

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université des Sciences et de la Technologie Houari Boumediene

Faculté d'Electronique et Informatique



Mémoire
Présenté pour l'obtention du diplôme de MAGISTER
En : INFORMATIQUE
Spécialité : Informatique Mobile

Par : Hayet GHRIBI

Sujet

**Traitement de pannes isolées dans les
réseaux de capteurs et de robots sans fil
(WSRN)**

Soutenu publiquement le 12/01/2015 devant le Jury Composé de :

| | |
|---|----------------------|
| Mr Abdelmadjid Boukra, Professeur - USTHB | Président, |
| Mr Mahfoud Benchaïba, Maitre de Conférence/A à l'USTHB | Directeur de Mémoire |
| Mr Youcef Zaffoune, Maitre de Conférence/A à l'USTHB | Examineur, |
| Melle Chafika Benzaïd, Maitre de Conférence/B à l'USTHB | Invitée, |

Dédicace

Je dédie ce modeste travail à

mes parents,

mon mari,

mes enfants,

mes frères et sœurs

Remerciements

Il est difficile de rédiger cette page tant qu'il ne faut oublier personne! Alors, je commence par mes excuses à toutes les personnes qui m'ont aidé et leurs noms ne figurent pas dans cette page. Vous êtes nombreux, je suis très reconnaissante, je vous dis: Merci!

- *Je tiens à remercier mon directeur de recherche Mahfoud BENCHAÏBA pour ses conseils et sa patience. Ses remarques constructives étaient le fil directeur et le secret qui ont permis l'aboutissement de ce travail.*
- *Je tiens à remercier les membres du jury d'avoir acceptés de juger mon mémoire de Magister. Certainement, leurs remarques et le débat seront fructueux pour moi, mais aussi, à l'enrichissement de mon travail de recherche.*
- *Je tiens à remercier tous les enseignants (les professeurs et les experts externes) de notre promotion pour leurs efforts tout au long de notre formation.*
- *Je tiens à remercier mes amies et les collègues de l'USTHB pour le soutien moral et l'aide surtout pendant la réalisation de mini – projet de Magister.*

Sommaire

| | |
|---|-----|
| Liste de figures | i |
| Liste de tables | iii |
| Introduction générale | 1 |
| Chapitre 1: | |
| Redéploiement d'un réseau de capteurs sans fil en utilisant les robots | |
| 1. Introduction..... | 4 |
| 2. Réseau de capteurs sans fil..... | 4 |
| 2.1 Définitions..... | 4 |
| 2.2. Critères de classements des réseaux de capteurs..... | 5 |
| 2.3 Caractéristiques des RCSFs..... | 6 |
| 2.4 Domaines d'applications..... | 7 |
| 2.4.1 Applications militaires..... | 7 |
| 2.4.2 Applications dans l'environnement..... | 8 |
| 2.4.3 Applications dans le domaine médical..... | 8 |
| 2.4.4 Applications dans les accueils et les maisons..... | 8 |
| 2.5 Critères de performances dans la conception d'un réseau de capteurs..... | 8 |
| 2.5.1 Tolérance aux pannes..... | 8 |
| 2.5.2 Topologie maintenable..... | 8 |
| 2.5.3 Scalabilité..... | 8 |
| 2.5.4 Gestion d'accès au Support de transmission..... | 10 |
| 2.5.5 Les contraintes matérielles..... | 10 |
| 2.5.6 L'efficacité énergétique..... | 10 |
| 2.5.7 Latence | 10 |
| 2.5.8 Précision..... | 10 |
| 2.5.9 Le coup de production..... | 10 |
| 2.6 Capteur..... | 10 |
| 2.6.1 Caractéristiques d'un capteur..... | 11 |
| 2.6.2 La pile protocolaire implémentée au niveau d'un capteur..... | 11 |
| 2.6.3 Les sources de perte d'énergie d'après Yaser. Y, (2010)..... | 15 |
| 2.7 Durée de vie d'un réseau de capteurs..... | 16 |
| 2.7.1 Définition générale..... | 16 |
| 2.7.2 Quelques définitions existantes..... | 16 |
| 2.8 Importance de la couverture et la connectivité dans un RCSF..... | 17 |
| 2.8.1 Connectivité..... | 17 |
| 2.8.2 Couverture..... | 17 |
| 2.8.3 Relation entre la couverture et la connectivité..... | 17 |
| 3. Panne d'un capteur..... | 17 |
| 3.1 Les pannes | 17 |
| 3.2 Causes d'une panne..... | 18 |
| 3.3 Durée d'une panne | 18 |
| 3.4 Comportant résultant d'une panne..... | 19 |
| 3.4.1 Pannes selon le comportement résultant (Challal. Y, 2008) | 19 |
| 3.4.2 Panne isolée..... | 19 |

| | |
|--|----|
| 4. Les différentes phases de déploiement d'un RCSF..... | 21 |
| 4.1 Phase de déploiement..... | 21 |
| 4.2 Phase de pré-déploiement et de déploiement | 21 |
| 4.3 Phase de post-déploiement | 23 |
| 4.4 Phase de redéploiement des nouveaux nœuds..... | 23 |
| 5. Les robots | 24 |
| 5.1 Types de robots utilisés..... | 24 |
| 5.2 Exemples de robots..... | 25 |
| 5.2.1 Le robot "AVATAR" | 25 |
| 5.2.2 Le robot "AMOUR" d'après Falcon Martinez. R-J, (2012)..... | 25 |
| 5.2.3 Robots de systèmes de jardinage..... | 26 |
| 6. Le redéploiement utilisant les robots | 27 |
| 7. Conclusion..... | 29 |

Chapitre 2:

Classification des approches de traitement de pannes isolées utilisant les robots

| | |
|--|----|
| 1. Introduction..... | 30 |
| 2. Nature et acteurs des décisions pour le redéploiement des pannes isolées utilisant les robots | 31 |
| 2.1 Etapes générales de traitement de pannes | 31 |
| 2.1.1 La détection de la panne | 32 |
| 2.1.2 La réparation..... | 32 |
| 2.2 Défis existants..... | 33 |
| 3. Classification des approches de traitement de pannes isolées | 34 |
| 3.1 Critères de Classification..... | 34 |
| 3.2 La classification selon l'équilibre de charge (Falcon. M, 2012)..... | 34 |
| 3.2.1 Les algorithmes centralisés | 34 |
| 3.2.2 Les algorithmes distribués | 35 |
| 3.2.3 Les algorithmes localisés..... | 35 |
| 3.3 Classification de solutions de traitement des pannes isolées | 35 |
| 3.3.1 Les capteurs détectent les pannes | 35 |
| 3.3.1.1 Sans exploration | 36 |
| 3.3.1.2 Avec exploration | 45 |
| 3.3.1.3 Synthèse..... | 52 |
| 3.3.2 Le(s) sink(s) évalue(nt) les pannes..... | 53 |
| 3.3.3 Le (s) robot(s) évalue(nt) les pannes..... | 55 |
| 4. Modèles de réparation | 60 |
| 4.1 Liste FIFO (First In First Out) | 60 |
| 4.2 Recherche du plus court chemin..... | 60 |
| 5. Discussion..... | 62 |
| 6. Conclusion | 63 |

Chapitre 3:

