réaliser les nombreuses applications envisagées. Cela nécessite la modification des protocoles existants pour les réseaux Ad-hoc sans fil classiques [22]ou la conception de nouveaux protocoles de communication et d'algorithmes efficaces [5,20].

Défis	***Mécanismes nécessaires***
Les contraintes de ressources	Utilisation efficace des ressources
Conditions d'environnement dynamique et extrêmes	Fonctionnement adaptatif du réseau
Redondance des données	Fusion et compression des données, traitement localisé
Communication sans fil peu fiable	Mécanisme de fiabilité
Identification non globale (ID) pour les nœuds capteurs	Paradigme de communication des données centrées
Défaillance des nœuds capteurs	Tolérance aux fautes
Déploiement à grande échelle	Faible coût, petite taille des capteurs avec auto-configuration et auto-organisation

Table2.1 – Défis vs Mécanismes nécessaires aux RCSFs

La table 2.1 résume d'importants défis ainsi que les mécanismes nécessaires leurs correspondants dans un RCSF.
Les nœuds capteurs possèdent des contraintes de ressources limitées telles que l'énergie, la mémoire ainsi que la capacité de calcul. L'énergie limitée des nœuds capteurs dans le réseau impose des contraintes sur la durée de vie d'un RCSF qui est liée directement à la durée de vie des nœuds capteurs. Le problème des ressources limitées peut être résolu en les utilisant efficacement. Les opérations d'économie d'énergie sont nécessaires pour maximiser la durée de vie du réseau en implémentant des protocoles à efficacité énergétique. Par exemple, le

protocole (EAR en anglais *Energy-Aware Routing*) [23] dans la couche réseau, le mode économie d'énergie au niveau de la couche MAC (*Medium Access Control*), etc. L'utilisation efficace de la mémoire limitée des capteurs est nécessaire en tenant compte de la problématique de la mémoire consommée telles que les tables de routage, les données dupliquées, la sécurité, etc. La topologie dynamique du réseau ainsi que les conditions d'environnement difficiles peuvent causer la défaillance des nœuds capteurs et la dégradation des performances du réseau. Cela oblige les RCSFs à prendre en compte des opérations adaptatives du réseau, y compris les algorithmes de traitement de signal adaptatifs ainsi que des protocoles de communication pour permettre aux utilisateurs finaux de faire face aux conditions du canal sans fil dynamique et aux variations de la connectivité [24].
La communication dans un RCSF n'est pas fiable en raison d'une erreur du support sans fil avec un taux d'erreurs élevé et une capacité de liens variable. Ainsi, un RCSF doit être fiable dans le but de fonctionner correctement et dépend des exigences de l'application, les données mesurées doivent être acheminées de manière fiable au nœud *Sink*.

Les RCSFs sont généralement sujets à des défaillances inattendues des nœuds dus à différentes raisons telles que : les nœuds peuvent manquer d'énergie ou peuvent être endommagés (des conditions d'environnement extrêmes), ou bien une communication sans fil entre deux nœuds peut être interrompue de façon permanente. Cela oblige les RCSFs à être robuste à la défaillance des nœuds. Dans un RCSF, la tolérance aux pannes peut être améliorée grâce à un niveau élevé de redondance en déployant des nœuds supplémentaires si tous les nœuds fonctionnent correctement. En cas de déploiement dense, les observations des capteurs peuvent être fortement corrélées dans le domaine spatial. La fusion de données et les processus de localisation sont tenus de régler le problème de redondance des données de telle sorte que seules les informations nécessaires sont délivrées à l'utilisateur final et la communication Overhead peut être réduite.
Étant donné que les RCSFs peuvent contenir un grand nombre de nœuds capteurs, l'architecture et les protocoles employés doivent être capables de mesurer des tailles de milliers ou plus. Par ailleurs, un déploiement à grande échelle de RCSF nécessite un faible coût et une petite taille des nœuds capteurs.
Un RCSF doit être en mesure de s'auto-organiser, comme la topologie du réseau change pour des raisons telles que la défaillance d'un nœud, la mobilité ainsi que les déploiements à grande échelle.
En outre, de nouveaux nœuds peuvent rejoindre le réseau, par exemple, remplacer les nœuds défaillants, ainsi un RCSF doit pouvoir s'auto-reconfigurer. Ça doit être coûteux de donner une adresse unique à chaque nœud (*paradigme Addess-Centric*) [23], surtout lorsque des milliers de nœuds sont déployés dans l'application. L'identification globale des capteurs dans un RCSF conduit à un important Overhead. Cependant, en raison de la mémoire limitée et de la faible puissance de calcul, il n'est pas fiable de dépendre du contenu d'un seul nœud capteur. Ainsi, les RCSFs sont tenus d'utiliser le (*paradigme de Data-Centric*) [25], qui se focalise sur les données générées par un groupe de capteurs.

2.2.3 Nœud capteur

Un nœud capteur est un dispositif électronique doté d'une autonomie et ayant des capacités de stockage, de calcul et d'énergie limitées. Ce dispositif est aussi capable de collecter des informations et de les communiquer à un centre de contrôle via un *Sink*.

Les principaux avantages des nœuds capteurs sont : leur taille réduite, leur très faible consommation électrique et surtout leur capacité à communiquer sans fil (ce qui permet une grande liberté de mouvement par rapport aux nœuds filaires) [23].

2.2.4 Composants d'un nœud capteur

Un nœud capteur est composé de quatre unités de base (voir figure 2.3):

- ***L'unité de calcul (Processing unit) :*** elle est généralement associée à une mémoire pour le stockage des programmes et elle permet de contrôler les procédures de collaboration entre les nœuds capteurs.
- ***L'unité de communication (Transceiver unit) :*** appelée aussi module radio, elle comprend un émetteur/récepteur permettant la communication entre les différents nœuds du réseau via un support de communication radio de faible portée. Les radios fonctionnent en plusieurs modes: transmission (*Tx*), réception (*Rx*), écoute de la porteuse (*idle*) et inactif (*sleep*) pour la conservation d'énergie [24**]**.
- ***L'unité d'énergie (Power unit) :*** disposée souvent sous forme de batteries pouvant alimenter les unités citées ci-dessus et sont souvent non rechargeables et non remplaçables. C'est l'unité critique dont dépend la durée de vie du nœud capteur et par conséquent celle du réseau tout entier.
- ***L'unité de détection (Sensing unit) :*** appelée aussi ***unité de capture***. Cette unité possède les valeurs sur les paramètres à mesurer et se compose de deux sous-unités :
  - Sous-unité « Capteurs » constituée d'un ou de plusieurs capteurs
  - Sous-unité « Analogue To Digital Converter : ADC » responsable de la conversion du signal analogique en signal numérique.

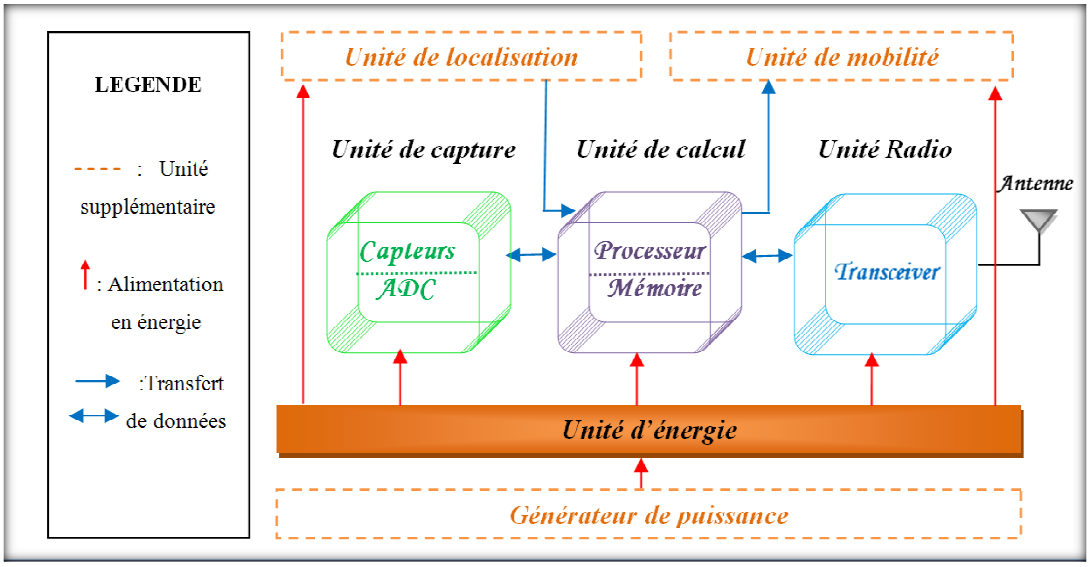


Figure 2.3– Architecture matérielle d'un nœud capteur

Suivant le domaine d'application; le capteur peut contenir d'autres unités telles que [26]:

- ***L'unité de localisation (Location finding system) :*** fournit des informations sur la localisation des cibles.
- ***Le générateur de puissance (Power générator) :*** permet de générer de l'énergie à partir de l'environnement.
- ***L'unité de mobilité (Mobilizer) :*** indispensable, si les nœuds doivent se déplacer, elle leur donne la possibilité de changer d'emplacement durant leurs mouvements fréquents.

Cette unité est utilisée avec une forte intensivité dans les RCSFs-EM, type de réseaux auquel nous nous intéressons dans ce travail.

2.2.5 Différents types de nœud

Pour mieux comprendre les systèmes physiques et par la suite, les différentes stratégies adoptées pour dimensionner et architecturer un RCSF, nous utilisons des modèles aussi simples que possibles. Dans cette section, nous définissons les différentsmodèles de nœuds utilisés dans les RCSFs [27].
Selon l'application et la structure choisie, un RCSF peut contenir différents types de nœuds.

- ***Noeud régulier :*** c'est un nœud doté d'une unité de transmission et d'une unité de traitement de données. L'unité de transmission de données est responsable de toutes les émissions et réceptions de données via un support de communication sans fil pouvant être de type optique (comme dans les nœuds Smart Dust[28]) ou de type radiofréquence (comme dans les nœuds Stargate[29]). L'unité de traitement de données est composée d'une mémoire, d'un microcontrôleur et d'un système d'exploitation spécifique (comme TinyOS[30], développé à l'université de Berkeley et actuellement utilisé par plus de 500 universités et centres de recherches à travers le monde). Elle est responsable du traitement des données en

provenance ou à partir de l'unité de transmission. Ces deux unités sont alimentées par une batterie embarquée comme le montre la figure 2.3. Selon le domaine d'application, un nœud peut être équipé d'unités supplémentaires ou optionnelles comme un système de localisation GPS (*Global Positioning System* [31], etc.) pour déterminer sa position, ou bien un système générateur d'énergie (cellule photovoltaïque, etc.), ou encore un système mobile pour lui permettre de changer sa position ou sa configuration en cas de nécessité.

- ***Nœud capteur*** ou nœud source : c'est un nœud régulier équipé d'une unité d'acquisition ou de détection. L'unité d'acquisition est généralement dotée d'un ou de plusieurs capteurs qui obtiennent des mesures analogiques (physiques et physiologiques) et d'un convertisseur Analogique/Numérique qui convertit l'information relevée en un signal numérique compréhensible par l'unité de traitement afin de les transmettre, directement ou via une communication multi-sauts à un utilisateur final.

- ***Nœud actionneur* :**ou robot est un nœud régulier doté d'une unité lui permettant d'exécuter certaines tâches spécifiques comme des tâches mécaniques (se déplacer, combattre un incendie, piloter un automate, etc.).

- ***Nœud puits (Sink) :***c'est un nœud régulier doté d'un convertisseur série connecté à une seconde unité de communication (GPRS en anglais *Global Packet Radio Service*[32]), Wi-Fi[33], WiMax en anglais *Worldwide Interoperability for Microwave Access*) IEEE 802.16 [34], etc.). La seconde unité de communication fournit une retransmission transparente des données provenant de nœuds capteurs à un utilisateur final ou d'autres réseaux comme internet.

- ***Nœud passerelle (ou gateway) :***c'est un nœud régulier permettant de relayer le trafic dans le réseau sur le même canal de communication.

Nous pouvons voir l'architecture de chacun de ces nœuds (nœud régulier, nœud capteur, nœud puits, nœud robot et nœud passerelle) sur la figure 2.4 [27] :

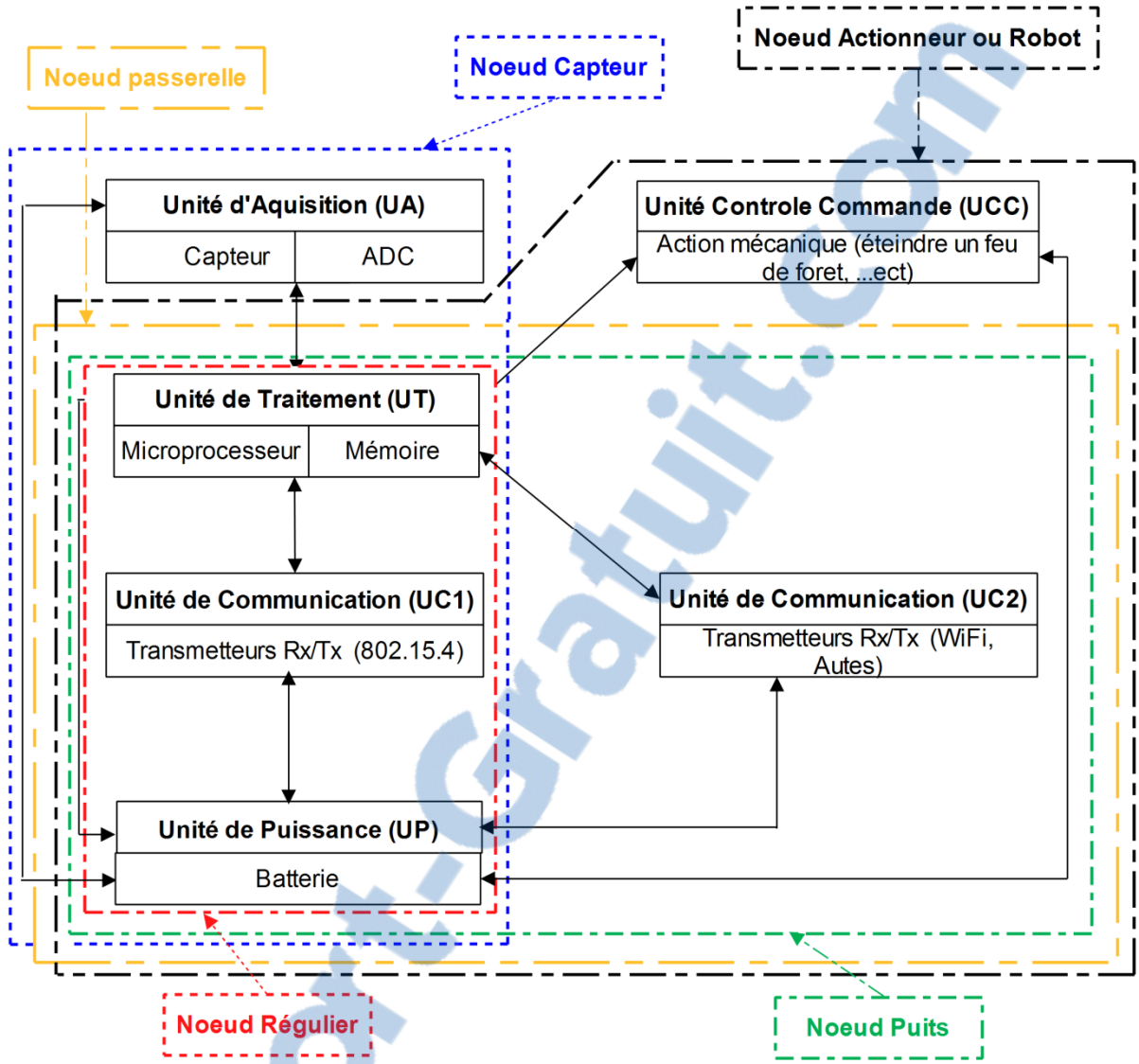


Figure 2.4 – Architecture des différents types de nœuds : régulier, capteur, robot, puits, passerelle

2.2.6 Plates-formes existantes de nœuds capteurs sans fil

Dans certaines technologies d'aujourd'hui, les nœuds capteurs sans fil sont nés d'un projet militaire, ce qui entrave la mise en place d'une chronographie précise de leur développement. Néanmoins, le premier prototype de nœuds capteurs sans fil identifiable dans la bibliographie correspond au module LWIM (Low-power Wireless Integrated Microsensors) développé au milieu des années 90 par l'Agence des Projets de Recherche Avancée de Défense(*DARPA*) des Etats-Unis [35]. Il s'agissait d'un géophone équipé d'un capteur de transmission radio et d'un contrôleur PIC. Depuis plus de 10 ans, la technologie des capteurs sans fil a beaucoup évolué. Les modules deviennent de plus en plus petits et les durées de vie prévues augmentent. Aujourd'hui, le marché des capteurs a été ouvert à l'industrie. Le fournisseur le plus connu est Crossbow Inc[36], avec ses capteurs Mica2 et MicaZ[37]. (Voir figure 2.5).

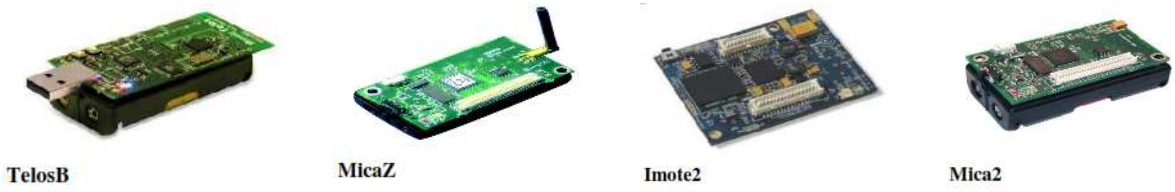


Figure 2.5 – Quelques modèles de capteurs sans fil

La Table 2.2 recense différents composants actuellement disponibles sur le marché. Parmi les modèles les plus courants, on retrouve les capteurs MICA développés par l'université de Berkeley et commercialisés par Crossbow, les capteurs Imote (Imote 2 et Tmote sky [38,39]) sont aussi commercialisés par Crossbow. Les capteurs TinyNode développés par la compagnie Shockfish SA [40], pour des applications réelles liées à l'industrie. Notons que bien qu'ils soient différents, ces modèles ont tous en commun les mêmes composants de base [35].

	Unité de Traitement				**Unité de Transmission**	**Unité de Captage**	**Unité de Puissance**
Nœud Capteur sans fil	**Micro-Contrôleur**	**RAM**	**Flash**	**EEprom**	**Type Radio**		
MICA2 (Crossbow)	**ATmega 128L (8bits) 3.37 MHz**	**4KB**	**128KB**	**4KB**	**Chipcon CC1000** 38kbps	**Connecteur pour carte de capteurs externe**	2xAA
MICAZ (Crossbow)	**ATmega 128L (8bits) 3.37 MHz**	**4KB**	**128KB**	**4KB**	**Chipcon CC2420** 250kbps	**Connecteur pour carte de capteurs externe**	2xAA
Imote2 (Crossbow)	**Intel PXA271 XScale (32bits) 13-416 MHz**	**256KB + 32MB SDRAM**	**32KB**	**32KB**	**Chipcon CC2420** 250kbps	**Connecteur pour carte de capteurs externe**	3xAAA
TeloSB [41] (Crossbow)	**TI MSP 430 (16bits) 8MHz**	**10KB**	**48KB**	**16KB**	**Chipcon CC2420** 250kbps	**Connecteur pour carte de capteurs externe**	2xAA
TinyNode (Shockfish SA)	**TI MSP 430**	**10KB**	**48KB**	**16KB**	**Semtech XE 1205** 153kbps	**Connecteur pour carte de capteurs externe**	2/3xAA
BTnode 3 (ETH)	**ATmega 1281**	**64KB**	**128KB**	**4KB**	**Chipcon CC1000/Bluetooth**	**Connecteur pour carte de capteurs externe**	2xAA
Tmote Sky (Moteiv)	**TI MSP 430 F1611(16bits)** 8MHz	**10KB**	**48KB**	**128KB**	**Chipcon CC2420**	**Connecteur pour carte de capteurs externe**	2xAA

Table 2.2 – Caractéristiques des capteurs existant actuellement

Le concept prévalant dans le développement des nœuds capteurs est la conception modulaire. En effet, tous les nœuds de la table 2.2 sont en fait des cartes intégrées qui regroupent l'unité de communication et l'unité de traitement, tandis que l'unité de captage est conçue comme une carte distincte qui peut être attachée à l'unité principale. Cela permet bien sûr de pouvoir réutiliser les mêmes unités dans différentes applications.

2.2.7 Différents types de réseaux de capteurs sans fil (RCSFs)

Actuellement de nombreux RCSFs sont déployés sur terre, sous terre et sous l'eau. Ils font face à différents challenges et contraintes dépendant de leur environnement. Nous présentons ici cinq types de réseaux de capteurs [12,20] comme le montre la figure 2.6.

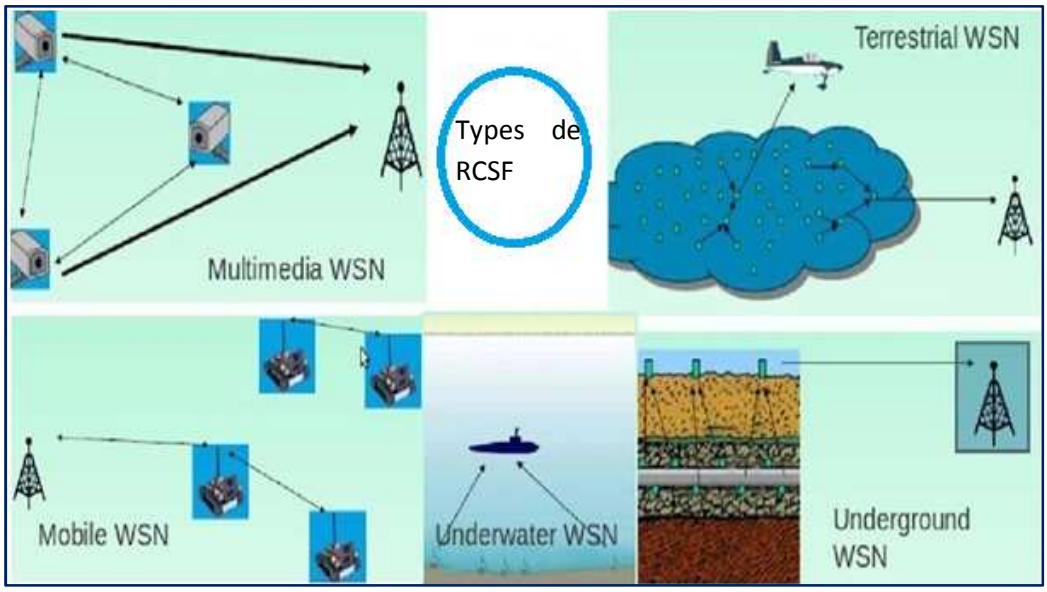


Figure 2.6 – Types de RCSFs

2.2.7.1 RCSF Terrestre (Terrestrial WSN)

Un RCSF terrestre est constitué d'un grand nombre (des centaines de milliers) de nœuds à faible coût déployés sur un espace dans une zone donnée, généralement d'une manière Ad-hoc (par exemple, les nœuds largués à partir d'un avion). Dans les réseaux de capteurs sans fil terrestres [42], les nœuds capteurs doivent être capables de communiquer les données efficacement à la station de base dans un environnement dense. Depuis que la puissance de la batterie est limitée et généralement non-rechargeable, les nœuds capteurs terrestres peuvent être équipés d'une source d'alimentation secondaire telles que les cellules solaires. L'énergie peut être conservée avec un routage multi-sauts optimal, une portée de transmission courte, l'agrégation des données dans le réseau, l'utilisation d'opérations à faible Duty-cycle. Les applications communes entre les RCSFs terrestres sont la détection environnementale et la surveillance, le contrôle industriel, ainsi que l'exploration de surfaces.

2.2.7.2 RCSF Sous-terrain (Underground WSN)

Un RCSF sous-terrain se compose d'un certain nombre de nœuds capteurs déployés dans des grottes, des mines ou des métros pour surveiller les conditions sous-terraines [43, 44]. Afin de relayer l'information à partir des nœuds capteurs souterrains à la station de base, des nœuds puits (*Sinks*) supplémentaires sont déployés au dessus du sol. Ils sont plus chers que les RCSFs terrestres car ils ont besoin d'équipements appropriés afin d'assurer une communication fiable à travers le sol, les roches et l'eau. La communication sans fil est un défi dans un tel environnement en raison de la forte atténuation et de la perte de signal. En outre, il est difficile de recharger ou de remplacer la batterie des nœuds enterrés, il est donc important de concevoir un protocole de communication énergétique efficace afin de prolonger la durée de vie du réseau. Les RCSFs sous-terrains sont utilisés dans de nombreuses applications telles que la surveillance de l'agriculture, la gestion du paysage, la surveillance souterraine du sol, de l'eau ou minéral ainsi que la surveillance militaire d'une frontière.

2.2.7.3 RCSF Sous-marin (Underwater WSN)

Un RCSF sous-marin se compose de nœuds capteurs déployés sous l'eau, par exemple, dans un environnement d'océan [45, 46]. Ces nœuds étant assez coûteux, seulement quelques nœuds sont déployés et des véhicules sous-marins autonomes sont utilisés pour explorer ou collecter les données de ces nœuds. Une communication sans fil sous-marine utilise des ondes acoustiques qui représentent divers challenges tels que la bande passante limitée, le long délai de propagation, une latence élevée, et les problèmes du signal fading. Ces nœuds doivent être capables de s'auto-configurer et de s'adapter à des conditions extrêmes de l'environnement de l'océan. Les nœuds sont équipés d'une batterie limitée qui ne peut être ni remplacée ni rechargée nécessitant une communication énergétique efficace sous-marine et des techniques de réseau. Les applications des RCSFs sous-marins incluent la surveillance de la pollution, la surveillance et l'exploration sous-marine, la prévention de catastrophes, la surveillance sismique, la surveillance d'équipements ainsi que la robotique sous-marine.

2.2.7.4 RCSF Multimédia (Multimedia WSN)

Un RCSF multimédia se compose de nœuds capteurs à faible coût équipés de caméras et de micros, déployés d'une manière planifiée afin de garantir une bonne couverture [47]. Les capteurs multimédia sont capables de stocker, traiter et récupérer des données multimédia telles que la vidéo, audio et images. Ils doivent faire face à différents défis tels qu'une forte demande de bande passante, une consommation d'énergie élevée, fournir une qualité de service (*QoS*), traitement des données et techniques de compression, ainsi que la conception inter-couches (cross-layer). Il est primordial de développer des techniques de transmission qui prennent en charge la bande passante élevée et la faible consommation d'énergie afin de fournir un contenu multimédia tel qu'un flux vidéo. Bien que fournir la qualité de service (*QoS*) soit difficile à réaliser dans les réseaux de capteurs multimédia en raison de la capacité de liaison variable et de délai, un certain niveau de qualité de service doit être atteint pour la livraison d'un contenu fiable. Les RCSFs multimédia améliorent les applications de RCSFs existantes tels que le suivi et la surveillance.

2.2.7.5 RCSF Mobile (Mobile WSN)

Un RCSF mobile se compose de nœuds capteurs mobiles qui peuvent se déplacer et interagir avec l'environnement physique [48]. Les nœuds mobiles peuvent se repositionner et s'organiser dans le réseau, de plus ils sont capables de détecter, calculer et communiquer les données.

Un algorithme de routage dynamique doit, par conséquent, être utilisé contrairement au routage fixe dans les RCSFs stationnaires (RCSF-S).

Les RCSFs mobiles font face à divers défis tels que : le déploiement, la gestion de la mobilité, la localisation de l'élément mobile, la navigation et le contrôle de robots mobiles, la minimisation de la consommation d'énergie pendant les déplacements, le maintien de la connectivité du réseau ainsi que la distribution de données. Les principaux exemples d'application de RCSFs mobiles sont dédiés à la surveillance (environnement, l'habitat, sous-marin), surveillance militaire, le suivi de cible, recherche et sauvetage. Un degré plus élevé de couverture et de connectivité peut être atteint avec des nœuds capteurs mobiles par rapport aux nœuds statiques.

Ces réseaux, objet de notre étude, représentent la technologie la plus attractive et la plus célèbre des réseaux fondés sur le paradigme réseaux Ad-hocs à l'heure actuelle. Nous leur réservons tout le reste de ce chapitre.

De manière générale, le concept de mobilité fait référence à la capacité d'une entité à se mouvoir. L'atout majeur de la robotique mobile réside ainsi dans la capacité à évoluer face à la dangerosité du milieu, sa nocivité ou sa difficulté d'accès. Le déploiement de robots mobiles pour une intervention humaine dangereuse ou très risquée. La figure 2.7dépeint une variété de robots mobiles spécifiés par le milieu sollicité, leur taille et leurs capacités. Par analogie aux capteurs ne représentant individuellement que des machines rudimentaires, une intelligence ambiante émerge de l'assemblage des robots [5].

Figure 2.7– Exemples de robots mobiles [5]

2.2.8 Domaines d'application des RCSFs (du traditionnel vers le moderne)

Par leur réalisme et leur apport concret, les RCSFs ont su se démarquer de leur origine Ad-hoc, de leur facilité de déploiement ainsi que leur faible cout, ils ont su attirer un grand nombre d'industriels et d'organisations civiles et militaires. Nous les retrouvons également dans le domaine médical, commercial ainsi qu'environnemental où la surveillance et la reconnaissance de phénomènes physiques demeurent une tache prioritaire [5,49]. Nous distinguons plusieurs domaines d'application répandus actuellement [20].
Nous pouvons voir sur la figure 2.8 des exemples d'applications de RCSFs déployés dans un environnement réel.

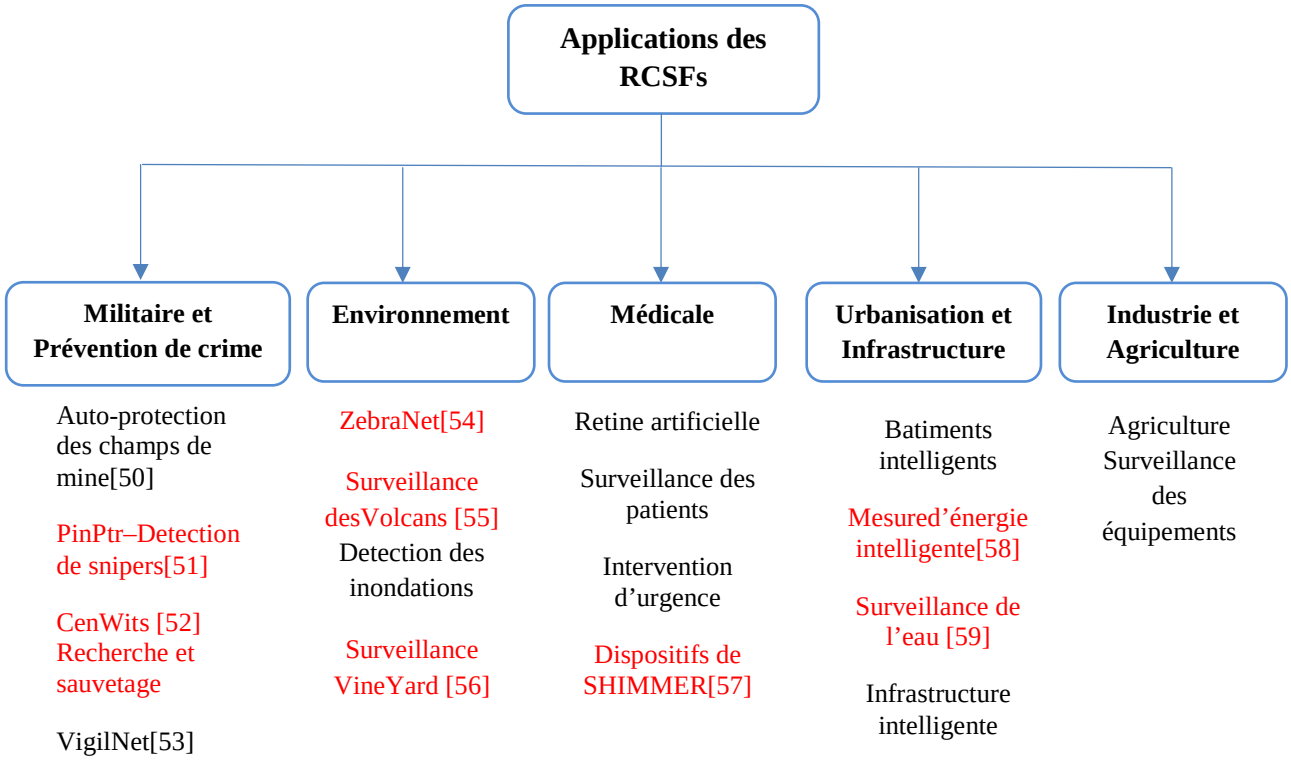


Figure 2.8 – Exemples d'applications de RSCFs déployés dans un environnement réel

- ***Applications militaires :*** l'utilisation des capteurs dans le domaine militaire est en pleine expansion. Ces dispositifs peuvent être utilisés dans les opérations de surveillance des champs de bataille, la détection d'intrusion et reconnaissance des forces amies et ennemies, un déploiement dans un endroit stratégique ou difficile d'accès, afin de surveiller toutes les activités des forces ennemies ou d'analyser le terrain avant d'y envoyer des troupes (par la détection d'agents chimiques, biologiques ou de radiations par exemple).
Parmi les travaux concrétisés dans ce domaine, nous pouvons citer les projets phares suivants: le projet WATS (*Wide Area Tracking System*) pour la détection des dispositifs nucléaires développés par le laboratoire *Lawrence Livermore National* [60].

- ***Applications médicales :***dans le domaine médical, les capteurs sont utilisés pour la surveillance des données physiologiques d'un patient. A titre d'exemple, la référence [61] propose une nouvelle plateforme pour la surveillance des personnes cardiaques en utilisant les capteurs pour la collecte des données à partir d'un Électrocardiogramme *ECG* (la durée QRS,

la durée entre deux piques R, l'amplitude du pique R) et le téléphone mobile pour la détection des pathologies cardiaques.

- ***Applications commerciales* :** dans les entreprises, les RCSFs permettent de suivre le procédé de production à partir des matières premières jusqu'au produit final livré. Grâce aux réseaux de capteurs, les entreprises peuvent offrir une meilleure qualité de service tout en réduisant les coûts, en évitant les pannes et ceci en surveillant l'état des machines [62,63].
- ***Applications environnementales :*** les RCSFs peuvent être utilisés pour surveiller les changements environnementaux [19,64]. Ils servent à déterminer les valeurs de certains paramètres à un endroit donné, comme par exemple : la température, la pression atmosphérique, à surveiller le taux de pesticides dans l'eau potable, le degré d'érosion, le niveau de pollution de l'air en temps réel etc. En dispersant des nœuds capteurs dans la nature, nous pouvons détecter des événements tels que des feux de forêts, des tempêtes ou des inondations. Ceci permet une intervention beaucoup plus rapide et efficace des secours. Avec les RCSFs, nous pouvons contrôler la pollution, par exemple en déposant des capteurs au-dessus d'un emplacement industriel pour détecter et surveiller des fuites de gaz ou de produits chimiques. Dans la référence [65], les auteurs décrivent leurs efforts de mise en œuvre d'un RCSF sur le volcan « Reventator » dans la partie occidentale de l'amazone en Équateur. Un autre exemple concernant les applications environnementales est le projet ARGO [66]. Le but de ce projet est de surveiller l'eau de l'océan, sa température, sa salinité ainsi que sa vélocité. Le projet utilise des nœuds équipés de capteurs de température et de salinité. Les nœuds sont déployés à partir de navires ou bien d'avions. Ils s'enfoncent à une profondeur de 2000 mètres tous les dix jours. Les données collectées au cours des déplacements sont transmises à un satellite tandis que des nœuds sont toujours à la surface. La durée de vie des nœuds est d'environ 45 ans.
- ***Applications domestiques :*** les RCSFs peuvent également être utilisés dans la domotique et l'environnement intelligent. Ils jouent un rôle essentiel dans les grandes usines et les entrepôts en surveillant les changements climatiques. Par exemple, des capteurs peuvent être utilisés pour contrôler les vibrations susceptibles d'endommager la structure d'un bâtiment. Dans la référence[67], les auteurs décrivent une application qui surveille l'état de grandes structures comme des immeubles administratifs. Ils exploitent les avantages d'un réseau de capteurs tels que le déploiement rapide (environ une demi-heure face à plusieurs jours pour l'installation des réseaux filaires). Un RCSF a été déployé dans un campus universitaire [68], permettant aux différentes machines (serveurs, imprimantes, etc) de tous les départements de communiquer ensemble.
- ***Applications dans le domaine sportif* :** l'évolution des RCSFs est utilisée de plus en plus dans le domaine sportif, à savoir les systèmes de surveillance, les systèmes de calcul de trajectoires (par exemple dans le tennis, ils utilisent un système appelé Hawk eye : l'œil du Faucon) [69], les systèmes de détection d'erreurs d'arbitrage (dans le football, ils indiquent si le balle a franchi la ligne de but).

- ***Autres applications :*** parmi les autres applications de RCSFs, nous avons: l'agriculture, où des capteurs sont incorporés dans la terre pour donner des informations sur l'état du champ. La protection des barrages pourrait être accomplie en y introduisant des capteurs. La détection

prompte de fuites d'eau permettrait d'éviter des dégâts. Les êtres humains sont conscients des risques et attaques qui les menacent. Ainsi, ils mettent à disposition toutes les ressources humaines et financières nécessaires pour leur sécurité. Grâce aux RCSFs, les entreprises pourraient offrir une meilleure qualité de service tout en réduisant leurs coûts. Les RCSFs peuvent également être utilisés pour surveiller l'infrastructure, lutter contre le terrorisme, contrôler le trafic, détecter des intrusions (en plaçant, à différents points stratégiques, des capteurs), contrôler les stocks (savoir le lieu, la quantité, la forme de tous les produits) [4 ,19]. Nous trouvons aussi la gestion de crises et de catastrophes naturelles (séismes, inondations, éruptions volcaniques, glissements de terrains, tempête, tsunamis, incendies de forêts, cyclones...). En effet ces phénomènes, en nette accentuation de nos jours, corroborent les prévisions de leur aggravation dans le futur avec comme principal facteur le changement global du climat et exigent l'intervention rapide et efficace des secours. La figure 2.9 représente plusieurs domaines d'application des RCSFs.

Figure 2.9 – Domaine d'Applications des RCSFs

2.2.9 Architecture protocolaire des RCSFs

La pile de protocoles utilisée par le puits (*Sink*) ainsi que par tous les nœuds-capteurs est donnée dans la figure 2.10. Cette pile de protocoles combine routage, gestion d'énergie et intègre les données avec les protocoles réseaux. Elle communique de manière efficace (en termes d'énergie) à travers le support sans fil et favorise les efforts de coopération entre les nœuds-capteurs.

La pile de protocoles comprend une couche application, une couche transport, une couche réseau, une couche liaison de données, une couche physique, un plan de gestion d'énergie, un plan de gestion de mobilité et un plan de gestion de tâches. Selon les tâches de détection, différents types de logiciels d'application peuvent être construits et utilisés dans la couche application. La couche transport contribue au maintien du flux de données si l'application du

RCSF l'exige. La couche réseau s'occupe de l'acheminement des données fournies par la couche transport. Comme l'environnement est sujet au bruit et que les nœuds-capteurs peuvent être mobiles, le protocole MAC doit tenir compte de la consommation d'énergie et doit être en mesure de réduire les collisions entre les nœuds voisins lors d'une diffusion par exemple [70].

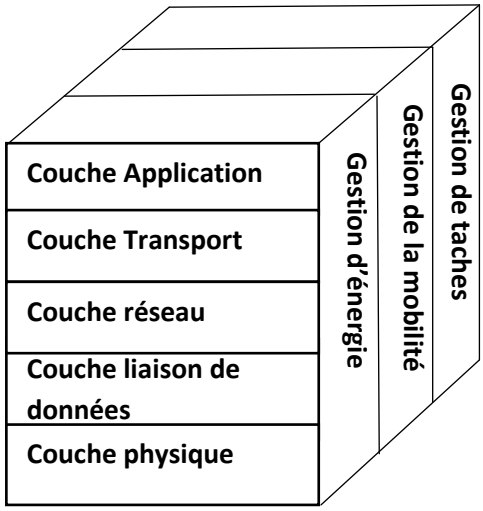


Figure 2.10 – Pile protocolaire des RCSFs

- ***Couche physique :*** certains circuits radio permettent de paramétrer leur mode d'opération à l'instar du CC1101 de Texas Instruments [71 ,72]. Parmi ces modes, nous trouvons le choix de la modulation, le type de codage correcteur et la puissance de transmission. Ces éléments de configuration permettent d'influer sur la fiabilité et le débit des communications.

- ***Couche liaison :*** dans la pile de communication, la couche MAC définit le protocole d'accès aux ressources radio. Elle définit ainsi l'ensemble des opérations qui mènent à la transmission d'une trame d'information entre deux nœuds à portée de communication. Le protocole d'établissement d'une communication affecte de manière évidente la fiabilité et le délai de transmissions.

- ***Couche réseau* :** la couche réseau, et plus particulièrement le protocole de routage définit la succession de transmissions qui doivent être exécutées afin d'acheminer une information d'un nœud à un autre dans un réseau multi-sauts. Cette tâche est assurée par deux mécanismes : le plan de contrôle, qui construit et maintient la topologie de routage, et le plan de données, qui assure l'acheminement des trames dans la topologie.

- ***Couche transport* :** la couche transport est en charge du contrôle de bout en bout de la communication entre un émetteur et son destinataire. Traditionnellement, dans les réseaux filaires, la Qualité de Service est implémentée à ce niveau. Les protocoles associés peuvent, à l'instar de (*TCP* en anglais *Transmission Control Protocol*) [73], prendre en

charge le contrôle du flux de données en implémentant un ordonnancement des trames de données, des mécanismes d'acquittement (*ACK*,*NACK*), et du contrôle de congestion, etc.

- ***Couche application :*** afin de compléter le recensement des mécanismes de Qualité de Service, nous citons ceux propres à l'application. Une application peut en effet intégrer dans son fonctionnement des mécanismes d'acquittement de données, de contrôle d'erreur sur les trames, et ce, similairement aux mécanismes disponibles dans la couche liaison.
- ***Plans de gestion d'énergie, de mobilité et des tâches :*** surveillent et gèrent la consommation d'énergie, les mouvements ainsi que la répartition des tâches entre les nœuds-capteurs. Ces plans aident les nœuds-capteurs à coordonner les tâches de détection et à réduire la consommation d'énergie.

2.2.10 Contraintes des RCSFs

Les RCSFs forment une branche à part, bien qu'ils fassent partie du domaine des réseaux Ad-Hoc. Cependant ils ont une contrainte commune qui est le partage du média sans fil. Ce partage implique une limitation de la bande passante réservée à un nœud. Et parmi les autres contraintes nous pouvons citer:[70,74,4]

- ***Durée de vie du réseau :*** elle commence dès le déploiement du réseau et dure jusqu'à l'épuisement de l'énergie du premier nœud, cela peut durer quelques jours à plusieurs années selon l'application pour laquelle le réseau est utilisé.
- ***Ressources énergétiques limitées :*** vu que la taille du nœud capteur est réduite, la batterie a une faible capacité et l'énergie disponible est très limitée, malgré cela le réseau doit fonctionner pendant une durée relativement longue. Le problème aussi du rechargement/remplacement des batteries qui est quasiment impossible et l'objectif principal dans la conception de ces protocoles est d'exploiter cette énergie limitée de manière efficace.
- ***Bande passante limitée :*** les capteurs communiquent avec un faible débit de quelques dizaines de Kb/s afin de préserver leur énergie, cela n'influe en aucun cas sur leur fonctionnement car les fréquences de transmissions ne sont pas très importantes.
- ***Faible coût de production :*** puisque les capteurs sont déployés en grand nombre, leur coût de production doit rester aussi faible que possible. Ceci implique d'autres contraintes principalement matérielles telles que (faible capacité de mémoire, faible puissance de calcul, etc…). En effet optimiser le coût de production revient à optimiser le coût des composants matériels [4].
- ***Tolérance aux pannes :*** un RCSF doit être en mesure de continuer à fonctionner normalement malgré la défaillance de certains nœuds qui peuvent générer des erreurs dus à un problème d'énergie, physique ou bien à cause d'interférences.
- ***Temps de réponse :*** ce facteur est une métrique de performance des capteurs, surtout lorsqu'ils sont déployés dans des applications d'alarmes où le besoin d'agir immédiatement est primordial. Cette métrique est aussi importante dans le cas où les capteurs sont utilisés pour le contrôle des systèmes ou des équipements.
- ***Passage à l'échelle ou Scalabilité*** : cette contrainte est considérée comme un facteur critique pour les RCSFs. Toute solution de protocoles acceptable proposée à une certaine échelle doit garantir au réseau des performances stables lorsque le réseau passe à une

échelle supérieure. Ce facteur doit être pris explicitement en considération pendant la phase de conception [10].

2.3 Consommation d'énergie d'un nœud capteur

2.3.1 Formes de dissipation d'énergie

Les nœuds-capteurs sont alimentés principalement par des batteries. Ils doivent donc fonctionner avec un bilan énergétique frugal. En outre, ils doiventle plus souvent, avoir une durée de vie de l'ordre de plusieurs mois, voire de quelques années, puisque le remplacement des batteries n'est pas une option envisageable pour des réseaux avec des milliers de nœuds.
Afin de concevoir des solutions efficaces en énergie, il est extrêmement important de faire d'abord une analyse des différents facteurs provoquant la dissipation de l'énergie d'un nœud-capteur [70].
Cette dissipation d'énergie se fait de manière générale selon plusieurs modes :

- ***MCU (Unité du microcontrôleur ou Micro Controller Unit) :*** généralement les MCUs possèdent divers modes de fonctionnement : « actif », « idle », et « sommeil », à des fins de gestion d'énergie. Chaque mode est caractérisé par une quantité différente de consommation d'énergie. Par exemple, le MSP430 (Les capteurs Tmote Sky[8]sont équipés de ce microcontrôleur) consomme 3 mW en mode actif, 98 µW en mode "*idle*" et seulement 15 µW en mode sommeil. Toutefois, la transition entre les modes de fonctionnement implique un surplus d'énergie et de latence. Ainsi, les niveaux de consommation d'énergie des différents modes, les coûts de transition entre les modes ou encore le temps passé par le MCU dans chaque mode ont une incidence importante sur la consommation totale d'énergie d'un nœud-capteur.
- ***Radio :*** opère dans quatre modes de fonctionnement : émission, réception, idle, et réveil. Une observation importante dans le cas de la plupart des radios est que le mode "*idle*" implique une consommation d'énergie significative, presque égale à la consommation en mode réception [75]. Ainsi, il est plus judicieux d'éteindre complètement la radio plutôt que de passer en mode "*idle*" en l'absence d'émission et de réception de données. Un autre facteur déterminant est que, le passage de la radio d'un mode à un autre, engendre une dissipation d'énergie importante due à l'activité des circuits électroniques. Par exemple, quand la radio passe du mode sommeil au mode émission pour envoyer un paquet, une importante quantité d'énergie est consommée pour le démarrage de l'émetteur luimême. Un autre point important est que les données des constructeurs sous-estiment assez régulièrement ces différentes consommations comme ont pu le montrer les auteurs de [76], en particulier concernant la consommation d'énergie dans le mode "*idle*"(écoute de la porteuse).
- ***Détecteur ou capteur proprement dit :*** Il existe plusieurs sources de consommation d'énergie par le module de détection, notamment l'échantillonnage et la conversion des signaux physiques en signaux électriques, le conditionnement des signaux et la conversion analogique-numérique.

Etant donné la diversité des capteurs, il n'y a pas de valeurs typiques de l'énergie consommée. En revanche, les capteurs passifs (de température, sismiques, ...) consomment

souvent moins d'énergie que d'autres nœuds capteurs. Notons que les capteurs actifs tels que les sonars, les capteurs d'images peuvent consommer beaucoup plus d'énergie.
En outre, il existe d'autres formes de dissipation d'énergie telles que les lectures et les écritures mémoire.
Un autre aspect non négligeable est le phénomène d'auto-chargement de la batterie. En effet, cette dernière se décharge d'elle même et perd de sa capacité au fil du temps.
Il est difficile d'apporter ici une étude quantitative et comparative précise de la consommation de chaque composant d'un nœud-capteur en raison du grand nombre de plates-formes commerciales existantes. Cependant, des expérimentations ont montré que c'est la transmission de données qui est la plus consommatrice en énergie. Le coût d'une transmission d'un bit d'information est approximativement le même que le coût nécessaire pour le calcul d'un millier d'opérations [77]. La consommation du module de détection dépend du type spécifique du nœud-capteur.

2.3.2 Sources de gaspillage d'énergie

Nous appelons gaspillage d'énergie toute consommation inutile que l'on peut éviter afin de conserver l'énergie d'un nœud-capteur. Les sources de ce gaspillage sont nombreuses, elles peuvent être engendrées lors de la détection lorsque celle-ci est mal gérée (par exemple par une fréquence d'échantillonnage mal contrôlée) [78].
Le gaspillage concerne également la partie communication. En effet, cette dernière est sujette à plusieurs phénomènes qui gaspillent de l'énergie surtout au niveau de la couche MAC où se déroule le contrôle d'accès au support sans fil. Cette couche a suscité l'intérêt de plusieurs chercheurs ces dernières années du fait qu'elle soit à l'origine des différentes sources de perte d'énergie que nous pouvons résumer selon [10,79,80,81] dans ce qui suit :

- ***Collisions***: puisque le canal radio est partagé par plusieurs nœuds capteurs, une collision aura lieu chaque fois que plus de deux nœuds tentent de transmettre simultanément leurs paquets. Les collisions augmentent à la fois la consommation d'énergie et la latence de délivrance des paquets.
- ***Ecoute de la porteuse à vide (Idle listening)*** :un nœud dans un état de non activité écoute en permanence la porteuse pour savoir s'il est ou non récepteur d'un éventuel trafic. Dans cette situation, la quantité d'énergie dépensée est équivalente à une réception normale.
- ***Réception indésirable (Overhearing)*** :se produit lorsqu'un nœud capteur reçoit des paquets qui ne lui sont pas destinés. Ceci est dû essentiellement à la nature de la transmission radio qui est omni directionnelle obligeant chacun des nœuds du voisinage à gaspiller de l'énergie en recevant cette radio.
- ***Non disponibilité du récepteur (Over emitting)*** : ce cas se produit lorsqu'un nœud capteur reçoit un paquet alors qu'il se trouve toujours en état de veille. Cette situation oblige l'émetteur à effectuer de nouvelles retransmissions afin de réussir sa transmission. Ces retransmissions, qui sont étroitement liées au problème de la désynchronisation, consomment plus d'énergie.
- ***Paquets de contrôle (Overhead)***: les entêtes des paquets MAC et les paquets de contrôle utilisés pour les signalisations (*RTS/CTS/ACK* :*Request-To-Send*/*Clear-To Send/ACKnowledgement*) ne contiennent pas les données propres à l'application et sont

donc considérés comme des données supplémentaires (overhead). Ces paquets de contrôle peuvent être significatifs car la plupart des applications utilisent des paquets de données de taille très réduite.

2.4 Techniques de conservation d'énergie

Plusieurs solutions de conservation d'énergie ont été proposées dans le domaine des RCSFs, partant de la couche physique jusqu'à la couche applicationen passant par des techniques de modulation et développement de logiciels spécialisés. Une classification des différentes techniques de conservation d'énergie est donnée dans la figure 2.11[82]:

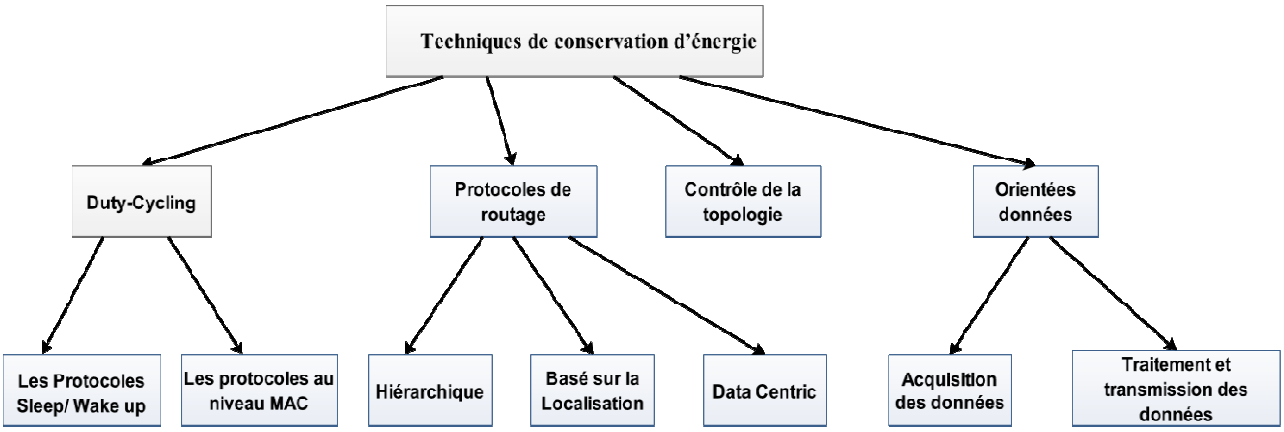


Figure 2.11 – Techniques de conservation d'énergie

- **Cycle d'activité (Duty-cycle) :** cette technique est principalement utilisée dans l'activité du réseau. Le moyen le plus efficace pour conserver l'énergie est de mettre la radio de l'émetteur en mode veille (low-power) à chaque fois que la communication n'est pas nécessaire. Idéalement, la radio doit être éteinte dès qu'il n'y a plus de données à envoyer et/ou à recevoir, et devrait être prête dès qu'un nouveau paquet de données doit être envoyé ou reçu.
  Ainsi, les nœuds alternent entre périodes d'éveil et de sommeil en fonction de l'activité du réseau. Un cycle d'activité est défini comme étant la fraction de temps où les nœuds sont actifs. Comme les nœuds-capteurs effectuent des tâches en coopération, ils doivent coordonner leurs dates de sommeil et de réveil. Un algorithme d'ordonnancement Sleep/Wake up(sommeil/réveil) accompagne donc tout plan de Duty-cycle. Il s'agit généralement d'un algorithme distribué reposant sur les dates auxquelles des nœuds décident de passer de l'état actif à l'état de sommeil. Il permet aux nœuds voisins d'être actifs en même temps, ce qui rend possible l'échange de paquets, même si les nœuds ont un faible Duty-cycle (i.e., ils sont en mode sommeil la plupart du temps) [70,83].
- **Protocoles de routage à économie d'énergie** : ces protocoles de routage déterminent les chemins jusqu'au *Sink* en considérant plusieurs métriques relatives à la consommation d'énergie tels que le coût, la fiabilité des liens radio (les erreurs de transmission entrainent

des retransmissions qui consomment beaucoup d'énergie), le débit de transmission sur les liens radios et le niveau de charge des batteries des nœuds. La solution la plus triviale est de choisir un chemin avec un minimum de sauts.
Ces protocoles vont être vus en détail dans la section 2.4.3.

- **Contrôle de la topologie** : le contrôle de la topologie consiste à éliminer du réseau les nœuds inutiles (par la mise en sommeil des nœuds redondants) et les liens inutiles (par l'ajustement de la puissance de l'émetteur radio, et donc de la portée de communication) pour diminuer la dépense d'énergie dans le réseau. Il s'agit donc de faire une réduction de la topologie initiale du réseau tout en préservant la couverture de la zone d'intérêt et la connectivité du réseau. Cette topologie réduite doit être maintenue à jour de temps en temps car le réseau a besoin d'évoluer au fur et `a mesure que des nœuds actifs arrivent à épuisement [7].
- **Techniques de compression des données** *:* les techniques de compression des données dans les RCSFs visent à réduire la quantité de données à traiter et à transmettre. Ces techniques peuvent être classées, selon les étapes de traitement des données, en trois catégories : acquisition des données, traitement et transmission.
  Les algorithmes suivants proposent des solutions à ce type de problème. Nous pouvons citer le F & G (Flooding and Gossiping [23 , 84]), DD (Directed Diffusion [28, 85]), SPIN (Sensor Protocol for Information via Negociation [86]), GBR (Gradient Based Routing[87]), RR (Rumor Routing [88, 69]), CADR (Constrained Anisotropic Diffusion Routing[81]), ACQUIRE (Active Query forwarding in Sensor Networks[89]).

2.4.1 Protocoles MAC à conservation d'énergie sans mobilité

Les protocoles en adéquation avec les contraintes des RCSFs et plus précisément la contrainte d'économie d'énergie se fait au niveau des couches liaison de données et réseau car la minimisation de la consommation d'énergie pendant la communication est étroitement liée aux protocoles développés pour ces deux couches. Une forte consommation d'énergie est observée dans la sous couche MAC et dans la couche réseau contrairement aux autres couches de la pile protocolaire (couche physique, couche transport et la couche application) où la consommation d'énergie est négligeable. Nous résumons ici quelques motivations d'un tel choix.
En premier lieu, nous avons choisi de nous intéresser aux protocoles MAC car ils régissent l'activité du circuit radio et l'établissement d'une communication entre émetteurs et récepteurs, les protocoles de contrôle d'accès au médium permettent de conjuguer les contraintes de consommation d'énergie et la durée de vie des nœuds capteurs.
La plupart des protocoles MAC pour les RCSFs ont été proposés par les chercheurs et une étude [90] datant de 2009a recensé plus de 70 protocoles MAC dans le but d'améliorer l'efficacité énergétique en consommant moins d'énergie pendant la transmission des paquets entre les nœuds. Ces protocoles ont également comme objectifs la réduction du retard et la perte de paquets. Un protocole MAC décide, quand les nœuds concurrents peuvent accéder au support partagé, et tente de s'assurer que deux nœuds n'interfèrent pas entre eux. Pour les applications de surveillance continue, les nœuds génèrent le trafic périodiquement, tandis que dans les applications événementielles, le trafic tend à être sporadique. La méthode la plus

courante de conservation d'énergie est d'utiliser le mécanisme Duty-cycle périodique; en désactivant la radio quand il n'y a aucun échange de paquets. Les protocoles MAC existants peuvent être classés en protocoles synchrones, asynchrones (à préambule)[1,91]ne supportant pas la mobilité et en protocoles synchrones, asynchrones et hybrides supportant la mobilité comme nous pouvons le voir sur la figure 2.12.

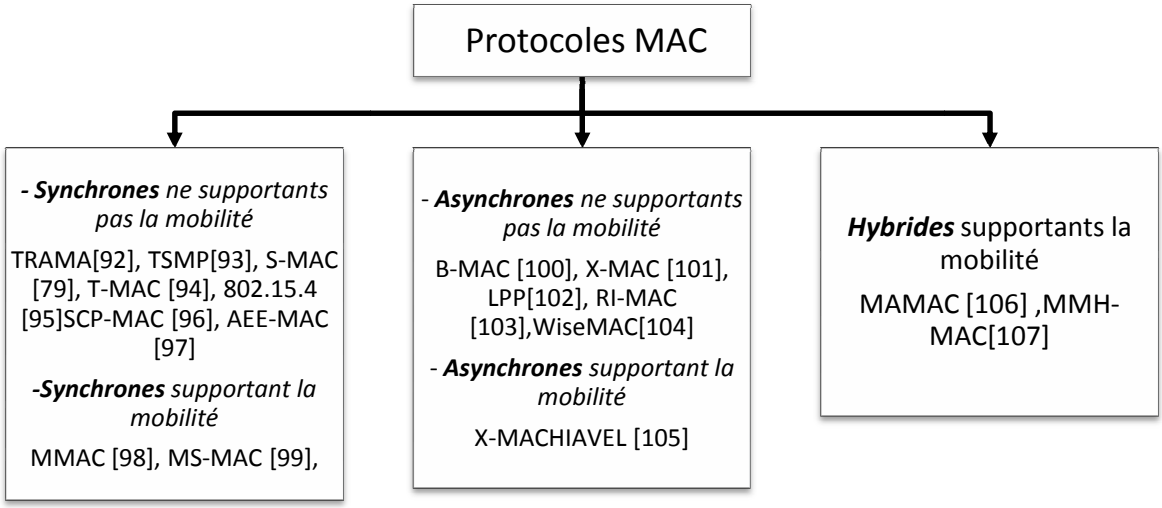


Figure 2.12 – Protocoles MAC

- ***Protocoles MAC Synchrones*** : Généralement, les protocoles MAC à conservation d'énergie synchrones ont besoin d'un mécanisme pour établir un calendrier de régulation non contradictoire qui aide à décider « Quelle ressource utiliser et à quel moment l'utiliser ». Le Schedule peut être fixe ou calculé à la demande. Le temps de synchronisation est nécessaire et ce temps est divisé en slot. Dans les protocoles MAC à conservation d'énergie ne supportant pas la mobilité comme dans S-MAC (*Sensor MAC* [79]), T-MAC(Timeout MAC[94]) qui est une amélioration de S-MAC, SCP-MAC [96], AEE-MAC [97], TRAMA [108], TSMP[109]il ya le sommeil périodique, soutenu par quelques mécanismes pour synchroniser le réveil des nœuds, pour assurer la connexion entre l'émetteur et le récepteur.
- ***Protocoles MAC Asynchrones:*** les mécanismes asynchrones sont généralement fondés sur les préambules, longs ou courts, qui précèdent les transmissions. Les préambules constituent l'Overhead, et donc le prix à payer pour ne pas compter sur la synchronisation d'horloge est une diminution de l'efficacité dans la transmission de données, soit plus de bits sont transmis pour envoyer la même quantité de données. Plusieurs propositions pour les RCSFs supportant la mobilité au niveau de la couche MAC ont été construites sur l'adaptation des solutions existantes de couche MAC conçue pour les réseaux stationnaires.

  Parmi les protocoles asynchrones les plus connus basés sur l'approche préambule: le protocole X-MAC[101],B-MAC(Berkeley MAC) [100] qui est une couche MAC basée sur le standard IEEE 802.15.4 (Wi-Fi) et Wise MAC, LPP ainsi que le protocole RI-MAC.

