Algorithmes de diffusion dans les réseaux dynamiques de capteurs sans fil

Tarek Moulahi

▶ **To cite this version:**

Tarek Moulahi. Algorithmes de diffusion dans les réseaux dynamiques de capteurs sans fil. Algorithme et structure de données [cs.DS]. Université de Franche-Comté; Université de Sfax (Tunisie), 2015. Français. NNT : 2015BESA2021 . tel-01508835

HAL Id: tel-01508835

https://theses.hal.science/tel-01508835

Submitted on 14 Apr 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Université de Franche-Comté
UFR des sciences et techniques

Ecole Doctorale Sciences Pour l'Ingénieur et Microtechniques

N° d'ordre: _____

Université de Sfax
Ecole Nationale d'Ingénieurs de Sfax

Ecole Doctorale Sciences et Technologies

N° d'ordre: _____

THESE Présentée par

Tarek MOULAHI

En vue de l'obtention du:

Grade de docteur de l'université de Franche-Comté.

Spécialité: **Informatique**

ET

Grade de docteur de l'université de Sfax.

Spécialité: **Ingénierie des Systèmes Informatiques**

Algorithmes de diffusion dans les réseaux de capteurs à topologie dynamique

Soutenue publiquement le 6 Mars 2015 devant le Jury composé de :

Pascal LORENZ	Professeur à l'université de Haute Alsace, France	Président
Zoubir MAMMERI	Professeur à l'université de Toulouse, France	Rapporteur
Lamia CHAARI EP FOURATI	M. C., HDR à l'université de Sfax, Tunisie	Rapporteur
Mohamed ABID	Professeur à l'université de Sfax, Tunisie	Examinateur
Hervé GUYENNET	Professeur à l'université de Franche-Comté, France	Directeur de thèse
Salem NASRI	Professeur à l'université de Monastir, Tunisie	Directeur de thèse

Remerciements

J'adresse mes remerciements á toutes les personnes qui m'ont permis d'accomplir cette thèse. En effet, en réalisant ce travail de recherche, je suis convaincu que la thèse est loin d'être un travail solitaire.

Je tiens tout d'abord á remercier M. Hervé GUYENNET, pour la confiance qu'il m'a accordée en acceptant d'encadrer ce travail doctoral, pour ses conseils précieux, son encadrement professionnel, sa disponibilité et ses qualités humaines.

Mes remerciements vont également á M. Salem NASRI, qui m'a dirigé aussi tout au long de ces trois années de thèse. Il a toujours été disponible, à l'écoute de mes questions. Je voudrais le remercier aussi pour ses multiples conseils, pour toutes les heures qu'il a consacrées à diriger cette recherche, ses qualités humaines et surtout sa patience.

Je voudrais également exprimer ma gratitude á Mme Lamia CHAARI FOURATI et M. Zoubir MAMMERI qui ont accepté d'être les rapporteurs de cette thèse, et je les en remercie, de même que pour leur participation au Jury. Ils ont également contribué par leurs consignes et suggestions à améliorer la qualité de ce manuscrit, et je leur en suis très reconnaissant.

Je remercie profondément M. Pascal Lorenz et M. Mohamed Abid qui m'ont fait l'honneur de participer au Jury de soutenance.

Je remercie également tous les membres de l'équipe CES Lab à Sfax et de l'équipe du DISC/FEMTO à Besançon et j'exprime mes gratitudes profondes à M. Hédi ben Messaoud et M. Mohamed Abid pour leurs aides administratives.

Et bien sûr je n'oublie jamais de remercier toutes les personnes qui m'aiment, toutes les personnes que j'aime, toute ma famille et tous mes enseignants.

Dédicaces

Je dédie ce travail à toute ma famille.

Sommaire

Remerciements 1

Dédicaces 2

Sommaire 3

Liste des figures 7

Liste des tableaux 10

Abréviations 11

Introduction générale 13

Première partie: 16

État de l'art 16

Chapitre 1 17

Les réseaux de capteurs et la diffusion 17

1.1. Introduction 17

1.2. RCSF: Rappel, importance et sujets ouverts de recherche 18

1.2.1. Définition des RCSF 18

1.2.2. Caractéristiques d'un capteur 18

1.2.3. Exemple de capteurs 20

1.2.4. Quelques exemples d'utilisation des RCSF 21

1.2.5. Environnements de simulation pour les RCSF 22

1.2.6. Comparaison des données centralisées à celles des RCSF 24

1.2.7. Classification des capteurs 25

Pour la classification des RCSF selon les données on distingue: 25

1.2.8. Abstraction des RCSF 29

1.2.9. Défis et sujets de recherche ouverts dans les RCSF 29

1.3. Importance de la tâche de diffusion dans les RCSF 30

1.3.1. Définition de la diffusion 30

1.3.2. Caractéristiques d'une méthode de diffusion 31

1.3.3. Découverte du voisinage 32

1.3.4. Rôle de la diffusion pour réaliser le routage 32

1.3.5. Distribution d'informations 33

1.3.6. Distribution des clés 34
1.3.8. Synchronisation du temps 34
1.4. Conclusion 35
Chapitre 2 37
Étude des méthodes de diffusion 37
Résumé: 37
2.1. Introduction 37
2.2. Présentation 38
2.2.1. Rappels et définitions. 38
2.2.2. Classification des méthodes de diffusion 40
2.2.3. Notion de backbone 40
2.3. Taxonomie des méthodes de diffusion 42
2.3.1. Diffusion basée sur le relai. 42
2.3.2. Diffusion basée sur les ensembles dominants connexes. 45
2.3.3. Diffusion basée sur des techniques hybrides. 47
2.3.4. Diffusion basée sur les ensembles dominants connexes à r sauts. 49
2.3.5. Diffusion basée sur les clusters. 51
2.3.6. Diffusion hiérarchique. 53
2.3.7. Diffusion basée sur les arbres. 54
2.4. Étude comparative 55
2.4.1. La minimisation 55
2.4.2. Tolérance aux fautes 56
2.4.3. Equilibrage de charge 57
2.5. Conclusion 58
Deuxième Partie: Contributions 59
Chapitre 3 60
Diffusion avec une couche physique réaliste 60
Résumé: 60
3.1. Introduction 60
3.2. Etude de la méthode DS-MPR 61
3.2.1. La méthode DS-MPR 61
3.2.2. Modèle LNM 62

3.2.3. Mesure de l'accessibilité de DS-MPR dans le modèle LNM 64
3.3. Contribution 1: RDS-MPR 66
3.3.1. Poids des nœuds dans RDS-MPR 66
3.3.2. Algorithme RDS-MPR 69
3.3.3. Comparaison de l'accessibilité de RDS-MPR avec celle de DS-MPR 70
3.4. Contribution 2: eRDS-MPR 71
3.4.1. Poids des nœuds dans eRDS-MPR 71
3.4.2. Algorithme eRDS-MPR 74
3.4.3. Comparaison de l'accessibilité d'eRDS-MPR à celle de RDS-MPR et DS-MPR 75
3.5. Etude comparative 76
3.5.1. Comparaison de l'énergie 76
3.5.2. Comparaison de performance 78
3.6. Conclusion 80
Chapitre 4 82
Optimisation de la diffusion multi-sauts 82
Résumé: 82
4.1. Introduction 82
4.2. Amélioration de la méthode MPR 83
4.2.1. modèle énergétique 83
4.2.2. Rappel et inconvénient de MPR 84
4.2.3. Amélioration proposée 86
4.2.4. Etude comparative 89
4.3. Amélioration de la méthode RDS-MPR 90
4.3.1. Rappel sur RDS-MPR 90
4.3.2. SCA : une nouvelle méthode d'auto-organisation 93
4.3.3. La méthode MCR 99
4.3.4. Etude comparative 100
4.4. Etude expérimentale avec las capteurs Arduino 103
4.4.1. Problème de la redondance circulaire 103
4.4.2. Le capteur Arduino 103
4.4.3. Etude comparative de MCR et 3B sur 3 capteurs Arduino. 104
4.5. Conclusion 105

Chapitre 5 ..... **107**
L'équilibrage de charge et la tolérance aux fautes pour la dissémination **107**
Résumé: 107
5.1. Introduction 107
5.2. Diffusion par la méthode LB(k,r)-CDS. 108
5.2.1. Définition du problème. 108
5.2.2. Algorithme LB(k,r)-CDS. 111
5.2.3. Evaluation du LB(k,r)-CDS. 114
5.3. Diffusion par relai tolérante aux fautes. 116
5.3.1. Indice de confiance. 116
5.3.2. Application de FCA pour effectuer la diffusion avec le relai. 117
5.3.3. Algorithme : diffusion par relai tolérant aux fautes avec FCA 118
5.3.4. Etude de performance 123
5.4. Conclusion 126
Conclusions et perspectives **127**
Publications **130**
Références **132**

Liste des figures

Figure 1.1. *Architecture d'un RCSF* 18
Figure 1.2. *Comparaison de la consommation d'énergie du MicaZ et TelosB* 20
Figure 1.3. *Classification des réseaux de capteurs selon les données* 1
Figure 1.4. *Capteur multimédia Cyclops* 26
Figure 1.5. *Classification des réseaux de capteurs selon la mobilité* 1
Figure 1.6. *Classification des réseaux de capteurs selon l'architecture d'organisation* 1
Figure 1.7. *Classification des réseaux de capteurs selon leur type* 1
Figure 1.8. *Classification des réseaux de capteurs selon les lieux de déploiement* 1
Figure 1.9. *(a) Capteur sous-marin LinkQuest et (b) le robot volant Swarm* 28
Figure 1.10. *Abstraction d'un RCS* 29
Figure 2.1. *L'ensemble {b,c,f,h,i} est un CDS du graphe* 1
Figure 2.2. *L'ensemble {b,c,e} est un (2,2-saut)-CDS du graphe* 1
Figure 2.3. *Classification des techniques de diffusion dans les RCSF* 1
Figure 2.4. *Exemple d'application de la méthode MPR* 1
Figure 2.5. *CDS dans un RCSF* 1
Figure 2.6. *Communication à travers un CDS de A vers B* 1
Figure 2.7. *Exemple d'un CDS (à droite) et d'un ensemble 2-saut CDS (à gauche) dans un RCSF* 1
Figure 2.8. *Organisation d'un RCSF (à gauche) en clusters (à droite)* 1
Figure 2.9. *Concept de minimisation dans les RCSF* 1
Figure 3.1. *Déploiement de 300 nœuds dans une zone de 1km^2* 61
Figure 3.2. *Principe de la diffusion basée sur le relai* 1
Figure 3.3. *Comparaison de probabilité de réception entre LNM et UDM pour différents rayons de transmission* 63
Figure 3.4. *La distance moyenne entre les nœuds selon leur rayon de communication* 65
Figure 3.5. *Comparaison de l'accessibilité de la méthode DS-MPR dans les deux modèles LNM & UDM* 65
Figure 3.6. *Comparaison de l'accessibilité de la méthode DS-MPR et RDS-MPR dans le modèle LNM*.. 70
Figure 3.7. *Choix des nœuds de relai basé sur les informations a 3 niveaux* 71
Figure 3.8. *Comparaison de l'accessibilité des méthodes DS-MPR, RDS-MPR et eRDS-MPR dans le modèle LNM* 75
Figure 3.9. *Comparaison d'énergie totale consommée pour accomplir la diffusion des méthodes DS-MPR, RDS-MPR et eRDS-MPR.* 77

Figure 3.10. *Messages échangés* 78
Figure 3.11. *Temps de communication* 79
Figure 3.12. *Espace mémoire* 80
Figure 4.1. *Comparaison entre les énergies de réception et d'émission* 84
Figure 4.2. (a) *Inondation aveugle*, (b) *Inondation MPR* 85
Figure 4.3. *Energie de diffusion consommée dans les réseaux de la figure.4.2 avec et sans les liens en bleu* 85
Figure 4.4. *Organisation du réseau sous forme de cluster* 88
Figure 4.5. *Comparaison de l'énergie totale de diffusion pour un réseau de 150 nœuds* 89
Figure 4.6. *Comparaison de l'énergie totale de diffusion pour un réseau de 250 nœuds* 90
Figure 4.7. *Choix des nœuds de relai par RDS-MPR* 91
Figure 4.8. *Energie totale consommée de diffusion dans le réseau de la figure.4.7 en appliquant RDS-MPR avec et sans liens intra-voisinage* 92
Figure 4.9. *Différents cas d'identification d'un nouveau nœud* 95
Figure 4.10. *Application de SCA à un réseau: les liens intra-niveaux du réseau d'origine sont éliminés mais les liens d'un niveau à l'autre dans le même cluster sont conservés.* 97
Figure 4.11. *Energie totale consommée avec RDS-MRR pour accomplir la diffusion avec des densités différentes et des tailles différentes de paquets* 102
Figure 4.12. *Energie totale consommée avec MCR pour accomplir la diffusion avec des densités différentes et des tailles différentes de paquets* 102
Figure 4.13. *Redondance circulaire* 1
Figure 4.14. *Platforme d'expérience* 104
Figure 4.15. *Identifient des nœuds: à droite si on applique MCR et à gauche avec 3B* 1
Figure 4.16. *Résultats expérimentaux* 105
Figure 5.1. *Exemple d'un graphe représentant un RCSF. L'ensemble (1,2-saut)-CDS est {c,f}.* 1
Figure 5.2. *Exemple d'un graphe représentant un RCSF. L'ensemble (1,2-saut)-CDS est {c,d,e}.* 1
Figure 5.3. *Affectation des nœuds dominés à leur dominant approprié : 1ère possibilité* 1
Figure 5.4. *Affectation des nœuds dominés à leur dominant approprié: 2ème possibilité* 1
Figure 5.5. *Evolution de l'énergie dans le nœud qui a fait plus de communication dans (1,2)-CDS avec une densité égale à 12 et une taille de paquet égale à 200 octets* 114
Figure 5.6. *Evolution de l'énergie dans le nœud qui a fait plus de communication dans (1,2)-CDS avec une densité égale à 200 et une taille de paquet égale à 200 octets* 115
Figure 5.7. *Evolution de l'énergie dans le nœud qui a fait plus de communication dans (1,2)-CDS avec une densité égale à 200 et une taille de paquet égale à 400 octets* 115

Figure 5.8. *Fonction d'indice de confiance entre Max TI=1 et Min TI=0 lorsque v(u)>=0* 117
Figure 5.9. *Exemple d'un RCSF* 1
Figure 5.10. *Treillis Relai* 1
Figure 5.11. *Taux de réussite de RRB comparé à RDS-MPR pour un taux de défaillance de nœuds qui varie entre 10 et 50% du nombre total des nœuds-capteur* 124
Figure 5.12. *Taux de réussite de RRB comparé à RDS-MPR pour un taux de défaillance fixé à 20% et un nombre de nœuds qui varie de 50 à 300* 125

Liste des tableaux

Tableau 1.1. *Comparaison de consommation d'énergie du MicaZ et TelosB* 19
Tableau 1.2. *Exemple de nœuds capteurs.* 20
Tableau 1.3. *Comparaison des données traditionnelles à celles des RCSF.* 24
Tableau 2.1. *Elément du backbone selon la méthode de diffusion* 41
Tableau 2.2. *Différentes types de liaison entre les têtes de clusters* 52
Tableau 3.1. *Choix du nombre de nœuds selon les distances entre eux* 64
Tableau 3.2. *Symboles et descriptions* 66
Tableau 3.3. *Paramètres de simulation pour les trois méthodes* 76
Tableau 3.4. *Notations de comparaisons* 78
Tableau 4.1. *Liste des symboles pour la méthode de construction des groupes* 86
Tableau 4.2. *Nombre total de réceptions* 91
Tableau 4.3. *liste des symboles dans SCA* 93
Tableau 4.4. *Notations pour la complexité* 100
Tableau 5.1. *liste des symboles* 109
Tableau 5.2. *Contexte Relai* 120
Tableau 5.3. *Choix des nœuds de relai* 121
Tableau 5.4. *Symboles et notations* 121
Tableau 5.5. *Paramètres de simulation* 123

Abréviations

RCSF: Réseau de Capteurs Sans Fil.

WSNs: Wireless Sensors Networks.

MWSNs: Mobile WSNs

MANET: Mobile Ad Hoc Network

VANET: Vehicular Ad Hoc Network

MPR: Multi Point Relay

DS-MPR: connected Dominating Set with MPR

RDS-MPR: Realistic DS-MPR

eRDS-MPR: extended RDS-MPR

CH: Cluster head

CM: Cluster Member

G: Gateway

SCA: Self Clustering Algorithm

MCR: Multi Cluster Relay

3B: Backbone-Based Broadcasting

CC: Clusters Construction

CDS: Connected Dominating Set

(k,m)-CDS: k-Connected m-Dominating Set

(k,r)-CDS: k-Connected r-hop Dominating Set

LB(k,r)-CDS: Load-Balanced (k,r)-CDS

RRB: Reliable Relay-based Broadcasting

TI: Trust Index

FCA: Formal Concept Analysis

AODV: Ad hoc On-Demand Distance Vector

LEACH: Low Energy Adaptive Clustering Hierarchy

DSR: Dynamic Source Routing

Introduction générale

Les réseaux de capteurs sont de plus en plus utilisés dans une variété de domaines: scientifiques, militaires, industriels, de santé, d'environnement, etc. Le besoin d'observer, d'analyser et de contrôler des phénomènes physiques sur des zones étendues est devenu essentiel pour de nombreuses applications.

D'après une étude du cabinet IDTechEx de 2012 [17,36] , le marché mondial des réseaux de capteurs sans fil devrait croitre de 0,45 milliard de $ à 2 milliards de $ entre 2012 et 2022. Cette croissance peut s'expliquer par la baisse de prix des composants avec une augmentation étendue de leurs performances et par l'augmentation du nombre d'applications de plus en plus variées. L'étude souligne le manque de solutions globales intégrées puisque pour chaque produit conçu il y a une nécessité de développements logiciels et matériels.

Par ailleurs, grâce aux récents développements réalisés dans le domaine de la micro-électronique et des technologies de communication, de nouvelles possibilités ont émergé poussant encore plus loin les limites d'utilisation des capteurs sans fil.

La gestion de l'énergie dans les réseaux de capteurs et la recherche de protocole de communication performants constituent aujourd'hui les préoccupations essentielles des chercheurs et les principaux enjeux des industriels.

Ces capteurs non-coûteux en fabrication et de petite taille fonctionnent avec une faible énergie et communiquent sans fil avec d'autres dispositifs dans leurs zones de communication. La fonction essentielle d'un capteur est de mesurer des grandeurs physiques dans l'environnement qui l'entoure telles que la température, le taux d'humidité, les vibrations, etc. Ensuite il les convertit en grandeurs numériques pouvant être traitées par des systèmes informatiques.

Un RCSF est un ensemble de capteurs (nœuds) contrôlés par un nœud principal appelé nœud de base ou puits. Ces capteurs coopèrent ensemble pour réaliser des tâches d'observation et de capture ensuite le transfert des données vers le nœud de base qui constitue la passerelle vers le reste du monde numérique.

Compte tenu de la diversité des applications, le déploiement d'un ensemble de capteurs est souvent aléatoire. Il peut aussi concerner des zones dangereuses non-contrôlées ou exigeant la mobilité. Ceci se traduit par une topologie dynamique, non-sécurisée et indéterminée en espace et en temps. Cela constitue aussi un défi pour réaliser les communications entre les capteurs, essentiellement les tâches de routage et de diffusion.

La diffusion dans les RCSF peut être définie comme étant la dissémination d'un paquet de données dans le réseau:

- Du nœud de base vers tous les nœuds,
- Du nœud de base vers un sous-ensemble de tous les nœuds,
- D'un nœud vers son voisinage.

Cette tâche est une opération essentielle et primordiale dans les RCSF. Elle est utilisée pour la découverte du voisinage ou de la totalité des nœuds du réseau. Elle permet aussi la distribution et la dissémination des informations dans le réseau comme les clés de sécurité. En outre, la diffusion est utilisée pour accomplir le routage et la localisation des nœuds. La diffusion permet aussi la synchronisation des horloges entre les différentes entités du réseau.

C'est dans ce cadre que se situent les travaux présentés dans ce manuscrit. On propose d'étudier la diffusion dans un RCSF en tenant compte de l'aspect dynamique du réseau et les caractéristiques des capteurs essentiellement la faible énergie, la puissance de calcul limitée, l'espace de stockage réduit et la vulnérabilité aux fautes de ces capteurs. Il en découle qu'une politique d'économie dans la consommation d'énergie est nécessaire pour prolonger la durée de vie du réseau. Pour cela, des méthodes d'optimisation des flux de communication sont proposées. Par ailleurs, il est aussi important de tenir compte de l'aspect évolutif dans le temps et l'espace d'un capteur pour garantir un transfert fiable des données. Toutefois, on considère que la diffusion basée sur le relai multi-saut et les approches d'auto-organisation des RCSF permettent d'offrir une économie de communication et de distribution des tâches.

Ce manuscrit est reparti en cinq chapitres définis comme suit:

Le chapitre 1 traite la diffusion comme étant une tâche primordiale dans les RCSF. Dans ce chapitre, on présente un rappel sur les RCSF, leurs fonctionnalités, leurs

caractéristiques et leur taxonomie selon différents points de vue. Ensuite, on définit la tâche de diffusion, sa position et son rôle dans ce type de réseau.

Dans le chapitre 2, on étudie et on classe les principales méthodes de diffusion existantes dans les RCSF selon les techniques utilisées pour résoudre le problème de dissémination. Ensuite, on réalise une étude comparative pour identifier les manques et les inconvénients de ces techniques afin de définir les orientations de notre travail de recherche.

Le chapitre 3 concerne notre première contribution sur la diffusion dans un environnement réel en tenant compte de l'environnement de la communication qui a une grande influence sur le taux d'accessibilité des méthodes de diffusion. Dans ce chapitre, on étudie une méthode de diffusion conçue dans un environnement idéal. Ensuite, on modifie cette méthode pour la rendre adéquate avec une couche physique réaliste décrite par un modèle probabiliste.

Dans le chapitre 4 on présente notre deuxième contribution. En effet, l'utilisation de la diffusion basée sur le multi-saut est la source d'une perte importante d'énergie des capteurs due aux émissions et réceptions redondantes. De plus ces méthodes ne tiennent pas compte de l'aspect dynamique du réseau. Dans ce chapitre, on utilise des méthodes d'auto-organisation dynamiques pour isoler les nœuds dans des clusters avant de réaliser la diffusion en évitant la redondance.

Enfin, le chapitre 5 décrit nos contributions pour:

- La sélection d'un sous-ensemble de capteurs pour faire la diffusion avec équilibrage de charge.
- La diffusion par relai, tolérante aux fautes en utilisant des nœuds avec des indices de confiance élevés pour qu'ils fassent partie d'un sous-ensemble du réseau afin de réaliser la diffusion.

On aboutit à la fin de ce manuscrit à une conclusion du travail de recherche présenté, essentiellement l'importance des contributions accomplies. De même, on présente comme perspectives l'intérêt de nos travaux envers les RCSF et les réseaux Ad Hoc en général.

Première partie:

État de l’art

Chapitre 1

Les réseaux de capteurs et la diffusion

1.1. Introduction

Au cours de la dernière décennie, les RCSF ont été un sujet de recherche très actif au niveau de la communauté scientifique à cause de leur coût raisonnable comparé à leur importante capacité d'observation. Le but est de transférer les informations collectées vers le monde numérique d'une façon coopérative entre les capteurs et avec une technologie sans fil.

Les capteurs possèdent une faible puissance de calcul et une mémoire limitée. La majorité de ceux-ci sont alimentés par des batteries avec une durée de vie restreinte. Dans beaucoup de cas le déploiement des capteurs se réalise dans un environnement n'offrant pas la possibilité de changer les batteries en cas d'épuisement. Ces caractéristiques constituent l'inconvénient majeur des RCSF.

Fréquemment, le déploiement des capteurs se fait aléatoirement dans les zones à contrôler ce qui constitue un défi pour réaliser la collecte des informations et la communication entre les différentes entités.

La diffusion est l'une des opérations fondamentales dans les RCSF. Elle permet:

- la découverte de voisinage,
- la réalisation du routage,
- la distribution d'informations dans tout le réseau (par exemple la clé de sécurité),
- la localisation des nœuds,
- la synchronisation du temps.

Ce chapitre est divisé en deux sections, la première concerne un rappel sur les RCSF. Dans la deuxième section, on s'intéresse à la diffusion en montrant son importance puis son implémentation dans les RCSF.

1.2. RCSF: Rappel, importance et sujets ouverts de recherche

Dans cette section, on présente un rappel et une classification des RCSF et on donne leurs caractéristiques, leurs types et leurs inconvénients constituant la base des travaux de recherche dans ce domaine.

1.2.1. Définition des RCSF

Les RCSF sont le résultat de la convergence des techniques de communication sans fil, des systèmes de technologie micro-électromécaniques, et de l'électronique numérique [3].

Un RCSF est un ensemble de capteurs collaborant ensemble sans fil afin de transmettre une information du monde physique vers le monde numérique. Dans un RCSF les capteurs sont considérés comme des nœuds contrôlés par un ou plusieurs nœuds de base appelés puits. Les puits sont les lieux de regroupement des données collectées et la passerelle entre les nœuds capteurs et le monde numérique [58]. Ci-après, on donne dans la figure 1.1, l'architecture de ce réseau.

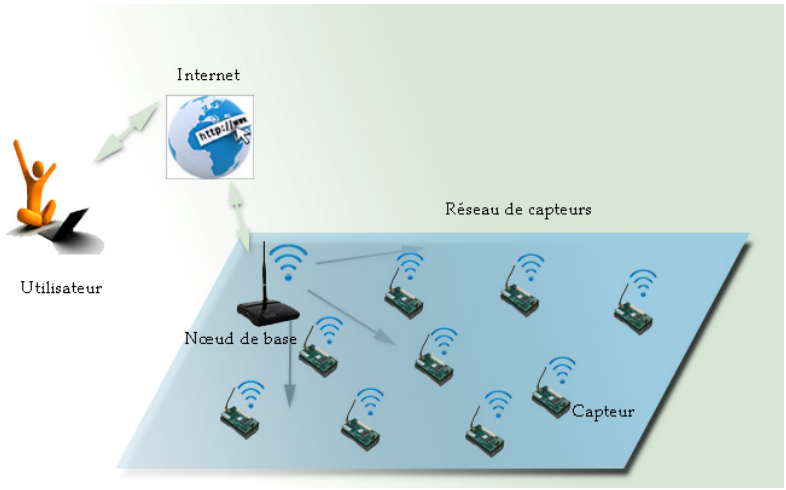


Figure 1.1 *Architecture__ d'un RCSF*

1.2.2. Caractéristiques d'un capteur

Un capteur est un dispositif intelligent caractérisé par sa petite taille, son coût raisonnable, son énergie et sa capacité de traitement et de stockage limitée. Il communique sans fil sur une distance courte. Il est appelé intelligent (*smart sensor*) parce qu'il est autonome et qu'il possède

un processeur. Il est capable de communiquer avec d'autres grâce à son système de communication radio.

Un capteur est capable de réaliser la capture des grandeurs physiques, le stockage ou le calcul sur les informations capturées et la communication sans fil. Un capteur peut traiter des données scalaires, exemple: température, humidité, mouvement, pression ou multimédia: image, son, vidéo.

La faiblesse de la capacité de stockage, de traitement et d'énergie constituent les inconvénients d'un capteur. Pour cela, la réalisation des tâches d'un capteur (collecte des données, routage d'information, diffusion, …) doit tenir compte de ces contraintes en conservant l'énergie au maximum afin d'augmenter sa durée de vie pour maximiser la longévité du réseau entier. Ci-après, on donne dans le tableau 1.1 une comparaison de consommation d'énergie [14] entre deux capteurs bien connus, MicaZ [31] et TelosB [32]. La communication est la source la plus importante de consommation d'énergie par rapport au calcul. Ainsi la majorité des travaux concernent la réduction de cette consommation d'énergie. Dans la figure 1.2, cette constatation est confirmée.

Tableau 1.1. *Comparaison de consommation d'énergie du MicaZ et TelosB*

Consommation d'énergie	MicaZ	TelosB
Calcul interne	26 mW (@ 7.37 MHZ)	4.8 mW (@ 4MHZ)
En veille	0.025 mW	0.035 mW
Communication= Emission + Réception	137 mW (@ Ptx= -5dBm)	115 mW (@ Ptx= -5dBm)

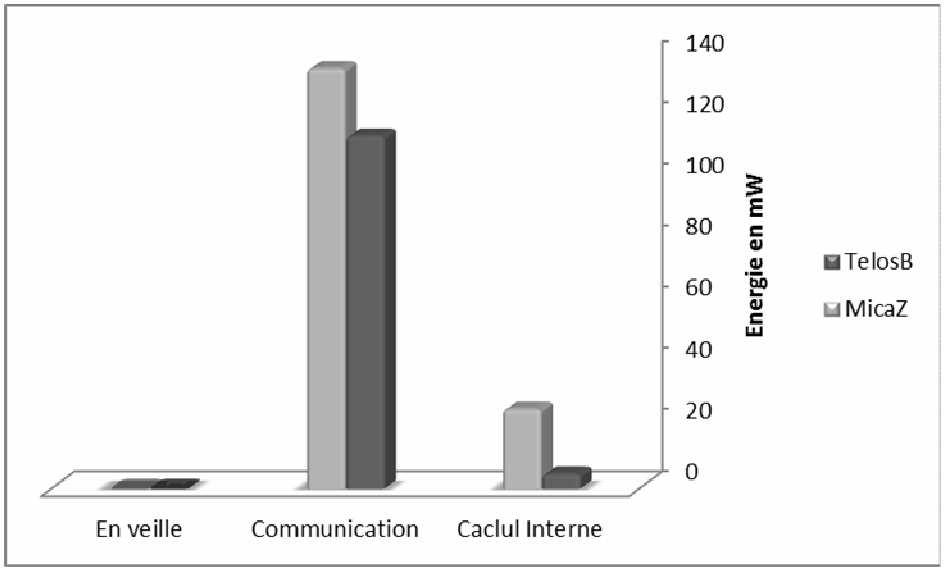


Figure 1.2. *Comparaison de la consommation d'énergie du MicaZ et TelosB*

1.2.3. Exemple de capteurs

Dans le paragraphe précédent, on a cité deux capteurs très connus, cependant il existe d'autres exemples. Chaque capteur a plusieurs caractéristiques comme la portée, la capacité de stockage et de calcul en plus du modèle énergétique, le système d'exploitation et le langage de programmation supporté. Dans le tableau 1.2, on donne quelques exemples de capteurs utilisés dans la littérature avec leurs caractéristiques [40].

Tableau 1.2. *Exemple de nœuds capteurs.*

Nom	*Mémoire de données*	*Mémoire externe*	*Programmation*	*Système d'exploitation*
Mica2	128+4 KO RAM	128 KO flash	nesC	TinyOS
Mica2dot	4 KO RAM	128 KO flash	nesC	TinyOS
MicaZ	4 KO RAM	128 KO flash	nesC	TinyOS, SOS et Mantis OS
Iris mote	8 KO RAM	128 KO flash	nesC	MoteRunner, TinyOS, MoteWorks

SunSPOT	512 KO RAM	4 MO flash	Java	Squawk Java ME Virtual Machine
TelosB	10 KO RAM	48 KO flash	nesC	Contiki, TinyOS, SOS, Mantis OS
TmoteSky	10 KO RAM	48 KO flash	nesC	Contiki, TinyOS, SOS, Mantis OS

1.2.4. Quelques exemples d'utilisation des RCSF

Les RCSF sont exploités dans beaucoup de domaines [3,58]:

- Militaire: par exemple pour contrôler les forces alliées ou pour surveiller les champs de bataille afin de connaître le déplacement de l'ennemi ou pour réaliser le ciblage.
- La surveillance d'environnement: le déploiement des capteurs dans des forets permet la détection des incendies ou la détection des fuites de gaz toxiques afin de protéger la faune et la flore.
- La médecine: les RCSF permettent la transmission des données vitales des patients en permanence aux médecins, ce qui rend plus facile la tâche de suivi en continu de certains patients, surtout les patients ayant une faible liberté de mouvement.
- Dans la domotique: par exemple la détection de vol de maison.
- Transport: surveillance des autoroutes.

Parmi les grands projets où les RCSF sont exploités on peut citer:

- SensorScope [33]: Sensor Networks for Environmental Monitoring, est un projet de surveillance de la nature. C'est un système distribué de mesure de l'environnement à grande échelle centré sur un RCSF pour produire des mesures de densité spatiale et temporelle. Ce système innovant est composé de stations de détection solaires multiples qui communiquent sans fil. SensorScope fournit un faible coût des systèmes de réseaux de capteurs sans fil fiables pour le suivi de l'environnement à une large communauté. Il améliore les techniques de collecte de données avec les technologies récentes, en répondant aux exigences des spécialistes de l'environnement.

- Libelium [34]: est un projet multidisciplinaire offrant plusieurs applications de RCSF dans la vie quotidienne.
- Citons le projet intelligent de contrôle de l'eau à Valencia en Espagne, utilisation du système EkoBus dans les villes de Belgrade et Pancevo en Serbie où les capteurs ont été installés sur les véhicules de transport public afin de suivre un ensemble de paramètres environnementaux sur une grande distance. Ils peuvent fournir des informations supplémentaires pour l'utilisateur comme l'emplacement des bus et l'heure d'arrivée approximative dans les arrêts de bus. On peut également citer l'exemple de contrôle des projets d'agriculture dans la ville du Galicia, etc.

1.2.5. Environnements de simulation pour les RCSF

La simulation des réseaux de capteurs est importante. Elle offre un environnement pour le développement de nouvelles solutions. De nouvelles architectures peuvent être évaluées sur une grande échelle à faible coût. Les simulateurs permettent aux utilisateurs d'isoler différents facteurs en réglant les paramètres configurables [62].

Un simulateur permet de tester les techniques et les protocoles conçus pour les RCSF tandis qu'un émulateur permet de combiner l'implémentation logicielle et matérielle ce qui permet d'offrir une précision plus importante.

Parmi les simulateurs et les émulateurs les plus utilisés on peut citer:

- NS-2: Network Simulator version 2, outil de simulation open source. La première version de ce simulateur est apparue en 1989. NS-2 permet la simulation de TCP, des protocoles de routage et de multicast des réseaux filaires et sans fil. Il offre aussi un modèle énergétique pour simuler les consommations d'énergie dans les nœuds. Son inconvénient majeur est la difficulté d'ajout de nouveaux protocoles. NS-2 est écrit en C++ est une version orientée objet de Tcl et appelée OTcl. On peut citer aussi NS-3 qui est la version 3 du NS.
- TOSSIM: est un émulateur conçu essentiellement pour les RCSF et fonctionnant sur TinyOS. TOSSIM peut être utilisé sur Cygwin (environnent Windows) ou sur environnement Linux. Cet émulateur possède une interface graphique TinyViz ce qui

facilite l'interaction et la réalisation des scénarios de simulation. TOSSIM est conçu pour simuler seulement les applications de TinyOS mais ne fournit pas la possibilité de faire les mesures de performance de nouveaux protocoles.