Directing Robots to Failures using Lists (DRFL)

| | |
|-----------------------|----|
| 1. Introduction | 64 |
|-----------------------|----|

| | |
|--|----|
| 2. Problématique et objectifs | 66 |
| 3. Notre solution | 66 |
| 3.1 Description de l'environnement de déploiement | 67 |
| 3.2 Description générale..... | 67 |
| 3.3 Apports de notre solution..... | 71 |
| 3.3.1 Le pas de déplacement..... | 71 |
| 3.3.2 Utilisation des listes pour la redirection des robots..... | 72 |
| 3.3.3 Utilisation des listes d'urgence..... | 73 |
| 3.4 Structure de données utilisées..... | 73 |
| 3.4.1 Messages..... | 74 |
| 3.4.2 Les différentes listes utilisées..... | 74 |
| 3.5 Mécanismes de base..... | 74 |
| 3.5.1 Calcul de point de la grille..... | 75 |
| 3.5.2 Attribution des rôles..... | 77 |
| 3.5.3 Calcul des directions permises..... | 78 |
| 3.5.4 Détection locale de la panne..... | 80 |
| 3.5.5 Routage géographique..... | 81 |
| 3.5.6 Méthode de mise à jour des listes..... | 82 |
| 3.6 Algorithme générale | 82 |
| 3.6.1 | |
| Initialisation..... | 83 |
| 3.6.2 Détection | 83 |
| 3.6.2.1 Détection des pannes | 83 |
| 3.6.2.2 Gestion des pannes d'urgence..... | 85 |
| 3.6.3 Information et redirection des robots..... | 86 |
| 3.6.4 Réparation | 88 |
| 3.6.5 Reprise | 88 |
| 4. La gestion des cas d'exceptions..... | 88 |
| 5. Discussions..... | 89 |
| 6. Conclusion | 89 |

Chapitre 4:

Evaluation des performances du protocole DRFL.

| | |
|--|-----|
| 1. Introduction..... | 91 |
| 2. Outils de simulation..... | 92 |
| 2.1 Le simulateur Omnet..... | 92 |
| 2.2 INET 2.4.01..... | 92 |
| 3. Méthodologie de simulation..... | 93 |
| 3.1 Implémentation de la méthode..... | 93 |
| 3.1.1 Implémentation des nœuds..... | 94 |
| 3.1.2 Etapes d'exécution de DRFL | 96 |
| 3.1.2 Génération des pannes et envois des rapports de pannes | 98 |
| 3.1.3 L'information du robot et la mise à jour des listes..... | 99 |
| 3.2 Paramètres de simulation utilisés | 99 |
| 3.3. Métriques de performance..... | 100 |
| 3.3.1 Nombre des points de la grille pour une zone de déploiement..... | 100 |
| 3.3.2 Temps de réparation moyen..... | 101 |

| | |
|--|------------|
| 3.3.3 Nombre de messages supplémentaires..... | 102 |
| 4. Résultats et interprétations..... | 102 |
| 4.1 Impact du rayon de communication d'un robot sur le nombre de points de la grille..... | 102 |
| 4.2 Impact de RCR sur le temps de réparation moyen d'une panne..... | 103 |
| 4.3 Impact du nombre de robots sur le temps de réparation moyen..... | 104 |
| 4.4 Messages supplémentaires..... | 105 |
| 5. Discussion | 106 |
| 6. Conclusion | 106 |
| Conclusion générale..... | 107 |
| Bibliographie..... | 109 |

Liste de Figures

| | |
|--|----|
| Figure 1.1: Architecture d'un réseau de capteurs (Challal.Y, 2008)..... | 5 |
| Figure 1.2: Les différentes applications des réseaux de capteurs (Challal.Y, 2008).... | 7 |
| Figure 1.3: Anatomie d'un réseau de capteurs (Challal.Y, 2008)..... | 11 |
| Figure 1.4: La pile protocolaire d'un réseau de capteur Akyildiz,W (2001)..... | 13 |
| Figure 1.5: Représentation d'une panne massive (Younis. B et Lee. S, 2014)..... | 20 |
| Figure 1.6: Exemple d'une panne isolée..... | 20 |
| Figure 1.7: Ordonnancement d'activités avec le critère de la couverture de surface...22 | |
| Figure 1.8.: AVATAR : hélicoptère autonome avec interface de capteurs pour le déploiement et le redéploiement des capteurs d'après Corke.P et al, (2006),25 | |
| Figure 1.9: Un réseau déployé par le AMOUR. | 26 |
| Figure 1.10 : Aquanodes drying..... | 26 |
| Figure 1.11: Robots tending to cherry tomato plants in the greenhouse..... | 27 |
| Figure 1.12: Un robot qui récolte de la tomate..... | 27 |
| Figure 1.13: Phase de redéploiement utilisant les robots dans le processus de déploiement pendant la durée de vie de RCSF..... | 28 |
| Figure 2.1: Prise de décision..... | 31 |
| Figure 2.2: Processus général de traitement de pannes selon le mode "sans exploration"..... | 36 |
| Figure 2.3: Création de chemins de routage..... | 38 |
| Figure 2.4: Subdivision de la zone de couverture en sous zones..... | 41 |
| Figure 2.5: Un graphe véronique dynamique..... | 42 |
| Figure 2.6: Stratégie de traitement des pannes dans l'approche de Hnin.Y et al (2008)..... | 44 |
| Figure 2.7: Déploiement déterministe des capteurs s_i, s_k, s_j qui assurent la couverture d'après Chang.C et al (2007)..... | 49 |
| Figure 2.8: Mouvement spirale d'après Chang. C et al (2007)..... | 49 |
| Figure 2.9: Architecture de communication balise dans l'approche de Xu.Li. et al(2012)..... | 50 |
| Figure 2.10: Politique de traversée utilisant LRV d'après Selon Fletcher. G, (2010)..51 | |
| Figure 2.11: Exemples de cartes iso-sensing et leurs contours d'après Pengju. Si et al (2012)..... | 56 |
| Figure 2.12: Exemple de trous de couverture extraits de Pengju. Si et al (2012)..... | 56 |
| Figure 2.13: Identification de trou de couverture d'après Yao. S et al (2009)..... | 58 |
| Figure 2.14: Position de redéploiement d'un nouveau capteur d'après Yao. S et al (2009)..... | 58 |
| Figure 2.15: Position du trou à l'aide des nœuds de bordure d'après Yao. S et al (2009)..... | 59 |
| Figure 2.16: Exemple de mouvement d'un robot dans 1-TSP-SELPD avec le nombre maximum de points = 3..... | 61 |
| Figure 2.17 : Le tour d'un véhicule pour attendre tous les points..... | 62 |
| Figure 3.1: Modèle de déplacement des robots..... | 68 |
| Figure 3.2: Rayon de communication d'un robot. | 68 |
| Figure 3.3: Division de la zone d'intérêt en sous zone selon la grille de déplacement des robots..... | 69 |
| Figure 3.4: Attribution des rôles aux capteurs..... | 70 |
| Figure 3.5: Chemin de réparation parcouru par un robot..... | 71 |
| Figure 3.6: Le pas de déplacement..... | 72 |

| | |
|--|-----|
| Figure 3.7: Calcul des numéros d'une sous zone selon l'axe X et l'axe Y | 75 |
| Figure 3.8: Exemple de calcul d'un point de la grille..... | 77 |
| Figure 3.9: Différence entre les directions permises selon la position de la sous zone..... | 78 |
| Figure 3.10: Exemple de calcul des directions permises..... | 79 |
| Figure 3.11: Identification des trous locaux Grid R3S2 (Xu. Li et <i>al</i> 2013)..... | 81 |
| Figure 3.12: Les capteurs A et B détectent la panne du capteur C et envoient les rapports de panne vers les capteurs responsables via le routage géographique..... | 84 |
| Figure 3.13: Etats des listes après l'envoi d'un message d'urgence..... | 85 |
| Figure 3.14: Etat du système avant que le robot diffuse son premier message de signalisation..... | 87 |
| Figure 3.15: Etat du système après que le robot diffuse son premier message de signalisation..... | 87 |
| Figure 4.1: Simple and compound modules..... | 92 |
| Figure 4.2: Architecture d'un capteur..... | 94 |
| Figure 4.3: Architecture d'un robot..... | 95 |
| Figure 4.4: Distribution initiale des capteurs et des robots..... | 97 |
| Figure 4.5: Calcul des directions permises par le capteur 106. | 97 |
| Figure 4.6: Etat du réseau dans $T=235.710290203104$ | 98 |
| Figure 4.7: Réception d'un rapport de panne..... | 98 |
| Figure 4.8: Mises à jour des listes des directions..... | 99 |
| Figure 4.9: Impact de RCR sur le nombre de points dans la grille..... | 103 |
| Figure 4.10: Impact de RCR sur le temps de réparation..... | 104 |
| Figure 4.11: Impact du nombre de robots sur le temps de réparation moyen..... | 105 |
| Figure 4.12: Messages supplémentaires..... | 105 |