- ***Protocoles MAC hybrides :*** l'idée de base des protocoles MAC hybrides est de changer le comportement du protocole entre TDMA et CSMA en fonction du niveau de contention n'est pas une idée nouvelle. Concernant les RCSFs, le protocole Z-MAC (Zebra-MAC)[110] est l'un des protocoles les plus intéressants. Afin de définir le schéma principal du contrôle de transmission, Z-MAC commence par une phase préliminaire de configuration. Chaque nœud construit une liste de voisins à deux sauts par le biais du processus de découverte de voisins. Puis, un algorithme distribué d'attribution des slots est appliqué pour faire en sorte que deux nœuds dans un voisinage à deux sauts ne soient pas affectés au même slot. Par conséquent, on est assuré qu'une transmission d'un nœud avec un de ses voisins à un saut n'interfère pas avec les transmissions de ses voisins à deux sauts. Le protocole Z-MAC permet à chaque nœud de maintenir son propre ordonnancement qui dépend du nombre de voisins et évite tout conflit avec ses voisins de contention. Les protocoles hybrides tentent de combiner les points forts des protocoles MAC synchrones et asynchrones tout en compensant leurs faiblesses. Toutefois, ces techniques semblent être complexes pour être réalisables dans un déploiement d'un grand nombre de nœuds [70].

2.4.2 Protocoles MAC à conservation d'énergie avec mobilité

De nos jours, les RCSFs ne sont plus constitués uniquement de capteurs fixes mais également de capteurs placés sur des éléments mobiles. Ces nœuds mobiles peuvent être utilisés dans des applications telles que la surveillance animale. Au niveau MAC, cette mobilité peut se traduire par des difficultés à émettre des données. En effet, le nœud mobile doit s'insérer dans les communications. Cette insertion peut être particulièrement complexe surtout si les communications entre les nœuds fixes sont organisées de manière précise [1].

- ***Protocoles MAC Synchrones*** **:** ces protocoles basés sur des slots vont organiser les communications de chaque nœud de manière précise.
  Cette forte synchronisation permet d'établir un schéma de communication sans collisions. L'insertion d'un nœud mobile dans les communications impose de renégocier ce programme établi, ce qui peut s'avérer coûteux et complexe. Le protocole M-MAC propose de redistribuer à intervalle régulier les slots et d'adapter la longueur des cycles pour permettre aux nœuds mobiles de s'y insérer. La fréquence de redistribution va dépendre du nombre de nœuds mobiles et de leur vitesse.
  Un autre exemple de protocole synchronisé, le protocole MS-MAC propose d'augmenter la fréquence des périodes de synchronisation. La fréquence des périodes est adaptée en fonction de la vitesse des nœuds mobiles obtenue à partir du RSSI des messages envoyés. Cependant, le RSSI utilisé pour détecter la mobilité d'un nœud n'est pas un indicateur idéal.
- ***Protocoles MAC Asynchrones*** **:** le protocole X-MACHIAVEL est un protocole MAC basé sur X-MAC dont l'objectif est d'optimiser l'accès au médium des nœuds mobiles. Le protocole X-MACHIAVEL prioritise les communications des nœuds mobiles et leur permet donc de subtiliser le médium radio à un nœud fixe. En pratique, si un nœud mobile souhaite transmettre un paquet de données, mais que le médium est déjà occupé car un préambule est en cours d'émission, le nœud mobile peut transmettre un acquittement

spécial. Cet acquittement annonce qu'il subtilise le canal et va transmettre directement son paquet de données au nœud émettant le préambule. De plus, pour accélérer la durée de transmission d'un paquet d'un nœud mobile, tout nœud fixe présent dans le voisinage est autorisé à acquitter le préambule et à devenir la destination du paquet de données. Ce comportement va permettre aux nœuds mobiles d'obtenir plus rapidement l'accès au canal et réduire ainsi les pertes dues à leur éloignement du destinataire.

- ***Protocoles MAC Hybrides*:** dans une tentative de profiter des aspects positifs des solutions MAC synchrones et asynchrones, des solutions hybrides ont été développées. Ce mixage de nœuds synchronisés dans le réseau avec des nœuds non synchronisés aux limites du réseau. MAMAC et MMH-MAC, que nous décrirons dans ce qui suit, sont deux propositions de synchronisation hybride pertinentes avec support de la mobilité.
  Le *protocole MAMAC* (Mobile Adaptive MAC) propose un simple mécanisme où les nœuds sont soit fixes soit mobiles, et n'utilise pas d'horloges synchronisées. Au lieu de cela, chaque nœud se réveille en un point aléatoire dans le temps et envoie une balise d'accusé de réception. Lorsqu'un nœud désire transmettre, il commence l'écoute de l'environnement jusqu'à ce qu'il reçoive la balise d'accusé de réception du nœud destinataire, au cours du quelle nœud commence la transmission [111].

  Aussi une solution hybride, le *protocole MMH-MAC* (Mobile Multimode Hybrid MAC protocol) vise à maintenir la faible consommation d'énergie des solutions asynchrones et le haut débit des solutions synchrones, même en présence de nœuds mobiles. Pour cela, il implémente à la fois le mode synchrone pour les nœuds fixes et le mode asynchrone pour les nœuds mobiles. Pour plus de détails sur ces deux modes se référer à l'article [111].

2.4.3 Routage dans les RCSFs

En second lieu, nous nous sommes intéressés aux protocoles de routage car ils prennent en charge l'établissement et le maintien d'un ensemble de chemins garantissant les exigences de la consommation d'énergie et permettant d'assurer les performances à long terme du réseau. C'est pour ces raisons, que nous allons voir dans la prochaine section les différents protocoles de routage.
La principale tâche des nœuds capteurs dans chaque application est de surveiller la zone cible et de transmettre leurs informations collectées vers le nœud *Sink* afin d'effectuer des traitements spécifiques en fonction des besoins de l'application. Les ressources limitées des nœuds capteurs et le manque de fiabilité des liens sans fil, en combinaison avec les différentes exigences de performance de différentes applications imposent de nombreux défis dans la conception de protocoles de routage efficaces pour les RCSFs. Dans ce contexte, les chercheurs ont proposé de nombreux protocoles de routage au niveau de la couche réseau de la pile protocolaire du nœud capteur, pour améliorer les exigences de performance des applications. La plupart des protocoles de routage existants dans les RCSFs sont conçus sur la base de la stratégie du routage mono-chemin. Ces protocoles de routage peuvent être classés selon différents paramètres [4].

Classiquement, selon le processus de création des chemins, trois grandes familles de protocoles peuvent être distinguées : les protocoles proactifs, réactifs et hybrides qui sont présentés dans la figure 2.13.

- ***Protocoles proactifs*** : chaque nœud maintient une table de routage contenant des chemins à tous les autres nœuds du réseau. ainsi, les chemins sont découverts et stockés même s'ils ne seront pas utilisés, cela entraine la diffusion de nombreux messages de contrôle, qui engendre du trafic dans le réseau, réduisant ainsi la bande passante disponible pour l'envoi des données. de plus, cette émission permanente de messages entraine une consommation d'énergie plus importante au niveau des nœuds du réseau, ce qui rend l'utilisation de ce type de protocole dans les RCSFs problématique. Les protocoles proactifs peuvent être inefficaces pour les réseaux dynamiques de grande taille. Des exemples de ce type de protocol: DSDV(Destination Sequence Distance Vector) [112],OLSR(Optimized Link State Routing Protocol)[113],LEACH (Low Energy Adaptive Clustering Hierarchy) [114] ,PEGASIS (Power Efficient Gathering in Sensor Information System) [115] etc.

- ***Protocoles réactifs*** : appelés aussi protocoles de routage à la demande, les chemins sont donc uniquement cherchés à la demande lorsqu'un nœud a besoin d'envoyer un message vers un autre nœud destinataire. le processus de découverte de chemins nécessite la transmission des requêtes de demande de chemins et l'attente d'une réponse fournissant un chemin vers la destination. A cause du temps de latence induit par ce processus de découverte, cette approche n'est pas adaptée pour les applications nécessitant une disponibilité immédiate de chemins. Les protocoles de routage AODV(Ad hoc On-demand Distance Vector) [116],DSR (Dynamic Source Routing) [117],SPIN[86],et DD[85] font partie de cette classe de protocoles.

- ***Protocoles hybrides*** : ce sont des protocoles basés sur les approches hybrides qui visent à fournir une solution optimale en combinant les avantages des deux approches proactives et réactives. Ils utilisent un protocole proactif pour connaitre le proche voisin, ainsi ils disposent de chemins immédiatement.au-delà de la zone du voisinage, le protocole hybride fait appel à un protocole réactif pour chercher des chemins. Les protocoles ZRP (Zone Routing Protocol)[118] et APTEEN (Adaptive-Periodic Threshold-Sensitive Energy Efficient Sensor Network Protocol) [119] sont des exemples de protocoles hybrides.

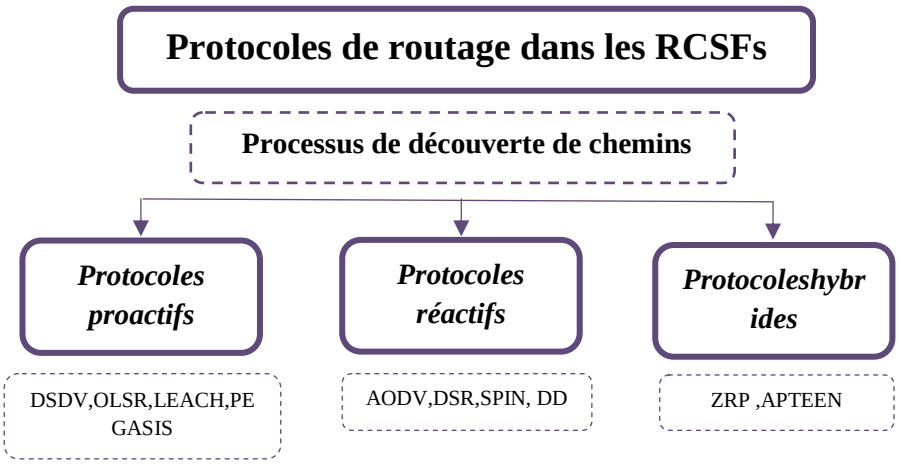


Figure 2.13 – Protocoles de routage selon le processus de découverte de chemins

Les auteurs de [120] ont proposé une autre classification de protocoles de routage selon deux points de vue différents : structure du réseau et stratégie de routage du protocole selon le point de vue structure de réseau, nous distinguons les protocoles de routage plat, hiérarchique et géographique [4]. La figure 2.14 résume les principaux protocoles de routage selon la structure du réseau.

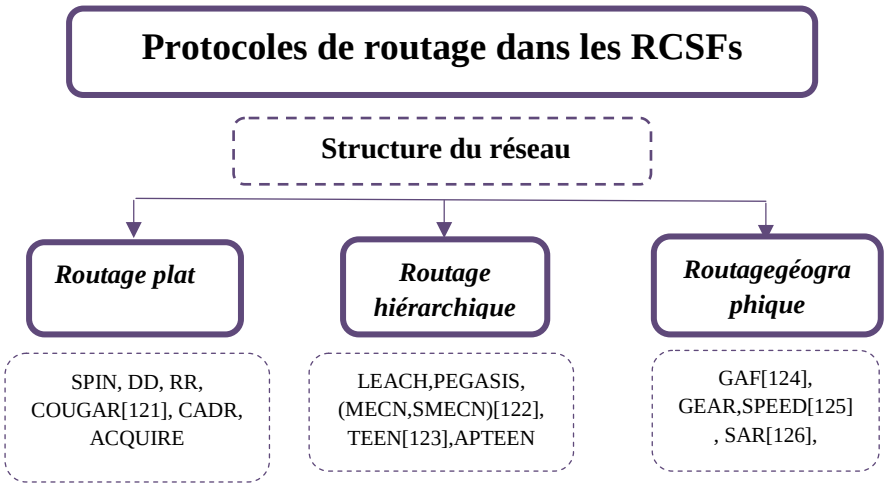


Figure 2.14 – Protocoles de routage selon la structure du réseau

- ***Routage plat* :** dans le routage plat, chaque nœud joue typiquement le même rôle et les nœuds capteurs collaborent pour accomplir la tâche globale du réseau. En raison du nombre important des nœuds capteurs, il n'est pas faisable d'assigner un identifiant global pour chaque nœud. Cette considération a mené au routage « Data-Centric », où la station de base envoie des requêtes à certaines régions du réseau et attend des retours de données à partir des nœuds capteurs situés dans ces régions. Puisque des données sont demandées par le biais des requêtes, la désignation des attributs est nécessaire pour indiquer les propriétés de ces données. Des premiers travaux sur le routage Data-Centric, tels que les protocoles

SPIN et DD, ont enregistré une économie d'énergie grâce à la négociation entre les nœuds du réseau et l'élimination des données redondantes. La figure 2.15 illustre le protocole plat dans les RCSFs.

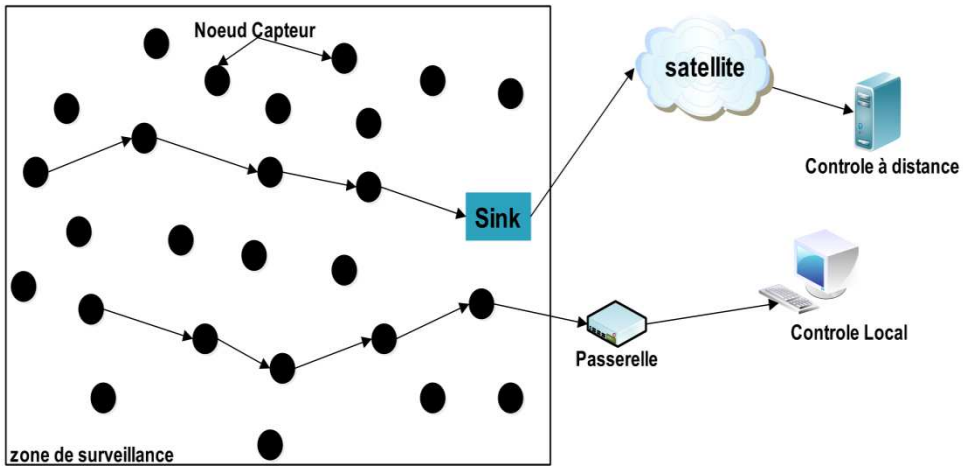


Figure 2.15 – Routage plat dans les RCSFs

- ***Routage hiérarchique:*** les méthodes de routage hiérarchique ont des avantages spéciaux liés au passage à l'échelle et à l'efficacité dans la communication. Par exemple, elles sont utilisées pour exécuter un routage avec économie d'énergie dans les RCSFs. Dans une architecture hiérarchique, des nœuds à grande énergie peuvent être employés pour traiter et envoyer l'information, alors que des nœuds à énergie réduite peuvent assurer la capture à proximité de la cible. La création des clusters et l'assignation des tâches spéciales aux têtes de clusters peuvent considérablement renforcer le passage à l'échelle, l'augmentation de la durée de vie et l'efficacité énergétique du système global. Le routage hiérarchique est une manière efficace de réduire la consommation énergétique dans un cluster en exécutant l'agrégation et la fusion de données afin de diminuer le nombre de messages transmis à la station de base [127].

  Parmi ces protocoles nous citons :le protocole LEACH que nous verrons plus en détail dans le chapitre 6 de cette thèse), PEGASIS, TEEN (Threshold-Sensitive Energy Efficient Sensor Network Protocol),APTEEN…

Nous pouvons voir sur la figure 2.16 le routage hiérarchique dans les RCSFs.

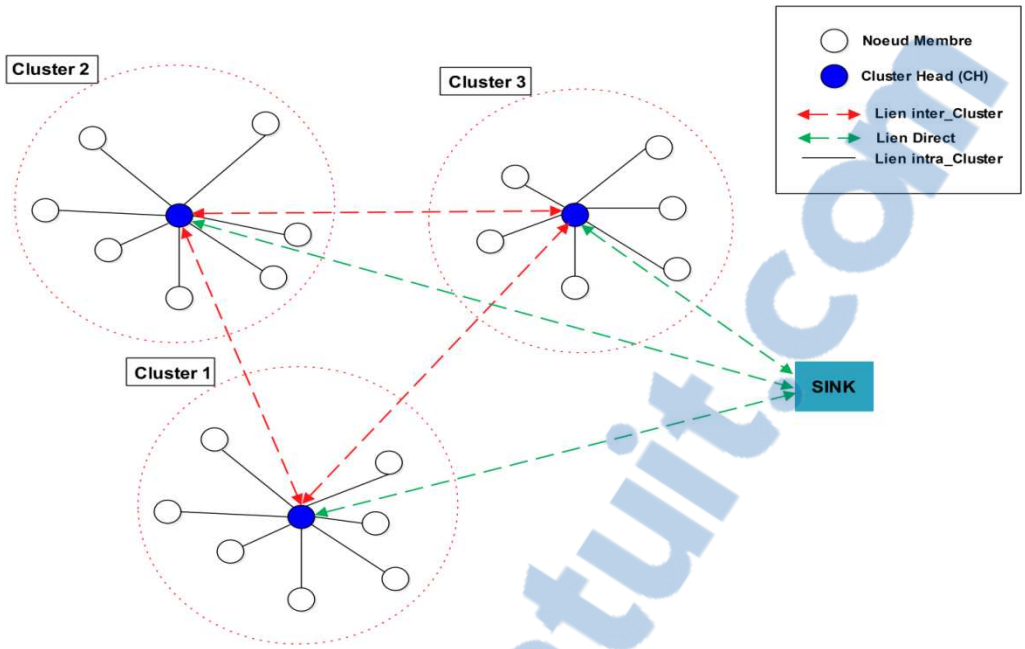


Figure 2.16 – Routage hiérarchique dans les RCSFs

Remarque : Nous avons étudié ces deux routages en détail car nous les avons utilisés dans notre travail.

- ***Routage géographique*****:** à l'inverse des approches traditionnelles, le routage géographique présente des propriétés intéressantes pour les réseaux maillés sans fil spontanés : il n'exige aucune information sur la topologie globale puisqu'un nœud choisit le prochain saut parmi ses voisins sur la base de la localisation de la destination. En conséquence, le mécanisme de routage supporte le passage à l'échelle, parce qu'il utilise seulement des décisions locales. Le routage géographique est simple, parce qu'il n'exige pas de tables de routage de sorte qu'il n'y ait aucune surcharge de contrôle pour leur création et maintenance. La jointure du réseau est également simple, parce qu'un nouveau nœud a besoin seulement d'une adresse basée sur sa localisation géographique. De telles adresses peuvent être obtenues à partir d'un dispositif dédié, par exemple GPS, ou par l'application de mécanismes d'auto-localisation qui sont très couteux pour des nœuds capteurs à ressources limitées. Nous citons comme protocole de routage géographique GEAR (Geografic and Energy-Aware Routing), GAF(Geografic Adaptive Fidelity), GPSR (Greedy Parameter Stateless Routing). La variante la plus familière du routage géographique est la transmission en mode glouton (Greedy) dans lequel un nœud transmet le paquet au voisin le plus proche de la destination, pour plus de détail sur ce mode voir [128].

Du point de vue stratégie de routage utilisée, tous les protocoles de routage existants dans les catégories mentionnées ci-dessous peuvent être également classés en protocoles de routage basés sur la négociation, les requêtes, la qualité de service, la cohérence et les chemins multiples. La figure 2.17 présente justement cette classification.

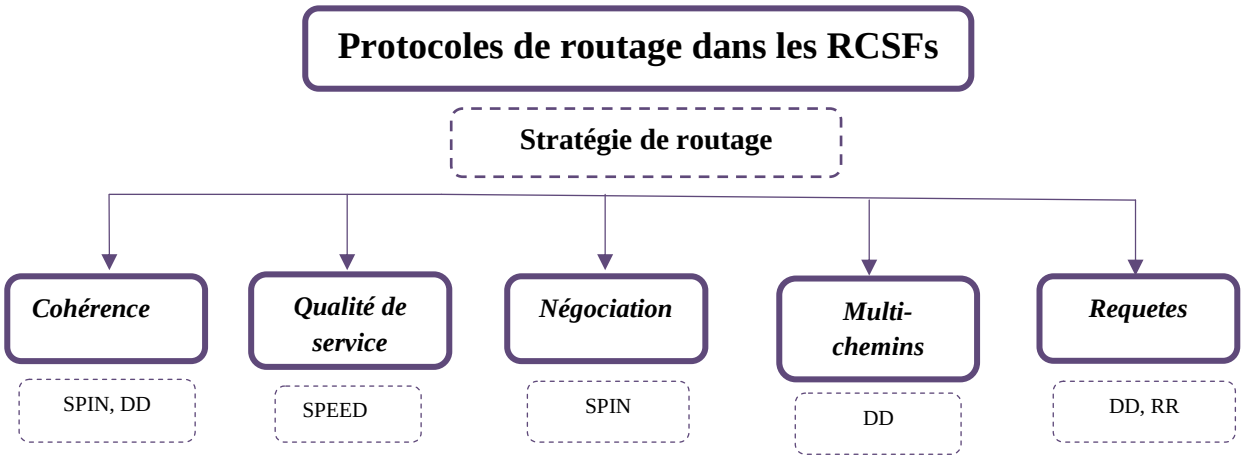


Figure 2.17 – Protocoles de routage selon la stratégie du réseau

2.5 Réseaux de capteurs sans fil à éléments mobiles (RCSFs-EM)

Un RCSF-EM est à l'origine un RCSF-S qui est constitué de nœuds, d'une zone d'intérêt, et d'un observateur qui peuvent être mobiles. Si les nœuds se déplacent, cela implique un changement de la zone d'intérêt qui devient alors « mobile ». Comme la zone change, l'observateur sera obligé de reconstruire le chemin d'accès et de briser l'ancien. Dans ce cas l'observateur a intérêt à construire plusieurs chemins entre lui et les nœuds et à choisir le plus bénéficiaire [19]. L'objectif des RCSFs-EM est de collecter plus d'informations sur un environnement en utilisant moins de nœuds capteurs, et de permettre au réseau d'organiser lui-même ses nœuds, de mieux assurer la connectivité et la couverture au sein du réseau et surtout de prolonger la durée de vie du réseau le plus longtemps possible pendant le processus de collecte de données.

En outre, il devient possible de déplacer ses capteurs dynamiquement selon les changements environnementaux, ce qui le rend adaptable à l'évolution de son environnement.

Ce type de réseau est composé des mêmes éléments que ceux d'un RCSF-S en plus d'une autre sorte de nœuds appelés *« **nœuds supports spéciaux** »*que nous allons voir en détail dans le chapitre suivant.

2.5.1 Différentes formes de mobilité

Nous pouvons distinguer trois formes de mobilité dans les RCSFs[7,129]:

2.5.1.1 Mobilité des nœuds capteurs

La mobilité des capteurs ne concerne qu'un petit nombre d'applications, par exemple des applications militaires (les capteurs sont attachés à des soldats), des applications de surveillance des animaux d'élevage, ou des oiseaux migrateurs (les capteurs sont portés par les animaux) ou encore des surveillances des milieux marins (les capteurs suivent le mouvement du courant).

Face à la mobilité des nœuds, le réseau doit se réorganiser assez fréquemment pour pouvoir fonctionner correctement. Le problème de la conservation de l'énergie des nœuds devient

aussi plus ardu car les techniques de géolocalisation coûtent de l'énergie et elles vont devoir être sollicitées plus fréquemment.

2.5.1.2 Mobilité de l'évènement

Ce type de mobilité existe essentiellement dans les applications de suivi de cibles. Dans ce type d'application, il est important que l'évènement observé soit couvert par un nombre suffisant de nœuds. Par conséquent, les nœuds vont se réveiller autour de l'objet, pour le surveiller avec un taux d'activité élevé, et le nœud capteur se mettra en mode Sleep lorsque la cible s'éloigne. Pendant que la source d'évènement se déplace à travers le réseau, elle est accompagnée d'un secteur d'activité dans le réseau qui la suit.

2.5.1.3 Mobilité du nœud Sink

C'est un cas spécial de mobilité où seul le point de collecte est mobile. Etant donné que le trafic converge vers le *Sink*, les nœuds proches du *Sink* transmettent plus de paquets que les autres nœuds et donc épuisent plus vite leurs batteries *(Effet Funneling)* [130]. Si ce *Sink* est mobile, les nœuds-capteurs peuvent attendre son passage pour lui envoyer leurs données directement ou sur un nombre de sauts réduit.
Cela va réduire considérablement le trafic et la consommation d'énergie dans le réseau.
Le *Sink* est un point de collecte statique, garant de l'exécution des opérations nécessaires de formation et de maintenance du réseau, est habilité à communiquer avec le centre de traitement via Internet ou par satellite pour l'analyse des informations remontées par les différents capteurs et la prise de décision ultérieure. Par conséquent, toute défectuosité de ce point névralgique, potentiellement induite par un mauvais positionnement, des changements dans l'environnement après le déploiement ou des défauts de connectivité, conduit à l'isolement du réseau, donc à son annihilation [5].
C'est ainsi que dans la plupart des contributions, certaines particularités exprimées le plus souvent sous forme d'hypothèses sont octroyées aux *Sinks*:

- ***Puissance de communication*** : le *Sink* est doté d'une puissance de transmission largement supérieure à celle des autres nœuds du réseau. D'aucuns stipulent même que son rayon de communication pourrait atteindre l'ensemble des capteurs répartis dans la zone d'intérêt. Ce qui est, en notre sens, irréaliste. La génération d'un système de coordonnées dynamiques où chaque nœud est affecté à un cluster, déduit des m émissions omnidirectionnelles (les couronnes) et n directionnelles (les secteurs) du puits, en est une parfaite illustration.
- ***Autonomie énergétique*** : la contrainte majeure dans les RCSFs est la capacité énergétique assez limitée des capteurs due en grande partie à leur taille. Le *Sink* demeure l'unique équipement du réseau à disposer suffisamment d'énergie afin de mener à bien les responsabilités qui lui incombent : passerelle avec le monde extérieur, initiateur des tâches de formation et de maintenance du réseau, etc. Toutefois, dans un contexte de difficulté ou de dangerosité d'accès de la zone de surveillance, le recours aux sources disponibles de l'environnement du *Sink* (vent, énergie solaire) pour son ravitaillement est envisageable.
- ***Capacité mémoire*** : un réseau de capteurs consiste en un déploiement de plusieurs milliers de capteurs dans une région d'intérêt : c'est la notion d'échelle. Cette forte densité du

réseau se traduit par une forte transmission de données vers le *Sink*. Ce phénomène est d'autant plus délicat si la nature des informations véhiculées concerne un contenu multimédia (image, audio, vidéo) compte tenu du volume de données mis en jeu. Par conséquent, contrairement aux nœuds capteurs dotés de capacités de mémoire et de calcul limitées, le *Sink* doit posséder une large mémoire pour le stockage des informations reçues.

2.5.2 Raisons de la mobilité dans les RCSFs

Les architectures de RCSFs traditionnels reposent sur l'hypothèse que le réseau est dense, de telle sorte que deux nœuds peuvent communiquer entre eux à travers une communication multi-sauts. Par conséquent, dans la plupart des cas les capteurs sont considérés comme statiques et la mobilité n'est pas considérée comme une option. Plus récemment, de manière similaire, la tendance de la recherche dans les réseaux mobiles Ad-hoc (*MANET*) des réseaux [131] et des réseaux tolérants au retard (*DTNs*) [132], la mobilité a également été introduite dans les RCSFs [111,133]. En fait, la mobilité dans les RCSFs est utile pour plusieurs raisons [134,135] comme c'est décrit dans ce qui suit :

- ***Connectivité*** : comme les nœuds sont mobiles, une architecture dense de RCSFs peut ne pas être une exigence. En fait, les éléments mobiles peuvent combler les régions isolées, de sorte que la contrainte de connectivité du réseau soit assouplie, également en termes de redéploiement des nœuds. Ainsi une architecture de RCSF dispersée devient une option réaliste.
- ***Coût*** : en déployant moins de nœuds dans un RCSFs-EM, le coût du réseau se voit réduit, bien que l'ajout de fonctionnalités de la mobilité aux nœuds peut être coûteux, dans de nombreux cas il est possible d'exploiter des éléments mobiles qui sont déjà présents dans la région (par exemple : trains, bus, navettes, ou les voitures), et leurs attacher des capteurs.
- ***Fiabilité*** : depuis que les RCSFs-S traditionnels sont denses et le paradigme de communication est souvent à multi-sauts (*Ad-hoc*), la fiabilité est compromise par les interférences et les collisions. En outre, la perte de messages augmente avec un nombre de sauts élevé. Les éléments mobiles visitent les nœuds dans le réseau et récupèrent les données directement via des transmissions à un seul saut. Ceci réduit non seulement les conflits et les collisions, mais aussi la perte de messages.
- ***Rendement énergétique*** : le modèle de trafic inhérent aux RCSFs est le convergecast, c'est-à-dire que les messages sont générés à partir des nœuds capteurs et sont collectés par le *Sink*. Par conséquent, les nœuds les plus proches du *Sink* sont surchargés par rapport aux autres, et donc épuisent leur énergie prématurément. Ce problème est connu sous le nom de*(Effet Funneling)*[130], car les voisins du *Sink* représentent le goulot d'étranglement du trafic. Les éléments mobiles peuvent aider à réduire *(Effet Funneling)* car ils peuvent visiter différentes régions du réseau et propager la consommation d'énergie de manière plus uniforme, même dans le cas d'une architecture de RCSF dense [136,137].

Cependant, la mobilité dans les RCSFs présente également des défis significatifs qui ne se posent pas dans les RCSFs-S. Ces défis sont décrits ici.

2.5.3 Challenges de la mobilité dans les RCSFs

L'usage d'un *Sink* mobile pour la collecte de données dans un RCSF présente plusieurs avantages mais n'est pas sans conséquence. En effet, la mobilité soulève plusieurs défis à relever pour une conception et une mise en œuvre efficace d'un schéma de collecte. Voici quelques défis spécifiques à l'introduction des *Sink* mobiles dans les RCSFs [11]:

- ***Détection et contact avec le Sink*** : étant donné que la communication n'est possible que lorsque les nœuds sont dans la phase de transmission, il est nécessaire de détecter la présence du *Sink* mobile de manière correcte et efficace. Cela est particulièrement vrai lorsque la durée de contact est courte. Une fois que le *Sink* mobile est détecté, la communication entre les capteurs stationnaires et le *Sink* mobile nécessite la présence de ce dernier dans leur portée radio. Cependant, cette détection du *Sink* mobile est fortement affectée par sa vitesse de déplacement et les cycles de réveil des nœuds du réseau. De plus, une vitesse de déplacement rapide du *Sink* conduit à de courtes liaisons entre ce dernier et son voisinage immédiat occasionnant potentiellement des pertes considérables. Ce mécanisme sera détaillé dans la Figure 3.4 du chapitre 3.

- ***Gestion de l'alimentation du courant de la mobilité*** : dans certains cas, il est possible d'exploiter la connaissance du schéma de mobilité afin d'optimiser la détection de l'élément mobile. En fait, si les temps de visite sont connus ou peuvent être prédits avec une certaine précision, les nœuds capteurs peuvent être éveillés seulement quand l'élément mobile est dans leur zone de transmission.

- ***Transfert fiable de données*** : comme les contacts disponibles sont rares et brefs, il est nécessaire de maximiser le nombre de messages transférés correctement au *Sink* d'une part, et d'autre part il faut également tenir compte de la mobilité pendant le transfert de données et l'échange de messages.
- ***Localisation du Sink*** : le but d'un RCSF est de remonter les informations collectées vers le *Sink*, cet objectif ne peut alors être atteint sans une localisation préalable de ce point focal. Contrairement à un réseau de capteurs à *Sink* statique, les nœuds n'ont à priori aucune connaissance de la position courante du *Sink* dans un contexte à puits mobile. Pour ce faire, à l'image des procédures de gestion de la mobilité (*Mobility Management*) dans les réseaux cellulaires où une mise à jour de localisation s'opère lorsqu'un terminal change de zone localisation, la position du *Sink* doit être mise à jour dans tout le réseau pour assurer la connectivité. Une fréquence régulière des mises à jour de la localisation du *Sink* dans le réseau pourrait réduire les délais de transmission car permettant l'ajustement des chemins en conséquence.

Cependant, elle génère un coût important en nombre de messages échangés, donc une consommation énergétique importante précipitant la déconnexion des éléments du réseau. D'un autre côté, des mises à jour peu fréquentes de la position courante du *Sink* conduiraient à des pertes de paquets considérables (paquets jetés) due à une progression importante du *Sink* depuis sa dernière position [5,11].

- ***Contrôle de la mobilité*** : lorsque le mouvement des éléments mobiles peut être contrôlé, il est primordial de définir une politique de visite au sein du réseau. A cette fin, le chemin, la vitesse ou le temps de séjour des nœuds mobiles doivent être définis afin d'améliorer et d'optimiser les performances du réseau. De là, on peut conclure que le contrôle de la mobilité dépend essentiellement des types d'éléments mobiles utilisés dans le réseau.

2.5.4 Mobilité au niveau de la pile protocolaire

Dans les RCSFs, la mobilité d'un équipement fait intervenir plusieurs niveaux de la pile de communication (modèle OSI en anglais *Open System Interconnection* [138]). Elle se gère tout d'abord au niveau 2, où un protocole d'accès au médium (*MAC*) doit permettre aux nœuds mobiles de transmettre leurs paquets sur le médium sans fil.
Plusieurs verrous scientifiques peuvent limiter les capacités de communication des nœuds mobiles au niveau MAC. En effet, un nœud mobile doit être capable de s'insérer dans les communications mais également de déterminer vers qui transmettre ses paquets. De nombreuses solutions facilitant l'insertion de nœuds mobiles ont été proposées dans la littérature [95,99,105] mais elles utilisent des informations provenant de protocoles de la couche réseau pour déterminer vers quel voisin transmettre les paquets. En raison du grand nombre de protocoles de routage existants [115], il est fort peu probable que celui utilisé parle nœud mobile soit identique à celui du réseau dans lequel il se trouve. En conséquence, le nœud mobile ne peut obtenir un prochain saut par des moyens conventionnels.

2.6 Stratégies de mobilité du Sink

Dans les sections précédentes, nous avons extrait des caractéristiques spécifiques de la mobilité, dans ce paragraphe nous allons parler des différents types de mobilité qui peuvent avoir un impact significatif sur les phases de collecte de données. Le chapitre suivant sera dédié au processus de collecte de données dans un RCSFs-EM. L'aspect de la mobilité qui a un impact encore plus significatif sur le processus de collecte des données est la « contrôlabilité », selon que le mouvement de l'élément mobile soit autonome ou pas.

2.6.1 Mobilité non contrôlée (non supervisée)

Il existe deux principaux modèles de mobilité non-contrôlée: déterministes et aléatoires que nous pouvons voir sur la figure 2.18.

- ***Modèle de mobilité déterministe :*** est caractérisé par la régularité des contacts de l'élément mobile, qui entre dans la zone de communication des nœuds capteurs à un temps spécifique (et habituellement périodiques). Cela peut se produire lorsque l'élément mobile est placé sur une navette pour les transports en commun, comme il est montré dans [139].
- ***Modèle de mobilité aléatoire :*** se caractérise par des contacts qui ont lieu régulièrement ou pas, mais avec une distribution probabiliste. Par exemple, les lois poissonnière d'un élément mobile ont été étudiées dans [140], tandis que la mobilité de la direction aléatoire de l'élément mobile a été prise en compte dans [141]. En général, un nœud devrait effectuer une découverte en continu, de sorte qu'il puisse augmenter les chances de détecter des contacts.

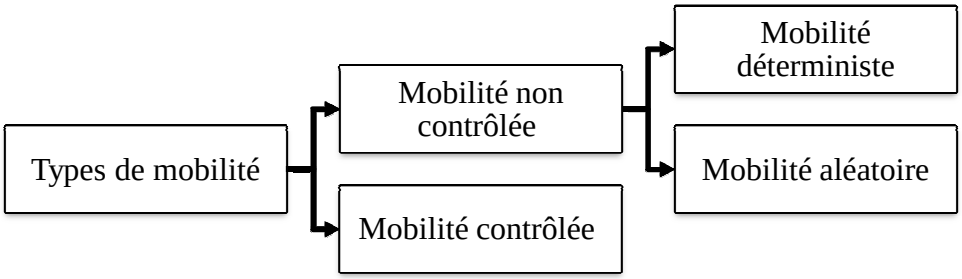


Figure 2.18– Types de mobilité

2.6.2 Mobilité contrôlée (supervisée)

Cette mobilité contrôlée exploite les nœuds qui peuvent changer activement leur emplacement, parce qu'ils peuvent contrôler leur trajectoire et leur vitesse. Par conséquent, le mouvement devient un facteur supplémentaire qui peut être exploité efficacement pour la conception de protocoles de collecte de données spécifiques au RCSFs – EMs. Il convient de noter que la mobilité contrôlée peut offrir des solutions à certains problèmes liés à une collecte de données moins pertinente. Par exemple, le problème de la découverte peut être quelque peu simplifié puisque les éléments mobiles peuvent être chargés de visiter les nœuds à des moments précis. En outre, la durée de contact est également moins problématique puisque les éléments mobiles peuvent s'arrêter au niveau des nœuds jusqu'à ce qu'ils collectent toutes les données mémorisées [140,141].

2.7 Gestion de la mobilité du Sink

La mobilité du *Sink* introduit une problématique liée à la gestion des communications entre les nœuds stationnaires du réseau et le *Sink* mobile. Considérons un scénario où le *Sink* se déplace aléatoirement à travers une zone géographique en émettant périodiquement une requête, afin de collecter les données générées par les capteurs avoisinants. Le nœud recevant la requête du *Sink* est appelé résolveur de requêtes (*query-resolver*), et a pour tâche d'acheminer cette requête vers les nœuds sources possédant les données correspondantes. À la réception des données, si le *Sink* n'est plus à portée de communication du résolveur de requêtes, la communication ne pourra probablement pas avoir lieu. Dans ce cas, de nouveaux mécanismes de communication doivent être mis en place afin d'assurer les communications entre les entités mobiles et le reste du réseau.

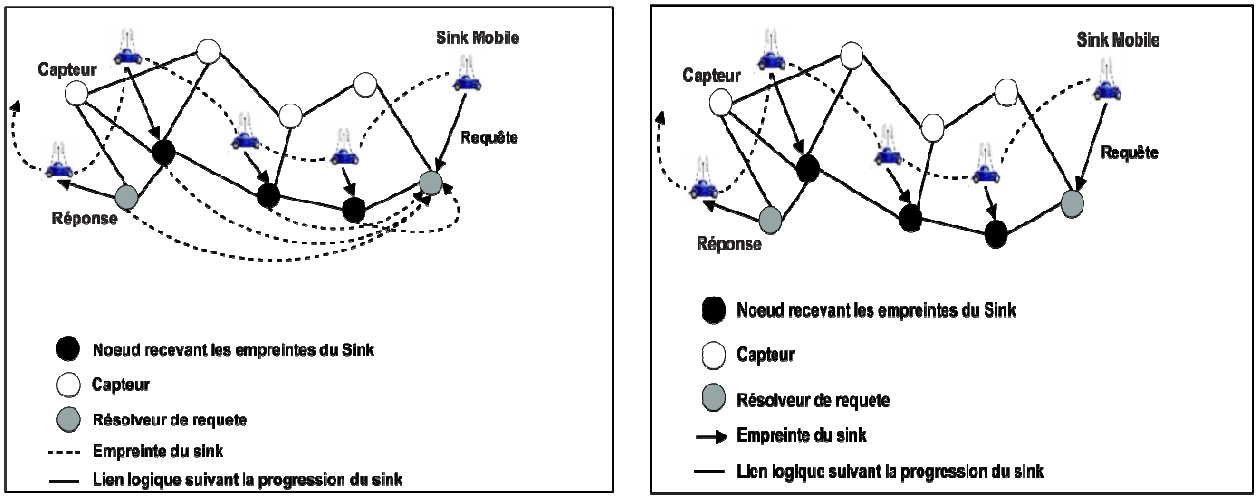


Figure 2.19–Gestion des communications entre Sink mobile et capteurs stationnaires.

Il existe dans la littérature deux principaux mécanismes permettant de gérer la mobilité d'entités communicantes [142]. Le premier mécanisme requiert de l'entité mobile de constamment mettre à jour son point de contact actuel auprès du résolveur de requêtes. Cela permet de construire un lien de communication virtuel vers l'ancien résolveur. Ce mécanisme est illustré sur la figure 2.19-(a). Étant donné que les mises à jour doivent traverser tout le réseau vers le résolveur, cette approche augmente la charge du trafic réseau.

Afin de remédier à ce problème, une deuxième approche a été proposée où l'entité mobile place régulièrement son empreinte ou Footprint afin de créer et maintenir un lien de communication logique vers le résolveur de requêtes. Toutes les mises à jour sont donc effectuées localement minimisant ainsi le trafic de contrôle. Cette méthode, qui est connue sous le nom de progressive Footprint chaining[143], est représentée sur la figure 2.19-(b).

2.7.1 Avantages des Sinks mobiles

Le *Sink* statique dans un RCSF traditionnel, demeure l'unique entité du réseau immunisée aux contraintes de ressources (capacités en mémoire, traitement, communication, bande passante, énergie, etc.). La mobilité du *Sink*, consistant à son déplacement périodique dans la zone d'intérêt pour la collecte de données, peut être réalisée par la fixation du *Sink* à une entité mobile telle qu'un humain, un animal, un véhicule ou un robot. Les auteurs de [12] résument comme suit les avantages potentiels expliquant l'intérêt d'un tel concept par rapport aux *Sinks* statiques dans les RCSFs:

- ***Prolongation de la durée de vie du réseau*** : la mobilité du *Sink* s'avère une très bonne alternative face à l'engorgement du voisinage immédiat du *Sink* occasionnant son isolement éventuel. L'autosuffisance énergétique demeure la contrainte majeure dans les RCSFs.

Le mode de communication plusieurs à un (many-to-one) à travers une architecture multi-sauts ainsi que la dissémination des données dans un RCSF stationnaire conduisent au drainage des ressources des nœuds, notamment leur énergie. Par conséquent, le recours au *Sink* mobile conduit également à la prolongation de la durée de vie globale du réseau par la répartition de la charge entre les nœuds et le rapprochement de plus près des sources réduisant considérablement le nombre de sauts à effectuer.

- ***Amélioration du débit et de la fiabilité des données*** : l'exploitation de puits mobiles peut également conduire à l'amélioration du débit et de la fiabilité des données véhiculées dans le réseau. En effet, le rapprochement d'au plus près des sources provoque non seulement la réduction de la distance de collecte(en nombre de sauts), mais aussi celle des erreurs de transmission et des risques de collision [13]. Cette minimisation des risques de retransmission est à l'origine de l'amélioration du débit. En outre, l'usage des *Sinks* mobiles augmente également la connectivité en permettant la collecte d'informations à partir des segments isolés du réseau [136].
- ***Sécurité*** : à l'inverse des *Sinks* statiques, la mobilité du *Sink* pose moins de problème de sécurité. En effet, dans un contexte de RCSFs à *Sinks* statiques, des attaques réseau peuvent cibler ce point névralgique facilement localisable.

2.8 Classification de la mobilité (objectif)

Dans la figure 2.20 l'objectif de la mobilité est recensé.

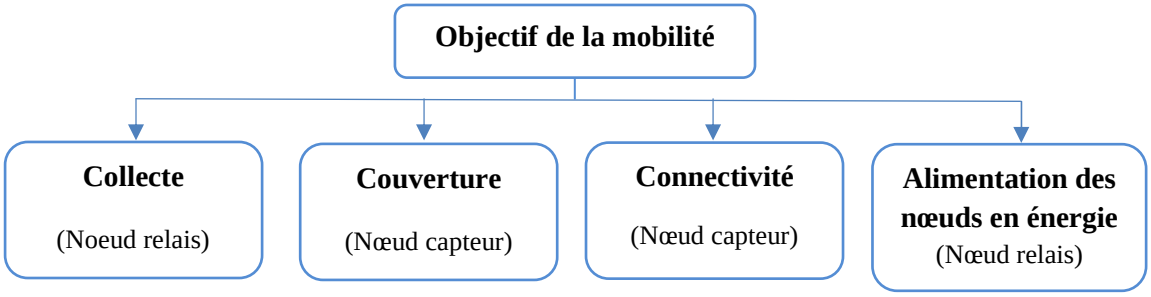


Figure 2.20 – Objectifs de la mobilité

- ***Collecteurs de données:*** ces collecteurs ont pour rôle de collecter les mesures provenant des nœuds sources et éventuellement de les agréger. Généralement, un "*Cluster-Head* " ou chef de cluster est utilisé comme NC dans une architecture hiérarchique où les NS sont partitionnés en plusieurs groupes. Cette catégorie de nœud sera vue en détail dans le chapitre 3 [11].
- ***Couverture et connexité :*** un nœud capteur permet de surveiller une zone appelée zone de couverture, cette zone est souvent considérée comme un disque de rayon *R*. Un nœud est capable de détecter n'importe quel événement qui se passe dans sa zone de couverture. D'un autre côté, la vision d'un capteur dépend du rayon de réception de son module de communication *Rs*. Un nœud peut communiquer avec un autre nœud que si ce dernier se trouve dans sa zone de communication, c'est à dire si la distance Euclidienne entre les deux nœuds est inférieure ou égale à *Rc*.

On dit qu'un réseau est connecté, si un nœud donné peut communiquer avec n'importe quel autre nœud soit directement soit par l'intermédiaire d'autres nœuds. La figure 2.21 illustre un exemple de la zone de couverture et de communication d'un capteur. Souvent on considère le rayon de communication plus grand que le rayon de couverture. La couverture de la zone d'intérêt est composée de l'union de toutes les zones de couverture des nœuds du réseau [19].

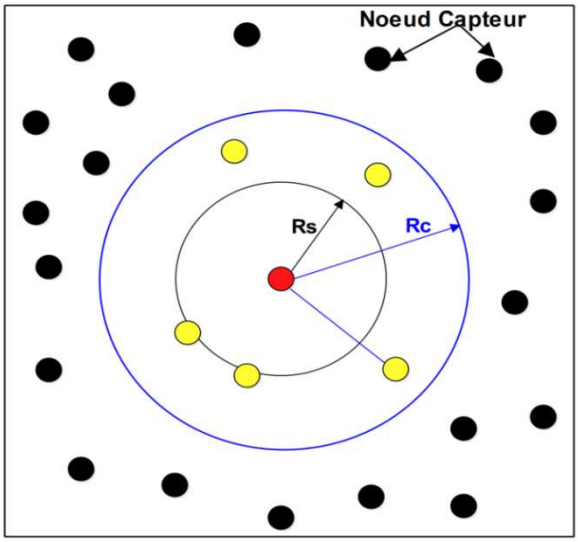


Figure 2.21 – Zones de couverture et de communication d'un capteur

Après un déploiement aléatoire, une des questions fondamentales qui se pose après la localisation est la couverture. Elle consiste à déterminer le degré de couverture de la zone d'intérêt.

Nous pouvons distinguer trois sortes de couverture ; la couverture « dispersée », où les nœuds déployés assurent la couverture d'une partie de la zone d'intérêt, la couverture « dense », dans ce cas la zone est presque complètement couverte. La couverture « redondante », où nous trouvons des zones couvertes plusieurs fois par plusieurs nœuds. En général, la couverture peut être considérée comme la mesure de la qualité de service d'un RCSF. Par ailleurs, une mauvaise répartition des capteurs engendrera une perte de certains nœuds qui n'auront pas de voisins et qui, par conséquent, seront isolés et déconnectés du réseau.

- ***Connectivité :*** les rayons de communication (désignés par *Rc* dans la figure 2.21) et les localisations physiques des nœuds individuels définissent la connectivité d'un réseau. S'il y a toujours une connexion réseau (même à travers plusieurs sauts) entre toute paire de nœuds capteurs du RCSF, le réseau est dit connexe. Les RCSFs sont généralement composés d'un large nombre de capteurs déployés sur une zone d'intérêt. De ce fait, ces réseaux sont caractérisés par une forte densité de nœuds assurant une bonne connectivité du réseau [10,142]. Néanmoins, vu la forte contrainte énergétique des capteurs, les nœuds peuvent disparaître du réseau en le partitionnant en plusieurs composantes connexes. Ainsi, les liens de bout en bout, du *Sink* vers la source, disparaissent. Dans ce cas, l'utilisation

d'un *Sink* mobile améliore la connectivité du réseau en permettant la collecte de données à partir de plusieurs composantes isolées du réseau, comme le montre la figure 2.22.

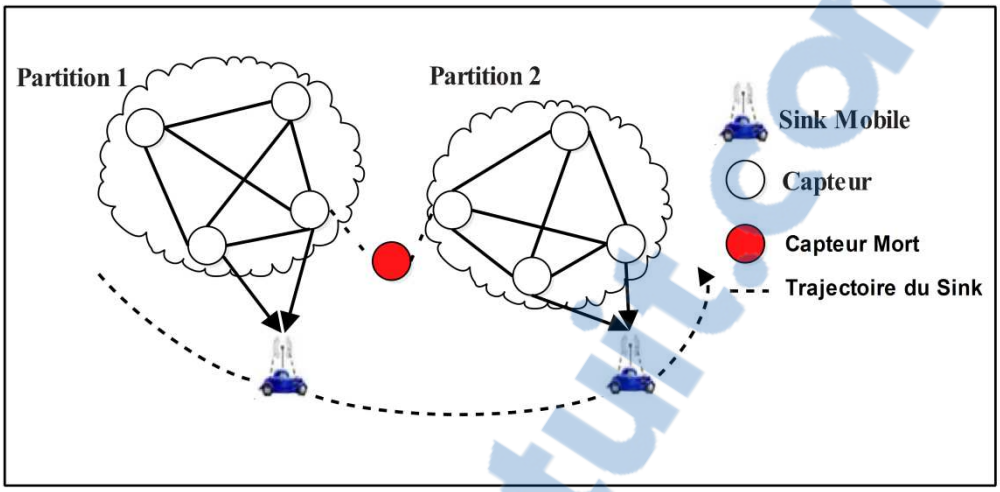


Figure 2.22 – Amélioration de la connectivité du réseau grâce à la mobilité du Sink

- ***Alimentation des nœuds en énergie :*** cette approche est très récente et traite simultanément le transfert d'énergie de plusieurs nœuds pour une recharge sans fil évolutive dans les RCSFs. Tous les schémas de rechargement existants reposent sur l'utilisation d'un chargeur mobile qui parcourt tous les points du réseau à travers plusieurs positions des nœuds rechargés. Cependant, ils se concentrent sur l'efficacité du transfert d'énergie et négligent l'énergie engendrée par le mouvement du chargeur. Dans cette approche, le chargement sans fil est modélisé par un problème d'optimisation de chemin pour le chargeur mobile, avec comme objectif une fonction qui minimise le nombre d'emplacements d'arrêt dans le chemin. En raison du problème NP-Hardness [144], ils proposent une heuristique simple mais efficace. Elle est basée sur un partitionnement cyclique pour trouver le nombre minimum d'emplacements permettant au chargeur mobile de reconstituer toutes les batteries des nœuds dans le réseau. Les résultats de l'évaluation montrent que l'approche proposée réduit significativement la consommation d'énergie totale du chargeur mobile, tout en utilisant une technique à faible complexité qui permet une évolutivité à un plus grand nombre de nœuds [145].

2.9 Conclusion

Dans ce chapitre, après avoir présenté brièvement les RCSFs, leurs caractéristiques ainsi que leurs applications, nous avons détaillé l'état de l'art sur les RCSFs-EM, objet de notre étude. Ce détail concerne essentiellement la classe particulière de RCSF que nous traitons tout le long de cette thèse, à savoir les RCSFs-EM dédiés aux applications critiques de surveillance. Une nouvelle classe de RCSFs suscite un immense intérêt des chercheurs universitaires, des industriels ainsi que des décideurs, étant donné sa large utilisation dans divers domaines très importants et d'actualité. Cette mobilité concerne les nœuds capteurs ou bien le nœud puits et dans un cas beaucoup plus complexe les deux en même temps. Dans notre étude, nous avons exploité la mobilité du *Sink* afin d'améliorer le processus de collecte, économiser de l'énergie,

éviter la perte de paquets, et par conséquent étendre la durée de vie du réseau. Nous avons donné les raisons ainsi que les défis de la mobilité ensuite nous avons décrit les différentes formes et classes de la mobilité en mettant l'accent sur la mobilité du *Sink* en explicitant ses avantages et ses inconvénients. Cependant, nous sommes arrivés à la conclusion que la mobilité réglait certains problèmes dans les RCSFs mais pour pouvoir l'utiliser efficacement, il fallait considérer plusieurs problématiques qui justement seront étudiées dans la suite de ce manuscrit et qui représentent le cœur de cette thèse.

Dans le chapitre suivant, nous entamerons le processus de collecte de données dans un RCSF-EM jouant le rôle du collecteur.

Chapitre 3

La collecte de données dans un RCSF avec élement mobile

Sommaire

Chapitre 3 La collecte de données dans un RCSF avec élément mobile

3.1 Introduction

La collecte de données est généralement réalisée dans les RCSFs par les nœuds relais qui transmettent les données vers un centre de contrôle statique (station de base). Motivés par d'importantes applications (principalement reliées à l'intelligence ambiante et la surveillance à distance) et comme c'est une première étape vers l'introduction de la mobilité, nous proposons une idée basique qui est d'avoir un *Sink* qui se déplace dans la zone du réseau et qui collecte les données à partir des nœuds capteurs [19].
L'une des premières problématiques qui se pose dans les RCSFs-EM est la collecte de données qui est d'ailleurs une tâche fondamentale des RCSFs. La collecte de données vise à collecter les données captées par les nœuds vers une station de base (*Sink*) prédéfinis pour analyse et traitement de ces données. Ce processus rend possible la communication entre ces nœuds. Un protocole de routage doit fournir des techniques rapides et sures pour la propagation de ces données [12]. La question primordiale qui se pose est "Comment réussir une collecte de données fiable et efficace en prévoyant un élément mobile dans le réseau". Cette mobilité concerne soit les nœuds capteurs ou bien le *Sink*. Nous nous focalisons dans cette thèse sur la mobilité du *Sink*.

Cependant, il existe plusieurs types d'éléments mobiles que nous avons évoqués en détail dans le chapitre précédent. Ainsi l'une des problématiques à laquelle nous devons faire face dans les RCSFs-EM est le choix de l'élément mobile (*Sink* ou noeud relais) à déplacer dans le réseau. Lorsque nous parlons du choix de cet élément mobile, nous entendons par là, un collecteur qui assure une collecte de données organisée, efficace et surtout fiable tout en respectant un temps raisonnable et un rendement énergétique positif, et par conséquent étendre la durée de vie du réseau.

Dans ce chapitre, nous allons faire un tour d'horizon des différentes phases de collecte ainsi que les principaux types de collecte de données. Les architectures de collecte de données via les *Sink*s mobiles, classifiées en fonction du type de mobilité : aléatoire, déterministe et contrôlée, seront aussi présentées.

3.2 Processus de collecte des données

Comme nous l'avons vu précédemment dans un RCSF, les nœuds capteurs sont déployés généralement de manière aléatoire et forment une zone de couverture. Le processus de collecte de données est réalisé dans cette zone, les nœuds sont chargés de collecter et d'acheminer les informations relevées par les capteurs vers le point de collecte « *Sink* » qui les récupère à son tour et les retransmet au centre de traitement. La collecte des données est considérée comme l'une des taches fondamentales dans les RCSFs
Afin de collecter toutes les données générées dans le réseau, dans [142], les auteurs utilisent un *Sink* mobile qui commence par élire parmi ses nœuds voisins un *Sink* -manager. Ce *Sink* -manager a pour principal objectif de traiter les requêtes du *Sink* en les envoyant vers la ligne centrale et en transmettant les réponses reçues vers le *Sink*. Puisque le *Sink* est mobile, il se peut que ce dernier soit hors de la portée radio de son *Sink*-manager, rendant ainsi la réception des messages impossible.

Pour résoudre ce problème, ils utilisent le principe du progressive *Footprint Chaining* [143], afin de gérer la mobilité du *Sink* et assurer la présence d'un chemin vers le *Sink* -manager.
Dès que le *Sink*-manager reçoit la requête du *Sink*, il la transmet en direction de la ligne centrale.
Le premier nœud communiquant qui reçoit la requête, la redirige vers son group-leader afin de vérifier la disponibilité de l'information demandée par le *Sink*. Si la donnée existe, un message réponse est renvoyé au *Sink*-manager. Sinon, la requête est transmise vers les deux group-leaders supérieur et inférieur.
Ce processus se répète jusqu'à trouver le chef de groupe possédant les données correspondantes, ou atteindre l'une des extrémités de la ligne virtuelle.
Dans la mobilité aléatoire, le *Sink* suit une trajectoire aléatoire dans le domaine des capteurs et d'importantes questions liées à la stratégie de collecte de données ont été posées. Habituellement, le *Sink* utilise une « Pull-Strategy » ou une collecte de données à partir des nœuds capteurs. Dans une « Pull-Strategy », un noeud transmet ses données uniquement lorsque le *Sink* lui initie une demande, alors que dans une « Push-Strategy », un nœud envoie ses données de manière proactive au *Sink*. Les auteurs dans [146,147] montrent que la mobilité aléatoire du *Sink* peut être utilisée pour réduire au maximum l'énergie dissipée et l'énergie dissipée moyenne par nœud en comparaison avec le cas d'un *Sink* statique. La collecte de données à un seul saut conduit à une plus forte réduction de la consommation d'énergie, parce qu'il n'existe aucune donnée chargée, relayée par les nœuds capteurs. Cependant, il peut également entraîner une collecte de données incomplète à partir du RCSF car avec un modèle de mobilité aléatoire il n'y a aucune garantie que le *Sink* atteigne tous les nœuds du réseau ou il pourrait prendre trop de temps à faire cela. Si le temps nécessaire pour une couverture complète de la zone doit être encore plus petit, alors le *Sink* peut être programmé pour collecter des données à partir de tous les nœuds qui ont un nombre maximal de sauts plus grands qu'un. Il en résulte une charge relais sur les nœuds capteurs, et donc l'énergie dissipée maximale et l'énergie dissipée moyenne augmentent par nœud par rapport au cas de la collecte de données à un seul saut [14].

La technique de collecte de données est utilisée pour collecter les données agrégées à partir du nœud capteur au nœud *Sink*. L'objectif principal du processus de collecte de données est de réduire le délai et d'améliorer la durée de vie des nœuds capteurs et par conséquent la durée de vie du réseau. Il existe différentes techniques utilisées pour collecter les données à partir du nœud source au *Sink*.
Tout d'abord, tous les capteurs sont statiques et le réseau est considéré comme stationnaire. Les nœuds capteurs statiques transfèrent les données au *Sink* par un ou plusieurs sauts [148,149]. Donc, les capteurs les plus proches du *Sink* s'épuisent rapidement.

Deuxièmement, la forme hiérarchique de collecte de données. Les nœuds peuvent être classés en couche inférieure et en couche supérieure. Les nœuds dans les couches de niveau inférieur sont des nœuds capteurs homogènes. Les nœuds dans la couche supérieure sont appelés des Clusterheads et sont plus puissants que les nœuds dans la couche inférieure.

Troisièmement, le collecteur mobile est utilisé pour collecter les données périodiquement. Un observateur de données mobile est utilisé pour collecter les données dynamiquement. Les

nœuds qui peuvent être situés plus près de l'observateur de données peuvent télécharger directement les données. Les nœuds qui sont situés loin de l'observateur peuvent transmettre les données par acheminement.

Le problème de collecte de données à un seul saut SHDGP (*Single Hop Data Gathering problem*) et la collecte de données mobile sont les deux approches qui peuvent être utilisées pour augmenter la durée de vie du réseau. La collecte de données à un seul saut (*SHDGP*) [148] est utilisée pour obtenir une consommation d'énergie uniforme. L'algorithme de collecte de données mobile est utilisé pour trouver l'ensemble minimal de points dans le RCSF. Ils servent de points de collecte de données pour un nœud mobile.

- ***Collecte de données à économie d'énergie :*** les auteurs dans [150,151] proposent un framework de collecte de données à économie d'énergie pour une corrélation spatio-temporelle parmi les données capturées. Pour cela, le réseau entier est divisé en plusieurs sous-régions où chaque région est couverte par un nœud cluster. Les clusters (grappes) sont basés sur les données d'échantillonnage, et l'ordonnancement est basé sur le principe de cluster qui est beaucoup plus précis qu'un ordonnancement basé uniquement sur la zone de détection des nœuds capteurs. Ceci permet d'économiser de l'énergie sans perdre la fiabilité de surveillance. Avec l'algorithme PLAMLIS[152], la restauration de données est améliorée et la consommation d'énergie est diminuée.

- ***Algorithme distribué pour un ensemble dominant connecté (CDS***: *Distributed algorithm for minimum Connected Dominating Set****) :*** les auteurs dans [153] proposent un algorithme distribué pour un ensemble minimum de dominants connectés (CDS). Cet algorithme fournit une amélioration du coefficient d'approximation égale à 6,91 qui est due à une analyse précise de la relation entre la taille d'un ensemble maximal indépendant et un CDS minimal dans l'unité graphique du disque. Chaque premier nœud diffuse (broadcast) à ses voisins et à l'ensemble de ses voisins. Après réception des informations adjacentes de tous ces voisins, il se déclare dominateur, si et seulement si il a deux voisins non adjacents.

- ***Routage opportuniste dans un RCSF :*** dans [154], les auteurs proposent un codage de source et un protocole de routage opportunistes pour une collecte de données corrélée dans un RCSF. Ce protocole améliore l'efficacité de la collecte de données en exploitant la compression opportuniste de données et la diversité coopérative associée à la diffusion sans fil. Il réduit ainsi la consommation d'énergie de près de 32% comparativement au schéma Greedy.

- ***Routage basé sur Cluster dirigé par la densité :*** dans [155], les auteurs proposent un protocole de routage basé sur cluster dirigé par la densité. Ce protocole organise les nœuds en Clusters, et à l'intérieur de chaque groupe de Custers un capteur de données est acheminé vers un Clusterhead, ainsi la donnée redondante est supprimée. Dans ce protocole il existe aussi un certain nombre de transmissions intégrales.