- EmStar: est un émulateur destiné essentiellement aux RCSF. Il est implémenté en C et a été développé à l'université de Californie. Parmi les avantages de cet émulateur c'est qu'il permet aux utilisateurs d'exécuter chaque module séparément sans sacrifier la réutilisation du logiciel en plus de son interface graphique qui facilite son utilisation. Pour les inconvénients, il ne peut pas supporter un grand nombre de nœuds et il possède une évolutivité limitée ce qui réduit la réalité de simulation. En outre, cet émulateur ne peut s'appliquer qu'aux capteurs iPAQ-class et les capteurs MICA2.
- OMNet++: est un simulateur de réseaux à événements discrets implémenté en C++. Il peut être utilisé soit avec un environnement Windows ou Linux. Il offre la possibilité de simuler les réseaux filaires et sans fil. Parmi ses avantages: l'interaction avec une interface graphique puissante, il offre la possibilité de simuler la mobilité et les protocoles MAC et les protocoles de localisation des RCSF. L'inconvénient majeur de ce simulateur est l'insuffisance du nombre de protocoles disponibles.
- J-Sim: est un simulateur de réseau à événements discrets implémenté en Java. Parmi ses avantages: il offre la réutilisabilité et l'interchangeabilité ce qui facilite la tâche de simulation. En plus, J-Sim contient un grand nombre de protocoles. Ce simulateur peut également permette des diffusions de données, des itinéraires et des simulations de localisation dans les RCSF. Il peut aussi simuler les communications et la consommation d'énergie. Parmi les inconvénients, ce simulateur est un peu lent comparé à NS-2 par exemple.
- ATEMU: est un émulateur pour les RCSF implémenté en C basé sur le microprocesseur AVR qui est un microcontrôleur à puce utilisé également par les plateformes MICA. Parmi ses avantages, il offre la possibilité de simuler de multiples capteurs en même temps où chaque capteur peut exécuter son propre programme. ATEMU permet aussi de simuler les RCSF homogènes et hétérogènes et différentes applications exécutées sur les capteurs MICA. Parmi les inconvénients, cet émulateur est un peu lent comparé à

d'autres, de plus il possède peu de fonction pour simuler les scenarios de *clustering* ou les problèmes de routage.

- Avrora: est un émulateur conçu pour les RCSF et implémenté en Java. Il est similaire à ATEMU c.-à-d. à base du microprocesseur AVR donc il peut émuler également les capteurs MICA2. Avrora a été développé à l'université de Californie. Il permet de combiner les avantages de TOSSIM et de ATEMU en limitant leurs inconvenants. Par contre il ne possède pas d'interface graphique, et il ne peut pas émuler les algorithmes de gestion de réseau, car il ne possède pas d'outils de communication.

1.2.6. Comparaison des données centralisées à celles des RCSF

Dans les RCSF, le traitement et la gestion des données se fait d'une façon distribuée et décentralisée contrairement au traitement traditionnel des données. Dans le tableau 1.3, on donne une comparaison entre les données traditionnelles et celles des RCSF [42].

Tableau 1.3. *Comparaison des données traditionnelles à celles des RCSF.*

	Données traditionnelles	Données dans les RCSF
Architecture de traitement	Centralisée	Distribuée
Type de données	Statique	Dynamique
Utilisation de la mémoire	Non-limitée	Limitée
Temps de traitement	Non-limité	Limité
Puissance de traitement	Elevée	Faible
Energie	Pas de contraintes	Limitée
Flux des données	Stationnaire	Continue

Le tableau précédent montre la différence entre le traitement des données traditionnelles et leurs caractéristiques par rapport aux données traitées dans les RCSF ce qui donne une vue plus précise des RCSF, de leurs caractéristiques spécifiques et de leurs limites.

1.2.7. Classification des capteurs

Les RCSF peuvent être classés en plusieurs catégories:

- Selon les données qu'ils gèrent: scalaires ou multimédia.
- La mobilité: rapide, moyenne, statique.
- L'architecture d'organisation: organisés selon une architecture ou non.
- Les types de capteurs: hétérogènes ou homogènes.
- Le lieu où ils sont déployés: terre, sous-marin, sous-terrain, espace.

Les figures 1.3, 1.5, 1.6, 1.7 et 1.8 donnent respectivement la classification des réseaux de capteurs selon: le type de données traitées, la mobilité, l'architecture d'organisation, le type de capteurs et le lieu où il sont déployés.

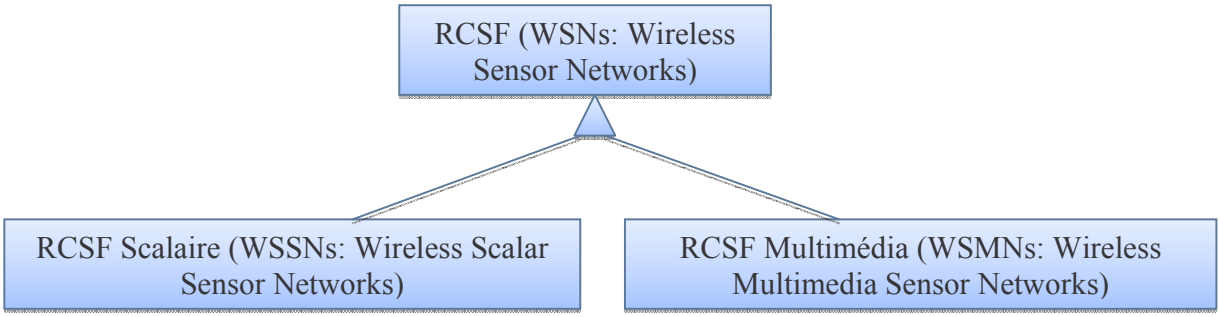


Figure 1.3. *Classification des réseaux de capteurs selon les données*

Pour la classification des RCSF selon les données on distingue:

- Les RCSF scalaires traitant des données scalaires simples tels que les capteurs de température ou d'humidité ou de pression.
- Les RCSF multimédia dont un exemple est donné dans la figure 1.4, traitant des données multimédia tels que les capteurs d'image, du son et du vidéo ce type peut aussi contenir des capteurs scalaires.

Figure 1.4. *Capteur multimédia Cyclops*

Concernant la classification des RCSF selon la mobilité on distingue:

- Les RCSF statiques (non mobiles): topologie fixe.
- Les RCSF mobiles: topologie variable.
- Les RCSF véhiculaires à mobilité plus rapide par rapport aux types précédents: la topologie change rapidement.

Toutefois, les réseaux de capteurs dynamiques peuvent être non mobiles, dans ce cas-là l'aspect dynamique est fonction de temps c.-à-d. le réseau change de topologie suite à l'absence d'un capteur pour de multiples raisons ou la présence d'un nouveau capteur ajouté par l'utilisateur. Les réseaux de capteurs dynamiques peuvent être aussi mobiles dans ce cas, l'aspect dynamique est fonction des changements spatio-temporels.

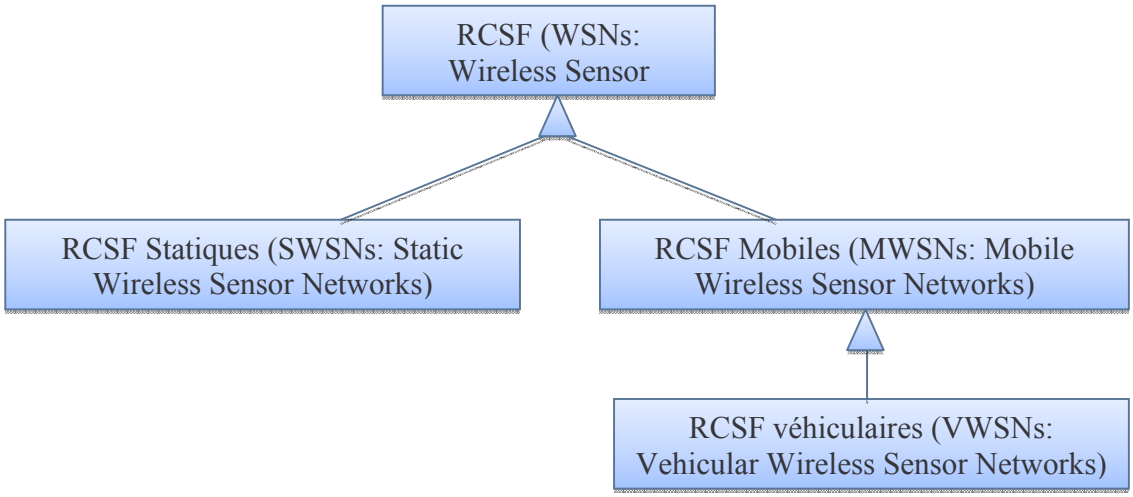


Figure 1.5. *Classification des réseaux de capteurs selon la mobilité*

Pour la classification des RCSF selon l'architecture d'organisation on distingue:

- Les RCSF non organisés c.-à-d. la formation du réseau après le déploiement des capteurs.
- Les RCSF organisés, où l'on distingue l'organisation hiérarchique (selon le type ou la fonction du capteur) et l'organisation en clusters. Pour ce type dernier, il y a plusieurs sortes d'organisation qu'on détaillera dans le chapitre suivant.

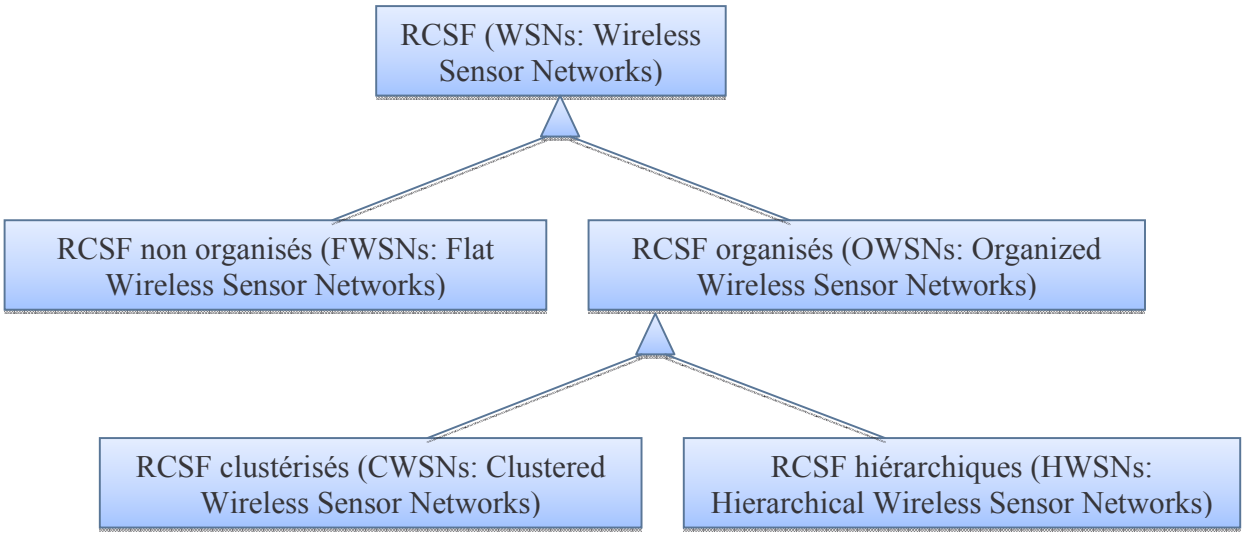


Figure 1.6. *Classification des réseaux de capteurs selon l'architecture d'organisation*

- Homogène: architecture un-tiers, tous les capteurs sont identiques c.-à-d. ont les mêmes caractéristiques matérielles
- Hétérogène: architecture multi-tiers, il y a plusieurs types de capteurs qui diffèrent selon leurs caractéristiques ou fonctionnalités.

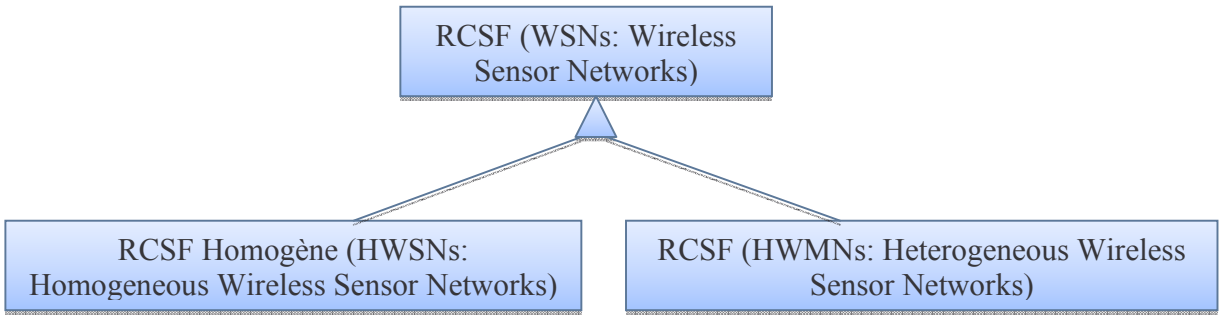


Figure 1.7. *Classification des réseaux de capteurs selon le type de capteur*

Finalement, les RCSF peuvent être classés selon les lieux où ils sont déployés voir figure 1.8. La figure 1.9 représente deux types de capteurs: sous-marin et aérien:

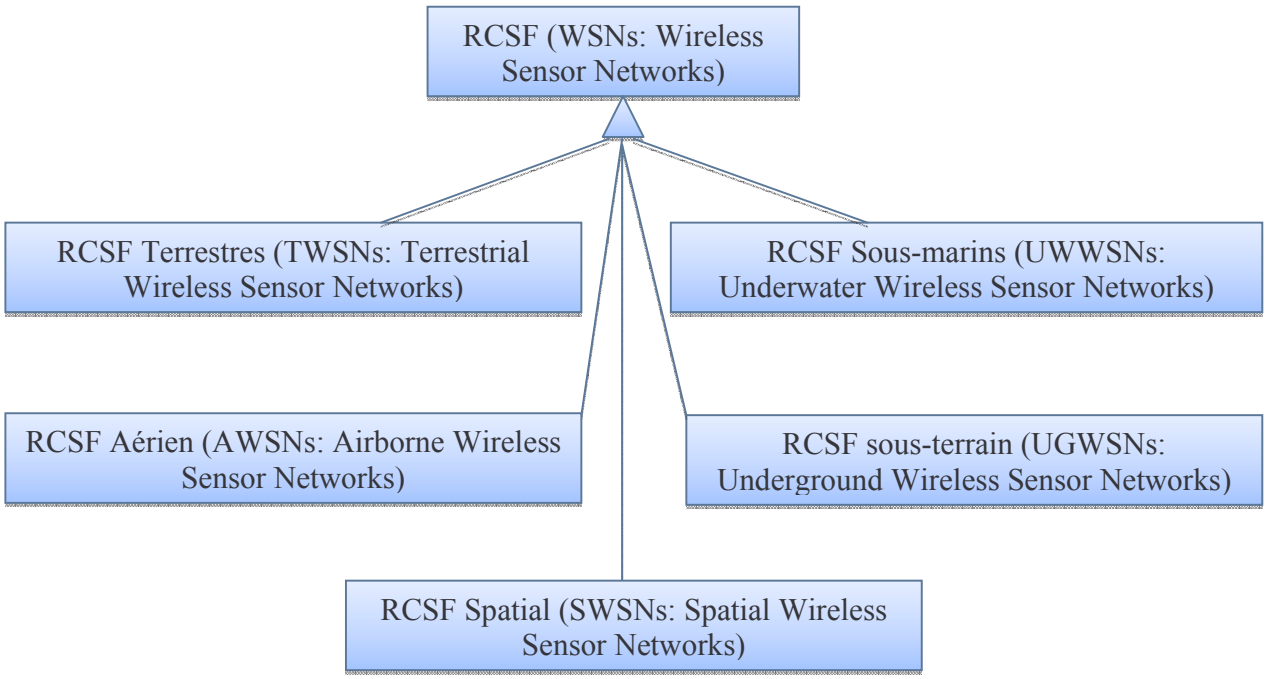


Figure 1.8. *Classification des réseaux de capteurs selon les lieux de déploiement*

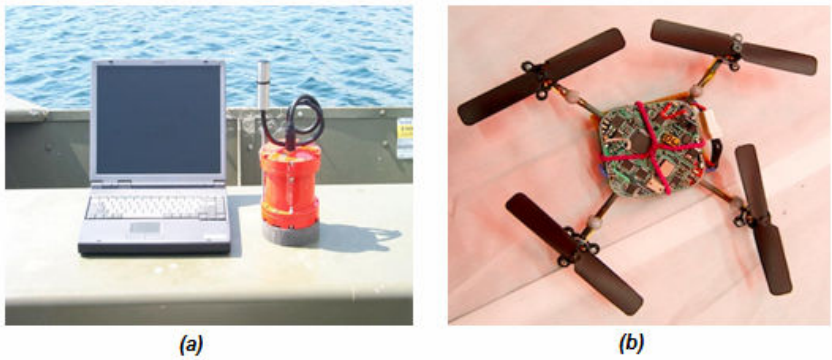


Figure 1.9. *(a) Capteur sous-marin LinkQuest et (b) le rebot volant Swarm*

1.2.8. Abstraction des RCSF

Les RCSF peuvent être représentés par un graphe non orienté *G(V,E)* où *V* représente l'ensemble des nœuds et *E* représente l'ensemble des arrêtes entre les nœuds *V*. Une arrête existe entre deux nœuds A et B si et seulement si B est dans la zone de couverture de A et vice-versa, la figure 1.10 représente cette abstraction.

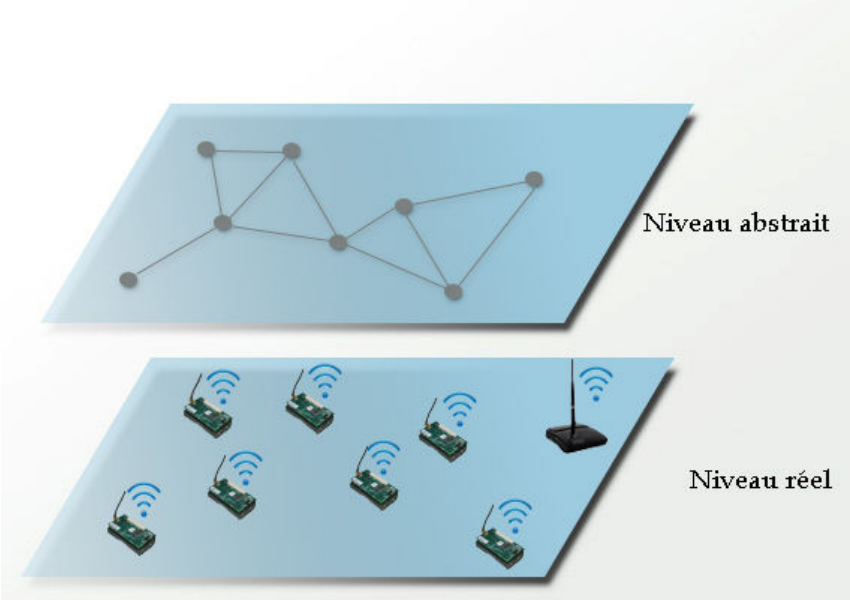


Figure 1.10. *Abstraction d'un RCSF*

1.2.9. Défis et sujets de recherche ouverts dans les RCSF

Malgré le grand apport de la communauté scientifique aux RCSF, il reste plusieurs thèmes à élaborer.

- Techniques de minimisation de la consommation d'énergie pour augmenter la longévité du réseau.
- Protocoles de routage prenant en compte les caractéristiques des RCSF.
- Méthodes de gestion de la couche MAC.
- Gestion de l'inter-couche
- Méthodes de clustering évoluées.
- Techniques d'auto-organisation.
- Utilisation des méthodes de datamining.
- Méthodes de cryptographie adaptées à ce type de réseau.

- Technique d'évitement d'attaques.
- Gestion de la tolérance aux fautes.
- La qualité de service.
- Méthodes pour réaliser la diffusion.
- Techniques d'équilibrage de charge entre les différentes entités du réseau.

Dans la section suivante on donne une définition de la tâche de diffusion et son importance dans les RCSF.

1.3. Importance de la tâche de diffusion dans les RCSF

Dans cette section, on se propose de définir la tâche de diffusion, son rôle et son importance dans les RCSF.

1.3.1. Définition de la diffusion

La diffusion, selon Larousse, est définie par:

"Un Phénomène par lequel un milieu de propagation produit une répartition...".

Selon wiki, la diffusion est définie comme suit:

"Dans le langage courant, le terme diffusion fait référence à une notion de «distribution», de «mise à disposition» (diffusion d'un produit, d'une information).

En informatique,

- *La diffusion (broadcast) consiste à envoyer le même paquet de données à tous les destinataires du réseau local.*
- *La multidiffusion (multicast) consiste à envoyer le même paquet de données à plusieurs destinataires.*
- *La rétrodiffusion (backscatter) se produit quand on envoie les messages d'erreur de remise d'un courrier électronique à quelqu'un qui ne l'a pas envoyé.*
- *En cryptographie, confusion et diffusion réfèrent à la dissipation des redondances statistiques en entrée d'une structure afin d'obtenir en sortie des statistiques «idéales»".*

Dans les RCSF, la diffusion est la dissémination d'un paquet de données dans le réseau:

- Du nœud de base vers tous les nœuds ou du nœud de base vers un sous-ensemble de tous les nœuds.

- D'un nœud vers son voisinage.

Dans le paragraphe suivant, on donne les caractéristiques d'une méthode de diffusion dans les RCSF.

1.3.2. Caractéristiques d'une méthode de diffusion

Une méthode de diffusion peut être décrite tout simplement par l'algorithme 1.1 qui est appelé l'inondation aveugle (Blind flooding).

Algorithme 1.1 : Diffusion par l'inondation aveugle

---

1. Un nœud fait la diffusion d'un paquet de données
2. Pour tous les autres nœuds:

   Rediffusion de ce paquet une et une seule fois après la première réception

---

Cet algorithme génère un grand nombre de réceptions redondantes du même paquet qui génère le problème de la tempête de diffusion (Broadcast Storm Problem) se traduisant par:

- La surconsommation d'énergie qui conduit à un épuisement rapide des batteries de capteurs.
- Des conflits lors de la réception provenant de plusieurs sources qui peuvent être traités au niveau de la couche MAC.
- Des problèmes de collusions qui peuvent être aussi traités au niveau de la couche MAC

En un mot on peut dire que:

Broadcast storm problem = {Collision + Redundancy + Contention}

Par Conséquent, les caractéristiques d'une bonne méthode de diffusion sont:

1. La réduction du problème de tempête de diffusion
2. La prise en compte des caractéristiques des nœuds capteurs et du RCSF par l'utilisation convenable de ses ressources.

3. La considération d'autre facteurs comme la vulnérabilité des capteurs et des communications entre eux face aux attaques et aux fautes.
4. La maximisation du taux de réussite dit aussi taux d'accessibilité c.-à-d. le pourcentage des nœuds capteurs ayant reçu le paquet diffusé par rapport au nombre total des nœuds capteurs concernés par la diffusion.

Après avoir défini la diffusion dans les RCSF, on va répondre à la question *"A quoi sert la diffusion dans les RCSF"*

1.3.3. Découverte du voisinage

La découverte du voisinage est réalisée par chaque nœud capteur pour déterminer l'ensemble des autres nœuds capteurs qui sont dans son rayon de communication. La détermination de l'ensemble des voisinages est importante lors d'un déploiement aléatoire et se fait périodiquement pour un RCSF dynamique puisque sa topologie est instable à cause des mouvements des capteurs ou à cause de la vulnérabilité des nœuds et des liens de communications.

La découverte du voisinage consiste à diffuser un petit paquet (Hello packet) contenant au moins l'identifiant du nœud capteur émetteur et ensuite attendre les réponses afin de remplir la table de voisinage.

La problématique de la détermination de la liste des voisinages, est la fréquence d'envoi de message de découverte. En effet, une fréquence faible peut conduire à une table de voisinage non à jour et une fréquence élevée peut conduire à une saturation de la bande passante des nœuds capteurs. Il est donc important de trouver un compromis entre ces deux critères en définissant la durée de vie des données dans la table de voisinage qui peut être déterminée selon le type du RCSF et la zone de déploiement.

La découverte de voisinage se fait généralement à 1-saut c.-à-d. découverte du voisinage direct, mais dans d'autres cas il est nécessaire de connaître en plus la liste de voisinage à 2-sauts. C'est le cas d'exécution d'une technique de diffusion basée sur le relai comme la méthode MPR [53] (Multipoints Relay) de relai multipoints.

1.3.4. Rôle de la diffusion pour réaliser le routage

La diffusion est utilisée dans les méthodes de routage pour:

- La découverte d'un chemin pour délivrer les paquets à router pour les protocoles de routage ordinaire.
- La découverte de plus qu'un chemin pour les méthodes de routage évoquant:
    - ✓ L'équilibrage de charge
    - ✓ La tolérance aux fautes
    - ✓ La sécurité
    - ✓ La qualité de service.
- La réparation des chemins défectueux.

Par exemple: la construction des routes pour le protocole AODV [52] est basée sur la diffusion des requêtes Route-Request/Route-Replay (RREQ/RREP) ensuite la participation des nœuds dans le RCSF permet de déterminer le chemin vers le destinataire.

Le protocole OLSR [13] est aussi l'un des protocoles qui utilise la diffusion en adoptant la méthode MPR. Ce protocole est basé sur deux types de messages:

- Message "Hello" diffusé par chaque nœud pour déterminer la liste des voisins.
- Message "TC: Topoly Control" envoyé périodiquement par chaque nœud contenant la liste de ses voisins choisis par la méthode MPR comme nœuds de relais afin de construire les tables de routage. Les messages TC sont diffusés dans le RCSF et ils sont rediffusés seulement par les nœuds MPR pour éviter la saturation du réseau.

1.3.5. Distribution d'informations

La distribution d'informations ou la dissémination des données dans les RCSF se fait généralement à partir du nœud de base vers le reste des nœuds ou localement. L'exemple typique de dissémination d'informations localement se fait lors de l'exécution des méthodes d'auto-organisation. En effet, l'élection des *cluster-head* est basée sur la dissémination par chaque nœud de ses informations personnelles comme son identifiant, son énergie restante et sa position vis-à-vis de ses voisins afin de faire une élection coopérative du *cluster-head*. La distribution d'informations est plus fréquente dans les réseaux mobiles et dynamiques que dans les réseaux statiques, puisque dans les réseaux qui changent de topologie, les techniques distribuées ou décentralisées sont plus appropriées que les techniques centralisées. En effet, plusieurs techniques de création des ensembles dominants connexes ou de détermination des nœuds de relai MPR pour les réseaux mobiles sont basées sur des algorithmes distribués qui mettent en

œuvre la participation coopérative d'un ensemble de nœuds par échange d'informations entre eux pour réaliser la tâche demandée.

1.3.6. Distribution des clés

Le déploiement des nœuds capteurs dans beaucoup de cas se fait dans des lieux incontrôlables et vulnérables aux attaques ce qui nécessite un certain niveau de sécurité des communications entre les différentes entités du RCSF surtout pour les applications sensibles comme dans le domaine militaire ou médical.

La cryptographie paraît comme solution primordiale pour beaucoup d'applications. En conséquence, les crypto-systèmes asymétriques basés sur les courbes elliptiques s'avèrent être une solution adéquate pour la sécurité dans les RCSF. En effet, ils offrent le même niveau de sécurité que RSA avec une taille de clés plus petite et un calcul moins coûteux.

Toutefois, la gestion et la distribution des clés [9] dans les RCSF est très importante avec un cycle de vie qui commence par leur production, leur enregistrement, leur protection, leur distribution, leur chargement, leur utilisation et finalement leur destruction.

La distribution de clés peut se faire par affectation et ceci avant le déploiement des capteurs mais cette solution, malgré son faible coût, présente beaucoup d'inconvénients surtout en cas de détection de clés.

En outre, la gestion et la distribution des clés peut se faire au fur et à mesure lors du fonctionnement du RCSF:

- Globalement: du nœud de base vers tous les nœuds, ensuite chaque nœud crypte ses informations collectées avant de les router vers le nœud de base.
- Localement: du *cluster-head* vers tous les nœuds membres du cluster, ensuite chaque nœud crypte ses informations collectées avant de les router vers le *cluster-head*.

1.3.8. Synchronisation du temps

Le but de la synchronisation du temps dans les RCSF est d'attribuer une base de temps commune pour les nœuds capteurs du réseau afin de dater ou d'ordonner les évènements ou évaluer le temps entre un ou plusieurs évènements.

La synchronisation a une grande importance dans les RCSF. Elle est basée sur un effort collaboratif de plusieurs nœuds capteurs permettant de fournir des agrégations de données, des ordonnancements d'évènements et des actions coordonnées. La synchronisation peut être globale pour les réseaux entiers ou globaux pour un sous-ensemble du réseau comme par exemple dans un cluster.

RBS (Reference Broadcast Synchronisation) [16] est l'une des techniques les plus connues de synchronisation de temps pour les RCSF. Le fonctionnement de RBS est simple. Tout d'abord, un émetteur diffuse un nombre M comme référence de broadcast. Ensuite chaque nœud capteur qui reçoit ce message diffuse vers ses voisins le temps de réception de ce message. Après, tous les nœuds capteurs calculent leur déphasage par rapport aux autres comme étant la moyenne des différences entre son temps local de réception du message contenant M et les autres temps de réception de même message contenant M.

1.4. Conclusion

Dans ce chapitre, on a défini le contexte de travail de la thèse en donnant un rappel sur les RCSF et l'importance de la tâche de diffusion comme tâche fondamentale et primordiale dans ce type de réseau.

En outre un RCSF peut être classé selon plusieurs dimensions: les données qu'il gère (scalaires ou multimédia), la mobilité (moyenne, rapide, statique), l'architecture d'organisation (organisé selon une architecture ou non), les types de capteurs (homogènes ou hétérogènes) et les lieux où ils sont déployés (terre, sous-marin, sous-terrain, espace…).

Dans cette thèse on s'intéresse aux RCSF dynamiques. Cet aspect dynamique peut être interprété selon deux axes:

- Axe du temps: où le RCSF change de topologie en fonction du temps lors de la disparition des nœuds capteurs à cause de multiples raisons (Epuisement de batterie, vol, mal-fonctionnement, panne matérielle ou logicielle, interférence d'onde radio, obstacle qui cause la coupure de la communication). La topologie peut encore changer par l'ajout d'un nouveau nœud capteur (absence d'obstacle qui cause la coupure de la communication, déploiement d'un ou d'un ensemble de nouveaux nœuds, changement d'une batterie épuisée d'un nœud).

- Axe spatio-temporel: prise en compte de la mobilité avec des changements de topologie décrite précédemment.

Par ailleurs, et comme développé dans ce chapitre, la diffusion est une tâche fondamentale et de même importance que la communication point à point dans les RCSF y compris les RCSF dynamiques où la tâche de diffusion reste toujours un sujet ouvert de recherche.

Dans le chapitre suivant on donne, l'état de l'art des techniques et des méthodes de diffusion dans les RCSF.

Chapitre 2

Étude des méthodes de diffusion

Résumé:

Dans ce chapitre on se propose d'étudier et de classer les principales méthodes de diffusion existantes dans les RCSF avec une étude comparative afin d'identifier les manques et les inconvénients de ces techniques pour introduire nos contributions dans les chapitres suivants.

2.1. Introduction

Comme on l'a mentionné dans le chapitre précédent, la diffusion est une tâche fondamentale et primordiale dans les RCSF. Dans ce chapitre, on va réaliser une étude à partir de la littérature pour extraire, classer et réaliser une comparaison des principales méthodes de diffusion.

Le déploiement aléatoire des nœuds capteurs d'un RCSF et les contraintes d'un capteur constituent un grand défi pour réaliser la diffusion. En effet, la réalisation de cette tâche peut se faire de deux façons:

- Centralisée, basée sur un seul nœud-capteur émetteur du paquet de données à diffuser.
- Décentralisée, basée sur la collaboration de plusieurs nœuds afin d'accomplir la diffusion.

Le but de ces méthodes est d'éviter les problèmes de diffusion aveugle en sélectionnant un sous-ensemble des nœuds capteurs qui soient connexes et permettant de couvrir le reste des nœuds. Ce sous-ensemble de nœuds est appelé *backbone*. Il permet de former une colonne vertébrale dans le réseau qui est seule responsable de la réalisation de la tâche de diffusion.

Dans ce chapitre, on donne tout d'abord des rappels et des définitions sur les théories des graphes. Ensuite, on donne une taxonomie des méthodes de diffusion dans les RCSF. La dernière partie du chapitre concerne une étude comparative des techniques de diffusion existantes afin de

dresser leurs avantages, leurs inconvénients et leurs points communs pour donner les orientations de ce travail de thèse.

2.2. Présentation

Dans cette section, on présente des rappels sur certains aspects de la théorie des graphes liées à la diffusion dans les RCSF. On donne de plus une taxonomie générale de ces méthodes et finalement on définit la notion du *backbone*.

2.2.1. Rappels et définitions.

- Modèle d'un RCSF :

  Comme on a mentionné dans le chapitre précèdent, les RCSF peuvent être représentés par un graphe non orienté *G(V,E)* où *V* représente l'ensemble des nœuds capteur et *E* représente l'ensemble des arrêtes entre les nœuds de *V*. Une arrête existe entre deux nœuds capteur *A* et *B* si et seulement si *B* est dans la zone de couverture de *A* et vice-versa
- Graphe k-connexe :

  un graphe non orienté *G(V,E)* est k-connexe si et seulement si il existe k chemins différents entre chaque couple de nœuds dans *G*.
- Connexion r sauts :

  deux nœuds *A* et *B* d'un graphe non orienté *G(V,E)* sont connectés à *r* sauts si et seulement si il existe au plus *r-1* nœuds dans le chemin qui relie *A* et *B*.
- Ensemble dominant connexe :

  CDS (Connected Dominating Set) est un sous-ensemble *C* connexe d'un graphe non orienté *G(V,E)* contenant des éléments dits dominants. Tous les éléments de *V* \ *C* sont dits dominés et sont connectés au moins à un nœud dans *C* (figure 2.1).

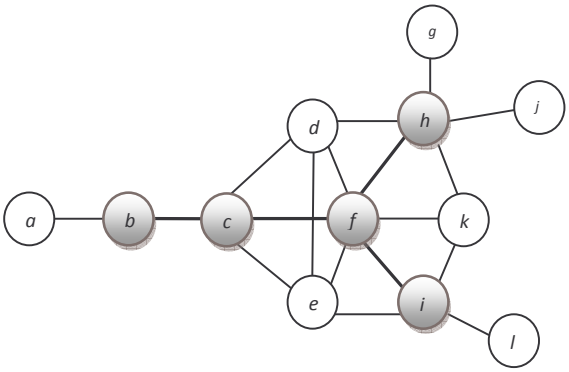


Figure 2.1. *L'ensemble {b,c,f,h,i} est un CDS du graphe*

– Ensemble dominant connexe à *r* sauts :

  r-CDS (*r*-hop Connected Dominating Set) est un sous-ensemble *D* connexe d'un graphe non orienté *G(V,E)* contenant des éléments dits dominants. Tous les éléments du *V* \ *D* sont dits dominés et sont connectés à r sauts à au moins un nœud dans *D*.

– Ensemble (k,m)-CDS:

  (k-Connected m-Dominating Set) est un sous-ensemble *C′* k-connexe d'un graphe k-connexe non orienté *G(V,E)* contenant des éléments dits dominants. Tous les éléments de *V* \ *C′* sont dits dominés et sont connectés au moins à m nœud dans *C′*.

– Ensemble (k,r)-CDS

  (r-hop k-Connected Dominating Set) est un sous-ensemble *D*' connexe d'un graphe k-connexe non orienté *G(V,E)* contenant des éléments dits dominants. Tous les éléments de *V**D′* sont dits dominés et sont connectés en r-saut à au moins k nœud dans *D′* (figure 2.2).

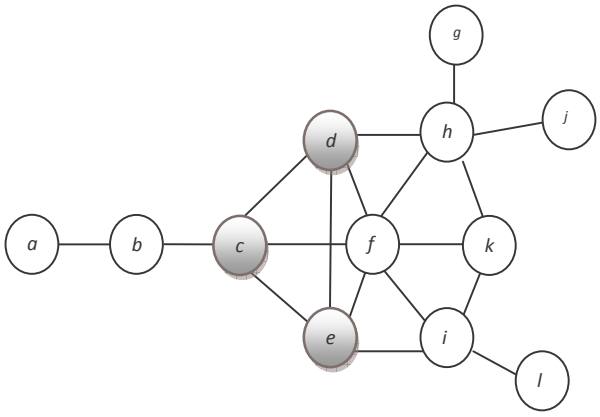


Figure 2.2. *L'ensemble {b,c,e} est un (2,2-saut)-CDS du graphe*

– Arbre couvrant :

  Un arbre couvrant est sous graphe non orienté *G(V,E)* acyclique dont le nombre de nœuds dans l'arbre est |*V*|.