Liste de Tables

| | |
|---|-----|
| Table 2.1: Comparaison entre les sous classes du mode sans exploration..... | 45 |
| Table 2.2: Comparaison des modes sans exploration et avec exploration..... | 53 |
| Table 2.3 : Comparaison entre modes en cours et après exploration..... | 60 |
| Table 3.1: Méthode de calcul des directions permises..... | 79 |
| Table 4.1: Paramètres de simulations..... | 100 |

Introduction générale

L'observation est l'une des méthodes utilisée dans plusieurs domaines, notamment, les études sur l'environnement en recherche scientifique, la détection des attaques dans le domaine militaire, la gestion des ressources dans les milieux urbains dans le domaine de la gestion. Le but de l'observation est la collecte des informations pour analyser et comprendre les phénomènes étudiés. Tandis que, certains phénomènes nécessitent la collection des informations pendant une période longue qui peut durer des années et dans un environnement qui peut être inconnu, difficile, critique, dangereux ou mortel pour les êtres humains. Cette situation nécessite le placement des unités qui possèdent une certaine autonomie de traitement et qui s'occupent de l'observation et de la transmission des informations nécessaires aux intéressés. Le réseau de capteurs sans fil, représente un outil efficace pour rassurer cette tâche.

Les capteurs sont des unités de traitement, de captage, et d'échange de messages, dont le rôle est l'observation de son environnement et la transmission des informations collectées vers un ou plusieurs puits. A cause de l'environnement de déploiement des capteurs qui est généralement dangereux et de ses capacités physiques limitées, ces derniers tombent en panne après une certaine période de leurs fonctionnements. Les pannes isolées (individuelles) représentent une des catégories des pannes qui sont dues généralement à la panne matérielle d'un capteur ou de l'épuisement de son énergie. La panne d'un ou plusieurs capteurs dans le réseau influe négativement sur la qualité de service fournie par le réseau de capteurs.

Afin d'éviter les conséquences catastrophiques des pannes, plusieurs stratégies sont appliquées dès le déploiement du réseau et tout au long de sa durée de vie. Par exemple, la stratégie de déploiement dense (dans la phase de déploiement), la relocation (dans la phase de post déploiement), le remplacement des nouveaux nœuds (redéploiement). Dans notre recherche, on est intéressé au redéploiement des nouveaux capteurs comme une méthode de réparation des pannes, cette stratégie va permettre la prolongation de la durée de vie du réseau de capteurs.

En parallèle, les recherches scientifiques en robotique ont permis la production des robots mobiles puissants qui possèdent des ressources d'énergie illimitées. Dans un environnement dur ou inaccessible, les robots sont utilisés dans les différentes phases de déploiement du réseau de capteurs. Le redéploiement utilisant les robots est parmi les mécanismes qui peuvent être utilisés dans un tel environnement pour le traitement des pannes isolées. Dans ce type des solutions, la mission principale des robots est le placement des nouveaux capteurs (redéploiement) aux emplacements calculés des pannes. Mais, avant de réussir le redéploiement, les robots doivent être informés de ces pannes qui sont déjà calculées par les nœuds (soit les capteurs ou les robots), et par la suite ils sont amenés à suivre un algorithme de redéploiement pour réparer ces pannes. Dans ce contexte, plusieurs décisions doivent être prises, afin de réussir le traitement des pannes. Parmi les questions qui peuvent être posées pendant le processus de traitement, on cite les questions suivantes : comment déterminer les emplacements des pannes? Quelles sont les nœuds qui prennent la responsabilité de ce calcul? Comment informer le(s) robot(s) des emplacements des pannes ? Comment les robots procèdent pour réparer ces pannes? Et bien d'autres questions.

La recherche bibliographique dans ce domaine nous a permis d'avoir une idée sur les efforts consacrés à la résolution de ce problème et qui sont représentés dans les différentes solutions proposées. Malheureusement, toutes les solutions apportées restent limitées.

Les étapes du processus de traitement des pannes utilisant les robots diffèrent selon les décisions prise pour chaque niveau. Dans notre classification de solutions connues dans la littérature, ces étapes diffèrent selon la source de détection de panne : Si le robot détecte la panne, l'algorithme de traitement est composé de deux étapes: la détection et la réparation, et si les capteurs(les capteurs ou le sink) détectent la panne, le traitement passe généralement par trois étapes: la détection, l'information de(s) robots des emplacements des pannes et finalement la réparation (le redéploiement des nouveaux capteurs).

Parmi ces solutions, nous avons choisi pour notre contribution le scénario suivant: Les robots explorent la zone de captage afin de communiquer localement avec les capteurs pour s'informer des positions des pannes déjà calculées localement par ces capteurs. Quand un robot est informé des pannes existantes, il se déplace vers les positions des pannes et il place de nouveaux nœuds, quand il termine il continue son mouvement pour découvrir d'autres pannes dans d'autres emplacements. Cette stratégie permet un traitement localisé et distribué des pannes avec la possibilité d'accroître le nombre de robots réparateurs sans modification de la solution. D'après les évaluations effectuées sur ce type des scénarios, le temps de réparation d'une panne et le coût en messages est proportionnellement acceptable. Mais, dans ce type de scénarios, le temps de traitement des pannes et le nombre des messages transmis par les capteurs dépend directement de l'algorithme de direction des robots et le nombre des arrêts effectués par ces derniers.

L'approche adoptée dans notre protocole, intitulé DRFL (*Directing Robots to Failures using List*), consiste à structurer la zone de déploiement sous forme d'une grille de cases carrées dont la taille du côté d'une case est le pas de déplacement d'un robot. Les capteurs de chaque case détectent les pannes à proximité, informent et d'orientent vers ces pannes les robots en nombre déterminé qui circulent de manière permanente dans la région d'intérêt. L'avantage en est l'économie d'énergie des capteurs grâce aux transmissions limitées d'informations. Un certain nombre de mécanismes inspirés de différentes solutions connues dans la littérature ont été utilisés dans le but d'optimiser les déplacements des robots et les communications dans le réseau. Nous citons par exemple : Structure de la zone, l'utilisation du routage géographique, la réparation de pannes dans l'ordre d'apparitions. D'autres mécanismes propres à la solution ont été introduits, nous citons entre autres :

- L'affectation de différents rôles aux capteurs,
- Les listes de pannes prioritaires (urgentes),
- La liste de directions des robots pour attirer les robots vers les zones de pannes et minimiser le temps de réparation maximal d'une panne.
- La minimisation du nombre d'arrêts de robots pour diminuer les communications,
- Etc...

Les performances de la solution ont été évaluées par des séries de simulations (par exemple : l'effet du rayon de communication d'un robot/capteur et nombre de robots sur le temps de réparation de pannes et sur la complexité en messages, le nombre

d'arrêts d'un robot ...). Les résultats de l'évaluation ont montré l'apport de notre solution.

La structure de ce mémoire traduit les différentes étapes de notre recherche. Le Chapitre 1 présente les aspects généraux liés au réseau de capteurs, les pannes, le déploiement, les robots, et l'importance de ces derniers pour la réparation des pannes. Dans le Chapitre 2, on expose les différents algorithmes qui proposent des solutions pour le traitement des pannes isolées dans un réseau de capteurs sans fils utilisant les robots vus dans la littérature, selon une classification que nous avons apportés. Dans le Chapitre 3, on présente les différentes étapes de notre algorithme (DRFL), et les différentes solutions qu'ils proposent pour le traitement des pannes isolées de capteurs. Le Chapitre 4 présente les étapes d'évaluation de notre solution en commençant par son implémentation jusqu'aux tests d'évaluation des performances et les résultats obtenus ainsi que leurs interprétions.