- ***Arbre minimum Steiner avec un nombre minimum de points Steiner :*** les auteurs dans [150], proposent un arbre minimum Steiner avec un nombre minimum de points Steiner. Le nombre minimum de nœuds relais pour connecter tous les nœuds capteurs revient à utiliser un nombre minimum de points Steiner et à relier tous les sommets donnés. L'idée principale de l'algorithme d'approximation est d'appliquer l'algorithme d'arbre minimum Steiner couvrant tous les sommets donnés et d'insérer une étape intermédiaire lorsque les arêtes restantes entre les sommets donnés sont plus longues que la zone de communication. Un point Steiner (avec trois arêtes) est ajouté pour connecter trois composants connectés en un seul. Si le point Steiner peut connecter chaque composant avec une arête, alors la longueur est égale à la zone de communication. Par contre, si une arête est plus longue que la zone de communication sélectionnée par l'algorithme, le nombre minimal de points Steiner est également ajouté à l'arête pour la casser en plus petites parties avec une longueur inférieure ou égale à la zone de communication.

3.3 Types de données à collecter

- ***Données Multimédias :*** **l**es applications multimédia sont caractérisées par un volume conséquent de données : alors que 2 à 3 octets suffisent à un capteur scalaire pour représenter des données, l'ordre de grandeur du codage d'une image, intrinsèque à sa taille et à sa résolution, peut avoisiner les milliers d'octets. Cet ordre de grandeur devient plus élevé dans le cas d'une vidéo. Ainsi, l'introduction d'aspects multimédia pour la surveillance avec un RCSF nécessite des efforts significatifs pour développer des mécanismes de contrôle adaptés, et ce à tous les niveaux des couches protocolaires. Avec des données multimédias, la qualité de service (*QoS*) devient désormais une exigence fondamentale pour la transmission dans un environnement contraint en ressources [5]. Les capteurs vidéo et audio permettent de capturer du son, des images fixes ou animées de l'événement détecté. Ils peuvent être disposés dans un réseau à un seul niveau ou d'une manière hiérarchique.
- ***Données Scalaires* :** ce sont des mesures de phénomènes physique telles que : la température, la lumière, la pression, l'humidité. Ce sont des données envoyées par un nœud émetteur qui charge cette donnée dans un paquet au *Sink* de façon périodique à la demande, suite à un évènement ou en continu. Un capteur scalaire a besoin de quelques octets seulement pour représenter des données scalaires.

3.4 Différents types de collecte de données

Selon [24,156], il existe trois (03) méthodes de collecte de données, comme nous pouvons le voir sur la figure suivante :

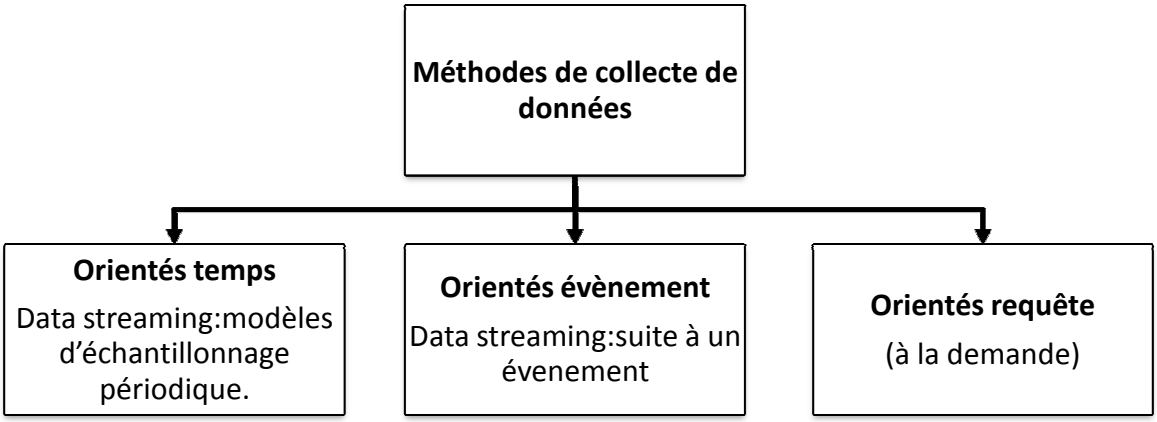


Figure 3.1 – Types de collecte de données dans les RCSFs

3.4.1 Collecte de données à la demande

Lorsque l'on souhaite connaitre l'état de la zone de couverture à un moment donné ***"t"***, le *Sink* diffuse des demandes vers cette zone pour que tous les nœuds capteurs remontent leurs données capturées. Ces données sont alors acheminées à travers une communication multi-sauts vers le *Sink* comme c'est illustré dans la figure 3.2. Cette collecte représente un dialogue bi-directionnel entre les nœuds et le *Sink*. Une requête peut être alors envoyée sur demande de l'utilisateur via le *Sink* vers les nœuds qui, à leur tour, transmettent leurs informations à l'utilisateur [108,20].

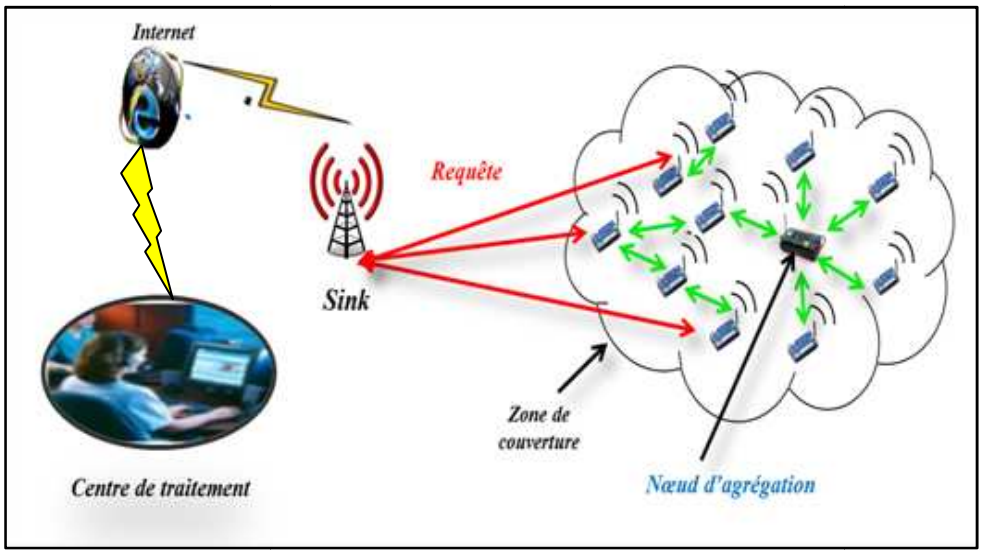


Figure 3.2 – Collecte de données à la demande

Cependant, une seconde catégorie de collecte de données peut être étudiée: le **« *Data streaming* »**.

3.4.2 Collecte de données de type « Data streaming »

Dans cette catégorie, les nœuds envoient leurs flux de données au *Sink* sans qu'il y ait une demande. Cette méthode est très utilisée surtout dans les applications de surveillance. Dans cette catégorie, nous distinguons des modèles basés sur les évènements « *event driven* » et des modèles **d'échantillonnage périodique.**

3.4.2.1 Collecte de données suite à un événement

Si un événement se produit en un point dans la zone de couverture (changement brusque de température, mouvement, ...), les capteurs situés à proximité remontent alors les informations relevées et les acheminent jusqu'au *Sink*. Pour supporter ce modèle en étant efficace en termes d'énergie et de rapidité de réponse, un nœud capteur doit être conçu de telle sorte que sa consommation d'énergie soit très faible en l'absence de tout événement (mode en veille), et que sa durée d'activité soit réduite dans le cas de détection des évènements. De nombreuses applications exigent la combinaison des deux modèles, le modèle « event driven» et le modèle d'échantillonnage périodique en même temps [19]. La figure 3.3 représente ce type de collecte.

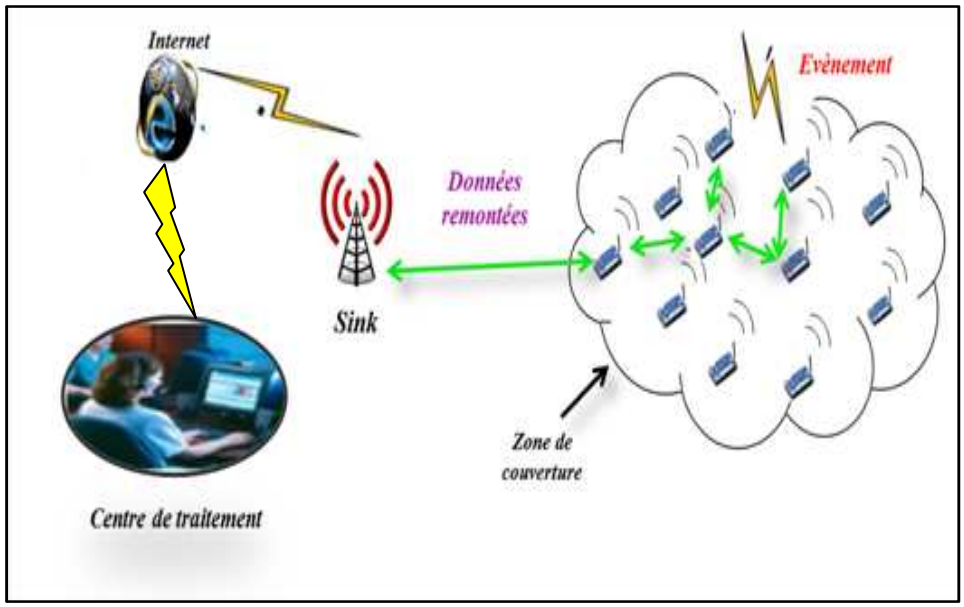


Figure 3.3 – Collecte de données suite à un événement

3.4.2.2 Collecte de données avec modèle d'échantillonnage périodique

Ce type de collecte est caractérisé par l'acquisition de données par un certain nombre de nœuds capteurs à distance et leur transmission au *Sink* d'une manière périodique. La période d'échantillonnage dépend principalement de la vitesse des changements environnementaux et des caractéristiques intrinsèques que nous voulons capturer. Ce modèle de données est approprié aux applications dans lesquelles certains phénomènes doivent être surveillés constamment, comme la température ou l'humidité dans un espace. La conception du modèle

périodique de données doit prendre en considération plusieurs facteurs influençant son fonctionnement. Parfois, les changements dynamiques de l'état de la zone à surveiller peuvent ralentir ou accélérer l'échantillonnage; si le nœud capteur peut adapter son taux d'échantillonnage à l'évolution dynamique de l'environnement, nous réduisons le sur-échantillonnage et par conséquent nous réduisons la quantité d'énergie consommée. Une autre question critique de conception se pose à savoir « la relation entre deux ou plusieurs nœuds ». Si deux nœuds fonctionnent avec des périodes identiques, des collisions de paquets de ces deux nœuds sont susceptibles de se produire périodiquement. Il est donc essentiel pour les nœuds d'être capables de détecter ce genre de collisions et d'introduire un décalage entre les séquences de transmission afin d'éviter ces dernières [20].

Ce type de collecte est utilisé, lorsque nous souhaitons observer le changement d'état d'un phénomène donné (tel que la surveillance du mouvement du personnel d'une entreprise). Ainsi le collecteur récolte les données récupérées par les nœuds capteurs après chaque intervalle de temps, c'est-à-dire, de façon cyclique.

3.5 Eléments mobiles dans un RCSF

Pour mieux comprendre les caractéristiques spécifiques aux RCSF-EM, nous allons présenter les principaux éléments mobiles dans un RCSF ainsi que l'architecture du réseau avec ces différents éléments [11].

- ***Nœuds capteurs réguliers*** ou nœuds : ce sont des sources d'information dont la tâche principale est la capture. Ils peuvent également être transmetteurs ou bien des messages relais dans le réseau, selon le paradigme de communication adopté.
- ***Sinks (Stations de base) :*** ces nœuds sont les destinations de l'information. Ils collectent les données capturées par les nœuds capteurs soit directement (en parcourant et en collectant les données de chaque capteur dans le réseau) ou bien indirectement (à travers des nœuds intermédiaires). Ils peuvent utiliser les données provenant des capteurs de manière autonome ou les rendre disponibles pour les utilisateurs intéressés à travers une connexion internet.
- ***Nœuds supports spéciaux (SSN) :*** Ce sont des nœuds capteurs dotés d'un mécanisme de mobilité et souvent caractérisés par une capacité de stockage et de traitement élevée. Ils sont déployés dans le réseau comme étant des nœuds collecteurs intermédiaires, où ils exploitent la mobilité soit pour assurer la collecte de données ***(nous parlons alors de nœuds relais mobiles),*** soit pour jouer le rôle de passerelles mobiles ***(et nous parlons alors de nœuds délocalisables).***

Dans un RCSF, la mobilité peut être intégrée selon différentes sortes :

- Les nœuds possèdent un module de mobilité intégré (moteur,…),
- Les nœuds sont embarqués dans un élément qui bouge (être humain, navette de transport,…),
- Les nœuds bougent grâce à une force externe (vent, inondation,..).
- La mobilité dans les RCSFs constitue ces derniers temps un axe de recherche prometteur car elle est exploitée pour résoudre plusieurs problématiques liées aux

RCSFs (consommation d'énergie, latence, collecte de données et durée de vie du réseau.

3.6 Architectures d'un RCSF avec éléments mobiles

Dans cette section, nous présentons les différents types d'éléments mobiles (EMs) en considérant la mobilité dans les RCSFs, et en mettant l'accent sur les aspects architecturaux. Selon le scénario spécifique, les nœuds supports peuvent être présents ou non. Lorsqu'il n'y a que les nœuds réguliers, l'architecture résultante du RCSF-EM est homogène ou plate. Par contre, lorsque les nœuds supports sont présents, l'architecture résultante du RCSF-EM est non homogène ou à plusieurs niveaux. En outre, elle est différente du RCSF traditionnel, qui est généralement limité en densité ; les RCSFs-EM peuvent également être dispersés. Comme l'architecture du réseau dépend fortement du rôle de l'élément mobile [11].

3.6.1 Nœuds délocalisables (*Relocatable Nodes)*

Les nœuds délocalisables sont des nœuds mobiles qui changent leur emplacement pour mieux déterminer la zone de capture, ou transmettre des données à partir des nœuds source au *Sink*. En contraste avec les collecteurs de données mobiles (qui seront discutées dans la section suivante), les noeuds délocalisables ne comportent pas de données lorsqu''ils se déplacent dans le réseau. En effet, ils ne changent que la topologie du réseau qui est supposée être assez dense pour assurer la connectivité et la couverture du réseau. Plus précisément, après le déplacement vers le nouvel emplacement, ils restent généralement fixes et transmettent les données le long de chemins multi-sauts [11]. Une architecture de RCSF-EM basée sur des nœuds délocalisables est représentée dans la figure 3.4.

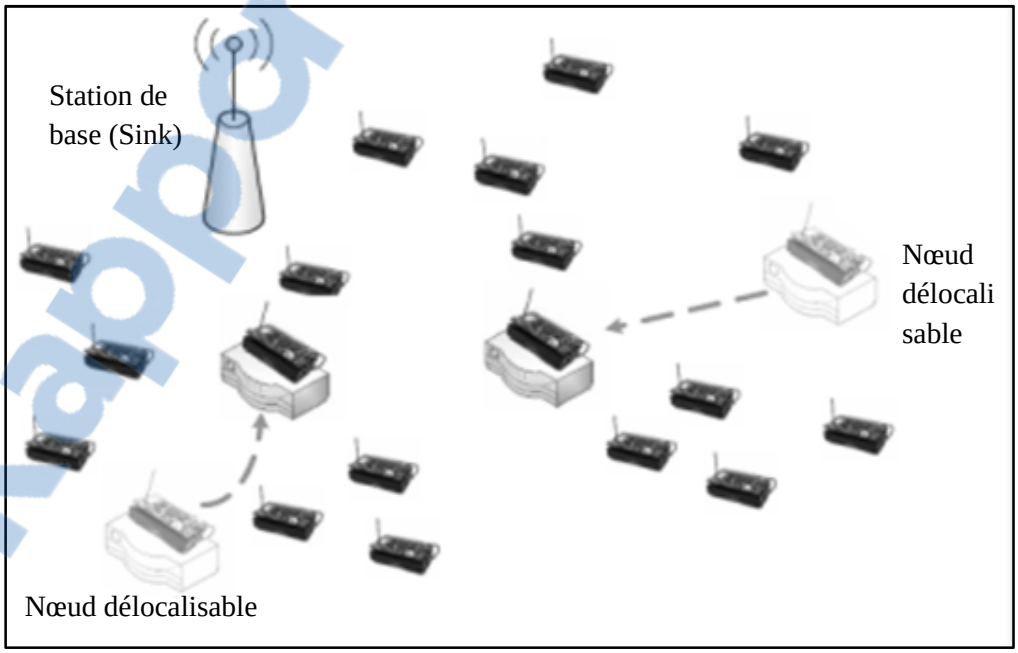


Figure. 3.4 – Architecture d'un RSCF avec nœuds délocalisables.

En théorie, les nœuds peuvent être délocalisables et dans la plupart des cas, les éléments mobiles spéciaux (par exemple les nœuds supports) sont utilisés. Un système avec des nœuds délocalisables, spécifique à la gestion de la topologie, a été proposé dans [157]. En particulier, les nœuds PILOTES sont spécialement prédéfinis, intelligents, gèrent une topologie légère. Ces nœuds sont utilisés pour rétablir la connectivité des liens défectueux du réseau. Dans le détail, les nœuds PILOTES[158] se déplacent vers des régions où la connexion entre nœuds est instable ou défectueuse, et agissent comme des ponts. Par conséquent, ils changent activement la topologie du RCSF afin d'améliorer à la fois la fiabilité de communication et l'efficacité énergétique. Quelques algorithmes de placement des nœuds délocalisables ont été étudiés [159,160] afin d'améliorer la connectivité du réseau. Les nœuds délocalisables peuvent également être utilisés pour traiter le problème de couverture des zones de capture.
Dans ce cas de figure, la principale préoccupation n'est pas d'assurer la connectivité du réseau, mais d'éviter la couverture des vides où la densité des nœuds n'est pas suffisante pour caractériser correctement un phénomène ou détecter un événement. Des approches spécifiques à la couverture de la zone de capture peuvent se concentrer sur le déploiement des nœuds capteurs [161,162], le capteur de délocalisation et de dissémination [163,164], ou les deux à la fois [165,166]. Les nœuds délocalisables fournissent une approche basée sur la mobilité pour les RCSFs ; c'est dans ce sens que les éléments mobiles ne sont pas exploités activement pour la collecte des données.
Par conséquent, dans ce qui suit nous n'allons pas discuter des solutions basées sur les nœuds délocalisables. Nous allons plutôt voir en détail les approches basées sur la mobilité où les EMs sont activement utilisés pour la collecte des données.

3.6.2 Collecteurs de données mobiles (CDMs)

Ce sont des éléments mobiles qui visitent le réseau pour collecter des données générées à partir des nœuds sources. Selon la façon dont ils gèrent les données collectées, les collecteurs mobiles peuvent être soit des *Sinks* mobiles ou bien des relais mobiles.

3.6.2.1 Collecteurs de données mobiles de type Sinks mobiles (SMs)

Ce sont des nœuds mobiles qui représentent la destination finale des messages générés par les nœuds capteurs, de ce fait, ils représentent le point final de la collecte de données dans les RCSFs-EM. Ils peuvent soit consommer de façon autonome les données collectées pour leurs propres besoins ou les rendre accessibles et disponible aux utilisateurs distants en utilisant une connexion Internet sans fil à longue portée [11]. L'architecture d'un RCSF avec un collecteur de données mobile de type *Sink* mobile est représentée sur la figure 3.5.

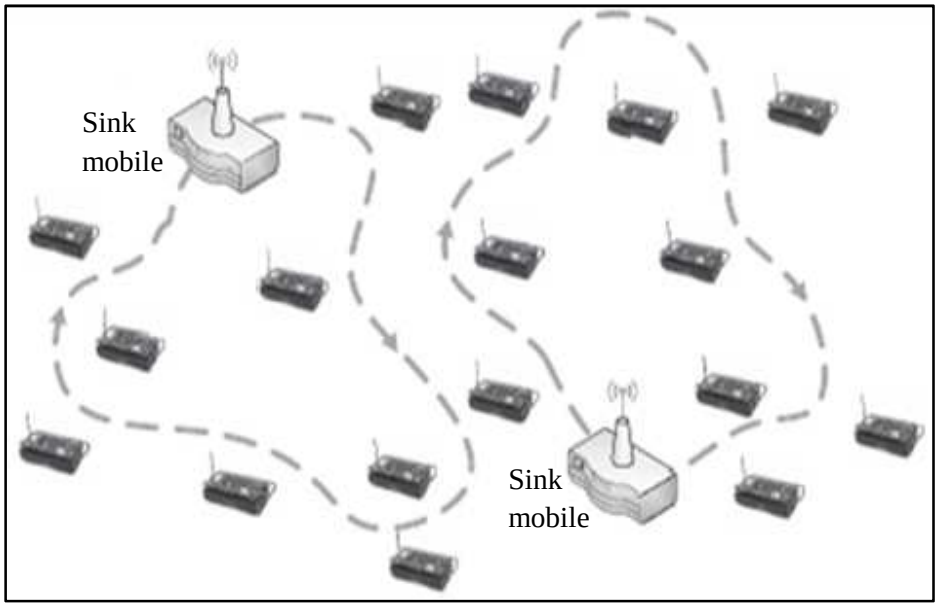


Figure 3.5 – Architecture d'un RCSF avec CDMs de type Sink mobile

Les *Sinks* mobiles ont été largement employés dans la littérature [146,147,11,91,134]. Dans ces cas de figures, les nœuds capteurs ordinaires sont statiques et déployés d'une manière très dense dans la zone de déploiement. Un ou plusieurs *Sinks* mobiles se déplacent à travers le RCSF pour collecter des données provenant des nœuds capteurs. Il est à noter que malgré que le chemin entre les nœuds sources et le *Sink* mobile soit multi-sauts, et bien que le chemin réel change avec le temps, la position du *Sink* mobile n'est pas fixe.

Une approche différente ciblée pour la collecte de données dans des scénarios urbains a été considérée dans [167]. Dans ce cas, les personnes mobiles agissent comme des *Sinks* mobiles collectant des données environnementales (telles que la concentration des polluants et des conditions climatiques) pour leurs propres objectifs. Le scénario de référence d'un RCSF est représenté par un RCSF dispersé où plusieurs *Sinks* mobiles peuvent être en contact avec un seul nœud capteur à la fois [11].

3.6.2.2 Collecteurs de données mobiles de type Relais Mobiles (RMs)

Ce sont des nœuds supports qui rassemblent les messages des nœuds capteurs, les stockent et transmettent ces données collectées vers la station de base (*Sink*). Ils ne sont pas les points finaux de communication, mais agissent comme des transmetteurs mobiles. Cela signifie que les données collectées se déplacent avec eux, jusqu'à ce que le relais mobile entre en contact avec le *Sink* ou la station de base. L'architecture de base d'un RCSF-EM avec collecteurs de données mobiles de type nœud relais mobile est représentée sur la figure 3.6.

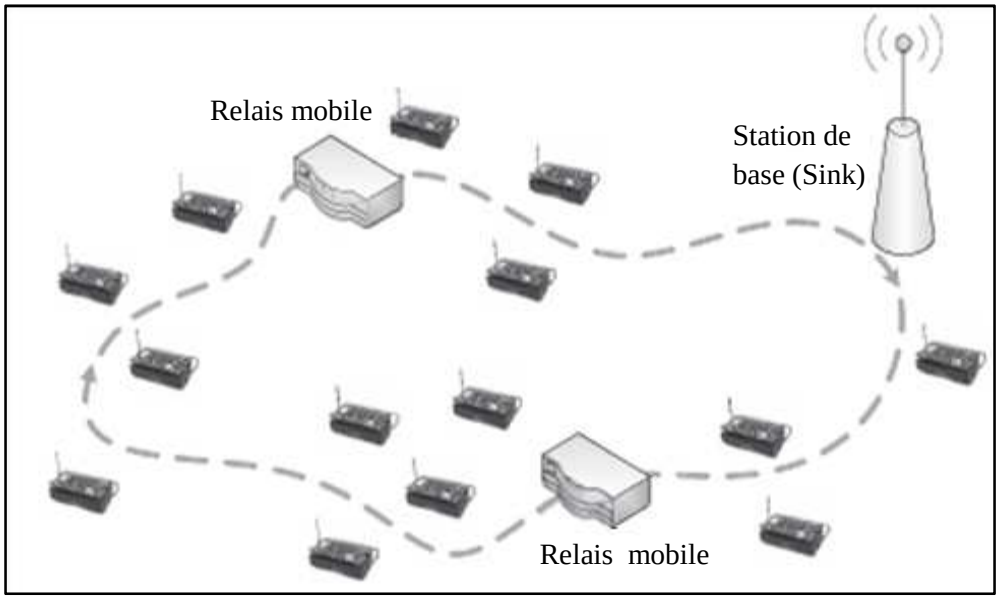


Figure 3.6 – Architecture d'un RCSF avec CDMs de type nœud Relais mobile

La collecte de données avec nœuds relais mobiles dans les RCSFs a été proposée dans le système de données MULE [133]. Le système de données MULE se compose d'une architecture à trois niveaux, où le niveau intermédiaire est représenté par des relais, appelé mobiles ubiquitaires LAN extensions (MULEs). Les nœuds capteurs, qui sont supposés être statiques- attendent qu'un MULE soit en contact avant de commencer les communications. Alors, le MULE collecte les données et se dirige vers un emplacement différent, éventuellement en passant par la station de base, où les données collectées sont stockées et mises à la disposition des utilisateurs distants. Des approches similaires ont été également utilisées dans un contexte de réseaux opportunistes [168]. L'une des approches la plus connue est donnée par le schéma de messages Ferrying [131]. Les messages Ferrying assurent le service de message relayé dans les réseaux Ad-hoc dispersés et mobiles. Les messages Ferrying se font autour de la zone du réseau tout en collectant les données provenant des sources. Ils transportent les données stockées et les transmettent vers les destinations. Ainsi les messages Ferrying peut être considéré comme une infrastructure de communication mobile qui permet le transfert de données dans un RCSF dispersé.

- **Comparaison entre RCSF avec CDMs de type Sink mobile et RCSF avec CDMs de type nœud Relais mobile dans le processus de collecte**

La table 3.1 présente une comparaison entre les deux RCSFs de type *Sink* mobile et nœud relais mobile ainsi que leur impact sur le processus de collecte de données et qui fera l'objet d'une partie de notre étude dans le chapitre 4) [14].

Modèles de collecte avec CDMs / ***Caractéristiques***		***RCSF avec CDMs de type Sink mobile (SM)***	***RCSF avec CDMs de type nœud(s) Relais mobile(s) (MRs)***
Eléments mobiles		*Sink*	Nœuds supports (Nœuds relais)
Eléments stationnaires		Nœuds capteurs réguliers (NR)	NR et *Sink*
Nœud source		Nœuds capteurs réguliers (NR)	NR (par rapport aux nœuds relais). Nœuds relais (par rapport aux *Sinks*).
Nœud destinataire		*Sink*	Nœuds relais (intermédiaires, depuis les NRs) et *Sink* (final, depuis les nœuds relais)
Paradigme de collecte de données	***1ère étape***	Le *Sink* est en contact avec les NRs disponibles dans sa région, dès qu'il entre dans leur portée radio ***(phase de découverte)***	Les nœuds relais sont en contact avec les NRs disponibles dans leur région, dès qu'ils entrent dans leur portée radio ***(phase de découverte)***
	2ème étape	Début du transfert de données depuis les NRs vers le *Sink* ***(phase de transmission)***	Début du transfert de données depuis les NRs vers le nœud relais ***(phase de transmission)***
	3ème étape	Le Sink reçoit les données qu'il garde à son niveau ou bien les retransmet à un site distant	Le nœud relais stocke les données collectées et rejoint le *Sink* afin de les lui transmettre (il agit seulement comme expéditeur mobile)
	4ème étape	/	Les nœuds relais sont en contact avec le *Sink* ***(phase de découverte)***
	5ème étape	/	Début de transfert de données depuis le nœud relais vers le *Sink* ***(phase de communication)***
Impact sur le réseau		Changements fréquents de chemins dus à la position variante des EMs dans le cas d'une communication multi-sauts	

Table 3.1 – Comparaison entre RCSFs avec CDMs de type Sink mobile et RCSFs avec CDMs de type nœud relais mobile

3.6.3 Pairs mobiles (PMs)

Contrairement aux collecteurs de données mobiles, qui peuvent être soit des *Sinks* ou des nœuds relais spéciaux, les pairs mobiles sont des nœuds capteurs mobiles ordinaires dans un RCSF-EM. Comme ils ne peuvent être à la fois expéditeurs et relayeurs de messages dans le réseau, leurs interactions sont symétriques car le *Sink* lui-même peut être également mobile. Lorsqu'un nœud pair est dans la portée radio du *Sink*, le nœud pair transfère ses propres données ainsi que celles collectées par les autres nœuds pairs tout en se déplaçant dans la zone de capture. Une architecture d'un RCSF-EM avec des nœuds pairs mobiles est représentée dans la figure 3.7. Dans ce cas, le réseau est homogène et plutôt dispersé [11].

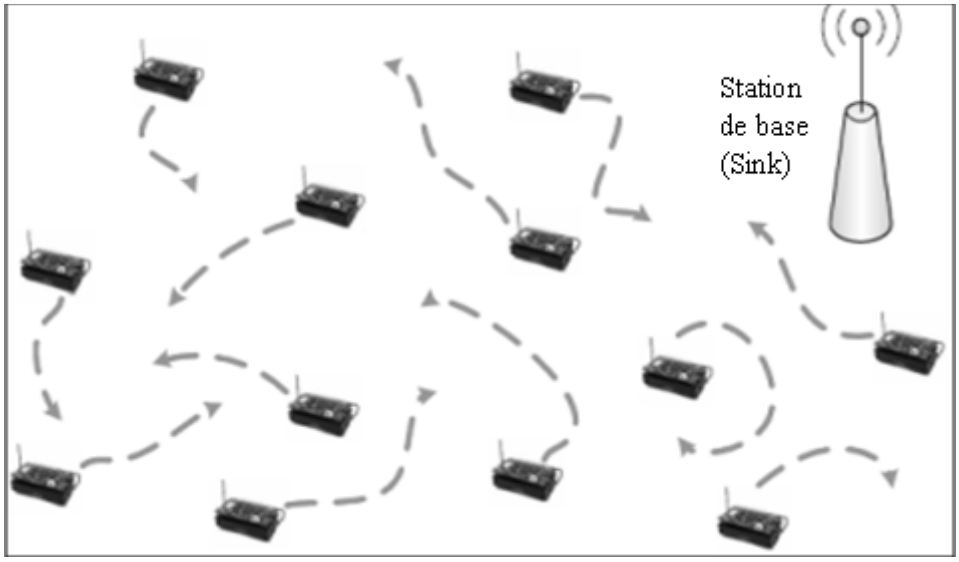


Figure 3.7 – Architecture d'un RCSF avec des pairs mobiles.

Les Pairs mobiles ont été employés avec succès dans un contexte d'applications de surveillance de faune par exemple, telles que le suivi de zèbres dans le projet ZebraNet [54] ou de baleines dans le système SWIM [170]. Les nœuds capteurs sont attachés aux animaux et agissent comme des pairs, de telle manière à ce qu'ils génèrent non seulement leurs propres données, mais transmettent toutes les données provenant d'autres nœuds avec lesquels ils étaient en contact précédemment. Lorsque les pairs mobiles se rapprochent du *Sink*, ils transfèrent toutes les données collectées. Les données qui ont déjà été transmises au *Sink* sont vidées par les nœuds pairs afin de sauvegarder le stockage. Les pairs mobiles peuvent également être utilisés pour la collecte de données opportuniste, pour des scénarios de capture urbaine [169,170]. Les exemples d'applications incluent la surveillance personnelle (par exemple, le suivi de l'exercice physique), la défense civile (par exemple, les dangers par rapport aux agents de police), et les applications collaboratives (par exemple, le partage d'informations à des fins touristiques).

Dans ce contexte, les capteurs ne sont pas utilisés principalement pour la surveillance de l'environnement, mais sont plutôt exploitées pour caractériser les gens en termes d'interactions et de contexte des informations. Un exemple est représenté par l'ordinateur de poche mobiscopes [171] où les appareils de poche tels que les téléphones cellulaires ou PDA, collectent les données provenant de l'environnement et les transmettent aux serveurs, qui offrent des services aux utilisateurs distants. Comme la plupart des problématiques dans le contexte des RCSFs-EM avec des pairs mobiles sont similaires aux problèmes des réseaux tolérants au délai (DTNs) classiques, nous n'allons pas les développer d'avantage. Les lecteurs intéressés par ce sujet peuvent se référer à la littérature [168] pour plus d'informations.

3.7 Architecture de collecte via des Sinks mobiles (SMs)

Dans cette section, nous présentons un éventail des architectures de collecte de données via des *Sinks* mobiles [5] car nous avons opté pour cette architecture pour effectuer la suite de notre travail. La découverte de la position courante du *Sink* mobile, l'établissement des routes et la transmission des données constituent, de manière générale, les étapes déterminantes d'un processus de collecte via des *Sinks* mobiles dans les RCSFs. Ces dernières années, de nombreuses propositions sur la dissémination de données via des *Sinks* mobiles ont vu le jour. Ces architectures de collecte peuvent être classées en trois grandes catégories en fonction du modèle de mobilité [13] :

- ***Architecture à mobilité aléatoire*** : le modèle de mobilité considère que le *Sink* est une entité mobile n'ayant aucune notion sur sa mobilité (trajectoire, vitesse, . . .) : un animal par exemple. Les architectures de collecte basées sur ce type de modèles de mobilité du *Sink* sont caractérisées par une autonomie totale de ce dernier en termes de vitesse et de trajectoire, rendant sa position prochaine fortement imprévisible [172].

- ***Architecture à mobilité déterministe*** : le modèle à mobilité déterministe, le plus simple d'entre eux, préconise que le *Sink* suit toujours une certaine trajectoire. Cette trajectoire pourrait être la périphérie de la zone de surveillance, une ligne droite, un chemin circulaire, ou une zone de rendez-vous consistant en un sous-ensemble de nœuds destinés à collecter les données du reste du réseau. Dans les schémas de collecte basés sur ce modèle, les capteurs le long de la trajectoire du *Sink* sont capables de calculer son temps de passage et optimiser en conséquence leurs tâches de collecte et de transmission de données : [173, 174]. Les parkings "intelligents" le long des routes où le *Sink* mobile (le conducteur) collecte des informations sur la disponibilité des places de stationnement d'une région d'intérêt est un excellent exemple d'application de cette famille.

- ***Architecture à mobilité contrôlée*** : le modèle à mobilité contrôlée désigne une situation où le *Sink* (ses mouvements) est sous le contrôle d'une entité extérieure au réseau [16, 175]. En fonction des objectifs de l'application et de la connaissance de l'environnement, le *Sink* est capable d'adapter ses mouvements (vitesse et trajectoire) pour atteindre un objectif précis.

3.8 Phases de Collecte de données dans un RCSF-EM

Dans cette section, nous décrivons les différentes phases du processus de collecte de données en soulignant les principaux enjeux. Pour plus de commodités, sans perte de généralité, nous allons nous référer à la situation représentée dans la figure 3.8, où un contact se produit entre l'élément mobile (*EM*) et un nœud capteur statique. La description peut être facilement étendue au cas où les nœuds capteurs sont également mobiles. En se référant à la figure 3.8, l'élément mobile est en contact avec un capteur quand ils peuvent s'atteindre mutuellement à travers des communications sans fil.

En général, un contact se produit lorsque deux ou plusieurs nœuds sont dans une zone de communication commune. La durée pendant laquelle les nœuds sont en contact est définie par un temps de contact. Nous définissons la zone de contact d'un nœud par la région où ce nœud peut éventuellement être en contact avec d'autres nœuds, par exemple, le cercle en pointillés dans la Figure 3.8. Depuis, les nœuds ne peuvent pas communiquer, sauf s'ils sont dans la zone de contact. Nous définissons alors la phase de découverte comme un processus qui permet à un nœud de détecter un contact, qui est, la présence d'un élément mobile dans sa portée radio. D'autre part, nous définissons le transfert de données comme l'échange de messages entre les nœuds qui sont en contact.

Notons bien que cette définition de « transfert de données » ne couvre que les transmissions à un seul saut, ce qui implique que deux ou plusieurs nœuds, où au moins un nœud est mobile. Nous définissons également le temps de contact résiduel comme une durée de temps qui est effectivement utilisée pour le transfert de données au cours d'un contact. Le temps de contact résiduel est généralement plus court que le temps de contact, étant donné qu'un nœud doit détecter la présence d'un élément mobile avant de commencer l'échange de messages. Enfin, nous indiquons que le processus de routage des données envoyées vers l'élément mobile, est la sélection d'un chemin d'accès à la destination souhaitée. Sur la base de cette discussion, trois phases principales associées à la collecte de données dans un RSCS-EM ont émergé: la phase de découverte, de transfert de données, et de routage pour les éléments mobiles [11].

Chaque phase a ses propres problèmes et exigences, que nous allons brièvement voir dans les prochaines sections.

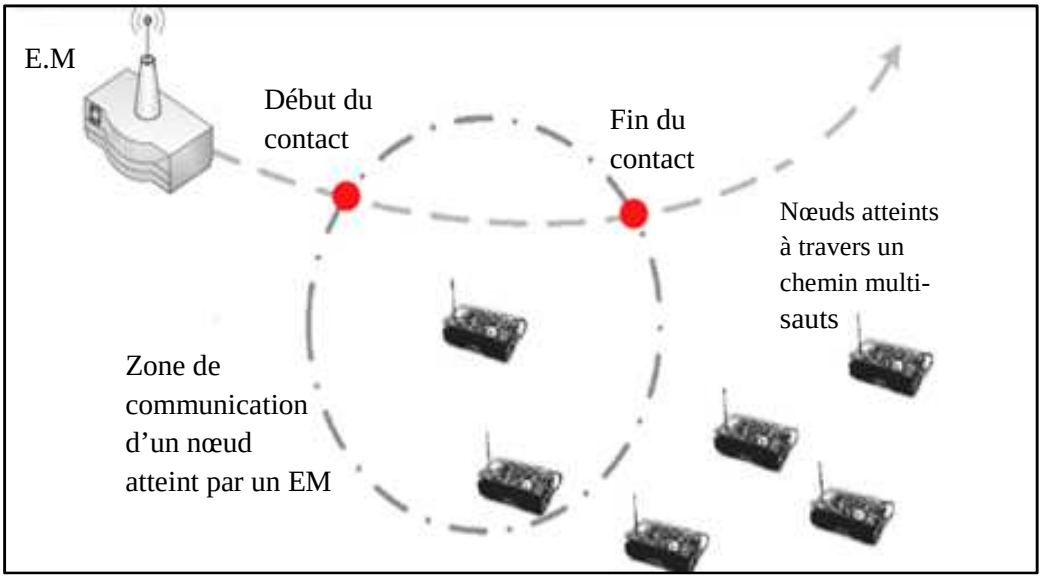


Figure 3.8 – Scénarios de collecte de données dans un RCSF-EM

Cette figure résume les différents aspects de collecte de données dans un RCSF-EM et illustre une taxonomie des approches utilisées dans les différentes phases. Dans ce qui suit, nous allons étudier avec soin ces aspects.

3.8.1 Phase de Découverte

C'est la première étape de collecte de données dans un RCSF-EM, car la présence de l'élément mobile dans la zone de contact est généralement inconnue aux capteurs. Le but du protocole de découverte est de détecter des contacts dès qu'ils y en a, et avec une faible dépense énergétique.
En d'autres termes, la découverte doit maximiser le nombre de contacts détectés, et également le temps de contact résiduel, tout en minimisant la consommation d'énergie.

3.8.2 Phase de Transfert de données

Cette phase vient immédiatement après la phase découverte. Le but du protocole de transfert de données est de tirer le meilleur parti du temps de contact résiduel. En d'autres termes il faudrait minimiser la perte de messages (messages transférés avec succès durant l'intervalle de contact) tout en minimisant la consommation d'énergie.

3.8.3 Phase de Routage de l'élément mobile

Cette phase est en fait possible que lorsque la densité du réseau est suffisante pour permettre (même partiellement) de trouver des chemins multi-sauts. Cela est vrai pour les RCSFs-EMs denses, où le routage de l'élément mobile est toujours possible. Effectivement, cela peut arriver même avec un RCSF-EM dispersé, où les nœuds peuvent organiser des Clusters déconnectés. Dans ce cas, le routage n'est possible que lorsque l'EM est en contact avec au moins un ClusterHead. Cependant, certains nœuds peuvent être élus comme ponts et agissent comme des passerelles entre les ClusterHeads et l'élément mobile.

Dans les deux cas, l'objectif de routage est de trouver le meilleur chemin multi-sauts, tant en termes de taux de livraison et de consommation d'énergie soit vers l'élément mobile ou bien vers le nœud, qui peut être en contact avec l'EM [11].

3.9 Approches de collecte de données dans un RCSF-EM

La figure 3.9 présente les différents aspects de la collecte de données dans un réseau de capteurs sans fil à élément mobile (RCSF-EM) et illustre une taxonomie des approches utilisées dans les différentes phases.

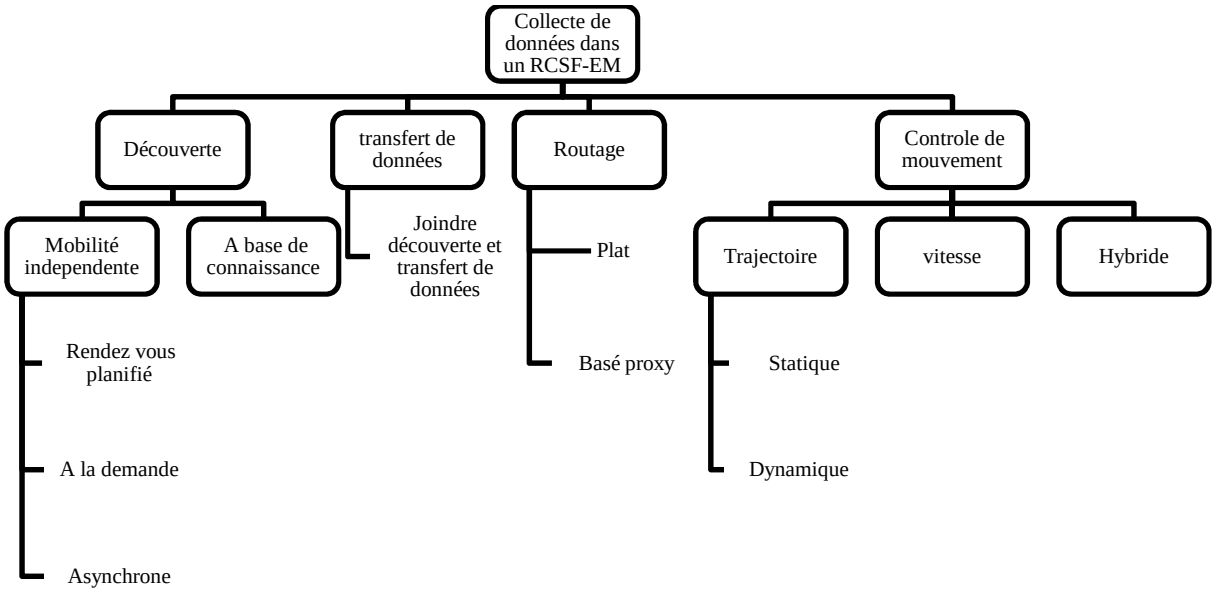


Figure 3.9 – Taxonomie des approches de collecte de données

Dans cette section, nous analysons et présentons les différentes approches de collecte en détail.

3.9.1 Découverte

Cette approche permet aux nœuds de détecter la présence de l'élément mobile pendant qu'il est dans la zone de contact. Par conséquent, la communication n'est possible que lors des contacts. L'approche de découverte permet de détecter correctement la présence de l'élément mobile mais doit aussi s'exécuter rapidement afin de permettre d'exploiter pleinement le temps de contact.

En effet, les contacts doivent être détectés, même quand ils sont très fréquents. Afin de réduire la consommation d'énergie due à la découverte, deux approches complémentaires peuvent être utilisées. Premièrement, il est possible de concevoir des protocoles de mobilité indépendante à faible puissance, qui peuvent détecter l'élément mobile indépendamment de leur modèle de mobilité. Deuxièmement, il est également possible d'exploiter une certaine connaissance de la mobilité des nœuds, de telle sorte que les capteurs soient actifs uniquement lorsque l'élément mobile est en contact.

3.9.2 Transfert de données

Après avoir détecté la présence de l'élément mobile, un nœud capteur émetteur peut procéder au transfert réel des données en utilisant un protocole de collecte. Nous voulons dire par transfert de données « processus de communication entre élément mobile et ses voisins à un-saut ». Comme conséquence, le protocole de transfert de données doit tenir compte des problèmes résultants de la mobilité. En fait, le processus de communication est affecté non seulement par les conditions du canal, mais aussi par la distance entre la source et le récepteur, qui change avec le temps en fonction de sa vitesse.

Quelques approches dans la littérature ont caractérisé la mobilité qui affecte les communications entre un élément mobile et les nœuds capteurs statiques. Des expériences effectuées par [135] ont montré qu'un élément mobile robotisé se déplace lentement entre 30-150 cm/s et collecte une quantité de données qui est indépendante de sa vitesse. La situation devient très différente, cependant, quand l'élément mobile peut se déplacer à des vitesses plus élevées. Cela se produit dans les scénarios urbains comme celui considéré dans [177], où un élément mobile se déplace le long d'une rue afin de collecter les données provenant des capteurs statiques, comme nous pouvons le voir sur la figure 3.10.

Le protocole de transfert de données doit tenir compte également du mécanisme du Handover, problème très connu lors du transfert des données avec élément mobile. En fonction de la quantité de données stockées au niveau du nœud capteur qui désire transmettre des données au *Sink* à un instant donné, il arrive que le *Sink* quitte la portée radio du nœud capteur sans récupérer toutes les données stockées; cela peut être interprété comme une collecte de données incomplète et donc peu fiable. L'idéal serait de fournir une solution innovante capable de garantir une collecte de données efficace pour chaque passage du *Sink* à travers le réseau. Cette solution initie un (Handover) transparent permettant à un nœud de percevoir les changements de la qualité d'une liaison (QL :quality link, qui peut être RSSI, LQI :link quality indicator ou les deux à la fois).

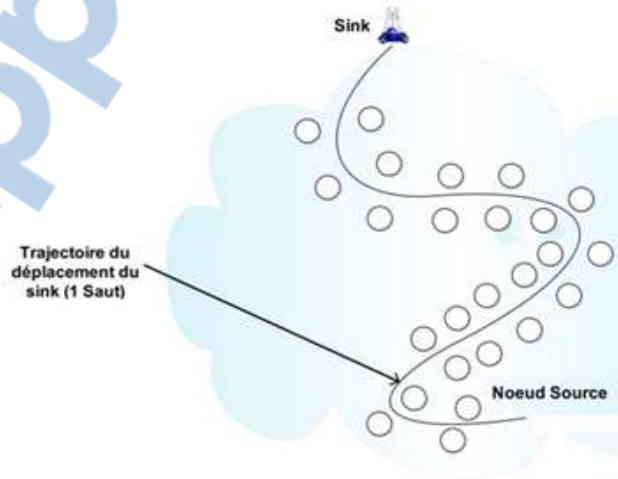


Figure 3.10 – Trajectoire de déplacement du Sink (communication à un-saut)

3.9.3 Routage pour éléments mobiles

Dans cette section, nous discutons les approches pour le routage des messages vers l'élément mobile, en mettant l'accent sur le collecteur de données mobile (que nous avons déjà vu en détail dans la section 3.5.2). Cela implique que le réseau est suffisamment dense pour adopter soit un paradigme de communication Ad-hoc multi-sauts total ou partiel. Nous supposons que l'élément mobile n'est pas contrôlable, de telle sorte que le protocole de routage s'adapte au mouvement de l'élément mobile. Il existe deux grandes classes de techniques de routage pour un élément mobile incontrôlable, à savoir, le routage plat et le routage basé sur le proxy [178]. Dans les deux cas, le chemin de routage de l'élément mobile est adaptatif, calculé et mis à jour, de sorte qu'il puisse être atteint en parcourant le réseau.
Le routage plat est caractérisé par le fait que tous les nœuds se comportent de la même façon et, par conséquent, il n'y a pas de capteurs avec des rôles particuliers. Le routage basé sur le proxy, au contraire, procède à l'élection d'un nombre de proxy ou de passerelles parmi les nœuds capteurs. Les communications se déroulent en multi-sauts via les ponts proxy entre les capteurs statiques et l'élément mobile (voir figure 3.11).

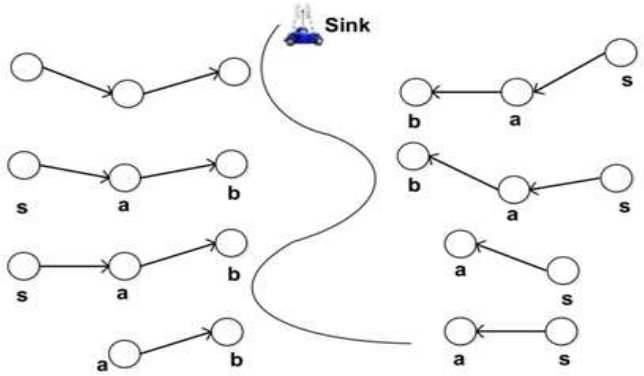


Figure 3.11 – Trajectoire de déplacement du Sink (ponts Proxy)

3.9.4 Contrôle de mouvement

La mobilité des nœuds peut être contrôlée ou non. Lorsque la mobilité n'est pas contrôlable, les nœuds capteurs peuvent être en conformité avec la façon dont l'EM se déplace dans le réseau. De toute évidence, le modèle de mobilité de l'EM a un impact significatif sur le schéma optimal de collecte de données. Dans [147], les différentes stratégies de collecte de données ont été considérées: passive (initiée par l'élément mobile), multi-sauts partiel, et multi-sauts total. Les schémas de collecte ont été évalués par simulation pour les différentes variantes déterministes et aléatoires des modèles de mobilité de l'élément mobile. Les résultats obtenus montrent que, lorsque la latence n'est pas critique, l'option la plus efficace en termes de consommation d'énergie est donnée par l'élément mobile traversant l'ensemble du réseau. Sinon, une trajectoire fixe avec collecte de données multi-sauts partielle peut atteindre une faible latence au détriment d'une meilleure consommation d'énergie et perte de message. D'autre part, lorsque la mobilité est contrôlable, les mouvements de l'élément mobile peuvent être établis de manière à atteindre les objectifs spécifiques et optimiser les

paramètres de performance donnés. Les différents problèmes du contrôle coordonné de plusieurs pairs mobiles ont été examinés dans [179], où un squelette était formé afin de maintenir la connectivité parmi un groupe de nœuds, ainsi que pour obtenir une formation spécifique de cibles. Nous pouvons voir clairement que l'élément mobile contrôlé donne plus de souplesse et de flexibilité à la conception d'un système de collecte de données.

3.10 Conclusion

Une description de quelques travaux qui traitent le processus de collecte de données avec élément mobile a été présentée. Ensuite nous avons décrit les différentes phases du processus de collecte de données ainsi que ses principaux enjeux. Nous avons vu à travers ce chapitre les différentes techniques de collecte de données ainsi qu'une classification détaillée des collecteurs de données mobiles (CDMs) dans un RCSF avec différents types d'éléments mobiles. Ainsi la problématique qui s'est posée à travers cette classification est le choix du meilleur élément à déplacer dans le réseau pour réaliser ou assurer une collecte de données fiable et efficace, « Est ce un CDM de type *Sink* mobile ou bien un CDM de type nœud relais mobile?» ceci fera l'objet d'étude du chapitre suivant. Notre objectif à travers le processus de collecte est de satisfaire certaines exigences des RCSFs. Parmi ces exigences : diminuer la consommation d'énergie, éviter la perte de données, assurer la sécurité et la fiabilité des données et bien sûr étendre la durée de vie du réseau. Cependant, un élément mobile dans le réseau peut poser à son tour quelques problématiques que nous allons voir dans les prochains chapitres tels que « comment choisir le meilleur modèle de mobilité selon lequel l'élément mobile devra se déplacer dans le réseau ? » et « comment optimiser ses déplacements ? » afin de gagner en latence, en consommation d'énergie ainsi qu'en qualité de service. Les réponses à ces questions constituent l'essentiel de nos contributions dans cette thèse.

Chapitre 4

Quel élément mobile à déplacer dans un RCSF

Sommaire

4.1 Introduction

Malgré les nombreuses applications des RCSFs, ces réseaux présentent plusieurs contraintes en termes d'énergie, de calcul et de bande passante. Le principal objectif de tels réseaux est d'assurer les communications et la collecte de données entre les capteurs sans fil et le *Sink* tout en essayant de prolonger la durée de vie du réseau et d'éviter les problèmes des goulots d'étranglement résultant des approches statiques qui peuvent sévèrement dégrader les performances du réseau. Plusieurs raisons expliquent l'intérêt de l'utilisation des éléments mobiles par rapport aux nœuds puits statiques, notamment l'amélioration des performances réseau.

C'est pour ces raisons que la mobilité dans les RCSFs s'avère un sujet d'actualité d'une grande importance et un centre d'intérêt d'un grand nombre de chercheurs. Les méthodes proposées jusqu'à ce jour, selon la nature et le nombre d'éléments mobiles employés [11,14,147,160,163,168], peuvent être classées en deux catégories : « les méthodes basées sur la mobilité du *Sink* et les méthodes basées sur la mobilité des nœuds relais.

Ne sachant pas laquelle de ces deux approches satisfaisait le mieux nos exigences, nous avons commencé notre travail par l'étude de ces dernières.

Dans ce chapitre, nous présentons quelques travaux réalisés qui correspondent à cette problématique, nous détaillerons notre contribution au niveau implémentation en adaptant les différentes couches protocolaires. Ensuite, nous décrirons les scénarios réalisés sous l'environnement de simulation choisi, à savoir OMNeT++ et les frameworks de mobilité MiXiM et INET [180,181]. Ces scénarios représentent deux instances de collecte de données, à savoir : un RCSF avec un *Sink* mobile et un RCSF avec un nœud relais mobile. Dans le scénario relatif au nœud relais, nous avons augmenté le nombre de relais mobiles à deux et à quatre. Enfin nous conclurons ce chapitre par une évaluation des résultats de simulation obtenus en déduisant lequel de ces scénarios répond le mieux à nos exigences (minimisation de la consommation d'énergie des nœuds capteurs et de la perte de messages). Cette contribution [3,14,15] est une contribution d'implémentation (logiciel) et un bon début pour étudier la mobilité sous différents angles.

4.2 Travaux réalisés

Les applications sensibles aux délais, telles que les interventions d'urgence et de surveillance d'environnements hostiles constituent des challenges difficiles pour les protocoles de routage dans les RCSFs. Un protocole de routage doit fournir des techniques rapides et fiables pour la propagation des données.

De nombreux travaux de recherche ont proposé une solution au problème du déploiement des collecteurs de données mobiles afin d'alléger la charge de trafic élevée dans le voisinage du *Sink* causé par les approches statiques. Dans [12,14],les auteurs ont proposé un protocole CDM / PEQ qui utilise les collecteurs de données mobiles (*CDM*) et qui diffusent périodiquement des paquets beacons[1]. Les nœuds capteurs qui reçoivent un beacon rejoignent le groupe (*Cluster CDM*) et mettent à jour leurs informations de routage afin de relayer les paquets de données au CDM. Les nœuds capteurs utilisent la force du signal du beacon afin

[1]Un Beacon représente une balise

d'établir une route de reconfiguration simple mais efficace (Handoff). Plusieurs expérimentations et simulations ont été menées et les résultats obtenus confirment que l'introduction des collecteurs de données mobiles dans les RCSFs réduit le goulot d'étranglement au niveau des nœuds proches du *Sink*. Les avantages recensés dans cette étude nous ont encouragé à utiliser les collecteurs de données mobiles, cependant dans notre étude nous nous sommes intéressé à deux types de CDMs (nœud relais et *Sink*).

Les auteurs dans [85] emploient la diffusion dirigée comme protocole de routage. Ainsi, la solution proposée est une approche basée sur le *Sink*. Le *Sink* diffuse un message d'intérêt pour les nœuds capteurs voisins en se déplaçant sur une ligne droite. Les nœuds capteurs reçoivent le message d'intérêt et le transmettent à leurs propres voisins. Chaque nœud capteur commence à transmettre les données au *Sink* mobile lorsqu'un événement se produit. Le *Sink* mobile se trouve hors de portée lors de la transmission d'événement et donc les paquets seront perdus. La solution utilise des accusés de réception pour s'assurer que le *Sink* a reçu le paquet avec succès, et un nœud capteur transmet d'autres paquets seulement après qu'il ait reçu un accusé de réception du *Sink*.

Tous les travaux cités formulent le problème de la collecte de données comme étant un problème de sélection de chemins optimaux avec différentes hypothèses et contraintes. Une méthode différente employant les nœuds mobiles intermédiaires est d'abord proposée pour les RCSFs dans [133]. Ils ont proposé d'utiliser un mouvement aléatoire « Data Mules» pour la collecte de données. Data Mules dans le domaine des capteurs sont utilisés comme agents transmetteurs. L'idée consiste à économiser l'énergie en utilisant un routage à un seul saut (du capteur au "Mule" directement) au lieu d'un routage multi-sauts plus coûteux (du capteur vers le *Sink*). L'approche à base de *Sink* "Mule" fournit toutes les données collectées au *Sink*.

Les auteurs dans [182] se sont intéressées à la performance d'un réseau dense avec un nœud relais mobile et ont montré que la durée de vie des nœuds dans le réseau augmentait. La communication impliquant les nœuds relais dans toutes les approches considèrent deux scénarios : un seul capteur à un relais et un relais à un *Sink*. La communication de relais à relais qui pourrait être potentiellement utile, est négligée. D'après les auteurs, aucun travail n'a examiné les avantages de laisser les nœuds de collecte de données communiquer pour former un réseau. Une idée similaire à leur contribution est présentée dans [183]. Cependant, les nœuds relais dans [183] ne peuvent pas obtenir directement des données à partir des capteurs. Mais, ils et se déplacent pour faciliter la communication entre les nœuds statiques d'agrégation de données et les différents nœuds Clusters. Ils ont été confrontés à une situation dynamique lorsque tous les collecteurs de données sont mobiles et considèrent conjointement les problèmes des données collectées (à partir des capteurs), et des données acheminées (parmi les nœuds relais mobiles).

Une première lecture de cet état de l'art présenté ne nous permet pas de répondre clairement à la question que nous nous sommes posés dans ce travail. Cette question concerne *le choix du meilleur élément mobile à déplacer dans un RCSF*. La mobilité est une nouvelle thématique abordée depuis peu dans les RCSFs et donc il n'existe pas beaucoup de travaux la concernant. Nous avons jugé utile de recourir à la simulation en implémentant les scénarios répondant à nos propres besoins.

Ainsi, cette contribution est considérée comme une validation par simulation. En effet, il fallait faire des tests de simulation à chaque fois qu'on se posait une question fondamentale liée à l'élément mobile et au fur et à mesure que le travail avançait car il n'existe pas de modèle prédéfini qu'on aurait pu modifier ou améliorer.
Afin de mener à bien les simulations des différents scénarios de mobilité réalisés, nous avons été obligés d'adapter plusieurs paramètres de chaque nœud capteur à notre approche.

4.3 Adaptation au niveau des couches protocolaires

4.3.1 Couche réseau

Pour réaliser ce présent travail, nous avons utilisé deux algorithmes de routage propres à MiXiM [181], le « *Flooding*[2] » pour les scénarios des RCSFs mobile et le « *WiseRoute* » pour le dernier scénario du RCSF stationnaire.

L'algorithme du *« Flooding »* de MiXiM : consiste à implémenter le Flooding classique, avec quelques améliorations qui permettent d'éviter les boucles dans le réseau d'une manière intelligente et de rendre possible une communication unicast qui n'est pas prévue dans cet outil. C'est un algorithme simple et s'exécute comme suit :

Un nœud capteur qui génère un message le diffuse en broadcast[3] à tous ses proches voisins (voisins à un saut). Lorsqu'un nœud reçoit un message, il teste si c'est lui le destinataire et le consomme sans le rediffuser. Sinon, il va scruter la liste des messages déjà envoyés. S'il ne trouve pas ce message dans la liste, il le rediffuse à ses voisins, ou alors il supprime le message.

Algorithme 1«Flooding »

```
1 : début
2 :pour chaque nœud i qui génère un message
3 :le nœud diffuse le message en broadcast à tous ses voisins à un saut;
4 :si (nœud qui a reçu le message=destinataire)
5 :   alors il le consomme sans le rediffuser;
6 :   sinon
7 :    il scrute la liste des messages déjà envoyés;
8 :     si(ce message n'existe pas dans la liste)
9 :     alors il le rediffuse à ses voisins;
10:     sinon
11:      il supprime le message;
12:     finsi
13:finsi
14:finpour
15: fin
```

L'Algorithme *« WiseRoute »:*est un protocole de routage simple « *Convergecast* » utilisé dans MIXIM. Dans cet algorithme, une génération d'arbres commence par la génération d'un nœud racine comme un paquet de route inondé. Chaque nœud qui reçoit un paquet de route inondé,

[2]Flooding définit une inondation

[3]Brodcast définit une diffusion d'un émetteur vers plusieurs récepteurs

contrôle la valeur du RSSI[4] (Received Signal Strength Indication) qu'il a reçu. Si la valeur du RSSI est supérieure à un seuil prédéfini, le nœud émetteur le sélectionne comme un nœud parents pour la transmission des paquets de données. Ce paquet de route inondé sera alors transmis.
Si un nœud reçoit un paquet dupliqué, il le rejette. Ce dernier sera rejeté, même si le paquet reçu récemment possède un RSSI supérieur à celui du précédent qui a été utilisé pour sélectionner le nœud parents. Ainsi, tout paquet qui a été reçu en premier sera utilisé pour sélectionner le nœud parent et non pas le paquet avec la meilleure valeur de RSSI.