– K arbres couvrant :

  Les k arbres couvrants dans un graphe k-connexe non orienté *G(V,E),* sont k arbres distincts dans *G*.

2.2.2. Classification des méthodes de diffusion

Les méthodes de diffusion dans les RCSF peuvent être classées en plusieurs catégories. Dans la figure 2.3, on donne les principales techniques et dans la section qui suit on détaille ces techniques.

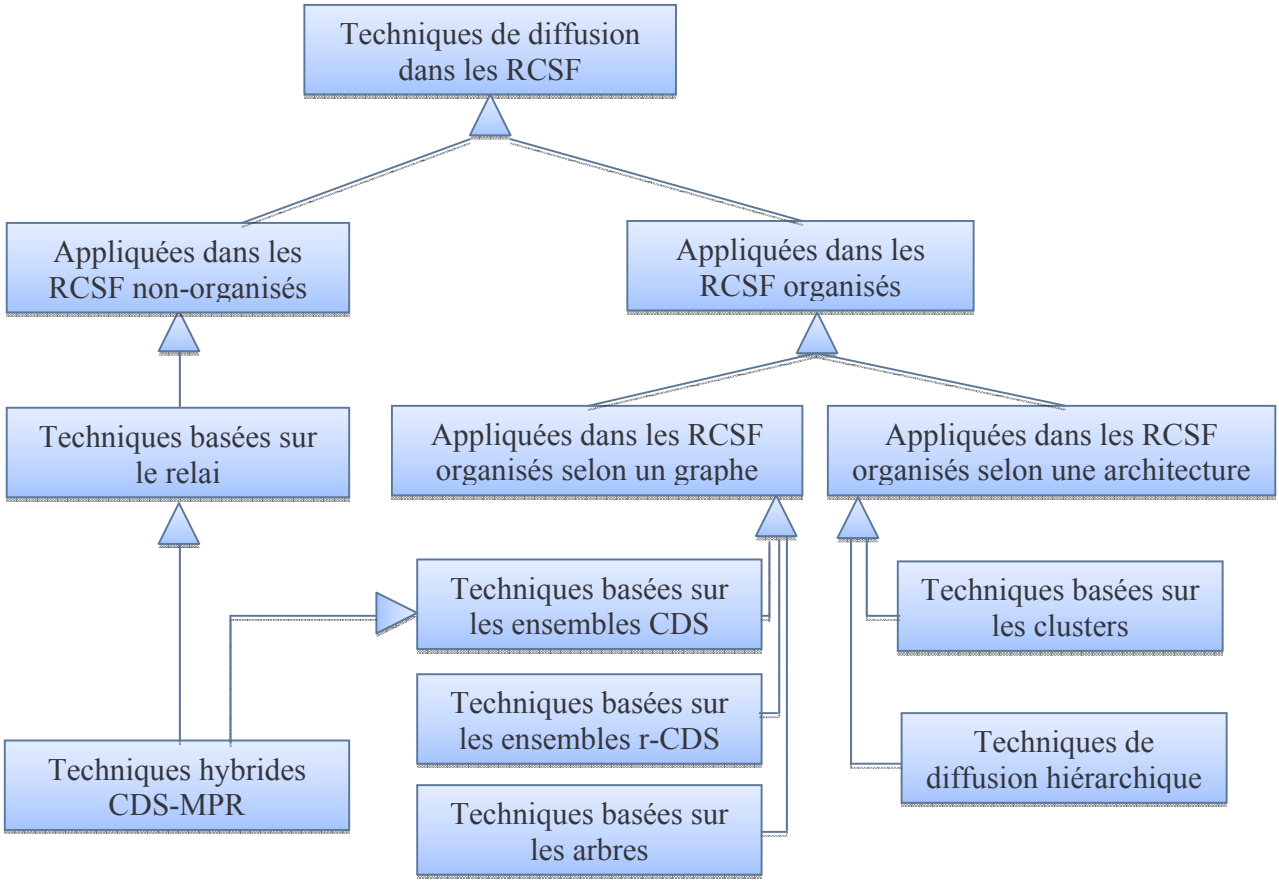


Figure 2.3. *Classification des techniques de diffusion dans les RCSF*

2.2.3. Notion de backbone

Les différentes techniques de diffusion sont appliquées pour permettre:

- La réduction du problème de tempête de diffusion en minimisant les nœuds-capteurs qui rediffusent.
- La prise en compte des caractéristiques de nœuds capteurs par l'utilisation convenable de leurs ressources et par la considération d'autres facteurs comme la vulnérabilité des capteurs et des communications entre eux face aux attaques et aux fautes.

– La maximisation du taux de réussite appelé également taux d'accessibilité.

Ces techniques permettent de sélectionner un sous-ensemble de nœuds capteurs qui soient connexes et qui permettent de couvrir le reste des nœuds. Ces sous-ensembles de nœuds sont appelés *backbone* permettant de former une colonne vertébrale dans le réseau qui est responsable pour réaliser la tâche de diffusion. Dans le tableau 2.1, on donne la composition d'un *backbone* selon la méthode choisie pour réaliser la diffusion.

Tableau 1.1. *Elément du backbone selon la méthode de diffusion*

Technique de diffusion	***backbone***
Basée sur le relai	Ensemble des nœuds sélectionnés pour faire le relai plus le nœud initiateur de diffusion.
Basée sur les clusters	Selon l'architecture de *clustering* : – Ensemble des têtes de clusters (si les têtes sont directement liées). – Ensemble des têtes de clusters plus les passerelles (si les têtes sont liées à travers des passerelles).
Diffusion hiérarchique	Ensemble des nœuds capteurs choisis leur type ou leur fonctionnalité pour faire la diffusion.
Basée sur les arbres	Tous les nœuds capteurs dans l'arbre sauf les feuilles.
Basée sur les CDS	Les nœuds capteurs dominants.
Basée sur les r-CDS	Les nœuds capteurs dominants + les nœuds dominés dans le sens du dominant vers les nœuds a plus que 2-sauts.

2.3. Taxonomie des méthodes de diffusion

Dans cette section, on donne un aperçu sur les techniques de diffusion selon la taxonomie présentée précédemment dans ce chapitre.

2.3.1. Diffusion basée sur le relai.

La diffusion par relai MPR (MultiPoint Relay : Relai Multi Points) a été proposée en 2002 dans un article de Qayyum et al. [53] . Dans cette technique, la diffusion se fait par niveau en commençant par le nœud source de diffusion. Le premier niveau est l'ensemble du premier voisinage du nœud source et ainsi de suite. Le nœud source, en connaissant les informations de son voisinage à deux sauts, peut sélectionner la liste des nœuds de son premier voisinage pour relayer les messages à diffuser. Chaque nœud qui reçoit le message à diffuser et qui est choisi comme nœud de relai exécute la même procédure réalisée au niveau du nœud source pour sélectionner les nœuds relais. Dans la figure 2.4. on donne un exemple du choix des nœuds de relai.

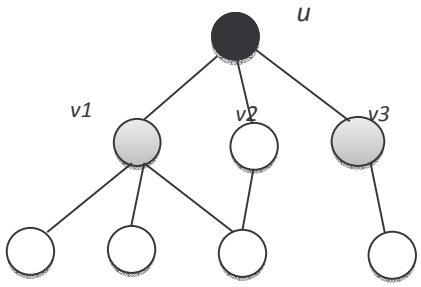


Figure 2.4. *Exemple d'application de la méthode MPR*

La méthode MPR est basée sur la réduction de redondance en sélectionnant seulement un sous-ensemble des nœuds du premier voisinage pour relayer le paquet à diffuser. Dans la figure 2.4. seulement *{v1,v3}* relayent. Ci-dessous on donne l'algorithme MPR.

Dans l'algorithme 2.1, *N1(u)* représente l'ensemble de nœuds du premier voisinage du nœud *u*, *N2(u)* représente l'ensemble de nœuds du deuxième voisinage du nœud *u* et *S2(u)* représente l'ensemble temporaire de nœuds du deuxième voisinage du nœud *u* qui ne sont pas couverts par les nœuds choisis pour faire le relai.

***Algorithme 2.1 : MPR* [53]**

1. Initialement

MPR(u)=Ø

S2(u)= N2(u)

2. Sélectionner les nœuds imposés de *S2 (u)* c á d ayant un seul parent dans *N1(u),* ensuite éliminer ces nœuds de *S2(u)* et ajouter leur parent á *MPR (u)*

3. Tant que *S2(u) ≠ Ø*

   Choisir le nœud *v* dans *N1(u)\ MPR(u)* qui couvre le maximum de nœuds dans *S2(u)*

   Ajouter *v* à *MPR(u)*

   Eliminer *N1(v)* du *S2(u)*

Si on applique l'algorithme précèdent à la figure 2.4, les nœuds de relai seront *v3* puisqu'il est imposé et *v1* puisqu'il couvre plus de nœuds dans le deuxième voisinage de *u*.

La méthode MPR a réduit remarquablement l'énergie totale pour réaliser la diffusion ce qui a permis d'augmenter la longévité de réseau et de réduire le problème de la tempête de diffusion (*Broadcast Storm Problem*).

Durant les années qui ont suivi son apparition, la méthode MPR a subi beaucoup d'amélioration. Dans [2] cette technique a été utilisée pour la première fois pour sélectionner un ensemble dominant connexe permettant à la fois de réaliser le routage et la diffusion. La méthode MPR a été aussi combinée avec le calcul des ensembles indépendants maximums pour réaliser la méthode GMPR (Getaway MPR) qui permet de minimiser l'ensemble des nœuds relai [38].

Dans [39] les auteurs présentent les principales méthodes basées sur MPR en donnant une comparaison en termes de complexité temporelle, complexité en nombre de messages, de taille

de l'information qui vient du voisinage d'un nœud et de l'indépendance de la source. Parmi ces méthodes on peut citer :

- L'heuristique ID-MPR (in-degree MPR) [39,43].

  Elle a pour but de minimiser la complexité de la méthode originale MPR à cause de la valeur maximale du degré sortant du premier voisinage d'un nœud choisi pour faire le relai. ID-MPR est basée sur une valeur appelée *in-degree* qui pour un nœud *A* est égale au nombre des voisins partagés entre le nœud *A* et un nœud *B* du premier voisinage de la source de diffusion. *A* est un nœud du deuxième voisinage de la source de diffusion.

- MO-MPR : (Minimum Overlapping MPR) [39].

  Amélioration de MPR afin de minimiser les collisions et minimiser les chevauchements c.-à-d. le nombre de nœuds qui sont liés à plus d'un nœud choisi comme nœuds MPR. Cette heuristique est basée sur la répartition maximum des nœuds MPR autour du nœud initiateur de diffusion afin de réduire les interférences dans le réseau et la collision lors de la réception du message diffusé.

- P-MPR: (Prioritized MPR) [39].

  Heuristique liée à MO-MPR afin de réduire les chevauchements mais sans augmentation du nombre total de nœuds MPR. Ceci est réalisé par la modification des critères de choix c.-à-d. dans P-MPR. Quand il y a plus d'un nœud du premier voisinage permettant de couvrir les nœuds du deuxième voisinage, le nœud qui a le minimum de *out-degree* (qui permet de couvrir plus de nœuds du deuxième voisinage) est choisi.

- RP-MPR (Random Prioritized MPR) [39].

  Heuristique combinant MPR et MO-MPR afin de réduire les chevauchements mais sans augmentation du nombre total des nœuds MPR. Ceci est réalisé par la modification des critères de choix c.-à-d. dans RP-MPR quand il y a plus qu'un nœud du premier voisinage permettant de couvrir les nœuds du deuxième voisinage, le nœud MPR est choisi aléatoirement. Les techniques P-MPR et RP-MPR sont identiques et ont le même but sauf dans le choix du nœud MPR quand il y a plus qu'un choix.

- U-MPR : (Utility-based MPR)[39].

  Le but essentiel de U-MPR est l'utilisation convenable de l'énergie afin de minimiser la consommation inutile pour augmenter la longévité du réseau. Ceci est réalisé par la réduction des retransmissions non nécessaires. Dans cette heuristique, les nœuds sont considérés hétérogènes bien qu'ils soient homogènes du point de vue matériel. U-MPR essaye de faire une distribution équitable de la charge en commençant par les nœuds les plus appropriés.

- UBF : (Utility Based Flooding) [39].

  Le but de UBF est d'améliorer la méthode U-MPR en donnant une autonomie totale aux ressources du réseau. Pour U-MPR le choix de faire la distribution est réalisé seulement après l'application de la méthode MPR aux nœuds qui restent et qui possèdent plusieurs possibilités de choix, mais pour UBF le choix d'équilibrage de charge est réalisé dès le début et même avant l'application de MPR.

La méthode MPR a été aussi améliorée selon un autre critère qui est l'adoption d'une couche physique réaliste ce qui permet d'améliorer nettement le taux de réussite de cette méthode c.-à-d. le nombre total des nœuds capteurs dans le réseau qui reçoivent correctement les messages diffusés. Ces améliorations sont basées sur un modèle probabiliste permettant de tenir compte des facteurs qui ont un impact sur la communication dans le monde réel comme le facteur d'atténuation du signal selon l'environnement de communication et la distance qui sépare l'émetteur et le récepteur [19] et [35].

2.3.2. Diffusion basée sur les ensembles dominants connexes.

Beaucoup d'efforts de recherche ont été réalisés pour la résolution du problème de sélection d'un CDS dans un réseau représenté par un graphe *G(V,E)*. L'importance du CDS vient de son utilité à la fois pour le routage et la diffusion dans un réseau à topologie aléatoire. La figure 2.5 représente un CDS dans un graphe de RCSF.

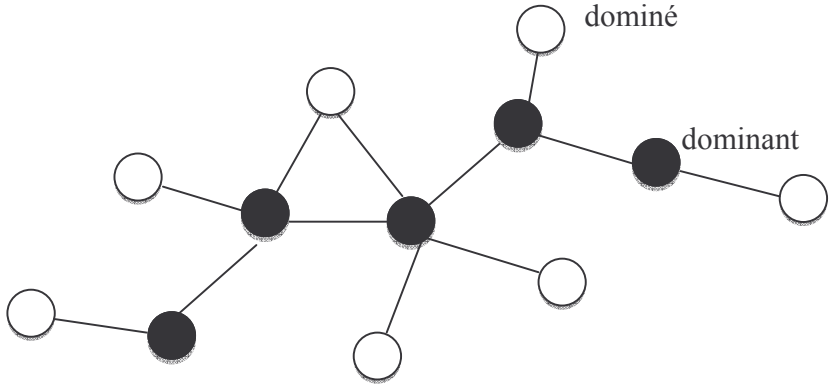


Figure 2.5. *CDS dans un RCSF*

Le principe de la communication entre deux nœuds *A* et *B* dans le RCSF selon un ensemble CDS est basé sur le fait que chaque nœud dominé connaît son nœud dominant et que chaque nœud dominant connaît son ensemble de nœuds dominés. La communication commence lorsque *A* envoie le paquet vers son nœud dominant ensuite ce paquet sera routé tout au long de l'ensemble CDS jusqu'à arriver au nœud dominant de *B* qui envoie le paquet vers *B*. Dans la figure 2.6, on donne les étapes de cette communication.

Pour réduire le coût de la communication (moins d'énergie, de temps, de communications et de surcharge du réseau) le but de la plupart des travaux de recherche est de sélectionner un CDS de taille minimale (MCDS) c.-à-d. minimiser le nombre des nœuds dominants dans le réseau de capteurs ou maximiser le nombre des nœuds dominés. C'est un problème NP-complet [22].

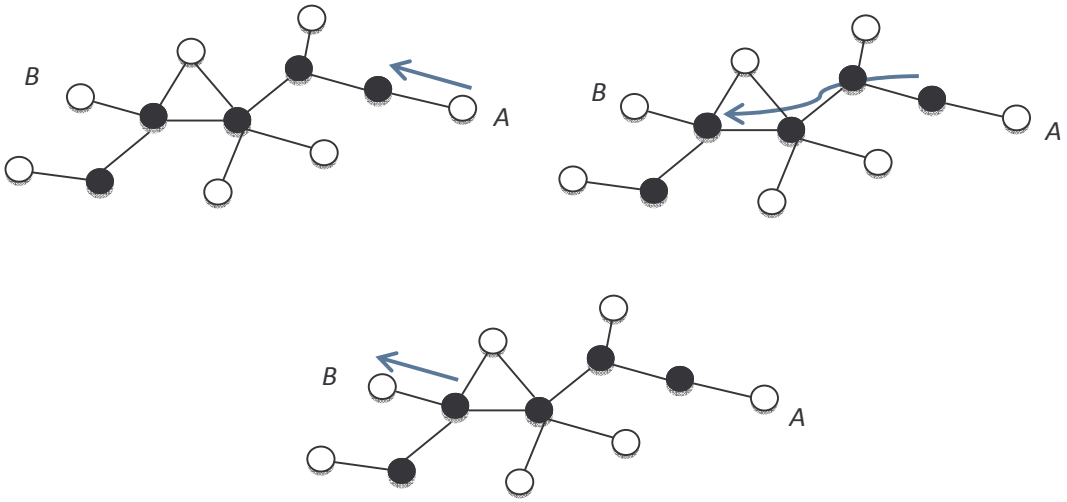


Figure 2.6. *Communication à travers un CDS de A vers B*

Vue l'importance de sélection du CDS, beaucoup de techniques ont été proposées pour résoudre ce problème [73,25]. Ci-après on cite les principales méthodes de construction des CDS selon leur type:

- Méthode centralisée ou décentralisée :
  Les techniques centralisées permettent généralement de construire un CDS de taille inférieure et de performance plus importante que les techniques décentralisées. Toutefois ce type de sélection est basé sur la supposition de la connaissance de la topologie entière du réseau. Pour la construction décentralisée on peut distinguer deux catégories : construction distribuée où le processus de décision se fait d'une façon décentralisée et construction localisée où le processus de décision n'est pas seulement distribué mais demande un certain nombre de communications.
- Méthode basée sur les ensembles maximaux indépendants ou non :
  Les techniques basées sur les ensembles maximaux indépendants MI permettent essentiellement de minimiser la taille du CDS en maximisant le nombre de nœuds dominés en premier lieu par la construction des ensembles MI ensuite en connectant ces ensembles pour construire un CDS. L'utilisation de cette technique peut être centralisée ou décentralisée selon le modèle du réseau.

Le but essentiel de toutes les méthodes de construction des CDS est la minimisation. Des travaux de recherche récents introduisent un nouveau concept qui est l'équilibrage de charge ou lieu de la minimisation. On traite ce concept plus tard dans ce chapitre.

2.3.3. Diffusion basée sur des techniques hybrides.

Les techniques hybrides de diffusion sont basées sur la combinaison de plus d'une méthode pour sélectionner un *backbone* pour réaliser toutes sortes de communications y compris la diffusion. Comme exemple de ce type des méthodes on peut citer la méthode CDS-MPR qui est la construction d'un ensemble dominant connexe par l'utilisation de relais multipoints [2]. Cette technique est basée sur la connaissance des nœuds de relai sélectionnés par la méthode MPR et l'identifiant ID des nœuds du premier et du second voisinage. Un nœud *A* appartient au CDS si et seulement si il respecte les règles suivantes :

- Règle 1: il a le plus petit ID de son premier voisinage.
- Règle 2: il a été sélectionné comme un nœud MPR et son sélecteur a le plus petit ID du premier voisinage de *A*.

La première règle s'applique à tous les nœuds pour sélectionner l'ensemble des nœuds MPR, mais la deuxième s'applique seulement aux nœuds MPR, donc MPR est un cas spécial du CDS-MPR dans le cas où tous les nœuds MPR vérifient la deuxième règle. La méthode CDS-MPR comme la méthode MPR s'exécute par étape en connaissant seulement les informations à deux voisinages d'un nœud ce qui la rend applicable pour les réseaux à topologie inconnue spécialement les réseaux dynamiques et mobiles.

L'inconvénient majeur de la méthode CDS-MPR est qu'elle augmente le nombre de retransmissions entre les nœuds.

Dans [69] les auteurs proposent EMPR (Enhanced MPR) qui est une amélioration de CDS-MPR dans le but de réduire les retransmissions en améliorant la première règle de sélection des dominants qui devient :

- Règle 1: Le nœud qui a le plus petit ID parmi tous les nœuds de son premier voisinage et qui a deux voisins non connectés.

EMPR propose aussi une amélioration de la méthode MPR comme suit :

***Algorithme 2.2 : EMPR* [69]**

---

1. Initialement

   Ajouter tous les nœuds non-connectés à la source aux nœuds MPR et éliminer les nœuds à 2-sauts qui sont couverts par ces nœuds non-connectés.

2. Appliquer l'heuristique MPR aux nœuds à 1-saut restants pour couvrir tout le reste des nœuds à 2-sauts et utiliser l'ID des nœuds comme critère de choix dans le cas où deux nœuds peuvent couvrir le même nombre des nœuds à 2-sauts non encore couverts.

Dans [12], les auteurs étudient ce même problème afin de réduire encore plus la taille du CDS sélectionné en proposant la méthode DEMPR (Degree-based Enhanced MPR). Cette amélioration a été réalisée à travers la modification des deux règles de sélection des nœuds du CDS qui sont devenues:

- Règle 1 étendue: un nœud fait partie du CDS s'il a le plus grand degré parmi tout son premier voisinage et s'il a deux voisins non connectés.
- Règle 2 étendue: un nœud fait partie du CDS s'il a été sélectionné comme nœud MPR et son sélecteur a le plus grand degré des nœuds de ses voisins à un seul saut.

Dans [70], il y a une proposition de EEMPR (Extended Enhanced MPR) permettant d'améliorer toutes les méthodes précédentes par l'utilisation des informations à trois voisinages afin d'extraire un CDS de taille inférieure. Cette amélioration a été réalisée en utilisant une amélioration de MPR, la même règle 1 de la méthode EMPR et en modifiant la règle 2 comme suit :

- Règle 2 améliorée: un nœud *A* fait partie du CDS si:
  1. Il a été sélectionné directement comme nœud MPR et son sélecteur a le plus petit ID des nœuds du premier voisinage de *A*.
  2. Il a été sélectionné indirectement comme nœud MPR et son sélecteur a le plus petit ID des nœuds du premier voisinage de *A*.

La méthode EEMPR a permis de sélectionner un CDS de taille nettement inférieure que celle des techniques précédentes

2.3.4. Diffusion basée sur les ensembles dominants connexes à r sauts.

Comme définis au début de ce chapitre, les ensembles r-CDS sont des CDS spécifiques où chaque nœud dominé est à r-1 sauts de son dominant. Dans la littérature ces ensembles sont utilisés pour la réduction de taille des CDS [51]. Dans la figure 2.7, on donne un exemple pour voir la différence entre un CDS et un 2-CDS.

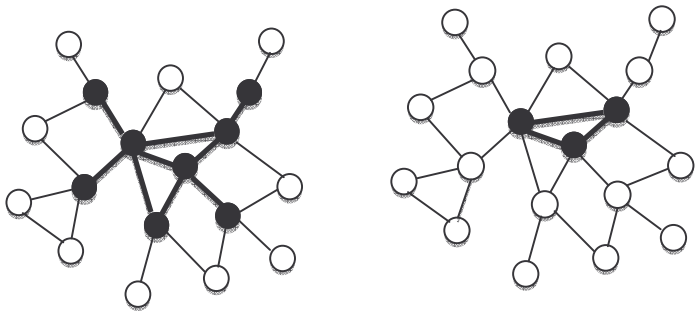


Figure 2.7. *Exemple d'un CDS (à gauche) et d'un ensemble 2-saut CDS (à droite) dans un RCSF*

Les ensembles r-CDS sont introduits essentiellement pour minimiser la taille des CDS afin de minimiser le coût des communications.

Les auteurs de [21] proposent une méthode pour le calcul d'un ensemble d-hop CDS qui est appliquée dans un RCSF organisé en clusters. Cette technique est réalisée en sélectionnant un d-CDS de chaque groupe. Ensuite, si la combinaison de ces d-CDS ne donne pas un ensemble connexe on ajoute certains nœuds qui minimisent les chemins entre les ensembles d-CDS afin de garantir la connectivité et la minimisation de l'ensemble sélectionné.

Dans la littérature on trouve, en particulier, les ensembles (k,r)-CDS où la variable k désigne la connectivité de l'ensemble r-CDS c.-à-d. le nombre de chemins indépendants entre deux sommets de cet ensemble. L'utilisation des (k,r)-CDS a été introduite pour la minimisation de la taille du CDS comme dit précédemment. De plus, on augmente la fiabilité de la communication à travers cet ensemble, puisque en cas de problème dans un chemin entre un couple de nœuds dominants on peut le substituer par k-1 autres chemins indépendants.

Parmi les méthodes de construction des ensembles (k,r)-CDS on peut citer celle de [37] où les auteurs proposent deux algorithmes approximatifs pour résoudre le problème. Le premier algorithme est basé sur un graphe pondéré en s'appuyant sur les énergies des nœuds capteurs ensuite réaliser la construction par la combinaison entre l'ensemble (k,r)-CDS du graphe d'origine et celui du graphe pondéré. Le second algorithme s'exécute en deux phases la première consiste à trouver un ensemble r-hop k-dominants indépendant ensuite connecter cet ensemble. Dans [50] les auteurs introduisent un algorithme distribué afin de construire des ensembles d-dominant qui sont connectés à d sauts et ayant les caractéristiques de plus court chemin. L'idée de construction est basée sur une distance égale à r sauts et étant le plus court chemin entre eux.

Ensuite, il faut trouver le nœud avec le plus grand ID parmi l'ensemble des nœuds qui forment le plus court chemin. Finalement les nœuds de plus grand ID forment l'ensemble à déterminer.

2.3.5. Diffusion basée sur les clusters.

La diffusion dans un réseau organisé en clusters dépend essentiellement de la méthode d'organisation choisie.

Tout d'abord l'approche d'auto-organisation dans les RCSF est la technique d'organisation des nœuds capteurs permettant de minimiser les communications réalisées en cas de redondance des données, la consommation inutile d'énergie et des collisions lors de multiples réceptions simultanées. De plus l'auto-organisation permet de mieux distribuer les tâches entre les entités du réseau. Elle permet, en particulier, d'éviter la surcharge inutile des communications en adoptant les approches d'agrégation lors de la collecte des données.

Le *clustering* a été largement étudié dans les RCSF. Il consiste à diviser le réseau en groupes. Chaque groupe est dirigé par un nœud appelé *cluster-head* (*CH*). Les nœuds restants dans le même groupe sont dits membres du cluster (*CM*). Le lien entre les groupes se fait soit directement entre les *CH* où à travers des nœuds spéciaux dits passerelles (*G*). Dans la figure 2.8, on donne un exemple de transformation d'un RCSF en un RCSF auto-organisé en clusters et dans le tableau 2.2 on donne les différentes possibilités pour connecter les clusters.

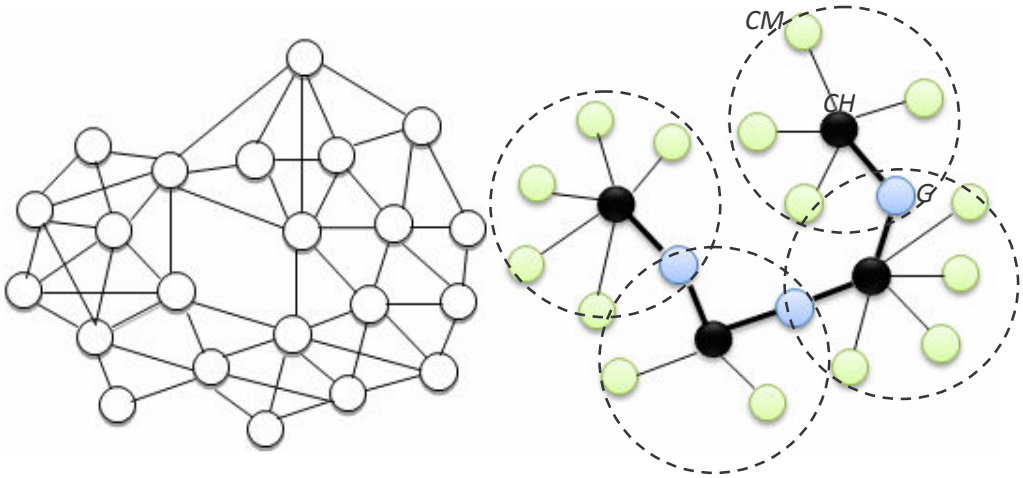
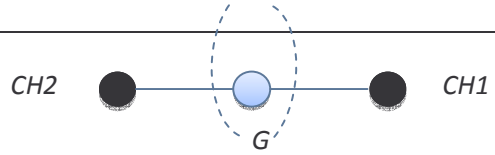
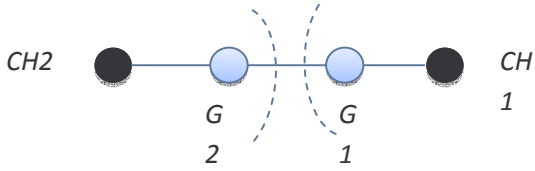


Figure 2.8. *Organisation d'un RCSF (à gauche) en clusters (à droite)*

Tableau 2.2. *Différents types de liaison entre les têtes de clusters*

Liaison	Schéma
Directe entre les *cluster-head*	
A travers une passerelle commune	
A travers deux passerelles indépendantes	

Le principe de diffusion en utilisant les clusters est basé sur l'utilisation du *backbone* formé par les *cluster-head* et les nœuds de liaison entre eux pour disséminer les données tout au long du réseau.

Vu ses avantages présentés précédemment, beaucoup d'algorithmes d'auto-organisation ont été proposé dans les RCSF. Selon [1] et [8] les algorithmes les plus connus et importants d'auto-organisation sont :

- LEACH (Low-Energy Adaptive Clustering Hierarchy)[1,8]:

  LEACH est un algorithme distribué où les nœuds réalisent des décisions autonomes. Initialement, un nœud décide d'être un *CH* avec une probabilité *p* et diffuse sa décision. Pour le reste des nœuds qui ne sont pas des *CH,* ils choisissent un *CH* qui minimise le coût de communication en énergie. LEACH fournit trois avantages qui sont:

  (1) le choix du *CH* pour chaque nœud n'est pas coûteux en communication puisque la décision se fait d'une façon autonome,

(2) chaque nœud calcule l'énergie minimale pour faire le choix de son *cluster-head*,

(3) les clusters fonctionnent indépendamment les uns des autres. Malgré ses avantages, LEACH n'est pas applicable si la station de base est loin du *CH*. Pour cela, beaucoup d'améliorations de cet algorithme ont été proposées comme LEACH-C et TL-LEACH.

- HEED (Hybrid Energy Efficient Distributed Clustering) [61]

  HEED est une technique où l'élection des *cluster-heads* se fait en fonction des distances entre les nœuds, le but de HEED est:

  (1) faire la distribution de la consommation d'énergie pour augmenter la longévité du réseau,

  (2) minimiser le gaspillage d'énergie lors de la sélection des têtes des clusters,

  (3) réduire les communications.

- EECS (Energy Efficient Clustering Scheme) [57]

  La minimisation de la consommation d'énergie dans EECS se fait de la même façon que dans LEACH. Mais l'amélioration principale de LEACH réside dans l'utilisation des tailles de clusters dynamiques dépendant de la distance entre la station de base et les clusters.

- WCA (Weighted Clustering Algorithm) [11]

  Il s'agit de calculer un poids pour tous les nœuds afin de sélectionner les meilleurs nœuds comme des *cluster-head*, ce poids est fonction du degré du nœud, son énergie restante, sa mobilité et les distances de ses voisins.

2.3.6. Diffusion hiérarchique.

Les architectures hiérarchiques sont appropriées dans les RCSF dans deux cas :

- Le RCSF est hétérogène. Donc les capteurs sont organisés selon leur type et leurs caractéristiques.
- Les capteurs n'ont pas la même fonctionnalité. Donc ils sont organisés selon leurs activités.

Dans beaucoup de cas, les méthodes de diffusion hiérarchique sont appliquées dans un RCSF organisé selon une méthode hiérarchique comme LEACH.

L'un des protocoles les plus connus de dissémination hiérarchique est TTDD (Two-Tier Data Dissemination) [56]. Il permet de minimiser l'explosion de messages d'inondation en se basant sur la technologie des grilles afin de diviser le réseau en petites grilles et faire l'inondation seulement au sein de ces grilles.

Dans [55], les auteurs proposent trois méthodes de dissémination des données dans un RCSF à grande échelle :

- FDDD (Fully Distinct Data Dissemination Hierarchy)
  Dans cette méthode chaque source qui génère les données sélectionne un nouveau nœud primaire dans sa région rectangulaire pour faire la diffusion, ce qui conduit à une hiérarchie différente pour chaque nœud.
- FSDD (Fully Sharing Data Dissemination Hierarchy)
  Pour cette technique de dissémination hiérarchique partagée, la sélection du nœud primaire est un peu différente. En effet, c'est le premier nœud source qui commence la diffusion qui sélectionne le nœud primaire.
- LSDD (Limited Sharing Data Dissemination Hierarchy)
  C'est une technique hybride entre les deux premières techniques en utilisant à la fois la dissémination hiérarchique complètement distincte pour chaque source et une dissémination hiérarchique partagée.

Dans ces techniques, les auteurs prennent en considération la mobilité, la minimisation de la consommation d'énergie et la tolérance aux fautes.

2.3.7. Diffusion basée sur les arbres.

La diffusion basée sur les arbres utilise les techniques de sélection des arbres couvrants dans un graphe pour définir un *backbone* virtuel formé par l'arbre privé de ses feuilles [24,4,41]. Dans la littérature, il y a plusieurs techniques qui utilisent cette approche parmi lesquelles on peut citer:

- Dans [24], les auteurs présentent une méthode de construction des arbres couvrants de poids minimal où le poids d'une arrête *(u,v)* est l'énergie nécessaire pour envoyer un paquet de *u* à *v*. Cette technique est utilisée pour les réseaux ad hoc mobile MANET.

- Dans [4], on trouve une proposition d'une technique de construction d'un arbre couvrant efficace qui est stable, sûr et prend en considération l'aspect dynamique du réseau.
- Un algorithme distribué, sûr et permettant de trouver un arbre de coût minimal a été proposé dans [42] (Broadcast et Multicast) dans les RCSF tout en tenant compte de l'aspect énergétique des capteurs.

2.4. Étude comparative

D'après ce qui précède, on remarque que la diffusion est un problème qui a été bien étudié par la communauté scientifique. Ainsi, dans la section suivante, on présente une étude des solutions de diffusion listées précédemment.

2.4.1. La minimisation

La minimisation de la taille du *backbone* a été le but de tous les travaux cités précédemment. En effet, il faut minimiser la taille du *backbone* pour minimiser le coût des communications afin d'augmenter la longévité du réseau. Dans la figure 2.9, on donne l'évolution des principales méthodes de sélection des *backbone*s virtuels dans les RCSF.