Chapitre 1:

Redéploiement d'un réseau de capteurs sans fil en utilisant les robots

1. Introduction

Les réseaux de capteurs sans fil (RCSF) jouent un rôle clé dans la détection, la collecte et la diffusion d'informations sur les phénomènes environnementaux (Tilak .S et al, 2002)^[2]. Il doit garantir de servir pendant une période proportionnellement grande. Malheureusement, les capteurs de ce réseau sont victimes de plusieurs facteurs qui provoquent leurs pannes après une certaine période de son fonctionnement. Les pannes influent sur la qualité de service fournie par le réseau, ils causent généralement un manque de couverture ou de connectivité ou une mauvaise qualité de transmission des données. Il existe plusieurs stratégies pour éviter ou prévenir les conséquences d'une panne, comme la gestion d'énergie des capteurs, le clustering et le déploiement dense. Mais, après une certaine période ces stratégies ne deviennent plus efficaces; d'où l'ajout des nouveaux nœuds devient nécessaire, ils s'agit du redéploiement. Le redéploiement représente la dernière étape de déploiement du réseau de capteurs sans fil où de nouveaux capteurs sont ajoutés afin de réparer les pannes qui se sont produites. Mais dans un environnement dangereux, l'utilisation d'un ou plusieurs robots mobiles pour le redéploiement est effective pour assurer la maintenance du réseau et d'allonger sa durée de vie.

Dans ce Chapitre, on aborde une discussion conceptuelle sur les concepts fondamentaux des réseaux de capteurs sans fil. Entre autres, on présente la notion de pannes, les stratégies existantes concernant la tolérance aux pannes, on introduit les notions de redéploiement et de redéploiement utilisant les robots. Enfin, nous présentons quelques modèles de robots utilisés dans le contexte de recherche et de réparations de pannes de capteurs.

2. Réseau de capteurs sans fil

2.1 Définitions

- Un réseau de capteurs est un outil de mesure et de transmission de l'information sur le phénomène à observer (Tilak .S et al, 2002).
- Les réseaux de capteurs sans fil sont considérés comme un type spécial des réseaux Ad hoc où l'infrastructure fixe de communication et l'administration centralisée sont absentes et les nœuds jouent, à la fois, le rôle des hôtes et des routeurs (Khelladi.L et Badache.N, 2010).

- Il est composé d'un ensemble des capteurs. Chacun de ces nœuds (capteur) a la capacité de collecter des données et de les transférer au nœud passerelle (dit "sink" en anglais ou puits) par l'intermédiaire d'une architecture multi-sauts. Le puits transmet ensuite, ces données par Internet ou par satellite à l'ordinateur central «Gestionnaire de tâches» pour analyser les données et prendre des décisions (Challal.Y, 2008). La Figure 1.1 montre l'architecture générale d'un réseau de capteurs sans fil.

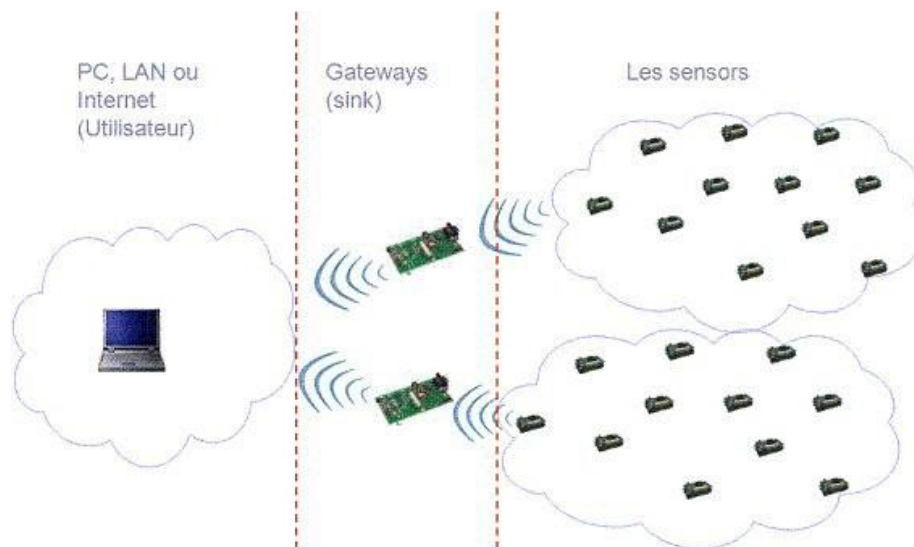


Figure 1.1: Architecture d'un réseau de capteurs (Challal.Y, 2008).

Le nœud puits est responsable, en plus de la collecte des rapports, de la diffusion des demandes sur les types de données requises aux capteurs via des messages de requêtes. Un réseau de capteurs peut contenir plusieurs nœuds puits diffusant des intérêts différents.

2.2. Critères de classements des réseaux de capteurs

Un réseau de capteurs sans fil peut être classé selon différents critères. Ci-dessous, on présente une classification selon cinq critères: la mobilité des nœuds, la topologie, le modèle de communication, le paradigme de communication, le modèle de transmission des données.

1. Selon la mobilité des nœuds, on peut déterminer les types de réseaux qui sont classées en trois types : mobile, statique et hybride (quelques nœuds sont mobiles et les autres sont statiques).
2. La topologie détermine l'organisation des capteurs dans le réseau. Il existe deux principales topologies dans les protocoles de routage pour les RCSFs: La topologie plate (tous les nœuds possèdent le même rôle), les nœuds sont semblables en termes de ressources, et la topologie hiérarchique (les nœuds sont divisés en plusieurs niveaux de responsabilités).
3. Selon le but de communication, il existe deux types: type application et type infrastructure. Dans le type application, les capteurs communiquent entre eux pour transférer les données captées. La transmission de ces données se fait soit

d'une façon coopérative entre les nœuds ou non coopérative. Le type infrastructure se réfère à la communication nécessaire pour la configuration, la maintenance, et l'optimisation des opérations. La communication d'infrastructure est utilisée pour l'optimisation de réseau (Tilak .S et al, 2002).

4. Selon le paradigme de communication : Le paradigme de communication représente l'information nécessaire pour la reconnaissance d'une information. Selon Challal.Y, (2008)¹⁸, il existe trois modes: - *Node centric* où les communications se basent sur l'identification des nœuds participants qui se fait à l'aide d'adresses IPs. - *Data centric* où les communicants sont identifiés par leurs données et l'identification des nœuds est inutile, *Position centric* où les positions des nœuds représentent le moyen principal d'adressage et de routage.
5. Selon le modèle livraison des données ainsi que les propos de Mouad. S, (2008); on distingue quatre modèles: Le modèle orienté temps (périodique) où l'acquisition et la transmission des données capturées sont liées au temps (instant précis ou période d'acquisition). Le modèle orienté événement (*event driven* ou *data driven*) où les capteurs envoient leurs données seulement si un événement spécifique se produit. Le modèle orienté requête dans lequel un capteur envoie de l'information uniquement suite à une demande explicite de la station de base. Le modèle hybride: Il met en œuvre une combinaison des trois modes de fonctionnement.

2.3 Caractéristiques des RCSFs

Un réseau de capteurs est une variante de réseau mobile ad hoc, il est composé d'un grand nombre de nœuds de capteurs, qui sont déployés d'une manière dense. Le nombre des nœuds de capteurs dans un réseau est plus élevé que les nœuds d'un réseau ad hoc (Akyildiz, W et al, 2001). Les capteurs sont répartis tout au long d'un espace physique (la zone de déploiement), ils se connectent entre eux avec une connexion sans fil (Wendi B et al, 2004). Ci-dessous, on présente quelques caractéristiques principales d'un réseau de capteurs sans fil:

- **Autonomie de fonctionnement:** Les nœuds capteurs sont équipés d'un processeur embarqué. Au lieu d'envoyer les données brutes vers les nœuds responsables de la fusion, les nœuds de capteurs utilisent leurs capacités de traitement localement et effectuent des calculs simples pour ne transmettre que les données requises et partiellement traitées (Akyildiz, W et al, 2001).
- **Changement dynamique de topologie:** Il est du soit à la mobilité des capteurs à travers l'espace, soit à l'addition de nouveaux nœuds, soit à la perte de nœuds déjà existants à cause d'une panne ou de l'épuisement d'énergie (Wendi B et al, 2004).
- **Assurer un minimum de qualité de service:** Ce niveau de QoS doit être maintenu sur une longue période de temps (Wendi B et al, 2004).
- **La limitation des ressources et des capacités physiques:** La bande passante du réseau et l'énergie de ses capteurs sont limités (Wendi B et al, 2004); la capacité de traitement est aussi limitée. Cela est, particulièrement, vrai car souvent les piles des capteurs et la taille de leurs mémoires sont limitées (Wendi B et al, 2004).
- **Les applications coopératives :** Les applications supportées par un réseau de capteurs implique une collaboration pour atteindre un objectif unique et partagé

les ressources disponibles (par exemple, l'énergie du capteur, la bande passante) (Wendi B et al, 2004).