Nous avons utilisé cet algorithme pour assurer l'acheminement multi-sauts des paquets émis par un nœud source jusqu'au *Sink* dans le scénario correspondant au RCSF stationnaire.

Algorithme 2 «WiseRoute »

```
1:début
2: génération d'un nœud racine comme un paquet de route inondé ;
3: Pour chaque nœud qui reçoit un paquet de route inondé ;  faire
4: contrôle la valeur du RSSI qu'il a reçu ;
5: Si la valeur du RSSI>seuil prédéfini     alors
6: le nœud émetteur le sélectionnera comme un nœud parents pour la transmission
des paquets de données (Ce paquet de route inondé sera alors transmis) ;
7: Si un nœud reçoit un paquet dupliqué alors
8:    ce dernier sera rejeté ;
9: Pour tout paquet qui a été reçu en premier sera utilisé pour sélectionner le
nœud parent et non pas le paquet avec la meilleure valeur de RSSI,
10:     finpour
11:   finsi
12:finsi
13:finpour
14:fin
```

4.3.2 Couche application

Pour adapter cette couche à notre travail nous avons ajouté plusieurs fonctions

- La fonction ***« PeriodicHello »*** : permet au *Sink* d'émettre des paquets « *Hello* » dans le réseau afin d'informer les nœuds capteurs de son arrivée.
- La fonction ***« ActiveSendData »*** : sert à synchroniser l'envoi des messages générés par les nœuds capteurs avec l'arrivée du *Sink.*
- La fonction ***« Throughput »*** : calcule le nombre de messages perdus.
- La fonction ***«Aggregate »*** : permet aux nœuds relais d'agréger les messages reçus.

4.3.3 Plan de mobilité

Pour représenter la trajectoire empruntée par les CDMs, nous avons développé quatre (04) modèles de mobilité appropriés aux différents scénarios simulés (que nous allons évoquer dans ce qui suit).
Ces modèles s'inspirent du modèle de mobilité « *TractorMobility* » implémenté sous Inet.

[4] RSSI fournit une indication sur l'intensité du signal reçu

Cet algorithme lit la position initiale de l'élément mobile sur l'axe des «*X*» et l'axe des «*Y*» ainsi que l'extrémité où doit s'arrêter cet élément (destination finale).
L'élément mobile se déplace dans le réseau en calculant à chaque fois sa prochaine position à partir de sa position initiale. Tant que la prochaine position n'est pas l'extrémité d'arrêt, il continue son parcours jusqu'à ce qu'il atteigne la destination finale.
Un extrait de l'algorithme correspondant au modèle de mobilité « *TractorMobility* » est représenté dans ce qui suit et est implémenté sous Inet :

Algorithme 3 « Tractor Mobility »

```
1 : début
2 : lire la position initiale(x,y)de l'EM ;
3 : écrire la position initiale de l'EM en fonction de sa position sur les deux
axes « X » et « Y » ;
4 : lire l'extrémité d'arrêt de l'EM ;
5 : Calculer la prochaine position de l'EM en fonction de sa position initiale ;
6 :tant que  (prochaine Position ≠ l'extrémité d'arrêt de l'EM)
7 :début
8 :calculer prochaine position et prochain temps ;
9:   fin
10:fin tantque
11: fin
```

Remarque : Nous avons effectué quelques modifications sur cet algorithme de manière à obtenir quatre nouveaux algorithmes de mouvement pour chacun des scénarios réalisés : un *Sink* mobile, un relais mobile, deux relais mobiles et quatre relais mobiles.

Deux approches de collecte sont considérées dans ce travail: une approche avec CDM de type *Sink* mobile et une approche avec CDM de type nœud relais mobile.

Dans la première approche, nous avons utilisé un seul *Sink* contrairement à l'approche avec nœud relais où nous avons testé un, deux et quatre relais mobiles.

4.3.3.1 Modèle de mobilité « Sink mobile »

Cet algorithme définit la trajectoire que doit emprunter le *Sink* mobile en s'inspirant du modèle ***Tractor Mobility***. La modification apportée à cet algorithme par rapport au modèle ***Tractor Mobilit***y, est que nous devons tester si cet élément est un nœud *Sink* ou bien un nœud *relais*. Le cycle doit être également incrémenté et un temps de pause doit être considéré.
Cet algorithme s'exécute comme suit :
Lorsque le *Sink* atteint la deuxième extrémité de la grille, et que la valeur de la prochaine position est égale à celle de la valeur de ***« l'extrémité d'arrêt »*** ; le *Sink* effectue un demi tour jusqu'à ce qu'il rejoigne sa position initiale (ce schéma est illustré sur la figure 4.1).
A ce moment-là, nous considérons que le *Sink* a fait un cycle (round), et que la valeur de la variable ***« cycle »*** s'incrémente de 1.
L'algorithme correspondant à ce modèle est représenté dans ce qui suit :

Algorithme 4 « Modèle de mobilité du ***Sink »***

```
1 : début
2 :lire la vitesse du Sink
3 : lire la position initiale du Sink (x0,y0)
4 : lire la position finale du Sink(x1,y1)
5 : Si(modèle du Sink)alors
6 :  calculer la nouvelle position à partir de la position initiale et de la
                    vitesse
7:   si(prochaine Position=l'extrémité d'arrêt)alors
8:Faire une pause
9:incrémenter le cycle de 1
10:finsi
11:finsi
12: fin
```

4.3.3.2 Modèle de mobilité « Relais mobile »

Ce modèle introduit un nouveau type de CDM qui correspond au nœud relais mobile. Ce nœud occupe une position initiale dans le réseau, et lors du lancement de l'exécution de la simulation, il commence à se déplacer en empruntant une trajectoire linéaire (déplacement sur ligne droite).

Etant donné que ce nœud est un collecteur intermédiaire, à chaque fois qu'il rejoint sa position, il se déplace en direction du *Sink* (récepteur final dans le réseau d'acquisition).

L'algorithme de ce modèle est représenté dans ce qui suit :

Algorithme 5 «Modèle de mobilité du ***nœud relais*** »

```
1 :début
2 : lire la vitesse de l'EM
3 : lirela position initiale du nœud relais(x1,y1)
4 : lire la position initiale du Sink(x0,y0)
5 : lire la position finale du nœud relais(x2,y2)
6 : si(modèle du nœud relais)            alors
7 :  calculer la prochaine position
8 :     si (prochaine Position=l'extrémité d'arrêt) alors
9:    rejoindre le Sink
10:    finsi
11: finsi
12: fin
```

Remarque* :** Les modèles de mobilité, relatifs à ***deux et à quatre nœuds relais possèdent le même principe que le modèle de mobilité d'un relais mobile, mis à part la valeur de la variable *« cycle »* qui ne s'incrémente que lorsque tous les nœuds relais atteignent le *Sink*.

Après l'exécution des simulations de tous les scénarios proposés dans ce travail, nous avons obtenu des résultats que nous allons analyser et évaluer dans les prochaines sections.

4.4 Simulation et résultats préliminaires

4.4.1 Environnement de simulation

L'apparition des RCSFs a ouvert de nombreux aspects pour les concepteurs des réseaux. Traditionnellement, les trois principales techniques pour analyser la performance des réseaux filaires ou sans fil sont les méthodes d'analyse, les simulations informatiques et les mesures

physiques. Toutefois, en raison de nombreuses contraintes imposées par les RCSFs, comme la limitation d'énergie, la collaboration décentralisée et la tolérance aux pannes, les algorithmes de RCSFs ont tendance à être plus complexes et défient souvent les méthodes d'analyse qui se sont révélées assez efficaces pour les réseaux traditionnels.
Divers environnements de simulation de réseaux existent pour tester les algorithmes dans les RCSFs [19], incluant OMNET++ [184], GloMoSim, OPNET, Java-Sim, SensorSim, NS2,Castalia [185],et bien d'autres.
Nous avons choisi la simulation informatique qui est une reproduction d'un phénomène (naturel, physique, informatique) sur ordinateur. Elle aboutit à la description du résultat de ce phénomène, comme s'il s'était réellement déroulé. Cette représentation peut être une série de données, une image, une vidéo ou même le fonctionnement d'un RCSF.
Les simulations reposent sur la mise en œuvre de *modèles théoriques*, la mise en œuvre d'infrastructures et de protocoles de RCSF nécessite une phase de tests avant sa mise en place afin de s'assurer du bon fonctionnement du réseau. L'expérimentation réelle sur le terrain s'avère très coûteuse. Alors, avant la mise en œuvre d'une telle structure, le bon fonctionnement du réseau doit être assuré. Cette étape est incontournable pour l'évaluation des modèles d'application ou des protocoles de communication ainsi que la consommation d'énergie, qui est un élément crucial dans les RCSFs. De plus, la simulation offre un gain considérable en temps, une flexibilité permettant la variation des paramètres et une meilleure visualisation des résultats.
Cette section sera consacrée à l'environnement de simulation (OMNeT++/MiXiM) en justifiant notre choix. Nous avons opté pour le simulateur OMNeT++ car il propose des modules permettant de simuler la mobilité des nœuds capteurs et offre également une meilleure flexibilité du code source.

4.4.1.1 OMNeT++ (Objective ModularNetwork Test-bed in C++) : est un simulateur dédié aux RCSFs et il est basé sur la simulation à événement discret. OMNET++ est un projet open source dont le développement a vu le jour en 1992 par Andras Vargas à l'université de Budapest, il possède une hiérarchie de modules séparés en deux catégories simple et composée. Les modules simples sont programmés en C++, tandis que les modules composés, constitués d'unou de plusieurs modules simples, sont programmés dans un langage de haut-niveau (NED) [184].
Pour la communication entre les nœuds, nous échangeons des messages qui représentent les paquets du réseau. OMNeT++ propose aux utilisateurs la possibilité d'étudier l'effet du passage à l'échelle, l'architecture du nœud, l'efficacité énergétique, l'architecture de communication, l'architecture du système, les protocoles, etc.
Une extension de mobilité pour OMNeT est conçue et destinée à simuler les réseaux sans fil et mobiles.
OMNeT++ est un outil open source et semble être le meilleur parmi les solutions open source et freeware, il dispose de plusieurs modules de mobilité tels que « *MobilityFramework* » , « *INET* », « *MiXiM* » , et même une version très récente a vu le jour qui combine INET avec MiXiM dans un seul produit appelé ***« MiXiM-INET-bundlle »***(version produite en Mars 2013).

4.4.1.2 Mobility Framework (MF) : la librairie MF a été développée par une équipe de chercheurs à l'université de Berlin et représente une extension du simulateur OMNeT++. Cette librairie est destinée à soutenir les RCSFs mobiles au niveau du simulateur OMNeT++. En effet, elle permet une bonne manipulation des nœuds mobiles et une gestion des connexions dynamiques pour avoir un modèle de mobilité de réseau sans fil qui fournit des résultats plus réalistes. En outre, la librairie MF prévoit des modules de base qui peuvent être utilisés pour former de nouveaux modules. Avec ce concept, un programmeur peut facilement développer ses propres protocoles avec l'interface nécessaire.

- ***INET :*** **e**st une librairie open source pour la simulation des réseaux informatiques dans l'environnement OMNeT++. Elle contient IPv4, IPv6, TCP, UDP, des protocoles implémentés, et plusieurs modèles d'application. Elle comprend également la couche liaison de données et des modèles de PPP, Ethernet et 802.11. Le routage statique peut être configuré à l'aide d'un réseau auto-configurateur ou, nous pouvons utiliser le protocole de routage mis en œuvre. INET prend en charge les simulations des réseaux ad-hoc ainsi que les réseaux sans fil mobiles.
- ***MiXiM :***est un framework d'OMNeT ++, créé pour modéliser les RCSFs, body area networks (BAN), Ad-hoc networks, vehicular networks, etc..), Il offre des modèles détaillés de propagation d'onde hertzienne, l'évaluation d'interférence, la consommation électrique d'émission-réception de la radio et des protocoles MAC sans fil (par exemple. Zigbee)**.** C'est une fusion de plusieurs frameworks d'OMNeT ++, développés pour supporter des simulations mobiles et sans fil. Aussi, il a une infrastructure de soutien et peut supporter la simulation de réseaux consistants jusqu'à 1000 nœuds.

4.4.2 Description du réseau et des scénarios de collecte

Nous avons considéré un réseau constitué de vingt (20) nœuds capteurs sans fil et d'un élément mobile. Les nœuds sont déployés alternativement sur les colonnes d'une grille de dimension (4*5) intercalées par une colonne vide.

Nous considérons dans ce travail six (05) scénarios: dans les quatre premiers scénarios, nous avons utilisé un RCSF-EM par contre dans le dernier scénario, nous avons utilisé un RCSF stationnaire. Voici les cinq scénarios réalisés dans cette première contribution :

- ***Scénario 1*** *« 1-SM » :* correspond à un RCSF mobile où la collecte de données est assurée par un seul *Sink* mobile (CDM=*Sink* mobile),
- ***Scénario 2*** *«1-RM » :* représente un RCSF mobile dans lequel l'élément mobile est un nœud relais (CDM=Relais mobile), dans ce scénario nous avons opté pour une architecture linéaire.
- ***Scénario 3*** *« 2-RM » :* comporte deux nœuds relais collecteurs de données, où chacun de ces nœuds est chargé de retransmettre les données récupérées par *Sink* (CDM=Relais mobile),
- ***Scénario 4*** *« 4-RM » :* consiste en un RCSF mobile comportant quatre nœuds relais mobiles comme collecteurs intermédiaires (CDM=Relais mobile),
- ***Scénario 5****« RCSF-S » :* correspond à un RCSF stationnaire (pas de CDM).

4.4.3 Scénarios réalisés dans le RCSF mobile *(1-SM, 1-RM, 2-RM, 4-RM)*

Dans ces quatre scénarios, nous considérons les nœuds du réseau de type *« Host 802.15.4 »* dotés d'une batterie simple, celle des nœuds relais est d'une capacité supérieure, cependant la batterie du *Sink* est considérée comme illimitée.

Ces nœuds échangent les informations via une communication mono-saut, régie par le protocole de routage *« Flooding »*.

Les valeurs des paramètres sont les mêmes dans tous les scénarios de simulation et sont présentées dans la table 4.1.

Paramètres	***Valeurs***
Nombre de paquets généré	10
Type d'application	« SensorApplLayer »
Capacité de la batterie	99999 mAh
Voltage de la batterie	3.3 V
Vitesse du CDM	1000/5000mps

Table 4.1– Paramètres de simulation utilisés dans les différents scénarios

4.4.3.1 Scénarios simulés avec des CDMs de type Sink mobile (1-SM) : dans ce scénario, le *Sink* est mobile tandis que les autres nœuds capteurs sont stationnaires. Initialement, pendant le processus de collecte de données le *Sink* est positionné à un endroit éloigné de l'ensemble des nœuds capteurs. Mais une fois la simulation lancée, le *Sink* parcoure les colonnes de la grille selon la trajectoire qui lui a été donnée dans ce modèle. Ainsi il collecte les données capturées par les nœuds se trouvant dans cette grille comme nous pouvons le voir sur la figure 4.1.

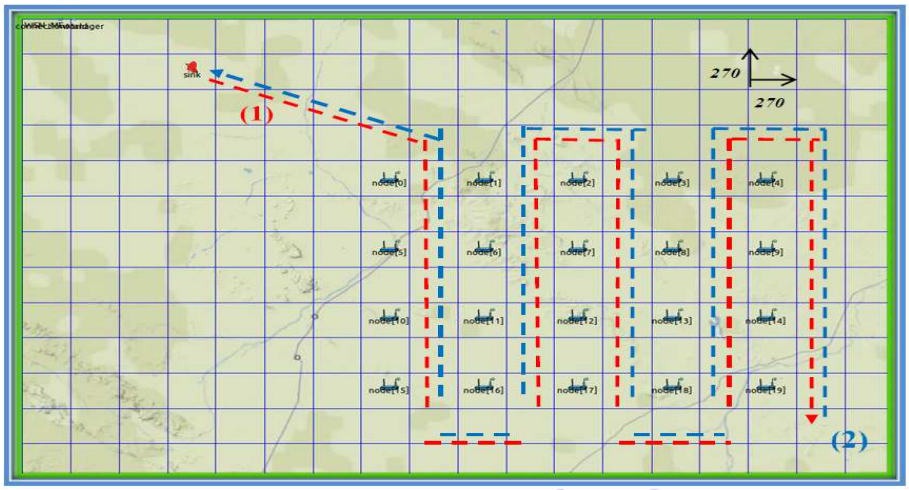


Figure 4.1 – Capture écran du scénario d'un RCSF avec Sink mobile

Une fois que le *Sink* atteigne la frontière droite de la grille selon la ligne en pointillée rouge (1) qui représente le chemin aller du *Sink*, il fait demi-tour et retourne à sa position initiale selon la ligne en pointillée bleu où il fera une pause durant une période déterminée. Après l'écoulement de celle-ci, le *Sink* refait ce même processus (c'est à dire un nouveau round de collecte).

4.4.3.2 Scénarios simulés avec des CDMs de type Relais Mobile : cette approche diffère de la précédente par l'utilisation d'un autre type de CDMs qui est un nœud relais. Ainsi, nous avons choisi de faire bouger un ou plusieurs nœuds relais au lieu du *Sink*. Les valeurs des paramètres du réseau restent toujours les mêmes à l'exception d'une capacité plus élevée en matière de stockage de la batterie des nœuds capteurs.

- ***Un relais mobile (1-RM)***

  Dans ce scénario, le réseau est composé de 19 nœuds capteurs, d'un *Sink* stationnaires et éventuellement d'un seul nœud relais mobile comme l'illustre la figure 4.2.

Figure 4.2 – Capture écran du scénario d'un RCSF avec un seul relais mobile

Le processus de collecte de données se déroule comme suit : dans ce scénario nous avons choisi de varier la topologie du réseau, afin d'étudier aussi son impact sur la mobilité. Donc, nous avons utilisé une architecture linéaire du réseau dans laquelle les nœuds capteurs sont déployés sur une ligne droite. Initialement, le relais est situé prêt du premier nœud capteur se trouvant sur sa droite. Ce dernier commence à se déplacer en allant vers la droite avec une vitesse fixe, en collectant à chaque fois les données capturées par les nœuds capteurs jusqu'à ce qu'il atteigne le dernier nœud du réseau. Ensuite, il revient sur ses pas après avoir récupérer les données de celui-ci .Quand le relais arrive à sa position initiale, il rejoint le *Sink* pour lui transmettre les messages qu'il a collectés, puis refait un nouveau round de collecte.

- ***Deux relais mobiles (2-RM)*** : dans ce scénario, nous avons divisé le réseau en deux parties de collecte. Ce dernier est composé de 18 nœuds capteurs stationnaires, d'un *Sink* stationnaire et de deux relais mobiles. Chacun des nœuds relais assure la collecte des données capturées par les seuls nœuds se trouvant dans sa région.

Ces relais effectuent une visite de collecte de données dans la région où ils opèrent, rassemblent les données ensuite rejoignent le *Sink* afin de les lui retransmettre. La trajectoire empruntée par chacun des nœuds relais est illustrée dans la figure 4.3.

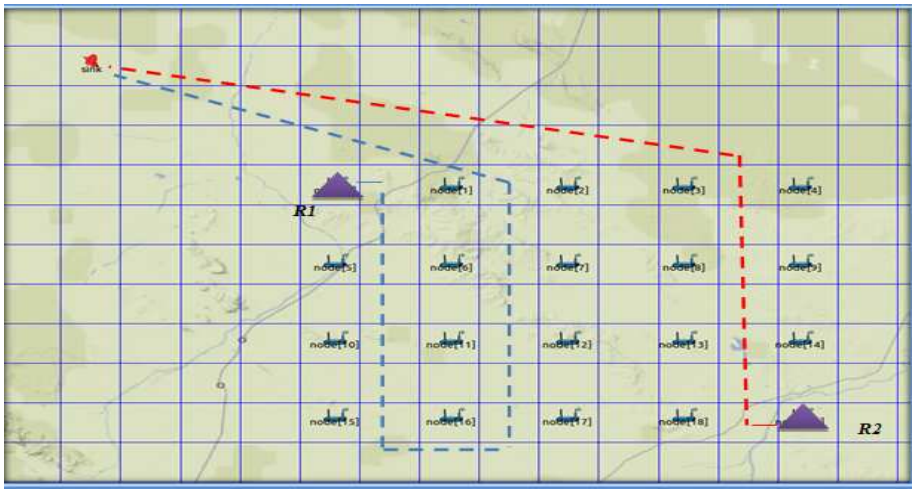


Figure 4.3 – Capture écran du scénario d'un RCSF avec deux relais mobiles

- ***Quatre Relais mobiles (4-RM) :*** dans ce scénario, nous avons augmenté le nombre de nœuds relais jusqu'à quatre, pour mieux étudier l'impact de l'élément mobile dans le réseau. Ainsi nous avons divisé le réseau en quatre parties de collecte. Ce réseau est composé de 16 nœuds capteurs et d'un *Sink* stationnaire, en plus de quatre nœuds relais mobiles. Chacun de ces relais mobiles assure la collecte de données capturées par les nœuds capteurs se trouvant uniquement dans sa trajectoire.

  Cette configuration est illustrée dans la figure 4.4.

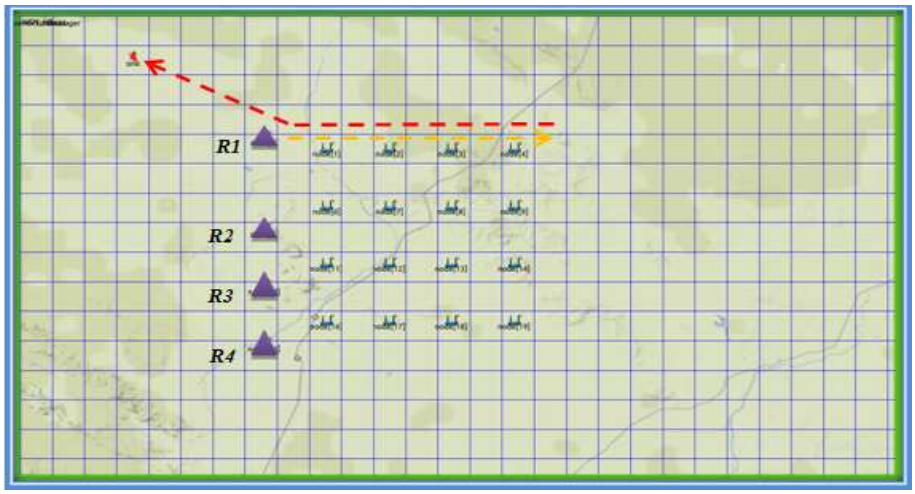


Figure 4.4 – Capture écran du scénario d'un RCSF avec quatre relais mobiles

4.4.4 Scénario réalisé dans le RCSF Stationnaire

Dans le présent travail, nous avons envisagé une seule solution de ce type dans le but de comparer les résultats obtenus de celle-ci avec ceux obtenus dans les scénarios utilisant un RCSF mobile. Le réseau est constitué de 20 nœuds capteurs et d'un seul *Sink* où tous les éléments sont stationnaires.

Contrairement aux solutions basées sur un RCSF mobile, la communication entre les nœuds capteurs et le *Sink* est une communication multi-sauts.
Les paramètres de cette configuration sont les mêmes que ceux présentés dans la table 4.1, sauf qu'au niveau du protocole de routage, nous avons utilisé le protocole ***« WiseRoute »***.
Le processus de collecte de données se déroule comme suit: les nœuds capteurs génèrent aléatoirement des messages de données dont la destination finale est toujours le *Sink*. Les paquets sont ensuite acheminés vers le *Sink* en empruntant le chemin établi par la couche réseau.

4.4.5 Suppositions du réseau étudié

Dans cette partie, nous présentons les principales suppositions utilisées dans les scénarios:

Supposition 1 : l'application de RCSF considérée dans ce travail est une application de surveillance (température, humidité par exemple) où le temps réel n'est pas exigé.

Supposition 2 : les nœuds capteurs et les *CDMs* sont considérés comme homogènes dans un même scénario. Cependant au niveau des composants, nous supposons dans certains cas, que seuls les *CDMs* sont dotés d'un sous-système supplémentaire, celui de la mobilité.

Supposition 3 : le *Sink* est un nœud ayant des capacités suffisantes en termes de calcul, de communication, de mémoire de stockage et d'autonomie d'énergie.

Supposition 4: la zone de déploiement est représentée sous forme d'un espace à deux dimensions 2D (une grille) où les nœuds capteurs sont déployés dans des positions fixes (notons que cette étude peut être étendue à un espace à 3dimensions 3D).

Supposition 5 : l'énergie consommée par chaque élément du réseau est calculée proportionnellement à la distance entre l'émetteur et le récepteur.

Supposition 6 : le modèle de mobilité est non contrôlable (aléatoire) et la trajectoire des éléments mobiles est déterministe (précise).

Supposition 7 : la vitesse de déplacement du *CDM* est constante.

Supposition8 : la communication est multi-sauts pour le scénario avec RCSF stationnaire et mono-saut pour les scénarios avec RCSFs mobiles.

Supposition 9 : les *CDMs* sont soit de type *Sink* mobile ou bien de type nœuds relais mobile.

Supposition 10:la collecte de données se fait d'une manière périodique (voir les détails dans le chapitre3.

Supposition 11: le temps de pause est considéré le même durant toutes les simulations

Supposition 12: dans le scénario du RCSF mobile, le *CDM* doit se déplacer dans l'espace de déploiement jusqu'à atteindre la portée radio d'un nœud capteur désirant

transmettre des données capturées et stockées à son niveau vers le *Sink*. Une fois dans la portée radio, le *Sink* peut communiquer avec le ou les nœuds concernés. En cas de concurrence, la couche MAC intervient pour coordonner les différentes communications.

4.4.6 Métriques de performance

Nous avons pris en considération trois critères d'évaluation afin de pouvoir comparer les résultats obtenus. Les métriques de performance suivantes sont les plus utilisées pour évaluer les protocoles des RCSFs:

- **L'énergie dissipée (Ed):** puisque les nœuds capteurs sont alimentés par des batteries, les protocoles doivent être à économie d'énergie pour maximiser la durée de vie du réseau. Il s'agit d'une mesure de l'ensemble des énergies dissipées par tous les nœuds du réseau par rapport au nombre de nœuds capteurs présents dans le réseau.

$$Ed = \frac{\sum énergies\ dissipées\ par\ tous\ les\ nœuds}{nombre\ total\ de\ noeuds\ capteurs}$$

- **Perte de message (Throughput)**: le taux de perte de paquet est le pourcentage de paquets perdus lors de la transmission de données. Ce taux représente la différence entre les paquets générés et les paquets reçus par la *Sink*. Il s'agit d'un critère de qualité de services d'un réseau et est calculée comme suit :

$$Thr = Nombre\ de\ massages\ générés - nombre\ de\ messages\ reçus\ par\ le\ sink$$

- **Latence (Lat)** : représente le temps nécessaire pour véhiculer un paquet dans un réseau, les paquets émis doivent être correctement reçus par le *Sink* dans les délais impartis; autrement ils deviennent obsolètes. Par exemple, les applications de streaming vidéo temps réel requièrent des garanties strictes de délai de bout en bout, de bande passante et de gigue. La sémantique précise de cette métrique dépend fortement de l'application. La latence est calculée par la formule :

$$Lat = \frac{\sum latences\ des\ messages\ calculées\ par\ tous\ les\ nœuds}{nombre\ de\ messages\ reçus}$$

4.4.7 Evaluation et interprétation des résultats dans les différents scénarios

Après les simulations effectuées sur les scénarios détaillés dans la section précédente (Un *Sink* mobile ***(1-SM)***, un relais mobile ***(1-RM)***, deux relais mobiles ***(2-RM)***, quatre relais mobiles ***(4-RM)*** et RCSF stationnaire ***(RCSF-S)***), les résultats obtenus sont résumés dans la table 4.2.

	Latence(s)	*Perte de messages*	*Consommation d'énergie)(mW/h)*
1-SM	14.06	24	62.038598009923
1-RM	16.42	29	62.038706562718
2-RM	3.79	19	62.038665262718
4-RM	1.5	16	62.037237836023
RCSF-S	1.07	26	62.500000018296

Table 4.2 – Résultats de simulation

- **Energie dissipée**

La figure 4.5 représente l'énergie dissipée dans chacun des cinq scénarios (*1-SM, 1-RM, 2-RM, 4-RM et RCSF-S*) correspondants aux RCSFs-EM et Stationnaires.

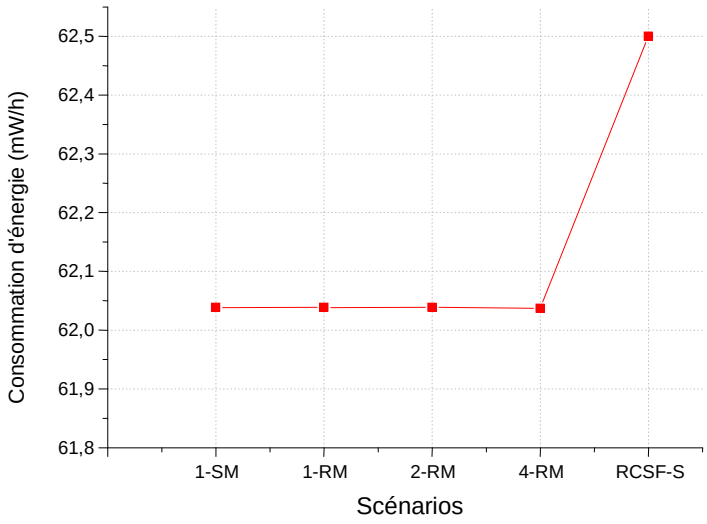


Figure 4.5 – Energie dissipée

Nous remarquons à travers ces résultats que:
L'énergie dissipée par les nœuds capteurs dans les différentes configurations est donnée selon un ordre croissant comme suit:

1. Quatre relais mobiles (4-RM)
2. Un *Sink* mobile (1-SM)
3. Deux relais mobiles (2-RM)
4. Un relais mobile (1-RM)
5. Un RCSF stationnaire (RCSF-S)

Nous constatons à travers cette courbe (figure 4.5) que quel que soit le type d'élément mobile utilisé, il existe un grand écart entre la quantité d'énergie dissipée par le réseau stationnaire et celle consommée par le réseau mobile. Le taux de consommation d'énergie élevé dans le réseau stationnaire est dû à la nature statique du réseau qui exige une communication multi-sauts avant d'arriver au *Sink*, ce qui implique le passage d'un même message par plusieurs nœuds.

De ce fait, les nœuds intermédiaires consomment plus d'énergie pendant la transmission des messages. Ceci augmente l'énergie dissipée dans le réseau par rapport à celle que consomme un RCSF-EM.

Tandis que dans un RCSF-EM, nous remarquons une légère différence d'énergie dissipée par rapport aux quatre scénarios d'RCSF-EM, un nœud relais mobile récupère un message à travers une communication à un seul saut. Ceci préserve encore plus l'énergie du réseau d'une part et d'autre part les *CDMs* de type relais sont supposés être dotés d'une plus grande capacité d'énergie par rapport aux nœuds ordinaires.

Plus nous augmentons le nombre de nœuds relais (*4-RM*), plus l'énergie dissipée diminue et ceci s'explique par la trajectoire empruntée par les nœuds relais (distance, temps de collecte pendant un cycle...).

Cependant le coût élevé des relais nous pousse à nous intéresser à la seconde meilleure solution car l'énergie dissipée dans le scénario avec *CDM* de type *Sink* mobile (*1-SM*) s'avère minimale par rapport aux autres scénarios. Nous expliquons ces résultats par le fait que le *Sink* soit le destinataire final. Tandis que dans un RCSF avec *CDMs* de type relais, le collecteur est le destinataire intermédiaire et par conséquent les nœuds relais consomment plus d'énergie afin de rejoindre le *Sink*.

- **Perte de messages**

La figure 4.6 représente le nombre de messages perdus dans chacun des cinq scénarios (*1-SM, 1-RM, 2-RM, 4-RM et RCSF-S*) correspondants aux RCSFs-EMs et stationnaires.

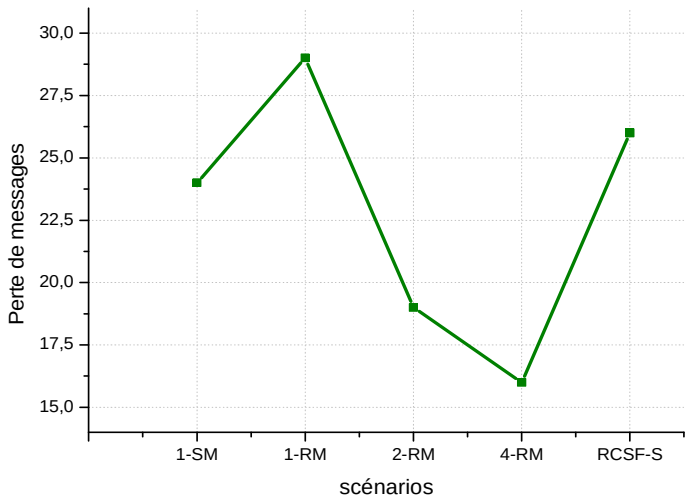


Figure 4.6 – Nombre de messages perdus

Les résultats concernant la perte de messages obtenus et illustrés dans la figure 4.6 permettent la classification suivante et par ordre croissant:

1. 4-RM
2. 2-RM
3. 1-SM
4. RCSF-S
5. 1-RM

Nous observons dans la figure 4.6 qu'il existe une perte considérable de messages dans le scénario (*1-RM*). Ce scénario utilise la solution existante dictée par le simulateur MiXiM. Ce mécanisme considère que tout message généré par le nœud capteur à un nœud collecteur hors de sa portée radio est un message perdu (principe de réception de message instantanée).

Nous allons étudier l'impact de la variation de la vitesse des éléments mobiles notée « *v* », entre 1000 et 5000 mps ainsi que le temps de génération des messages par les nœuds capteurs dans un RCSF-EM noté «*T*», pour l'amélioration des performances du réseau. Les résultats obtenus au bout de deux cycles montrent que plus la vitesse du collecteur augmente plus la perte est importante (voir figure 4.7).

Nous remarquons sur cet histogramme que le scénario qui offre le moins de perte de message est celui qui correspond à une vitesse plus élevée, conditionnée par un intervalle de temps plus long. Ce cas est spécifique à la méthode de génération de message de MiXiM et ne peut être généralisé.

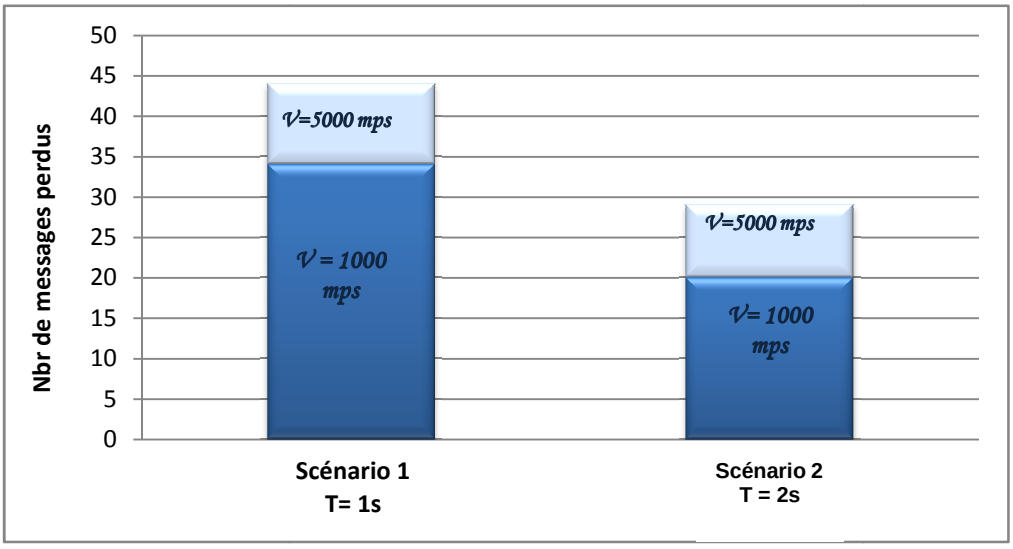


Figure 4.7 – Nombre de messages perdus en variant la vitesse et le temps dans le scénario du *Sink* mobile

La conséquence est une perte de messages dans le réseau, ce qui est probablement dû au temps résiduel assez court, c'est-à-dire que le *Sink* ressort de la portée radio du nœud avant l'achèvement du transfert de message.

Pour pallier à ce problème, nous proposons une solution basée sur le scénario suivant : L'EM se déplace selon une trajectoire donnée, lorsqu'il détecte un élément statique dans sa portée radio il commence à lui envoyer un message « Hello ». Dès la réception du message, l'élément statique à son tour transmet à l'EM les messages disponibles à son niveau.

Remarque : Dans ce travail, nous avons réalisé une partie de la solution que nous venons de proposer. Cette solution consiste à minimiser la perte de messages au niveau des nœuds relais mobiles (collecteurs intermédiaires).
Pour cela:

- Nous avons modifié la structure du message en ajoutant d'autres champs nécessaires à la collecte de données avec relais mobiles.
- Nous avons doté chaque nœud relais d'un buffer. A chaque fois que le nœud relais reçoit un message, il le stocke dans ce buffer. Dans ce cas là, le nœud devient aggrégateur de message.
- Nous avons également tenté de réduire la consommation d'énergie ainsi que la latence. Pour cela, nous avons procédé à la concaténation des messages reçus par le relais en un seul message. De ce fait, l'élément mobile retransmet au *Sink* un seul et unique message à la fois (ceci est proportionnel à la taille du buffer).Cette solution s'avère plus ou moins économe en termes de :

  Energie : le nœud relais peut stocker un nombre important de messages. De là, le collecteur peut effectuer plus d'un cycle sans être obligé de les communiquer au *Sink* après chaque cycle. Cette solution permet d'économiser de l'énergie au réseau en évitant des opérations (émission et réception) et des mouvements inutiles de l'*EM*.
  Perte de messages : en évitant les allers-retours après chaque cycle vers le *Sink*, l'*EM* rassemble plus de messages.

Remarque: Dans ce cas, il est primordial d'ajuster le temps de contact résiduel entre le nœud relais et le *Sink* selon la taille du message à retransmettre afin d'éviter la perte de celui-ci. Pour cela, il faudrait par exemple programmer des temps de pauses pour l'*EM* une fois qu'il est en contact avec le *Sink*.

- **Latence**

Le troisième paramètre que nous avons considéré dans notre étude est la « *latence* ».Celle-ci est calculée comme étant le temps écoulé entre la génération du message et sa réception par le *Sink* (collecteur). Les résultats obtenus sont illustrés dans la figure 4.8.

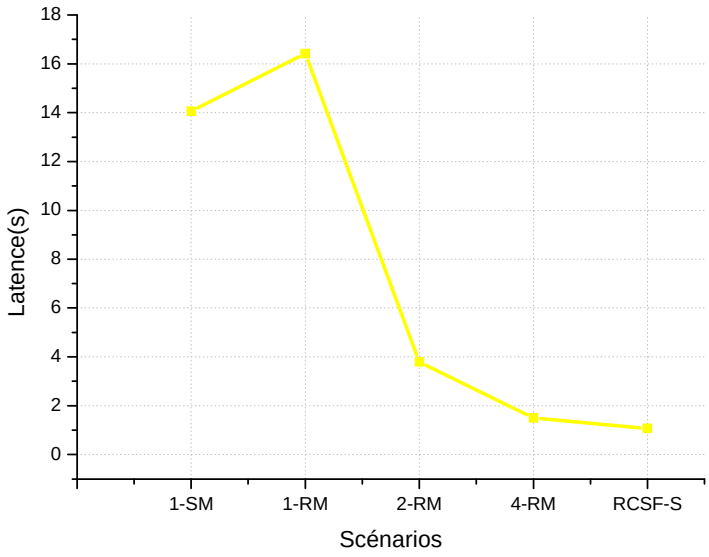


Figure 4.8 – Latence des messages de données

Les valeurs relatives à la latence obtenues dans les scénarios réalisés sont données ci-dessous par ordre croissant :

1. RCSF-S
2. 4-RM
3. 2-RM
4. 1-SM
5. 1-RM

La latence est beaucoup plus importante dans les scénarios d'un (*1-RM, 1-SM*) et généralement dans toutes les configurations des RCSFs-EMs. Ceci est dû au temps de séjour des messages collectés au niveau de l'élément mobile pendant son parcours jusqu'à leurs retransmission au *Sink*. Le scénario du RCSF stationnaire offre de meilleurs résultats en termes de latence alors que dans les scénarios des RCSFs-EM, la latence est beaucoup plus importante, cette perte de temps s'explique par le déplacement de l'élément mobile. Donc tous ces paramètres dépendent du type d'application que l'on vise. La meilleure solution est de trouver un compromis entre les trois paramètres vus précédemment (***énergie consommée, perte de messages et latence***).

D'après les résultats discutés précédemment, la solution basée sur quatre relais mobiles s'avère la plus intéressante en termes de consommation d'énergie, de latence et de nombres de messages collectés et reçus avec fiabilité. Mais il faut toujours prendre en considération que les relais possèdent une batterie limitée en énergie et qu'ils sont assez coûteux. Alors nous avons préféré nous orienter vers une configuration avec *CDM* de type *Sink* mobile.

4.5 Conclusion

Dans ce chapitre, nous avons présenté les différentes simulations réalisées traitant les méthodes de collecte de données dans un RCSF-EM en utilisant soit des nœuds relais mobiles (un, deux ou quatre), soit un *Sink* mobile. Nous avons également vu ce processus dans un RCSF stationnaire.

Dans ces scénarios, nous avons pris en considération trois paramètres de performance du réseau à savoir : la consommation d'énergie, la perte de messages et la latence.

Selon les résultats obtenus et discutés dans la section précédente, nous avons observé que les méthodes à base de *CDMs* consomment beaucoup moins d'énergie que la méthode statique (RCSF stationnaire).

Aussi dans les RCSFs-EM, nous avons constaté que la méthode qui permet de conserver plus d'énergie et qui garantit plus de fiabilité de transfert des messages et également de latence est celle des ***quatre nœuds relais mobiles (4-RM)***. La seconde meilleure solution est celle du *Sink* mobile et vu les avantages de cet élément mobile (capacité énergétique et de stockage illimitée) alors nous avons opté pour cette solution.

Néanmoins, il est à noter que la méthode statique offre une meilleure latence par rapport à celle des éléments mobiles. Donc la sélection de la meilleure méthode de collecte de données repose sur le domaine d'utilisation de celle-ci, est-ce que l'on veut gagner en latence, en énergie ou bien en nombre de messages reçus. Dans notre cas, nous nous intéressons plus à l'énergie dissipée par le réseau pendant le processus de collecte de données afin de prolonger la durée de vie du réseau.

Chapitre 5

Quel modèle de mobilité pour le déplacement du Sink

Sommaire

Chapitre 5 Quel modèle de mobilité pour le déplacement du Sink

5.1 Introduction

La principale problématique dans un RCSF est la consommation d'énergie des nœuds capteurs. En effet, la transmission des données d'un capteur représente environ 70% de sa consommation d'énergie selon les auteurs de [114].
Malgré cet handicap, les nœuds capteurs doivent rester opérationnels le plus longtemps possible et dans des conditions parfois difficiles (ex : lâchés par avion sur les parois d'un volcan, etc….).Afin de satisfaire cette exigence, nous avons pensé à exploiter la mobilité pour essayer de pallier à ces problèmes.
Cependant, la gestion de la mobilité d'un collecteur mobile dans un RCSF génère plusieurs problématiques liées au déplacement de ce *Sink* mobile. Parmi ces problématiques : le choix de l'élément mobile à déplacer dans le réseau (*Sink* ou nœud relais, problématique traitée dans le chapitre 4 et les résultats ont montré qu'il était préférable de déplacer un nœud *Sink* plutôt qu'un nœud relais). Dans ce chapitre, nous nous intéressons au choix du modèle de mobilité selon lequel nous devons déplacer le *Sink* afin de collecter les données et qui donnerait de meilleures performances au réseau [15].

Nous allons présenter un survol des différents travaux réalisés dans le domaine des modèles de mobilité utilisés dans les RCSFs mobiles, ensuite nous donnerons une classification de ces modèles (deux catégories de modèles : de groupes et d'entité) en s'intéressant particulièrement aux modèles de mobilité d'entité. Parmi cette catégorie, nous avons choisi trois modèles que nous avons étudiés avec soin (*Random WayPoint, Gauss Markov et Random Walk*) pour le déplacement du *Sink* dans un RCSF.
Nous avons simulé plusieurs scénarios avec des réseaux constitués de plusieurs nœuds capteurs et des vitesses de déplacement différentes (basse, moyenne et élevée) et en changeant dans chaque scénario le modèle de mobilité (parmi les trois modèles choisis) afin de pouvoir étudier leur impact sur le mécanisme de collecte de données. Dans cette contribution, nous avons considéré quelques métriques de performance importantes telles que le nombre de paquets reçus, le nombre de paquets perdus ainsi que le nombre de collisions. Ces simulations ont été réalisées avec le simulateur OMNET [184], et la plateforme INET.
Le but de notre travail est de déduire le meilleur modèle de mobilité qui offre :

- Moins de perte de paquets
- Une optimisation de la consommation d'énergie
- Une collecte de données fiable et efficace
- Gain de temps (moins de latence)
- Etendre la durée de vie du réseau
- Eviter au maximum les collisions

5.2 Travaux réalisés

Dans cette section, nous passons en revue les travaux effectués sur les modèles de mobilité utilisés pour le déplacement de *Sink*s mobiles pour la collecte de données dans les RCSFs.

Une approche pour prolonger la durée de vie des nœuds est étudiée dans [146], C'est l'utilisation d'un *Sink* mobile. Sur certains aspects, il est similaire d'utiliser plusieurs *Sink*s statiques; Cependant, l'utilisation de plusieurs *Sink*s statiques nécessite une communication globale supplémentaire pour collecter toutes les données en un seul point final. Afin de combler les lacunes observées par l'utilisation d'un *Sink* statique, l'utilisation d'un *Sink* mobile a été proposée. Un *Sink* mobile peut suivre différents types de modèle de mobilité dans le domaine des capteurs, tels que la mobilité aléatoire, prévisible/chemin fixe de mobilité, ou mobilité contrôlée, ce qui a des conséquences en terme de stratégie d'efficacité énergétique et de collecte de données. Dans ce qui suit, nous résumons quelques solutions proposées pour chaque type de mobilité.

Un réseau Ad-hoc mobile est une collection d'auto configuration et d'adaptation de liaison sans fil entre appareils communicants (dispositifs mobiles) pour former une topologie arbitraire et une connectivité sans fil multi-sauts sans l'utilisation d'infrastructure existante. Il nécessite un protocole de routage dynamique efficace pour déterminer les routes suivant un ensemble de règles qui permettent à deux ou plusieurs dispositifs de communiquer entre eux. Les auteurs dans [186], classent essentiellement et évaluent les métriques de mobilité en deux catégories: les métriques de mobilité directe et les métriques de mobilité dérivée. Ces deux métriques de mobilité ont été utilisées pour mesurer les différents modèles de mobilité. Cet article examine certains modèles tels que « *Random Waypoint Model, Reference Point Group Mobility Model, Random Direction Mobility Model, Random Walk Mobility Model, Probabilistic Random Walk, Gauss Markov, Column Mobility Model, Nomadic Community Mobility Model and Manhattan Grid Model* ».

Dans l'évaluation des performances d'un protocole pour un réseau Ad-hoc, le protocole doit être testé dans des conditions réelles, mais sans se limiter à une plage de transmission sensible, à l'espace limité du tampon (buffer) pour le stockage des messages; à des modèles représentatifs de trafique des données; et aux mouvements réalistes des utilisateurs mobiles (soit un modèle de mobilité). Cette étude [187], est un état de l'art des modèles de mobilité qui sont utilisés dans la simulation des réseaux Ad-hoc. Ils décrivent plusieurs modèles de mobilité qui représentent les nœuds mobiles dont les mouvements sont indépendants les uns des autres (c'est à dire, des modèles de mobilité d'entité) et plusieurs modèles de mobilité qui représentent les nœuds mobiles dont les mouvements sont dépendants les uns des autres (les modèles de mobilité de groupe). L'objectif principal de cette étude est de présenter un certain nombre de modèles de mobilité afin d'offrir aux chercheurs des choix plus éclairés quand à l'utilisation du modèle de mobilité choisie.

5.3 Modèles de mobilité

Un modèle de mobilité doit être étudié dans des conditions réalistes en tenant compte d'une zone de transmission variable en raison d'une possible présence d'obstacles, d'espace mémoire limité au stockage des messages, des modèles de trafic de données ainsi que des mouvements réalistes des nœuds mobiles. Il existe plusieurs modèles de mobilité qui proposent des stratégies de mouvements des capteurs [188]. Certains modèles prennent en charge le déplacement de l'ensemble des nœuds, tant dis que d'autres prennent en charge un seul nœud. La première catégorie de modèles est la mobilité de groupe, la deuxième catégorie est la mobilité d'entité. Nous pouvons voir cette classification sur la figure 5.1.
La catégorie qui nous intéresse dans notre travail est la deuxième catégorie « mobilité d'entité » car nous nous intéressons au déplacement du nœud *Sink* qui représente une seule entité et non pas un groupe. Les autres nœuds du réseau sont considérés comme statiques.

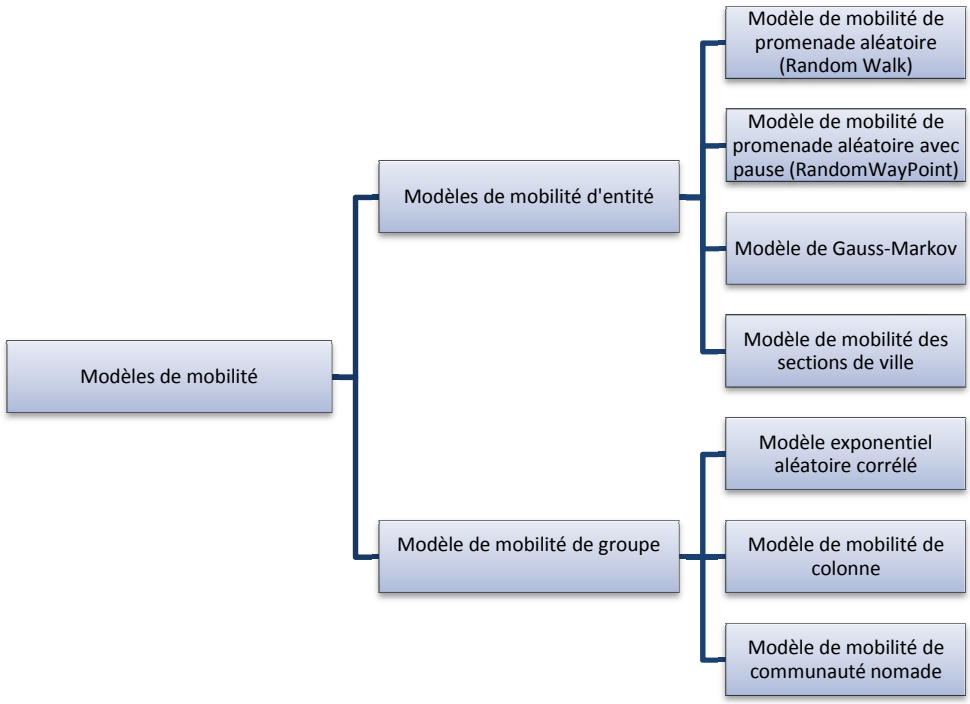


Figure 5.1 – Classification des modèles de mobilité dans les RCSFs

5.3.1 Modèles de mobilité d'entité

Il existe plusieurs modèles de mobilité d'entité, cependant dans cette section nous présentons trois modèles de mobilité d'entité que nous avons proposés pour l'évaluation des performances d'un protocole de RCSF. Parmi les modèles les plus utilisés actuellement et que nous allons voir en détail dans la prochaine section: le modèle de promenade aléatoire :Random walk (*RWMM*), le modèle de mobilité de promenade aléatoire avec temps de pause :RandomWay point (*RWpMM*) ainsi que le modèle *Gauss Markov*.

5.3.1.1 Promenade aléatoire avec pause : le modèle de mobilité de promenade aléatoire avec pause noté RWpMM (*Random Waypoint Mobility Model*) inclut des temps de pause à chaque changement de direction et/ou de vitesse. Un nœud mobile commence à se déplacer, en attendant, dans un endroit pendant une certaine période de temps (temps de pause) [189].

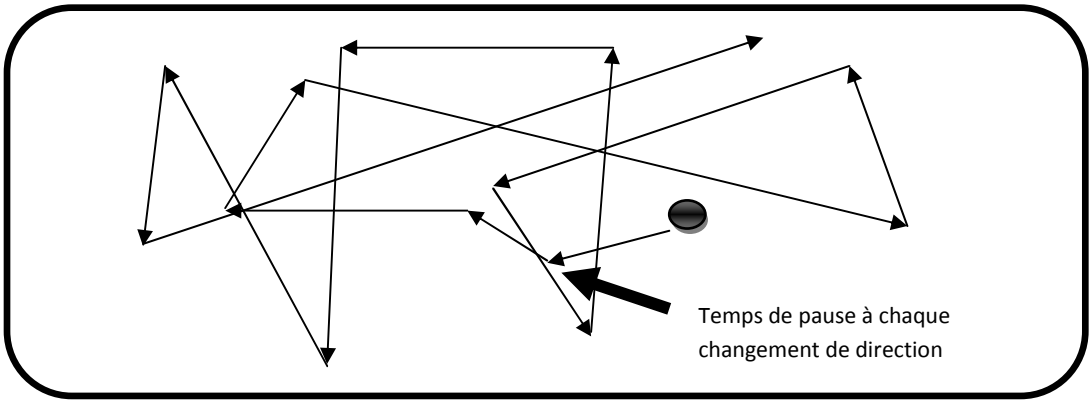


Figure 5.2 – Modèle de promenade aléatoire avec pause (RandomWayPoint)

Une fois la période de temps expirée, le nœud mobile choisit une destination aléatoire et une vitesse distribuée dans$[minspeed \dots maxspeed]$.
Le nœud se déplace alors vers la nouvelle destination choisie avec la vitesse choisie. À l'arrivée, il fait une pause pendant une période de temps définie, avant de recommencer le processus.
La figure 5.2 illustre le déplacement d'un nœud mobile suivant le modèle RWpMM, ce modèle est largement répandu et est souvent utilisé pour l'évaluation des performances des algorithmes et des protocoles de routage. Dans la majorité des tests de performance qui utilisent le modèle RWpMM, les nœuds mobiles sont d'abord distribués aléatoirement dans le secteur de simulation.

5.3.1.2 Promenade aléatoire : le modèle de mobilité de promenade aléatoire noté RWMM (*Random Walk Mobility Model*) a été décrit pour la première fois mathématiquement par Einstein en 1926. Puisque beaucoup d'entités dans la nature se déplacent de manière extrêmement imprévisible, le RWMM a été développé pour imiter ce mouvement erratique. Dans ce modèle, un nœud mobile se déplace de son endroit actuel à un nouvel endroit en choisissant aléatoirement une direction et une vitesse de déplacement. Les nouvelles vitesses de direction sont toutes les deux choisies dans un intervalle prédéfini, $[speedmin \dots speedmax]$et $[0 \dots 2\pi]$ respectivement [189,190].
Chaque mouvement se produit pendant un intervalle de temps constant $'t'$ ou sur une distance constante $'d'$ traversée au-delà desquels une nouvelle direction et une nouvelle vitesse sont choisies. Si un nœud mobile qui se déplace selon ce modèle et qui arrive à la frontière de la zone de simulation, il rebondit contre elle, avec un angle déterminé par la direction d'incidence. Le nœud mobile continue son déplacement suivant ce nouveau chemin comme le montre la figure 5.3:

Le modèle RWpMM ressemble au modèle RWMM si le temps est nul et $minspeed = speedmin$ et $maxspeed = speedmax$.

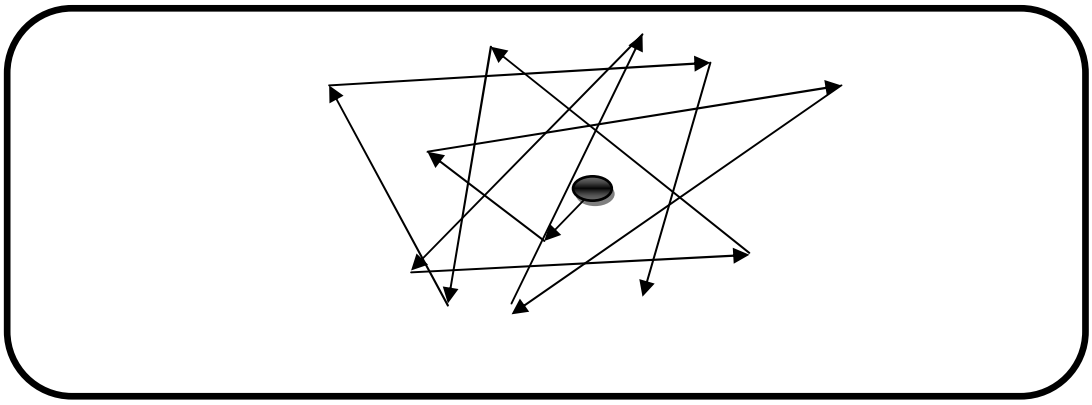


Figure 5.3 – Modèle de promenade aléatoire (Random Walk)

Le modèle 2D *RWMM* est un modèle largement répandu et utilisé. Il est connu parfois sous le nom de « mouvement brownien ». Pour son utilisation, le modèle est parfois simplifié. Ce modèle est sans mémoire, car la vitesse et la direction courantes des nœuds mobiles sont indépendantes des vitesses et des directions précédentes. Ceci peut produire des mouvements peu réalistes tels que des arrêts soudains et des retours à angle aigu.

***5.3.1.3 Gauss-Markov* :** le modèle Gauss-Markov a été proposé pour la simulation d'un réseau PCS (Personnal Communications Services), mais il a aussi été utilisé pour la simulation des protocoles de routage Ad-hoc. Ce modèle a été conçu pour s'adapter aux différents niveaux d'aspect aléatoires, par l'intermédiaire d'un paramètre d'accord. Les mouvements générés par ce modèle évitent les arrêts soudains tout en étant proches de la réalité [191].

Initialement, une vitesse et une direction sont assignées à chaque nœud mobile. À intervalle de temps fixe, un mouvement se produit qui met à jour la vitesse et la direction de chaque nœud. Plus précisément, la valeur de la vitesse et la direction au n^{ième} intervalle de temps sont calculées en fonction des valeurs de la vitesse et de la direction du $(n-1)$ i-ème intervalle de temps et d'une variable aléatoire, en utilisant les équations suivantes :

$$S_n = \alpha . S_{n-1} + (1-\alpha).\bar{S} + \sqrt{(1-\alpha^2)S_{n-1}} \quad (5.1)$$

$$d_n = \alpha . d_{n-1} + (1-\alpha).\bar{d} + \sqrt{(1-\alpha^2)d_{n-1}} \quad (5.2)$$

où S_n et d_n sont les nouvelles vitesses et direction du nœud mobile à l'instant n.
α, $0 \leq \alpha \leq 1$, est le paramètre d'accord utilisé pour l'aspect aléatoire ;$\bar{S}$ et $\bar{d}$ sont des constantes représentant la valeur moyenne de la vitesse, de la direction quand n tend vers l'infini; et S_{n-1}et d_{n-1} sont des variables aléatoires d'une distribution de Gauss. A chaque intervalle de temps, la prochaine destination est calculée en utilisant la destination, la vitesse et la direction courantes du mouvement. Plus précisément, pendant un intervalle de temps n, la position d'un nœud mobile est donnée par les équations suivantes :

$$X_n = X_{n-1} + S_{n-1} cos\ d_{n-1} \quad (5.3)$$
$$Y_n = Y_{n-1} + S_{n-1} sin\ d_{n-1} \quad (5.4)$$

où (X_n, Y_n) $et(X_{n-1}, Y_{n-1})$ sont respectivement les coordonnées X et Y de la position du nœud au n-ième intervalle ; S_{n-1}, d_{n-1}sont la vitesse et la direction du nœud mobile pendant le $(n-1)$-ième intervalle.
La figure 5.4 représente la trajectoire suivie par l'élément mobile selon le modèle Gauss-Markov.

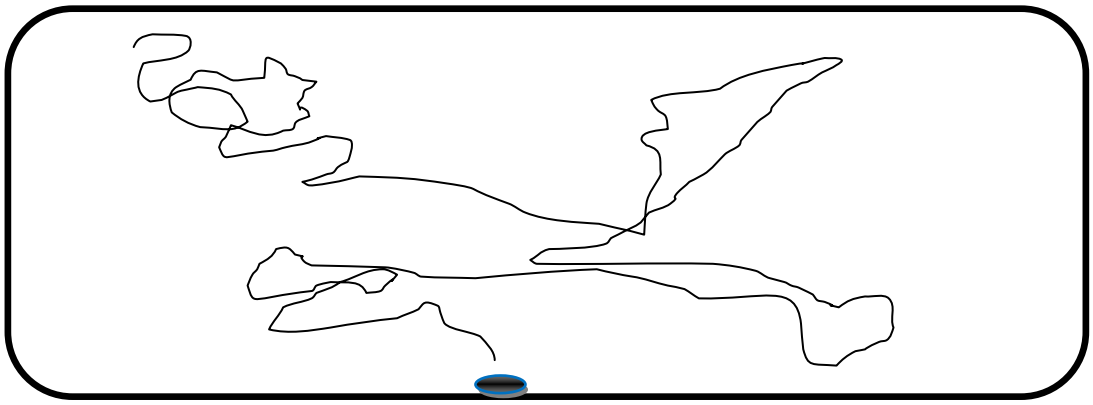


Figure 5.4 – Modèle de mobilité Gauss-Markov

5.3.2 Modèle de mobilité de groupe

Dans ce type de mobilité, le déplacement d'un nœud dépend des déplacements d'un ensemble de nœuds. Dans le domaine militaire par exemple, un groupe de soldats peut effectuer une mission dans une partie de terrain afin de détruire des mines anti-personnelles, capturer des ennemis assaillants ou travailler simplement ensemble de façon coopérative. Il existe plusieurs modèles de mobilité de groupe [188,192], tels que le modèle exponentiel aléatoire corrélé (*ECRMM)*, le modèle de mobilité de colonne (CMM), le modèle de mobilité de communauté nomade (NCMM), le modèle de mobilité de poursuite (*PMM)* , le modèle de mobilité d'un groupe avec point de référence (RPGMM) ainsi que le modèle de mobilité avec obstacles Voronoi.