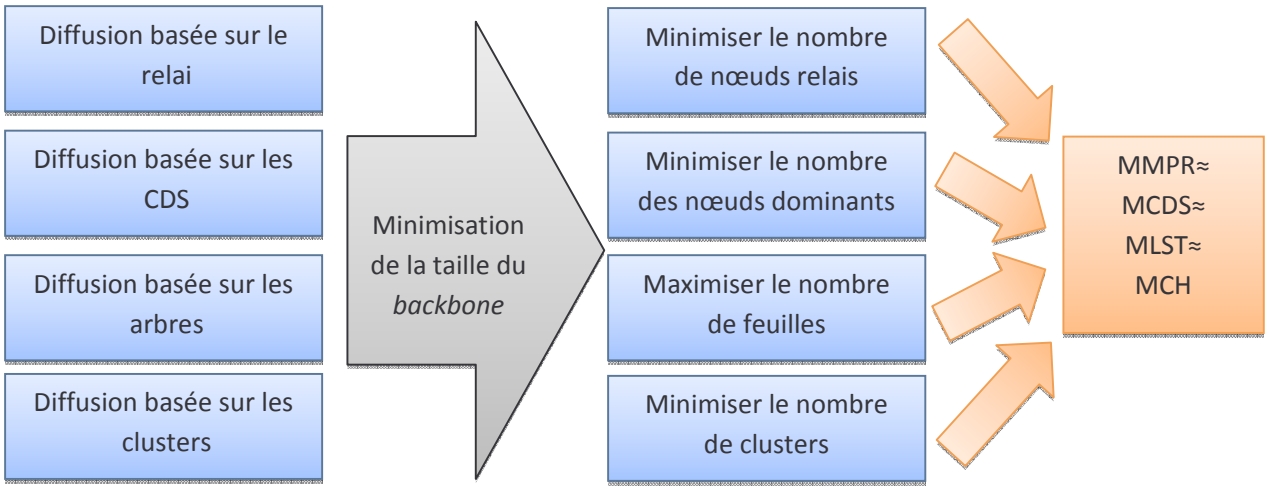


Figure 2.9. *Concept de minimisation dans les RCSF*

Afin de minimiser la taille du *backbone* beaucoup de travaux ont été réalisés sur ce problème qui est NP-Complet. En effet, MCDS (Minimum Connected Dominating Set) est équivalent à MLST (Maximum Leaf Spaning Tree) lui aussi équivalent à MMPR (Minimum

relay node in MPR) qui est aussi équivalent à minimiser le nombre de cluster : MCH (Minimum Cluster Head).

2.4.2. Tolérance aux fautes

La tolérance aux fautes dans les RCSF a une grande importance. En effet, le déploiement et le fonctionnement d'un RCSF sont soumis aux pannes dues au disfonctionnement du matériel, aux problèmes logiciels, aux dangers environnementaux, etc. [15]. De ce fait, garantir un bon niveau de tolérance aux fautes est très important pour un fonctionnement normal d'un RCSF. Dans ce cadre, beaucoup de travaux de recherche ont été réalisés [23].

Ci-après on cite les différents types de fautes dans les RCSF selon les couches.

- Couche matérielle : Les fautes dans cette couche sont dues aux disfonctionnements d'un composant. Exemple: microprocesseur, batterie, unité de capture, interférence réseaux. La cause de ces fautes est due à la contrainte d'énergie; par exemple pour un certain niveau d'énergie dans le capteur il y a des erreurs de lecture.
- Couche logicielle : les logiciels dans un capteur peuvent comporter des bugs ce qui conduit à des fautes.
- Couche de communication : Les pannes dans la couche de communication sont dues principalement aux fautes générées par le sans fil. Ces pannes sont liées à l'environnement externe par exemple aux interférences des ondes.

Compte-tenu de ces différents types de fautes il devient important de définir des techniques de diffusion tolérantes aux fautes comme suit:

- Définir plus qu'un chemin pour rejoindre un nœud capteur.
- Eviter de choisir les capteurs dans un *backbone* pour ceux possédant une probabilité élevée d'être en faute.

Dans notre travail on utilise un indice représentant le degré de confiance pour un nœud. Cet indice qui varie entre 0 et 1 évolue selon le comportement du nœud. Un nœud est plus sûr lorsque son indice de confiance est plus élevé. Plusieurs fautes peuvent être la cause d'un tel mauvais comportement par exemple: un nœuds peut être placé dans un milieu dangereux, à un bas niveau d'énergie un nœud peut donner des mauvais mesures, ou des autres fautes liées à la couche matérielle ou logicielle du nœud-capteur.

Dans la littérature les travaux les plus importants dans ce domaine sont:

- Les ensembles km-CDS permettant de définir plus qu'un chemin pour rejoindre un nœud capteur. En effet, il y a *k* chemins différents entre chaque nœud dominant. De plus, chaque nœud dominé est connecté à *m* nœuds dominants [65,59,60,18].
- La sélection de k-Tree a été utilisée pour définir des chemins distincts entre les nœuds qui sont utiles en cas de rupture de connexion [6].
- L'utilisation d'un *backbone* formé par des groupes de nœuds pour remplacer un nœud en panne par un autre dans le même groupe [63].
- Les ensembles (k,r)-CDS pour jouer le même rôle que les ensembles km-CDS mais avec une taille plus petite pour minimiser les coûts des communications [10,72].

2.4.3. Equilibrage de charge

Récemment des travaux de recherche ont montré que la sélection d'un *backbone* qui offre un équilibrage de charge est plus important que minimiser sa taille et permet réellement d'augmenter la longévité du réseau par rapport à un *backbone* de taille minimale.

En effet Jing He et al. ont proposé de nouvelles techniques de sélection des ensembles CDS en se basant sur l'équilibrage de charge.

Jing He et al. [26,27,28] proposent la sélection d'un *backbone* virtuel parmi les nœuds d'un réseau de capteurs en se basant sur l'équilibrage de charge et non pas sur la minimisation de taille comme critère essentiel de sélection ce qui a permis d'améliorer nettement la durée de vie du réseau entier [26]. Ils utilisent le critère d'équilibrage de charge pour définir une technique de choix d'un CDS. En effet, les auteurs divisent le problème de LB-CDS (Laod Balancing CDS) en deux sous-problèmes [27]:

- Sélection des nœuds dominants en tenant compte de l'équilibrage de charge
- Affectation équitable des nœuds dominés aux nœuds dominants pour garantir, au maximum, le critère d'équilibrage de charge.

Les mêmes auteurs évoquent les algorithmes génétiques pour faire la sélection d'un LB-CDS [28]. Ils ont montré que le critère d'équilibrage de charge lors de sélection des CDS est plus important que la minimisation de sa taille afin d'augmenter la longévité du réseau.

2.5. Conclusion

Dans ce chapitre, on a cité et classé les principales méthodes de diffusion dans les RCSF. De plus, une analyse a été réalisée pour montrer les points communs de ces méthodes et les points de différences. En outre, on a tenu compte de l'évolution chronologique de certaines techniques de diffusion.

Toutefois, le concept de minimisation, a été trop souvent étudié afin d'augmenter la longévité du réseau. Ce problème est un problème NP-Complet qui est équivalent pour presque toutes le méthodes de sélection des *backbones* dans les RCSF. Récemment, des travaux ont montré que l'équilibrage de charge est plus important que la minimisation car celle-ci peut amener à des épuisements rapides des nœuds dans le réseau.

Enfin, on peut déduire que la diffusion hybride regroupant plusieurs techniques peut être envisagée et améliorée. Ainsi une diffusion pourrait regrouper le relai, les ensemble CDS et le *clustering*. On peut améliorer également les méthodes de diffusion pour prendre en compte les facteurs de communication dans un monde réel, l'équilibrage de charge et la tolérance aux fautes.

Deuxième Partie:

Contributions

Chapitre 3

Diffusion avec une couche physique réaliste

Résumé:

Ce chapitre présente notre première contribution. A partir d'une méthode de diffusion basée sur le relai et conçue pour un environnement idéal, on l'a modifié pour la rendre applicable à une couche physique réaliste. Cette idée est inspirée de [2,19,35].

3.1. Introduction

La plupart des méthodes existantes de diffusion dans les RCSF considèrent un environnement idéal, en se basant, sur le fait qu'un nœud-capteur récepteur va certainement acquérir des données envoyées par un autre nœud-capteur émetteur s'il est dans son rayon de couverture. Cette supposition est modélisée fréquemment par la transmission dans un modèle disque unitaire (UDM: Unit Disque Model). Le modèle UDM signifie que la probabilité de réception entre deux nœuds est supérieure à 0 si la distance entre eux ne dépasse pas leur rayon de transmission. Récemment, certains chercheurs ont proposé un nouveau modèle de transmission plus proche de la réalité (LNM: LogNormal Model). Ce modèle tient compte de la plupart des facteurs influant sur la communication et il a été utilisé dans [19] et [35] pour donner un aspect plus réaliste à la méthode MPR [53]. Il a été utilisé aussi dans [67] pour définir une nouvelle méthode de diffusion avec une couche physique réaliste. L'utilisation du LNM dans certaines méthodes et algorithmes donne un aspect réaliste et avec des résultats plus cohérents. En effet, cette méthode définit une probabilité de réception entre les nœuds du réseau prenant en compte la distance entre les entités communicantes, le rayon de transmission du nœud émetteur et le facteur d'atténuation du signal. En plus, le modèle LNM décrit la probabilité de réception même si le récepteur dépasse le rayon de transmission de l'émetteur. Cette probabilité est égale à zéro seulement si le récepteur est situé à une distance double rayon de transmission de l'émetteur.

Dans ce chapitre, on se propose d'étudier la méthode DS-MPR, utilisée pour déterminer l'ensemble dominant connexe à travers le relai multipoint et conçue dans un modèle UDM. Ensuite, on teste cette méthode dans un modèle LNM pour voir ses limites. Dans la section 3 de

ce chapitre, nous proposons une nouvelle méthode de diffusion RDS-MPR qui est une amélioration de DS-MPR dans un model LNM. Une autre amélioration, relative à l'accessibilité, est proposée dans la section 4. Cette méthode est appelée eRDS-MPR qui est une extension du RDS-MPR. Finalement, une étude comparative entre les trois méthodes est donnée dans la section 5. Cette étude a montré l'importance et l'efficacité de nos contributions.

3.2. Etude de la méthode DS-MPR

La méthode DS-MPR [2], ou calcul de l'ensemble dominant connexe en utilisant le relai multipoint, permet d'extraire un ensemble connexe de nœuds constituant un *backbone*. Ce *backbone* est utilisable pour le routage et la diffusion. Dans cette section, on présente l'idée de l'algorithme DS-MPR, suivi du modèle LNM. Enfin, on mesure l'accessibilité de DS-MPR dans les deux modèles LNM et UDM.

3.2.1. La méthode DS-MPR

La méthode DS-MPR [2] consiste à trouver un ensemble connexe de nœuds en utilisant le relai MPR [53]. Cet ensemble peut être utilisé pour relayer un message envoyé initialement du nœud de base vers tous les nœuds. La figure 3.1 représente un déploiement aléatoire de 300 nœuds dans une zone de 1 km^2 en utilisant NS-2. Le problème à résoudre est: comment envoyer un paquet de données du nœud de base vers tous les nœuds tout en évitant les problèmes d'inondation aveugle

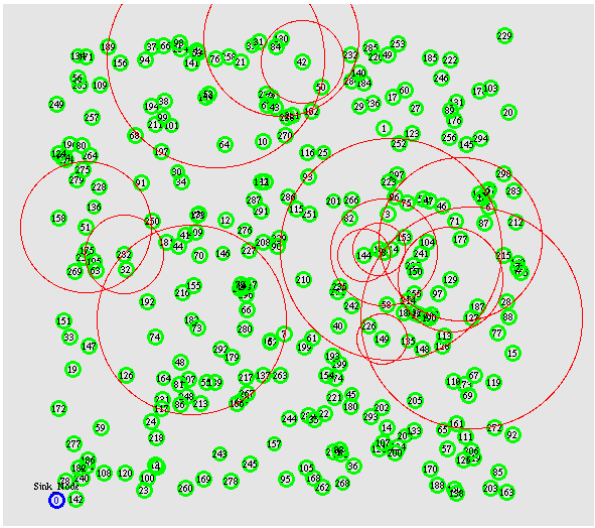


Figure 3.1. *Déploiement de 300 nœuds dans une zone de 1km²*

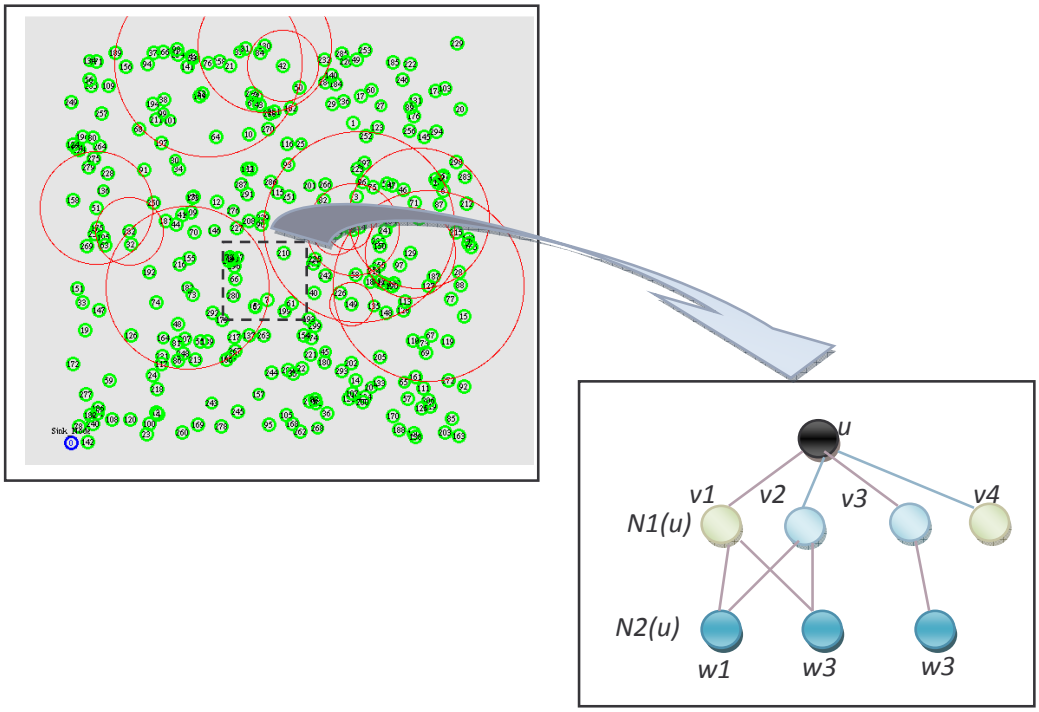


Figure 3.2. *Principe de la diffusion basée sur le relai*

En partant du nœud *u*, à chaque niveau, DS-MPR permet de choisir les meilleurs nœuds pour relayer le message. Le choix des nœuds est basé sur leur poids qui est une fonction de :

- L'énergie restante: niveau d'énergie de la batterie du capteur.
- Le degré: nombre de voisinages du nœud dans le niveau suivant.

Les nœuds de relai sont choisis au niveau *N1(u)*. Ils peuvent couvrir par la suite tous les nœuds du *N2(u)*. L'algorithme commence par les nœuds imposés, qui sont les nœuds possédant un seul parent, ils ont donc une seule source pour recevoir le message en cours de diffusion. Ensuite, le choix du reste des nœuds est effectué.

On présente dans le paragraphe suivant un modèle de communication qui a été utilisé pour décrire une communication avec une couche physique réelle.

3.2.2. Modèle LNM

Ce modèle a été introduit par [30] et a été utilisé dans [19] et [35]. LNM est représenté par la fonction (1) suivante:

$$P(x) = \begin{cases} 1 - \frac{(\frac{x}{R_c})^{2\alpha}}{2} & if \quad 0 \prec x \preceq R_c \\ \frac{(\frac{2R_c - x}{R_c})^{2\alpha}}{2} & if \quad R_c \prec x \preceq 2R_c \\ 0 & otherwise \end{cases} \tag{1}$$

Avec :

- x: Distance entre l'émetteur et le récepteur.
- R_C: Rayon de communication de l'émetteur.
- α: Facteur d'atténuation du signal.

Une comparaison entre le modèle UDM et LNM pour α=1 et R_C =25 ensuite R_C =30 est donnée dans la figure 3.4.

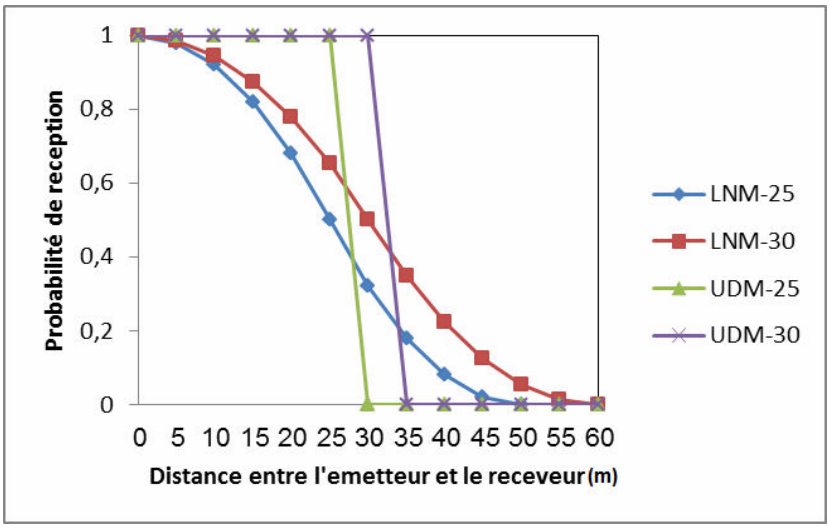


Figure 3.3. *Comparaison de probabilité de réception entre LNM et UDM pour différents rayons de transmission*

D'après la courbe de la figure 3.3, on remarque que la probabilité de réception est égale à 1 jusqu'à une distance ne dépassant pas le rayon de transmission de l'émetteur et zéro au-delà de cette distance pour le model UDM. Ce n'est pas le cas du modèle LNM, où cette probabilité commence par 1 et décroit suivant l'accroissement de la distance entre les entités communicantes. Ce résultat parait logique et proche de la réalité. Pour cela, on adopte ce modèle

dans la suite de ce chapitre. Dans le paragraphe suivant on se propose de tester la méthode DS-MPR avec une couche physique décrite par le modèle LNM.

3.2.3. Mesure de l'accessibilité de DS-MPR dans le modèle LNM

Afin de mesurer l'accessibilité de la méthode DS-MPR en utilisant le modèle LNM, on choisit un nombre de nœuds qui varie entre *16m* et *21m*. Dans le tableau 3.1 on décrit notre choix des nœuds dans la zone de dimension *250m* × *250m* avec des nœuds identiques de rayon *25m* et formant un graphe connexe.

Tableau 3.1. *Choix du nombre de nœuds selon les distances entre eux*

	D=21	*D=20*	*D=19*	*D=18*	*D=17*	*D=16*
Nombre de nœuds sur la ligne=*250/2D*, on suppose que la distance entre les nœuds est la même.	5,95	6,25	6,57	6,94	7,35	7,81
Nombre de nœuds dans la surface = $(250/2D)^2$	35.43	39.06	43,28	48,22	54,06	61,03
Nombre de nœuds choisis	6×5=30	6×6=36	6×7=42	7×7=49	7×8=56	8×8=64

Le tableau précédent décrit notre manière de choisir le nombre de nœuds dans la zone, la figure 3.4 contient la courbe représentant la moyenne des distances entre les nœuds selon leur rayon de communication. Le même nombre de nœuds va être utilisés par la suite dans toutes les simulations de ce chapitre.

La figure 3.5 représente la mesure de l'accessibilité de la méthode DS-MPR dans les deux modèles. Au niveau théorique, l'accessibilité de DS-MPR, avec une probabilité de réception toujours égale à 1, équivaut à un taux de réussite égal à 100% si la distance entre les entités communicantes ne dépasse pas leur rayon de communication. Mais au niveau pratique avec LNM, le taux de réussite de DS-MPR, qui est la moyenne de 10 simulations de cette méthode avec NS-2, commence par une valeur inférieure á 60% pour un nombre de nœuds égal à 30 et au maximum 81% pour un nombre de nœuds égal à 64.

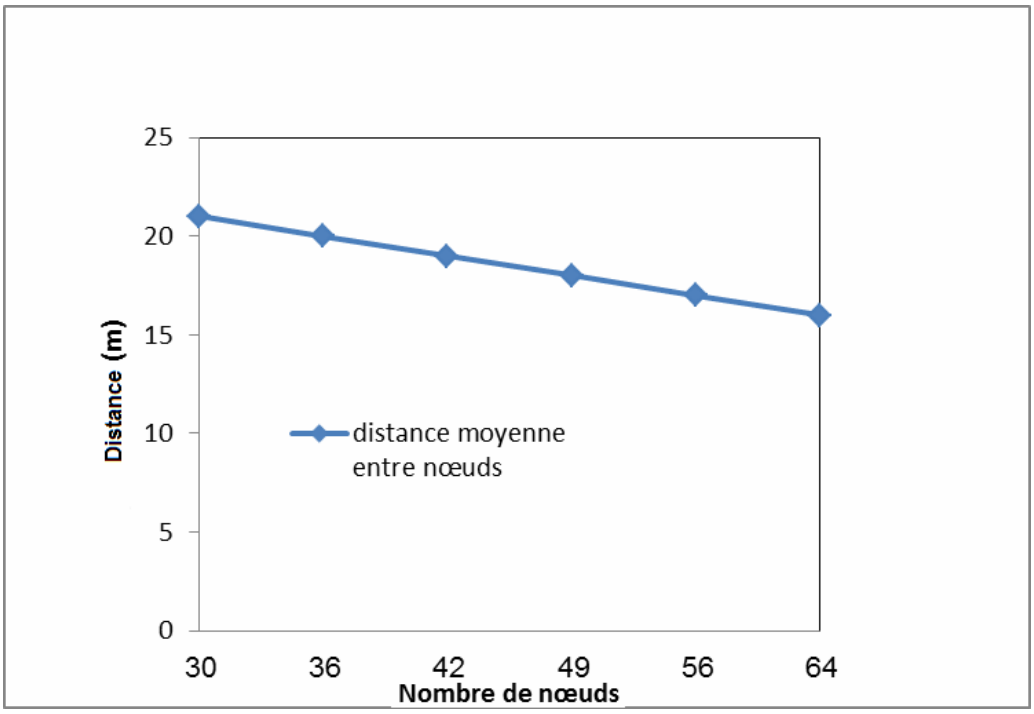


Figure 3.4. *La distance moyenne entre les nœuds selon leur rayon de communication*

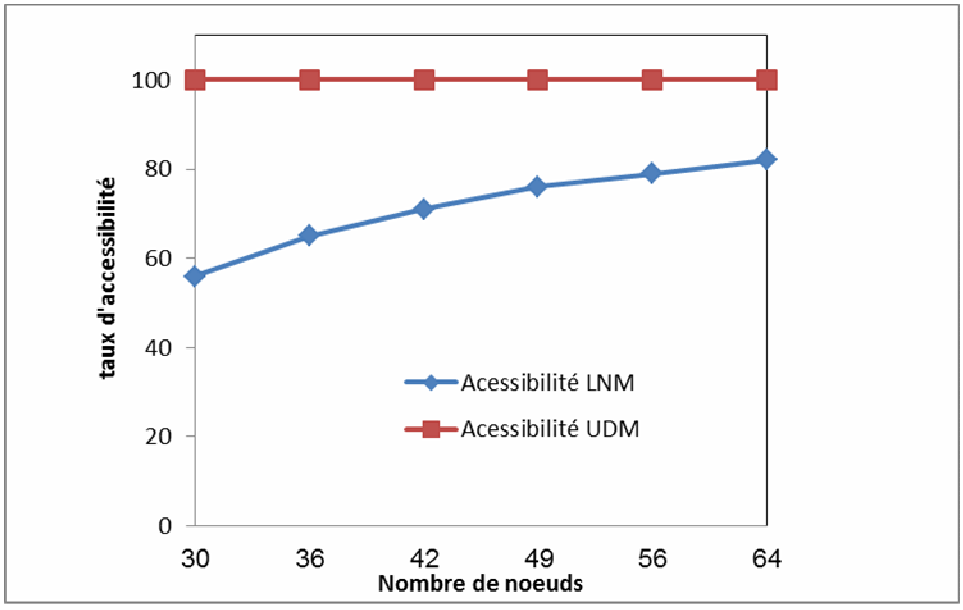


Figure 3.5. *Comparaison de l'accessibilité de la méthode DS-MPR dans les deux modèles LNM & UDM*

Le but de notre travail est de définir une amélioration de DS-MPR pour élever au maximum cette insuffisante accessibilité. Dans le paragraphe suivant, on donne notre première contribution pour résoudre ce problème.

3.3. Contribution 1: RDS-MPR

Dans cette section, on présente RDS-MPR (Realistic DS-MPR) [45,47]. Tout d'abord, la nouvelle fonction de calcul du poids des nœuds est expliquée. Ensuite, l'algorithme de notre contribution est présenté. Finalement, on donne une comparaison d'accessibilité de notre contribution par rapport à DS-MPR.

3.3.1. Poids des nœuds dans RDS-MPR

Dans le tableau 3.2, on donne une liste des symboles utilisés dans le reste de ce chapitre et leurs descriptions.

Tableau 3.2. *Symboles et descriptions*

Symbole	**Description**
G(V,E)	Graphe de connectivité représentant le réseau de capteurs, où *V* décrit l'ensemble des nœuds et *E* l'ensemble des arrêtes entre les nœuds *V*.
u	Nœud initial d'envoi
N1(u)	L'ensemble de voisinage d'ordre 1 de *u*
N2(u)	L'ensemble de voisinage d'ordre 2 de *u*
N3(u)	L'ensemble de voisinage d'ordre 3 de *u*
v	Un élément de *N1(u)*
w	Un élément de *N2(u)* ou *N1(v)*
x	Un élément de *N3(u)* ou *N2(v)* ou *N1(w)*

S(u)	Sous-ensemble de *N1(u)* choisi pour faire le relai
S1(u)	Sous-ensemble de *N2(u)* non encore couvert par les nœuds dans *S(u)*

Dans notre contribution, on définit une nouvelle fonction du poids *RW* des nœuds intégrant trois critères (2):

- Le degré
- L'énergie restante
- La probabilité de réception entre les nœuds

Le poids *RW* est décrit par une fonction linéaire sommant les trois poids décrits précédemment après la multiplication de chacun d'eux par un facteur de pondération. La somme des facteurs de pondération est égale à 1 (3). Ces facteurs permettent de définir la priorité de chaque poids. Si ces facteurs sont égaux tous à 1/3 alors les poids ont tous la même priorité.

$$RW(u) = \alpha \times RW_{Degree}(u) + \beta \times RW_{Energy}(u) + \gamma \times RW_{Probability}(u) \quad (2)$$

Avec:

$$\alpha + \beta + \gamma = 1 \quad (3)$$

$RW_{\text{Degree}}(v)$ représente le poids du degré d'un nœud *v*. Il est égal á la cardinalité de l'ensemble des nœuds du premier voisinage de *v* non couvert par les nœuds choisis pour relayer divisé par le maximum de cardinalité de ce même ensemble pour les nœuds du premier voisinage de *u* hors de la liste de nœuds choisis pour relayer. Ce poids décrit donc, la possibilité de choisir *v* comme nœud relai vu son degré de couverture (4).

$$RW_{Degree}(v) = \frac{|N1(v) \bigcap S1(u)|}{Max\{|N1(v_i \bigcap S1(u))|; v_i \in N1(u) \setminus S(u)\}} \quad (4)$$

$RW_{\text{Energy}}(v)$ est le poids de l'énergie restante d'un nœud *v*. Il est égal á l'énergie restante du nœud *v* divisée par le maximum des énergies restantes des nœuds du premier voisinage de *u* hors

de la liste de nœuds choisis pour relayer. Ce poids décrit donc, la possibilité de choisir *v* comme nœud relai vu son énergie restante (5).

$$RW_{Energy}(v) = \frac{E(v)}{Max\{E(v_i); v_i \in N1(u) \setminus S(u)\}} \quad (5)$$

$RW_{\text{Probability}}(v)$ est le poids de probabilité de réception d'un nœud *v*. Il est égal à la probabilité de réception du nœud *v* notée *q(v)* divisé par le maximum de cette même probabilité de réception des nœuds du premier voisinage de *u* hors de la liste de nœuds choisis pour relayer. Ce poids décrit donc, la possibilité de choisir *v* comme nœud relai vu sa probabilité de réception (6).

$$RW_{Probability}(v) = \frac{q(v)}{Max\{q(v_i); v_i \in N1(u) \setminus S(u)\}} \quad (6)$$

La probabilité *q(v)* d'un nœud *v* est la probabilité de réception, selon le modèle LNM, de *u* vers *v* noté *p(u,v)* multipliée par la valeur moyenne des probabilités de *v* vers les nœuds w_i , avec:

- *v* non encore choisi pour relayer
- w_i appartenant à la liste des nœuds du premier voisinage de *v* et non encore couverte par les nœuds déjà sélectionnés pour faire le relai.

$$q(v) = \begin{cases} p(u,v) \times \dfrac{\sum_{i=1}^{|N1(v) \bigcap S1(u)|} p(v,w_i)}{|N1(v) \bigcap S1(u)|} & if |N1(v) \bigcap S1(u)| \neq 0 \\ 0 & otherwise \end{cases} \quad (7)$$

$RW_{\text{Probability}}(v)$ est le poids de la probabilité de réception d'un nœud qui peut être écrit aussi sous la forme de (8).

$$RW_{Probability}(v) = \begin{cases} \dfrac{p(u,v) \times \dfrac{\sum_{i=1}^{|N1(v)\bigcap S1(u)|} p(v,w_i)}{|N1(v)\bigcap S1(u)|}}{Max\left[p(u,v_j) \times \dfrac{\sum_{i=1}^{|N1(v_j)\bigcap S1(u)|} p(v_j,w_i)}{|N1(v_j)\bigcap S1(u)|} ; v_j \in N1(u) \setminus S(u) \right]} & if |N1(v)\bigcap S1(u)| \neq 0 \\ 0 & otherwise \end{cases} \quad (8)$$

3.3.2. Algorithme RDS-MPR

Apres avoir défini les facteurs interagissant dans le calcul des poids des nœuds, on donne notre contribution pour la diffusion avec une couche physique réaliste. Dans l'algorithme qu'on propose on utilise une probabilité de réception notée *p0* que l'on considère comme la probabilité minimale de réception correcte c.-à-d. une réception avec une probabilité moins que *p0* est considérée fausse ou insuffisante. Dans l'algorithme RDS-MPR on utilise aussi une probabilité temporaire de bonne réception qui est supérieure à *p0*, pour garantir la bonne communication, notée *pt.*

Algorithme 3.1 : RDS-MPR

---

3. Initialement *S(u)=Ø* et *S1(u)= N2(u)*
4. Sélectionner les nœuds imposés de *S1(u)* c á d ayant un seul parent dans *N1(u),* ensuite éliminer ces nœuds de *S1(u)* et ajouter leur parent á *S(u)*
5. Tant que *S1(u)* ≠ *Ø*
   (a) Pour tous les nœuds n_i dans *N1(u)\ S(u)*

   Calculer les poids des nœuds n_i

   Pour le nœud *v* de poids max et *p(u,v)>pt*

   Ajouter *v* a *S(u)*

   Eliminer *N1(v)* du *S1(u)*

   (b) Fin Pour tous

(c) Si Accessibilité<90% Et pt>p0 Et γ<1 alors

décroitre légèrement pt et/ou accroitre légèrement γ

Si non

sort de la boucle tant que

Fin Si

3.3.3. Comparaison de l'accessibilité de RDS-MPR avec celle de DS-MPR

Pour mesurer la performance de notre contribution, on met une comparaison d'accessibilité entre les méthodes DS-MPR et RDS-MPR dans le modèle LNM avec les mêmes nombres de nœuds choisis dans les mesures précédentes dans la figure 3.6. On remarque que pour un *p0=0.6* et *γ=1/3* on a bien amélioré le taux d'accessibilité qui est devenu 71% pour un nombre de nœuds égal à 30 et 87% pour un nombre de nœuds égal à 64 donc l'accessibilité, est en moyenne, améliorée de 12.35%. Cette amélioration devient plus importante pour un *p0=0.5*et *γ=1/2,* où le taux d'accessibilité est devenu 80% pour un nombre de nœuds égal à 30 et 94% pour un nombre de nœuds égal à 64 donc l'accessibilité, est en moyenne, améliorée de 22.84%.

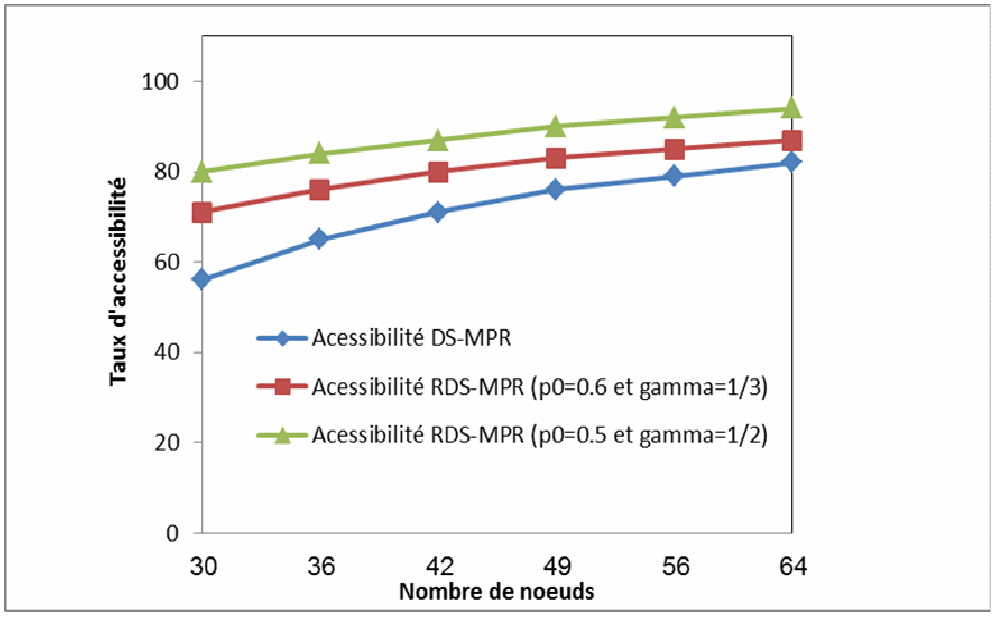


Figure 3.6. *Comparaison de l'accessibilité de la méthode DS-MPR et RDS-MPR dans le modèle LNM*

Dans la section suivante, on présente notre deuxième contribution pour la diffusion dans un environnement réaliste.

3.4. Contribution 2: eRDS-MPR

Dans cette section, on présente notre nouvelle fonction de calcul du poids des nœuds. Ensuite, on donne le nouvel algorithme. Finalement, on fait une mesure d'accessibilité de la nouvelle méthode comparée à DS-MPR et RDS-MPR.

3.4.1. Poids des nœuds dans eRDS-MPR

Pour la méthode eRDS-MPR [47] (extended Realistic DS-MPR) on réalise une sélection plus profonde c.-à-d. en tenant compte des informations des nœuds des trois niveaux au lieu de calculer le poids des nœuds de deux premiers niveaux seulement.

Cette idée est expliquée dans la figure 3.7, où on a utilisé un troisième niveau à partir de *u* pour décider des meilleurs nœuds de relai.