- Les Capteurs sont sujets à des défaillances (Akyildiz, W et al, 2001).
- Les capteurs utilisent la diffusion comme un paradigme de communication principale.
- Les nœuds capteurs peuvent ne pas avoir un identificateur global (ID) en raison de la grande quantité de frais généraux et le nombre important de capteurs.

2.4 Domaines d'applications

Le nouveau concept de micro-capteurs connectés via un support sans fil permet la réalisation de plusieurs types d'applications selon le domaine dans lequel ils sont utilisés par Akyildiz, W, (2001) et Khelladi. L, (2010). Pour plus de détails, voir Figure 1.2.

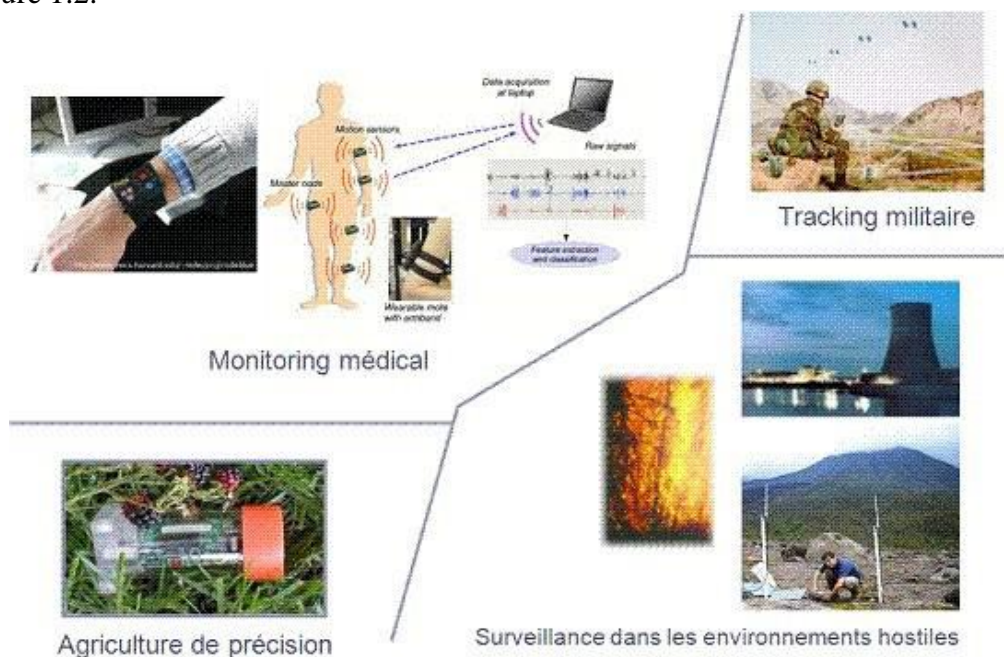


Figure 1.2: Les différentes applications des réseaux de capteurs (Challal.Y, 2008).

2.4.1 Applications militaires

Les réseaux de capteurs sans fil sont capables de constituer des modules intégraux dans les systèmes militaires de commandes, de contrôle, de communication, de calcul, d'intelligence, de surveillance, de reconnaissance et de ciblage. Ces systèmes sont communément nommés par les systèmes Akyildiz, W, (2001) et Khelladi. L, (2010) C4ISR (command, control, communications, computing, intelligence, surveillance, reconnaissance, and targeting).

Le déploiement rapide, l'auto organisation et la tolérance aux pannes de ce type de réseaux les rendent une solution prometteuse dans un tel domaine.

2.4.2 Applications dans l'environnement

Diverses sont les applications d'environnement qui peuvent bénéficier de la technologie des réseaux de capteurs sans fil. On peut citer par exemple, le cheminement des mouvements d'oiseaux, des petits animaux et des insectes. Le contrôle des aspects environnementaux qui peuvent affecter les récoltes et le bétail, l'irrigation, les macro-instruments utilisés pour la surveillance des terrains à grande échelle et les explorations planétaires, la détection chimique et biologique, la détection des incendies dans les forêts, les travaux de recherche météorologiques et géophysiques, la détection des inondations, les études de pollution et la schématisation de la bio-complexité de l'environnement d'après Akyildiz, W, (2001) et Khelladi. L, (2010).

2.4.3 Applications dans le domaine médical

Les réseaux de capteurs peuvent être très utiles dans le domaine médical. Ils peuvent, par exemple, fournir des interfaces d'aides aux handicapés, la surveillance intégrée des patients, le diagnostic, l'administration des médicaments au sein de l'hôpital, le contrôle des mouvements et processus internes des insectes ou des autres petits animaux, le contrôle des médecins et des patients dans l'hôpital ainsi que la télésurveillance et la collecte des informations physiologiques humaines (Khelladi. L et Badache. N, 2010).

2.4.4 Applications dans les accueils et les maisons

Les systèmes de sécurité sont de plus en plus complexes. La gestion ne concerne pas seulement l'intrusion dans les espaces sécurisés mais il est évident de prendre en considération la présence de substances telles que les incendies ou le gaz monoxyde de carbone. Pour être en mesure de surveiller les variables de l'application, le système de sécurité doit obtenir des données provenant des capteurs hétérogènes comme les capteurs acoustiques, les capteurs de mouvement, les capteurs de chaleur et de vibrations dispersés dans toute la maison / bureau.

L'alimentation de ces capteurs par batteries rend leur placement facile dans les maisons existantes déjà sans procéder à des modifications majeures par les ménages. Pour que la durée de vie de réseau de capteurs dure longtemps, l'application doit activer un sous-ensemble de capteurs à tout moment. Une fois l'activation d'un capteur sera déclenchée par un événement, l'application doit analyser les données et décider la manière pour modifier la configuration des capteurs actifs.

Les capteurs qui ont les données sont vraiment importants dans l'application, tels que les capteurs de vidéo, ils ne doivent pas être utilisés en tant que routeurs ou des nœuds de commande, de sorte que leur énergie soit conservée (Wendi. B, 2004).

2.5 Critères de performances dans la conception d'un réseau de capteurs

2.5.1 Tolérance aux pannes

Les nœuds peuvent être sujets à des pannes dues à leur fabrication (ce sont des produits de série bon marché, il peut donc y avoir des capteurs défectueux) ou plus fréquemment à un manque d'énergie. Les interactions externes (chocs, interférences)

peuvent aussi être la cause des dysfonctionnements. Afin que les pannes n'affectent pas la tâche première du réseau, il faut évaluer la capacité du réseau à fonctionner sans interruption (Kacimi. R, 2009).

En effet, si le réseau de capteurs est destiné aux environnements avec un faible degré d'interférences, tel que ceux utilisés dans les bâtiments pour surveiller le taux d'humidité et le degré de température, les protocoles utilisés ne doivent pas cibler une grande tolérance aux pannes, car dans ce type de réseau, il n'existe pas une grande interférence avec l'environnement, et ses nœuds ne sont pas exposés au risque d'endommagement.