5.3.3 Discussion et classification des modèles de mobilité

La performance d'un protocole de routage dans les RCSFs peut changer de manière significative lorsque nous le testons avec différents types de modèles de mobilité, mais aussi quand le même modèle de mobilité est utilisé avec différents paramètres. Aussi, ce modèle doit se rapprocher le plus de la réalité [193]. Le modèle de mobilité de promenade aléatoire, *RWMM* avec comme paramètre d'entrée (distance ou temps) produit un mouvement Brownien et par conséquent évalue un réseau statique lorsqu'il est utilisé pour l'évaluation des performances. Cependant, avec l'utilisation d'un grand nombre de paramètres d'entrée, le

modèle *RWMM* ressemble au modèle de mobilité *RWpMM* si nous ajoutons des temps de pause.
La principale différence entre ces deux modèles est que les nœuds composant le modèle *RWpMM* ont plus tendance à se regrouper au centre du secteur de simulation que les nœuds composant le modèle *RWMM*. Le modèle *RWpMM* est utilisé dans plusieurs études de protocoles dédiés aux RCSFs. C'est un modèle flexible et réaliste mais l'inconvénient majeur de ce modèle est la ligne droite du mouvement, suivi par le nœud mobile qui se déplace vers la prochaine destination choisie.
Le modèle de mobilité *Gauss-Markov* fournit des modèles de mouvement auxquels nous pouvons s'attendre dans la réalité si nous choisissons les bons paramètres. En outre, la méthode utilisée pour forcer les nœuds à partir des bords du secteur de simulation (évitant ainsi les effets du bord de secteur) est intéressante.

La table 5.1 présente de manière compacte les principales caractéristiques, comme la vitesse, la direction et la nature (réaliste ou pas) des modèles de mobilité étudiés dans cette thèse.

Type du modèle de mobilité	**Nom du modèle**	**Direction aléatoire**	**Vitesse aléatoire**	**Réalisme**	**Type d'application**
Entité	RWMM	Oui	Oui	Peu	Mouvement brownien
	RWpMM	Oui	Oui	Oui	WLAN
	GaussMarkov	Non	Non	Oui	PCS (Personal communication service)
Groupe	ECRMM	Non	Oui	Non	
	NCMM	Non	Oui	Oui	Visite touristique
	RPGMM	Oui	Oui	Oui	Situation de secours

Table 5.1 – Caractéristiques des modèles de mobilité

5.3.4 Algorithmes de mobilité des trois modèles étudiés

Il existe dans la littérature plusieurs modèles de mobilité, chaque modèle possède ses propres caractéristiques, ses avantages, et ses inconvénients. Malgré leur diversité, la majorité des algorithmes de mobilité sont stochastiques. Dans cette section, nous allons décrire les algorithmes des modèles de mobilité étudiés dans cette thèse (*Random WayPoint*, *Random Walk* et *Gauss-Markov*).

- Dans ***l'algorithme*** *« Random WayPoint »*, la mobilité des nœuds est aléatoire et tous les nœuds sont distribués uniformément dans l'espace de déploiement.
  L'affectation d'une position, d'une vitesse et d'une destination initiale est aléatoire pour chaque nœud mobile. A chaque fois qu'un nœud mobile atteigne sa destination dans la zone de déploiement, il repart vers une autre destination choisie aléatoirement après un éventuel temps de pause. Ce processus se répète tant que la condition d'arrêt (destination finale) n'est pas satisfaite.

Algorithme 1 « Random WayPoint »

```
1 :Position au sol_x=600 ;
2 :Position au sol_y=400 ;
3 :Position_X=uniform(0,Position au sol_x) ;
4 :Position_Y=uniform(0,Position au sol_y) ;
5 :Pause_time=30 ;
6 :tant que(!Condition d'arret)
7 :faire
8 :   destination_X=uniform(0, Position au sol_x) ;
9 :   destination_Y=uniform(0,Position au sol_y) ;
10:   aller à (Position_X, Position_Y, destination_X, destination_Y) ;
11:   Position_X=destinationt_X ;
12:   Position_Y=destination_Y ;
13:   Pause(Pause_time) ;
14:fin tant que
```

- ***L'algorithme*** *« Random Walk »* est basé sur un mouvement imprévisible. Un nœud mobile se déplace de sa position initiale vers une nouvelle position en choisissant aléatoirement une direction et une vitesse. Le déplacement de ce nœud se fait pendant un temps *t*. Un nœud mobile qui atteint la limite de la zone de déploiement, recommence le processus en choisissant une nouvelle direction et une nouvelle vitesse indépendamment du choix précédent.

Algorithme 2 « Random Walk »

```
1:Position au sol_x=600;
2:Position au sol_y=400;
3:Position_X=uniform(0, Position au sol _x);
4:Position_Y=uniform(0, Position au sol _y);
5:tant que(!Condition d'arret)
6:faire
7:    destination_X==uniform(0, Position au sol _x);
8:    destination_Y=uniform(0,Position au sol _y);
9:    aller à(Position_X, Position_Y, destination_X, destination_Y) ;
10:   Position_X=destination_X;
11:   Position_Y=destination_Y;
12:fin tantque
```

- Dans ***l'algorithme*** *« Gauss Markov »*, la position et la vitesse d'un nœud à tout instant dépendent de la position et de la vitesse au moment précédent. Pour s'assurer qu'un nœud

ne reste pas près de la bordure de simulation, les nœuds sont poussés loin de la bordure lorsqu'ils sont à moins d'une certaine distance. Ce principe est réalisé en modifiant la valeur de la direction moyenne au cours de la simulation. Par exemple, lorsqu'un nœud est proche de la bordure droite, la valeur de la direction moyenne change à 180°, alors la nouvelle direction du nœud l'éloigne de cette bordure.

Algorithme 3 « Gauss-Markov »

```
1:Position au sol_x=600 ;
2:Position au sol_y=400 ;
3:x=uniform(0, Position au sol_x) ;
4:y=uniform(0, Position au sol_y) ;
5: tant que (Condition d'arrêt non atteinte)
6: faire
7: x_temp=position_x+vitesse*cos(direction) ;
8: y_temp=position_y+vitesse*sin(direction) ;
9: vitesse_normale=random(0,1);
10:direction_normale=random(0,1);
11:vitesse_temp=alpha*vitesse+(1-alpha)*vitesse_moyenne+sqrt(1-alphaau
carré)*vitesse_normale;
12:direction_temp =alpha*direction+(1-alpha)*direction_moyenne+sqrt(1-
alpha au carré)*direction_normale ;
//  gestion des bordures
13:si (x_temp<dist_bord) alors direction_pricipale =0;
14:si (x_temp>position au sol_x- dist_bord) alors direction_principale
=180˚;
15:si (y_temp<dist_bord) alors direction_principale =90˚;
16:fin tant que
```

5.3.5 Etapes de l'approche proposée

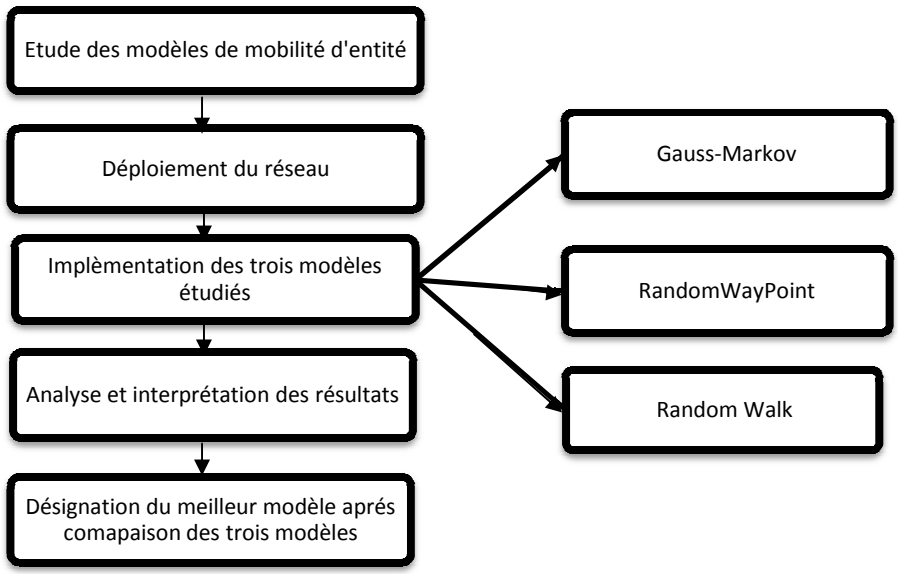


Figure 5.5 – Etapes de l'approche proposée

La figure 5.5 représente les étapes de l'approche proposée dans ce chapitre. Après avoir étudié les modèles de mobilité de groupe et d'entité, nous avons choisi les modèles d'entité pour leur adéquation avec notre problématique (un seul nœud mobile). Pour cela, nous avons déployé un certain nombre de capteurs où seul le *Sink* est mobile. L'implémentation de ces trois modèles: Gauss-Markov ; Random Way point et Random walk sous l'environnement Inet a conduit aux résultats de simulation qui seront analysés et interprétés dans les prochaines sections. A la fin, une comparaison a été faite pour désigner le meilleur modèle qui répond le mieux à nos exigences.

5.4 Simulation et résultats préliminaires

5.4.1 Environnement de simulation

Dans cette deuxième contribution, nous avons également réalisé nos simulations à l'aide du simulateur OMNET avec sa plateforme INET. Ce simulateur est riche en termes de modèles de mobilité et a été présenté en détail dans le chapitre 4.

5.4.2 Description du réseau et des scénarios réalisés

Dans ce travail, nous considérons un réseau composé d'un ensemble de capteurs statiques et d'un collecteur de données de type *Sink* mobile pour assurer le processus de collecte de données. La figure 5.6 montre les différents composants du réseau.

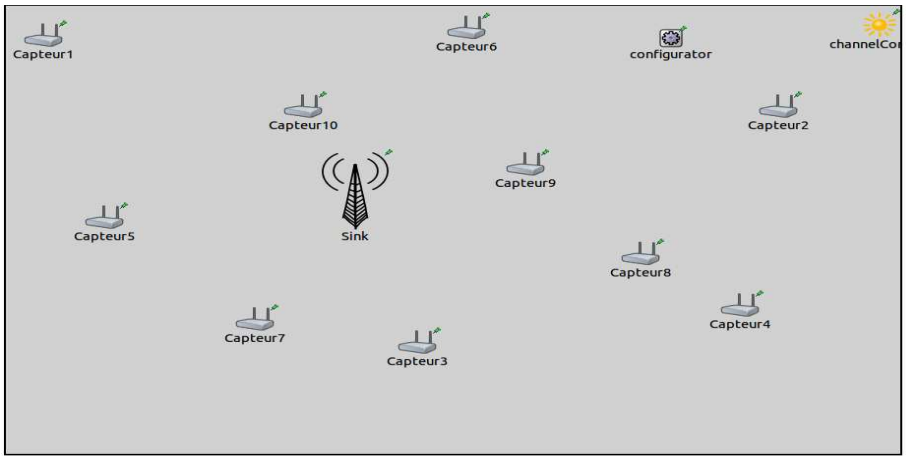


Figure 5.6 – Composants du réseau

5.4.3 Scénarios de simulation

Plusieurs scénarios ont été réalisés, dans chaque scénario nous faisons varier le type de modèle de mobilité, la vitesse du *Sink*, le temps de simulation, et le nombre de capteurs dans le réseau ainsi que le nombre de canal radio dans le *Sink*. Nous avons fait varier ce dernier paramètre afin de pouvoir mettre l'ensemble des capteurs dans des domaines de collision

différents. Par conséquent étudier la collecte de données sans l'influence de la perte de paquets par une collision.

Nous avons testé trois (3) modèles de mobilité sur des réseaux contenant 3 et 10 capteurs avec des vitesses de déplacement différentes.

5.4.3.1 Scénario 1

La table 5.2 présente les différents paramètres de simulation du scénario 1.

Modèle de mobilité	**Gauss Markov**	**RandomWaypoint**	**RandomWalk**
Nombre de capteurs	3	3	3
Protocole de routage	AODV	AODV	AODV
Couche Mac	IEEE802.11a	IEEE802.11a	IEEE802.11a
Type d'application	TCP Session APP	TCP Session APP	TCP Session APP
Type de communication	Capteur->Sink	Capteur->Sink	Capteur->Sink
Temps de simulation	500s	500s	500s
Vitesse de déplacement (mètre par seconde)	10 mps	[1 mps, 10 mps]	[1 mps, 10 mps]

Table 5.2 **–** Paramètres de simulation du scénario 1

a) Interprétation des résultats du scénario 1 : les résultats du scénario 1 sont représentés dans la table 5.3. Les trois figures 5.7, 5.8, et 5.9 montrent l'évolution du nombre de paquets reçus en fonction du temps pour les trois modèles de mobilité étudiés.

Modèle de mobilité	**Gauss Markov**	**RandomWay Point**	**Random Walk**
Nombre de paquets reçus	5704	5711	6300
Nombre de paquets perdus	225	180	550
Nombre de collisions	2	0	2

Table 5.3 **–** Résultats de simulation du scénario 1

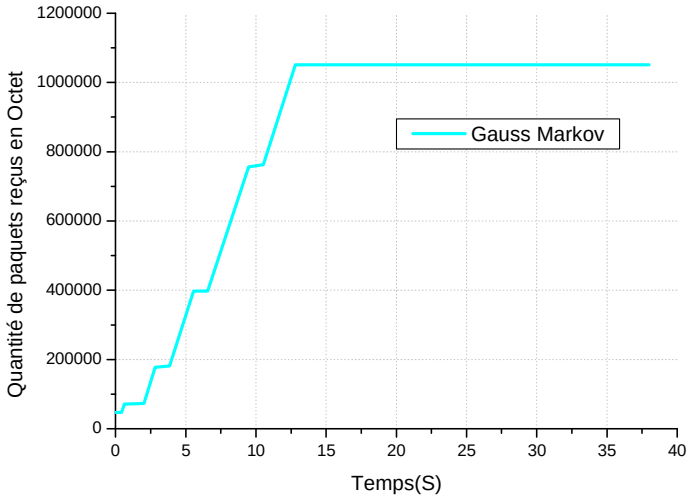


Figure 5.7 – Nombre de parquets reçus/ temps (Gauss Markov)

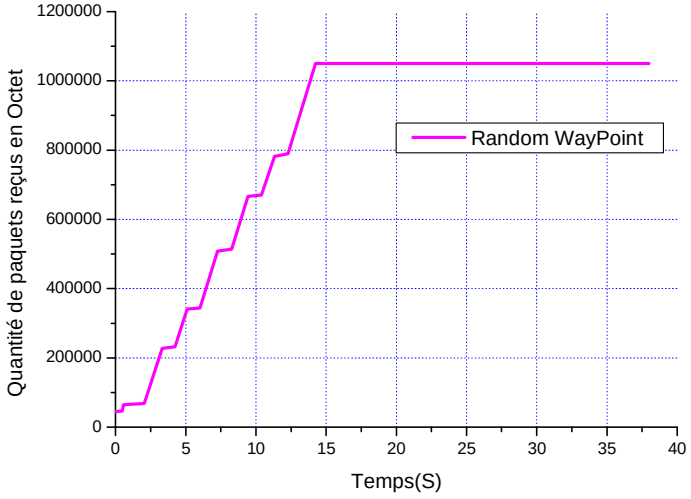


Figure 5.8 – Nombre de parquets reçus/ temps (RandomWayPoint)

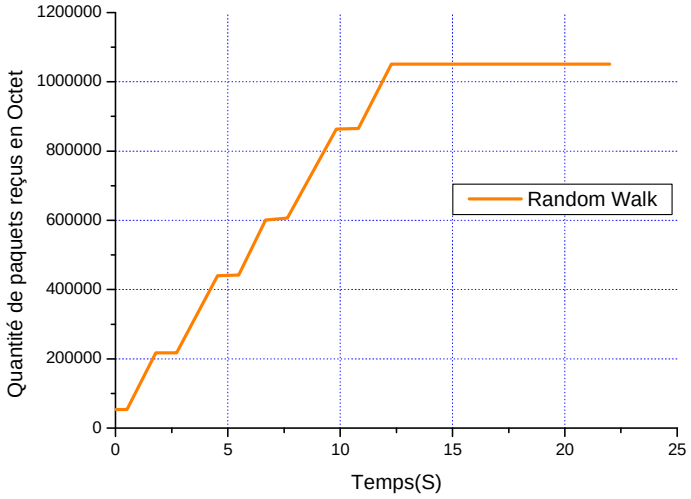


Figure 5.9 – Nombre de parquets reçus/ temps (RandomWalk)

- **« *Gauss Markov* »**, dans ce modèle le mouvement aléatoire dirigé du *Sink* ainsi que son ancienne position lui permettent de recevoir le maximum de messages pour la simple raison qu'il ne revisite pas tout de suite un capteur qu'il vient de visiter. Le nombre de paquets reçus augmente plus rapidement (voir figure 5.7).
- **« *Random WayPoint* »** est un modèle à mouvement aléatoire, donc le *Sink* se déplace aléatoirement dans l'espace de déploiement, et marque une pause une fois arrivé à destination, ceci explique la fragmentation du graphe illustré dans la figure 5.8. Le *Sink* entre dans la zone radio d'un capteur, puis ressort de cette dernière aléatoirement.
- **« *Random Walk* »** ressemble beaucoup au modèle *Random Waypoint*, sauf qu'il est moins stable en raison de l'absence d'un temps de pause. Une fois qu'il entre dans une zone radio, il la quitte. Son comportement ne peut pas être prédit où expliqué tout au long de la simulation comme c'est illustré dans la figure 5.9.

b) Comparaison entre les trois modèles de mobilité : A travers les résultats présentés dans la figure 5.10, nous remarquons que parmi les trois modèles de mobilité, ceux de *Gauss Markov* et *Random WayPoint* sont les modèles qui offrent les meilleurs résultats en termes de paquets reçus et perdus. Malgré que l'on remarque une légère différence qui peut être expliquée par leur aspect aléatoire dirigé. Cependant, le modèle *Random Walk* qui est un modèle purement aléatoire, peut donner de mauvais résultats (inexplicables) vu sa nature stochastique. Selon les résultats obtenus et pour répondre à notre question « Quel est le meilleur modèle de mobilité», le modèle *RandomWayPoint* affiche les meilleurs résultats dans le scénario 1(nombre de paquets reçus élevé, nombre de paquets perdu minime et pas du tout de collisions).

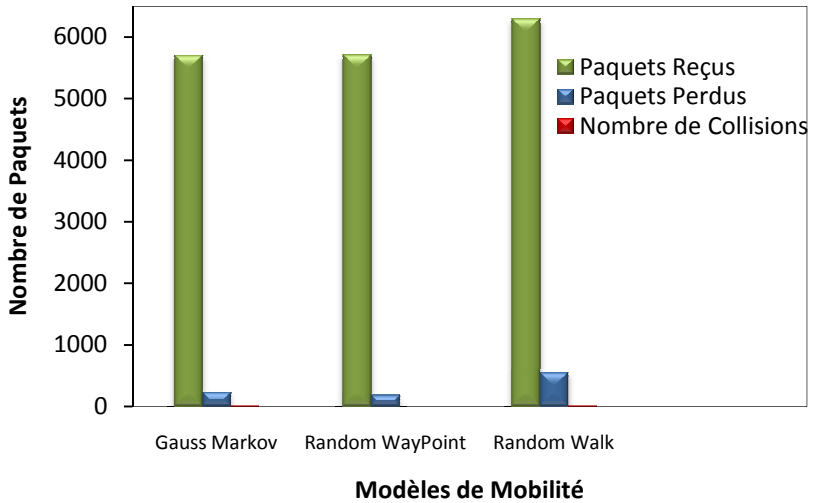


Figure 5.10 – Comparaison entre les trois modèles dans le scénario 1

5.4.3.2 Scénario 2

Dans ce scénario, nous avons considéré les mêmes paramètres de simulation que dans le scénario 1 à l'exception du nombre de capteurs et le temps de simulation qui varient. La table5.4 montre les valeurs des paramètres modifiés :

Modèle de mobilité	**Gauss Markov**	**Random Waypoint**	**Random Walk**
Nombre de capteurs	10	10	10
Temps de simulation	100s	100s	100s

Table 5.4 – Paramètres de simulation du scénario 2

a) Interprétation des résultats du scénario 2 : les résultats du scénario 2 sont décrits dans la table 5.5. Les trois figures 5.11, 5.12 et 5.13 montrent l'évolution du nombre de paquets reçus en fonction du temps pour les trois modèles de mobilité.

Modèle de mobilité	**Gauss Markov**	**Random WayPoint**	**Random Walk**
Nombre de paquets reçus	1980	1969	1990
Nombre de paquets perdus	158	78	53
Nombre de collisions	20	17	13

Table 5.5 – Résultats de simulation du scénario 2

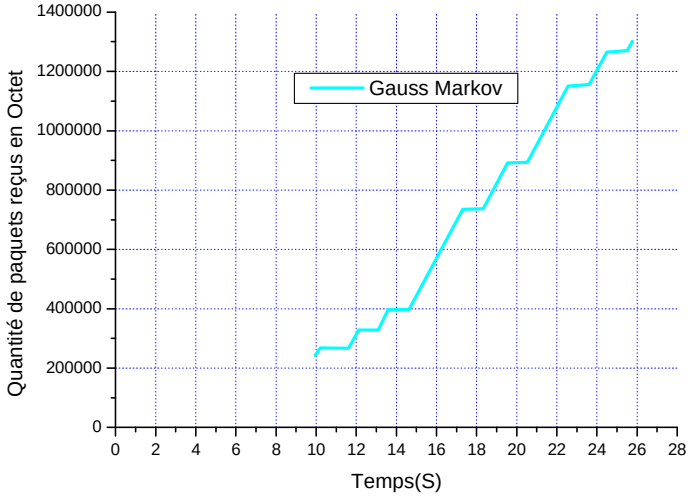


Figure 5.11 – Nombre de parquets reçus/ temps (Gauss Markov)

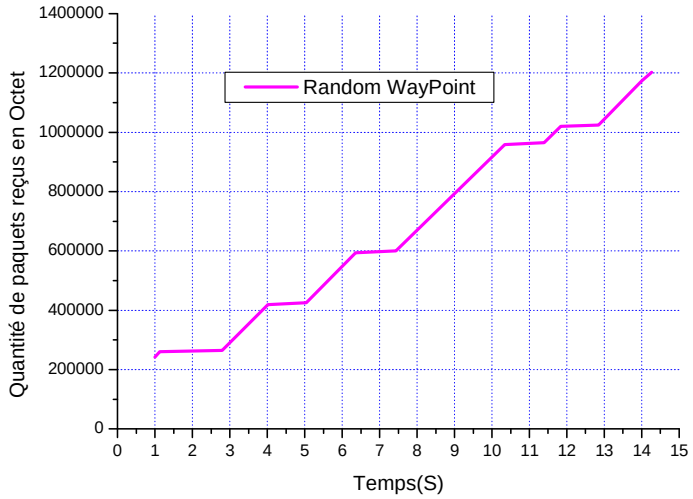


Figure 5.12 – Nombre de paquets reçus/ temps (RandomWayPoint)

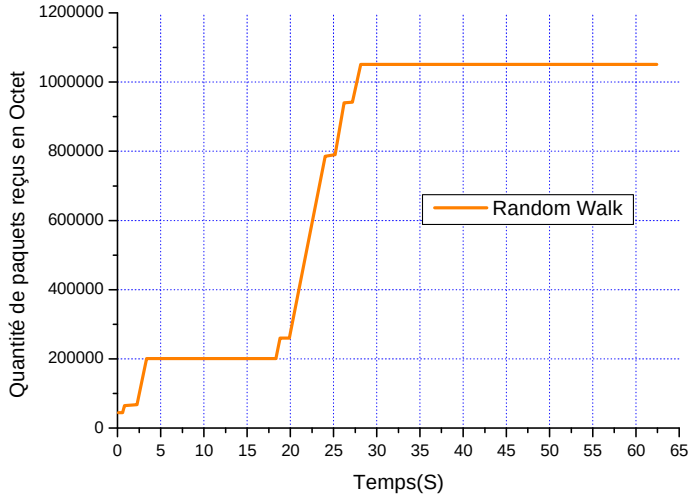


Figure 5.13– Nombre de parquets reçus/ temps (RandomWalk)

- « ***Gauss Markov*** », dans ce modèle l'existence de plusieurs échanges de données simultanés a forcé l'apparition de collisions, ce qui signifie automatiquement l'application de l'algorithme du *back off*[1]. Ceci explique la réception tardive des données par le *Sink* (jusqu'à la 10^{e} seconde de simulation). Cependant, la collecte de données commence à partir de la 10^{e} seconde et est remarquablement croissante dans la figure 5.11.
- « ***Random WayPoint*** » présente moins de collisions que les autres modèles et cela est dû au temps de pause qu'il fait au niveau de chaque capteur. Mais dans certains cas, il

[1]C'est un algorithme utilisé pour limiter la charge du réseau quand une collision se produit entre deux messages émis simultanément par deux stations.

peut devenir un goulot d'étranglement si le *Sink* se trouve dans la portée radio de plusieurs capteurs à la fois lors de son temps de pause. Cela peut mener à plusieurs collisions.

- **« *Random Walk* »** avec plusieurs capteurs permet une bonne recherche de l'information grâce à son aspect stochastique.

b) Comparaison entre les trois modèles de mobilité: la figure 5.14 présente une comparaison entre les trois modèles de mobilité dans le scénario 2.

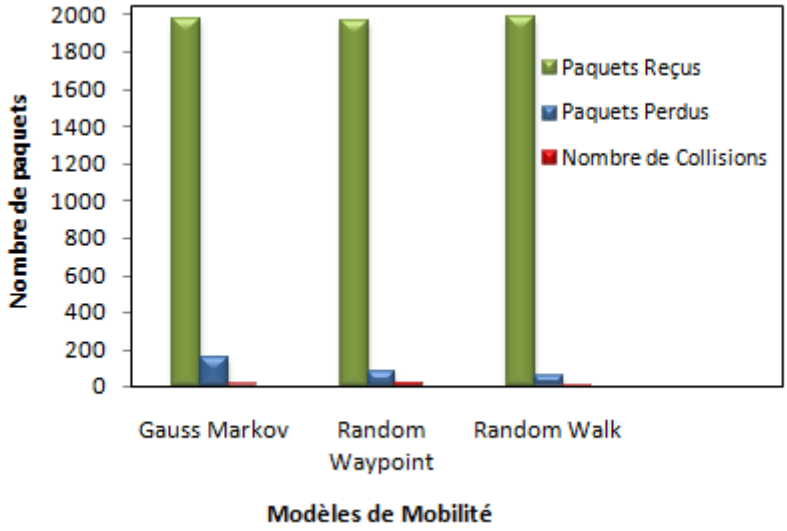


Figure 5.14 – Comparaison entre les trois modèles dans le scénario 2

Nous remarquons à partir des résultats obtenus que les modèles à base de mouvements aléatoires maintiennent toujours leurs bonnes performances (*Random Walk et Random WayPoint*).

5.4.3.3 Scénario 3

Après avoir exécuté les différents scénarios en changeant le nombre de capteurs et le modèle de mobilité dans les scénarios précédents. Dans le scénario 3, nous voulons étudier maintenant l'impact du changement de vitesse de déplacement du *Sink* dans le processus de collecte de données avec les trois modèles de mobilité (*Random WayPoint, Random Walk* et *Gauss Markov*) et avec 10 capteurs déployés dans le réseau.

Les résultats sont présentés dans la table5.6 :

Vitesse de déplacement	[1mps, 4mps]		4mps	[4mps,10mp]		10mps	[10mps,15mp]		15mps
Modèle de mobilité	**Random Walk**	**Random WayPoint**	**Gauss-Markov**	**Random Walk**	**Random WayPoint**	**Gauss-Markov**	**Random Walk**	**Random WayPoint**	**Gauss-Markov**
Nombre de paquets reçus	1105	2231	2147	3217	3325	3115	1300	1750	1120
Nombre de paquets perdus	30	20	35	87	83	77	195	125	251
Nombre de Collisions	3	17	11	2	19	9	2	18	8
Temps de simulation	200s	200s	200s	200s	200s	200s	200s	200s	200s

Table 5.6 –Résultats de simulation du scénario 3

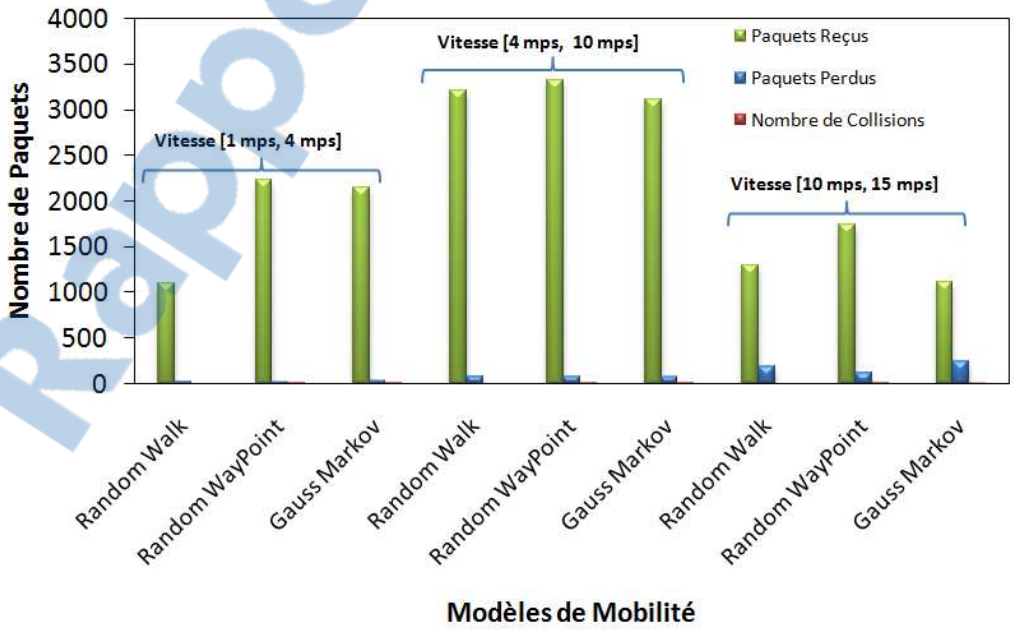


Figure 5.15 – Comparaison entre les trois modèles dans le scénario 3

Comparaison et Interprétation des trois modèles de mobilité

Nous pouvons voir sur la figure 5.15 :

- Dans l'intervalle ***[1 mps, 4 mps]***, nous remarquons que les deux modèles *Gauss Markov* et *Random WayPoint* ont donné de bons résultats par rapport au modèle *Random walk* en terme de paquets reçus, cependant nous remarquons une perte de paquets importante dans *Gauss Markov* et dans *Random walk*.
- Dans l'intervalle ***[4 mps, 10 mps]***, les trois modèles donnent de bons résultats pour la quantité de paquets reçus, par contre le nombre de paquets perdus a augmenté.
- Dans l'intervalle ***[10 mps, 15 mps]***, le nombre de paquets perdus augmente. En effet, nous remarquons que les deux modèle *Gauss Markov* et *Random Walk* enregistrent une importante perte de paquets, cependant dans le modèle *Random WayPoint* les paquets perdus diminuent, ceci est dû à son temps de pause qu'il marque à chaque déplacement. Nous constatons aussi que le nombre de paquets reçus diminue dans les trois modèles, ceci est dû à l'augmentation de la vitesse de déplacement du *Sink* qui n'a pas pu recevoir tous les paquets à temps.

Donc les trois modèles, avec une vitesse de déplacement moyenne présentent de meilleures résultats par rapport aux modèles où la vitesse est soit basse soit élevée ; cela est dû au fait que dans un modèle avec une faible vitesse le *Sink* visite moins de capteurs que dans un modèle avec une vitesse plus élevée. Cependant si la vitesse devient trop élevée, le *Sink* quitte la zone radio d'un capteur plus rapidement, ainsi il faut trouver un bon compromis entre la vitesse de déplacement et la collecte de données efficace.

5.5 Conclusion

Nous avons étudié dans ce chapitre l'impact du modèle de mobilité sur le processus de collecte de données dans un RCSF avec *Sink* mobile. Pour cela, avons mené un certain nombre de simulations représentant différents types de scénarios où les nœuds capteurs sont stationnaires et où le *Sink* est mobile, en variant dans chaque scénario quelques paramètres tels que le temps de simulation, le nombre de capteurs ainsi que la vitesse de déplacement du *Sink*. Les résultats de simulation montrent clairement que les modèles à base de mouvement aléatoire (*Random WayPoint, Random Walk)* donnent de meilleurs résultats que les modèles à chemins définis à l'avance (*Gauss-Markov*). La vitesse de déplacement du *Sink* ainsi que son temps de pause sont aussi des facteurs importants qui influent sur la réception et la perte de paquets dans le réseau pendant la collecte de données. Nous avons pu voir que les modèles ayant une vitesse moyenne et un temps de pause raisonnable présentent de meilleurs résultats qu'avec une vitesse basse ou élevée. Nous pouvons dire que l'effort qui a été investi dans cette contribution est un effort d'implémentation pour répondre à notre question qui est de savoir : quel est le meilleur modèle de mobilité à utiliser pour déplacer le *Sink* dans notre réseau afin de collecter les données capturées par les nœuds.

Comme perspective, nous pouvons tester d'autres modèles de mobilité plus complexes et mener des expérimentations réelles avec un robot navigateur représentant le *Sink* dans notre cas.

Chapitre 6

Comment peut-on optimiser les déplacements du Sink dans un RCSF mobile

Sommaire

6.1 Introduction

Dans la première contribution, nous avons prouvé que la mobilité du *Sink* était plus intéressante à considérer dans un RCSF pendant le processus de collecte de données. Cette mobilité à permis d'assurer la connectivité et la couverture du réseau, de minimiser la consommation d'énergie et de prolonger la durée de vie du réseau. Cependant, cette approche présente aussi des défis significatifs qui n'apparaissent pas dans les RCSFs statiques.

- ***Modèle de mobilité du Sink : quel modèle de mobilité choisir?***
- ***Optimisation de la trajectoire de déplacement du Sink ?***

Après avoir traité la première problématique dans le précédent chapitre, nous nous sommes intéressé à la troisième problématique qui est « *Comment optimiser les déplacements du Sink dans le réseau afin qu'il puisse réaliser une collecte de données fiable, efficace et intelligente* ». Nous visons dans ce travail une application de détection de la présence de survivants sous des décombres après un séisme, inondation, etc. Au lieu de parcourir tous les nœuds du réseau, le *Sink* ne se déplacera que vers les capteurs qui ont détecté la présence de survivants et ceci en introduisant un mécanisme de priorité. Nous avons optimisé la trajectoire du *Sink* en passant par deux étapes: par Simulation pure et par les Méthodes formelles.

Dans la première étape, nous présentons quelques travaux réalisés correspondant à cette problématique, ensuite nous détaillons notre contribution en adaptant les différentes couches protocolaires. Nous présentons notre propre modèle de mobilité implémenté et simulé en utilisant des paramètres prédéfinis de déplacements du Sink (trajectoire, vitesse et temps de pause du Sink etc..) et en prenant en considération toutes nos exigences dans le but d'évaluer les performances de ce modèle et pouvoir les comparer à celles des autres modèles de mobilité étudiés dans le chapitre 5. Ce modèle de mobilité permettra au Sink d'effectuer des déplacements intelligents dans le réseau. Cette simulation est réalisée avec le simulateur Castalia [185] qui est basé sur la plate-forme OMNeT++, et que nous allons présenter dans la prochaine section.

Nous avons implémenté aussi notre propre outil de visualisation « CastaliaViz» pour étendre l'outil Castalia afin de pouvoir visualiser le réseau.

La deuxième étape consiste à utiliser des méta-heuristiques. Ce sont des méthodes approchées (incomplètes) utilisées pour trouver l'optimum qui représente dans notre étude la plus courte trajectoire du Sink en parcourant tous les nœuds capteurs du réseau. En considérant quelques paramètres tels que la consommation d'énergie, le taux d'erreur…..

Nous commençons par présenter les différents travaux réalisés concernant cette problématique. Nous détaillerons ensuite nos contributions en présentant les modèles de mobilité implémentés. Une description du réseau et des scénarios réalisés sera donnée. A la fin, nous analyserons et discuterons les résultats de simulation obtenus.

6.2 Approche basée sur la simulation

6.2.1 Travaux réalisés

Dans cette section, nous présentons quelques travaux réalisés dans le domaine des réseaux de capteurs avec *Sink* mobile (détermination et optimisation de la trajectoire du *Sink* en utilisant le principe du Clustering)

Les chercheurs dans [194] ont essayé de maximiser la durée de vie du réseau et d'équilibrer la consommation d'énergie lorsque le *Sink* se déplace. La méthode soustractive de Clustering modifiée, la méthode k-means[1], et la méthode d'interpolation du plus proche voisin sont utilisées pour déterminer la trajectoire de mouvement. Le modèle d'optimisation de la durée de vie est établi sous les contraintes suivantes : la contrainte de consommation d'énergie, de transmission de liaison ainsi que d'autres contraintes.

La méthode de sous-gradient est utilisée pour résoudre le modèle d'optimisation de la durée de vie lorsque le *Sink* reste à un emplacement fixe. La méthode géométrique est utilisée pour évaluer la collecte de données lorsque le *Sink* est en mouvement. Enfin, tous les nœuds capteurs transmettent des données selon le schéma optimal de transmission de données. Le *Sink* doit collecter les données sur les chemins les plus courts.

Pour prolonger la durée de vie du réseau, les auteurs dans [195] ont utilisé l'approche du *Sink* mobile, cette mobilité rend le réseau dynamique. Trouver une route dans un réseau dynamique est une tache difficile. Dans ce document, ils ont proposé un protocole de diffusion de données arborescente distribuée (TEDD) avec *Sink* mobile. Le protocole est validé par simulation et comparé avec les protocoles existants en utilisant certains paramètres tels que la consommation d'énergie, délai moyen de bout en bout et débit. Les résultats de l'expérience montrent que le protocole proposé surpasse de loin les protocoles existants.

L'optimisation du mouvement d'un *Sink* pour la minimisation de l'énergie est d'abord étudiée dans [136]. Les auteurs ont utilisé une programmation linéaire pour déterminer les emplacements de chaque *Sink* mobile, périodiquement dans l'opération de collecte de données. D'autres, dans [196], ont étudié le même problème en utilisant un modèle décentralisé de programmation linéaire mixte. Les auteurs de [197] ont étudié ce problème en tenant compte des situations les plus réalistes en présence d'obstacles. Ainsi, les *Sinks* mobiles ne peuvent se déplacer le long de certains chemins pour collecter les données. Un travail assez récent dans les RCSFs, se basant sur la maximisation de la durée de vie, en utilisant plusieurs *Sinks* mobiles est présenté dans[198]. Ils ont formulé le problème comme une optimisation de la contrainte multi-sauts des mouvements des *Sinks* mobiles.

6.2.2 Etapes de l'approche proposée

Notre approche est décrite par les étapes résumées dans la figure 6.1.

Pour optimiser les déplacements du *Sink*, il faut lui éviter des déplacements inutiles. A partir de ce principe, nous avons construit notre approche. Nous avons déployé un certain nombre de capteurs que nous avons organisés en Clusters en appliquant le protocole LEACH que nous

[1]C'est une méthode de partitionnement de données et un problème d'optimisation combinatoire.

avons modifié selon nos besoins pour permettre au *Sink* de communiquer qu'avec les ClusterHeads (minimisation de ses déplacements). Des propriétés seront ensuite attribuées à ces ClusterHeads afin que le *Sink* puisse se déplacer en premier lieu vers les CHs prioritaires (déplacements intelligents). Pour faire cette optimisation, nous avons utilisé l'approche par simulation en implémentant un modèle selon nos exigences sous Castalia. Dans l'approche formelle, nous avons choisi deux méthodes (recherche tabou et recuit simulé pour trouver la meilleure trajectoire de déplacement du *Sink*). Une analyse et une interprétation des résultats sera effectuée à la fin.

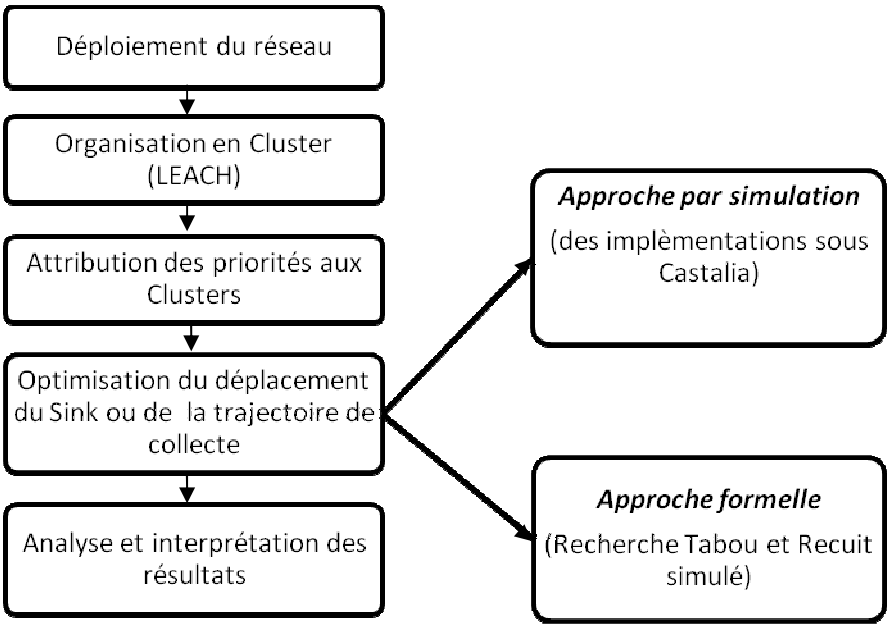


Figure 6.1 – Etapes de l'approche proposée

6.2.3 Adaptation au niveau des couches protocolaires

6.2.3.1 Couche application

En s'inspirant des deux applications de Castalia (ValueReporting et ThroughputTest).

- *L'application ValueReporting* : dans cette application, chaque nœud définit une minuterie pour la détection. Lorsque la minuterie expire, le nœud appelle la fonction qui demande la lecture du capteur avec un indice donné et une autre minuterie est alors réglée. Le résultat de la fonction de lecture sera retourné avec le rappel d'une fonction de contrôle de la valeur détectée, il doit alors transmettre le paquet qui contient l'Id du nœud, sa position (x, y) et les données (valeur détectée) au Sink.
- *L'application ThroughputTest*: est une application où les nœuds envoient au Sink des paquets à intervalle régulier. Pour chaque nœud, nous spécifions un débit d'envoi et la destination du paquet envoyé (Sink), une minuterie est calculée à partir du débit d'envoi, et lancer au début de l'application. Quand cette minuterie expire, le nœud génère un paquet et l'envoie au Sink.

Nous avons implémenté une nouvelle application dédiée au Clustering et qui communique avec le module de mobilité « ClusteringApplication »

6.2.3.2 Couche routage

- Les protocoles de routage, fournis par Castalia, ne prennent pas en charge la mobilité. Donc nous avons intégré le protocole AODV.
- Implémentation d'un protocole de routage à un saut (Capteur->*Sink*) qui supporte la mobilité.

6.2.4 Adaptation au niveau des modèles de mobilité

6.2.4.1 Protocole LEACH (Low-Energy Adaptive Clustering Hierarchy)

Nous allons exploiter le principe de l'auto organisation du réseau (*Clustering*) pour optimiser les déplacements du *Sink* comme nous pouvons le voir sur la figure 6.2. En effet, le protocole de routage est déterminant pour définir la trajectoire du *Sink*.

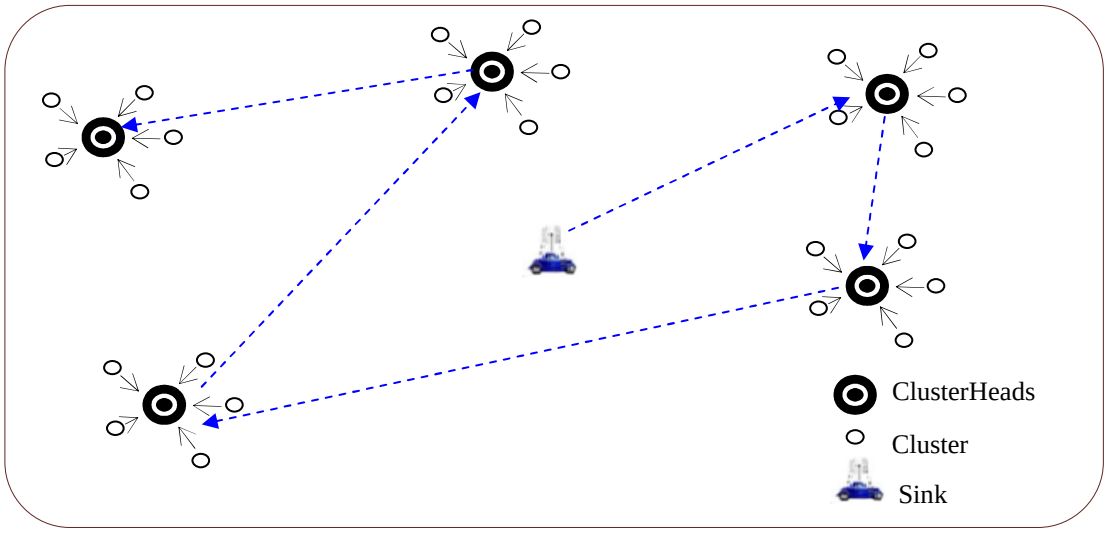


Figure 6.2 – Trajectoire de déplacement du Sink

Le *Protocole LEACH* [199,200] est un algorithme de Clustering distribué pour le routage des RCSFs homogènes. Il divise le réseau en deux parties : les ClusterHeads (chefs de clusters) et les nœuds membres. Il choisit aléatoirement les ClusterHeads et attribue ce rôle aux différents nœuds selon la politique de gestion Round-Robin[2] pour garantir une dissipation équitable d'énergie entre les nœuds. Dans le but de réduire la quantité d'information transmise au *Sink.*, les ClusterHeads agrègent les données capturées par les nœuds membres qui appartiennent à leur propre Cluster, et envoient un paquet agrégé au *Sink.* Le protocole LEACH s'exécute en deux phases : comme nous pouvons le voir sur la figure 6.3.

- *Phase « Set-up ou construction»*, les ClusterHeads sont sélectionnés et les clusters sont formés, le processus d'élection des ClusterHeads est déclenché pour choisir le futur

[2]Est un algorithme d'ordonnancement qui attribue des tranches de temps à chaque processus en proportion égale, sans accorder de priorité aux processus.

ClusterHead. Ainsi, une fraction prédéterminée de nœuds s'élisent comme ClusterHeads selon le schéma d'exécution suivant : durant une période T, un nœud n choisit un nombre aléatoire nb dont la valeur est comprise entre 0 et 1 (0 < nb < 1). Si nb est inférieure à une valeur seuil, le nœud n deviendra ClusterHead durant la période courante, sinon le nœud n devra rejoindre le ClusterHead le plus proche dans son voisinage.

- *Phase « Steady-state ou communication»*, les communications à l'intérieur d'un cluster sont effectuées avec la méthode TDMA[3] (Time Division Multiple Access) où chaque chef du cluster établit un *Schedule TDMA* pour les membres de son cluster en indiquant pour chaque nœud son slot d'émission. Les membres émettent donc les données capturées pendant ce slot d'émission. Ce mécanisme permet aux différents capteurs d'éteindre leur interface de communication en dehors de leur slot d'émission, ce qui a pour effet d'économiser leur énergie. Ensuite les données agrégées (dans les ClusterHeads) sont envoyées directement au *Sink*.

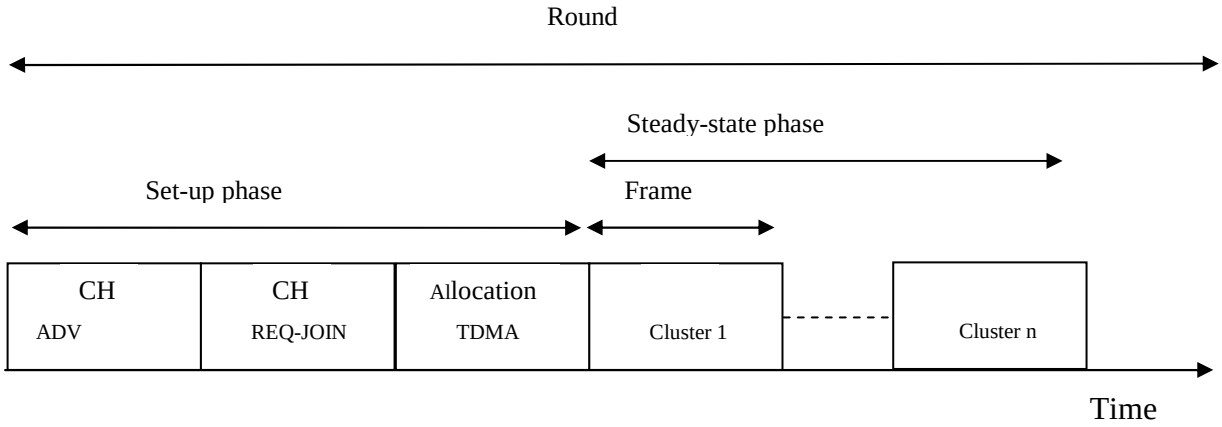


Figure 6.3 – Formation des Clusters dans le protocole LEACH

Bien que le protocole LEACH augmente la durée de vie du réseau, il présente certains inconvénients cités ci-dessous

- Le protocole LEACH suppose que tous les nœuds transmettent des données avec une grande puissance pour atteindre le *Sin*k et que chaque nœud possède une puissance de calcul lui permettant de supporter différentes couches MAC. En outre, LEACH choisit aléatoirement la liste des ClusterHeads et ne pose aucune contrainte sur leur distribution ainsi que sur leur niveau d'énergie. Ainsi, les ClusterHeads peuvent se concentrer dans un même endroit et par conséquent, il pourrait y avoir des nœuds isolés (sans ClusterHead) pouvant se déclarer.
- L'agrégation des données est centralisée et est exécutée périodiquement dans ce protocole. Or, dans certains cas, la transmission périodique des données pourrait ne pas être nécessaire, ce qui épuise rapidement l'énergie limitée des capteurs.
-

[3]Mode permettant de transmettre plusieurs signaux sur un seul canal

6.2.4.2 Modification et adaptation du protocole LEACH à notre travail

Bien que LEACH soit le protocole de routage le plus adapté à notre travail, les inconvénients que nous venons de citer ne peuvent pas être négligés, par exemple :

- ***Inconvénient 1*** : LEACH suppose que tous les nœuds transmettent des données avec une grande puissance pour atteindre le *Sink* alors que dans notre cas le *Sink* n'est pas toujours en contact avec tous les capteurs. Si les ClusterHeads envoient les données directement elles seront perdues.
- ***Solution1*** *à l'inconvénient 1* : nous avons effectué des modifications de telle sorte que les ClusterHeads n'envoient pas leurs données au *Sink* directement, mais attendent l'arrivée du *Sink* pour lui transmettre leurs données.

La détection des nœuds n'est possible que dans la phase de transmission, pour cela nous avons utilisé la technique du Hello Message dont le principe est le suivant :

1. Le *Sink* envoie le HELLO message.
2. Un ClusterHead le reçoit et lui envoie un ACK_HELLO pour informer le *Sink* qu'il est prêt à envoyer les données.
3. Le *Sink* se met en mode réception pour recevoir les données du ClusterHead.

- ***Inconvénient 2*** : les ClusterHeads peuvent se concentrer dans un même endroit et par conséquent, il pourrait exister des nœuds isolés (sans ClusterHeads) pouvant se déclarer. Dans ce sens, la trajectoire du *Sink* ne va pas couvrir toute la zone de déploiement.
- ***Solution 2*** *à l'inconvénient 2* : nous avons établi notre propre méthode de désignation des ClusterHeads qui se déroule comme suit :Chaque nœud tente de devenir un ClusterHead, alors il lance un timer (intervalle de temps) d'une durée aléatoire entre 0 et 1s; s'il a consommé son timer, il devient ClusterHead et envoie en broadcast son ADV, par contre s'il reçoit un ADV d'un autre nœud (un nœud voisin qui a consommé son timer avant lui) alors il doit annuler son timer (abandonner la tentative de devenir ClusterHead) et envoyer un REQ-JOIN au ClusterHead dont il a reçu l'ADV.

Afin de bien étudier l'impact de mobilité, nous avons utilisé le protocole LEACH sans la notion d'agrégation, c'est-à-dire les ClusterHeads qui ont reçu des données, les retransmettent au *Sink* sans les agréger (telles qu'elles sont).

6.2.5 Fonctionnement du modèle proposé

1. Formation des Clusters.
2. Une fois les clusters formés, le *Sink* parcourt le réseau pour découvrir les Clusters.
3. Transmission des positions et des priorités de chaque ClusterHead au *Sink*.
4. Une fois que le *Sink* connait toutes les positions et les priorités des ClusterHeads, il parcourt tous les ClusterHeads selon un ordre préemptif tout en marquant des temps de pause qui varient en fonction de la priorité du ClusterHead.
5. Si le temps de parcours est écoulé alors retour à 1.

Les figures 6.4 et 6.5 représentent respectivement l'organigramme de fonctionnement du modèle implémenté et le diagramme de séquence du protocole LEACH modifié.

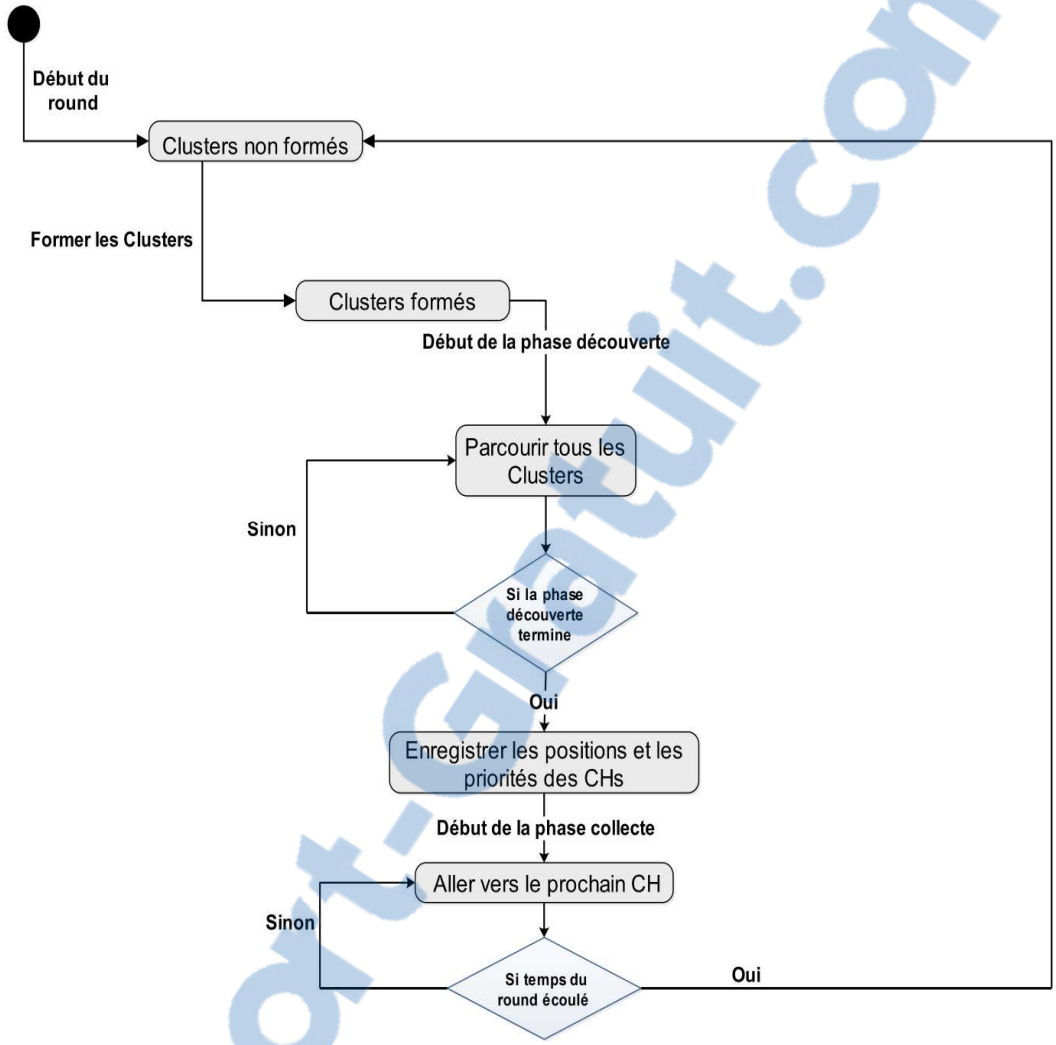


Figure 6.4 – Organigramme du modèle implémenté

Diagramme de séquence représentant le LEACH modifié

Nous allons décrire quelques fonctions représentées dans ce diagramme.

- ***Hello Message*** : représente le message que le *Sink* diffuse régulièrement pour informer les autres nœuds de son arrivée.
- ***Duty-Cycle*** : temps pendant lequel le capteur est en mode sommeil.
- ***TDMA*** : un Schedule qui permet d'assigner à chaque nœud un intervalle fini de temps appelé slot.
- ***Clustering*** : permet de diviser le réseau en plusieurs groupes de capteurs, où chaque groupe est représenté par un nœud chef appelé ClusterHead. Les capteurs communiquent avec leur ClusterHead, ensuite ce dernier communique avec le *Sink*.
- ***Timer*** : une minuterie qu'un nœud lance et lorsque le temps attribué est consommé, un évènement se déclenche.

- ***Round*** : c'est le temps pendant lequel les Clusters restent formés et que les ClusterHeads gardent leurs statuts. A la fin de ce temps, le processus reprend (un nouveau ClusterHead sera élu et de nouveaux Clusters seront formés).
- ***ADV*** : c'est un paquet qu'un nœud envoie en broadcast pour informer les autres nœuds qu'il est le ClusterHead.
- ***REQ-JOIN*** : c'est un paquet qu'un nœud non ClusterHead envoie à un nœud ClusterHead pour lui demander de faire partie de son Cluster.

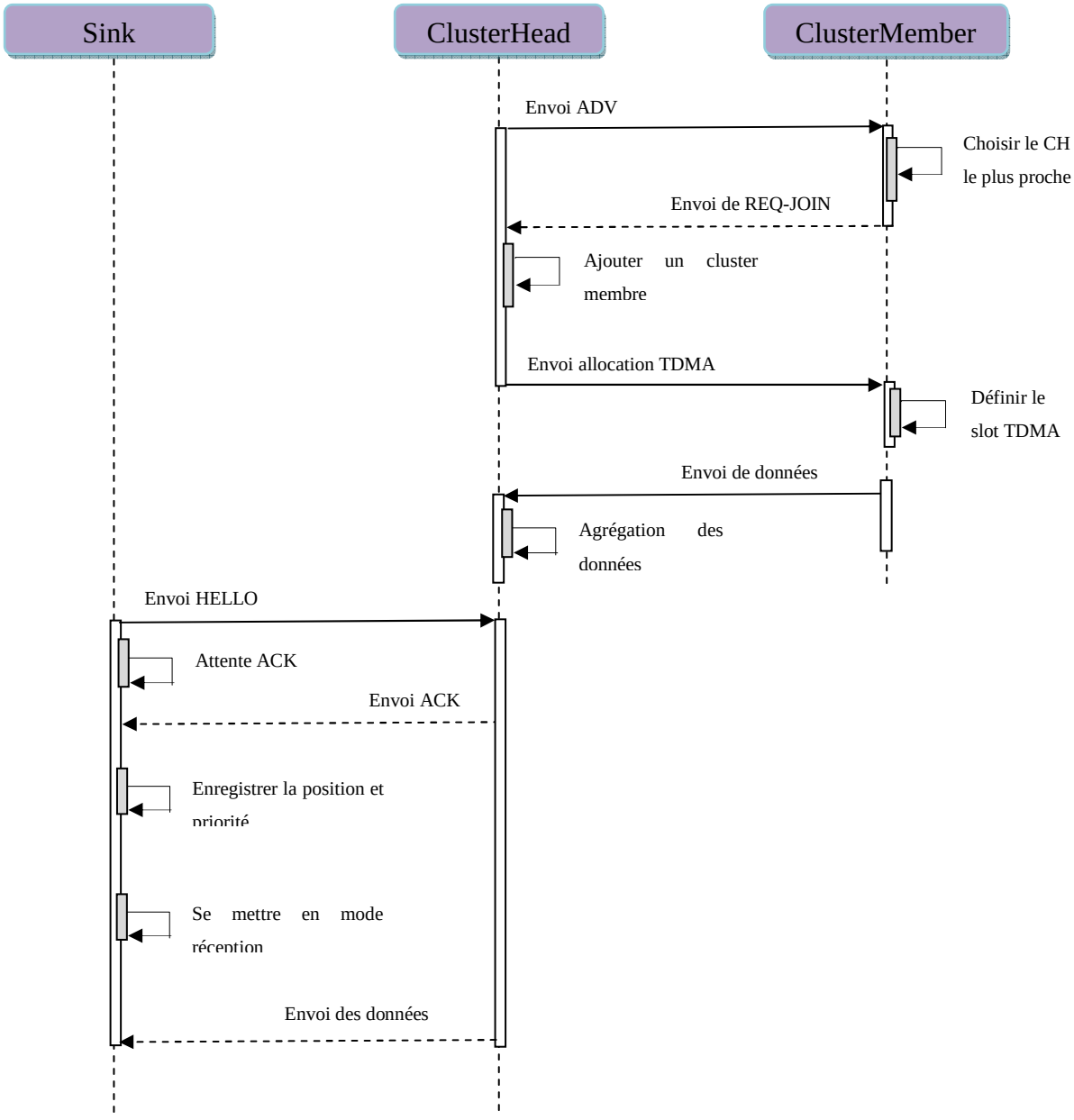


Figure 6.5 – Diagramme de séquence :Sink, ClusterHead et ClusterMember

Afin d'évaluer notre modèle de mobilité, nous l'avons comparé à d'autres modèles que nous avons implémentés sous Castalia.