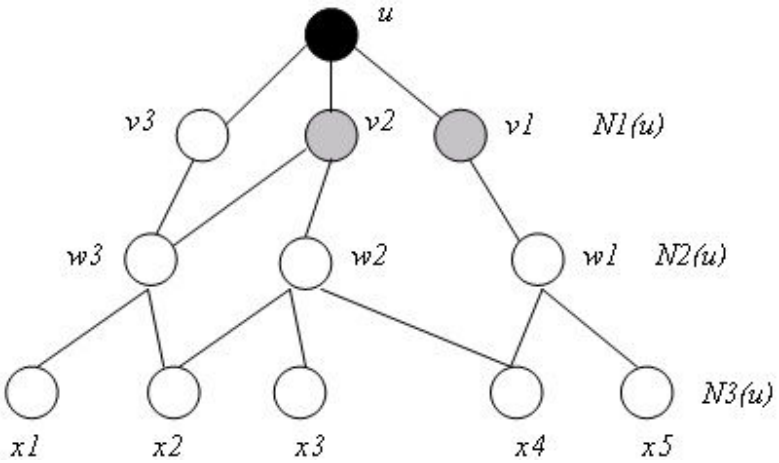


Figure 3.7. *Choix des nœuds de relai basé sur les informations a 3 niveaux*

Dans la méthode eRDS-MPR on a introduit une nouvelle fonction de calcul de poids des nœuds notée *EW* qui prend en compte:

- Le poids du degré du nœud noté EW_{Degree}.
- Le poids de l'énergie restante dans le nœud noté EW_{Energy}
- Le poids de la probabilité de réception selon les modèles LNM noté $EW_{\text{Probability}}$

Le nouveau poids des nœuds, qui est une fonction linéaire regroupant les précédents poids, est décrit par les équations (9) et (10)

$$EW(u) = \alpha \times EW_{Degree}(u) + \beta \times EW_{Energy}(u) + \gamma \times EW_{Probability}(u) \quad (9)$$

Avec:

$$\alpha + \beta + \gamma = 1 \quad (10)$$

Où:

$EW_{\text{Degree}}(v)$ est le poids du degré de couverture d'un nœud v. Il est égal au poids du degré du v noté $EW_{\text{Degree}}'(v)$ divisé par le maximum de ce même poids du degré pour les nœuds du premier voisinage de u hors de la liste des nœuds choisis pour relayer. Ce poids décrit donc, la possibilité de choisir v comme nœud relai vu son degré de couverture du niveau suivant (11).

$$EW_{Degree}(v) = \frac{EW'_{Degree}(v)}{Max\{EW'_{Degree}(v_i); v_i \in N1(u) \setminus S(u)\}} \quad (11)$$

Le poids $EW_{\text{Degree}}'(v)$ d'un nœud v est le nombre de nœuds non encore couverts par les nœuds déjà choisis pour relayer et que v peut les couvrir multiplié par la valeur moyenne des degrés des nœuds couverts par $N1(w_i)$, (12) avec :

- v non encore choisi pour relayer
- w_i appartient à la liste des nœuds du premier voisinage de v et non encore couvert par les nœuds déjà sélectionnés pour faire le relai.

$$EW'_{Degree}(v) = \begin{cases} |N1(v) \bigcap S1(u)| \times \dfrac{\sum_{i=1}^{|N1(v) \bigcap S1(u)|} |N1(w_i)|}{|N1(v) \bigcap S1(u)|} & if |N1(v) \bigcap S1(u)| \neq 0 \\ 0 & otherwise \end{cases} \quad (12)$$

$EW_{\text{Energy}}(v)$ est le poids de l'énergie restante d'un nœud v, il décrit un poids d'énergie de v noté $EW_{\text{Energy}}'(v)$ divisé par le maximum de ce même poids relatif à l'énergie restante pour les nœuds du premier voisinage de u hors de la liste des nœuds choisis pour relayer. Ce poids décrit donc, la possibilité de choisir v comme nœud relai vu son énergie restante (13).

$$EW_{Energy}(v) = \frac{EW'_{Energy}(v)}{Max\{EW'_{Energy}(v_i); v_i \in N1(u) \setminus S(u)\}} \quad (13)$$

Le poids $EW'_{\text{Energy}}(v)$ d'un nœud v est l'énergie restante de v multiplié par la valeur moyenne des énergies restantes des nœuds w_i, (14) avec:

- v non encore choisi pour relayer
- w_i appartient à la liste des nœuds du premier voisinage de v et non encore couvert par les nœuds déjà sélectionnés pour faire le relai.

$$EW'_{Energy}(v) = \begin{cases} E(v) \times \dfrac{\sum_{i=1}^{|N1(v) \bigcap S1(u)|} E(w_i)}{|N1(v) \bigcap S1(u)|} & if |N1(v) \bigcap S1(u)| \neq 0 \\ 0 & otherwise \end{cases} \quad (14)$$

$EW_{\text{Probability}}(v)$ est le poids de la probabilité de réception d'un nœud v. Il est égal á la probabilité de réception du v noté $EW'_{\text{Probability}}(v)$ divisé par le maximum de cette même probabilité de réception des nœuds du premier voisinage de u hors de la liste de nœuds choisis pour relayer. Ce poids décrit donc, la possibilité de choisir v comme nœud de relai vu sa probabilité de réception (15).

$$EW_{Probability}(v) = \frac{EW'_{Probability}(v)}{Max\{EW'_{Probability}(v_i); v_i \in N1(u) \setminus S(u)\}} \quad (15)$$

Le poids de la probabilité $EW'_{\text{Probability}}(v)$ d'un nœud v est la probabilité de réception, selon le modèle LNM, de u vers v notée $p(u,v)$ multipliée par la valeur moyenne des probabilités de v vers les nœuds w_i, multipliée par la valeur moyenne des probabilités de w_i vers les nœuds x_i, avec:

- v non encore choisi pour relayer
- w_i appartient à la liste des nœuds du premier voisinage de v et non encore couverte par les nœuds déjà sélectionnés pour faire le relai.
- x_i appartient à la liste des nœuds du premier voisinage de w_i

$$EW'_{Probability}(v) = \begin{cases} p(u,v) \times \dfrac{\sum_{i=1}^{|N1(v)\bigcap S1(u)|} p(v,w_i) \times \dfrac{\sum_{j=1}^{|N1(w_i)|} p(w_i,x_j)}{|N1(w_i)|}}{|N1(v)\bigcap S1(u)|} \\ if |N1(v)\bigcap S1(u)| \times |N1(w_i)| \neq 0 \\ 0 \qquad otherwise \end{cases} \tag{16}$$

3.4.2. Algorithme eRDS-MPR

Apres avoir défini les entités interagissant dans le calcul des poids des nœuds, dans ce paragraphe on donne notre deuxième contribution pour la diffusion avec une couche physique réaliste. Dans l'algorithme qu'on propose, on utilise une probabilité de réception noté *p0* que l'on considère comme la probabilité minimale de réception correcte c.-à-d. une réception avec une probabilité inferieur á *p0* sera considérée fausse ou insuffisante. Dans l'algorithme eRDS-MPR on utilise aussi une probabilité temporaire de bonne réception qui est supérieur à *p0*, pour garantir la bonne communication, notée *pt.*

Algorithme 3.2 :e RDS-MPR

---

1. Initialement *S(u)=Ø* et *S1(u)= N2(u)*
2. Sélectionner les nœuds imposés de *S1(u)* c á d ayant un seul parent dans *N1(u),* ensuite éliminer ces nœuds de *S1(u)* et ajouter leur parent á *S(u)*
3. Tant que *S1(u)* ≠ *Ø*
   (a) Pour tous les nœuds n_i dans *N1(u)\ S(u)*
   Calculer les poids des nœuds n_i
   Pour le nœud *v* du poids max et $p(u,v)>pt$ et $p(v,w_i)>pt$
   Ajouter *v* a *S(u)*
   Eliminer *N1(v)* de *S1(u)*
   (b) Fin Pour tous
   (c) Si Accessibilité<90% et pt>p0 et γ<1 alors

---

décroitre légèrement pt et/ou accroitre légèrement γ

Si non

sort de la boucle tant que

Fin Si

3.4.3. Comparaison de l'accessibilité d'eRDS-MPR à celle de RDS-MPR et DS-MPR

Pour mesurer la performance de notre contribution, dans la figure 3.8 on met une comparaison d'accessibilité entre les méthodes DS-MPR, RDS-MPR et eRDS-MPR dans le modèle LNM avec le même nombre de nœuds choisis dans les mesures précédentes.

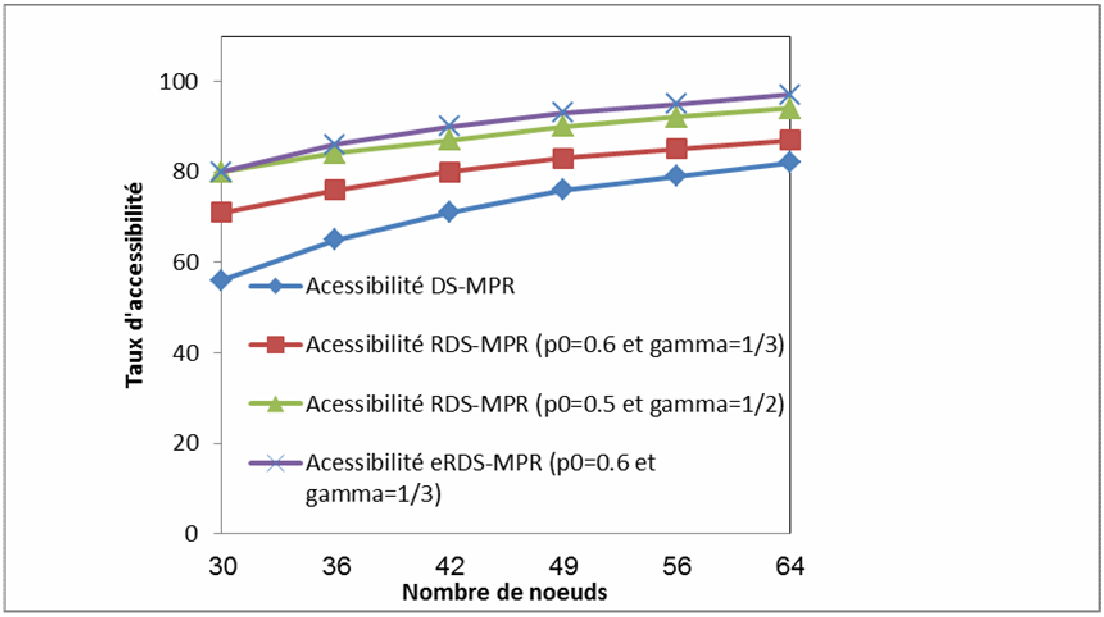


Figure 3.8. *Comparaison de l'accessibilité des méthodes DS-MPR, RDS-MPR et eRDS-MPR dans le modèle LNM*

On remarque que pour un *p0=0.6* et *γ=1/3* on a bien amélioré le taux d'accessibilité qui est devenu 71% pour un nombre de nœuds égal à 30 et 87% pour un nombre de nœuds égal à 64 donc l'accessibilité est en moyenne, améliorée de 12.35%. Cette amélioration devient plus importante pour un *p0=0.5*et *γ=1/2,* où le taux d'accessibilité est devenu 80% pour un nombre de

nœuds égal à 30 et 94% pour un nombre de nœuds égal à 64 donc l'accessibilité est en moyenne, améliorée de 22.84%.

3.5. Etude comparative

Dans cette section du chapitre, en utilisant le simulateur NS-2, on réalise une étude comparative entre DS-MPR, RDS-MPR et eRDS-MPR. Premièrement, on compare les trois heuristiques au niveau de la consommation d'énergie, et par la suite on compare leurs performances du point de vue messages échangés, temps de communication et espace mémoire utilisé pour accomplir la diffusion. Cette étude comparative est réalisée selon les paramètres donnés dans le tableau 3.3

Tableau 3.3. *Paramètres de simulation pour les trois méthodes*

Paramètre	**Valeur**
Taille du champ	250m × 250m
Nombre des nœuds	30; 36; 42; 49; 56; 64
Rayon de transmission	25m
Protocole de routage	SDR
Protocole MAC	MAC/802.11
Energie Initiale du nœud	30j
Puissance Tx	2.0 mW
Puissance Rx	0.4 mW
Taille des paquets	200b

3.5.1. Comparaison de l'énergie

La figure 3.9 donne une comparaison de l'énergie consommée dans les trois méthodes. On remarque que l'amélioration de l'accessibilité a causé une augmentation de consommation de

l'énergie. Pour DS-MPR, il n'y a pas beaucoup de perte d'énergie puisque il y a plein de nœuds inaccessibles.

Si on compare les trois courbes avec un nombre de nœuds égal à 63:

- Selon la figure 3.9, RDS-MPR et eRDS-MPR accroissent la consommation d'énergie totale pour accomplir la diffusion comparé à DS-MPR respectivement avec (1,32-1,15)/1,15= 14,7 % et (1,35-1,15)/1,15= 17.3 %.
- RDS-MPR et eRDS-MPR accroissent l'accessibilité du DS-MPR respectivement avec (94-80)/80= 17,5 % et (97-80)/80= 21.2 %.

On peut donc conclure que l'augmentation de l'accessibilité par nos contributions est réalisée avec une augmentation moins importante d'énergie.

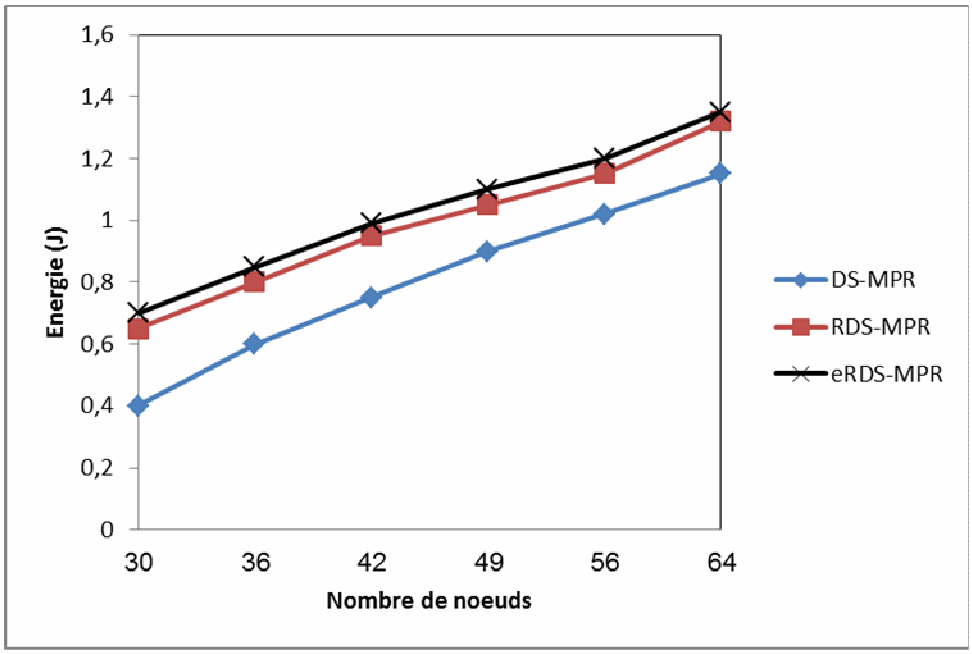


Figure 3.9. *Comparaison d'énergie totale consommée pour accomplir la diffusion des méthodes DS-MPR, RDS-MPR et eRDS-MPR.*

Pour réaliser une analyse complète de nos contributions dans le paragraphe suivant on mesure trois autres critères pour les trois méthodes.

3.5.2. Comparaison *de performance*

Si on néglige les collusions, afin de mieux raffiner la comparaison de performance entre les trois heuristiques, on présente dans ce paragraphe: une comparaison théorique de la taille de la mémoire utilisée, le nombre de messages échangés et le temps nécessaire pour faire les communications.

On donne dans le tableau 3.4, les notations utilisées dans ce paragraphe pour effectuer cette étude comparative.

Tableau 3.4 ***Notations de comparaisons***

Paramètre	Désignation
d	Degré moyen dans les nœuds du graphe du réseau
t	Temps moyen nécessaire pour réaliser une communication entre deux nœuds du réseau.
x	Taille moyenne pour stocker une information relative à un poids dans la mémoire d'un nœud du réseau.

– Pour le nombre de messages échangés (figure 3.10), il est clair que DS-MPR et RDS-MPR utilisent le même nombre puisqu'ils ont besoin seulement des informations a deux sauts du nœud sélecteur des nœuds relais. Ce nombre est multiplié par deux pour eRDS-MPR.

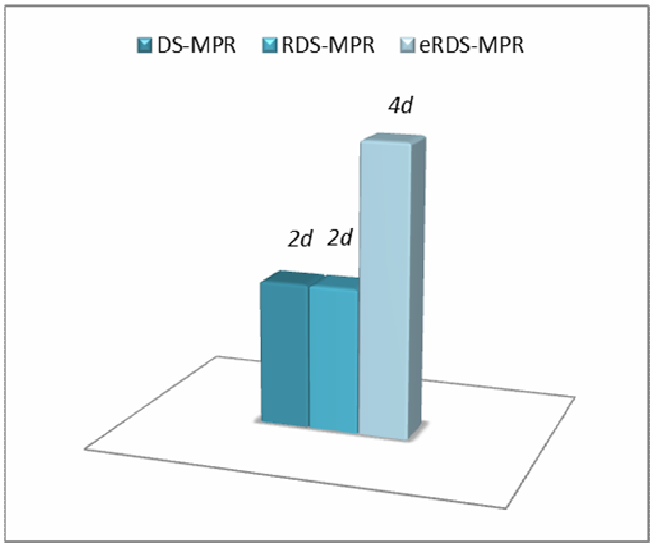


Figure 3.10. *Messages échangés*

- Pour le temps total de la communication (figure 3.11), DS-MPR et RDS-MPR ont besoin du même temps puisque ils ont besoin seulement des informations a deux sauts du nœud sélecteur des nœuds relai. Ce nombre est multiplié par deux pour eRDS-MPR.

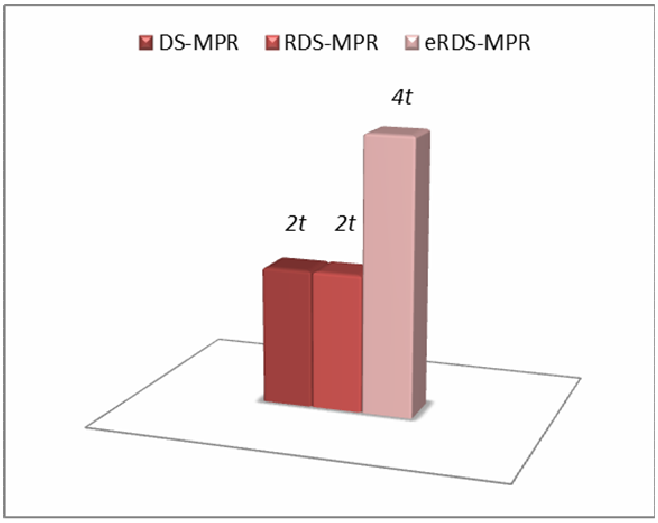


Figure 3.11. *Temps de communication*

- Pour l'espace mémoire nécessaire (figure 3.12), RDS-MPR utilise un poids de plus que DS-MPR tandis que eRDS-MPR a besoin de stoker le double des poids du RDS-MPR.

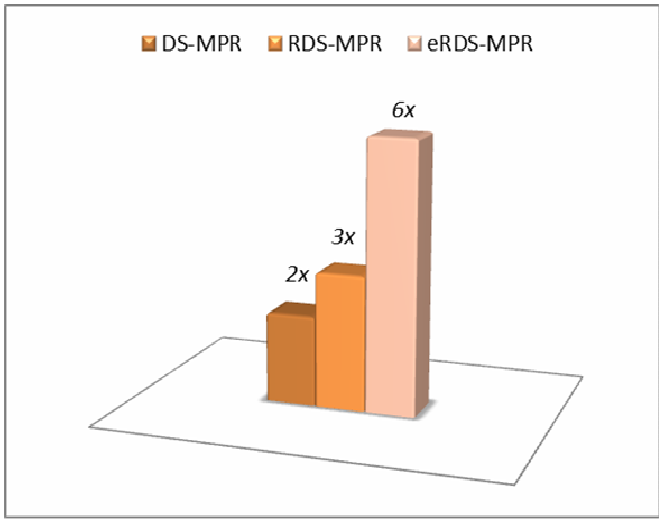


Figure 3.12. *Espace mémoire*

On peut conclure d'après cette comparaison, que malgré que eRDS-MPR fournisse le taux d'accessibilité le plus élevé allant jusqu'à 97% il est trop coûteux pour un capteur limité en ressources. Ainsi RDS-MPR parait plus adéquat en rapport accessibilité-légèreté (accessibilité =94% avec moitié des ressources utilisées dans eRDS-MPR).

3.6. Conclusion

Beaucoup d'efforts de recherche ont été réalisés pour résoudre le problème de la diffusion dans les RCSF. La but de la plupart de ces travaux est de résoudre les problèmes de congestion et de collision sans prise en compte de l'environnement de la communication qui a une grande influence sur le taux d'accessibilité des méthodes de diffusion.

Dans ce chapitre, on a étudié une méthode de diffusion conçue dans un UDM, afin de la modifier pour la rendre adéquate avec une couche physique réaliste décrite par un modèle probabiliste appelé LNM.

Le taux d'accessibilité de la méthode d'origine DS-MPR est égal à 80%, dans le meilleur des cas. Il a été amélioré nettement par nos deux contributions, RDS-MPR et eRDS-MPR respectivement 94% et 97%. Une étude comparative entre les trois méthodes a montré que bien qu'eRDS-MPR fournisse le taux d'accessibilité le plus élevé, il est trop coûteux pour un capteur limité en ressources. Ainsi RDS-MPR paraît plus adéquat en rapport accessibilité-légèreté.

Dans le chapitre suivant, on propose une amélioration des méthodes de diffusion qui sont basées sur le relai multi-saut, en particulier, RDS-MPR. Cette amélioration consiste à utiliser les moyens d'auto-organisation dans les RCSF pour éviter les réceptions redondantes qui causent une perte d'énergie lors des diffusions.

Chapitre 4

Optimisation de la diffusion multi-sauts

Résumé:

L'utilisation de la diffusion basée sur le multi-saut est la source d'une perte importante d'énergie des capteurs à cause des réceptions redondantes. De plus, ces méthodes ne tiennent pas compte de l'aspect dynamique du réseau. Ainsi, dans ce chapitre, on décrit notre deuxième contribution où on définit des méthodes d'auto-organisation dynamiques, afin de répartir les nœuds dans des clusters avant la diffusion.

4.1. Introduction

L'énergie consommée par un capteur lors des opérations de calcul interne peut être considérée comme négligeable comparée à celle nécessaire pour réaliser la communication. La minimisation des communications redondantes permet donc de prolonger remarquablement la longévité d'un capteur et donc la longévité du réseau entier.

En particulier, la réception dupliquée est l'un des problèmes de la diffusion basée sur le relai multipoints qui cause beaucoup de pertes d'énergie.

De ce fait, l'utilisation des méthodes d'auto-organisation est une solution qui permet d'isoler les capteurs dans des clusters et d'éliminer les liaisons causant les redondances.

D'autre part un RCSF peut être un réseau dynamique lorsque les capteurs sont déployés dans des zones inconnues et parfois dangereuses. Il y a donc une possibilité de disparition de certains de ces capteurs c.-à-d. rupture de certains liens dans le réseau. Il y a d'autre part une possibilité d'apparition de nouveaux capteurs donc de nouveaux liens entre les entités du réseau.

A partir de ces remarques, notre contribution consiste en un raffinement des méthodes basées sur le relai, à savoir :

- La réduction des réceptions redondantes.

- La prise en compte de l'aspect dynamique.

Cette contribution est basée sur la combinaison entre les méthodes d'auto-organisation dynamiques et les méthodes de diffusion basées sur le relai.

Dans ce chapitre, on présente tout d'abord notre proposition pour améliorer la méthode MPR afin de diffuser avec moins d'énergie. Ensuite, on définit une nouvelle méthode dite MCR : Multi Cluster Relay qui est l'amélioration de RDS-MPR, où l'on minimise la consommation d'énergie tout en tenant compte de l'aspect dynamique du réseau. Nos contributions sont illustrées en utilisant un modèle énergétique. De plus, on a utilisé 3 capteurs *Arduino* [7,5] pour voir une exécution réelle de nos algorithmes.

4.2. Amélioration de la méthode MPR

Dans cette section, on présente le modèle énergétique utilisé. Ensuite, on réalise un rappel de la méthode MPR afin de montrer son inconvénient et finalement on propose notre solution pour supprimer cet inconvénient.

4.2.1. modèle énergétique

Le modèle énergétique utilisé dans [64] est donné par les équations suivantes (1,2):

- Energie consommée lors d'une émission

$$t_i = s.(\alpha + \beta.d^2) \quad (1)$$

Avec :

α=50nj/b

β=100pj/b/m²

s : taille du message (b)

d : distance de transmission (m)

- Energie consommée lors d'une réception

$$v_i = s.\gamma \quad (2)$$

Avec :

γ=50nj/b

s : taille du message (b)

Dans la figure 4.1 on donne la comparaison entre les énergies de réception et de l'émission en utilisant équations (1) et (2) .

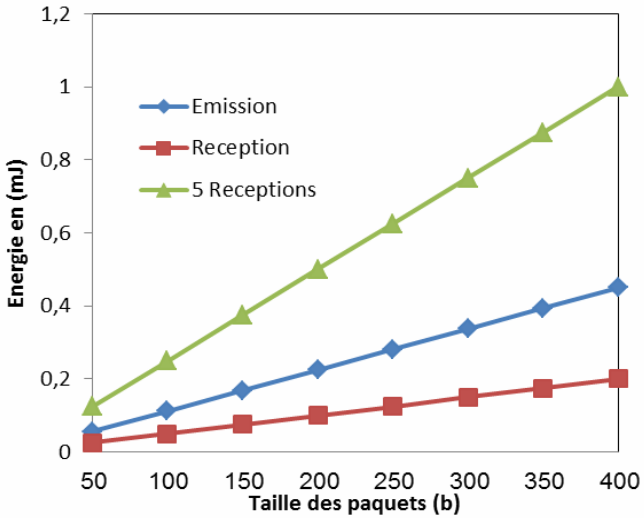


Figure 4.1. *Comparaison entre les énergies de réception et d'émission*

D'après la figure 4.1, qui représente l'énergie consommée pour l'émission, la réception selon les équations (1) et (2), on voit sans surprise que la réception redondante est une cause de perte inutile d'énergie, par conséquent réduire les redondances permet d'augmenter la durée de vie d'un capteur.

4.2.2. Rappel et inconvénient de MPR

La méthode de diffusion MPR est une méthode distribuée permettant d'accomplir la diffusion par étape. Dans chaque étape, en connaissant des informations sur ses voisinages à deux sauts, un nœud choisit ses meilleurs nœuds de relai du niveau suivant.

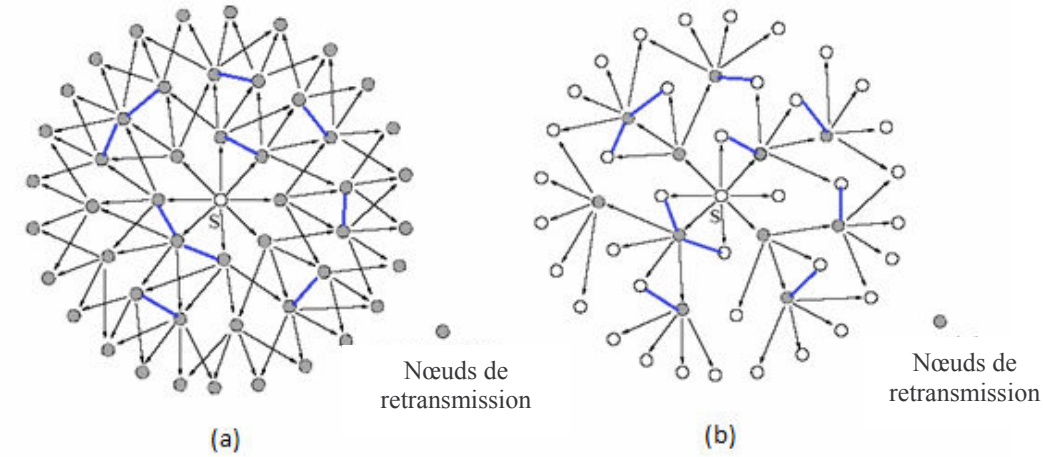


Figure 4.2. (a) *Inondation aveugle*, (b) *Inondation MPR*

Dans la figure 4.2 (a), en réalisant une inondation aveugle (*blind flooding*) tous les nœuds vont retransmettre le message initialement envoyé par *S*.

Dans la figure 4.2 (b), on applique la diffusion MPR où le nombre de nœuds de relai est beaucoup plus faible que lors d'une inondation aveugle.

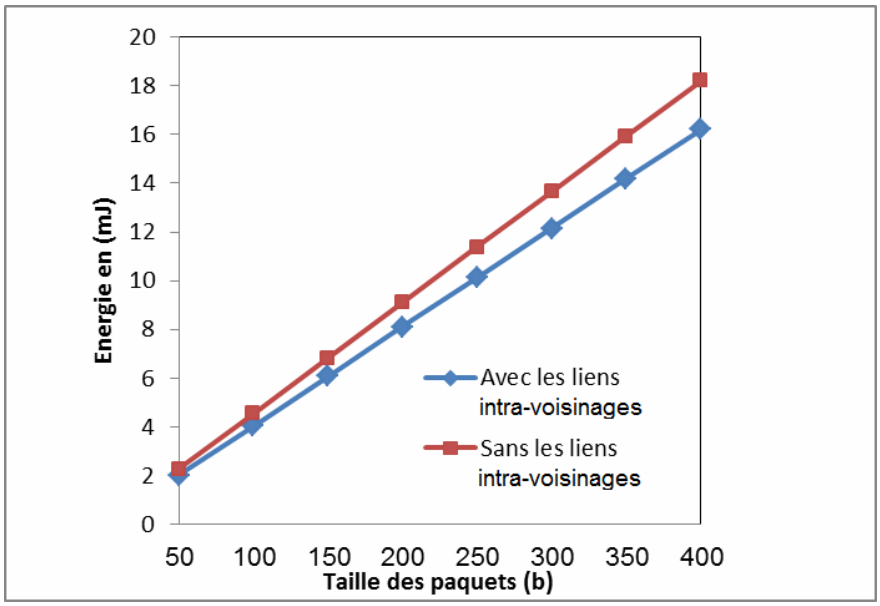


Figure 4.3. *Energie de diffusion consommée dans les réseaux de la figure.4.2 avec et sans les liens intra-voisinages*

Malgré le grand apport de la méthode de diffusion multi-saut MPR, elle possède quelques lacunes, puisque:

- Elle ne tient pas compte des liens entre les nœuds du même niveau (liens intra-voisinages en bleu dans la figure 4.2).
- Un nœud peut recevoir plus d'une fois le même message.

Ces deux problèmes causent beaucoup de réceptions redondantes et par conséquent une perte d'énergie. La figure 4.3 décrit l'énergie consommée avec et sans les liens en bleu (intra-niveau), on remarque que la réduction des redondances permet d'éviter une perte de 12.34% d'énergie totale pour réaliser la diffusion.

Dans le paragraphe suivant on se propose de résoudre ce problème en utilisant une nouvelle méthode d'auto-organisation.

4.2.3. Amélioration proposée

Pour éviter les réceptions redondantes on propose une nouvelle méthode d'auto-organisation en vue d'isoler les nœuds dans des clusters suivant l'algorithme 4.1 [48]. En effet, notre méthode sélectionne le nœud de base et les nœuds relais choisis par la méthode MPR comme des *cluster-heads*. Ensuite, chaque autre nœud est identifié comme *cluster-member*. Ensuite, il est affecté à un et un seul *cluster-head* en respectant l'équilibrage de charge entre les différents *cluster-heads*.

Tableau 4.1. ***Liste des symboles pour la méthode de construction des groupes***

Symbole	**Description**
G(V,E)	Graphe de connectivité représentant le réseau de capteurs, où *V* représente l'ensemble des nœuds et *E* l'ensemble des arrêtes entre les nœuds *V*.
Sink	Nœud initial d'envoi
Root	Identifiant du nœud initial d'envoi

Gc(Vc,Ec)	Graphe de connectivité représentant le réseau de capteurs après l'auto-organisation, où *Vc* représente l'ensemble des nœuds et *Ec* l'ensemble des arrêtes entre les nœuds *Vc*.
Ch	Identifiant du *cluster-head*
Leaf	Identifiant d'un membre du groupe
Temp	Ensemble temporaire des *cluster-head*s

Algorithme 4.1 : Construction des groupes

L'identifiant de nœud du base est *root*, ensuite ajouter *root* et les nœuds de relai choisis par l'application de la méthode MPR á l'ensemble des nœuds *Vc* du graphe auto-organisé.

1: id(sink) ← root

2: Vc← Vc U {root}

3: Vc←Vc U MPR(G)

L'identifiant des nœuds dans *Vc* autre que le sink, est *cluster-head* "ch"

4: **Pour tout** noeud n dans Vc\ {sink} **faire**

5: id(n)←ch

6: **Fin Pour tout**

Le reste des nœuds de graphe est considéré comme des feuilles, si une feuille peut se connecter à plus d'un *cluster-head*, elle garde la connexion avec le capteur d'énergie maximale.

7: Temp← Ø

8: **Pour tout** noeud n dans G \ Gc et ch dans Gc **faire**

9: **Si** (n,ch) appartient à E **Alors**

10: Temp← Temp U {ch}

11: **Fin Si**

12: choisir ch dans Temp ayant le maximum d'énergie

13: id(n)←leaf

14: Ec←Ec U (n,ch)

15: Temp← Ø

16: **Fin Pour tous**

L'algorithme 4.1 permet de faire une organisation du réseau sous forme d'un ensemble de clusters ce qui permet d'éviter les liens dupliqués entre les nœuds. Cette organisation est réalisée en utilisant la méthode MPR lors du choix des *cluster-head*s, ensuite chaque nœud garde la connexion avec un et un seul *cluster-head*. Par conséquent, il recevra un message diffusé par le nœud de base une et une seule fois. Le résultat de cette organisation est un *backbone* formé par le nœud de base « *sink* » et les *cluster-head*s «*ch*».

Dans la figure 4.4, on applique l'algorithme précédent á un réseau de capteurs. On remarque un *backbone* formé par le nœud de base et les nœuds de relai. En plus on a bien éliminé les liens dupliqués c.-à-d. chaque nœud feuille est connecté à un seul nœud de relai.

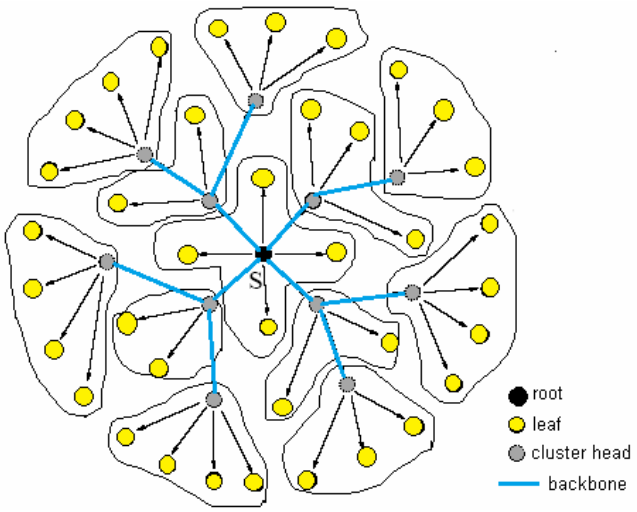


Figure 4.4. *Organisation du réseau sous forme de cluster*

Apres l'application de l'algorithme d'organisation ci-après l'algorithme de diffusion est donné [48].

Algorithme 4.2 : 3B Backbone Bazed Broadcasting

1 : Appliquer l'algorithme 4.1 pour organiser le réseau

2: Créer le *backbone* formé par le nœud de base *sink* et les *cluster-head ch*

3 : Faire la diffusion du *sink* vers le réseau en utilisant le *backbone* crée dans 2

4.2.4. Etude comparative

Afin de montrer l'apport de notre méthode, en utilisant le modèle énergétique décrit précédemment, on mesure l'énergie consommée en utilisant la méthode MPR et on la compare avec celle consommée en utilisant la méthode 3B. Dans les figures 4.5 et 4.6 on réalise cette mesure avec des tailles différentes de paquets et avec des densités de 150 nœuds et de 250 nœuds. Cette comparaison montre que le pourcentage d'énergie consommée peut être réduite de 17% et 23% pour des réseaux de tailles respectives 150 et 250 nœuds.