Par contre, si le réseau est destiné aux applications militaires telle que la surveillance et le contrôle d'un champs de bataille, le niveau de tolérance aux pannes visé par les protocoles employés doit être très élevé, car les nœuds sont exposés à un grand risque d'endommagement par des actions hostiles, et les informations captées sont très critiques. Par conséquent, le niveau de tolérance aux pannes requis dépend de l'application du réseau de capteurs conçu, et les schémas de conception doivent prendre en charge ce paramètre (Khelladi. L et Badache.N, 2010).

Il est difficile de remplacer les capteurs existants, le réseau doit être tolérant aux pannes, tels que les échecs non catastrophiques qui seront cachés à partir de l'application. La tolérance aux pannes peut être réalisée par exemple, grâce à la réplication de données. Cependant, la réplication de données elle-même exige l'énergie, ce qui incite à réaliser un compromis entre la réplication des données et l'efficacité énergétique. Nous suggérons que la réplication de données soit spécifiée par l'application. Les données qui ont une priorité plus élevée selon l'application pourraient être reproduites pour la tolérance aux pannes et les autres données pourraient ne pas être (Tilak. S et *al*, 2002).

2.5.2 Topologie maintenable

Les caractéristiques de déploiement aléatoire, le fonctionnement autonome et la fréquence élevée de pannes rendent la maintenance de la topologie d'un réseau de capteurs une tâche complexe. En effet, plusieurs centaines de capteurs sont déployés avec une densité qui pouvant être supérieure à 20 nœuds par m³. Ceci exige une bonne gestion de la maintenance de la topologie du réseau déployé (Khelladi. L et Badache. N, 2010).

2.5.3 Scalabilité

La surveillance d'un phénomène nécessite le déploiement d'un nombre de nœuds qui est de l'ordre de plusieurs milliers de capteurs. Suivant l'application, ce nombre peut encore augmenter jusqu'à des millions de capteurs. Cette augmentation doit pouvoir garantir un bon fonctionnement avec ce nombre de capteurs élevés. On doit, aussi, exploiter la nature fortement dense des réseaux de capteurs.

L'évolutivité des réseaux de capteurs est également un facteur critique. Pour les grands réseaux, il est probable que la localisation des interactions dans la hiérarchie et l'agrégation soit critique pour assurer l'évolutivité (Tilak. S et *al*, 2002).

2.5.4 Gestion d'accès au Support de transmission

Dans un réseau de capteurs, les nœuds sont liés via un moyen de communication sans fil, et ceci, en utilisant un support optique ou des fréquences radio. Cependant, il faut s'assurer de la disponibilité du moyen de transmission choisi dans l'environnement de captage afin de permettre au réseau d'accomplir la totalité de ses tâches (Khelladi. L et Badache. N, 2010).

2.5.5 Les contraintes matérielles

Les nœuds capteurs sont des petites unités de traitement et de transmission, grâce à leur petit volume, les unités intégrées possèdent aussi un petit volume. Ceux la influent directement sur leurs capacités de traitement, i.e. ne permettent pas de faire des exécutions multitâches et d'enregistrer un volume important d'informations.

2.5.6 L'efficacité énergétique

Comme les nœuds capteurs sont alimentés par les piles, les protocoles doivent être économes en énergie afin de maximiser la durée de vie du système (Tilak. S et *al*, 2002).

2.5.7 Latence

L'observateur est intéressé de connaître les phénomènes dans un délai donné et bien déterminé. Les sémantiques de précision de latence d'pendent de l'application (Tilak. S et *al*, 2002).

2.5.8 Précision

L'obtention des informations précises est l'objectif principal de l'observateur. Comme nous savons, la précision est déterminée par l'application des données observées. Il y a un compromis entre la précision, la latence et l'efficacité énergétique. L'infrastructure donnée doit être adaptée de sorte que l'application obtienne la précision et le retard souhaités avec un minimum d'énergie dépensée. (Tilak. S et *al*, 2002).

2.5.9 Le coût de production

Le coût de production d'un micro-capteur est très important pour l'évaluation du coût global du réseau. Sachant qu'un nœud contient d'autres systèmes que celui de transmission radio tel que les unités de captage et de traitement des données. De plus, le nœud peut être équipé d'éléments additionnels tel qu'un system de localisation GPS ou un système de rechargement d'énergie. Dès lors, la minimisation du coût de production du nœud capteur constitue un grand défi mené par les chercheurs, vu les fonctionnalités que doit comporter ce type de nœud et l'objectif désiré pour un coût global minimal.

2.6 Capteur

Un capteur est un dispositif ayant pour tâche la transformation d'une mesure physique observée en une mesure électrique. Cette mesure sera traduite en une donnée binaire exploitable et compréhensible pour un système d'information. Les capteurs actuels gèrent des fonctionnalités qui leurs permettent le traitement de ces données et leurs

communications vers un autre dispositif, en plus de l'enregistrement et la détection d'événements mesurables. On parle alors de capteur intelligent, capable à la fois de mesurer des données et de les communiquer avec d'autres capteurs au sein d'un réseau (Martins. D, 2010).

Selon Akyildiz, W et *al* (2001), et selon les types des capteurs, un capteur détecte: la température, l'humidité, le mouvement des véhicules, la pression, le taux de bruits, la présence ou absence de certains types d'objets, le taux de frottement sur des objets attachés. Il peut détecter aussi d'autres caractéristiques telles que la vitesse, la direction et le volume d'un objet donné.

2.6.1 Caractéristiques d'un capteur

Selon Akyildiz, W et *al* (2001), les capteurs:

- sont des unités de petite taille,
- consomment de l'énergie extrêmement faible,
- utilisent des densités volumétriques élevées,
- indispensable et doivent être de coût de production bas.
- doivent être autonomes et doivent fonctionner sans surveillance,
- doivent être adaptables à l'environnement.

Un capteur (voir Figure 1.3) est constitué des composants suivants :

- **L'unité de traitement:** C'est l'unité principale du capteur, généralement un processeur couplé à une mémoire vive. Son rôle consiste à contrôler le bon fonctionnement des autres unités. Sur certains capteurs, elle peut embarquer un système d'exploitation pour faire fonctionner le capteur. Elle peut aussi être couplée à une unité de stockage, qui servira à y enregistrer les informations transmises par l'unité de capture (Martins. D, 2010).

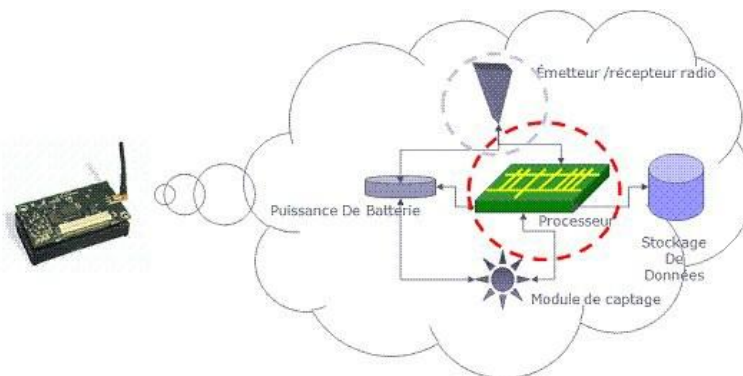


Figure 1.3: Anatomie d'un réseau de capteurs (Challal.Y, 2008).

- **Les microcontrôleurs :** Utilisés dans le cadre de réseaux de capteurs, ces dispositifs sont à faible consommation d'énergie. Leurs fréquences sont assez faibles, moins de 10 MHz pour une consommation de l'ordre de 1 mW. Une autre caractéristique est la taille de leur mémoire qui est de l'ordre de 10 Ko de RAM

pour les données et de 10 Ko de ROM pour les programmes (Naskath. J et *al.* 2011).