6.2.6 Implémentation et adaptation du modèle « RandomWayPoint»

Dans le chapitre 5, les résultats ont montré que le modèle de mobilité « *RandomWayPoint* » donnait de meilleurs résultats que les autres modèles étudiés, c'est pour cela que nous allons utiliser ce modèle pour réaliser notre comparaison. Pour voir les détails sur l'algorithme de mobilité « *RandomWayPoint*», se référer à la section 5.3.4 du chapitre 5. Au début de la simulation, les paramètres de déplacement du *Sink* tels que la vitesse et le temps de pause sont fixes. En adaptant ce modèle à notre travail et dans le but de le rendre paramétrable, nous avons fait varier le temps de pause et/ou la vitesse du *Sink* pendant la simulation. A chaque fois que le *Sink* atteint une destination intermédiaire, il choisit une nouvelle vitesse et/ou un nouveau temps de pause pour son prochain départ.

Le ***modèle*** « *RandomWayPoint* » est un modèle de mobilité non déterministe (trajectoire aléatoire), donc nous ne pouvons pas prédire la trajectoire du *Sink*. Il se peut que le *Sink* ne parcoure pas tous les nœuds du réseau, ce qui nous conduit à notre deuxième problématique, qui est de trouver une façon de fixer la trajectoire du *Sink* de telle sorte qu'il parcourt tous les nœuds du réseau.

6.2.7 Implémentation du modèle de mobilité « GridMobility »

Dans ce modèle, nous avons représenté le réseau sous forme de grille (divisée en lignes et en colonnes) pour déterminer la trajectoire du *Sink*. Dans un premier temps, nous remplissons la grille avec les positions où le *Sink* doit se rendre. Ensuite le *Sink* parcourt la grille de façon circulaire comme nous pouvons le voir sur la figure 6.6.

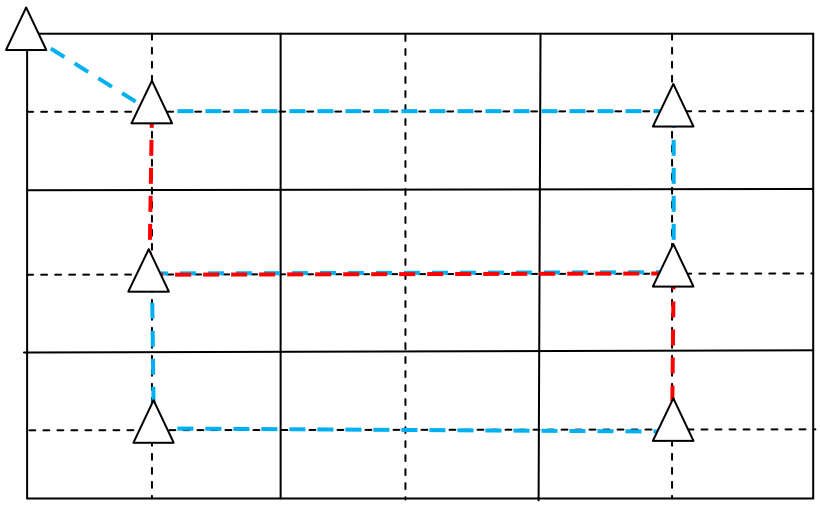


Figure 6.6 – Modèle de mobilité « GridMobility »

L'algorithme « GridMobility » est proposé pour modéliser le mouvement des nœuds sur les routes d'un segment de ville. Par exemple au début, chaque nœud choisit un point dans une route quelconque pour commencer. Puis il choisit une destination aléatoire et prend une direction vers cette destination, sans vitesse définie au préalable. Lorsqu'un nœud mobile atteint sa destination, il reprend le processus.

Algorithme1« GridMobility »

```
1: début
2:lire la vitesse et le temps de pause du Sink ;
3:lire la taille du réseau sur les axes «X» et «Y»;
4:lire toute les colonnes et les lignes de la grille;
5: Pa_x = (taille du réseau_X / (NColones -1))/2;
6: Pa_y = (taille du réseau_Y / (numLignes -1))/2;
7: Aller à (Pa_x,Pa_y) ;
8: Créer le tableau TableDestination[((numLignes - 1) * 4)-1] d'entiers ;
9: Remplir ce tableau avec les différents endroits où doit se rendre le Sink
10: Index = 0;
11:Arrêt = faux;
12:tantque (!Arrêt)
13:faire
14: Nouvelle Destination X = TableDestination[X] ;
15: Nouvelle Destination Y = TableDestination[Y] ;
16: Aller à la nouvelle destination sur les axes «X» et «Y»;
17: Localisation du Nœud «X» = Nouvelle Destination de «X» ;
18  Localisation du Nœud «Y» = Nouvelle Destination de «Y»;
19: Incrémenter l'index ;
20:fin tant que
21: fin
```

6.2.8 Outil de visualisation pour le simulateur Castalia « CastaliaViz »

Contrairement aux logiciels NS (à titre d'exemple) qui sont fournis avec une interface graphique nommée NAM (Network Animator Manager) qui permet de visualiser le fonctionnement du réseau (mobilité, communication,…etc.).Certains simulateurs ne disposent pas d'un tel outil. Ils génèrent seulement un ou plusieurs fichiers trace qui contiennent les évènements qui se sont déroulés lors de la simulation, le simulateur Castalia en fait partie. Nous avons ajouté quelques classes au simulateur Castalia pour générer des fichiers formatés, l'outil va lire ces fichiers pour faire la visualisation des évènements qui se déroule lors de la simulation. La façon dont l'outil est lancé est la même que pour le simulateur.

Une fois ces modifications faites dans Castalia, il suffit juste de mettre le fichier sous forme exécutable (.jar) pour ensuite l'appeler à partir d'un script python (comme les scriptes Castalia, CastaliaResults et CastaliaPlot) que nous avons placé dans le Bin de Castalia.

Voici le fichier python qui lance l'animation, nous l'avons appelé « CastaliaViz ».

```
1. #!/usr/bin/python
2. import subprocess, os, sys
3.
4. subprocess.call(['java', '-jar', '/home/derdour/Castalia-3.3/bin/CaVis.jar'])
```

La figure 6.7 présente un aperçu de cet outil.

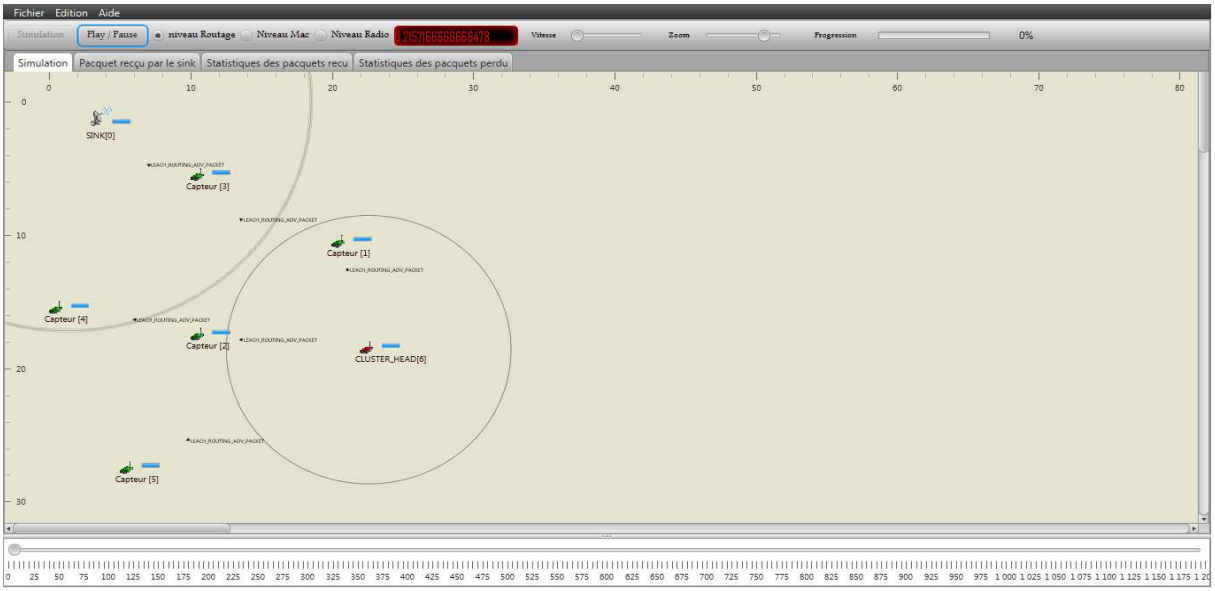


Figure 6.7 – Aperçu de l'outil de visualisation implémenté

Après le déroulement d'une simulation, il faut taper la commande « CastaliaViz » dans le terminal pour lancer l'animation.

6.2.9 Simulation et résultats préliminaires

6.2.9.1 Environnement de simulation

Pour cette contribution, nous avons opté pour le simulateur Castalia qui est un simulateur dédié aux RCSFs, aux réseaux BAN (BAN : Body Area Networks) et généralement des réseaux de systèmes embarqués à faible puissance. Ce simulateur est développé par NICTA (National ICT Australia), publié pour la première fois en 2007, dédié aux RCSFs, supportant la mobilité et permettant l'intégration et la modification de modules. La version de Castalia que nous avons utilisée est la version 3.3 (sortie en 2013).
Castalia est basé sur la plateforme OMNeT++ et peut être utilisé pour tester des algorithmes distribués et/ou des protocoles de communication sans fil, avec un comportement réaliste, surtout en ce qui concerne l'accès à la couche radio.

Le simulateur Castalia offre la possibilité de manipuler différentes couches du modèle OSI, la couche MAC (accès au support de communication), la couche réseau (routage des paquets) et la couche application, permettant de créer des réseaux avec nœuds statiques.

6.2.9.2 Description du réseau

Notre réseau est représenté par un espace à deux dimensions (500m x 200m) de la zone de déploiement et nous avons considéré les suppositions suivantes:

Supposition 1: le *Sink* est le seul nœud doté d'un sous-système supplémentaire, celui de la mobilité.
Supposition 2: la mobilité du *Sink* est contrôlée.
Supposition 3: le module de mobilité permet au *Sink* de se déplacer dans le réseau (déplacement aérien ou sur une surface sans obstacle).

Supposition 4: tous les capteurs ont la même quantité d'énergie. Initialement, la quantité d'énergie du nœud proposée par Castalia est par défaut 18720 Joules, elle correspond à deux piles AA.

Supposition 5: la collecte de données se fait d'une manière périodique.

Ces suppositions restent valables dans les différents scénarios réalisés dans ce travail. Après le déroulement des simulations, nous utilisons le ScriptCastaliaResults[4] pour calculer les différentes métriques de performance (Energie consommée, Perte de messages ainsi que la Latence).

Nous avons utilisé les mêmes formules pour les métriques considérées que celle décrites dans le chapitre 4.

6.2.9.3 Métriques de performance

- La perte de messages de données n'est pas fournie par Castalia, car ce paramètre n'est donné qu'au niveau physique (Radio). Alors, nous avons réalisé nos propres algorithmes pour calculer les messages perdus au niveau de la couche physique.
- Latence (d'un message) qui égale au temps de réception du message par le *Sink* moins le temps où le message a été généré. Pour réaliser cela nous avons ajouté un champ au paquet, application indiquant le temps de création du message. Ces métriques ont été vues en détail dans le chapitre 4.

6.2.9.4 Scénarios de simulation

Nous avons réalisé plusieurs scénarios afin d'évaluer les performances du modèle implémenté et optimiser la trajectoire de déplacement du *Sink* en considérant plusieurs métriques de performance (Consommation d'énergie, le nombre de paquets perdus, reçus ainsi que la latence).

- **Scénarios S1,S2** : dans cette partie, nous avons simulé deux scénarios, le premier scénario ***(S1)*** est réalisé avec le principe du Clustering et le deuxième ***(S2)*** avec une topologie plate. Afin de pouvoir comparer les deux topologies et expliquer notre intérêt pour le Clustering et son impact sur la collecte de données. La tables 6.1 présente les détails relatifs aux scénarios S1,S2 et la table 6.2 résume les paramètres de simulation leurs correspondants.

	Communication		***Mobilité***	
	Type	Protocole	Type	Modèle
S1(Clustering)	2 sauts (CM -> CH ->Sink)	LEACH	Contrôlée	Notre modèle
S2 (Plat)	1 saut (C ->Sink)	ShRouting[5]	Aléatoire	RandomWayPoint

Table 6.1 – Scénarios (S1,S2)

[4] Permet de voir les résultats de simulation (paquets reçus, paquets perdus, qualité du signal, les interférences…
[5]SingleHopRouting :Routage à un saut

	Vitesse	Temps de pause	Duty-Cycle	Déploiement	Temps de simulation	Nombre de capteurs
S1	15mps	Adaptatif	TDMA	Uniforme	800s	24 + Sink
S2	15mps	1s	75 %	Uniforme	800s	24 + Sink

Table 6.2 – Paramètres de simulation (S1,S2)

Dans le but de réduire les déplacements du *Sink.*, nous avons divisé le réseau en Clusters c'est-à dire en un ensemble de capteurs proches les uns des autres et à leur tête un CH comme le montre la figure 6.8 qui représente le déploiement des Clusters au sein du réseau (S1).

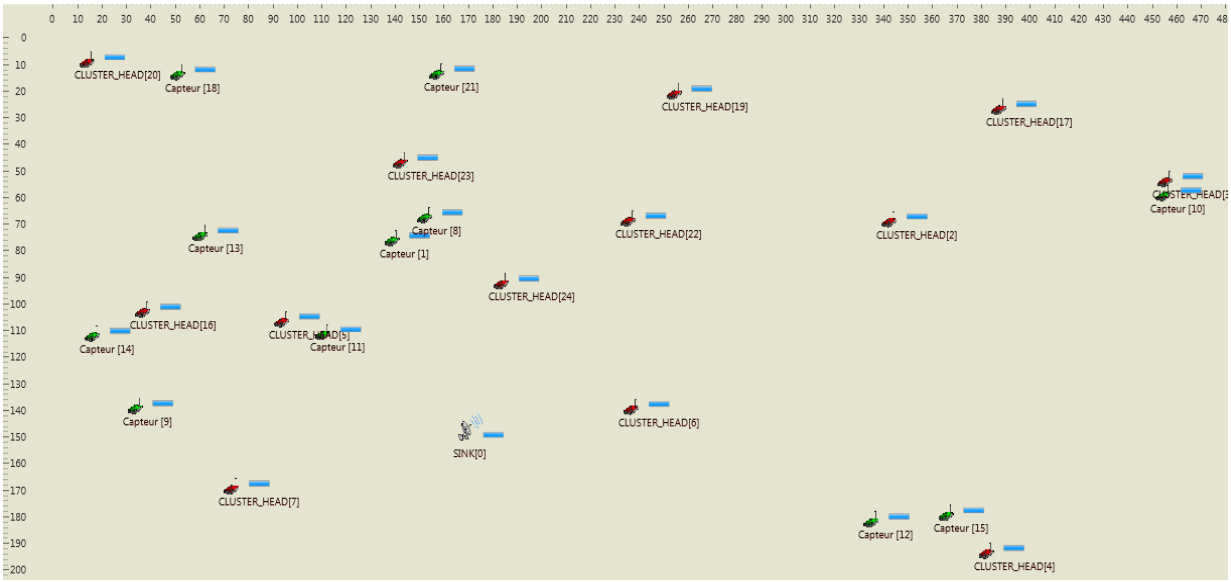


Figure 6.8 – Déploiement des Clusters au sein du réseau (S1)

6.2.10 Evaluation des résultats de simulation (S1, S2)

La table 6.3 présente les résultats obtenus dans les deux scénarios S1 et S2

	Scénario 1	***Scénario 2***
Paquets générés par la couche application	2400	2400
Paquets envoyés	2400	2400
Paquets reçus par le Sink	2399	2114
Paquets perdus	1	286
Energie consommée par le réseau	175 joules	196 joules
Latence	37.74s	139.36s
Trajectoire parcourue par le Sink	46694.9m	49729.4m

Table 6.3 – Résultats de simulation (S1,S2)

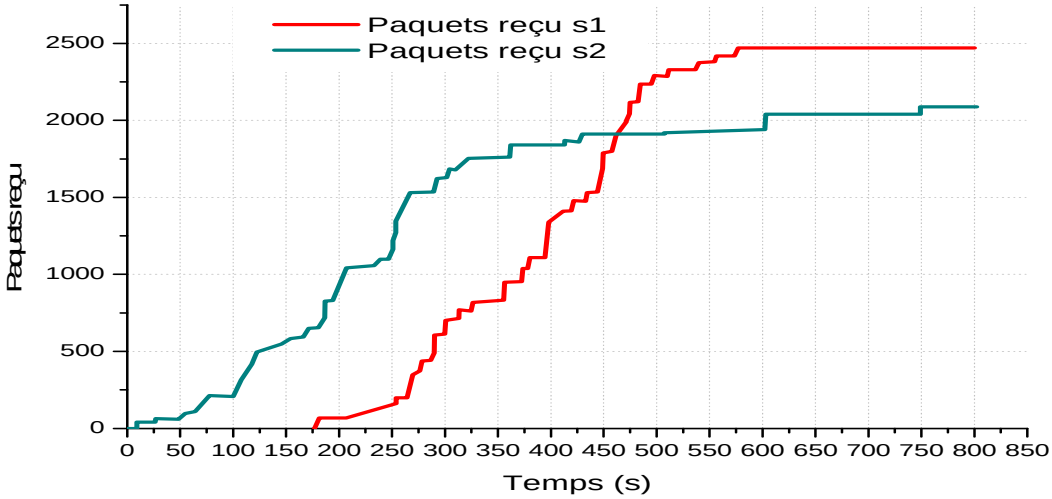


Figure 6.9 – Paquets de données reçus par le Sink (S1,S2)

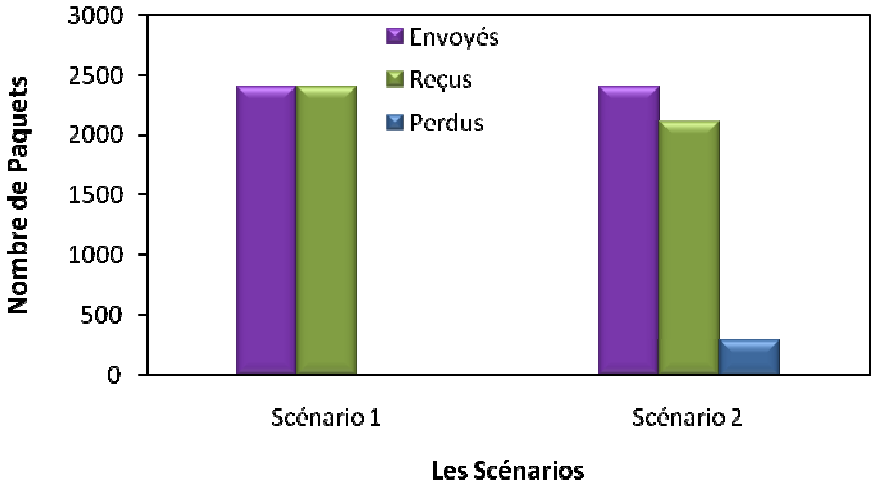


Figure 6.10 – Comparaison entre S1et S2 par rapport au nombre de paquets

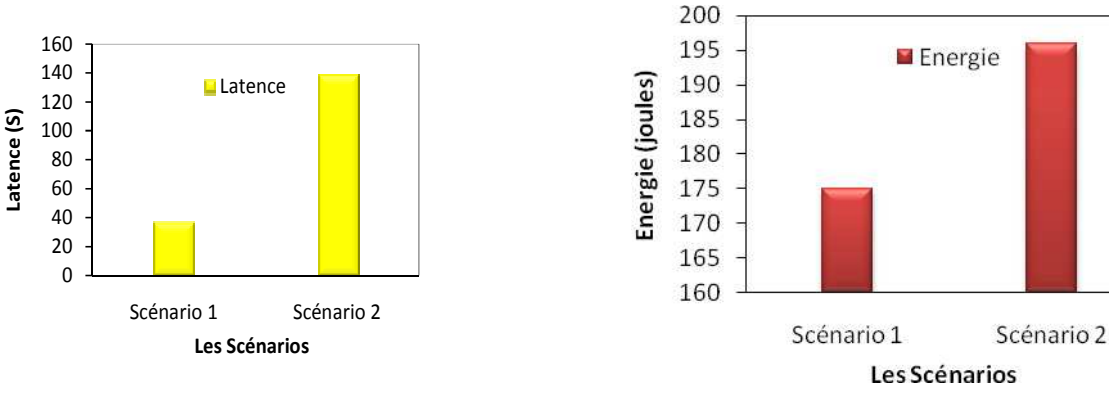


Figure 6.11 – Comparaison entre S1et S2 par rapport à la latence et à l'énergie

Nous constatons d'après les figures 6.9, 6.10 et 6.11, que le scénario S1 avec principe de Clustering donne de meilleurs résultats que le scénario S2 avec routage plat en termes de nombre de messages perdus, d'énergie et de temps de latence.

- ***Perte de messages*** : dans les RCSFs-EMs, le *Sink* reçoit les données des nœuds capteurs pendant son déplacement et surtout lorsqu'il sort de la portée radio de l'émetteur, c'est le cas du scenario2. Par conséquent, le taux de réception baisse (88%), contrairement au cas du scenario1 où le *Sink* marque un temps de pause pour réceptionner les données et donc le taux de réception est plus élevé (99.98%). Nous constatons aussi que la réception des données dans S1 commence à l'instant 175s, c'est le temps nécessaire au *Sink* pour terminer son premier tour (phase de découverte du réseau).
- ***Energie consommée*** : le protocole LEACH est basé sur le TDMA entre le ClusterHead et ces membres, donc un nœud ne reste éveillé que pendant le temps qu'il lui est attribué par le ClusterHead, ce qui permet d'économiser de l'énergie plus que le Duty-Cycle classique, utilisé dans le scénario2.
- ***Latence*** : le mouvement aléatoire du *Sink* dans S2 ne permet pas une collecte immédiate, ce qui augmente considérablement le temps de latence, contrairement au S1où le *Sink* se dirige directement vers les CHs pour collecter les données. Dans S1, la collecte dure 417s alors que dans S2 elle dure 745s, ceci explique le temps de latence obtenu.
- ***Trajectoire parcourue par le Sink :*** nous remarquons selon les résultats représentés dans la table 6.3 que la trajectoire parcoure est plus grande dans le scénario (S2 : topologie plate) où le *Sink* est obligé de parcourir tous les nœuds du réseau contrairement au scénario (S1 : topologie en Clusters), le *Sink* ne se déplace que vers les CHs pour collecter les données et donc sa trajectoire est réduite.

Ces résultats révèlent que la collecte de données avec une topologie de Clustering donne de meilleurs résultats. Ainsi, nous utiliserons cette topologie dans les prochains scénarios.

- **Scénarios S3, S4, S5 :** en seconde partie, nous étudions l'impact et les performances du modèle de mobilité avec une topologie de Clustering en utilisant la mobilité contrôlée. Nous proposons trois scénarios S3,S4 et S5, dans chacun d'eux nous faisons varier le type de mobilité (contrôlée, aléatoire et déterministe) et le modèle de mobilité, comme illustré dans la table 6.4.

	Communication		***Mobilité***	
	Type	Protocole	Type	Modèle
S3	2 sauts (CM -> CH ->Sink)	LEACH	Contrôlée	Notre modèle
S4			Aléatoire	RandomWayPoint
S5			Déterministe	GridMobility

Table 6.4 – Scenarios S3,S4 et S5

La table 6.5 présente les paramètres correspondants à ces trois scénarios.

	Vitesse	Temps de pause	DutyCycle	Déploiement	Temps de simulation	Nombre de capteurs
S3	15mps	Adaptatif	TDMA	En cluster	800s	48 + Sink
S4	15mps	1s	TDMA	En cluster	800s	48 + Sink
S5	15mps	1s	TDMA	En cluster	800s	48 + Sink

Table 6.5 – Paramètres de simulation (S3,S4 et S5)

Priorité des ClusterHeads

Une fois que les Clusters et les ClustersHeads sont formés, nous avons attribué à chaque ClusterHead une priorité, comme nous pouvons le voir sur la table 6.6 et la figure 6.12.

CH6	CH8	CH15	CH22	CH29	CH33	CH38	CH43	CH46
2	6	1	5	3	2	4	3	2

Table 6.6 – Priorités attribuées à chaque Cluster

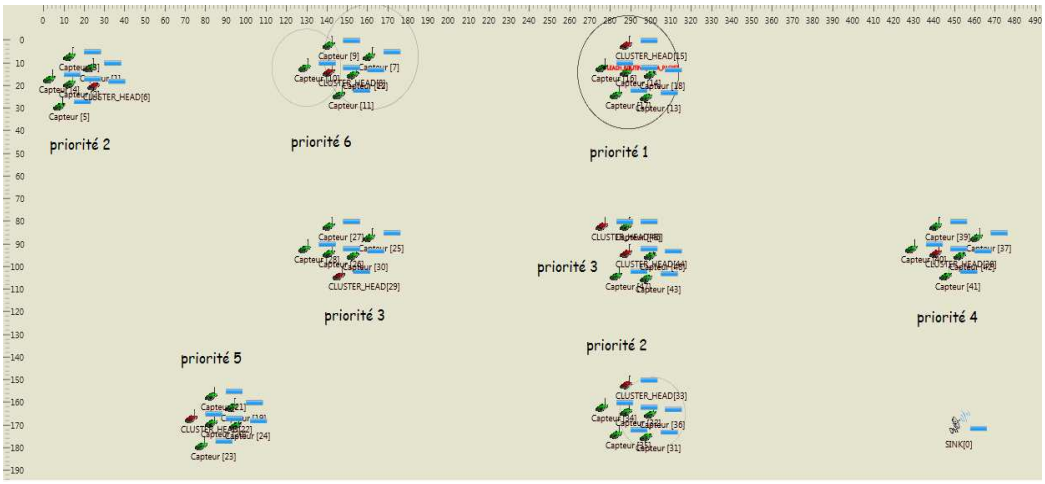


Figure 6.12 – Déploiement des Clusters au sein du réseau (S3,S4 et S5)

6.2.11 Evaluation des résultats de simulation (S3,S4 et S5)

La table 6.7 illustre les différents résultats de simulation des scénarios S3,S4 et S5.

		Les différents scénarios		
		S3	S4	S5
Les différents résultats	Paquets générés par la couche application	4772	4772	4772
	Paquets envoyés	4073	3772	4293
	Paquets reçus par le Sink	4072	3472	3978
	Paquets perdus	1	300	315

	Paquets encore stockés dans le buffer	695	996	475
	Energie consommée par le réseau	25.42 joules	25.14 joules	25.40 joules
	Latence	96.8s	153.69s	134.71s
	Trajectoire parcourue par le Sink	8103.9m	11981.4m	11983.8m

Table 6.7 – Résultats de simulation (S3,S4 et S5)

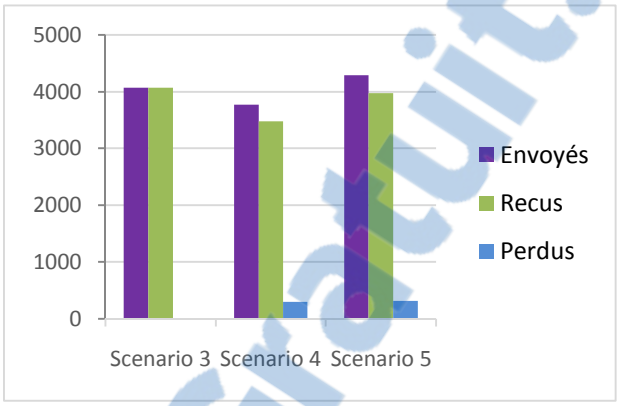


Figure 6.13 – Comparaison entre S3, S4 et S5 par rapport au nombre de paquets

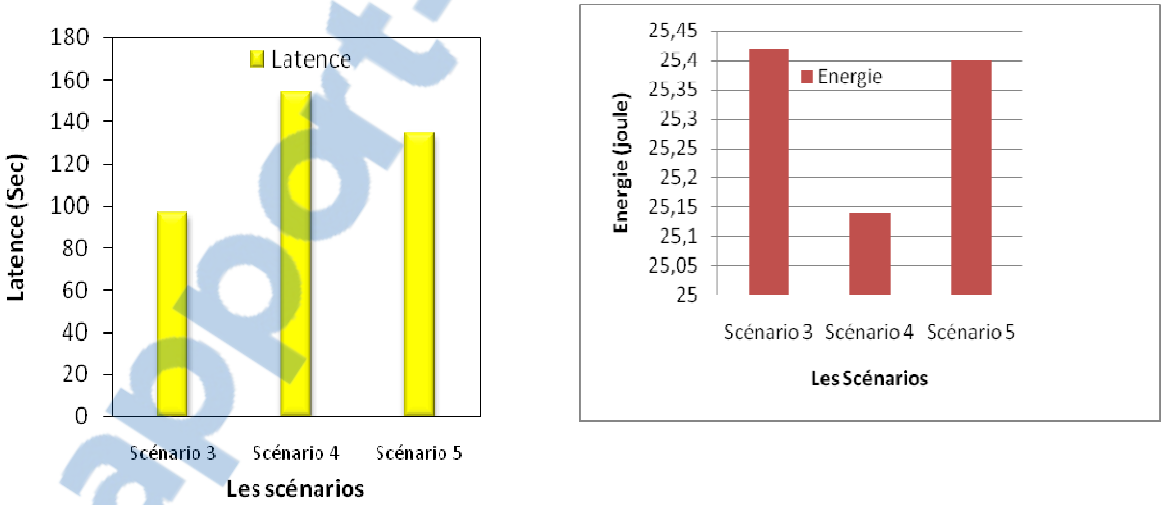


Figure 6.14 – Comparaison entre S3, S4 et S5 par rapport à la latence et à l'énergie

Dans les trois scenarios, nous constatons sur la figure 6.13que la mobilité contrôlée avec notre propre modèle de mobilité (S3) donne de meilleurs résultats dans une collecte avec un minimum de perte. Ceci est l'un des avantages de ce type de mobilité. En effet le *Sink* ne reçoit les données que lorsqu'il est en pause, et même s'il entre en contact avec d'autres CHs durant son trajet. Il ne communique pas avec eux, il se dirige directement vers le CH le plus prioritaire.

Cependant, avec les autres types et modèles de mobilité (S4,S5), le *Sink* peut recevoir les paquets de données pendant son déplacement; arrivé à un certain seuil il commence à perdre ces paquets. Nous remarquons aussi un temps de latence plus élevé pour les mêmes raisons déjà expliquées précédemment.

- **Scénario S6 (Impact de la densité)** : dans cette troisième et dernière partie, nous allons étudier l'impact de la densité des nœuds dans notre réseau. Pour cela, nous avons proposé un sixième scénario où nous avons fait varier le nombre de nœuds de 25-150 nœuds.

Les résultats sont présentés dans la table 6.8 :

	25 nœuds	50 nœuds	75 nœuds	100 nœuds	125 nœuds	150 nœuds
Taux de réception	99,95%	99,57%	99,83%	99,58%	97,73%	97,51%
Latence (s)	37,47s	76,63s	112,77s	149,46s	188,41s	225,16s
Energie (J)	175,89 j	154,36 j	104,05 j	87,49 j	77,36 j	77,95 j

Table 6.8 – Résultats de simulation (S6)

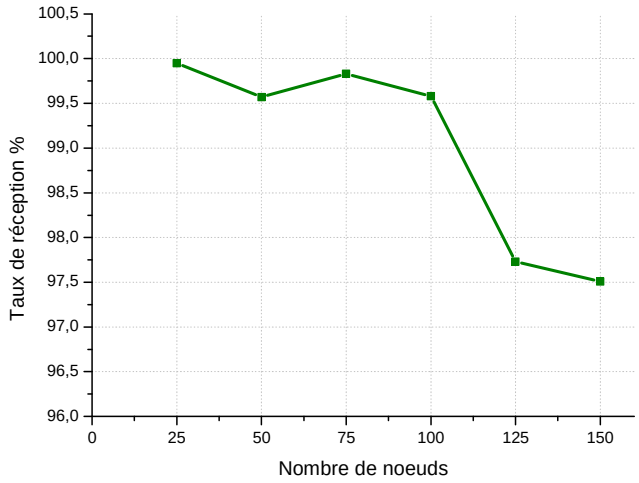


Figure 6.15 – Taux de réception par apport à la densité des nœuds

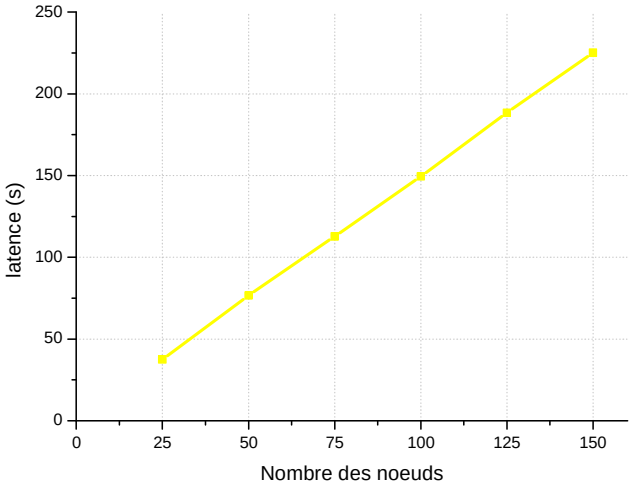


Figure 6.16 – Latence par apport à la densité des nœuds

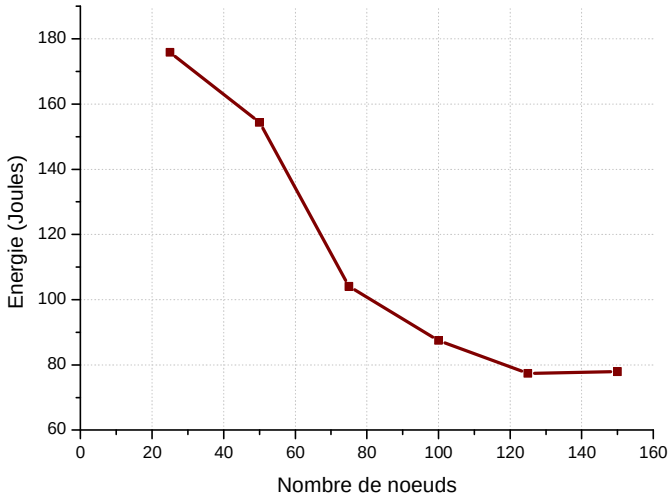


Figure 6.17 – Consommation d'énergie par rapport à la densité des nœuds

6.2.12 Evaluation des résultats de simulation (S6)

Nous remarquons sur la figure 6.15, que le taux de réception des paquets diminue légèrement dans les deux derniers scenarios, ceci s'explique par la congestion des paquets au niveau inter-Cluster.

Cependant sur la figure 6.16, nous remarquons que la densité a un effet négatif sur le temps de latence. En effet, en augmentant le nombre de nœuds, les CHs augmentent et donc plus de données à transmettre, du coup des temps de pause plus longs, ce qui augmente le temps de latence considérablement.

Nous constatons aussi sur la figure 6.17est que lorsque le nombre de nœud augmente l'énergie consommée diminue. En effet, en augmentant la densité des nœuds, la taille des Clusters ainsi que le temps de sommeil des nœuds augmentent aussi.

6.2.13 Conclusion

Au vu des résultats obtenus dans cette première étape, nous pouvons conclure que la mobilité du *Sink* dans un RCSF apporte des avantages considérables, que ce soit au niveau de la collecte de données ou bien en termes d'économie d'énergie. Nous avons implémenté notre propre modèle de mobilité, basé sur le protocole de routage hiérarchique LEACH (Clustering) dans Castalia que nous avons modifié pour nos propres besoins afin de déterminer et d'optimiser la trajectoire du *Sink*.

Les résultats de simulation révèlent que le Clustering avec TDMA offre un gain considérable au niveau de l'économie d'énergie par rapport aux approches de Duty-Cycle avec topologie plate. Ainsi, la mobilité contrôlée du *Sink* améliore nettement la collecte de données et minimise la perte de paquets.

Pour pouvoir visualiser toutes ces simulations dans Castalia, nous avons été contraints d'implémenter un outil de visualisation qui n'existait pas auparavant. Cet outil nous a permis d'avoir une meilleure vue du réseau ainsi que les interactions qui s'y produisent.

Nous cherchons maintenant à améliorer formellement notre solution dans cette deuxième étape. Pour cela, nous allons utiliser une méthode de résolution approchée (Méta-heuristique) pour optimiser la trajectoire de déplacement du *Sink*. Nous allons présenter, dans ce qui suit, quelques définitions des méthodes de résolution ainsi qu'une classification de ces méthodes en mettant l'accent sur les Méta-heuristiques.

Lors de nos précédentes simulations, nous avons fixé la trajectoire du *Sink* pendant le processus de collecte. Cette approche nous a permis de trouver une trajectoire optimale mais la question qui se pose est :« Est ce la meilleure trajectoire » ?

Aussi, pour répondre à cette question et confirmer nos résultats, nous avons jugé utile de choisir la trajectoire de mobilité du *Sink* en utilisant les techniques d'optimisation pour au moins se rapprocher de l'optimum.

Dans les sections suivantes, nous allons présenter cette technique et nous donnons une classification de ces méthodes de résolution exactes et approchées.

Comme les méthodes exactes posent un problème d'explosion combinatoire (facteur d'échelle), nous avons opté pour les méthodes approchées. Parmi celles-ci, nous avons choisi deux méthodes « la Recherche tabou et le Recuit simulé » qui sont en adéquation avec notre problématique. Nous présenterons par la suite les algorithmes correspondants à ces deux techniques.

6.3 Approche basée sur les méthodes formelles

6.3.1 Méthodes de résolution approchées

Les méthodes dites approchées sont utilisées lorsqu'il n'est pas nécessaire d'obtenir des solutions forcément exactes, pour des problèmes de grande taille et pendant des temps restreints. Les méthodes approchées peuvent être divisées en deux grandes familles : les

heuristiques et les méthodes d'approximation. La différence entre ces deux méthodes vient du fait que les méthodes d'approximation assurent une garantie des solutions obtenues par rapport à l'optimum global, alors que les heuristiques ne donnent aucune indication sur cet aspect-là. Ces dernières peuvent être divisées à leur tour en deux classes : les méta-heuristiques qui sont des méthodes génériques non dédiées à un problème spécifique et les heuristiques qui sont dédiées et qui concernent des méthodes spécifiques à un problème donné. La notion d'heuristique date de 1945 [201] et les premières méta-heuristiques, les algorithmes évolutionnaires et les premières recherches locales, datent des années 60. Depuis cette époque, les méta-heuristiques rencontrent un succès fulgurant et ont vu leur nombre se multiplier. Cela est dû à leur capacité à résoudre des problèmes de taille importante pendant des temps raisonnables en apportant des solutions de qualité satisfaisante. Il est bien sûr possible d'obtenir des solutions de meilleure qualité mais cela ne peut se faire qu'au détriment du coût de la méthode utilisée. Deux grandes familles de méta-heuristiques existent : les méta-heuristiques à base de solution unique et celles à base de population de solutions. Les premières consistent en une amélioration d'une seule solution existante tant que cela est possible, ce sont donc des méthodes avec un fort pouvoir d'intensification. Cependant, les méta-heuristiques, à base de population, permettent de mieux appréhender les grands espaces de la recherche de part leur pouvoir de diversification. Même si les méta-heuristiques sont applicables sur n'importe quel type de problème, il convient toujours d'analyser les caractéristiques du problème à traiter pour évaluer la meilleure méthode à utiliser [201].

La figure 6.18 présente une classification des méthodes de résolution les plus utilisées.

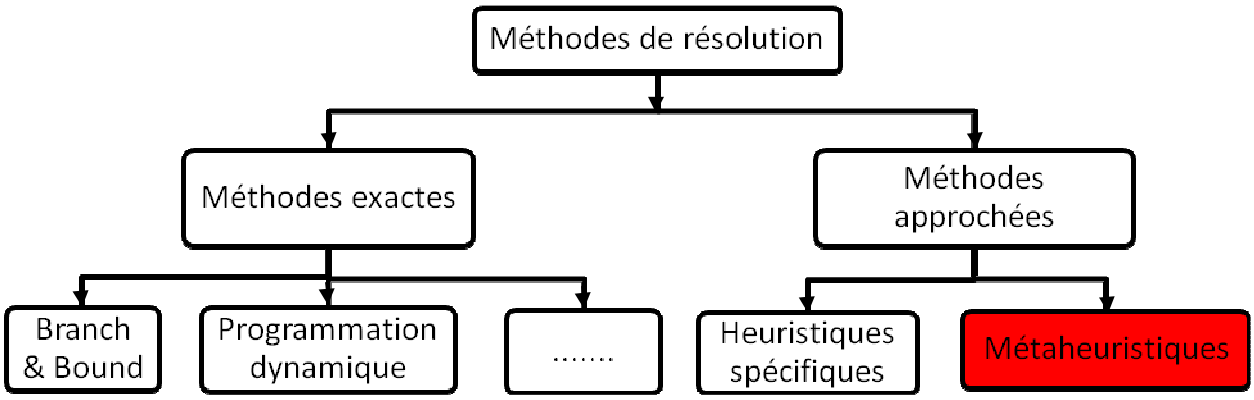


Figure 6.18 – Classes des méthodes de résolution

6.3.1.1 Méta-heuristiques

Les méta-heuristiques forment un ensemble de méthodes utilisées en recherche opérationnelle et en intelligence artificielle pour résoudre des problèmes d'optimisation, considérés difficiles. Résoudre un problème d'optimisation combinatoire, revient à trouver l'optimum d'une fonction, parmi un nombre fini de choix, souvent très grand. Les applications concrètes sont nombreuses, que ce soit dans le domaine de la production industrielle, des transports ou

de l'économie (partout où se fait sentir le besoin de minimiser des fonctions numériques, dans des systèmes où interviennent simultanément un grand nombre de paramètres).

Une heuristique est un algorithme qui a pour but de trouver une solution réalisable, sans garantie d'optimalité, contrairement aux méthodes exactes qui garantissent des solutions exactes. Comme les algorithmes de résolution exacte sont de complexité exponentielle pour les problèmes difficiles, il serait plus judicieux de faire appel aux heuristiques pour calculer une solution approchée d'un problème ou pour accélérer le processus de résolution exacte. Généralement une heuristique est conçue pour un problème particulier, mais les approches peuvent contenir des principes plus généraux. Nous parlons dans ce cas là de méta-heuristiques.

Les méta-heuristiques permettent de trouver des solutions, peut-être pas toujours optimales, mais très proches de l'optimum et en un temps raisonnable. Donc, nous distinguons des méthodes dites exactes qui garantissent certes la résolution d'un problème, mais au prix de temps de calcul prohibitif. Parmi leurs caractéristiques:

- Intensification (exploitation) : permet d'examiner en profondeur une zone particulière de l'espace de recherche.
- Diversification (ou exploration) : permet d'orienter la recherche vers de nouvelles zones (prometteuses) dans l'espace de recherche.

6.3.1.2 Classification des Méta-heuristiques

Les problèmes d'optimisation combinatoire sont souvent des problèmes très difficiles dont la résolution par les méthodes exactes peut s'avérer très longue ou peu réaliste. L'utilisation de méthodes heuristiques permet d'obtenir des solutions de bonne qualité, en un temps de résolution raisonnable. Les heuristiques sont aussi très utiles pour le développement de méthodes exactes, fondées sur des techniques d'évaluation et de séparation (Branch and Bound).

Les méta-heuristiques se composent de deux classes, celles avec une population de solutions et celles qui ne manipulent qu'une seule solution à la fois. Nous allons nous intéresser dans ce travail à la deuxième classe. Les méthodes qui tentent itérativement d'améliorer une solution sont appelées méthodes de recherche locale ou méthodes de trajectoire.

La méthode Tabou, le Recuit simulé et la Recherche locale sont des exemples typiques de méthodes de trajectoire. Ces méthodes construisent une trajectoire dans l'espace des solutions en tentant de se diriger vers des solutions optimales.

L'exemple le plus connu de méta-heuristique à méthodes avec une population de solutions est l'algorithme génétique. La figure 6.19 donne un panorama des méta-heuristiques les plus utilisées.

Cependant, notre modèle de déplacement du *Sink* s'inspire du modèle du voyageur de commerce dont l'objectif est de minimiser le chemin en démarrant d'un point initial, en passant par tous les autres points et en revenant au point de démarrage. Pour cela, nous allons utiliser pour étudier ces déplacements, les méta-heuristiques à solution unique « Recherche

Tabou et le Recuit Simulé » pour leurs efficacités en termes d'optimisation dans ce type de problèmes [202].

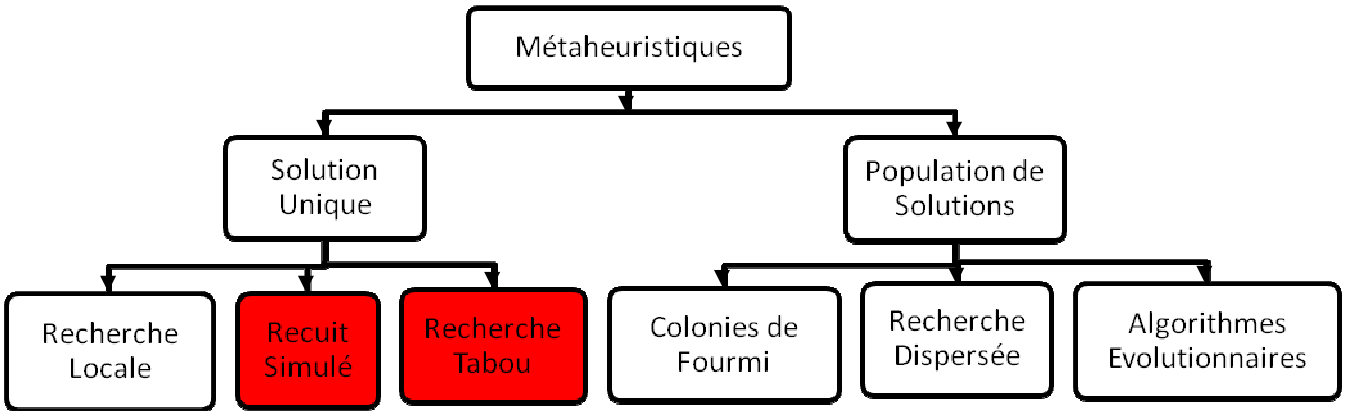


Figure 6.19 – Classes de Méta-heuristiques.

6.3.1.3 Méthodes à base de solution unique

Elles permettent de définir aisément des méthodes avec un fort pouvoir d'intensification. Ces méthodes permettent d'obtenir rapidement des solutions de bonne qualité. Depuis la recherche par descente, les méta-heuristiques à base de solution unique ont beaucoup évolué et sont capables aujourd'hui de combiner des phases de diversification, d'intensification et d'utiliser des voisinages multiples.

- ***Recherche Tabou :*** cette méthode a été proposée par Glover [201] et est devenu populaire dans les années 90. Cette méthode est une méta-heuristique itérative qualifiée de *recherche locale* au sens large. L'idée de la Recherche tabou consiste, à partir d'une position donnée, à en explorer le voisinage et à choisir la position dans ce voisinage qui minimise la fonction objectif. Il est essentiel de noter que cette opération peut conduire à l'augmentation de la valeur de la fonction (dans un problème de minimisation) ; c'est le cas lorsque tous les points du voisinage ont une valeur plus élevée. C'est à partir de ce mécanisme que l'on sort d'un minimum local. Le risque, cependant, est qu'à l'étape suivante, on retombe dans le minimum local auquel on vient d'échapper. C'est pourquoi il faut que l'heuristique ait de la mémoire : le mécanisme consiste à interdire (d'où le nom de *tabou*) de revenir sur les dernières positions explorées.

Le déroulement de cette méthode est le suivant :

1. Sélectionner comme solution le capteur le plus proche du *Sink*.
2. Déplacement du *Sink* vers ce capteur.
3. Le *Sink* cherche les capteurs qui se trouvent dans sa portée radio (Rechercher des solutions au voisinage de la solution courante « principe d'intensification »)
4. Communication entre le *Sink*, le capteur sélectionné et les capteurs qui sont dans sa portée radio (pour collecter les données).

5. Mettre à jour la liste tabou (avec les capteurs se trouvant sur le chemin) et la liste des capteurs ayant communiqué avec le *Sink*.
6. Après communication avec les capteurs qui sont dans le voisinage du capteur sélectionné, rechercher un autre capteur qui n'appartient pas à la liste et qui est plus proche du *Sink*. « Principe de Diversification » : rechercher des solutions dans des régions pas encore explorées. Dans cette étape, pendant le déplacement du *Sink*, le *Sink* peut communiquer avec des capteurs ayant déjà communiqué avec lui, donc nous acceptons cette solution« avec critère d'aspiration».
7. Aller vers l'étape 2.
8. Arrêter le processus de la recherche du chemin lorsque le *Sink* revient à sa position initiale où le nombre d'itérations = au nombre d'itérations maximale.

L'algorithme 2 détaille la méthode *Recherche Tabou* adaptée à nos besoins:

Algorithme 2 « Recherche Tabou»

```
1: Sol=C ; /*Solution initiale : Une trajectoire initiale */
2: f* = f(Sol) ; /*Calculer le coût de la solution Sol par la fonction f (Fonction
objectif) */
3:  T=∅ ; /* Initier la liste taboue*/
4:  I = 0 ;I_max = i_max; /* i_max le nombre d'itérations maximal pour trouver la solution
*/
5:  répéter
6:     Sol* = Le meilleur voisin de la solution Sol n'est pas tabou ou tabou en
utilisant un critère d'aspiration;
7:          si f(Sol) < f(Sol*) alors
8:            Sol* = Sol ;
9:f* = f(Sol) ;
10:         fin Si
11:  Mettre à jours la liste Tabou T ;
12:  I++ ;
13:  jusqu'à (critère d'arrêt(I = I_max)) ;
14:  retourner Sol*.
```

Un algorithme de Recherche Tabou comporte trois types de mémoire :

1. ***Mémoire à court terme*** **:** il s'agit de la liste taboue traditionnelle. Le principe est d'éviter de faire des mouvements inverses pour ne pas régénérer des solutions déjà rencontrées. La taille des listes taboues est une donnée critique. C'est pour cela qu'elle peut être fixe ou variable voir même adaptative. Dans ce dernier cas, la taille évolue en fonction des solutions qu'elle contient et du voisinage utilisé.
2. ***Mémoire à moyen terme*** **:** cette mémoire traduit le principe d'intensification pour la Recherche Tabou. Le but est de permettre de tirer la recherche vers des solutions ayant de bonnes propriétés. Le problème est bien sûr de trouver ses bonnes propriétés. Ce type de mémoire est forcément dédié au problème étudié et il faut pouvoir extraire, des meilleures solutions trouvées, l'information commune qui va guider la recherche.

3. ***Mémoire à long terme*** **:** il s'agit du mécanisme de diversification. Le principe est de pouvoir diriger la recherche vers les régions non encore explorées de l'espace de recherche. Cette étape de diversification est appliquée de temps en temps pour permettre de relancer la recherche.

L'organigramme suivant (Figure 6.20), illustre les différentes instructions et étapes de l'algorithme « méthode Recherche Tabou ».

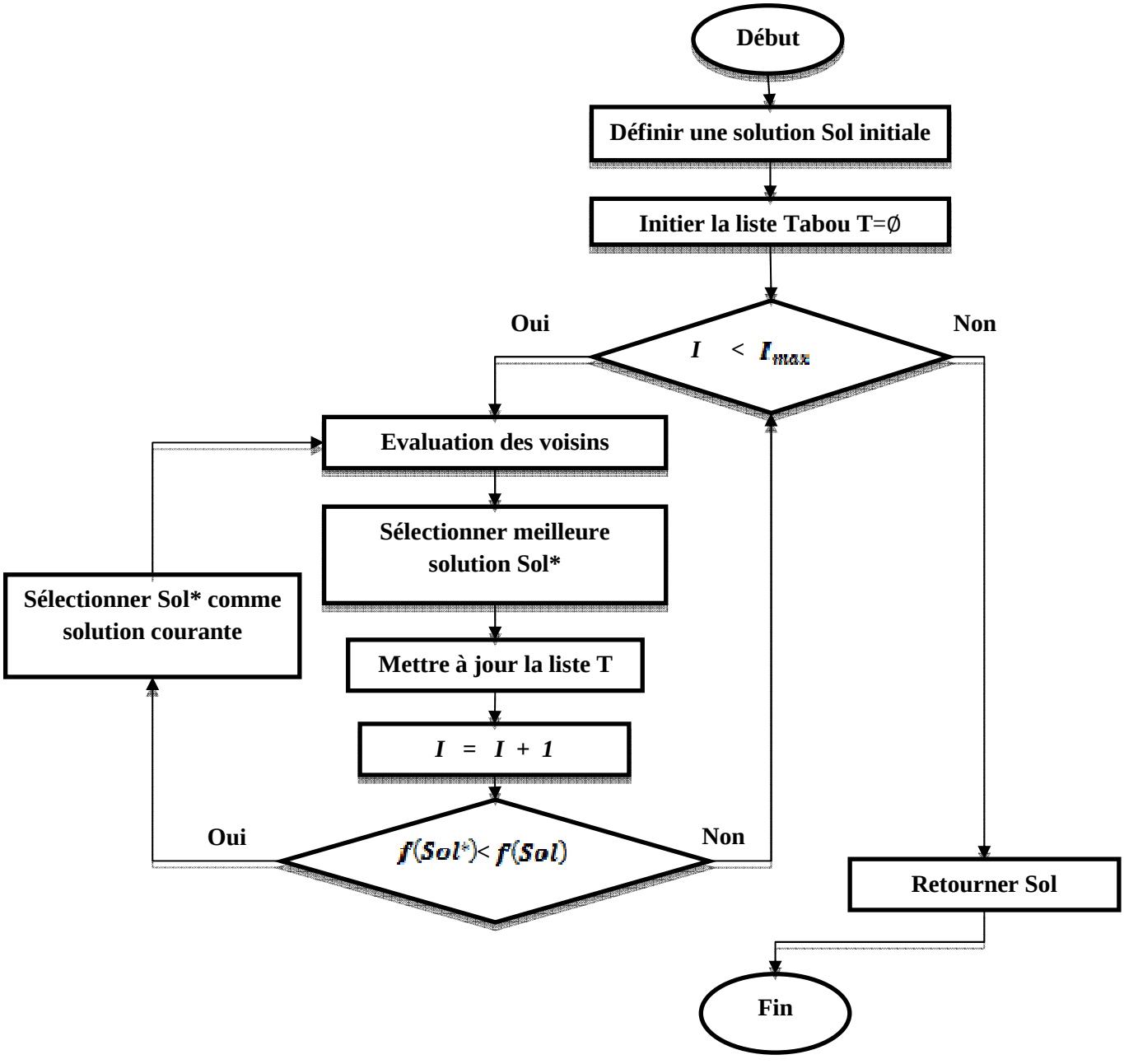


Figure 6.20 – Organigramme descriptif de la méthode « ***Recherche Tabou*** »

Dans cet organigramme, nous définissons une solution initiale et un nombre d'itérations maximal. Tant que le nombre d'itérations n'atteint pas le nombre maximal d'itérations, l'évaluation des voisins et la recherche de la meilleure solution (meilleure trajectoire) à l'aide

de la liste tabou qui a pour rôle d'éviter de traiter des solutions déjà obtenues au cours de la recherche, sont poursuivies jusqu'à ce qu'on aboutisse à une solution satisfaisante.

- ***Recuit Simulé***: l'algorithme de recuit simulé trouve ses origines dans les travaux de Kirkpatrick et Cerny [201]. Il doit son succès à son efficacité aussi bien pour des problèmes combinatoires que continus. Le recuit simulé retranscrit le processus physique permettant d'obtenir des structures cristallines. A partir d'une température élevée, un refroidissement contrôlé permet d'abaisser la température jusqu'à l'obtention d'un état d'équilibre. La température initiale et la vitesse de refroidissement sont des critères essentiels de ce processus. Dans l'algorithme du recuit simulé, la température intervient dans la probabilité d'accepter des voisins de moins bonne qualité que la solution courante. Généralement cette probabilité suit la probabilité de Boltzmann :

$$P = e^{-\frac{f(Sol') - f(Sol)}{T}} \qquad (6.1)$$

$f(x)$ est la valeur de qualité de la solution x représente *Sol* et T est la température courante. Ainsi, quand le meilleur voisin n'est pas de meilleure qualité que la solution courante, plus grande est la température, plus grande sera la probabilité de l'accepter. Par conséquent, le recuit simulé peut être vu comme un processus d'exploration qui, petit à petit, tend vers un processus d'intensification à la manière d'une recherche par descente. L'algorithme 3 donne le pseudo-code d'un recuit simulé.

Algorithme 3 « Recuit simulé »

```
1:  Sol=C ; /*Solution initiale : Une trajectoire initiale */
2:  f* = f(Sol) ; /*Calculer le coût de la solution Sol par la fonction f (Fonction
objectif) */
3 : T=Tmax ; /* Initier la Température */
4 :  répéter
5 :  Sol* = Un voisin aléatoire de la solution Sol;
6 :  δE = f(Sol*) – f(Sol) ;
7 : si ( δE ≤ 0) alors
8 :Sol = Sol* ;
9 : f* = f(Sol) ;
10 :sinon
11 :Sol = Sol* Avec la probabilité e^-δE/T ;
12 : f* = f(Sol) ;
13 :fin si
14 : T = T * α ;/*Mettre à jours la température T, avec 0 < α < 1 */
15 :jusqu'à (critère d'arrêt T > Tmin);
16 : retourner Sol*
```

L'organigramme suivant (Figure 6.21), illustre les différentes instructions et étapes de l'algorithme « méthode Recuit Simulé ».

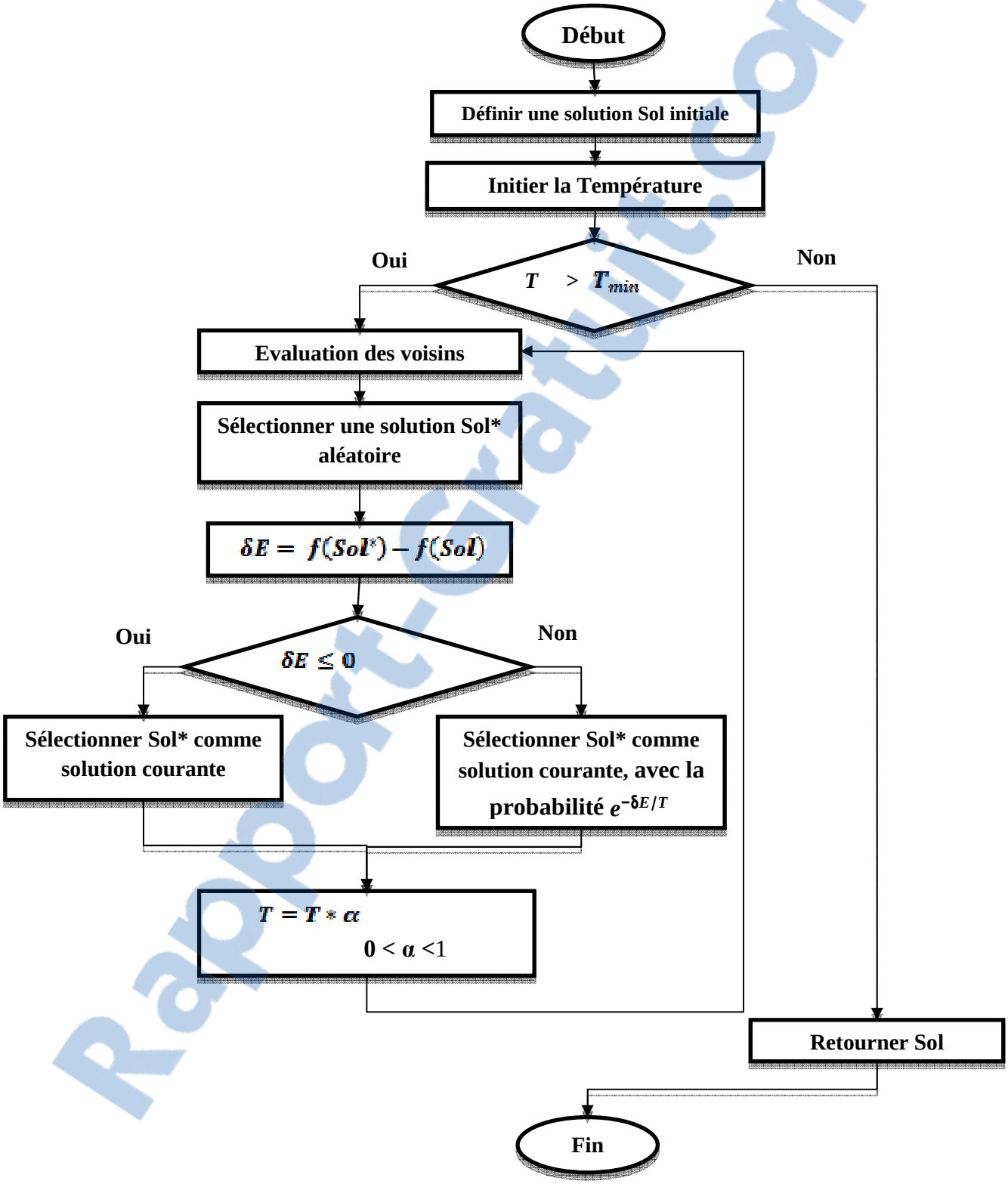


Figure 6.21– Organigramme descriptif de la méthode « ***Recuit Simulé*** »

Dans cet organigramme, à chaque lancement d'une recherche nous devons choisir initialement une solution et une température maximale. Cette température baisse au fur et à mesure que la recherche d'une meilleure solution se poursuive. Pendant le temps où la température est toujours supérieure à une température minimale, une génération de voisins est évaluée aléatoirement et la solution acceptée dépend de sa qualité.