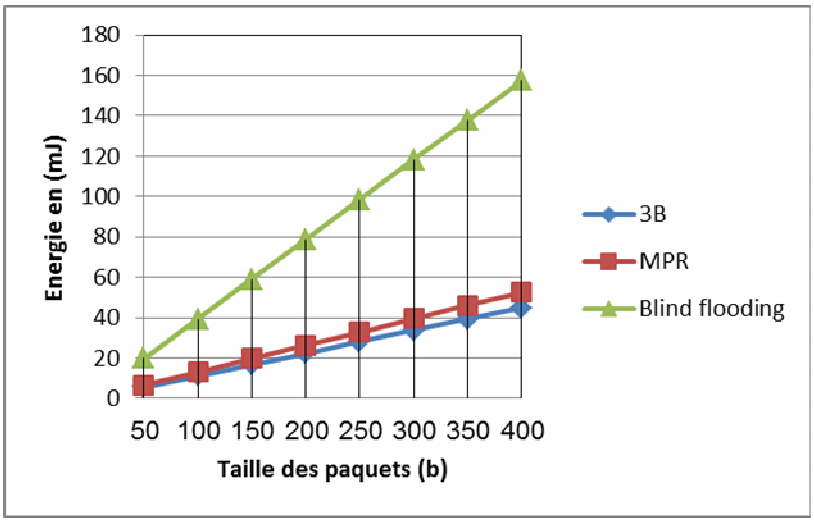


Figure 4.5. *Comparaison de l'énergie totale de diffusion pour un réseau de 150 nœuds*

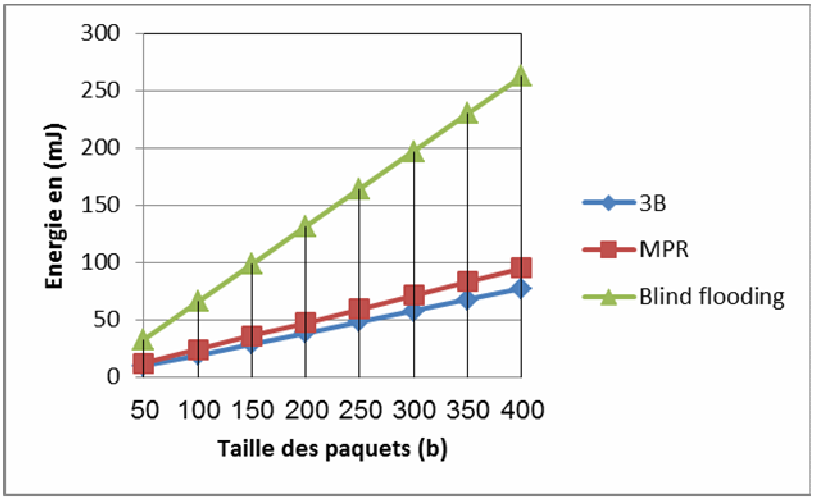


Figure 4.6. *Comparaison de l'énergie totale de diffusion pour un réseau de 250 nœuds*

Dans le paragraphe suivant, on donne notre deuxième contribution pour l'optimisation de la diffusion multi-saut dans un environnement réaliste et en tenant compte de l'aspect dynamique d'un RCSF.

4.3. Amélioration de la méthode RDS-MPR

4.3.1. Rappel sur RDS-MPR

La méthode RDS-MPR [47] de calcul de l'ensemble dominant connexe en utilisant le relai multipoint, permet d'extraire un ensemble connexe de nœuds constituant un *backbone*. Ce *backbone* est utilisable pour le routage et la diffusion. La sélection des nœuds de relai se fait par étape. Dans chaque étape, un nœud choisit ses meilleurs nœuds de relai du niveau suivant en connaissant des informations sur son voisinage à deux sauts.

La sélection des nœuds de relai est basée sur une fonction du poids *RW* des nœuds définit par trois paramètres (2):

- Le poids du degré,
- Le poids de l'énergie restante,
- Le poids de la probabilité de réception entre les nœuds.

$$RW(u) = \alpha \times RW_{Degree}(u) + \beta \times RW_{Energy}(u) + \gamma \times RW_{Probability}(u) \quad (2)$$

Avec:

$$\alpha + \beta + \gamma = 1 \quad (3)$$

Si on applique RDS-MPR au réseau de la figure 4.7 les nœuds de relai seront *v1,v2* et *v4* l'ensemble dominant connexe est *{u,v1,v2,v3}*. La méthode RDS-MPR ne tient pas compte des liens entre les nœuds du même niveau qui causent des réceptions redondantes causant une consommation inutile d'énergie. Dans le tableau 4.2 on se propose de compter le nombre total de réceptions d'un paquet diffusé une seule fois de la part du nœud *u* afin d'extraire le nombre total de réceptions redondantes causant une consommation inutile d'énergie.

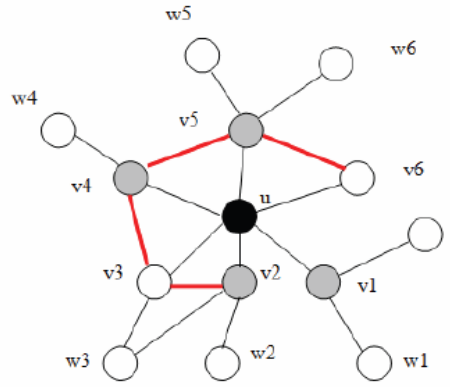


Figure 4.7. *Choix des nœuds de relai par RDS-MPR*

Tableau 4.2. *Nombre total de réceptions*

Nœud	**Nombre total de réceptions du même message envoyé par** *u*	**Provenance du message**
v1	1	*u*
v2	1	*u*
v3	3	*u,v2,v4*

v4	2	*u,v5*
v5	2	*u,v4*
v6	2	*u,v5*

La figure 4.8 décrit une comparaison de l'énergie totale pour réaliser la diffusion d'un paquet de taille comprise entre 50 et 400 octets en utilisant la méthode RDS-MPR puis en utilisant la méthode RDS-MPR avec élimination des liens intra-voisinage. On remarque d'après cette comparaison que l'élimination des liens intra-voisinage, cause des réceptions inutiles, a réduit l'énergie totale de 20% pour réaliser la diffusion (figure 4.8).

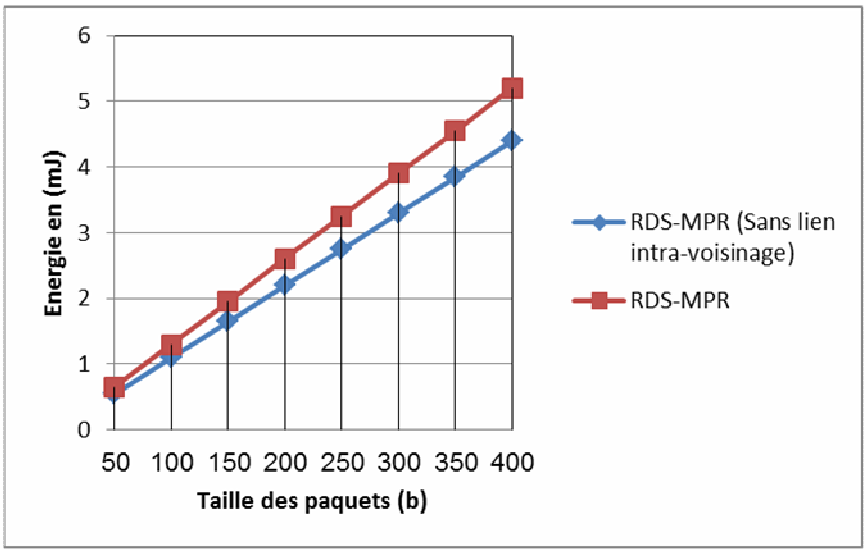


Figure 4.8. *Energie totale consommée de diffusion dans le réseau de la figure.4.7 en appliquant RDS-MPR avec et sans liens intra-voisinage*

Ainsi, l'insuffisance de la méthode RDS-MPR, comme pour des autres méthodes basées sur le relai, réside dans le fait qu'elles ne tiennent pas compte des facteurs suivants:

- Les liens entre les nœuds de même niveau par rapport au nœud initiateur de le diffusion,
- Les réceptions redondantes,
- L'aspect dynamique du RCSF.

Notre amélioration de RDS-MPR est réalisée en deux phases :

Phase1 : Organiser le réseau en clusters qui prend en considération la présence ou l'absence d'un nœud, (algorithmes SCA : Self-Clustering Algorithm, algorithmes: node-move-in et node-move-out).

Phase2 : Réaliser la diffusion en utilisant la méthode MCR Multi-Cluster Relay à travers le réseau auto-organisé dans la phase 1.

Cette méthode sera définie dans l'algorithme 4.6.

4.3.2. SCA : une nouvelle méthode d'auto-organisation

Dans le tableau 4.2, on définit la liste des symboles utilisés dans l'algorithme SCA [46].

Tableau 4.3. *liste des symboles dans SCA*

Symbole	**Description**
G(V,E)	Graphe de connectivité représentant le réseau de capteurs, où *V* décrit l'ensemble des nœuds et *E* l'ensemble des arrêtes entre les nœuds *V*.
u	Nœud initial d'envoi
N1(u)	L'ensemble de voisinage d'ordre 1 de *u*
N2(u)	L'ensemble de voisinage d'ordre 2 de *u*
v	Un élément de *N1(u)*
w	Un élément de *N2(u)* ou *N1(v)*
r	Identifiant du nœud initial d'envoi
n_t	Identifiant temporaire d'un nœud
n_n	Identifiant d'un nouveau nœud à ajouter dans le réseau

n_i	Identifiant d'un nœud intermédiaire
n_l	Identifiant d'un nœud feuille ou membre d'un cluster
n_{ch}	Identifiant d'un nœud *cluster-head*
n_g	Identifiant d'un nœud passerelle entre deux clusters
$G_{cluster}$	Sous-graphe de G organisé en clusters
G_{temp}	Sous-graphe de $G_{cluster}$ où n_n peut être connecté

Après avoir donné la liste des symboles et leurs descriptions, on donne notre algorithme d'auto-organisation des RCSF. La méthode SCA est distribuée et chaque nœud dans le réseau détermine lui-même son identifiant et son *cluster-head* proposé. L'auto-organisation commence par le nœud initial de diffusion en lui donnant r comme identifiant ensuite pour chaque nœud connecté à r, son identifiant sera n_i. Le graphe auto-organisé est $G_{cluster}$ qui est initialement formé par r et les n_i. Ensuite, pour tous les autres nœuds n_n dans G, le sous graphe où on peut connecter n_n sera nommée G_{temp}. Ensuite, on calcule le poids des nœuds dans G_{temp} selon les équations données ci-après et on extrait n_t de G_{temp} qui a le poids maximal. Finalement, n_n sera connecté à n_t, et aura un identifient selon la figure 4.9.

Le poids des nœuds dans G_{temp} est calculé selon l'énergie restante dans le nœud et la probabilité de réception entre le nouveau nœud á ajouter et les nœuds dans G_{temp}. Ce poids est défini dans les équations ci-après.

$$W(v) = \alpha \times W_{Energy}(v) + (1 - \alpha) \times W_{Probability}(v) \qquad (4)$$

Avec $\alpha \leq 1$, qui est un facteur de priorité entre le poids d'énergie et le poids de probabilité c.à.d. dans le cas où $\alpha=1/2$, la priorité des poids est équitable.

$$W_{Energy}(v) = \frac{E(v)}{Max(E(v_i); v_i \in G_{temp})} \qquad (5)$$

$$W_{Probability}(v) = \frac{p(v, n_n)}{Max(p(v_i, n_n); v_i \in G_{temp})} \qquad (6)$$

Où :

- *E(v)* est l'énergie restante du nœud *v*.
- *p(u,v)* est la probabilité de réception entre u et v selon le modèle LNM.

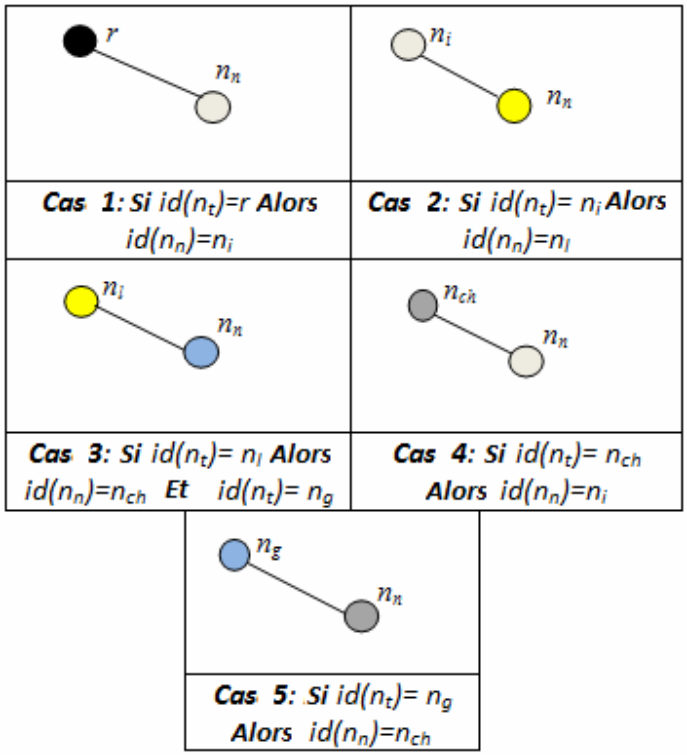


Figure 4.9. *Différents cas d'identification d'un nouveau nœud*

Algorithme 4.3 : SCA :Self-Clustering Algorithm

1: Initialement : $G_{cluster}$ est vide

2: Appeler *r* l'identifiant du nœud diffuseur et l'ajouter á $G_{cluster}$

3: **Pour tout** nœud n_n qui n'a pas d'identifiant **faire**

4: Ajouter les nœuds n de $G_{cluster}$ á G_{temp} si (n_n, n) est une arrête de *G*

5 : Calculer le poids des nœuds dans G_{temp}

6 : Choisir n_t dans G_{temp} qui a le poids maximal

---

7 : Ajouter n_n aux nœuds et (n_n, n) aux arrêtes de $G_{cluster}$

8 : **Selon** (id(n_l)) **Faire**

9 : **cas** r : id(n_n) = n_i

10 : **cas** n_i : id(n_n) = n_l

11 : **cas** n_l : id(n_n) = n_{ch} and (id(n_l)) = n_g

12 : **cas** n_{ch} : id(n_n) = n_i

13 : **cas** n_g : id(n_n) = n_{ch}

14 : **Fin Selon**

15 : Vider G_{temp}

16 : **Fin Pour tout**

17: **Pour tout** nœud n_1 et n_2 dans le même cluster **faire**

18: **Si** id(n_1)=n_i **et** [id(n_2)= n_l **ou** id(n_2)= n_g] **et** [(n_1,n_2) est une arrête de G] **alors**

19 : ajouter (n_1,n_2) á $G_{cluster}$

20 : **Fin Si**

21 : **Fin Pour tout**

Les étapes 17,18,19 et 20 de l'algorithme précédent sont très importantes afin d'appliquer la méthode RDS-MPR á l'intérieur des clusters. En effet, les liens entre les nœuds intermédiaires et les feuilles initialement présentes dans le graphe et qui sont dans le même cluster, sont ajoutés au graphe $G_{cluster}$.

Une illustration de notre algorithme d'auto-organisation est donnée dans la figure 4.10 ci-après.

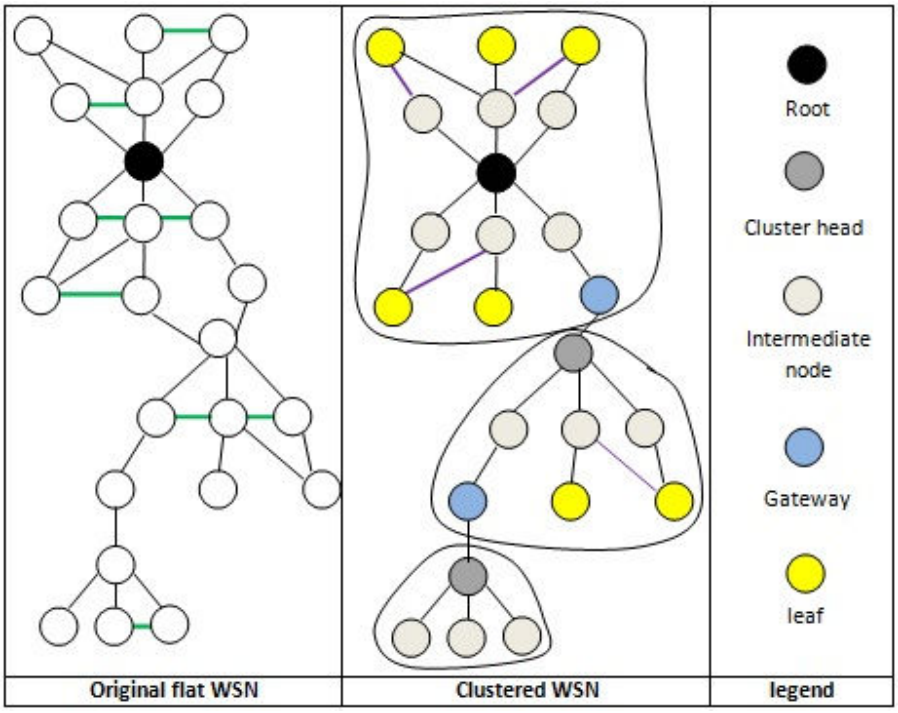


Figure 4.10. *Application de SCA à un réseau: les liens intra-niveaux du réseau d'origine sont éliminés mais les liens d'un niveau à l'autre dans le même cluster sont conservés.*

Pour tenir compte de la présence ou l'absence d'un nœud, on donne ci-dessous deux algorithmes *node-move-out* et *node-move-in* pour gérer respectivement la sortie et l'arrivée des nœuds.

Algorithme 4.4 : node-move-in

1: Appeler n_n le nœud à ajouter à $G_{cluster}$

2: Ajouter les nœuds n de $G_{cluster}$ à G_{temp} où n_n peut se connecter

3 : Calculer le poids des nœuds dans G_{temp}

4 : Choisir n_t dans G_{temp} qui a le poids maximal

5 : Ajouter n_n aux nœuds et (n_n, n) aux arrêtes de $G_{cluster}$

6 : **Selon** (id(n_t)) **Faire**

---

7 : **cas** r : id(n_n) = n_i

8 : **cas** n_i : id(n_n) = n_l

9 : **cas** n_l : id(n_n) = n_{ch} **Et** (id(n_l)) = n_g

10 : **cas** n_{ch} : id(n_n) = n_i

11 : **cas** n_g : id(n_n) = n_{ch}

12 : **Fin Selon**

L'ajout d'un nouveau nœud au réseau est facile à mettre en œuvre car c'est une continuité de la construction des clusters.

Pour la gestion de la disparition d'un nœud, deux cas peuvent se présenter :

- Cas simple : si le nœud qui quitte est une feuille ou un nœud intermédiaire non-connecté à aucune feuille, il suffit de mettre à jour le réseau sans modification.
- Cas critique : si le nœud qui quitte est une racine ou un *cluster-head* ou une passerelle ou un nœud intermédiaire connecté à des feuilles. Dans ce cas un traitement spécifique est nécessaire. En effet, la disparition d'un nœud du type cité précédemment peut donner un graphe non-connexe. Il faut donc faire un redémarrage de SCA pour sélectionner un nouveau *backbone* afin de rejoindre tous les nœuds.

Algorithme 4.5 : node-move-out

1: Appeler n_r le nœud à supprimer

2 : **Selon** (id(n_r)) **Faire**

3 : **cas** n_l : mettre à jour $G_{cluster}$

4 : **cas** n_i : **Si** $N1(n_r)$ est vide **Alors**

5 : mettre à jour $G_{cluster}$

6 : **Si non**

7 : Essayer de connecter les $N1(n_r)$ à un autre n_i dans le même cluster

8 : **Si** il existe encore des nœuds dans $N1(n_r)$ sans n_i **Alors**

9 : Redémarrer SCA

10 : **Fin Si**

11 **: Fin Si**

12 : **cas** *r* : Redémarrer SCA à partir de la nouvelle racine

13 : **cas** n_{ch} : **Si** il est impossible de choisir un nouveau *cluster-head* **alors**

14 : Redémarrer SCA

15 : **Fin Si**

16 : **cas** n_g : **Si** il est impossible de choisir une nouvelle passerelle **alors**

17 : Redémarrer SCA

18 : **Fin Si**

19 : **Fin Selon**

4.3.3. La méthode MCR

La méthode MCR [46] est appliquée à un RCSF organisé selon la méthode SCA. Elle est réalisée en quatre étapes où la diffusion est réalisée par des sauts entre les différents clusters.

Algorithme 4.6 : MCR Multi-Cluster Relay

1 : Appliquer l'algorithme SCA pour organiser le réseau

2: Appliquer RDS-MPR á l'intérieur des clusters pour choisir les nœuds intermédiaires de relai.

3 : Former le *backbone* depuis la racine, les passerelles, les *cluster-heads* et les nœuds intermédiaires de relai.

4 : Réaliser la diffusion de la racine vers tous les nœuds du réseau en utilisant le *backbone* créé dans (3).

La méthode MCR possède un aspect dynamique puisqu'elle tient compte de :

- L'apparition d'un nouveau nœud qui est automatiquement pris en compte selon les règles de construction de SCA. Donc l'ajout d'un nouveau nœud n'est plus une source de perturbation pour la méthode MCR.
- L'absence d'un nœud : SCA réagit automatiquement pour prendre en compte le manque d'un nœud donc MCR peut éviter les pertes de paquets lors de la diffusion et peut aussi éviter les attaques de trou noir « *black hole* ».

Cet aspect dynamique fournit plus de possibilités pour la gestion du réseau puisqu'un utilisateur peut facilement remplacer un nœud d'énergie épuisé. Les pannes des nœuds peuvent être traitées. D'autre part, la mobilité peut être gérée puisqu'elle est la disparition d'un nœud d'un emplacement (*node-move-out*) et/ou l'apparition d'un nœud dans une autre place (*node-move-in*).

4.3.4. Etude comparative

Dans ce paragraphe, on fait une étude comparative entre RDS-MPR et MCR. On compare tout d'abord la complexité puis l'énergie totale de diffusion.

Pour comparer la complexité temporelle entre RDS-MPR et MCR, on utilise les notations données dans le tableau 4.4.

Tableau 4.4. *Notations pour la complexité*

Paramètre	**Désignation**
round	Intervalle fixe où un nœud peut réaliser une transmission, une réception ou un calcul interne.
D	Degré maximal d'un nœud dans le graphe *G*
L	Le diamètre de *G*
d	Degré maximal d'un nœud dans $G_{cluster}$
h	La hauteur du backbone qui est le résultat de l'application de SCA a *G*

En outre, on suppose que tous les nœuds sont similaires c.-à-d. ayant tous le même rayon de transmission.

La méthode RDS-MPR est basée sur le relai et réalisée selon l'information relative à deux niveaux pour chaque étape. Donc, au maximum cette méthode a besoin de D^2 *round* pour chaque étape de diffusion. Finalement, RDS-MPR peut s'achever en $O(LD^2)$ *round*s.

La diffusion en utilisant la méthode MCR peut se faire en d^2 *round* pour chaque cluster donc cette méthode peut se réaliser en $O(hd^2)$ *rounds*. D'autre part SCA peut être achevée, dans le pire des cas, en $O(D^2)$ *round*s. Finalement, MCR peut être achevée en $O(hd^2+D^2)$ *round*s.

Si on compare les complexités, de MCR et de RDS-MPR

On a : $d \leq D$ et $h \leq L$

Alors: $D^2 + hd^2 \leq D^2 + LD^2$ et $O(D^2 + LD^2) \approx O(LD^2)$

Finalement: $O(hd^2+D^2) \leq O(LD^2)$

Donc on peut conclure, que la complexité de MCR est inférieure à celle de la méthode RDS-MPR.

Ci-après une comparaison de l'énergie totale pour faire la diffusion entre MCR et RDS-MPR. Elle est réalisée pour des réseaux de dimensions différentes et pour des tailles diffèrentes de paquets.

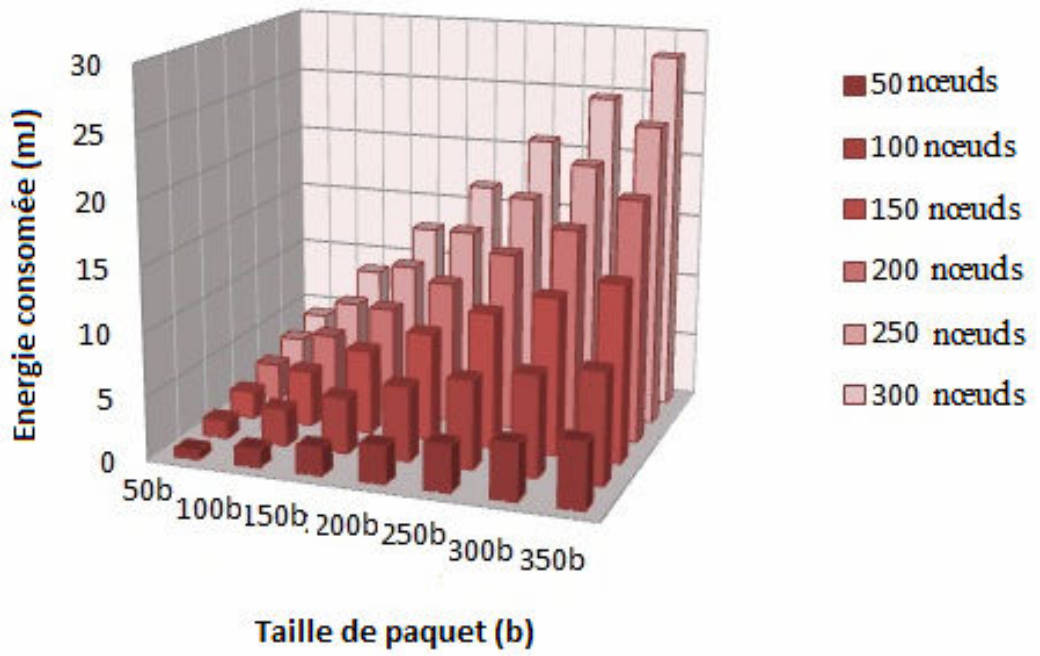


Figure 4.11. *Energie totale consommée avec RDS-MRR pour accomplir la diffusion avec des densités différentes et des tailles différentes de paquets*

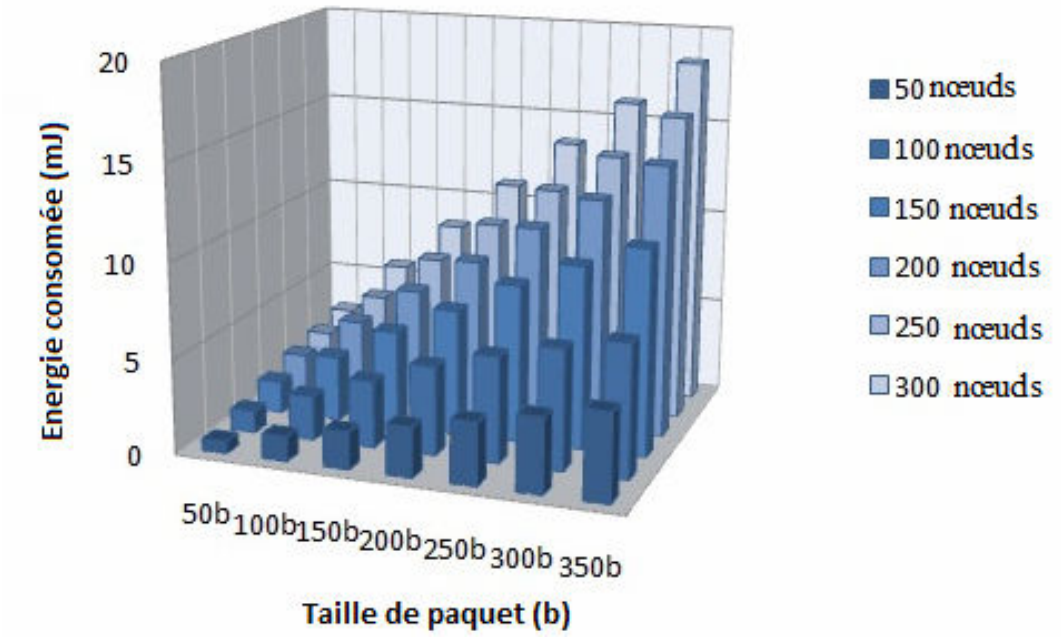


Figure 4.12. *Energie totale consommée avec MCR pour accomplir la diffusion avec des densités différentes et des tailles différentes de paquets*

Dans les deux figures précédentes, on a mesuré l'énergie totale pour réaliser une diffusion d'un paquet de données de taille variant de 50 octets a 350 octets par pas de 50 et pour des tailles de réseaux variant de 50 nœuds à 300 nœuds par pas de 50. On peut conclure, que MCR a diminué la consommation d'énergie d'une valeur moyenne variant entre 17% et 22%. Cette réduction est beaucoup plus importante tant qu' il y a des liens intra-niveaux entre les nœuds du RCSF. Cette amélioration est très importante pour un capteur contraint en énergie ce qui permet d'augmenter sa longévité donc la longévité du réseau entier. Dans la section suivante, on utilise 3 capteurs *Arduino* pour voir une expérimentation réelle de nos contributions.

4.4. Etude expérimentale avec las capteurs Arduino

4.4.1. Problème de la redondance circulaire

Le problème de la redondance circulaire [54] entre 3 nœuds-capteurs peut être décrite par la figure 4.13. lorsque le nœud *u* lance une diffusion en utilisant le *blind-flooding*. Ainsi, *v1* et *v2* auront une fonction de relai ce qui cause des réceptions redondantes au niveau de *u*, *v1* et *v2*. (Dans le reste de cette section on s'intéresse seulement aux réceptions redondantes au niveau de *v1* et *v2* puisque *u* est le nœud de base qui n'a pas de problème d'énergie limitée).

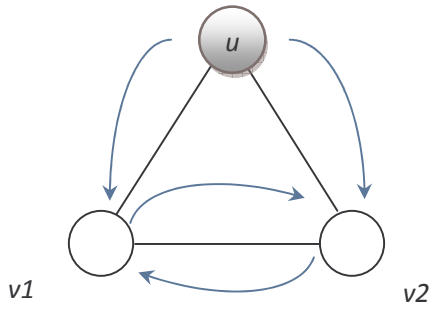


Figure 4.13. *Redondance circulaire*

Pour notre exécution réelle dans ce qui suit on remplace les 3 nœuds-capteur de la figure 4.13 par des capteurs *Arduino*.

4.4.2. Le capteur Arduino

Le capteur *Arduino* [5] est un capteur à faible coût, qui fonctionne avec Windows, Linux est Mac et qui est construit autour d'un microcontrôleur Atmel AVR.

4.4.3. Etude comparative de MCR et 3B sur 3 capteurs Arduino.

Dans la figure 4.14, on donne une image de la platforme utilisée dans l'expérience.

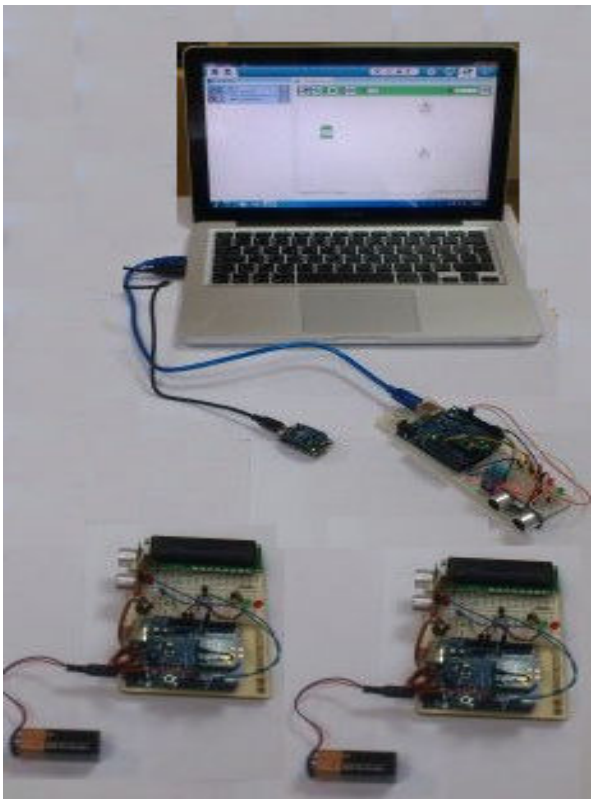


Figure 4.14. *Platforme d'expérimentation*

Pour tester les méthodes 3B et MCR on a positionné le nœud de base et les deux capteurs *Arduino* de la même façon que dans la figure 4.13, les identifiants des nœuds sont donnés dans la figure 4.15.

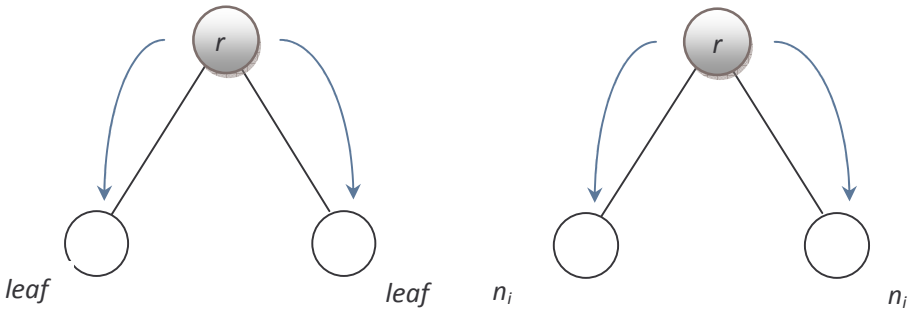


Figure 4.15. *Identifiants des nœuds: à droite si on applique MCR et à gauche avec 3B*

Pour notre expérience les deux algorithmes vont donner les mêmes résultats puisque les nœuds d'identifiant *leaf* ou n_i ne vont pas relayer les paquets à diffuser issus de *r*.

On réalise une diffusion et on compare nos algorithmes au *blind flooding,* on obtient les résultats dans la figure 4.16 qui donne l'énergie totale consommée pour réaliser la diffusion avec une taille de paquet égale à 16 Octets et deux rayons de transmission de 20m et 50m (on néglige l'énergie consommée au niveau du nœud de base puisqu'il n'a pas un problème de limitation d'énergie).

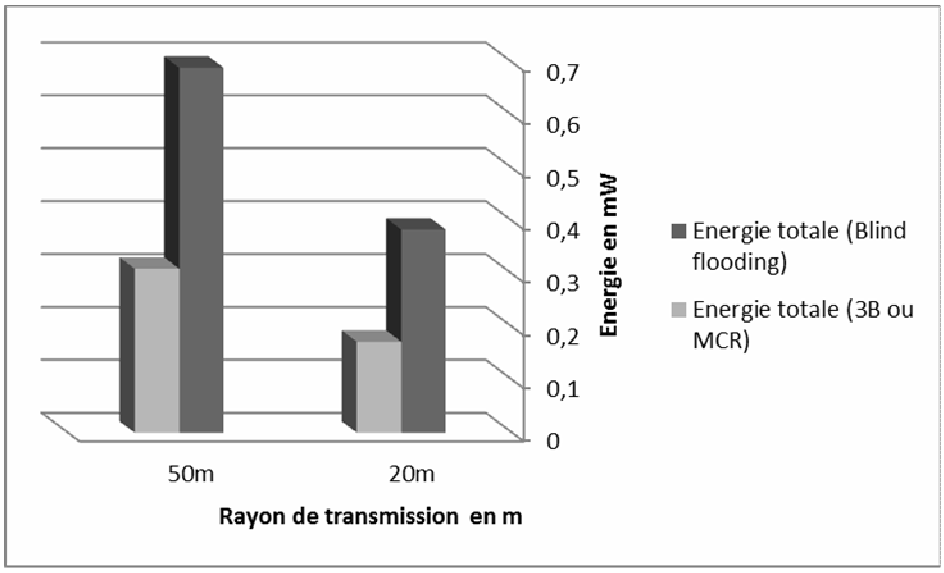


Figure 4.16. *Résultats expérimentaux*

Finalement, on peut conclure que nos deux contributions sont vérifiées expérimentalement avec un gain d'énergie de l'ordre de 44,87% par rapport au *blind flooding*.

4.5. Conclusion

Les inconvénients de la diffusion multi-sauts y compris MPR et RDS-MPR sont dus, d'une part, au fait, qu'ils ne tiennent pas compte:

- Des liens entre les nœuds de même niveau par rapport au nœud initiateur de le diffusion,
- Des réceptions redondantes,
- De l'aspect dynamique d'un RCSF.