- **L'unité de capture:** Elle permet la capture des données, c'est à dire la mesure des grandeurs physiques ou analogiques et leur conversion en données numériques. Elle est composée du capteur lui-même et de l'ADC 1. Le capteur est chargé de récupérer les signaux analogiques qu'il transmet à l'ADC qui a pour rôle de transformer et de communiquer les données analogiques en données numériques compréhensibles pour l'unité de traitement (Martins. D, 2010).
- **L'unité de communication:** Elle a pour fonction de transmettre et recevoir l'information. Elle est équipée d'un couple émetteur/récepteur, aussi appelé transceiver (terme anglophone issu de la contraction des mots transmitter et receiver). Elle permet la communication au sein du réseau (radiofréquence (ondes radios, optique, infrarouge, etc...)) (Martins. D, 2010).
- **L'unité d'alimentation:** C'est un élément primordial de l'architecture du capteur, elle fournit en énergie toutes les autres unités. Elle correspond le plus souvent à une batterie ou une pile alimentant le capteur, dont les ressources limitées en font une problématique propre à ce type de réseau. La réalisation récente d'unité d'alimentation à base de panneaux solaires tente d'apporter une solution pour prolonger sa durée de vie (Martins. D, 2010).
- Autre unité : unité de localisation, tel qu'un GPS, une unité de mobilité (Martins. D, 2010).
- Les capacités physiques diffèrent selon la mission de captage, l'environnement de déploiement et le traitement nécessaire effectué par les capteurs. Ci-dessous, on présente quelques exemples de capteurs : WINS (Wireless Integrated Network Sensors) selon Corke. P et *al.*, (2006), les capteurs de ce genre possèdent une portée de communication faible avec un faible débit (inférieure à 1kbps). Cette architecture exploite la distance faible entre les nœuds pour garantir une communication multi-sauts qui permet une réduction considérable dans l'énergie consommée (Corke. P et *al.*, 2006). Les Smart Dust, grâce à leur poids léger et le volume miniaturisé, ces micro-nœuds peuvent être attachés à des objets ou même se laisser flotter dans l'air. Ces nœuds utilisent des technologies MEMS (microelectro mechanical systems) pour implanter les fonctions de captage et de communication à travers un support optique. Ce type de communication nécessite, par contre, la disponibilité permanente d'une ligne de vision entre les nœuds communicants. Les capteurs Smart Dust peuvent porter plusieurs types de sources énergétiques telles que les cellules solaires ou les batteries génériques disponibles sur le marché (Corke. P et *al.*, 2006).

2.6.2 La pile protocolaire implémentée au niveau d'un capteur

La pile protocolaire utilisée par le nœud puits ainsi que tous les autres capteurs du réseau est donnée dans la Figure 1.4.

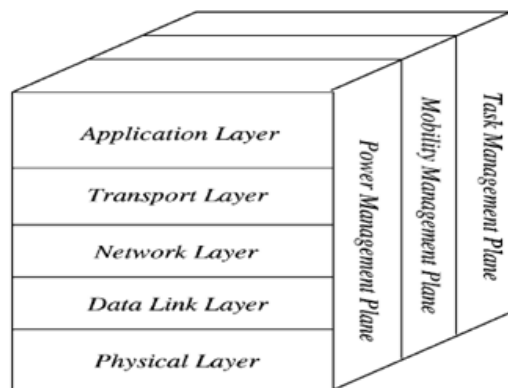


Figure 1.4: La pile protocolaire d'un réseau de capteur d'après Akyildiz,I-F (2001).

La pile protocolaire des réseaux de capteurs WINS est composée de 5 couches, la couche application, la couche réseau, la couche MAC et la couche physique (Corke.P et al.2006).

- **Gestion d'énergie**

Le but de cette couche est de minimiser la consommation d'énergie, par exemple éteindre l'interface de réception de capteur dès qu'il reçoit un message d'un nœud voisin afin d'éviter la réception des messages dupliqués, d'après Akyildiz, I-F et al (2001), Khelladi. L et Badache.N (2010).

- **Gestion de mobilité**

Ce niveau détecte et enregistre tous les mouvements des nœuds capteurs d'une manière à leur permettre de garder continuellement une route vers l'utilisateur final et maintenir une image récente sur les nœuds voisins. Cette image est nécessaire pour pouvoir équilibrer l'exécution des tâches et la consommation d'énergie (Corke.P et al, 2006).

- **Gestion des tâches**

Le niveau de gestion des tâches assure l'équilibrage et la distribution des tâches sur les différents nœuds du réseau, afin d'assurer un travail coopératif et efficace en matière de consommation d'énergie, et par conséquent, prolonger la durée de vie du réseau (Corke.P et al, 2006).

Les différentes tâches exécutées par un nœud sont le captage, la transmission/réception et le traitement.

- **La couche application**

La conception des protocoles agissants dans la couche application reste largement inexploitée. Nous donnons à titre d'exemples trois protocoles d'application pour les réseaux de capteurs : *Sensor management protocol* (SMP), *Task assignment and data advertisement protocol* (TADAP) et enfin *Sensor query and data dissemination protocol* (SQDDP). Ces protocoles sont nécessaires pour tous les réseaux de capteurs

basés sur le schéma des couches protocolaires décrites précédemment, d'après Akyildiz, W (2001) et Khelladi. L, (2010).

- **La couche transport**

Cette couche est, particulièrement, nécessaire lorsque le système est prévu pour être accessible par le biais d'Internet ou d'autres réseaux externes. Cependant, au meilleur de notre connaissance, il n'y a pas de tentative de proposition de schéma ou de discussion sur les questions liées à la couche de transport d'un réseau de capteurs dans la littérature (Akyildiz, W et al, 2001).

On peut faire la communication entre l'utilisateur et le nœud puits en UDP ou en TCP via l'Internet ou par satellite. D'autre part, la communication entre le(s) nœud(s) puits et les capteurs peut être, purement, par des protocoles de type UDP parce que chaque nœud capteur a une mémoire limitée (Akyildiz, I-F et al, 2001).

- **La couche réseau**

Le rôle de cette couche est de trouver un chemin entre le sink et le(s) capteur(s). La couche réseau est généralement conçue en fonction des principes suivants d'après Khelladi. L, (2010):

- L'efficacité en consommation d'énergie est une considération prioritaire.
- Tous les protocoles dans les réseaux de capteurs prennent en charge les données d'application qui circulent dans le réseau (Data centric).
- L'agrégation des données est utile quand elle ne cache pas l'effort collaboratif des nœuds capteurs.
- Un réseau de capteur idéal garanti un adressage basé-attributs et ses nœuds sont conscients de leur localisation.

- **La couche liaison d'après Khelladi. L et Badache. N, (2010)**

Un protocole MAC dans un réseau de capteurs sans fil auto-organisé doit accomplir deux objectifs essentiels :

- a) la création de l'infrastructure de communication :** Comme ce type de réseau est construit de milliers de nœuds capteurs éparpillés d'une manière dense sur le champ de captage, le protocole MAC doit assurer l'établissement des liens de communication pour le transfert des données entre ces nœuds. Ceci formera l'infrastructure de base requise pour la communication saut-par-saut et donnera au réseau une capacité d'auto-organisation.
- b) partage des canaux de communication :** Le second objectif du protocole MAC est le partage efficace et équitable des canaux de communication entre les nœuds capteurs. Les schémas traditionnels des protocoles MAC, peuvent être classés suivant les mécanismes adoptés pour le partage des ressources de communication.

- **La couche physique d'après Khelladi. L et Badache. N, (2010)**

La couche physique est responsable de la sélection des fréquences, la génération des ondes porteuses, la détection du signal, la modulation et le cryptage des données. La

génération des ondes porteuses et la détection des signaux sont beaucoup plus liés au matériel utilisé et à l'architecture des interfaces de transmission.

Il est très bien connu que les communications sans fil à longue distance peuvent être coûteuses, en termes de consommation d'énergie et de complexité de réalisation. En effet, la consommation d'énergie est d'une importance majeure lors de l'implantation de la couche physique pour les réseaux de capteurs.

2.6.3 Les sources de perte d'énergie d'après Yaser. Y, (2010)

- **Consommation d'énergie dans les RCSFs**

La première étape dans la conception de système énergétique de capteurs consiste à analyser les caractéristiques de consommation d'énergie d'un nœud de capteur sans fil. Cette analyse systématique de l'énergie d'un nœud capteur est extrêmement importante pour identifier les problèmes dans le système énergétique pour permettre une optimisation efficace.