6.3.2 Description du réseau

Notre réseau est constitué de 25 nœuds capteurs et d'un Sink mobile comme nous pouvons le voir sur la figure 6.22. Notre but est d'optimiser les déplacements du Sink en trouvant la meilleure trajectoire dans la zone de déploiement des capteurs où chaque capteur est caractérisé par un identifiant (ID), des coordonnées (X et Y) ainsi qu'un taux d'erreur (TE). Nous allons utiliser les deux méthodes « Recherche Tabou et Recuit Simulé » pour résoudre notre problématique.

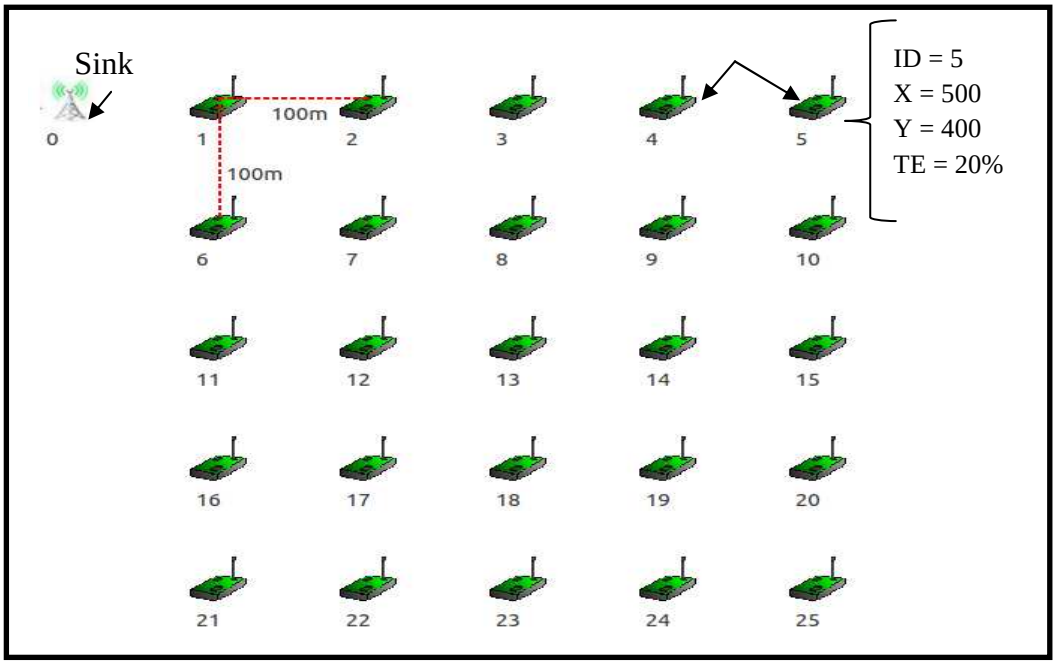


Figure 6.22 – Description de notre réseau

6.3.2.1 Modèle mathématique proposé pour la mobilité du Sink

Dans cette section, nous présentons notre modèle de mobilité pour les déplacements du Sink pendant le processus de collecte.

Il s'agit de trouver la plus courte trajectoire en parcourant tous les capteurs se trouvant dans le l'espace de déploiement, de minimiser le déplacement vers les capteurs qui ont le taux d'erreur le plus faible tout en maximisant la qualité du signal.

$$\min \quad \alpha[Dist(S,C_1) + Dist(S,C_m) + \sum_{i=1}^{m-1} Dist(C_i,C_{i+1})] + \beta[\sum_{i=1}^{n} Dist(S,C_i) \times TE_i] + \gamma[\sum_{i=1}^{n} \frac{1}{QS_i}] \quad (6.2)$$

$Dist_{ij}$: La distance entre un capteur i et un capteur $j$est donnée par la formule suivante :

$$Dist_{ij} = \sqrt{(w-u)^2 + (z-v)^2} \forall i, j \in S, i \neq j. \forall (w,z), (u,v) \in \mathbb{R}^2 \quad (6.3)$$

Les coordonnées des capteurs i et j.

C_k : Un capteur parmi l'ensemble des capteurs et k indique son ordre d'apparition dans la tournée.

TE_i : Taux d'erreur d'un capteur i en %; S : Sink ;

QS_i : Qualité du signal d'un capteur i ; α, β et γ : Trois constantes avec $\alpha + \beta + \gamma = 1$.

Pour favoriser :

- La trajectoire, nous choisissons un α proche de 1.
- Le taux d'erreur, nous choisissons un β proche de 1.
- La qualité du signal, nous choisissons un γ proche de 1.

- **Les variables :**

d_{ij} : déplacement d'un capteur i vers un capteur j, $d_{ij} = 1$ si un déplacement est effectué et $d_{ij} = 0$ sinon.

- **Les constantes :**

$Dist_{ij}$, TE_i

R_i : Rayon de la portée radio d'un capteur i.

e_{min} : L'énergie minimale. Si l'énergie d'un capteur atteint cette limite, ce dernier ne pourra pas envoyer de données.

$d_{max} = n + 1$: le nombre maximal de déplacements d_{ij}entres les capteurs.

QS_{min} : la qualité du signal nécessaire pour qu'un capteur puisse envoyer ses données.

M_{max} : la taille maximale de la mémoire d'un capteur.

6.3.2.2 Formalisation du problème

1. *La fonction objectif*

$$\min \ \alpha \sum_{i,j \in S, i \neq j}^{n} c_{ij} d_{ij} + \beta \sum_{i,j \in S, i \neq j}^{n} c_{ij} d_{ij} TE_j + \gamma \sum_{i=1}^{n} \frac{1}{QS_i} \quad (6.4)$$

Où c_{ij} coût entre un capteur i et un capteur j.

$$d_{ij} = \begin{cases} 1, & S'il\ existe\ un\ déplacement\ de\ i\ vers\ j\ dans\ la\ trajectoire\ optimale. \\ 0, & Sinon. \end{cases}$$

2. Les contraintes

$Dist_{ij} \leq R_i \forall i,j \in S, i \neq j.$ (6.5)

$e_{Ri} \geq e_{min} \forall i \in S$. (6.6)

$d_{ij} = 1$ Si$[(Dist_{ij} \geq R_i) \wedge (QS_j \geq QS_{min}) \wedge (e_{Rj} \geq e_{min})] \vee$ (6.7)

$[(e_{Rj} < e_{min}) \wedge (M_j = M_{max})], \forall i,j \in S, i \neq j.$

$P_i \neq P_j$ Si $d_{ij} = 1, \forall i,j \in S, i \neq j.$ (6.8)

$\sum d_{ij} \leq d_{max} \forall i,j \in S, i \neq j.$ (6.9)

La fonction objectif (6.4) représente le coût de la trajectoire optimale pour le déplacement du Sink avec une faible consommation d'énergie et une bonne qualité de signal.

- La contrainte (6.5) permet de vérifier qu'un capteur se trouve dans la portée radio du Sink pour que la communication puisse se faire.
- La contrainte (6.6) concerne l'énergie. En effet, un capteur doit avoir une énergie suffisante pour pouvoir communiquer avec le Sink.
- La contrainte (6.7) permet le déplacement du Sink vers un capteur.

Un déplacement s'effectue :

Si la distance entre le Sink et un capteur, qui a une bonne qualité du signal et une énergie suffisante, est supérieure au rayon de la portée radio du Sink.
Si un capteur possède une énergie inférieure au seuil minimal et que sa mémoire est pleine.

- La contrainte (6.8) s'assure que le Sink ne chevauche pas avec un capteur lors de son déplacement.
- La contrainte (6.9) s'assure que le nombre optimal de déplacement du Sink ne dépasse pas le nombre maximal d_{max} tel que défini plus haut.

6.3.3 Etapes de l'approche proposée

Notre approche est représentée par des organigrammes et des diagrammes dans les figures suivantes :

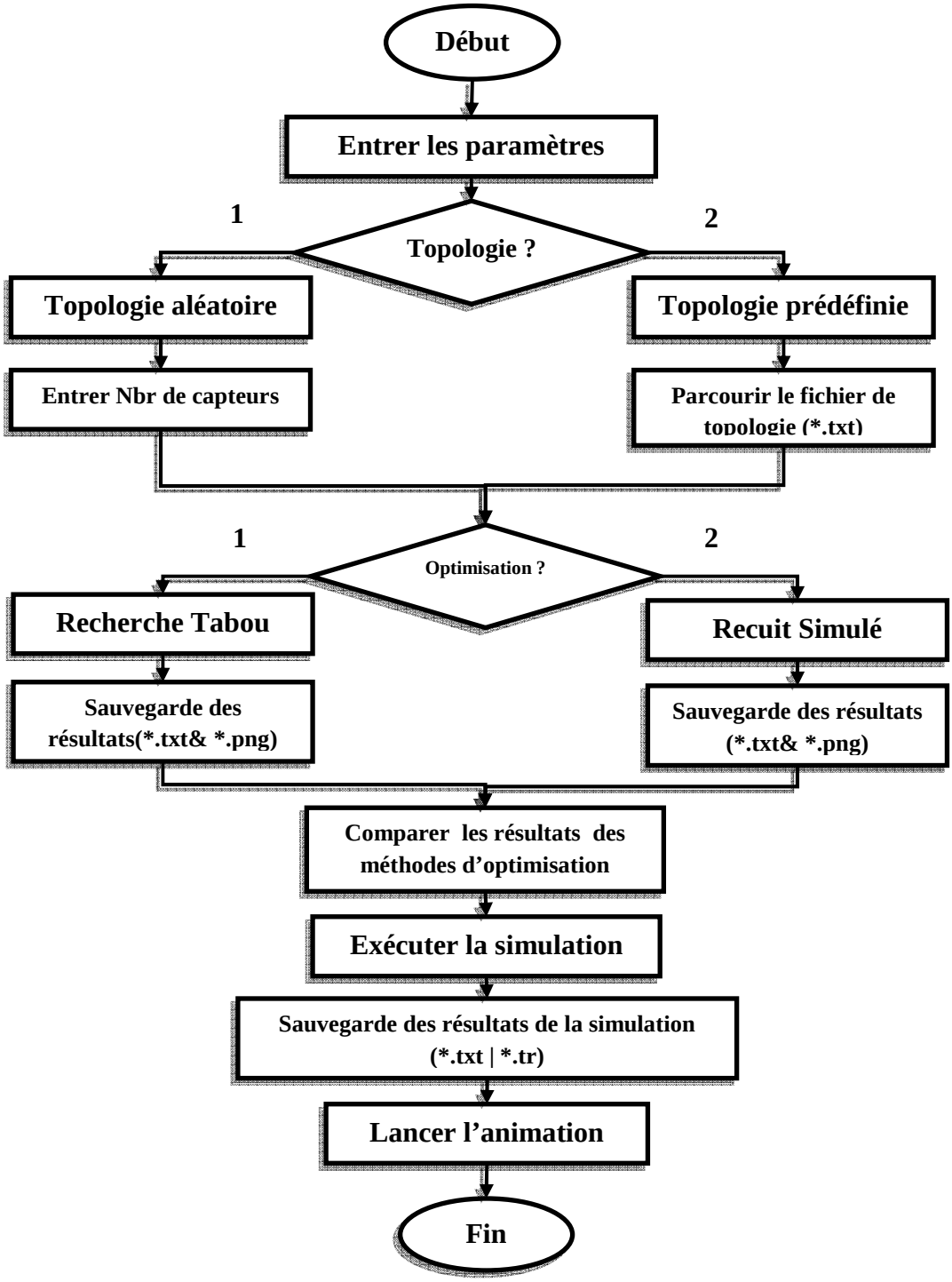


Figure 6.23 – Etapes de l'approche proposée

- **Entrer les paramètres :** Nous introduisons les paramètres nécessaires à l'optimisation. Ces paramètres sont : le nombre d'itérations dans la recherche tabou, la température dans le recuit simulé, les facteurs de la fonction objectif (distance et taux d'erreur) et la vitesse de déplacement du Sink.
- **Topologie :** Cette topologie est soit aléatoire et nécessite l'introduction d'un nombre de capteurs, ou bien prédéfinie, et nécessite le parcours d'un fichier «.txt » contenant le nombre de capteurs déployé dans le réseau.

- **Optimisation :** Nous choisissons la méthode d'optimisation (Recherche Tabou ou Recuit Simulé).
- **Sauvegarde des résultats :** Affiche et enregistre les résultats d'optimisation (le coût de la trajectoire, la latence…). La trajectoire ainsi calculée et enregistrée dans un fichier « .txt », pour être utilisée par la suite dans la simulation. Nous pouvons enregistrer également le chemin parcouru sous un fichier image « .png ».
- **Comparer les résultats d'optimisation :** Nous pouvons tracer les graphes et les histogrammes des résultats des méthodes d'optimisation (Temps de Calcul, coût des résultats obtenus et évolution de la fonction objectif pour chaque méthode).
- **Exécuter la simulation :** Nous pouvons lancer la simulation des résultats obtenus par les méthodes d'optimisation. La configuration de la simulation se fait par la lecture des paramètres entrés initialement.
- **Sauvegarde des résultats de simulation :** concerne la génération des fichiers traces de la simulation.
- **Lancer l'animation :** en lançant l'animation, nous pouvons visualiser la trajectoire obtenue par les méthodes d'optimisation utilisées dans notre application.

Diagramme de sequence

Le diagramme de séquence décrit le comportement de l'utilisateur vis-à-vis de notre outil d'optimisation que nous avons implémenté. Nous avons intégré tous les modules dans l'application pour permettre à l'utilisateur de traiter toutes les étapes inhérentes à l'optimisation, à la comparaison et à la simulation sans être amené à changer cet outil.

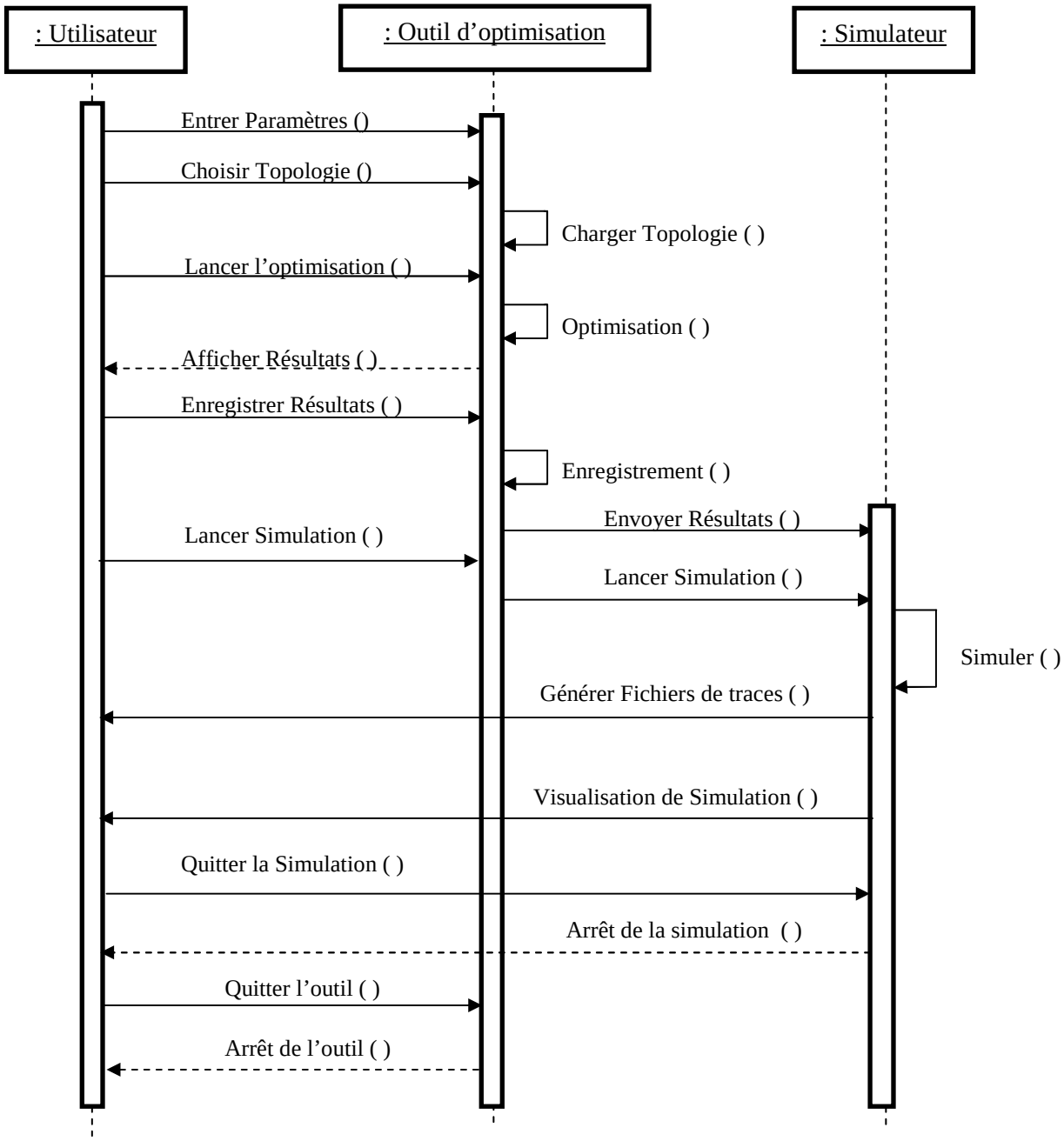


Figure 6.24 – Diagramme de séquence modélisant l'outil d'optimisation et de simulation

6.3.4 Architecture de l'outil d'optimisation « EIOSM » (Environnement Intégré d'Optimisation et de Simulation basé sur les Méta-heuristiques)

EIOSM, est l'outil d'implémentation que nous avons développé pour optimiser les déplacements du Sink mobile dans un RCSF par les méta-heuristiques, ensuite nous avons simulé ces résultats obtenus par optimisation.

Dans la figure suivante, nous présentons tous les modules implémentés et utilisés pour la réalisation de notre outil d'optimisation et de simulation EIOSM.

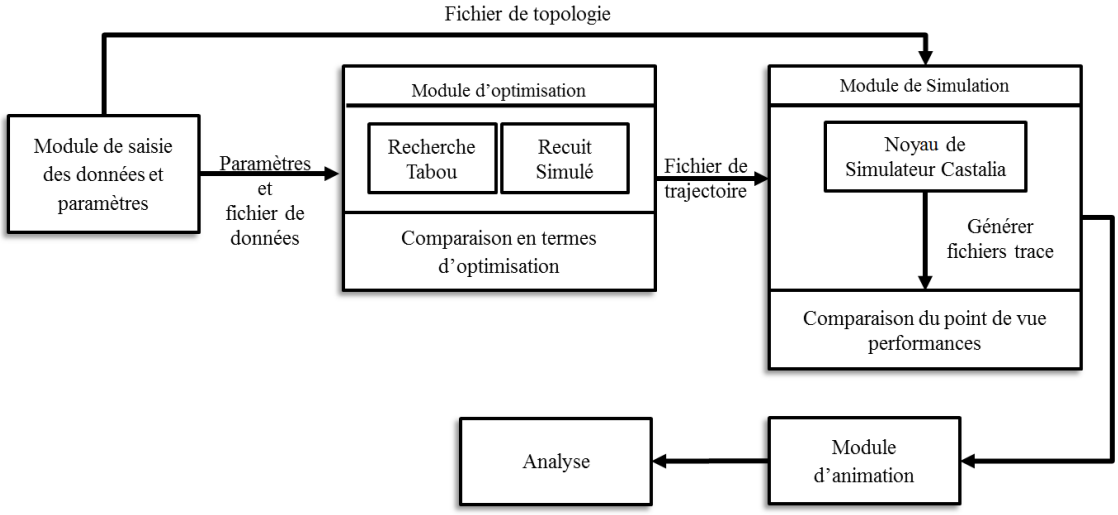


Figure 6.25 – Architecture d'EIOSM

La figure ci-dessous donne un aperçu de notre outil EIOSM.

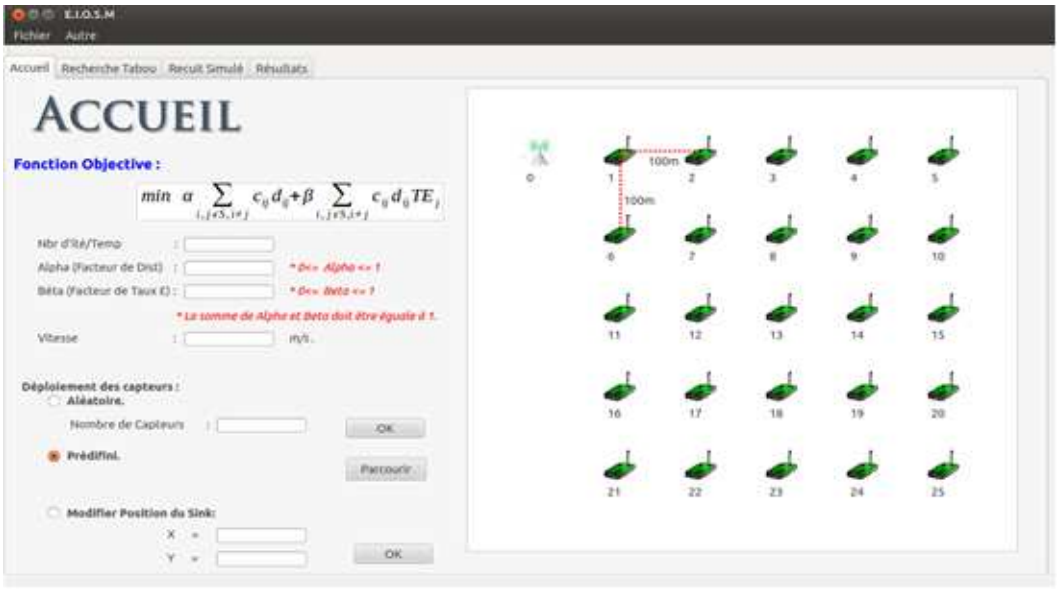


Figure 6.26 – L'outil EIOSM

6.3.4.1 Module de saisie des données et des paramètres

Dans ce module, nous introduisons les différents paramètres nécessaires pour lancer l'optimisation et charger la topologie du réseau dont nous voulons optimiser la trajectoire du Sink mobile. Les paramètres sont déjà cités plus haut.

6.3.4.2 Module d'optimisation

Ce module est responsable du calcul de la trajectoire du Sink mobile à partir de la fonction objectif. Nous avons défini la fonction objectif dans notre outil comme suit :

Algorithme 4 : Implémentation de la fonction objectif dans EIOSM

```
float FO(int *Sol, int N, float Alpha, float Beta) {
        float cost = 0;
        for (int i = 0; i < N; i++) {
            cost = cost+Alpha*dist[Sol[i]][Sol[i + 1]]
                +Beta*TE[Sol[i+1]]*dist[Sol[i]][Sol[i + 1]];
            }
        return cost;
}
```

La solution est exprimée en une structure de vecteur où les éléments de ce vecteur sont les identifiants des capteurs.

Méthode « Recherche Tabou »: la figure suivante présente l'interface d'optimisation avec la méthode « Recherche Tabou ».

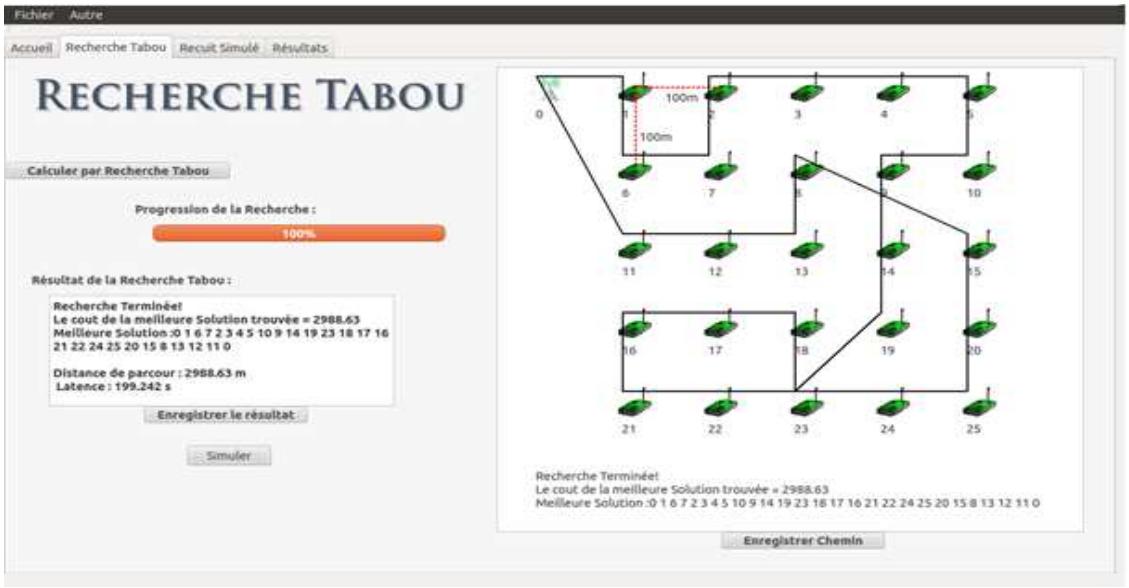


Figure 6.27 – Interface d'EIOSM avec la méthode RT

Algorithme 5 : Implémentation de la RT dans EIOSM

```
////////Générer la solution initiale
int *currSol;
currSol = new int[N + 1];
for (int i = 0; i < N + 1; i++) {
          currSol[i] = i;
    }
currSol[N] = 0;
////////Copier la solution initiale
int *bestSol = new int[N + 1];
for (int i = 0; i < N + 1; i++) {
          bestSol[i] = currSol[i];
    }
////////Calculer le cout de la solution
float bestCost = env.FO(bestSol, N, Alpha, Beta);
////////Lancement de la RT
for (int ite = 0; ite < I; ite++) {
            currSol = mVoisin(tabouList, env, currSol, N, Alpha, Beta);
            float currCost = env.FO(currSol, N, Alpha, Beta);
            if (currCost < bestCost) {
                for (int i = 0; i < N + 1; i++) {
                    bestSol[i] = currSol[i];
                };
                bestCost = currCost;
            }
        }
```

Méthode « Recuit Simulé »: la figure suivante présente l'interface d'optimisation avec la méthode « Recuit Simulé ».

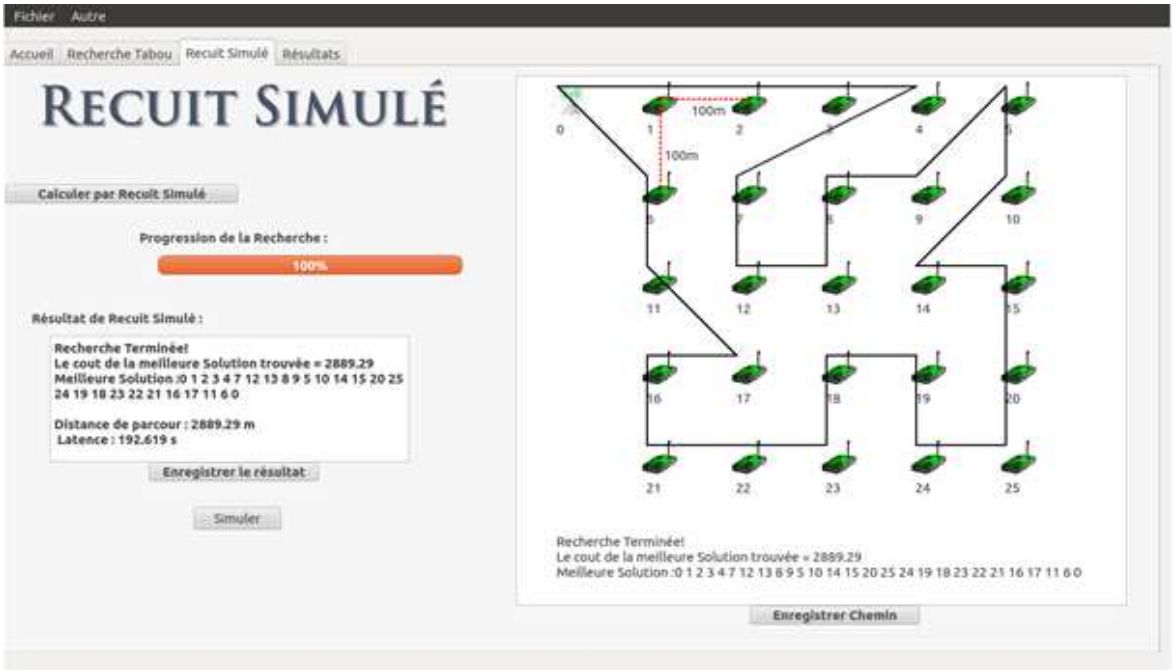


Figure 6.28 – Interface d'EIOSM de la méthode RS

La figure ci-dessous présente le code implémenté de la méthode Recuit Simulé dans notre outil d'optimisation EIOSM.

Algorithme 6 : Implémentation du RS dans EIOSM

```
1 //////////Générer la solution initiale
2 int *currSol;
3 currSol = new int[N+1];
4 for (int i = 0; i < N+1 ; i++) {
5     currSol[i] = i;
6 }
7 currSol[N]=0;
8 /////////Calculer le cout de la solution
9 float costd=env.FO(currSol,N, Alpha, Beta);
10 int *newbestSol;
11 newbestSol = new int[N+1];
12 /////////Lancement de la RS
13 while (temperature>temperatureAbsolute){
14     newbestSol=autreSol(currSol,N);
15     float newcostd=env.FO(newbestSol,N, Alpha, Beta);
16     delta= newcostd-costd;
17     double T = static_cast <double> (rand()) / static_cast <double> (RAND_MAX);
18     if ((delta < 0) || (costd > 0 && exp(-delta / temperature) > T))
19       {
20          for (int i = 0; i < N+1 ; i++) {
21                currSol[i] = newbestSol[i];
22          };
23    costd = delta + costd;
24    }
25     temperature *= alpha;
26 }
```

- **Comparaison en termes d'optimisation**

Notre outil permet de tracer le graphe de l'évolution de la fonction objectif pour chaque méthode d'optimisation et affiche les différents histogrammes nécessaires pour comparer les méthodes implémentées. Le temps d'exécution est calculé à partir du moment où la méthode d'optimisation est lancée jusqu'à sa fin.

6.3.4.3 Module de simulation

A partir des données et des paramètres introduits dans le module de saisie, le module d'optimisation procède au calcul de la solution selon les méthodes du RS ou de la RT. Le simulateur lit les données notamment celles de la topologie du réseau (module de saisie) et les résultats obtenus par le module d'optimisation. Enfin, ce module procède à la configuration de la simulation des résultats obtenus.

- **Comparaison en termes de performances**

A partir des fichiers trace générés après le lancement de la simulation des résultats obtenus par les méthodes d'optimisation que nous avons implémentés (Recherche Tabou et Recuit Simulé), nous pouvons extraire les performances qui nous intéressent dans la simulation. Nous pouvons aussi, à l'aide de ces fichiers résultants de la simulation, tracer les graphes et les courbes de performances en utilisant le Gnuplot où même l'EXCEL. Le but recherché, par l'extraction des fichiers trace obtenus par la simulation et la présentation des graphes, est de comparer les performances du réseau lors de l'utilisation des méta-heuristiques implémentés dans notre outil EIOSM.

6.3.4.4 Module d'animation

Ce module prend les résultats du module de simulation (les fichiers trace) pour visualiser le comportement du Sink mobile dans le réseau.

Donc ce module est un outil permettant la visualisation des évènements qui se sont déroulés lors de la simulation. Il peut être une réelle aide pendant le débogage des scénarios de simulation complexes, et il est beaucoup plus facile de passer par le texte de Castalia-trace.txt [185].

6.3.5 Hypothèses et scénarios

6.3.5.1 Hypothèses

Dans notre travail, nous avons considéré un RCSF avec les caractéristiques suivantes :

- Le Sink est le seul nœud mobile et est doté d'un système supplémentaire d'optimisation et de mobilité.
- Le module de mobilité du Sink lui permet de se déplacer sans gêne dans la zone de déploiement (déplacement aérien ou sur une surface sans obstacle), c'est pourquoi nous avons éliminé la partie de la qualité du signal dans la fonction objectif en fixant sa valeur à 0).

- Tous les capteurs ont la même quantité d'énergie. Au départ la quantité d'énergie du nœud est proposée par Castalia, sa valeur par défaut est 18720 Joules qui correspondent à deux piles AA.

- La zone de déploiement est représentée sous forme d'un espace à deux dimensions (600m x 600m).

6.3.5.2 Scénarios

Nous pouvons travailler sur une topologie aléatoire (Figure 6.29) ou bien sur une topologie prédéfinie (à partir du fichier de topologie comme nous pouvons le voir dans la figure 6.30).

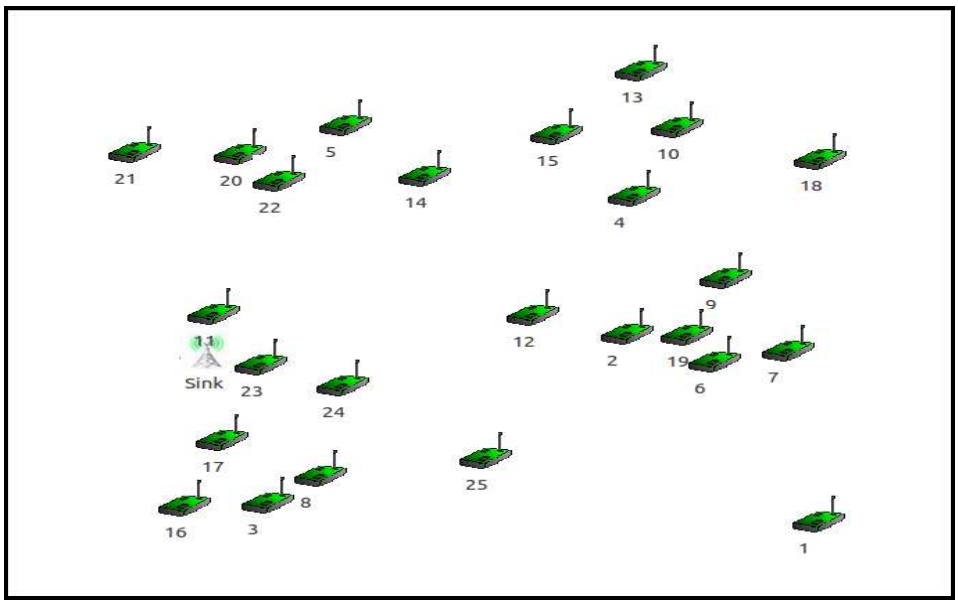


Figure 6.29 – Topologie aléatoire

Dans tous les scénarios, nous travaillons avec une topologie en grille et nous considérons que la vitesse de déplacement du Sink mobile est égale à 15m/s.

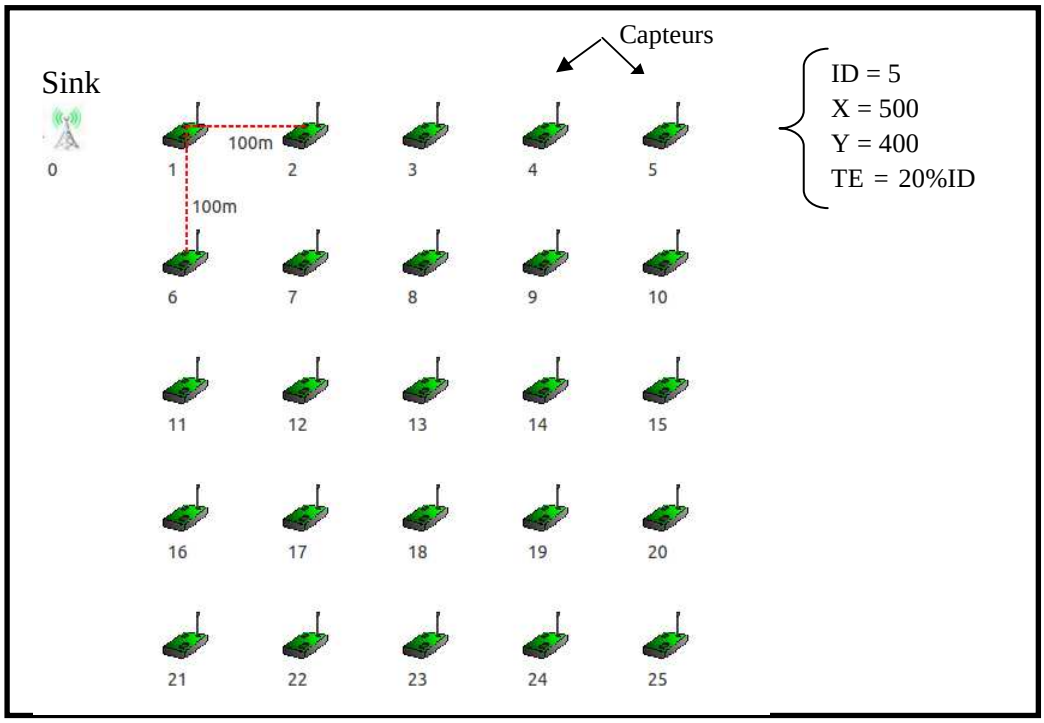


Figure 6.30 – Topologie en grille

La table 6.9 présente les différents scénarios utilisés dans notre travail.

	Caractéristiques	***α(Facteur de distance)***	***β (Facteur de taux d'erreur)***
S1	Minimiser la trajectoire.	$\alpha \approx 1$	$\beta \approx 0$
S2	Minimiser le taux d'erreur.	$\alpha \approx 0$	$\beta \approx 1$
S3	Equilibrer entre trajectoire et taux d'erreur.	$\alpha = 0.5$	$\beta = 0.5$

Table 6.9 – Scénarios S1,S2 et S3

6.3.6 Evaluation des résultats

Dans cette section, nous présentons les résultats obtenus par optimisation et simulation en termes : d'optimisation et de performances du réseau.

6.3.6.1 En terme d'optimisation

Nous évaluons les résultats obtenus selon trois métriques : Stabilité, Temps d'exécution et Meilleure solution obtenue (distance parcourue et latence).

- **Stabilité**

La figure 6.31 montre l'évolution de la fonction objectif dans l'espace des solutions, où nous observons que la courbe représentant la méthode « Recherche Tabou » est plus stable que celle du Recuit Simulé. Ceci s'explique par le fait que dans la Recherche Tabou, nous avons la mémoire Tabou qui contrôle la génération des solutions dans le voisinage, par contre dans

le Recuit Simulé, la génération des solutions se fait d'une manière aléatoire dans l'espace de recherche.

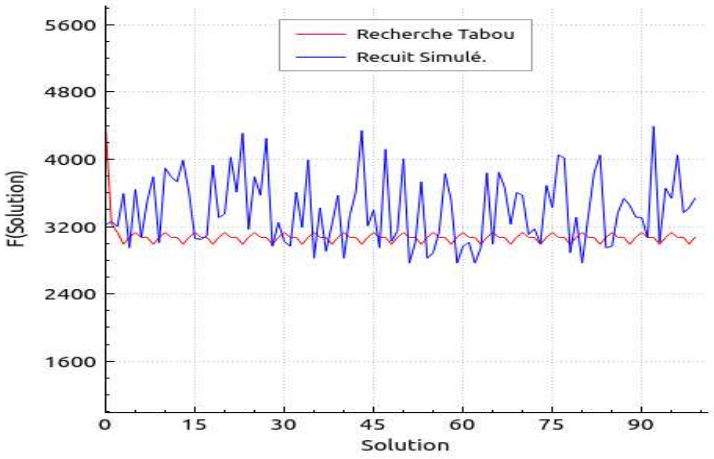


Figure 6.31 – Evolution de la fonction objectif dans l'espace de recherche en appliquant la RT et le RS

- **Temps d'exécution**

La table 6.10 et la figure 6.32 montrent le temps d'exécution de chaque méthode d'optimisation « Recherche Tabou et Recuit Simulé » utilisée dans tous les scénarios.

Méthodes	***Temps d'exécution (Sec)***
Recherche Tabou	0.17 Sec
Recuit Simulé	0.31 Sec

Table 6.10 – Temps d'exécution de la RT et le RS dans tous les scénarios

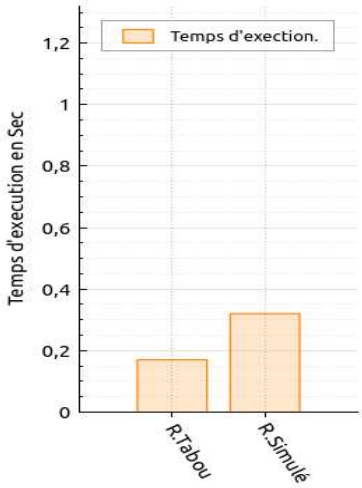


Figure 6.32 – Temps d'exécution de la RT et le RS dans tous les scénarios

Nous remarquons sur la table 6.10 et la figure 6.32 que la Recuit Simulé prend plus de temps de calcul que la Recherche Tabou parce qu'il s'agit d'un processus qui se révèle long et coûteux notamment à cause de la génération des nombres aléatoires et l'utilisation d'exponentiels.

- **Meilleure Solution**

La table 6.11 présente la distance parcourue ainsi que la latence (Temps de la tournée) selon les deux méthodes. La latence est calculée comme suit: ***Latence = Distance Parcourue / Vitesse***

Méthodes	***Distance Parcourue***	***Latence***
Recherche Tabou	S1 : 2988.63 m S2 : 3763.17 m S3 :3465.03 m	S1 : 199.24 Sec S2 : 250.88 Sec S3 : 231.00 Sec
Recuit Simulé	S1 : 2765.69 m S2 : 3223.16 m S3 : 2984.16 m	S1 : 184.38 Sec S2 : 214.88 Sec S3 : 198.94 Sec

Table 6.11 – Résultats de simulation obtenus dans tous les scénarios

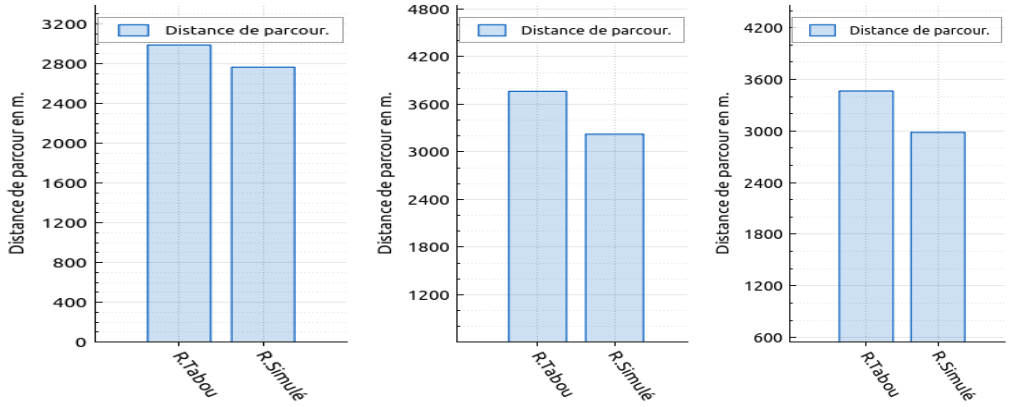


Figure 6.33 – Distance parcourue dans chaque scénario

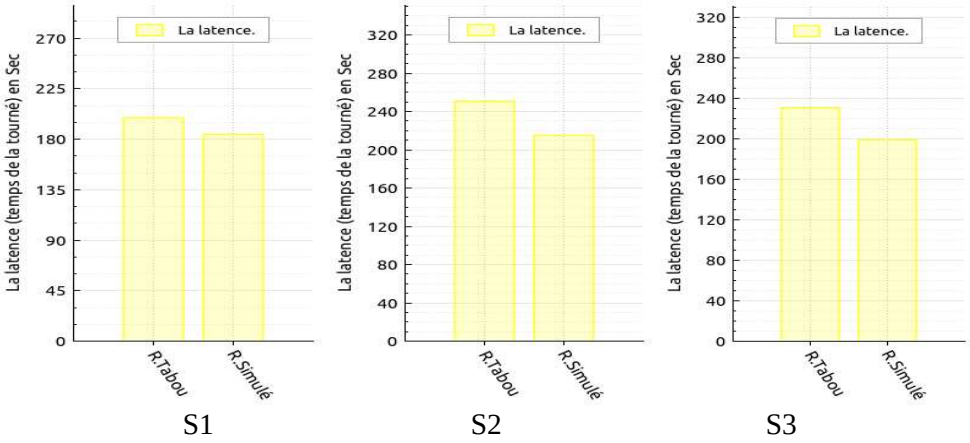


Figure 6.34 – La latence dans chaque scénario

Nous constatons dans tous les scénarios que le Recuit Simulé donne de meilleurs résultats en termes de distance parcourue et de latence, ceci est dû au fait qu'il trouve toujours les bonnes solutions par rapport à la Recherche Tabou et ceci grâce à la grande exploration de l'espace des solutions.

- **Impact de la densité**

La table suivante montre l'impact de la densité des nœuds avec les deux méthodes « Recherche Tabou et Recuit Simulé » dans un scénario où nous minimisons la distance.

	25 nœuds	50 nœuds	75 nœuds	100 nœuds
Temps d'exécution	RT : 0.17 Sec RS : 0.31 Sec	RT : 0.3 Sec RS : 0.96 Sec	RT : 0.64 Sec RS : 1.32 Sec	RT : 1.36 Sec RS : 2.44 Sec
Distance parcourue	RT :2988.63 m RS : 2765.69 m	RT : 5131.03 m RS : 4882.21 m	RT : 9005.7 m RS : 8721.48 m	RT : 13657 m RS : 12289.6 m
Latence	RT : 199.24 Sec RS : 184.38 Sec	RT : 342.06 Sec RS : 325.48 Sec	RT : 600.38 Sec RS : 581.43 Sec	RT : 910.46 Sec RS : 819.307 Sec

Table 6.12 – Impact de la densité des nœuds avec la RT et le RS

6.3.6.2 En termes de performance du réseau

Nous évaluons les résultats obtenus selon trois métriques : L'énergie consommée, La perte de paquets et la latence.

Les tables suivantes (6.13, 6.14 et 6.15) représentent les résultats de simulation de chaque scénario. Le temps de simulation dans tous les scénarios est de 850 secondes.

	Recuit Simulé	***Recherche Tabou***
Paquets générés par la couche application	5000	5000
Paquets envoyés	5000	5000
Paquets reçus par le Sink en %	4999 (99,98%)	4891 (97,82%)
Paquets perdus	1	109
Energie consommée	46,80 J	46,91 J
Latence	86,42 Sec	89,23 Sec
Distance parcourue	12714,1m	12716,5m

Table 6.13 – Résultats de simulation de S1

	Recuit Simulé	***Recherche Tabou***
Paquets générés à la couche application	5000	5000
Paquets envoyés	5000	5000
Paquets reçus par le Sink en %	5000 (100%)	4981 (99.62%)
Paquets perdus	0	19

Energie consommée	46,82 J	47,53 J
Latence	91,29 Sec	103,24 Sec
Distance parcourue	12718,9m	12723,5m

Table 6.14 – Résultats de simulation de S2

	Recuit Simulé	***Recherche Tabou***
Paquets générés à la couche application	5000	5000
Paquets envoyés	5000	5000
Paquets reçus par le Sink en %	4847 (96,94%)	4650 (93%)
Paquets perdus	153	350
Energie consommée	46,82 J	47,27 J
Latence	89,33 Sec	92,81 Sec
Distance parcourue	12716,5m	12721m

Table 6.15 – Résultats de simulation de S3

- **L'energie consommée**

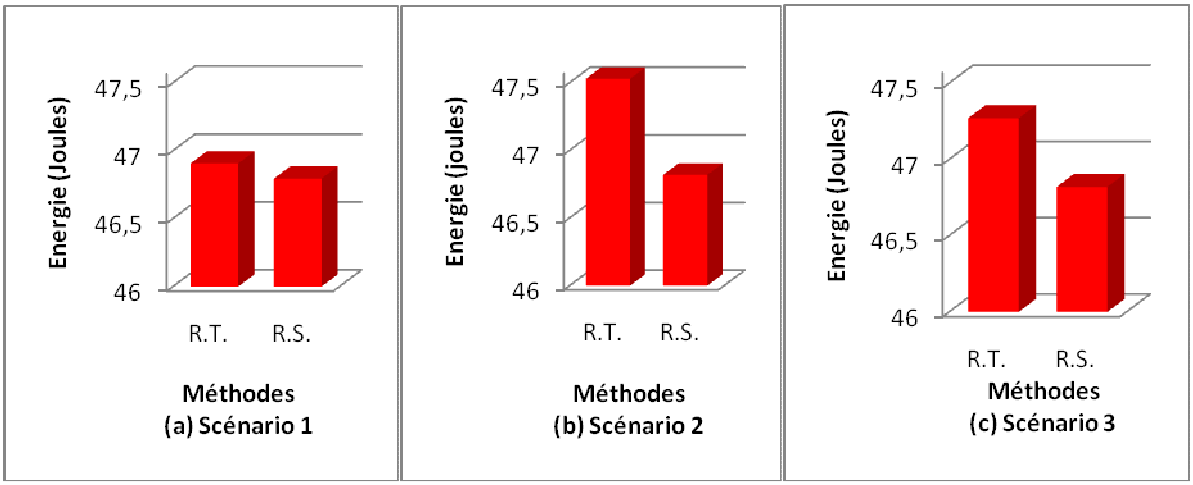


Figure 6.35 – Energie consommée dans chaque scénario

Nous remarquons sur la figure 6.35 que l'énergie consommée dans les deux méthodes est quasiment la même, mais l'énergie consommée dans la Recherche Tabou est légèrement supérieure à celle consommée dans le Recuit Simulé. Cette différence est due à une communication supplémentaire entre le Sink et les capteurs. Alors que dans le Recuit Simulé, pas plus d'une communication est établie entre un capteur et le Sink lors de son déplacement.

- **La perte de paquets**

La perte de paquets est calculée comme suit : ***Nombre de paquets perdus = Le nombre de paquets envoyés – Le nombre de paquets reçus***

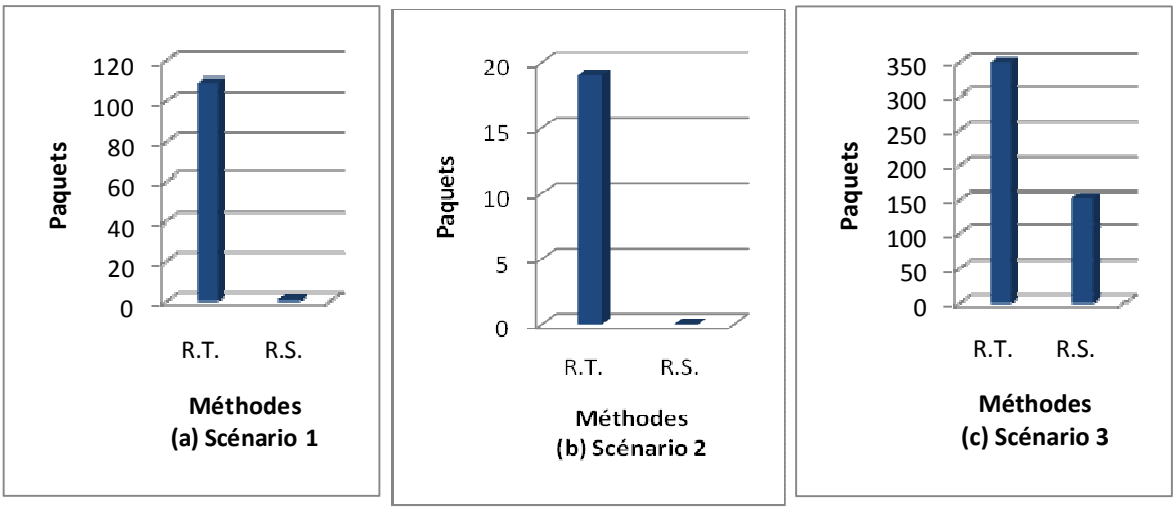


Figure 6.36 – Nombre de paquets perdus dans chaque scénario

La perte de paquets est due à la mobilité du Sink, ceci s'explique par le fait que ce dernier reçoit les données des capteurs alors qu'il est en déplacement ou lorsqu'il s'apprête à sortir de la portée radio de l'émetteur. Pour le taux de réception dans le S1, les deux méthodes donnent un taux appréciable qui est de 99.98% et 97.82% respectivement pour le RS et la RT.

Dans le S2, lorsque nous favorisons le taux d'erreur dans la fonction objectif, nous nous intéressons à la qualité de réception des données, c'est pour cette raison que les résultats sont obtenus avec un taux de réception de 100% pour le Recuit Simulé et de 99,62% pour la Recherche Tabou.

Le taux de réception diminue dans le S3 (96,94% pour le Recuit Simulé et 93% pour la Recherche Tabou) à cause des déplacements du Sink vers les nœuds destinataires lorsqu'il entre dans la portée radio des nœuds hors sa destination. Donc la communication entre le Sink et ces capteurs n'est pas établit complètement. La figure 6.40 décrit la trajectoire parcourue par le Sink et montre que ce dernier passe par des capteurs non concernés lors de son déplacement vers les nœuds destinataires.

- **Latence**

La latence dans les RCSFs est le temps nécessaire pour envoyer un paquet de données depuis un capteur (source) vers le Sink (destination). C'est-à-dire, de l'instant où le paquet est généré par la couche application jusqu'au moment où ce paquet est réceptionné par le Sink et est calculé comme suit :

Latence = Le temps de réception du paquet – Le temps de génération du paquet

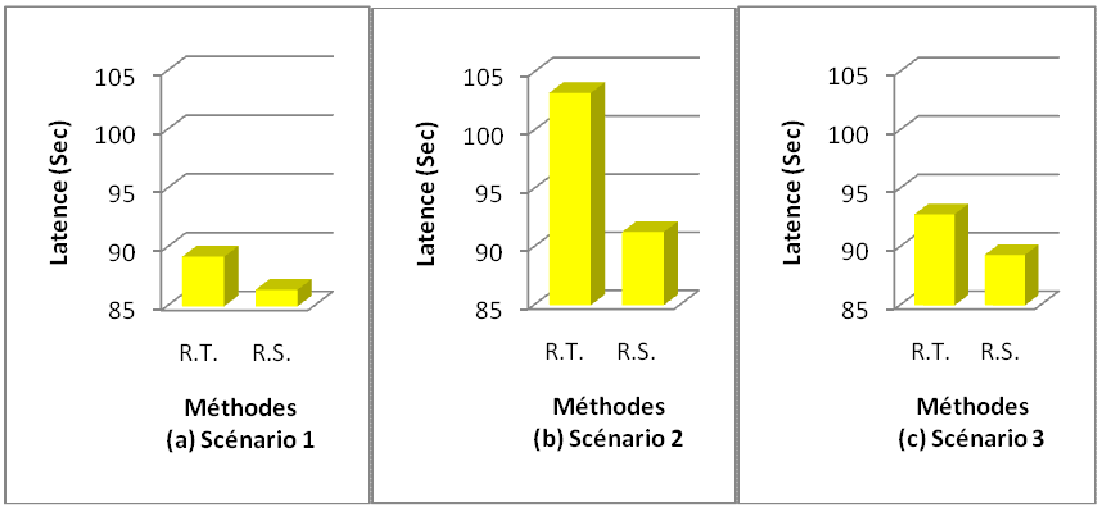
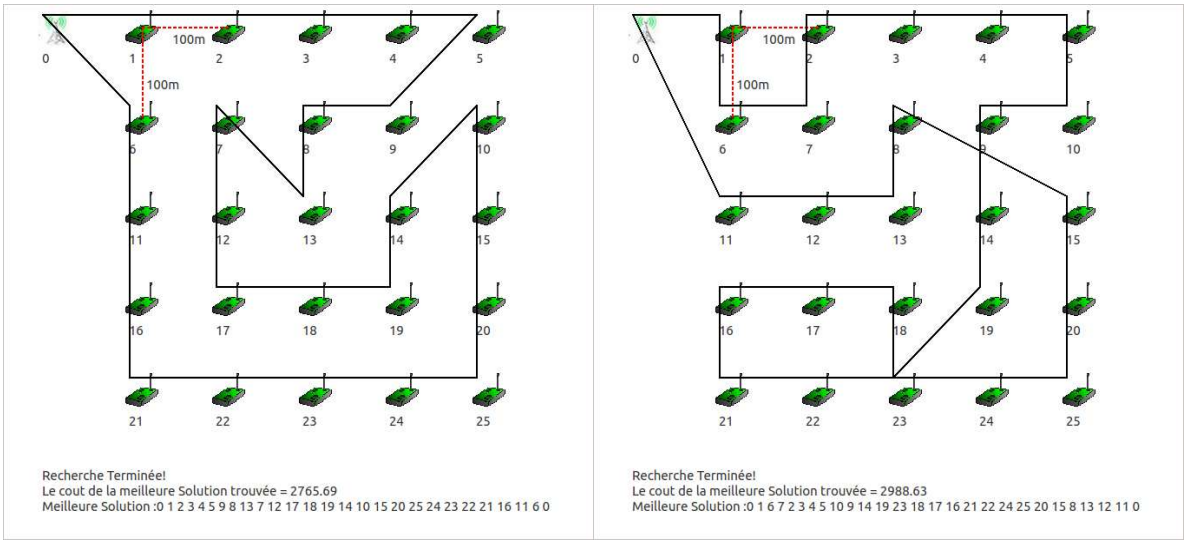


Figure 6.37 – Latence calculée dans les trois scénarios

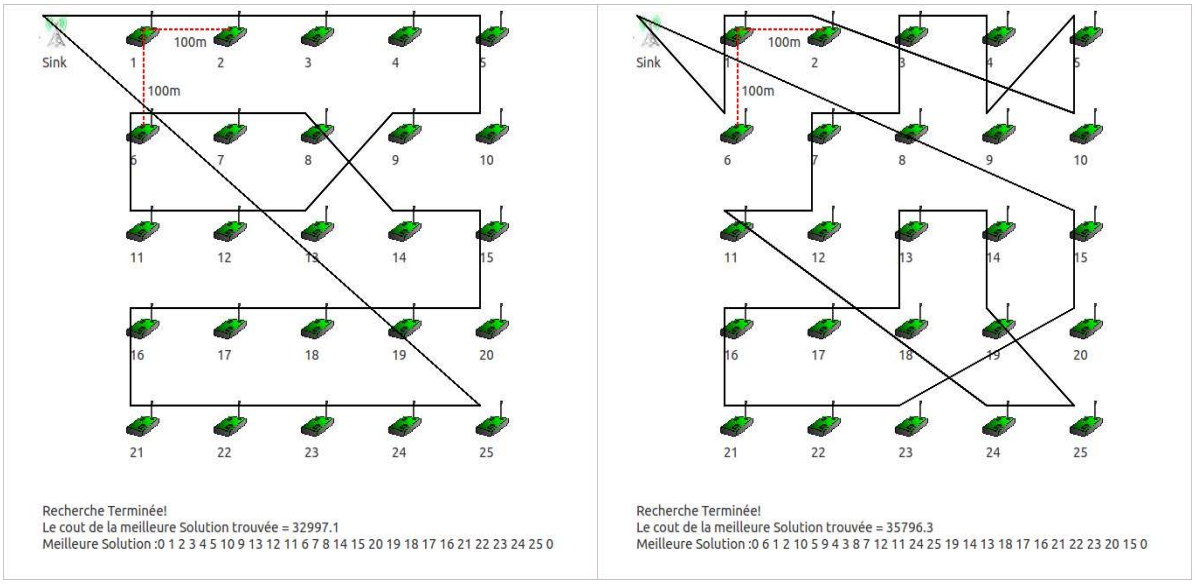
Le capteur génère ses paquets au niveau de la couche application et attend que le Sink entre dans sa portée radio pour lui envoyer les paquets de données. Donc la latence, dans notre cas, dépend du déplacement du Sink vers les nœuds capteurs. La Recherche Tabou donne une latence plus élevée que le Recuit Simulé, cette différence est due toujours au déplacement du Sink dans le réseau où le Sink se déplace d'une manière plus optimale avec la méthode du Recuit Simulé qu'avec celle de la Recherche Tabou.

Nous remarquons dans tous les scénarios que le Recuit Simulé donne de meilleures performances par rapport à la Recherche Tabou qui donne des résultats satisfaisants mais pas de bonne qualité que ceux obtenus par le Recuit Simulé, que ça soit pour la collecte de données, l'énergie ou bien pour la latence. Les figures 6.38, 6.39 et 6.40 montrent les trajectoires empruntées par le Sink mobile dans chaque scénario.