D’autre part les méthodes d’auto-organisation dans les RCSF permettent de mieux gérer les communications entre les entités du réseau et minimisent les collusions et les redondances ce qui permet d’augmenter la longévité des capteurs et l’efficacité des services fournis par le réseau.

Dans ce chapitre, on a proposé une amélioration de la méthode MPR qui consiste à transformer le réseau non structuré en un réseau structuré en clusters, les *cluster-heads* sont les nœuds choisis par MPR pour faire le relai. L’ensemble des *cluster-heads* et le nœud de base forment un *backbone* qui constitue l’axe de diffusion. Cette amélioration a permis d’éviter les réceptions redondantes permettant ainsi de conserver une partie de l’énergie du capteur.

La méthode RDS-MPR est également améliorée en adoptant une méthode d’auto-organisation dynamique du RCSF appelée MCR qui peut tenir compte de la présence ou l’absence d’un nœud automatiquement. Ensuite on applique RDS-MPR á l’intérieur des clusters créés par MCR. Finalement, la diffusion se fait à travers un *backbone* formé par le nœud initiateur de la diffusion, les nœuds intermédiaires de relai choisis par RDS-MPR, les passerelles entre les clusters et les *cluster-heads*.

La nouvelle méthode MCR fait le lien entre les clusters pour achever la diffusion. Elle constitue un couplage entre les méthodes de diffusion basées sur le multi-saut et les méthodes de diffusion basées sur l’utilisation des clusters. Cette méthode a permis d’améliorer RDS-MPR par la minimisation de l’énergie totale pour achever la diffusion et la minimisation de la complexité temporelle.

Nos deux contributions MCR [46] et 3B [48] constituent des améliorations respectives de RDS-MPR et MPR en réduisant l’énergie totale de diffusion par l’élimination des liens causant les réceptions redondantes. Mais cette amélioration n’a pas pris en compte la tolérance aux fautes puisque chaque nœud est lié à une seule source de données, ce qui constitue un problème en cas de panne de cette source.

Ainsi, dans le chapitre suivant on orientera notre travail vers la tolérance aux fautes. Pour cela, on proposera une amélioration des méthodes de diffusion qui consiste à gérer les pannes tout en continuant à fonctionner même lors de la présence de fautes.

Chapitre 5

L'équilibrage de charge et la tolérance aux fautes pour la dissémination

Résumé:

Ce chapitre présente nos deux autres contributions pour réaliser la diffusion :

- *L'équilibrage de charge en utilisant un backbone virtuel à travers les ensembles LB(k,r)-CDS [44].*
- *La tolérance aux fautes appliquée aux nœuds de relai en introduisant un indice de confiance pour chaque nœud [49].*

5.1. Introduction

De nombreux travaux se sont intéressés à la minimisation de la taille des *backbones* afin de minimiser le coût des communications lors du routage d'information ou de la diffusion. Récemment, des études ont montré que l'équilibrage de charge permet d'augmenter la longévité du réseau comparé à la minimisation de la taille du *backbone* qui peut causer un déséquilibre entre les tâches réalisées par chaque nœud [26,27,28]. En conséquence, l'épuisement rapide de l'énergie de certains nœuds peut aboutir à un réseau non connexe. C'est dans ce contexte que se situe notre première contribution de ce chapitre : la diffusion avec équilibrage de charge décrite par la méthode LB(k,r)-CDS : Load Balanced (k,r)-Connected Dominating Set (ensemble k-connexe à r sauts avec équilibrage de charge) [27].

Le déploiement et la mise en œuvre des RCSF peuvent être soumis à plusieurs pannes dues aux mauvais fonctionnement du matériel, à des problèmes logiciels ou à un problème environnemental [15]. De ce fait, garantir un bon niveau de tolérance aux fautes est très important pour un fonctionnement normal d'un RCSF. Les fautes dans les RCSF peuvent être dues au mauvais fonctionnement d'un composant (exemples: microprocesseur, batterie, unité de capture et interférence réseaux), ou lié à la contrainte d'énergie puisque pour un niveau d'énergie faible du capteur il peut avoir des erreurs de lecture. Les fautes peuvent être également dues aux logiciels qui peuvent être une source de bug, ou aux fautes liées à la communication sans fil.

On voit donc qu' il est très important de tenir compte de l'aspect « tolérance aux fautes » lors de la diffusion. C'est dans ce contexte que se situe notre deuxième contribution de ce chapitre qui est une amélioration de la diffusion basée sur le relai avec la prise en compte d'un indice de confiance pour chaque nœud-capteur [71].

5.2. Diffusion par la méthode LB(k,r)-CDS.

Les ensembles (k,r)-CDS sont utilisés afin de réduire les tailles des ensembles dominants connexes ce qui permet de sélectionner un *backbone* de petite taille. Dans ces ensembles chaque nœud dominé est lié au moins à k nœuds dominants. De plus, la distance entre un dominant et un dominé est au maximum égale à r sauts. L'existence de plusieurs chemins entre un nœud dominé (hors du *backbone*) et les nœuds dominants (nœuds constituant le *backbone*) permet de substituer un chemin défaillant par un autre chemin valide.

5.2.1. Définition du problème.

La plupart des méthodes existantes de sélection des ensembles dominants connexes se proposent de résoudre le problème MCDS qui est un problème NP-Complet [22]. MCDS veut dire sélectionner un CDS de taille minimum afin de minimiser la taille du *backbone*. Malgré les nombreux efforts de recherche réalisés pour la sélection d'un MCDS, celui-ci a conduit, dans la plupart des cas, a un déséquilibrage de charge ce qui touche la longévité du réseau et surtout sa connectivité. Récemment, des études ont proposé une résolution du LB-CDS au lieu du MCDS. Ainsi, la sélection d'un LB-CDS a permis l'amélioration de la longévité du RCSF avec un taux de 80% [27, 28].

Pour créer un LB(k,r)-CDS, on s'est inspiré des techniques de création du LBCDS dans [28] afin d'extraire un *backbone* qui possède trois critères:

- Tolérant aux fautes puisqu'on parle de l'ensemble (k,r)-CDS.
- Equilibrant la charge en utilisant les mêmes techniques du LBCDS.
- De taille inférieure à un *backbone* formé par la méthode LBCDS vu les distances entre les nœuds dominants et dominés.

Dans le tableau 5.1, on donne la liste des symboles et notations utilisés pour résoudre le problème LB(k,r)-CDS et dans la figure 5.1, un exemple de graphe du RCSF est présenté. Ainsi,

l'ensemble (1,2-saut)-CDS, en utilisant la dernière méthode de sélection du (k,r)-CDS proposée dans [10] est : *{c,f}*.

Tableau 5.1. *liste des symboles*

Symbole	**Description**
G(V,E)	Graphe de connectivité représentant le réseau de capteurs, où *V* représente l'ensemble des nœuds et *E* l'ensemble des arrêtes entre les nœuds *V*.
a,b,c,…	Identifient la liste des nœuds capteurs dans le réseau.
ND(x)	L'ensemble des nœuds dominés par le nœud *x*
|ND(x)|	Taille de l'ensemble *ND(x)*
p-norm	Défini pour calculer le facteur d'équilibrage de charge pour un sous-ensemble du RCSF ayant *n* nœuds, ce facteur a été proposé dans [29] et a été utilisé dans [27] et [28] , il est égal à $\| X \|_p = (\sum_{i=1}^{n} \| x_i \|^p)^{1/p}$ (5.1)
d(x)	Degré d'un nœud-capteur (Nombre de voisins)
$\overline{d}$	Partie entière supérieure de *d(x)*
D	Ensemble des dominants dans un graphe G

Pour mesurer le degré d'équilibrage de charge de cet ensemble on calcule son *p-norm*. Un ensemble fournit le meilleur équilibrage de charge s'il possède le *p-norm* le moins faible avec *p=2* [29]. on choisit :

$$x_i = | d_i - \overline{d} | \quad (5.2)$$

Donc , $p - norm(\{c, f\}) = \sqrt{| 4-3 |^2 + | 6-3 |^2} = \sqrt{10}$ (5.3)

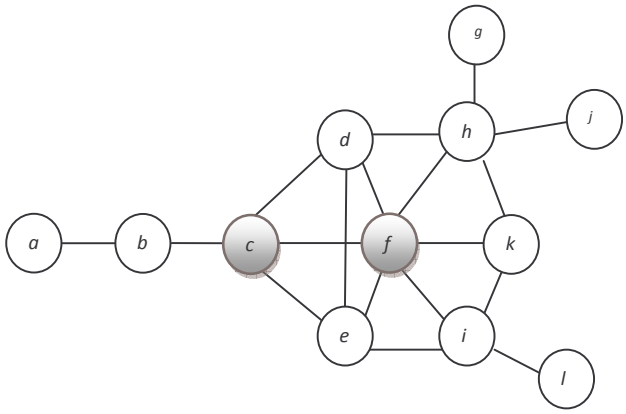


Figure 5..1 *Exemple d'un graphe représentant un RCSF. L'ensemble (1,2-saut)-CDS est {c,f}.*

Dans la figure 5.2, on choisit un nouvel ensemble qui est *{c,d,e}*. Le *p-norm* de cet ensemble est :

$$p-norm(\{c,d,e\}) = \sqrt{|4-3|^2 + |4-3|^2 + |4-3|^2} = \sqrt{3} \quad (5.4)$$

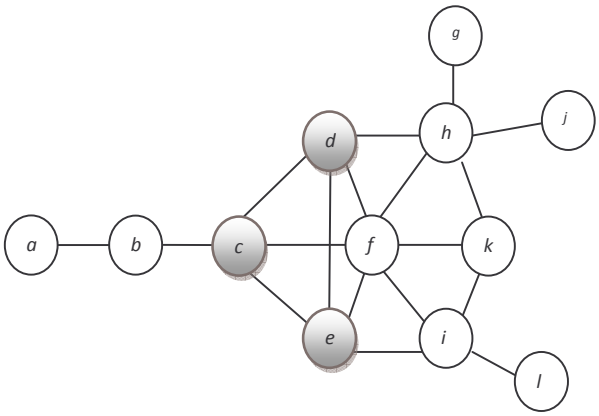


Figure 5..2 *Exemple d'un graphe représentant un RCSF. L'ensemble (1,2-saut)-CDS est {c,d,e}.*

Si on compare le *p-norm* des deux ensembles *{c,f}* et *{c,d,e},* on peut conclure que *{c,d,e}* offre un meilleur équilibrage de charge que *{c,f}* vu que son *p-norm* est inférieur. De ce fait, la sélection de l'ensemble des nœuds dominants doit tenir compte de la minimisation du *p-norm* .

L'affiliation des nœuds dominés doit aussi garantir l'équilibrage de charge. Ci-après on donne des différentes affiliations pour qu'on puisse choisir celle qui maximise l'équilibrage de charge.

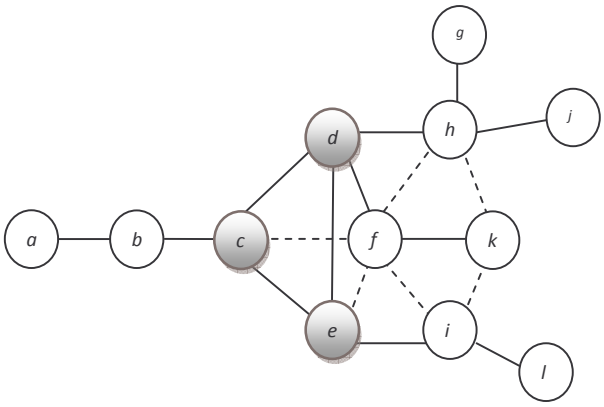


Figure 5..3 *Affectation des nœuds dominés à leur dominant approprié : 1ère possibilité*

D'après la figure 5.3, *ND(c)={b,a}; ND(d)={h,g,j,f,k}* et *ND(e) ={i,l}*. Dans la figure 5.4, on donne une autre affectation avec : *ND(c)={b,a}; ND(d)={h,g,j}* et *ND(e) ={i,l,f,k}*.

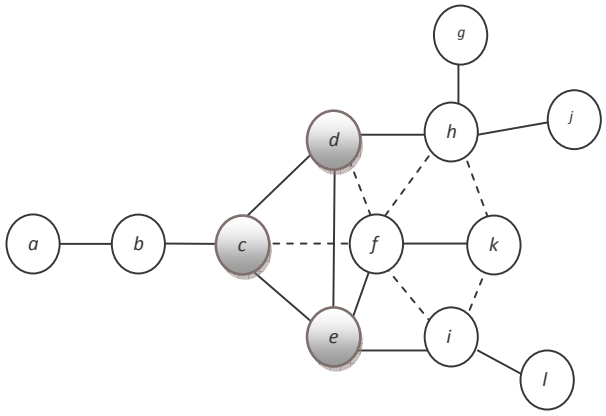


Figure 5..4 *Affectation des nœuds dominés à leur dominant approprié: 2ème possibilité*

Si on veut comparer laquelle des deux affectations précédentes est plus adéquate en fonction de l' équilibrage de charge, on mesure leur *p-norm* avec :

$$x_i = \| ND(i) \| - \overline{ND(i)} | \tag{5.5}$$

Pour la première affectation :

$$p-norm(\{c,d,e\}) = \sqrt{|2-3|^2 + |5-3|^2 + |2-3|^2} = \sqrt{6} \tag{5.6}$$

Pour la deuxième affectation :

$$p-norm(\{c,d,e\}) = \sqrt{|2-3|^2 + |3-3|^2 + |4-3|^2} = \sqrt{2} \tag{5.7}$$

Finalement, on peut conclure que la deuxième affectation fournit un meilleur équilibrage de charge.

5.2.2. Algorithme LB(k,r)-CDS.

Dans ce paragraphe, on donne un traitement formel du problème LB(k,r)-CDS. D'après ce qui précède, la construction d'un LB(k,r)-CDS doit tenir compte de la minimisation du *p-norm*. Celle-ci est réalisée en deux étapes :

- Sélection des nœuds dominants en maximisant l'équilibre de charge.
- Affectation équilibrée des nœuds dominés aux nœuds dominants.

(a) Sélection de l'ensemble dominant connexe des nœuds: D

L'ensemble *D* est un sous-ensemble de *V* dans un graphe *G(V,E)* respectant les propriétés suivantes :

- *D* doit être connexe.

- Pour tout nœud *v* dans *V* \ *D*, il existe au moins un nœud *u* dans *D,* où la distance entre *v* et *u* est au maximum r sauts.
- $Minimiser\ |D|_p = (\sum_{i=1}^{|D|} | d(x_i) - \overline{d} |^2)^{1/2}$ (5.8)

Ainsi, la sélection de l'ensemble *D* commence par des nœuds dans *V* qui minimisent la valeur $|D|_p$ et se termine quand *D* est connexe et lorsque chaque nœud dans *D* est au maximum à une distance r sauts des nœuds dans *V* \ *D*. Finalement, on optimise l'ensemble *D* c.-à-d. on extrait les nœuds *n* de *D* si *D* \ *{n}* respecte les critères mentionnés ci-dessus

Algorithme 5.1 : Sélection des nœuds dominants

Entrées: un graphe connexe *G(V,E)*

Sortie: la liste des nœuds dominants *D*

1: $D \leftarrow \emptyset$

2: calculer $\overline{d}$ dans *G*

3: **Pour tout** nœud *n* dans *G* **faire**

4: **Si** $d(n) = \overline{d}$ **alors** $D \leftarrow D \cup \{n\}$

5: **Si** *D* est connexe et tous les nœuds dans *V* \ *D* sont à une distance maximale r sauts de moins un nœud de *D* **alors**

6: **sort de la boucle**

7: **Si non** $\overline{d} \leftarrow \overline{d} + 1$

8: **Fin Si**

9: **Fin Pour tout**

10: **Pour tout** nœud *n* dans *D* **faire**

11: **Si** *D*\{*n*} est connexe et tous les nœuds dans *V**D* sont à une distance maximum r sauts d'au moins un nœud de *D*\{*n*} **alors**

12: $D \leftarrow D \setminus \{n\}$

13: **Fin Si**

14: **Fin Pour tout**

(b) Affectation des nœuds dominés aux nœuds dominants :

Le problème d'affectation des nœuds dominés aux nœuds dominants peut être définit ainsi:

- Chaque nœud dominé est affecté à un seul dominant.
- La distance entre un nœud dominé et son dominant ne doit pas dépasser r sauts.
- L'allocation doit maximiser

$$| D |_p = (\sum_{i=1}^{|D|} || ND(x_i) | - \overline{ND(x_i)} |^2)^{1/2}$$

L'algorithme suivant illustre les propriétés précédentes en commençant par l'allocation des nœuds imposés c.-à-d.:

- Les nœuds affectés à un seul nœud dominant,
- Les nœuds voisins d'un seul nœud dominant,
- Les nœuds voisins d'un seul nœud dominé affecté.

Pour le reste des nœuds l'affectation se fait selon le dominant qui minimise $|ND(x)|$.

Algorithme 5.2 : Affectation des dominés aux dominants

Entrées: un graphe connexe *G(V,E),* la liste des nœuds dominants *D*

Sortie: tous les dominés sont affectés aux dominants

1: $S \leftarrow V \setminus D$

2: $Temp \leftarrow G(V,E)$

3: Affecter des dominés imposés et les éliminer de *S*.

4: **Pour tout** nœud *n* dans *S* et *x* dans *D* **Faire**

5: r ←1

6: choisir l'ensemble *x* où on peut affecter n à r sauts dans *Temp*

7: choisir *x* où son $|ND(x)|$ est minimum

8: $ND(x) \leftarrow ND(x) \cup \{n\}$

9: $S \leftarrow S \setminus \{n\}$

10: Supprimer les arrêtes entre les autres *x* et *n* dans *Temp*

11: $r \leftarrow r+1$

12: **Si** *S* est vide **break**

13: **Fin Pour tout**

Si on applique l'algorithme 5.2 au réseau de la figure 5.2. pour faire l'affectation des nœuds dominés aux nœuds dominants, on commence par $ND(c)=\{a,b\}$ puisque a et b sont imposés à c. Ensuite, idem pour h et i qui sont imposés respectivement à d et e. Ensuite, l est imposé à e et g,j sont imposés à d. En conséquence, $ND(d)=\{h,g,j\}$ et $ND(e)=\{i,l\}$. le nœud f sera lié à e parce que $|ND(e)|<|ND(f)|$. Finalement, $ND(e)=\{i,l,f,k\}$ et, S sera vide donc l'algorithme se termine.

5.2.3. Evaluation du LB(k,r)-CDS.

La mesure de performance du LB(1,2)-CDS a été faite en utilisant le simulateur NS-2, avec des nœuds identiques de rayon de communication *50m*. Les tailles des paquets de données à transmettre sont entre 200 et 400 octets. On suppose dans ce test que le nœud de base est directement lié à un des nœuds dominants de l'ensemble LB(1,2)-CDS. Chaque 10 secondes, le nœud de base diffuse un paquet de données à tous les nœuds du réseau et après 10 secondes il reçoit les données collectées de tous les autres nœuds. Finalement, on extrait périodiquement l'énergie du nœud qui a fait plus de communication que les autres nœuds dans LB(1,2)-CDS. Ensuite, on compare cette énergie à l'énergie du nœud qui a fait plus de communication que les autres nœuds dans (1,2)-CDS [10]. Initialement, chaque nœud-capteur est alimenté par deux batteries AA de lithium dont l'énergie initiale est proche de *30kJ* . On prend en considération seulement les énergies de communication et on néglige l'énergie de capture et de traitement interne.

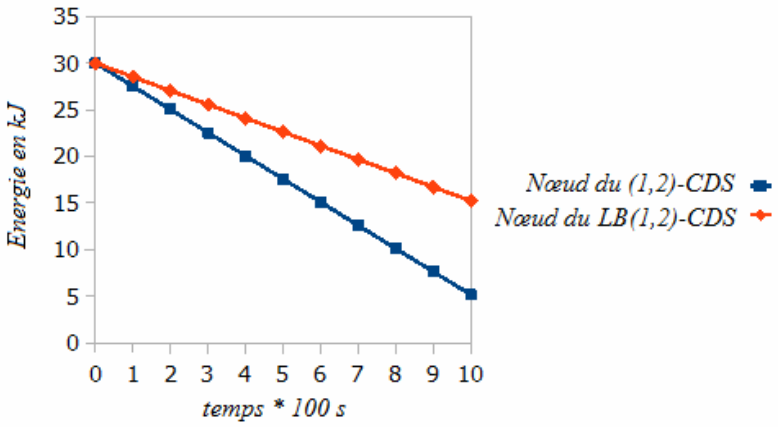


Figure 5.5. *Evolution de l'énergie dans le nœud qui a fait plus de communication dans (1,2)-CDS avec une densité égale à 12 et une taille de paquet égale à 200 octets*

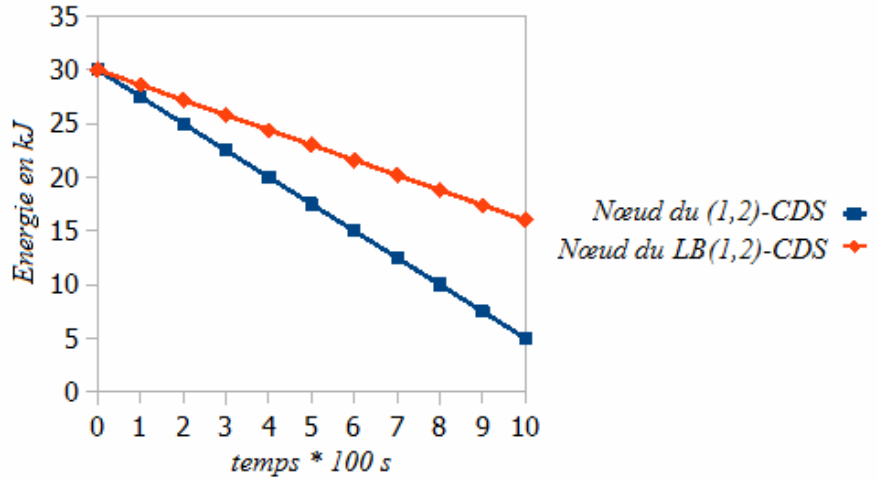


Figure 5.6. *Evolution de l'énergie dans le nœud qui a fait plus de communication dans (1,2)-CDS avec une densité égale à 200 et une taille de paquet égale à 200 octets*

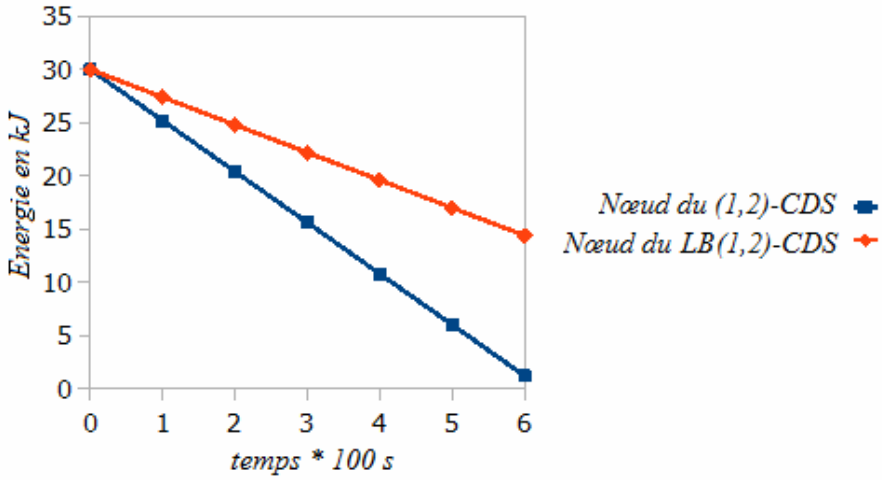


Figure 5.7. *Evolution de l'énergie dans le nœud qui a fait plus de communication dans (1,2)-CDS avec une densité égale à 200 et une taille de paquet égale à 400 octets*

Dans la figure 5.5, on peut voir une amélioration importante de longévité dans le temps du nœud dominant qui a fait plus de communication dans (1,2)-CDS. Les deux courbes sont linéaires donc on peut calculer le temps d'épuisement total du nœud. Ainsi, on peut conclure que notre contribution améliore la durée de vie du nœud de 68% comparé à la dernière méthode proposée de sélection du (1,2)-CDS [10]. Ce résultat est confirmé dans la figure 5.6 en utilisant un réseau de 200 nœuds et 200 octets pour la taille des paquets et dans la figure 5.7 en utilisant un réseau de 200 nœuds et 400 octets pour la taille des paquets.

L'augmentation de la durée de vie du nœud dominant qui a fait plus de communication dans l'ensemble (1,2)-CDS a un effet bénéfique concernant la longévité du réseau entier. En effet, si un nœud dans l'ensemble des dominants du (k,r)-CDS est épuisé on aura un *backbone* non connexe et un ensemble de nœuds capteurs ne sera pas couvert.

5.3. Diffusion par relai tolérante aux fautes.

La tolérance aux fautes est une propriété fondamentale pour la réalisation de la diffusion. Dans ce qui suit on donne notre contribution qui prend en compte la tolérance aux fautes.

5.3.1. Indice de confiance.

Dans [71] les auteurs donnent une définition d'un indice *TI(u)* représentant le degré de confiance pour un nœud *u*. Cet indice qui varie entre 0 et 1 évolue selon le comportement du nœud et il est défini pour un nœud *u* par l'équation :

$$TI(u) = e^{-\lambda v(u)} \tag{5.8}$$

Avec :

λ: une variable permettant de recalculer *TI* en jouant sur la rapidité de changer *TI(u)* lorsque *v(u)* croît ou décroît.

Initialement *TI=1* pour tous les nœuds-capteurs. Un nœud est plus sûr lorsque son indice de confiance *TI* est plus élevé. Plusieurs fautes peuvent être la cause d'un tel mauvais comportement par exemple: un nœuds peut être placé dans un milieu dangereux, à un bas niveau d'énergie un nœud peut donner des mauvais mesures, ou des autres fautes liées à la couche matérielle ou logicielle du nœud-capteur.

v(u) est initialement égal à 0 pour avoir un *TI* égal à 1 pour tous les nœuds. Ensuite, *v(u)* croît depuis le nœud de base avec un pas de 0.1 dans le cas où *u* est estimé faux ou erroné. Dans le cas contraire c.-à-d. lorsque *u* est estimé correct, *v(u)* décroît depuis le nœud de base avec un pas de 0.1. Ce comportement est décrit ci-après par l'équation:

$$\begin{cases} v_0(u)=0 \\ v(u)= \begin{cases} v(u)+0.1 & \text{Si } u \text{ est estimé erroné} \\ Max(v(u)-0.1;\ 0) & \text{Si non} \end{cases} \end{cases} \qquad (5.9)$$

Finalement, on déduit que plus *v(u)* augmente plus *TI* diminue. Inversement, pour des rapports positifs *v(u)* qui diminuent, *TI* augmente. Dans la figure 5.8, on donne une représentation de la fonction *TI* pour λ=1.

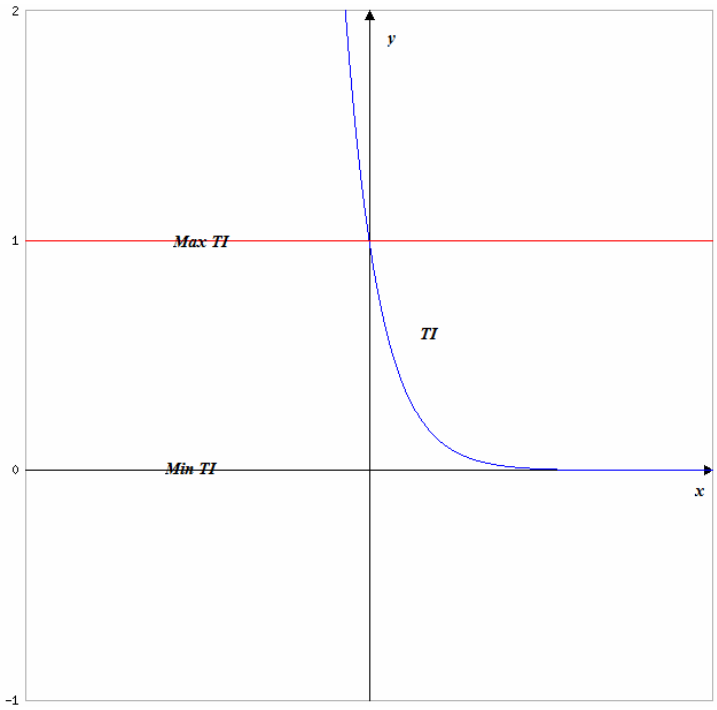


Figure 5..8. *Fonction d'indice de confiance entre Max TI=1 et Min TI=0 lorsque v(u)>=0*

5.3.2. Application de FCA pour effectuer la diffusion avec le relai.

L'analyse formelle des concepts (FCA Formal Concept Analysis) est un outil d'analyse des données utilisé dans beaucoup de domaines scientifiques permettant de grouper un nombre

d'objets possédant un groupe d'attributs en commun. Dans ce paragraphe on donne quelques définitions sur cet outil [66,68].

Définition 1 : Contexte

Un contexte est un type spécifique de données, qui est composé d'un ensemble d'objets (O) décrit par un ensemble d'attributs (A). Un contexte peut être représenté par un tableau n*m où les objets sont mis dans les lignes et les attributs dans les colonnes. Si un objet dans la colonne *i* possède l'attribut de la colonne *j*, ceci sera représenté par une croix dans la cellule *(i,j)* du tableau, si non la cellule sera vide. Au lieu des croix, le tableau peut être booléen avec des 1à la place des croix et 0 si non.

Définition 2 : Concept formel

Soient A un ensemble d'attributs et O un ensemble d'objets décrivant un contexte C alors

(a,o) est un concept formel de C, si et seulement si :

(1) $a \subset A$,

(2) $o \subset O$ et

(3) La liste des attributs *a* est commune pour la liste des objets *o*.

Définition 3 : FCA

FCA est une technique appliquée au contexte décrit précédemment et réalisée en deux phases.

(1) Extraire tous les concepts formels d'un contexte.

(2) Trouver les groupes d'objets qui partagent des groupes d'attributs en commun.

Définition 4 : Treillis de Galois

La classification du concept entier forme le treillis de Galois, qui est le résultat de l'application du FCA sur un contexte c.-à-d. extraire est classer tous les concepts formels d'un contexte selon leurs attributs en commun.

5.3.3. Algorithme : diffusion par relai tolérant aux fautes avec FCA

Dans cette section, on présente l'application du FCA pour réaliser l'algorithme de diffusion.

Contexte « Relai »

Le but de l'algorithme est de sélectionner un *backbone* pour faire la diffusion tout au long du réseau en utilisant les techniques de relai. Pour décider si un nœud fait partie du *backbone*, RDS-MPR utilise une fonction linéaire permettant de sommer les poids normalisés ensuite le nœud de poids maximum est sélectionné. Dans notre technique quatre critères sont utilisés:

- Poids de l'énergie,

- Poids du degré,

- Poids de la probabilité de réception,

- Poids de l'indice de confiance.

Le contexte *Relai* est donc formé par les objets qui sont les nœuds du premier niveau du nœud sélecteur (nœud en couleur foncée dans la figure 5.9). Les attributs sont les critères décrits précédemment.

On se propose de définir des seuils pour chaque critère appelés *e0,d0,p0* et *ti0* respectivement pour le poids de l'énergie, du degré, de la probabilité de réception et de l'indice de confiance.

Un nœud *n* vérifie le critère du poids d'énergie par exemple si et seulement si son énergie restante dépasse *e0*. Même chose pour les autres nœuds et pour les autres critères.

Soit le RCSF décrit par la figure 5.9, on se propose d'extraire le contexte relai relatif à cette figure.

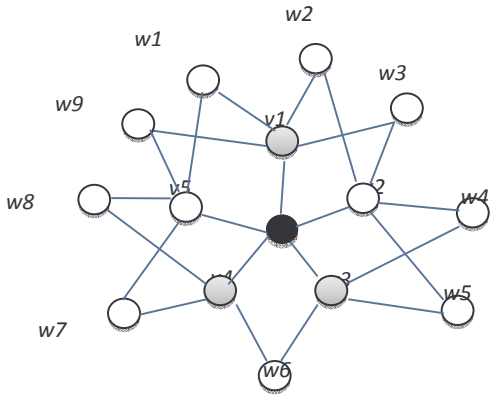


Figure 5.9. *Exemple d'un RCSF*

Le contexte *Relai* est décrit dans le tableau 5.2.

Tableau 5.2. *Contexte Relai*

	PE: Poids de l'énergie	PD: Poids du degré	PP: Poids de la probabilité de réception	PI: Poids d'indice de confiance
v1	x	x	X	x
v2		x	X	
v3	x	x	X	x
v4		x	X	x
v5	x	x		

Apres avoir déterminé le contexte *Relai* on peut générer ci-après le treillis de Gallois *Relai.*

Treillis de Galois Relai

Beaucoup d'outils permettent de générer le treillis de Galois à partir d'un contexte par exemple celui du [20]. Dans la figure 5.10, on donne le treillis *Relai* généré à partir du contexte *Relai*.

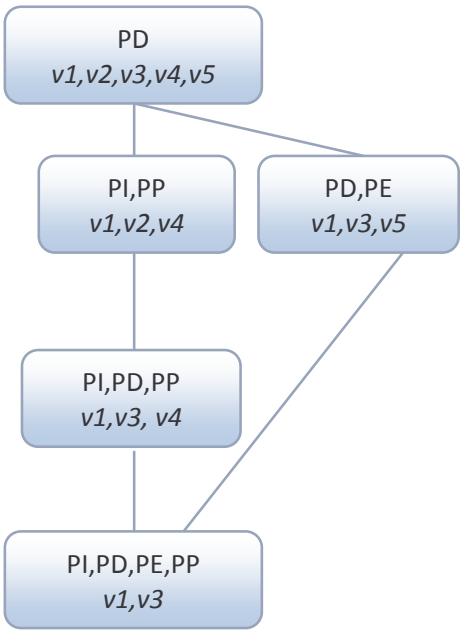


Figure 5.10. *Treillis Relai*

Avantages du treillis de Galois par rapport aux fonctions linéaires

Le treillis de Galois permet de donner un tri parfait des nœuds selon leurs critères ce qui permet de choisir les meilleurs nœuds de relai. Dans le tableau suivant, on donne des informations relatives à deux nœuds où les facteurs de pondération sont tous égaux à ¼ pour voir la différence de choix des nœuds de relai en utilisant FCA par rapport à l'utilisation des fonctions linéaires.

Tableau 5.3. *Choix des nœuds de relai*

	PE	PD	PP	PI	Somme des poids
v1	0.2	0.2	0.7	0.5	0.40
v2	0.3	0.4	0.4	0.3	0.35

D'après le tableau 5.3, l'utilisation d'une fonction linéaire favorise le nœud *v1* puisque son poids total est supérieur à celui de *v2* malgré son poids d'énergie et le degré qui sont faibles par rapport à *v2*. En effet, les poids faibles sont cachés par les poids forts puisque la décision du meilleur nœud se fait en se basant sur l'utilisation des fonctions linéaires qui somment l'ensemble des poids. Au contraire, si on utilise la technique FCA le nœud *v2* est favorisé. En effet, notre technique exige un seuil pour chaque poids. Ce qui est mieux vérifié par *v2*.

Algorithme RRB

Dans le tableau 5.4, on donne la liste des symboles et leurs descriptions qui sont utilisés dans notre algorithme.

Tableau 5.4. *Symboles et notations*

Symboles	**Descriptions**
U	Nœud sélecteur des nœuds de relai
N1(u)	Premier voisinage de *u*
N2(u)	Deuxième voisinage de *u*

S(u)	Sous ensemble de *N1(u)* choisi comme ensemble optimal pour faire le relai.
v	Un élément de *N1(u).*
w	Un élément de *N2(u).*
Relai	Contexte des nœuds dans *N1(u)*
LR	Treillis de Galois généré à partir de *Relai*
l	Un niveau de *LR*

Algorithme 5.3 : RRB Reliable Relay-based Broadcasting

Entrées:

un nœud *u* et les informations sur son voisinage a deux sauts dans un graphe connexe *G(V,E)*

Sortie:

la liste des nœuds de relai *S(u)*

1: *S1* ← *N2(u)* et

2 : *S(u)* ← *Ø*

3 : Ajouter tous les nœuds imposés à *S(u)* et éliminer leurs premiers voisinages de *S1(u)*

4 : Générer le contexte relai des nœuds dans *N1(u)* non encore choisis comme nœuds de relai.

5: Générer le treillis *LR*

4 : **Tant que** *S1(u)* ≠ *Ø* **Faire**

Commencer par les nœuds dans le niveau bas du *LR*, l'ajouter à *S(u)* et éliminer leur premier voisinage de *S1(u)* . Si le nombre de nœuds est supérieur à 1 choisir celui d'énergie maximum.

5 : **Fin Tant que**

Application du RRB

Si on applique notre algorithme au RCSF de la figure 5.9, l'ensemble *N1* est initialisé par *{v1,v2,v3,v4,v5}, S1* sera initialisé par *{w1,w2,w3,w4,w5,w6,w7,w8,w9}* et *S*

sera vide. Il n'y a pas de nœuds imposés donc l'algorithme générera le contexte *Relai* (tableau 5.3). Ensuite *LR* est généré (figure 5.10). Dans le niveau plus bas du *LR* on trouve *v1* et *v3*. Si l'énergie restante dans *v1* est plus élevée que celle de *v3* alors *S={v1}* et *S1={w4,w5,w6,w7,w8}*. *S1* n'est pas encore vide. En conséquence, on rajoute *v3* à *S* donc *S={ v1,v3}*. L'algorithme passe au niveau suivant plus haut dans *LR* où il y a seulement un nœud puisque *v3* et *v1* sont déjà sélectionnés. Finalement *S={v1,v3,v4}* et S2 sera vide donc l'algorithme se termine et l'ensemble optimal de relai sera *{v1,v3,v4}* .

5.3.4. Etude de performance

Dans ce paragraphe on va simuler notre algorithme avec les paramètres du tableau 5.4. On se propose dans cette simulation de fixer le champ de déploiement à 500 x 500 m et on fait varier le nombre des nœuds-capteurs entre 50 et 300 avec un pas de 50. Le rayon de transmission est fixé à 50m. le protocole de routage utilisé pour diffuser les paquets de données est AODV. On définit des énergies pour l'émission, la réception, le repos, le sommeil, et la transition entre l'état de repos et l'état de sommeil. On définit aussi des seuils pour chaque poids (seuil d'énergie, seuil de degré, seuil de probabilité et seuil d'indice de confiance).

L'ensemble des capteurs est déployé arbitrairement à condition de former un graphe connexe. Dans cette simulation on suppose de plus que chaque indice de confiance est une valeur générée aléatoirement pour chaque nœud et que chaque nœud-capteur connait l'indice de confiance des nœuds de son premier voisinage qui est mis à jour lors de la collecte des données.

Tableau 5.5. *Paramètres de simulation*

Paramètre	**Valeur**
Taille du champ	500/500 *m*
Nombre de nœuds	50 à 300
Rayon de transmission	50*m*
Protocole de routage	AODV
Energie initiale du nœud	10 *kJ*
Energie d'émission	2.0 *mW*
Energie de réception	1.0 *mW*
Energie à l'état de repos	0.2 *mW*
Energie à l'état de sommeil	0.02 *mW*

Energie de transition du sommeil au repos	2.0 *mW*
Temps de transition du sommeil au repos	0.05 *secondes*
Liste des seuils	
Seuil d'énergie	5.0 *mW*
Seuil de degré	2
Seuil de probabilité	0.5
Seuil d'indice de confiance	0.5

Dans la figure 5.11, on fixe le nombre de nœuds dans le champ à 150 nœuds et on fait varier le taux des nœuds défaillants entre 10% et 50% avec un pas de 10%. Dans la simulation, les nœuds d'indice de confiance faible sont les nœuds les plus susceptibles d'être défaillants. Ensuite, on calcule le taux de succès de chaque méthode qui est le nombre total de nœuds qui ont reçu le paquet de données envoyé par le nœud initiateur de diffusion. Dans notre technique *RRB* on se propose de sélectionner les nœuds avec un indice de confiance élevé. Ainsi, le taux de succès de RRB est nettement supérieur à celui de la méthode RDS-MPR. Selon la figure 5.11, RRB améliore le taux de succès de RDS-MPR, dans le meilleur des cas de 36.7%. Dans le pire des cas le taux de succès de RRB est le même que RDS-MPR.

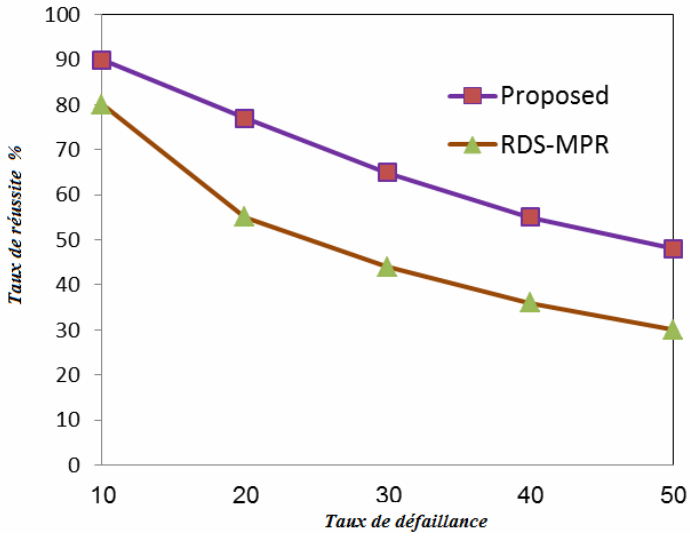


Figure 5.11. *Taux de réussite de RRB comparé à RDS-MPR pour un taux de défaillance de nœuds qui varie entre 10 et 50% du nombre total des nœuds-capteurs*

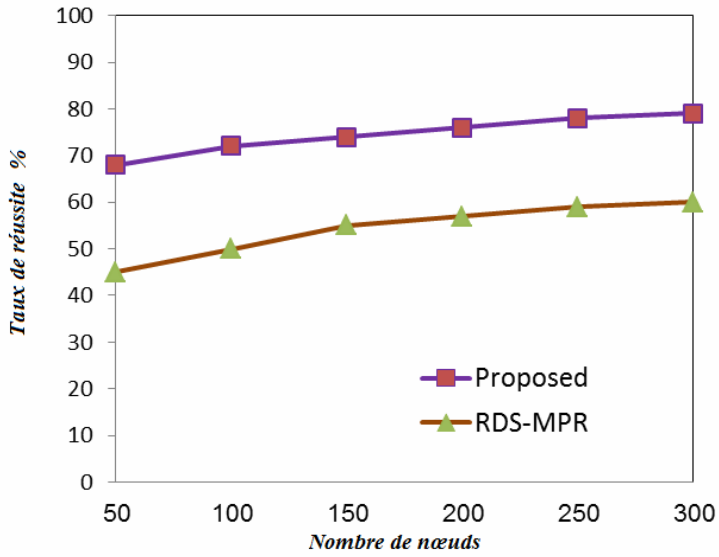


Figure 5.12. *Taux de réussite de RRB comparé à RDS-MPR pour un taux de défaillance fixé à 20% et un nombre de nœuds qui varie de 50 à 300*

Dans la figure 5.12, on fixe le taux des nœuds défaillants à 20% du nombre total des nœuds dans la zone de déploiement, et on fait varier le nombre total des nœuds-capteurs de 50 à 200 avec un pas de 50.

On remarque que le résultat précèdent est confirmé selon ce nouveau scenario de simulation. En effet, le taux de succès de notre technique améliore nettement celui de RDS-MPR.

L'énergie consommée pour sélectionner les nœuds de relai en utilisant RDS-MPR ou en utilisant RRB est la même puisque le même nombre de requêtes a est réalisé pour collecter les données des nœuds. Les données collectées sont :

- La position du nœud pour faire le calcul de la probabilité de réception.
- L'énergie restant dans les nœuds.

Le degré d'un nœud est déterminé selon le nombre de réponses aux messages de requêtes.

Pour comparer l'énergie consommée pour réaliser la sélection des nœuds on considère seulement l'énergie pour faire les communications, puisque l'énergie de traitement interne est négligeable devant l'énergie nécessaire pour faire les communications.

5.4. Conclusion

Dans ce chapitre on a présenté nos contributions suivant deux approches :

- La diffusion en utilisant un *backbone* qui garantit l'équilibrage de charge entre les entités du RCSF.
- La diffusion qui tient compte du facteur de défaillance de nœuds en introduisant un indice de confiance pour chaque nœud.

Nos deux contributions ont été validées via une étude par simulation.

Dans RRB on a introduit pour la première fois l'utilisation d'une méthode de *data mining* (FCA) pour améliorer la diffusion dans les RCSF. Cette technique nous a aidé à faire une sélection plus logique comparée à la fonction linéaire. En effet, dans la sélection à travers les fonctions linaires sommant plusieurs poids, les poids faibles seront cachés par les poids forts. Au contraire FCA exige un certain seuil pour chaque poids.

La mise à jour et le suivi des indices de confiance des nœuds peuvent être améliorés dans RRB par la mise en place d'une organisation en clusters où chaque *cluster-head*, selon le comportement des nœuds, recalcule l'indice de confiance de ses nœuds *cluster-member*. Cette amélioration donne un résultat plus réaliste à la méthode RRB.

Conclusions et perspectives

Au cours des dernières années, le développement des réseaux de communication sans fil, a connu un essor important grâce aux avancées technologiques dans divers domaines tels que la micro-électronique, l'informatique et les systèmes de communication. C'est ainsi que de nouvelles voies d'investigation ont été ouvertes avec l'émergence des réseaux de capteurs sans fil. Ces réseaux sont composés d'un nombre assez important de petits appareils fonctionnant de façon autonome et communiquant entre eux à travers des voies de transmission de courte portée.

Les nœuds capteurs sont conçus pour être déployés d'une manière dense et arbitraire dans des zones géographiques non contrôlées ou des environnements ouverts ou des endroits hostiles et difficiles d'accès ou encore sur des éléments mobiles offrant un aspect dynamique au réseau. De ceci découle la nécessité de limiter au maximum leurs dimensions physiques qui s'obtiennent au détriment des capacités de calcul, de traitement et de ressources énergétiques. Pour cela, ils sont conçus pour être peu gourmands en énergie, en capacité de calcul et en bande passante. On assiste ces derniers temps à des défis de plus en plus importants qu'il faut relever à chaque fois. Il s'agit en particulier des tâches d'organisation, de communication entre les capteurs et des services rendus.

En effet, les capteurs intelligents et autonomes permettent d'observer, d'analyser, de contrôler les phénomènes physiques sur des zones étendues. Ils convertissent les données collectées en format numérique exploitable directement par les systèmes informatiques et les applications associées.

On s'est intéressé dans ce manuscrit à la tâche de diffusion dans un RCSF dynamique comme étant une tâche essentielle puisqu'elle est nécessaire pour la dissémination des données, la localisation des nœuds, la réalisation de routage et la synchronisation du temps. L'aspect dynamique d'un RCSF est pris en considération parce qu'il peut être considéré comme une des caractéristiques essentielles d'un réseau déployé dans des zones géographiques non contrôlées, tout en prenant en compte la possibilité de panne ou d'épuisement d'énergie des capteurs.

C'est dans cette optique qu'on a travaillé en développant les points suivants:

- La diffusion pondérée RDS-MPR qui constitue une amélioration de la diffusion multi-sauts. En effet, on a défini des méthodes permettant de sélectionner les nœuds "relais" selon des poids qui intègrent les caractéristiques des nœuds dans le réseau. On obtient ainsi un aspect plus réaliste de notre méthode et on augmente nettement son taux de succès.

- L'approche clustérisée qui permet de partitionner le réseau en zones, est une approche prometteuse. Pour atteindre cet objectif, on a proposé la diffusion Muli-clusters MCR qui est une amélioration du relai multipoints par l'absence de redondances et la prise en compte de l'aspect dynamique du RCSF. La méthode MCR minimise l'énergie totale consommée pendant la diffusion. Par ailleurs on a pris en considération l'aspect dynamique du réseau d'une façon autonome.

- La diffusion avec équilibrage de charge en développant la méthode LB(k,r)-CDS permettant de sélectionner un *backbone* virtuel basé sur les ensembles dominants connexes et offrant au maximum l'équilibrage de charge dans le réseau, ce qui permet de prolonger la longévité du réseau par rapport à un *backbone* de taille minimale ((k,r)-CDS).

- La diffusion tolérante aux fautes en utilisant une méthode de *data-mining* basée sur le FCA permettant de faire une meilleure sélection des nœuds relais avec la prise en compte de l'aspect tolérance aux fautes. La méthode proposée améliore nettement le taux de succès de la diffusion malgré l'introduction de fautes dans certains nœuds du réseau.

Concernant les perspectives, les performances des méthodes de diffusion dans les RCSF peuvent être encore améliorées par l'introduction des méthodes de *data-mining*. Ces méthodes sont utilisées fréquemment dans le domaine du génie logiciel. Cependant elles sont moins étudiées dans le domaine des réseaux informatiques et en particulier dans les réseaux sans fil et Ad hoc.

Notons aussi que l'intégrité, la sécurité et l'authentification des données diffusées constitue l'un des problèmes qui restent à résoudre. Pour cela, l'utilisation des méthodes de *watermarking* semble être une solution prometteuse pour ce type de problème. En effet, les méthodes de *watermarking* permettent d'authentifier les données avec des techniques plus légères en complexité comparées aux méthodes de sécurité ordinaires.

Enfin, on peut dire que, d'une part, les réseaux de capteurs sans fil rencontrent plusieurs autres problèmes qui affectent leur bon fonctionnement dû à leurs caractéristiques, telles que les limitations en énergie, le type de communication, les environnements hostiles où sont déployés les capteurs ou encore leur faible coût.

D'autre part, ils sont caractérisés par les pannes des nœuds qui peuvent causer un dysfonctionnement du réseau entier. Dans cette optique, il est important de proposer des protocoles de routage tolérants aux fautes. Cet aspect est fondamental pour ce genre de réseau où le routage se réalise en collaboration avec les différents nœuds du réseau. De ce fait, un protocole de routage doit prendre en compte les contraintes matérielles d'un capteur : une énergie faible, une capacité de stockage modeste, une bande passante réduite, etc.

Pour finir cette conclusion on peut dire que les RCSF ont tendance à bouleverser le quotidien de l'être humain. Nous allons totalement vers un monde "sensorisé". Rien qu'avoir les applications de contrôle ou de tracking dans les automobiles, ou même dans les domaines militaire, médical, social, environnemental, on peut juger de l'importance et de l'intérêt qu'il faut accorder aux RCSF, et aux sujets de recherche qui leur sont associes.

Publications

Journaux

1. Moulahi, T., Guyennet, H, & Nasri, S., Improvement of RDS-MPR by a cluster based energy efficient broadcasting for dynamic WSNs, *Journal of ad hoc & sensor wireless networks*, Accepted in 2013 (To appear in March 2015).

2. Moulahi, T., Nasri, S., & Guyennet, H. (2012). Broadcasting based on dominated connecting sets with MPR in a realistic environment for WSNs & ad hoc. *Journal of Network and Computer Applications*, *35*(6), pp. 1720-1727.

3. Ghadi, M., Laouamer L., Moulahi, T. (2015). Securing data exchange in wireless multimedia sensor networks: perspectives and challenges, Multimedia Tools and Applications, pp. 1-26.

Conférences internationales

1. Moulahi, T., Guyennet, H. & Nasri, Maximizing clusters lifetime in Mobile Wireless Sensor Networks. To *The international Conference on Management of computational and collective intelligence in DigitalEcoSystems (MEDES'14)* (Accepted in 2014).

2. Moulahi, T., Nasri, S., & Guyennet, H. (2014, May). Fault tolerant flooding through formal concept analysis for WSNs. In *2014 International Conference on Telecommunications (ICT),* pp. 221-225.
3. Moulahi, T., Guyennet, H., Nasri, S., & Hajlaoui, R. (2012, December). On the construction of load-balanced (k, r-hop)-connected dominating set for WSNs. In *2012 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS),* Bengalore, India, pp. 76-80.
4. Moulahi, T., Nasri, S., & Guyennet, H. (2012, March). Amelioration of MPR by a backbone-based broadcasting algorithm for WSNs. In *2012 International Conference on Information Technology and e-Services (ICITeS)* , Sousse, Tunisia, pp. 1-5.
5. Moulahi, T., Guyennet, H., Lehsaini, M., & Nasri, S. (2011, October). An energy aware MPR-based broadcasting algorithms for wireless sensor networks. In *2011 International Conference on Selected Topics in Mobile and Wireless Networking (iCOST*), Shanghai, China, pp. 13-18.

Articles soumis (à soumettre)

1. Moulahi, T., Touil, S., Guyennet, H. & Nasri, Reliable Relay-based Broadcasting through Formal Concept Analysis for WSNs . *submitted To IEEE sensors journal.*

Références

1. Abbasi, A. A., & Younis, M. (2007). A survey on clustering algorithms for wireless sensor networks. *Computer communications*, *30*(14), pp. 2826-2841.
2. Adjih, C., Jaquet, P., and Viennot. L., (2005). "Computing Connected Dominated Sets with Multi-point Relays", *International Journal Ad Hoc & Sensor Wireless Networks*, vol.1no.1-2, pp. 27-39.
3. Akyildiz, I. F., Su, W., Sankarasubramaniam, Y., & Cayirci, E. (2002). Wireless sensor networks: a survey. *Computer networks*, 38(4), pp. 393-422.
4. Alpar Juttner and Adam Magi, (2005). Tree Based Broadcast in Ad Hoc Networks, *Mobile Networks and Applications*, v.10 n.5, pp.753-762.
5. Arduino : an open-source electronics prototyping platform , [online] Available: http://www.arduino.cc/
6. Avinash Srinivasan and Jie Wu, (2008). Secure and reliable broadcasting in wireless sensor networks using multi-parent trees, *Security And Communication Networks*. v. 2, pp. 293-253.
7. Bergmann, N. W., Wallace, M., & Calia, E. (2010, December). Low cost prototyping system for sensor networks. In *2010 Sixth International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP*), Brisbane, QLD, Australia, pp. 19-24.
8. Boyinbode, O.; Hanh Le; Mbogho, A.; Takizawa, M.; Poliah, R., (2010, September). A Survey on Clustering Algorithms for Wireless Sensor Networks, *2010 13th International Conference on Network-Based Information Systems (NBiS)*, Takayama, Japan, pp. 358-364.
9. Camtepe, S. A., & Yener, B. (2005). Key distribution mechanisms for wireless sensor networks: a survey. *Rensselaer Polytechnic Institute, Troy, New York, Technical Report*, 05-07.
10. Chan Zheng, Yiqing Zhang and Ling Yin, (2011). Constructing (k,r)-Connected Dominating Sets for Robust Backbone in Wireless Sensor Networks, *11th International Symposium on Communications and Information Technologies*, Hangzhou, China, pp. 174-177.
11. Chatterjee, M., Das, S. K., and Turgut, D., (2002). WCA: A Weighted Clustering Algorithm for Mobile Ad Hoc Networks, *Clustering Computing*, vol. 5, pp. 193–204.

12. Chen, X. and Shen, J., (2004, May). Reducing Connected Dominating Set Size with Multipoint Relays in Ad Hoc Wireless Networks, *Proceedings. 7th International Symposium on Parallel Architectures, Algorithms and Networks*, 2004., pp. 539–43.
13. Clausen, T., Jacquet, P., Adjih, C., Laouiti, A., Minet, P., Muhlethaler, P., & Viennot, L. (2003). Optimized link state routing protocol (OLSR). http://hal.inria.fr/docs/00/47/17/12/PDF/5145.pdf
14. De Meulenaer, G., Gosset, F., Standaert, O. X., and Pereira, O. (2008, October). On the energy cost of communication and cryptography in wireless sensor networks. In Networking and Communications, 2008. *WIMOB'08. IEEE International Conference on Wireless and Mobile Computing,* Avignon, France, pp. 580-585.
15. De Souza, Luciana Moreira Sá, Harald Vogt, and Michael Beigl. (2007). "A survey on fault tolerance in wireless sensor networks." *Sap research, braunschweig*, germany.
16. Elson, J., Girod, L., & Estrin, D. (2002). Fine-grained network time synchronization using reference broadcasts. *ACM SIGOPS Operating Systems Review*, *36*(SI), pp. 147-163.
17. Etude du cabinet IDTechEx, http://www.idtechex.com/research/reports/wireless-sensor-networks-wsn-2012-2022-forecasts-technologies-players-000314.asp
18. Feng Wang, My T. Thai and Ding-Zhu Du, (2009). On the Construction of 2-Connected Virtual Backbone in Wireless Networks, *IEEE Transactions on wireless communications*, pp.1230-1237.
19. Francois Ingelrest and David Simplot-Ryl. (2006). Maximizing the Probability of Delivery of Multipoint Relay Broadcast Protocol in Wireless Ad Hoc Networks with a Realistic Physical Layer, *Mobile Ad hoc and Sensor Networks, Lecture Notes in Computer Science (Springer)*, pp.143-154.
20. Galois lattice interactive constructor, http://www.iro.umontreal.ca/~galicia/ goals.html
21. Gao, X., Wang, W., Zhang, Z., Zhu, S., Wu, W., (2009). A PTAS for minimum d-hop connected dominating set in growth-bounded graphs, *Optimization Letters* 4 (3), pp. 321–333.
22. Guha, S., and Khuller, S., (1998, Apr). Approximation algorithms for connected dominating sets, *Algorithmica*, 20(4), pp. 374-387.

23. Hai Liu, Amiya Nayak, Ivan Stojmenović. (2009). "Fault Tolerant Algorithms/Protocols in Wireless Sensor Networks". *Computer Communications and Networks*, Springer, pp. 261-291.
24. Jeffrey E. Wieselthier, Gam D. Nguyen and Anthony Ephremides, (2000). On the Construction of Energy-Efficient Broadcast and Multicast Trees in Wireless Networks, *IEEE INFOCOM 2000*, Tel Aviv, pp. 585-594.
25. Jiguo Yu, Nannan Wang, Guanghui Wang, Dongxiao Yu, (2013, January). Connected dominating sets in wireless ad hoc and sensor networks – A comprehensive survey, *Computer Communications*, Volume 36, Issue 2, pp. 121-134.
26. Jing He, Shouling Ji, Pingzhi Fan, Yi Pan, and Yingshu Li, (2012). Constructing a Load-Balanced Virtual Backbone in Wireless Sensor Networks*, ICNC 2012*, Maui, Hawaii, USA, pp. 959-963.
27. Jing He, Shouling Ji, Mingyuan Yan, Yi Pan, and Yingshu Li, (2012). Load-Balanced CDS Construction in Wireless Sensor Networks Via Genetic Algorithm*, International Journal of Sensor Networks (IJSNET)*, Vol. 11, No. 3, pp. 166-178.
28. Jing He, Shouling Ji, Mingyuan Yan, Yi Pan, and Yingshu Li, (2011). Genetic-Algorithm-Based Construction of Load-Balanced CDSs in Wireless Sensor Networks, *MILCOM 2011*, Baltimore, Maryland, USA, pp. 667-672.
29. Kalantri, M., Haghpanahi, M., and Shayman. M., (2008). A p-norm Flow Optimaziation Problem in Dense Wireless Sensor Network, *INFOCOM*, 2008, Phoenix, AZ, USA, pp. 341-345.
30. Kuruvila, J.; Nayak, A.; Stojmenovic, I., (2005) Hop count optimal position-based packet routing algorithms for ad hoc wireless networks with a realistic physical Layer, *IEEE Journal on Selected Areas in Communications*, vol.23, no.6, pp. 1267-1275.
31. Le capteur MicaZ, http://www.openautomation.net/uploadsproductos/micaz_datasheet.pdf
32. Le capteur Telosb, http://www.memsic.com/userfiles/files/Datasheets/WSN/telosb_datasheet.pdf
33. Le projet Sensorscope, http://www.sensorscope.ch/fr/
34. Le projet Libelium, http://www.libelium.com/

35. Lehsaini, M., Guyennet, H., & Feham, M. (2007). MPR-based broadcasting in ad hoc and wireless sensor networks with a realistic environment. *International Journal of Computer Science and Network Security (IJCSNS)*, pp. 82-89.
36. Les réseaux de capteur, note de veille CAPTIVEN, Juillet 2013, http://www.captiven.fr/sites/default/files/pictures/Note%20de%20veille_R%C3%A9seaux decapteurs_Tech2Market_finale2.pdf
37. Li, D., Liu, L., and Yang, H., (2009). Minimum connected r-hop k-dominating set in wireless networks, *Discrete Mathematics, Algorithms and Applications* 1 (1), pp. 45–57.
38. Liang, O., Ahmet Sekercioglum, Y., and Nallasamy Mani. (2007). Gateway multipoint relays an-MPR-based broadcast algorithm for ad hoc networks, *IEEE Singapore International Conference on Communication systems*. Singapore, pp. 1-6.
39. Liang, O., Sekercioglu, Y. A., & Mani, N. (2006). A survey of multipoint relay based broadcast schemes in wireless ad hoc networks. *IEEE Communications Surveys and Tutorials*, *8*(1-4), pp. 30-46.
40. Liste de quelques capteurs
http://en.wikipedia.org/wiki/ List_of_wireless_sensor_nodes
41. Nazanin Rahnavard, Badri N. Vellambi and Faramarz Fekri, (2008). Distributed Protocols for Finding Low-Cost Broadcast and Multicast Trees in Wireless Networks, *5th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks*, San Francisco, CA, USA, pp. 551-559.
42. Mahmood, A., Shi, K., Khatoon, S., & Xiao, M. (2013). Data Mining Techniques for Wireless Sensor Networks: A Survey. *International Journal of Distributed Sensor Networks*, 2013, pp.1-24.
43. Mans, B., & Shrestha, N. (2004, June). Performance evaluation of approximation algorithms for multipoint relay selection. In *Proceedings of the 3rd Annual Mediterranean Ad Hoc Networking Workshop*, Bodrum, Turkey, pp. 480-491.
44. Moulahi, T., Guyennet, H., Nasri, S., & Hajlaoui, R. (2012, December). On the construction of load-balanced (k, r-hop)-connected dominating set for WSNs. In *2012 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS),* Bengalore, India, pp. 76-80.

45. Moulahi, T., Guyennet, H., Lehsaini, M., & Nasri, S. (2011, October). An energy aware MPR-based broadcasting algorithms for wireless sensor networks. In *2011 International Conference on Selected Topics in Mobile and Wireless Networking (iCOST*), Shanghai, China, pp. 13-18.
46. Moulahi, T., Guyennet, H, & Nasri, S., (2013). Improvement of RDS-MPR by a cluster based energy efficient broadcasting for dynamic WSNs, *Journal of ad hoc & sensor wireless networks*, Accepted in 2013.
47. Moulahi, T., Nasri, S., & Guyennet, H. (2012). Broadcasting based on dominated connecting sets with MPR in a realistic environment for WSNs & ad hoc. *Journal of Network and Computer Applications*, *35*(6), pp. 1720-1727.
48. Moulahi, T., Nasri, S., & Guyennet, H. (2012, March). Amelioration of MPR by a backbone-based broadcasting algorithm for WSNs. In *2012 International Conference on Information Technology and e-Services (ICITeS)* , Sousse, Tunisia, pp. 1-5.
49. Moulahi, T., Nasri, S., & Guyennet, H. (2014, May). Fault tolerant flooding through formal concept analysis for WSNs. In *2014 International Conference on Telecommunications (ICT),* pp. 221-225.
50. Pai, S., Rieck, M., and Dhar, S., (2005). Distributed routing algorithms for wireless ad hoc networks using d-hop connected d-hop dominating sets, *Computer Networks* 47 (6), pp. 221–225.
51. Perkins, C.; Belding-Royer, E.; Das, S. (July 2003). Ad hoc On-Demand Distance Vector (AODV) Routing. https://tools.ietf.org/html/rfc3561
52. Qayyum, A., Viennot, L., & Laouiti, A. (2002, January). Multipoint relaying for flooding broadcast messages in mobile wireless networks. In *Proceedings of the 35th Annual Hawaii International Conference on System Sciences, 2002. HICSS.*, Hawaii , USA, pp. 3866-3875.
53. Sivaraman, E. (2010, December). Dynamic cluster broadcasting for mobile ad hoc networks. In *2010 International Conference on Communication and Computational Intelligence (INCOCCI),* Erode, India, pp. 123-127.
54. Visvanathan, A.; Jong-Hoon Youn; Jitender Deogun, (2005, May). Hierarchical data dissemination scheme for large scale sensor networks*, 2005 IEEE International Conference on Communications*, Seoul, Korea, pp. 3030-3036, Vol. 5.

55. Ye, F., Luo, H., Cheng, J., Lu, S., and Zhang, L., (2002). A Two-Tier Data Dissemination Model for Large-scale Wireless Sensor Networks, *ACM International Conf. on Mobile Computing and Networking (Mobicom 2002)*. New York, USA, pp. 148-159.
56. Ye, M., Li, C., Chen, G., and Wu, J., (2006). An Energy Efficient Clustering Scheme in Wireless Sensor Networks, *Ad Hoc & Sensor Wireless Networks*, Vol.1, pp. 1–21.
57. Yick, J., Mukherjee, B., & Ghosal, D. (2008). Wireless sensor network survey. *Computer networks*, 52(12), pp. 2292-2330.
58. Yiwei Wu, Feng Wang, My T. Thai and Yingshu Li, (2007). Constructing k-connected m-dominating sets in wireless sensor networks, *Military Communications Conference*, Orlando, FL, USA, pp. 1–7.
59. Yiwei Wu and Yingshu Li, (2008). Constructing algorithms for k-connected m-dominating sets in wireless sensor networks, *Proceedings of the 9th ACM international symposium on Mobile ad hoc networking and computing*, Hong Kong, China, pp. 83–90.
60. Younis, O., and Fahmy, S., (2004). "HEED: "A Hybrid Energy-Efficient Distributed Clustering Approach for Ad Hoc Sensor Networks," *IEEE Transactions on Mobile Computing*, vol. 3, no. 4, pp. 366-379.
61. Yu, F., & Jain, P. R. (2009). A Survey of Wireless Sensor Network Simulation Tools. *Washington University in St. Louis, Department of Science and Engineering,* http://www. cse. wustl. edu/~ jain/cse567-11/ftp/sensor/index.html.
62. Yu-Yuan Lin and Kuo-Feng Ssu, (2009). Efficient Fault-Tolerant Backbone Construction in Tmote Sky Sensor Networks*, 15th IEEE Pacific Rim International Symposium on Dependable Computing*, Shanghai, China, pp. 319–323.
63. Wang, X., & Li, J. (2009, December). Precision constraint data aggregation for dynamic cluster-based wireless sensor networks. In *MSN'09. 5th International Conference on Mobile Ad-hoc and Sensor Networks, 2009.* Fujian, China, pp. 172-179.
64. Weiping Shang, Pengjun Wan, Frances Yao, Xiaodong Hu, (2007). Algorithms for minimum mconnected k-tuple dominating set problem, *Theoretical Computer Science*, pp. 241-247.
65. Wolff, Karl Erich, (1994). "A first course in Formal Concept Analysis", in F. Faulbaum, StatSoft'93, Gustav Fischer Verlag, pp. 429–438.

66. Wong, G. K., Liu, H., Chu, X., Leung, Y. W., & Xie, C. (2010). “Efficient broadcasting in multi-hop wireless networks with a realistic physical layer”. *Ad Hoc Networks*. Vol. 11, pp. 1305-1318.
67. Wormuth, B., and Becker, P., (2004). Introduction to Formal Concept Analysis , *2nd International Conference of Formal Concept Analysis*. Sydney, Australia. http://www.wormuth.info/ICFCA04/Introduction_to_FCA_ICFCA2004.pdf
68. Wu, J., (2002, September). “An Enhanced Approach to Determine a Small Forward Node Set Based on Multipoint Relays*,” IEEE Trans. Parallel and Distributed Systems*, pp. 866–81.
69. Wu, J., Lou, W., & Dai, F. (2006). Extended multipoint relays to determine connected dominating sets in MANETs*, IEEE Transactions on Computers*, *55*(3), pp. 334-347.
70. Xianghua XU1, Chunxiong FAN, Jian WAN, Wei ZHANG, Naixue XIONG, (2012). "TIM-SNAP: Fault Tolerant Event Location Estimation in Sensor Network"*,Journal of Computational Information Systems* 8: 8, pp. 3507–3517.
71. Xiuying Li and Zhao Zhang, (2010). Two algorithms for minimum 2-connected r-hop dominating set*, Information Processing Letters*, Volume 110, Issue 22, pp. 986-991.
72. Zhuo Liu, Bingwen Wang and Lejiang Guo, (2010). A Survey on Connected Dominating Set Construction Algorithm for Wireless Sensor Networks. *Information Technology Journal*, 9: pp. 1081-1092.

Résumé

Dans cette thèse, on s'intéresse à la tâche de diffusion dans les réseaux de capteurs sans fil (RCSF) dynamiques. C'est une tâche essentielle et primordiale puisqu'elle est nécessaire pour réaliser la découverte de voisinage, le routage, la distribution d'informations dans tout le réseau, la localisation des nœuds et la synchronisation du temps. L'aspect dynamique du RCSF est pris en considération parce qu'il constitue une des caractéristiques essentielles d'un réseau déployé dans des zones géographiques inconnues et non contrôlées. La possibilité de panne ou d'épuisement d'énergie des capteurs est également prise en compte. Ce manuscrit est composé de deux parties. La première concerne un état de l'art de la diffusion dans les RCSF et sa taxonomie, tandis que dans la deuxième partie, nous présentons nos contributions qui sont la diffusion pondérée RDS-MPR, la diffusion multi-clusters MCR, la diffusion avec équilibrage de charge qui utilise la méthode LB(k,r)-CDS et la diffusion tolérante aux fautes qui utilise une méthode de *data-mining* permettant d'améliorer le taux de succès malgré l'introduction de fautes dans certains nœuds du réseau.

Mots clés Réseaux de capteurs dynamiques sans fil; Diffusion; Backbone; Relai; Ensembles dominants connexes; Equilibrage de charge; Tolérance aux fautes

Abstract

We are interested by the broadcasting task in dynamic WSNs as an essential and primordial task since it is used to achieve the neighboring discovery, routing, data dissemination in the network, nodes localization and time synchronization. The dynamic behavior of WSNs is taken into consideration. Such an aspect can be considered as one of essential characteristics of a network deployed in unknown or unmonitored geographic zone, while taking into account the possibility of failure or nodes energy exhaustion. This manuscript is composed of two parts. The first one is a state of the art of broadcasting in WSNs and its taxonomy, while in the second part we give our contributions which are the weighted broadcasting RDS-MPR, The multi-clusters broadcasting, the load-balanced broadcasting by introducing LB(k,r)-CDS method, and the fault tolerant broadcasting through a data mining method which has enhanced the success ratio of broadcasting in spite of introducing some failure nodes in the network.

Key Words : Dynamic Wireless Sensor Networks; Broadcasting; Backbone; Relay; Connected Dominating Sets; Load-balancing; Fault tolerance

- Ecole doctorale SPIM 16 route de Gray F – 25030 Besançon Cedex, France
- Tel. +33 (0) 3 81 66 66 02 Ed-spim@univ-fcomte.fr http://www.ed-spim.univ-fcomte.fr
- Ecole Doctorale Sciences et Technologie ENIS, Route de Soukra Km 2.5, BP. 1173 - 3038, Sfax, Tunisie
- Tel : +216 74 274 60 - 74 274 418 Fax : +216 74275 595 contact.edst@gmail.com http://www.enis.rnu.tn/EDST