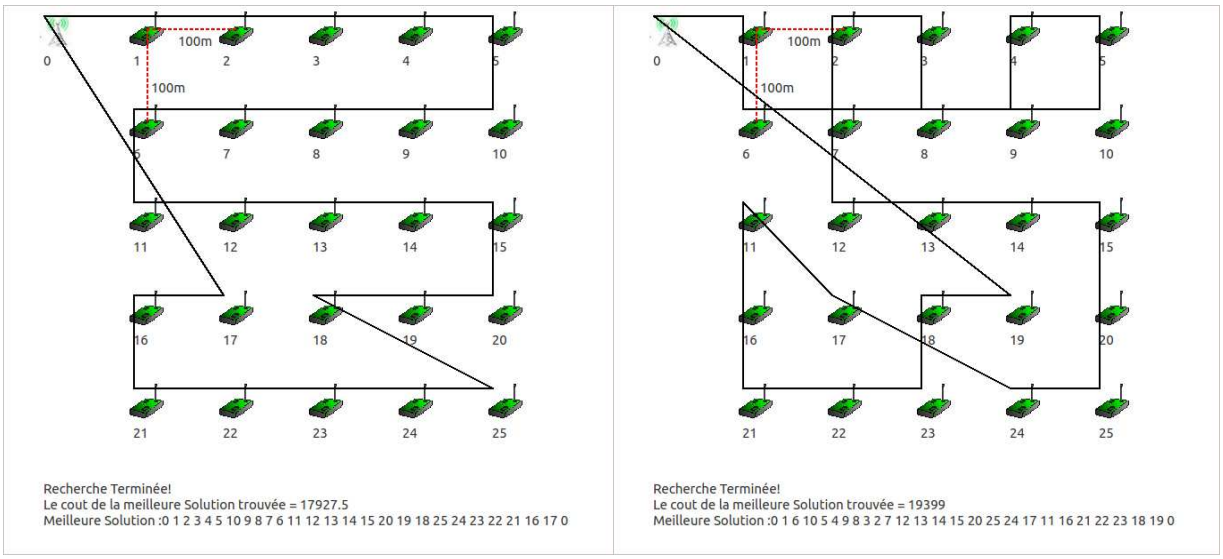
(a) Trajectoire calculée par le RS (b) Trajectoire calculée par la RT

Figure 6.38 – Trajectoire du SM calculée dans le scénario 1



(a) Trajectoire calculée par le RS (b) Trajectoire calculée par la RT

Figure 6.39 – Trajectoire du SM calculée dans le scénario 2



(a) Trajectoire calculée par le RS (b) Trajectoire calculée par la RT

Figure 6.40 – Trajectoire du SM calculée dans le scénario 3

6.3.7 Conclusion

Dans cette deuxième étape, nous avons utilisé deux méthodes de résolution approchées (Méta-heuristiques) et ceci en implémentant notre propre outil d'optimisation et de simulation EIOSM que nous avons étudiés en détail.

Les résultats obtenus par l'implémentation des méta-heuristiques (Recherche Tabou et Recuit Simulé) sont satisfaisants pour le calcul de la trajectoire du Sink mobile que ce soit en termes d'optimisation et de performance de réseau.

Le choix de la méthode d'optimisation reste le choix de l'utilisateur selon le cas d'utilisation et les objectifs visés. Par exemple, si l'on veut une intervention rapide sur le terrain, la Recherche Tabou offre alors une solution rapide et efficace. Si l'on fixe comme solution la bonne qualité, le Recuit Simulé donne alors de meilleures solutions grâce à la grande exploration de l'espace de recherche qui peut aboutir à un optimum global.

Comme perspective, nous proposons pour améliorer les déplacements du Sink mobile dans le RCSF, l'optimisation par contraintes en utilisant des méthodes exactes.

Chapitre 7

Conclusion générale

Sommaire

Conclusion

La technologie des RCSFs a connu ces dernières années un véritable engouement aussi bien dans l'industrie que dans le milieu universitaire grâce à leur très large spectre d'applications dans plusieurs domaines comme le monitoring environnemental, la surveillance, la médecine, domaine militaire, etc. Cependant, ces réseaux doivent aussi faire face à d'importants défis de conception en raison de leurs capacités limitées en termes de calcul, stockage, de bande passante et surtout en autonomie d'énergie (batteries). Ainsi, l'énergie est une ressource critique dans ces réseaux et constitue souvent un obstacle majeur de leur déploiement à grande échelle pour marquer leur omniprésence dans le monde de demain. Comme cette technologie est relativement récente (fin des années'90), plusieurs autres défis restent encore à relever en plus de la problématique énergétique qui influence énormément la durée de vie du réseau ; nous pouvons citer entre autres le routage avec qualité de service, connectivité et couverture, réduction du temps de latence, minimisation de la perte de paquets, gestion des incertitudes.
La nature stationnaire de ces réseaux a permis la génération de nouveaux problèmes liés essentiellement à la topologie et au fonctionnement du réseau. Parmi ces problèmes, nous pouvons citer : le goulot d'étranglement, la congestion du réseau, la non fiabilité des communications entre les nœuds capteurs et le nœud *Sink*. Ces problèmes conduisent à la dégradation sévère des performances du réseau et les rendent dans plusieurs cas quasi-inexploitables. Les développements, récents dans le domaine de la robotique mobile, ont permis l'émergence d'une nouvelle classe d'applications de RCSFs mobiles. Cette mobilité peut se traduite en termes de mobilité des nœuds capteurs, mobilité du *Sink* ou mobilité des nœuds relais. La mobilité des collecteurs (*Sink*, nœuds relais) avec des nœuds capteurs stationnaires constitue le principal objet de cette thèse.

L'utilisation des RCSFs à éléments mobiles (RCSF-EM) présente plusieurs avantages par rapport à l'utilisation des *Sinks* statiques notamment au niveau de l'amélioration des performances du réseau. Parmi ceux-ci nous citons :

- Faciliter le routage des données et assurer la connectivité et la connexité du réseau.
- Minimiser la consommation d'énergie des nœuds-capteurs qui constitue le cheval de bataille de toutes les solutions et protocoles proposés dans les RCSFs.
- Assurer une bonne couverture des zones à surveiller.
- Eviter la perte d'informations et gain de temps considérable.
- Augmenter la durée de vie des nœuds capteurs en réduisant les communications multi-sauts.
- Améliorer le débit ainsi que la sécurité et la fiabilité des données.

Cette nouvelle classe de RCSF-EM est ainsi un axe de recherche scientifique prometteur et à valeur ajoutée. Le nombre de travaux de recherche, relativement limité, que nous avons rencontrés dans notre étude bibliographique et pour lesquels il y a absence de résultats consensuels, nous a permis de se poser d'abord certaines questions de recherche relativement simples et pour lesquelles il n'y pas de réponses convaincantes scientifiquement avant de

s'attaquer aux véritables challenges. Ainsi, nous sommes convaincus que la démarche de recherche progressive et pédagogique adoptée dans cette thèse pour l'étude de ces RCSF-EM permet à notre avis d'améliorer sensiblement leur état de l'art.

7.1 Apports de la thèse

Nous avons étudié dans cette thèse trois problématiques liées à la gestion de la mobilité d'un élément mobile dans un RCSF et constaté son impact sur les performances du réseau.

Notre première contribution consiste à choisir le meilleur élément mobile à déplacer dans le réseau (nœud *Sink* ou nœud relais mobile) afin de collecter les données de manière fiable et efficace. Nous avons simulé un certain nombre de scénarios et lancé le processus de collecte avec deux variantes afin de déterminer l'approche la plus rentable : les collecteurs de données mobiles (*CDMs) de type Sink*, ou bien les *CDMs de type nœud relais*. Ces simulations ont été réalisées avec le simulateur OMNeT++ couplé à deux frameworks de mobilité « MiXiM » et « INeT» en utilisant « MiXiM-Inet-Buddle». Les résultats de simulation obtenus ont révélé que l'approche avec CDM de type Sink mobile offre plus de performance au réseau en termes de consommation d'énergie, de temps de latence et de perte de paquet.

Notre deuxième contribution consiste à choisir le meilleur modèle de mobilité parmi les trois modèles étudiés dans cette thèse (*Gauss-Markov,RandomWaypoint* et *Random Walk*) selon lequel nous devons déplacer notre élément mobile. Nous avons mis l'accent sur l'impact de ces modèles sur le processus de collecte des données. Plusieurs scénarios ont été testés sous le simulateur OMNET et la plateforme INET. Dans chaque scénario, nous avons fait varier le type de modèle de mobilité, la vitesse de déplacement du *Sink*, le temps de simulation ainsi que le nombre de capteurs. Les résultats de simulation ont démontré que le modèle à base de mouvement aléatoire avec temps de pause (*RandomWaypoint)* satisfait au mieux nos exigences et étend la durée de vie du réseau.

Notre troisième contribution propose une approche qui améliore et optimise la trajectoire de déplacement du *Sink*. En d'autres termes, obtenir un déplacement 'intelligent' pour gagner en énergie, en latence, éviter la perte de paquets, améliorer et faciliter la collecte de données ainsi que pour prolonger considérablement la durée de vie du réseau. Cette dernière représente l'ultime objectif de n'importe quelle application utilisant les RCSFs mobiles ou stationnaires.
L'approche proposée se base sur deux méthodes : méthode par simulation et méthode par résolution approchée.
La première méthode consiste à trouver un moyen efficace pour gérer et optimiser les déplacements du *Sink* dans un RCSF pendant la collecte de données. Nous avons implémenté notre propre modèle de mobilité basé sur le protocole de Clustering (LEACH) que nous avons modifié selon nos besoins. Ensuite, nous avons simulé un certain nombre de scénarios pour évaluer ses performances et les comparer avec d'autres modèles de mobilité sous l'environnement de simulation Castalia. Ce modèle permet au *Sink* d'emprunter une trajectoire optimale en utilisant les bons paramètres de déplacement (vitesse et temps de pause), type de mobilité contrôlée ainsi que le principe d'organiser le réseau en Clusters dans la perspective de limiter les déplacements entre le *Sink* et les capteurs pour une bonne collecte

de données. Les résultats obtenus ont montré que l'approche proposée (Clustering et mobilité contrôlée) réduit nettement les déplacements du *Sink* et la consommation d'énergie.

Néanmoins, cette approche augmente le temps de latence, comparé aux approches avec communication multi-sauts, et donc devient inefficace pour les applications sensibles aux délais et qui nécessitent l'envoi de données en temps réel.
L'objectif est donc de trouver un compromis entre l'énergie consommée et le temps de latence selon le type d'application pratique visée.

Dans la deuxième méthode, nous avons implémenté deux méta-heuristiques à savoir la *Recherche tabou* et le *Recuit simulé* que nous avons exécuté et simulé sur un environnement expérimental. Les résultats obtenus ont été comparés et analysés en termes d'optimisation (Meilleure solution obtenue, temps d'exécution…), et en termes de performance du réseau (Collecte des données, énergie consommée, latence…).

Dans les différents scénarios, les deux méthodes offrent des résultats appréciables. Le *Récuit simulé* donne de meilleurs résultats comparés à ceux obtenus dans la *Recherche tabou* grâce à la grande exploration de l'espace de recherche qui peut aboutir à un optimum global.

En termes de performance du réseau, nous avons remarqué que les deux méthodes utilisées pour le calcul de la trajectoire du *Sink* mobile, donnent des résultats satisfaisants que ce soit en termes de collecte de données, d'énergie et de latence grâce à l'optimisation de la trajectoire calculée.

7.2 Perspectives et futurs travaux

Nous proposons en perspective comme suite à ce travail de thèse les travaux de recherche futurs suivants :

- En plus des avantages apportés par la mobilité, évoqués dans cette thèse, nous pouvons en perspective exploiter aussi la mobilité pour pouvoir cette fois ci recharger des batteries des nœuds capteurs. L'opération de rechargement sans fil peut être étudiée par un problème d'optimisation de chemin emprunté par le chargeur mobile, avec comme objectif une fonction qui minimise le nombre d'emplacements d'arrêt dans le chemin.
- L'approche de collecte de données avec *Sinks*-multiples peut être aussi intéressante : essayer de diviser le réseau en plusieurs parties distinctes où chaque *Sink* opère dans une partie bien définie du réseau. Lorsque la quantité de données capturées par les nœuds capteurs est importante alors celle-ci sera transmise à l'un des *Sinks* multiples.
- La validation de nos résultats par expérimentation est envisageable. En effet, il s'agit de déployer un certain nombre de capteurs et de démarrer le processus de collecte de données en utilisant un *Sink* mobile représenté dans notre étude par un robot mobile. Le robot se déplace sur la zone de déploiement en passant par l'ensemble des capteurs afin de collecter les données qu'ils ont capturées, en considérant ses ressources illimitées. Ce futur travail se fera en collaboration avec le laboratoire de robotique de l'Université de l'USTO.MB

- Nous pouvons aussi augmenter la complexité de la mobilité en opérant sur un RCSF où tous les éléments qui le constituent sont mobiles. Dans ce cas, nous utiliserons un modèle de mobilité par groupes pour gérer les déplacements des nœuds capteurs, et un modèle de mobilité par entité pour les déplacements du *Sink*.
- Un autre aspect lié à la sécurité des communications et à l'authentification des éléments mobiles pourra faire l'objet de plusieurs études complémentaires à ce sujet.
- L'amélioration de la troisième contribution, en adoptant une optimisation par contraintes et en utilisant des méthodes exactes.

Communications Nationales et Internationales

1. **Y. Derdour**, B.Kechar, F. Khelfi. «Data Collection in WSNs with a Mobile Sink for a Supervision Application». Science and Information Conference (SAI 2013). IEEE Catalog Number: CFP13SAA-POD, ISBN: 978-1-4799-1272-8. ***Proceeding IEEE Explorer***. Londres du 7 au 9 octobre 2013.

2. **Y. Derdour**, B.Kechar, F. Khelfi. « Impact de l'Elément Mobile pour l'Amélioration des Performances d'un Réseau de Capteurs sans fil ». 6éme séminaire sur les systèmes de détection : Architecture et technologie DAT'2014, Alger (Algérie) du 17-19 février 2014.

3. **Y. Derdour**, B.Kechar, F. Khelfi. « Le Choix du Meilleur Elément Mobile pour Augmenter la Durée de Vie d'un Réseau de Capteur Sans Fil **».** La journée des doctorants, Université d'Oran. Algérie le 13 mars 2014.

4. **Y. Derdour**, B.Kechar, F. Khelfi. « The Impact of the Mobile Element on Performance Improvement in Wireless Sensor Network». The 5th International Conference on Ambient Systems, Networks and Technologies (ANT 2014). ***Proceeding Elsevier***. Hasselt, Belgique, Juin 2 - 5, 2014.

5. **Y. Derdour,** B.Kechar, F. Khelfi. « Choosing the Best Mobile Element for increasing the Performance of Wireless Sensor Network». International Conference on Advanced Networking, Distributed Systems and Applications (INDS 2014) Bejaia, Algérie, Juin 17-19, 2014.

Publication Internationale

Y. Derdour, B.Kechar, M.F.Khelfi . « Using Mobile Data Collectors to Enhance Energy Efficiency and Reliability in Delay Tolerant Wireless Sensor Networks ». Journal of Information Processing Systems (JIPS).ISSN :1976-913X (Print), ISSN :2092-805X (electronic). JIPS Journal is indexed in SCOPUS, DOI, DBLP, EBSCO, Google and Google Scholar. http://dx.doi.org/10.3745/JIPS.03.0032. (2015)

Références bibliographiques

[1] ***D. Roth.***«Gestion de la Mobilité dans les Réseaux de Capteurs sans Fil».Thèse de Doctorat. Université de Strasbourg, novembre 2012.

[2] ***J. Yick, B. Mukherjee, D. Ghosal.*** «Wireless Sensor Network Survey »Computer Networks, vol. 52, no. 12, pp. 2292–2330, 2008.

[3] ***Y. Derdour, B.Kechar, F. Khelfi***. **«**Impact de l'Elément Mobile pour l'Amélioration des Performances d'un Réseau de Capteurs sans fil ». 6éme séminaire sur les systèmes de détection : Architecture et technologie DAT'2014. , du 17-19, Algérie février 2014.

[4] ***Z. Bidai.***«Routage Multi-Chemin avec Qualité de Service pour le Transport d'un Trafic Scalaire/Multimédia dans un Réseau de Capteurs sans Fil ».Thèse de Doctorat. Université d'Oran, juin 2013.

[5] ***E.H.S. Mamour Diop. «*** Optimisation de la Transmission d'Images dans les Réseaux de Capteurs ». Thèse en cotutelle de doctorat en science : Université de Pau, France et Université Gaston Berger de Saint-Louis, Sénégal. Juin 2014.

[6] ***M. Aissani***. « Optimisation du Routage dans les Réseaux de Capteurs sans Fil pour les Applications Temps-Réel. Thèse en cotutelle de doctorat en science : Université de Paris-Est, France et Université USTHB, Algérie. Mars 2011.

[7] ***L. Makkaoui.*** « Compression d'Images dans les Réseaux de Capteurs sans Fil ». Thèse de Doctorat en Sciences de l'Université de Lorraine, spécialité Automatique. France, Novembre 2012

[8] ***V. Raghunathan, C. Schurgers, S. Park et M.B. Srivastava.*** « Energy-Aware Wireless Microsensor Networks ». Signal Processing Magazine, IEEE, 19(2),pp 40–50. 2002.

[9] ***E .Shih, S.H. Cho, N. Ickes, R. Min, A. Sinha, A. Wang et A. Chandrakasan*** « Physical Layer Driven Protocol and Algorithm Design for Energy-Efficient Wireless Sensor Networks ». In : Proceedings of the 7th annual international conference on Mobile computing and networking. ACM. pp. 272–287. 2001.

[10] ***B. Kechar***. «Problématique de la Consommation d'Energie dans les Réseaux de Capteurs sans Fil ».Thèse de doctorat d'état en informatique. Université d'Oran, Algérie ,2010.

[11] ***M. D. Francesco, S. K. Das, et Anastasi.*** « Data Collection in Wireless Sensor Networks with Mobile Elements : A Survey ». ACM Trans. Sens. Netw. TOSN, vol. 8, No 1, Article 7,aout 2011.

[12] ***N.Richard, N.Pazzi, A.Boukerche.*** « A Mobile Data Collector Strategy for Delay-Sensitive Applications over Wireless Sensor Networks ».Paradise Research Laboratory SITE, University of Ottawa, Canada, Available on line 3 January 2008.

[13] ***A. W. Khan, A. H. Abdullah, M. H. Anisi, J. I. Bangash. «*** A Comprehensive Study of Data Collection Schemes using Mobile Sinks in Wireless Sensor Networks », vol. 14(2), pp 2510–2548, 2014.

[14] ***Y. Derdour, B.Kechar, F. Khelfi***. « The Impact of the Mobile Element on Performance Improvement in Wireless Sensor Network». The 5th International Conference on Ambient Systems, Networks and Technologies (ANT 2014) Proceeding Elsevier. Hasselt, Belgique, Juin 2 - 5, 2014.

[15] ***Y. Derdour, B.Kechar, M.F.Khelfi***. « Using Mobile Data Collectors to Enhance Energy Efficiency and Reliability in Delay Tolerant Wireless Sensor Networks ». Journal of Information Processing Systems (JIPS). ISSN :1976-913X (Print), ISSN :2092-805X (electronic). JIPS Journal is indexed in SCOPUS, DOI, DBLP, EBSCO, Google and Google Scholar. 2015

[16] ***M.Becker, A.L.Beylot, R.Dhaou, A.Gupta, R. Kacimi, M.Marot***. « Experimental

Références bibliographiques

study : Link Quality and Deployment Issues in Wireless Sensor Networks ». In Proceedings of the 8th International IFIP-TC6 Networking Conference (Networking'09), volume 5550, pp 14_25, Aachen, Germany, Springer 2009.

[17] ***L.Jian, M.Prasant***. « Analytical Modeling and Mitigation Techniques for the Energy Hole Problem in Sensor Networks ». Pervasive and Mobile Computing, 3(3) pp:233_254, 2007.

[18] ***A. Kinalis, S. Nikoletseas, D. Patroumpa, J. Rolim. «*** Biased Sink Mobilitywith Adaptative Stop Times for Low Latency Data Collection in Sensor Networks », Information Fusion, vol. 15, pp. 56–63, 2012.

[19] ***A.Makhoul.*** « Réseaux de Capteurs: Localisation, Couverture et Fusion de Données ».Thèse de Doctorat. Université de Franche-Comté (LIFC), France .Novembre 2008

[20] ***P.Rawat, K. Deep Singh, H.Chaouchi, J.M.Bonnin.*** « Wireless Sensor Networks: A Survey on Recent Developments and Potential Synergies ».Revue : The Journal of super computing,Vol 68, No1, pp 1-48. Éditeur Springer US.2014

[21] ***F. Yu, S. Park, E. Lee, and S. H. Kim***. « Elastic Routing : A Novel Geographic Routing for Mobile Sinks in Wireless Sensor Networks » IET Communications, vol. 4, pp. 716–727, 2010.

[22] ***M. Ilyas***, « The Handbook of Ad hoc Wireless Networks », CRC Press, ISBN: 0-8493-1332-5, 2003.

[23] ***Y.Yousef.*** « Routage pour la Gestion de l'Energie dans les Réseaux de Capteurs Sans Fil ».Thèse de Doctorat d'état en informatique, Faculté des Sciences et Techniques Université de Haute Alsace, Juillet 2010.

[24] ***G. Chalhoub.*** « Routage et MAC dans les Réseaux de Capteurs sans Fil ».Thèse de doctorat.Université Blaise Pascal. Ecole Doctorale Sciences pour L'ingénieur de Clermont-Ferrand. 8 novembre 2010.

[25] ***B. Krishnamachari, D. Estrin, S. Wicker***, « Modelling Data-Centric Routing in Wireless Sensor Networks ». USC Computer Engineering Technical Report CENG pp,02-14, 2002.

[26] ***M.A .Matin.,M.M.Islam.*** «Overview of Wireless Sensor Network». Creative Commons Attribution License, 2012.

[27] ***C.T.KONE***. « Conception de l'Architecture d'un Réseau de Capteurs sans Fil de Grande Dimension ». Thèse de Docteur. Université Henri Poincaré, Nancy I,Octobre 2011

[28] ***J. M. Kahn, R. H. Katz, K. S. J. Pister***, « Next CenturChallenges:MobileNetworking for Smart Dust ». Proc. of the 5th Annual ACM/IEEE Int. Conf. on Mobile Computing and Networking (MobiCom 99), Seattle, WA, pp. 271, 278, August 1999.

[29] Mica Sensors Specification, http://www.xbow.com

[30] TinyOS Community Forum. http://www.tinyos.net

[31] ***P. Misra ,P. Enge***. « Global Positioning System: Signals, Measurements, and Performance ». Book Review, Ganga-Jamuna Press, Lincoln, Massachusetts, 2001

[32] ***T. Halonen, J. Romero, J. Melero***. « GSM, GPRS and EDGE Performance », John Wiley & Sons Ltd, ISBN: 0-470-86694-2, 2003.

[33] IEEE, Information Exchange between Systems Local, Metropolitan Area Network Specific Requirements Part 11 : Wireless LAN Medium Access Control (MAC), Physical Layer (PHY) Specifications. The Institute of Electrical, and Electronics Engineers, 1997.

[34] http://wirelessman.org/

[35] ***N. Labraoui***. « La Sécurité dans les Réseaux sans Fil Ad-hoc », thèse de doctorat. Université de Tlemcen Algérie 2012

Références bibliographiques

[36] « Crossbow technologie Inc », http://www.xbow.com

[37] « Wireless Sensors Platform,W.S.NMICA-Familly Wireless Mote Platform Specifications »

[38] « Wireless Sensors Platform c, Intel Platform » 2010, http://www.xbow.com/Products/productdetails.aspx?sid=280

[39] « Wireless Sensors Platform, W.S.N tmoteSky Platform specifications »,http//www.sentilla.com/moteiv-transition.html

[40] http://www.tinynode.com/uploads/media/TinyNode_Users_Manual_rev11.pdf

[41] http://www.memsic.com/userfiles/files/Datasheets/WSN/telosb_datasheet.pdf

[42] ***I.Akyildiz ,W.Su, Y. Sankarasubramaniam,E. Cayirci***. « A Survey on Sensor Networks ». IEEE Commun Mag 40(8):102–114. 2002

[43] ***I.Akyildiz ,E. Stuntebeck***. « Wireless Underground Sensor Networks: Research Challenges ». Ad-Hoc Netw 4(6):pp 669–686. 2006

[44] ***M.Li ,Y.Liu***. « Underground Structure Monitoring with Wireless Sensor Networks ». In: Proceedings of the 6th international conference on information processing in sensor networks. ACM, New York, p 78.

[45] ***I.Akyildiz ,D.Pompili ,T. Melodia.*** « Challenges for Efficient Communication in Underwater Acoustic Sensor Networks ». ACM Rev 1(2):8. 2004

[46] ***J.Heidemann,Y. Li ,A .Syed, J.Wills,W. Ye***. « Underwater Sensor Networking: Research Challenges and Potential Applications ». In: Proceedings of the IEEE wireless communications and networking conference. 2006

[47] ***I.Akyildiz ,T.Melodia T, K.Chowdhury***. « A Survey on Wireless Multimedia Sensor Networks ». Comput Network 51(4)pp:921–960. 2007

[48] ***J.Yick , B.Mukherjee, D.Ghosal .*** « Wireless Sensor Network Survey ». Computing Network 52(12):pp2292–2330. 2008

[49] ***S. A.H.Sedjelmaci***. « Mise en Œuvre de Mécanisme de Sécurité Basée sur les IDS pour les Réseaux de Capteurs sans Fil », Thèse de doctorat; Février 2013.

[50] ***H.Saito,S. Shimogawa ,S.Tanaka, S.Shioda***. « Estimating Parameters of Multiple Heterogeneous Target Objects using Composite Sensor Nodes ». IEEE Trans Mob Comput 11(1):pp125–138.2012

[51] ***G.Simon et al***. « Sensor Network-Based Countersniper System ». In: Proceedings of the 2nd international conference on embedded networked sensor systems. ACM, New York, pp 1–12.2004

[52] ***J.Huang,S. Amjad ,S. Mishra***. CenWits: « A Sensor-Based Loosely Coupled Search and Rescue System using Witnesses ». In: Proceedings of the 3rd international conference on embedded networked sensor systems. ACM, New York, pp 191. 2005

[53] VigilNet. http://www.cs.virginia.edu/wsn/vigilnet/

[54] ***P.Zhang, C.Sadler, S.Lyon ,M.Martonosi***. « Hardware Design Experiences in ZebraNet ». In:Proceedings of the 2nd international conference on embedded networked sensor systems. ACM,New York, pp 227–238. 2004

[55] Harvard Sensor Networks Lab. « Volcano Monitoring ». Available online: http://fiji.eecs.harvard.edu/Volcano

[56] ***J.Burrell,T. Brooke,R. Beckwith***. « Vineyard Computing: Sensor Networks in Agricultural Production ». IEEE Pervasive Comput 3(1):pp38–45. 2004

[57] ***K.Lorincz et al*** « Mercury: A Wearable Sensor Network Platform for HighFidelity Motion Analysis. In: Proceedings of the 7th ACM conference on embedded networked sensor systems. ACM, New York, pp 183–196. 2009

[58] ***V.Weber V***. « Smart Sensor Networks: Technologies and Applications for Green Growth. The Organization for Economic Cooperation and Development »2009

[59] ***Y .Kim , T.Schmid, Z.M.Charbiwala,J. Friedman,M.B.Srivastava.*** « NAWMS:

Nonintrusive Autonomous Water Monitoring System ». In: Proceedings of the 6th ACM conference on embedded network sensor systems. ACM, New York, pp 309–322. 2008

[60] ***T.B. Gosnell, J.M. Hall, C.L. Ham, D.A. Knapp, Z.M. Koenig, S.J. Luke, B.A. Pohl,A. Schach von Wittenau, J.K. Wolford***. « Gamma-Ray Identification of Nuclear Weapon Materials ». Technical Report DE97053424. Lawrence Livermore National Lab., Livermore, CA (USA). February 1997

[61] ***R.Merzougui, M.Feham, H.Sedjelmaci***. «Design and Implementation of An Algorithm for Cardiac Pathologies Detection on Mobile Phone». International Journal of Wireless InformationNetworks, 18 (1):pp11-23, 2011.

[62] ***K.Beydoun***, « Conception d'un Protocole de Routage Hiérarchique pour les Réseaux de Capteurs ». Thèse de Doctorat en informatique, Université de Franche-Comté, France, Décembre 2009.

[63] ***M. Fitzgerald***. «Technnology Review: Tracking a Shopper's Habits », August 2008. http://www.technologyreview.com/computing/21161/

[64] ***A.Loureiro, R.Linnyer.*** «Autonomic Wireless Networks in Smart Environments », Fifth Annual Conference on Communication Networks and Services Research, CNSR apos,, pp. 5–7.2007

[65] ***G.Werner-Allen, K.Lorincz, M.Welsh et al.,*** « Deploying a Wireless Sensor Network on an Active Volcano». IEEE Internet Computing, vol.10, no2,pp. 18–25. April 2006

[66] « http ://www.argo.ucsd.edu/ [accessed 17 06 2008] ».

[67] ***K.Chintalapudi,T. Fu, J.Paek et al.*** « Monitoring Civil Structures with a Wireless Sensor Network », IEEE Internet Computing, vol. 10, no2, pp. 26–34.2006

[68] ***C.Herring, S.Kaplan***. «Component Based Software Systems for Smart Environments », IEEE Personal Communications, pp. 60–61. October 2000

[69] **N.Kouadria. «** Codage et Compression d'Images pour des Réseaux de Capteurs sans fil Multimédia et Communications » .Doctorat 3éme cycle 2013/2014

[70] ***R.KACIMI***. « Techniques de Conservation d'Energie pour les Réseaux de Capteurs sans fil ». Thèse de Doctorat, Institut National Polytechnique de Toulouse Spécialité : Réseaux et Télécommunications, 28/09/2009.

[71] ***Q.Lampin***. « Réseaux Urbains de Capteurs sans-fil : Applications, Caractérisation et Protocoles ». Thèse de doctorat. Institut National des Sciences Appliquées de Lyon. Projet INRIA Urbanet, Laboratoire CITI, INSA de Lyon30 janvier 2014

[72] Texas Instruments. CC1011 chip specifications. [Online]. Available : {http://focus.ti.com/docs/prod/folders/print/cc1101.html} june,2011.

[73] ***J. Postel***, «Transmission Control Protocol (TCP) », RFC 6550, 1981.

[74] ***C.DINI***, « Mécanismes de Traitement des Données dans les Réseaux de Capteurs sans fils dans le cas d'Accès Intermittent à la Station de Base ». Thèse de Doctorat, Université de Haute Alsace. Faculté des Sciences et Techniques, année 2010

[75] ***Y. Xu, J.Heidemann, D.Estrin.*** « Geography-Informed Energy Conservation for Ad-hoc Routing ». In Proceedings of the 7th annual international conference on Mobile Computing and networking (MobiCom'01), pp 70_84, New York, NY, USA, .ACM 2001.

[76] ***H.Philipp ,T.Braun***. « Calibrating Wireless Sensor Network Simulation Models with Real-World Experiments ». In Proceedings of the 8th International IFIP-TC6 Networking Conference (Networking'09), volume 5550, pp 1_13, Singapore, May 5-6 2009.Springer. 2009

[77] ***G .J. Pottie,W. J. Kaiser***. « Wireless Integrated Network Sensors ». Communications of the ACM, 43(5) :pp51_58, 2000.

[78] ***C. Alippi, G.Anastasi, C.Galperti, F.Mancini, M.Roveri***. « Adaptive Sampling for

Références bibliographiques

Energy Conservation in Wireless Sensor Networks for Snow Monitoring Applications ». In Proceedings of the 4th IEEE International Conference on Mobile Ad-hoc and Sensor Systems (MASS'07), pp 1_6, Pisa, Italy, October 2007.

[79] ***W.Ye, J. Heidemann, D.Estrin***. « Medium Access Control with Coordinated Adaptive Sleeping for Wireless Sensor Networks ». IEEE/ACM Transactions on Networking,12(3) :pp493_506. 2004

[80] ***M.Ali, A. Böhm, M.Jonsson***. « Wireless Sensor Networks for Surveillance Applications - A comparative Survey of MAC Protocols ». In Proceedings of the 4th International Conference on Wireless and Mobile Communications (ICWMC '08), pages 399_403, Washington, DC, USA. IEEE Computer Society. 2008

[81] ***K. Holger, A.Willig***. « Protocols and Architectures for Wireless Sensor Networks ». Wiley, 2005.

[82] ***Y. Benabassi***. « Application de la Redondance pour la Surveillance par Réseau de Capteurs sans fil : Cas du Réseau de Capteurs Images sans fil ». Doctorat en science à l'Université d'Oran.2014

[83] ***T. Armstrong***. « Wake-up Based Power Management in Multi-hop Wireless Networks », Term Survey Paper, University of Toronto, available at http ://www.eecg.toronto.edu/ trevor/Wakeup/index.html. 2005

[84] ***B.Blywis ,M.Günes, F.Juraschek, O.Hahm, N. Schmittberger*** . « Flooding and Gossiping : A Survey of Flooding, Gossip Routing, and Related Schemes for Wireless Multi-Hop Networks ». Computer Systems and Telematics. Institute of Computer Science. Freie Universität Berlin, Germany October 10, 2011

[85] ***C. Intanagonwiwat, R. Govindan, D. Estrin***. « Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks ». In the Proceedings of the 6th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom'00), Boston, MA, August 2000.

[86] ***H. Qi, P. T. Kuruganti, Y. Xu***. « The Development of Localized Algorithms in Wireless Sensor Networks ». Published on 202 Sensors ISSN, 22 July, pp. 1424-8220.2002

[87] ***C. Schurgers, M. B. Srivastava***. « Energy Efficient Routing in Wireless Sensor Networks ». In the MILCOM Proceedings on Communications for Network-Centric Operations: Creating the Information Force, McLean, VA, 2001.

[88] ***E. M. Royer, C. K. Toh. «*** A Review of Current Routing Protocols for Ad-Hoc Mobile Wireless Networks ». IEEE Personal Communications, Vol. 6, No. 2, April 1999,pp. 46-55.

[89] ***S. A. Ahsan Rajon***. « Energy Efficient Data Communication for Resource Constrained Systems ». Livre Lambert academic publishing 2014.

[90] ***A. Bachir, M. Dohler, T. Watteyne, K. Leung***. « MAC Essentials for Wireless Sensor Networks ». IEEE Communications Surveys Tutorials, vol. 12, no. 2, pp. 222 –248, 2010.

[91] ***Y. Derdour, B.Kechar, F. Khelfi***. « Data Collection in WSNs with a Mobile Sink for a supervision application » .Science and Information Conference(SAI 2013).**,**IEEE Catalog Number: CFP13SAA-POD ,ISBN:978-1-4799-1272-8 .Proceeding IEEE Explorer, Londres, du 7 au 9 octobre 2013.

[92] ***G.Chauhan***. « Traffic-adaptive Medium Access Protocol Published » Project Trainee at Nirma University of Science and Technology. 04 mars 2013

[93] ***K.S. J. Pister, L.Doherty***. « TSMP: Time Synchronized Mesh Protocol » Proceeding of the IASTED international symposium Distributed sensors networks Orlondo, Florida USA. (DSN 2008)

[94] ***T. van Dam,K. Langendoen***. « An Adaptive Energy Efficient MAC Protocol for

Wireless Sensor Networks » in Proceedings of the *1*st international conference on Embedded networked sensor systems (SenSys'*03*), (New York, NY, USA), pp. 171–180, ACM, Novembre 2003.

[95] IEEE Computer Society, "IEEE Standard for Information Technology - Telecommunications and Information Exchange Between Systems- Local and Metropolitan Area Networks- Specific Requirements Part 15.4 : Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs)," IEEE Std *802.15.4-2006* (Revision of IEEE Std *802.15.4-2003*), pp. 1–305, Septembre 2006.

[96] ***W.Ye,J.Heidemann*** . « SCP-MAC: Reaching Ultra-Low Duty Cycles ». http://secon2005.ieee-secon.org

[97] ***A.Roy, N.Sarma***. « AEEMAC: Adaptive Energy Efficient MAC Protocol for Wireless Sensor Networks ». India Conference (INDICON), 2011 Annual IEEE, pp 1-6, 16-18,Dec. 2011.

[98] ***M. Ali, T. Suleman, Z. Uzmi***. « MMAC : A Mobility-Adaptive, Collision-Free MAC Protocol for Wireless SensorNetworks » in Proceeding of the 24th IEEE InternationalPerformance, Computing, and Communications Conference(IPCCC'05), pp. 401–407, Avril 2005.

[99] ***H. Pham, S. Jha***. « An Adaptive Mobility-Aware MAC Protocol for Sensor Networks (MS-MAC) » in IEEE International Conference on Mobile Ad-hoc and Sensor Systems (MAHSS'*04*), pp. 558–560, Octobre 2004.

[100] ***J. Polastre, J. Hill, D. Culler***. « Versatile Low Power Media Access for Wireless Sensor Networks » in Proceedings of the *2*nd international conference on Embedded networked sensor systems (SenSys'*04*), pp. 95–107, ACM, Novembre 2004.

[101] ***M. Buettner, G. V. Yee, E. Anderson, R. Han***. « X-MAC : A Short Preamble MAC Protocol for Duty-Cycled Wireless Sensor Networks » in proceedings of the *4*th international conference on embedded networked sensor systems (SenSys'06), (New York, NY, USA), pp. 307–320, ACM, Novembre 2006.

[102] ***R. Musaloiu-E., C.-J. M. Liang, A. Terzis***. « Ultra-Low Power Data Retrieval in Wireless Sensor Networks » in Proceedings of the *7*th international conference on Information processing in sensor networks (IPSN'*08*), (Washington, DC, USA), pp. 421–432, IEEE Computer Society, Avril 2008.

[103] ***Y. Sun, O. Gurewitz, D. B. Johnson***. « RI-MAC : AReceiver-Initiated Asynchronous Duty-Cycle MAC Protocolfor Dynamic Traffic Loads in Wireless Sensor Networks » inProceedings of the 6th ACM conference on Embedded networksensor systems (SenSys'*08*), (New York, NY, USA), pp. 1–14,ACM, Novembre 2008.

[104] ***A. El-Hoiydi, J.-D. Decotignie***, « WiseMAC : An Ultra Low Power MAC Protocol for Multi-hop Wireless Sensor Networks », in Algorithmic Aspects of Wireless Sensor Networks (S. Nikoletseas and J. Rolim, eds.), vol. 3121 of Lecture Notes in Computer Science, pp. 18–31, Springer Berlin /Heidelberg, 2004.

[105] ***R. Kuntz, J. Montavont, T. Noël***. «Improving the Medium Access in Highly Mobile Wireless Sensor Networks ». Telecommunication Systems, pp. 1–22, 2011.

[106] ***L. Bing, Y. Ke, Z. Lin, Z. Huimin***, « Mac Performance and Improvement in Mobile Wireless Sensor Networks », in: Eighth ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing, SNPD 2007, vol. 3, 2007, pp. 109–114, doi:http://dx.doi.org/10.1109/SNPD.2007.

[107] ***L. Bernardo, H. gua, M. Pereira, R. Oliveira, R. Dinis, P. Pinto,*** « A Mac Protocol for Mobile Wireless Sensor Networks with Bursty Traffic », in: Wireless Communications and Networking Conference (WCNC), 2010 IEEE, 2010, pp. 1–6, doi:http://dx.doi.org/10.1109/WCNC.2010.5506145.

Références bibliographiques

[108] ***V. Rajendran, K. Obraczka, J. J. Garcia-Luna-Aceves***. « Energy-Efficient, Collision-Free Medium Access Controlfor Wireless Sensor Networks ». Wireless Networks, vol. 12,pp. 63–78, Février 2006.

[109] ***K. S. J. Pister ,L. Doherty***, «TSMP : Time Synchronized Mesh Protocol», in Proceedings of International Symposium on Distributed Sensor Networks (DSN'*08*), pp. 391— 398, IEEE/IFIP, Juin 2008.

[110] ***I.Rhee, A. Warrier, M.Aia, J. Min, M. L. Sichitiu***. « Z-MAC :A Hybrid MAC for Wireless Sensor Networks ». IEEE/ACM Transactions on Networking, 16(3) :pp511_524. 25, 27. 2008

[111] ***R.Silva, J. Sa Silva, F.Boavida***. « Mobility in Wireless Sensor Networks – Survey and Proposal ». Contents lists available at ScienceDirect Computer Communications journal homepage: www.elsevier.com/locate/comcom. 28 May 2014

[112] ***C.Perkins, P.Bhagwat***, « Highly Dynamic Destination-Sequenced Distance-Vector Routing DSDV For Mobile Computers » ACM SIGCOMM Computer communication review,vol 24,n 4,pp 234-244,1994

[113] ***T.Clausen, P.Jacquet***, « Optimized Link State Routing Protocol OLSR,IETF RCF » 3626,Octobre 2003

[114] ***M. Ali, S. K. Ravula***. « Real-Time Support and Energy Efficiency in Wireless Sensor Networks ». Technical report, IDE0805, Janvier 2008.

[115] ***K. Akkaya, M. Younis***. « A Survey on Routing Protocols for Wireless Sensor Networks ». Journal of Ad Hoc Networks, Vol. 3, No. 3, May 2005.

[116] ***C.Perkins, E.Royer, S.Das***, « Ad-Hoc On Demand Distance Vector (AODV) Routing ».Online.available :http://www.ietf.org/internet-draft/draft-ieftmanet-aodv-03.txt.1999

[117] ***D.B.Johnson,D.A.Maltz, J.Broch***. « DSR :The Dynamic Source Routing Protocol For Multi-Hop Wireless Adhoc Networks ».Addison wesley,ch,5,pp,139-172.2001

[118] ***Z.Haas, M.Pearlman, P.Samar***. « The Zone Routing Protocol»,internet-Draft, draft,draftietfmanetzoneszrp04,txt,work in progress.2004

[119] ***N. V. Subramanian***. « Survey on Energy-Aware Routing and Routing Protocols for Sensor Networks ». Technical Report, Computer Science, University of North Carolina, Charlotte. 2004.

[120] ***J.N.Karaki, E.Kamal***, « Routing Techniques In Wireless Sensor Network :A Survey »IEEE Wirel.Commun,vol,11,n 6,pp 6-28,2004

[121] ***Y.Yao J.,Gehrke***. « The Cougar Approach to in Network Query Processing in Sensor Networks ». SIGMOD Rec 2002

[122] ***V. Rodoplu ,T. H. Ming***. « Minimum Energy Mobile Wireless Networks ». IEEE Journal of Selected Areas in Communications, Vol. 17, No. 8, 1999, pp. 1333-1344.

[123] ***A. Manjeshwar, D. P. Agrawal***. « TEEN : A Protocol for Enhanced Efficiency in Wireless Sensor Networks ». 1st International Workshop on Parallel and Distributed Computing Issues in Wireless Networks and Mobile Computing, San Francisco, CA, April 2001.

[124] ***Y. Xu, J. Heidemann, D. Estrin***. « Geography-Informed Energy Conservation for Ad-hoc Routing ». In Proceedings of the 7th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM'01), Rome, Italy, July 2001.

[125] ***Y. Yu, D. Estrin, R. Govindan***. « Geographical and Energy-Aware Routing: A Recursive Data Dissemination Protocol for Wireless Sensor Networks ». UCLA Computer Science Department Technical Report, UCLA-CSD TR-01-0023, Mai 2001.

[126] ***K. Sohrabi, J. Pottie***. « Protocols for Self-Organization of a Wireless Sensors

Network ». IEEE Personal Communications, Vol. 7(5), pp. 16–27, October 2000.

[127] ***L.T. Nguyen, X. Defago, R. Beuran, and Y. Shinoda***. « An Energy-Efficient Routing Scheme for Mobile Wireless Sensor Networks ». Proc. of the 5 IEEE Int'l Symposium on Wireless Communication Systems, pp.568-572, Reykjavik, Iceland, October 21-24, 2008.

[128] ***B. Karp and H. Kung***. « GPSR: Greedy Perimeter Stateless Routing for Wireless Networks ». Proc. of the ACM/IEEE Conference on Mobile Computing and Networking (MobiCom), pp. 243-254, Boston, Massachusetts, USA, August 6-11, 2000.

[129] ***H.Karl, A. Willig***. « Protocols and Architectures for Wireless Sensor Networks. Wiley-Interscience.2007

[130] ***J.LI, P. Mohapatra***. « Analytical Modeling and Mitigation Techniques for The Energy Hole Problem in Sensor Networks ». Pervasive Mob. Comp. 3, 3, pp233–254. 2007

[131] ***W.Zhao, M. Ammar*** « Message Ferrying: Proactive Routing in Highly-Partitioned Wireless Ad-hoc Networks ». In Proceedings of the 9th IEEE International Workshop on Future Trends of Distributed Computing Systems (FTDCS'03).pp 308–314. 2003

[132] ***Fall.*** « A Delay-Tolerant Network Architecture for Challenged Internets ». In Proceedings of the Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications (SIGCOMM'03).pp 27–34. 2003

[133] ***R.C.Shah, S.Roy, S.Jain, W.Brunette.*** « Data Mules: Modeling a Three-Tier Architecture for Sparse Sensor Networks ». In Proceedings of the 2nd ACM International Workshop on Wireless Sensor.2003

[134] ***G.Anastasi,M. Conti, M. DI Francesco, A. Passarella.*** « Energy Conservation in Wireless Sensor Networks: A Survey ». Ad Hoc Netw. 7, 3,pp 537–568. 2009

[135] ***A. Kansal, A. Somasundara, D.Jea ,M.Srivastava,D.Estrin.*** « Intelligent Fluid Infrastructure for Embedded Networks ». In Proceedings of the 2nd ACM International Conference on Mobile Systems, Applications, and Services (MobiSys'04).pp 111–124. 2004

[136] ***S.Gandham, M.Dawande,R. Prakash, S.Venkatesan.*** « Energy Efficient Schemes for Wireless Sensor Networks with Multiple Mobile Base Stations ». In Proceedings of the 46th IEEE Global Telecommunications Conference Workshops (GlobeCom'03).pp 377–381. 2003

[137] ***G. Wang, G.Cao, T.La Porta,W.Zhang.*** « Sensor Relocation in Mobile Sensor Networks ». In Proceedings of the 24th IEEE Conference on Computer Communications (INFOCOM'05). Vol. 4.pp 2302–2312. 2005

[138] ISO/IEC, "Reference Model of Open Systems Interconnection ; Part 1, Basic Reference Model (incorporating connectionless-mode transmission)," tech. rep., British Standards Institution, 2 Park Street, London W1A 2BS, UK, 1988. BS 6568 : Part 1 : 1988 (⌘ ISO 7498–1984 including Amendment *1*).

[139] ***A.Chakrabarti, A.Sabharwal, B.Aazhang***. « Using Predictable Observer Mobility for Power Efficient Design of Sensor Networks ». In Proceedings of the 2nd International Workshop on Information Processing in Sensor Networks (IPSN'03), pp 129_145, Palo Alto, CA, USA, 2003.

[140] ***A.Somasundara., A.Kansal, D.Jea,D.Estrin,M.Srivastava.*** « Controllably Mobile Infrastructure for Low Energy Embedded Networks ». *IEEE Trans. Mob. Comp. 5*, 8, pp1536–1233. 2006

[141] ***S.Poduri, G.S.Sukhatme.*** « Achieving Connectivity Through Coalescence in Mobile Robot Networks ». In Proceedings of the 1st International Conference on Robot Communication and Coordination (RoboComm'07). 1–6. 2007

Références bibliographiques

[142] ***E. B. Hamida***. « Modélisation Stochastique et Simulation des Réseaux sans fil Multi-Sauts ». Ph.D. dissertation, Institut National des Sciences Appliquées (INSA) de Lyon, France, Septembre 2009.

[143] ***C. C. Shen, C. Srisathapornphat, and C. Jaikaeo***. « Sensor Information Networking Architecture and Applications ». IEEE Personal Communications, pp 52–59, August 2001.

[144] ***S.Sharma.Y.Shi , Y.T.Hou, H.D.Sherali.*** « Joint Optimization of Session Grouping and Relay Node Selection for Network-Coded Cooperative Communications ». Mobile Computing, IEEE Transactions on (Volume:13, Issue: 9) pp: 2028 – 2041, ISSN :1536-1233, IEEE Computer Society. 30 juillet 2014

[145] ***L.Khelladi, D. Djenouri, N.Lasla, N.Badache*** .« MSR : Minimum-Stop Recharging Scheme for Wireless Rechargeable Sensor Networks » IEEE International Conference on Ubiquitous Intelligence and Computing/International Conference on Autonomic and Trusted Computing/International Conference on Scalable Computing and Communications and Its Associated Workshops. CERIST Center of Research. Algeria 2014.

[146] ***M.I. Khan a, N. Wilfried, B. Gansterer, G. Haring***. « Static Vs. Mobile Sink: The Influence Of Basic Parameters On Energy Efficiency In Wireless Sensor Networks ». Contents lists available at SciVerse ScienceDirect Computer Communications journal homepage: www.elsevier.com/locate/comcom,pp 14, Model 5G Elsevier. 15 novembre 2012.

[147] ***I. Chatzigiannakis, A. Kinalis, S. Nikoletseas***. « Sink Mobility Protocols for Data Collection in Wireless Sensor Networks », in: Proceedings of the international 1183 .Workshop on Mobility Management and Wireless Access, MobiWac '06, 1184 Terromolinos,, pp. 52–59 ,Spain, 2006.

[148] ***R.Rubia, .S.A.Selvan.*** « A Survey on Mobile Data Gathering in Wireless Sensor Networks » - Bounded Relay International Journal of Engineering Trends and Technology (IJETT) – Volume 7 Number 5- Jan. Kumaraguru College of Technology, Coimbatore-49tamil nadu india. 2014

[149] ***M. Ma and Y. Yang.*** « SenCar: An Energy-Efficient Data Gathering Mechanism for Large-Scale Multihop Sensor Net works ». IEEE Trans. Parallel and Distributed Systems, vol. 18, no. 10, pp. 1476-1488, Oct. 2007.

[150] ***V.Shrinithi, R.Rohini***. « A Survey on Data Collection in Wireless Sensor Network with Mobile Elements ».International Journal of Latest Trends in Engineering and Technology (IJLTET). College of Engineering for Women, Tiruchengode,Tamil Nadu, India.2013

[151] ***C.Liu, K. Wu,J.Pei***. « An Energy Efficient Data Collection Framework for Wireless Sensor Networks by Exploiting Spatiotemporal Correlation »2007.

[152] ***D.J.Parker***. « Exploiting Temporal and Spatial Correlation in Wireless Sensor Networks ». Thèse au Département d'Electronique et d'informatique .Northeastern Université de Boston, Massachusetts Avril, 2013

[153] ***M. Alzoubi ,P.J.Wan, O.Frieder***. « New Distributed Algorithm for Connected Dominating Set in Wireless Ad-Hoc Networks ». Department of Computer Science, Hawaii International Conference on System Sciences – 2002

[154] ***T.Cui, L.Chen, T.Ho, H.S.Low,L.H.A.Lachlan***. « Opportunistic Source Coding for Data Gathering in Wireless Sensor Networks » DARPA grant N66001-06-C- 2020, Caltech's Lee Center for Advanced Networking.2007

[155] ***S.Pattem, B.Krishnama chari, R.Govindan***. « The Impact of Spatial Correlation on Routing with Compression in Wireless Sensor Networks ». ACM Transactions on Sensor Networks (TOSN), vol. 4, no. 4, pp. 24,2008.

Références bibliographiques

[156] ***R.Bechar***. « Théorie de la Redondance pour la Reconfiguration des Systèmes – Application aux Réseaux de Capteurs sans fil ».Thèse de Magister, Université Abdelhamid Ibn Badis de Mostaganem, Ecole doctorale STIC, Algérie, 2009.

[157] ***T.Srinidhi.,G.Sridhar, V.Sridhar.*** « Topology Management In Ad-Hoc Mobile Wireless Networks ». In Proceedings of the 24th IEEE International Real-Time Systems Symposium (RTSS'03). (Work-in-progress session).pp 29–32. 2003.

[158] ***H.Ge, Q.Guo, H.Sun, B.Wang, B.Zhang, W .Wu***.«A Load Fluctuation Characteristic Index and its Application to Pilot Node Selection ISSN 1996-1073. www.mdpi.com/journal/energies. Janvier 2014

[159] ***G.Dini,M. Pelagatti,I.M. Savino.*** « An Algorithm for Reconnecting Wireless Sensor Network Partitions ». In Proceedings of the 5th European conference on Wireless Sensor Networks (EWSN'08).pp 253–267. 2008

[160] ***F.El-Moukaddem.,E.Torn.,G.Xing,S. Kulkarni.*** « Mobile Relay Configuration in Data Intensive Wireless Sensor Networks ». In Proceedings of the 6th IEEE International Conference on Mobile Ad Hoc and Sensor Systems (MASS'09).pp 80–89. 2009

[161] ***G.Wang, Cao, T.La Porta***. « Movement-Assisted Sensor Deployment ». IEEE Trans. Mob. Comp. 5, 6,pp 640–652. 2006

[162] ***G.Wang,G.Cao,P.Berman.,T.La Porta.*** « Bidding Protocols for Deploying Mobile Sensors ». IEEE Trans. Mob. Comp. 6, 5, pp563–576. 2007

[163] ***W. Wang, V. Srinivasan, K. C. Chua***. « Using Mobile Relays to Prolong the Lifetime of Wireless Sensor Networks » in Proceedings of the 11th Annual International Conference on Mobile Computing and Networking (MobiCom 05),pp. 270–283. (Cité en page 89.) 2005

[164] ***S.Yoon,O. Soysal,M.Demirbas,C.Qiao.*** « Coordinated Locomotion of Mobile Sensor Networks ». In Proceedings of the 5th IEEE Conference on Sensor and Ad Hoc Communications and Networks (SECON'08).pp 126–134. 2008.

[165] ***A.Deshpande, S.Poduri,D.Rus,G.S.Sukhatm***. « Distributed Coverage Control for Mobile Sensors with Location-Dependent Sensing Models ». In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA'09).pp 3493–3498. 2009

[166] ***Y.C.Wang, C.C.Hu.*** « Efficient Placement and Dispatch of Sensors in a Wireless Sensor Network ». IEEE Trans. Mob. Comp. 7, 2,pp 262–274. 2008.

[167] ***G. Anastasi.E.Borgia, M.Conti,E.Gregori***. « A Hybrid Adaptive Protocol for Reliable Data Delivery in WSNs with Multiple Mobile Sinks ».The Computer Journal.Vol 2,pp 213-229. Oxford University Press

[168] ***M.Conti,L.Pelusi,A.Passarella,G .Anastasi.«*** Mobile-RelayForwardingOpportunistic Networks ». In Adaptation and Cross Layer Design in Wireless Networks. CRC Press, Boca Raton, FL,pp389–418. 2008.

[169] ***A.T.Campbell,S.B.Eisenman, N.D. Lane, E.Miluzzo, R.A.Peterson***. « People Centric Urban Sensing ». In Proceedings of the 2nd International Workshop on Wireless Internet (WICON'06).Article No. 18. 2006

[170] ***A.T.Campbell, S.B.Eisenman,N.D.Lane,E.Miluzzo,R.A.Peterson,H.Lu,X.Zheng, M.Musolesi,K.Fodor,G.S.Ahn***. « The Rise of People-Centric Sensing ». *IEEE* InternetComp. *12,* 4,pp 12–21. 2008.

[171] ***T.Abdelzaher,Y.Anokwa,P.Boda,J.Burke,D.Estrin,L.Guibas,A.Kansal,S.Madden,J. Reich,***. « Mobiscopes For Human Spaces ». IEEE Pervas. Comp. 6, 2, pp20–29. 2007

[172] ***L. Shi, B. Zhang, H. T. Mouftah, J. Ma.*** « DDRP : An Efficient Data-Driven Routing Protocol for Wireless Sensor Networks with Mobile Sinks » Int. J. Commun. Syst., vol. 26, pp. 1341–1355. (Cité en page 92.) 2013

Références bibliographiques

[173] ***T. S. Chen, H. W. Tsai, Y. H. Chang,T. C. Chen***. « Geographic Convergecast Using Mobile Sink in Wireless Sensor Networks » Comput. Commun., vol. 36, pp. 445–458, (Cité en page 93.) 2013.

[174] ***B. Tang, J. Wang, X. Geng, Y. Zheng, J. U. Kim***. « A Novel Data Retrieving Mechanism in Wireless Sensor Networks with Path-Limited Mobile Sink ». Int. J. Grid Distrib. Comput., vol. 5, pp. 133–140, (Cité en page 93.) 2012.

[175] ***T. Banerjee, B. Xie, J. H. Jun, D. P. Agrawal***. « Increasing Lifetime of Wireless Sensor Networks Using Controllable Mobile Cluster-Heads ». Wirel. Commun. Mob. Comput., vol. 10, pp. 313–336. (Cité en page 93.) 2010

[176] ***C.Schurgers, V.Tsiatsis,S. Ganeriwal,M.B. Srivastava*** « Optimizing SensorNetworks in the Energy-Latency-Density Design Space ». *IEEE Trans. Mob. Comp. 1*, 1,pp 70–80. 2002

[177] ***G.Anastasi,M .Conti, E.Gregori,C. Spagoni,G. Valente.*** « Motes Sensor Networks in Dynamic Scenarios: An Experimental Study for Pervasive Applications in Urban Environments ». Int. J. Ubiq. Comp. Intell. 1. 2007.pp

[178] ***Y.Y. Lim,M.Messina, F.Kargl, L.Ganguli***. « SNMP-Proxy for Wireless Sensor Network ». Fifth International Conference on Information Technology :New Generations (ITNG 2008).Las Végas, Nevada USA.Source DBPL. 7-8 avril 2008

[179] ***Z.Yao,K.Gupta. «*** Backbone-Based Connectivity Control for Mobile Networks ». In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA'09).pp 2420–2426. 2009.

[180] ***Aditi ,Khadilkar , Nitin et Palan, «*** Media Access Control Protocol for MobileSensor Network- Modelling Using OMNeT++-MiXiM Network Simulator **».**(IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 2 (3), pp1154-1159, 2011

[181] MiXiM ,URL: htttp://www.MiXiM.mht.

[182] ***W.Wang, V.Srinivasan, K.C.Chua*** « Extending the Lifetime of Wireless Sensor Networks Through Mobile Relays ». IEEE/ACM Transactions on Networking,pp 1108–1120, 2008.

[183] ***Y.T.Hou, Y. Shi, H.D.Sherali, S.F.Midkiff***. « Prolonging Sensor Network Lifetime with Energy Provisioning and Relay Node Placement ». In: SECON 2005, pp. 295–304, 2005

[184] Omnet++, http ://www.omnetpp.org/.

[185] ***A. Boulis, Y.Tselishchev.***«Manual of Castalia

[186] ***S.Kumar,S.C.Sharma ,B.Suman***. «Classification and Evaluation of Mobility Metrics for Mobility Model Movement Patterns in Mobile Ad-Hoc Networks » International journal on applications of graph theory in wireless ad hoc networks and sensor networks(GRAPH-HOC) Vol.3, No.3. Indian Institute of Technology, Roorkee-India. 2011

[187] ***T.Camp, J.Boleng, V.Davies***.« A Survey of Mobility Models for Ad Hoc Network Research ». Dept. of Math. and Computer SciencesColorado School of Mines, Golden, CO. 10.Septembre 2002

[188] ***J.L.Huang ,M.S.Chen***. « On the Effect of Group Mobility to Data Replication in Ad-hoc Networks», IEEE Transactions on Mobile Computing, volume 5, pp 5, Mai 2006.

[189] ***F. Bai, A. Helmy***.«A Survey of Mobility Models in Wireless Ad-hoc Networks». , chapter book on Wireless Ad-Hoc and Sensor Networks, juin 2004.

[190] ***P.Bracka*** .« Une Architecture de Contrôle de Mobilité pour le Routage des Messagesdans les Réseaux Ad-hoc de Grande Taille » Thèse de doctorat. Université Marne la vallée. France

[191] ***L.Chen, W.B. Heinzelman***.«A Survey of Routing Protocols that Support QoS in

Références bibliographiques

Mobile Ad-Hoc Networks», IEEE Network, University de Rochester USA. November/December 2007.

[192] ***V.Lenders, J.Wagner,M.May***. «Analyzing the Impact of Mobility in Ad- hoc Networks»,In ACM/REALMAN, 2006.

[193] ***B.Dioum***. « Effets de la Mobilité sur les Protocoles de Routage dans les Réseaux Ad-hoc ».Université Mouloud Mammeri de Tizi Ouzou (Algerie) - Ingenieur d'état en Système d'information avancé 2007.

[194] ***Y.Chen, Z.Wang, T.Ren, Y.Liu, and H.Lv***. «Maximizing Lifetime of Wireless Sensor Networks with Mobile Sink Nodes ».College of Information Science & Technology, Zhejiang Shuren University, Hangzhou 310015, China. Published 30 Septembre 2014.

[195] ***S. Sharma, S. Kumar Jena***. « Data Dissemination Protocol for Mobile Sink in Wireless Sensor Networks ». Suraj Sharma1 and Sanjay Kumar Jena2 .1International Institute of Information Technology, Bhubaneswar 751003, India. 2 National Institute of Technology, Rourkela 769008, India . Published 27 Avril 2014.

[196] ***S.Basagni,A .Carosi, E.Melachrinoudis, C.Petrioli, Z.M.Wang***. « Controlled Sink Mobility for Prolonging Wireless Sensor Networks Lifetime ». Wireless Networks 14(6),pp 831–858 .2007.

[197] ***Tang, S., Yuan, J., Li, X., Liu, Y., Chen, G., Gu, M., Zhao, J., Dai, G.: DAWN. «*** Energy Efficient Data Aggregation in WSN with MobileSinks ».In:IEEE. InternationalWorkshoponQuality Of Service2010.

[198] ***L.Weifa, L.Jun***. « Network Lifetime Maximization in Sensor Networks with Multiple Mobile Sinks ». In: IEEE Conference on Local Computer Networks, LCN, 2011.

[199] ***P.Bakaraniya, S.Mehta***. « K-LEACH: An improved LEACH Protocol for Lifetime Improvement in WSN ».International Journal of Engineering Trends and Technology (IJETT) – Vol 4 Issue 5. ISSN: 2231-5381. Mai 2013. http://www.ijettjournal.org

[200] ***G. Renugadevi,M.G.Sumithra***. « An Analysis on LEACH-Mobile Protocol for Mobile Wireless Sensor Networks » .International Journal of Computer Applications (pp 0975 – 8887), Vol 65– No.21, Mars 2013.

[201] ***J.Charles.Boisson***. « Modélisation et Résolution par Méta-heuristiques Coopératives : de l'Atome à la Séquence Protéique ». Thèse de doctorat, Université des Sciences et Technologies de Lille. 2008

[202] ***H. Kwasnicka***. « Efficiency Of Selected Meta-Heuristics Applied To The TSP Problem: A Simulation Study », Department of Computer Science, Wroclaw University of Technology, Poland. September 2002.