L'énergie consommée par un capteur est principalement due aux opérations suivantes: la détection, le traitement et la communication.

- **Energie de capture**

Les sources de consommation d'énergie des nœuds pour les opérations de détection ou de capture sont : l'échantillonnage, la conversion analogique-numérique, le traitement de signal et l'activation de la sonde de capture.

- **Energie de traitement**

L'énergie de traitement est composée de deux sortes d'énergie: l'énergie de commutation et l'énergie de fuite. L'énergie de commutation est déterminée par la tension d'alimentation et la capacité totale commutée au niveau logiciel (en exécutant un logiciel). Par contre, l'énergie de fuite correspond à l'énergie consommée lorsque l'unité de calcul n'effectue aucun traitement. En général, l'énergie de traitement est faible par rapport à celle nécessaire pour la communication.

- **Energie de communication**

L'énergie de communication se décline en trois parties: l'énergie de réception, l'énergie de l'émission et l'énergie en état de veille. Cette énergie est déterminée par la quantité des données à communiquer et la distance de transmission, ainsi que par les propriétés physiques du module radio. L'émission d'un signal est caractérisée par sa puissance; quand la puissance d'émission est élevée, le signal aura une grande portée et l'énergie consommée sera plus élevée. Notons que l'énergie de communication représente la portion la plus grande de l'énergie consommée par un nœud capteur.

2.7 Durée de vie d'un réseau de capteurs

2.7.1 Définition générale

La vie d'un réseau de capteurs correspond à la période de temps durant laquelle le réseau peut, selon le cas, maintenir assez de connectivité, couvrir le domaine entier ou garder le taux de perte d'informations en-dessous d'un certain niveau (Kacimi. R, 2009).

2.7.2 Quelques définitions existantes

Selon Kacimi. R, (2009), il existe quatorze (14) définitions pour la durée de vie d'un réseau de capteurs :

1. La durée jusqu'à ce que le premier nœud épuise toute son énergie,
2. La durée jusqu'à ce que le premier *clusterhead* épuise toute son énergie,
3. La durée jusqu'à ce qu'il reste au plus une certaine fraction β de nœuds survivants dans le réseau,
4. *Demi-vie du réseau* : La durée jusqu'à ce que 50% des nœuds épuisent leurs batteries et s'arrêtent de fonctionner,
5. La durée jusqu'à ce que tous les capteurs épuisent leurs énergies;
6. La durée jusqu'à ce que le réseau soit partitionné : apparition de la première division du réseau en deux (ou plus). Cela peut correspondre aussi à la mort du premier nœud (si celui-ci tient une position centrale) ou plus tard si la topologie du réseau n'est plus robuste;
7. *k-couverture*: la durée pendant laquelle la zone d'intérêt est couverte par au moins k nœuds;
8. *100%-couverture*
 - La durée pendant laquelle chaque cible est couverte par au moins un nœud;
 - La durée pendant laquelle l'ensemble de la zone est couverte par au moins un nœud;
9. α *couverture*
 - La durée cumulée, au bout de laquelle au moins une portion α de la région est couverte par au moins un nœud ;
 - La durée pendant laquelle la couverture tombe en-dessous d'un seuil prédéfini α ;
 - La durée de fonctionnement continu du système avant que la couverture ou la proportion de paquets reçus (PDR pour Packet Delivery Ratio) tombent en-dessous d'un seuil prédéfini;
10. La durée pendant laquelle un pourcentage donné de nœuds possèdent un chemin vers la Station de Base
11. L'espérance de l'intervalle complet pendant lequel la probabilité de garantir simultanément une connectivité et une *k-couverture* est au moins α ;
12. La durée jusqu'à la perte de la connectivité ou de la couverture;
13. La durée jusqu'à ce que le réseau ne fournisse plus un taux acceptable de détection d'événements;
14. La durée pendant laquelle le réseau satisfait continuellement les besoins de l'application.

2.8 Importance de la couverture et la connectivité dans un RCSF

D'après les définitions précédentes, la durée de vie du réseau de capteurs termine généralement quand il ya un manque de couverture et de connectivité dans le réseau, ce qui exprime la nécessité de ces deux paramètres. Ci-dessous, on présente la définition de ces deux paramètres et la relation entre eux.

2.8.1 Connectivité

Selon Gallais. A, (2008), Deux nœuds sont dits connectés si la distance qui les sépare est inférieure au rayon de communication ou s'il existe un chemin multi-sauts composé de nœuds intermédiaires capables de communiquer deux à deux. Lorsque plusieurs chemins, k par exemple, existent entre deux nœuds, ces derniers sont dits k -connectés. Un réseau est k -connexe si chaque paire de nœuds est k -connectées.

2.8.2 Couverture

A) La couverture simple (Gallais. A, 2008) [9]

La forme de la surface couverte par un nœud dépend du matériel utilisé et de l'information captée. En général, cette zone est un disque dont le centre est la position du capteur et son rayon est égale au rayon de captage.

2.8.3 Relation entre la couverture et la connectivité

Selon Carle. J, (2008), si $CR \geq 2SR$. Autrement dit, si la surface est totalement couverte alors tous les nœuds sont connectés, autrement si $CR \geq 2SR \rightarrow$ connectivité, où CR est le rayon de communication d'un capteur et SR est son rayon de couverture.

3. Panne d'un capteur

3.1 Les pannes

Une faille (ou panne) du système se produit lorsque son comportement devient inconsistant et ne fournit pas le résultat voulu. La panne est une conséquence d'une ou plusieurs erreurs. Une erreur représente un état invalide du système dû à une faute (défaut). La faute est donc la première cause de l'erreur, cette dernière provoque la faille du système (Challal. Y, 2008).

Lorsqu'il y a une panne, il y aurait un risque que la zone ne soit pas couverte et connectée. Cela génère des conséquences catastrophiques sur l'application (manque d'information, paralysie du réseau).

La propriété de tolérance aux pannes est définie par: l'habilité du réseau à maintenir ses fonctionnalités sans interruptions provoquées par la panne des capteurs. Elle vise, donc, à minimiser l'influence de ces pannes sur la tâche globale du réseau (Khelladi. L et Badache. N, 2010).

Cette propriété $R(t)$ est modélisée par une distribution de poisson où $R(t)$ donne la probabilité de ne pas avoir une panne pour un nœud capteur pendant l'intervalle de temps $[0, t]$ (Khelladi. L et Badache. N, 2010)

$$R(t) = \exp(-\lambda_k t)$$

λ_k est le taux de pannes du nœud capteur k ,

t est la période de temps.

Alors, la probabilité d'avoir une panne est $R'(t)=1-R(t)$

Si le réseau de capteurs est destiné aux environnements avec un faible degré d'interférences, tel que ceux utilisés dans les bâtiments pour surveiller le taux d'humidité et le degré de température, les protocoles utilisés ne doivent pas cibler une grande tolérance aux pannes, car dans ce type de réseau, il n'existe pas une grande interférence avec l'environnement, et ses nœuds ne sont pas exposés au risque d'endommagement.

Par contre, si le réseau est destiné aux applications militaires telle que la surveillance et le contrôle d'un champs de bataille, le niveau de tolérance aux pannes visé par les protocoles employés doit être très élevé, car les nœuds sont exposés à un grand risque d'endommagement par des actions hostiles, et les informations captées sont très critiques.

Par conséquent, le niveau de tolérance aux pannes requis dépend de l'application du réseau de capteurs conçu, et les schémas de conception doivent prendre en charge ce paramètre (Khelladi. L et Badache. N, 2010).

3.2 Causes d'une panne

- **Panne de design** : Elle est due à une mauvaise structuration du réseau ou du composant en particulier. En pratique, ce genre de panne ne devrait pas exister grâce aux tests et simulations avant la réalisation finale du réseau;
- **Panne opérationnelle** : Elle se produit durant le fonctionnement du système. Elle est généralement due aux causes physiques. En outre, on peut distinguer, spécialement pour les réseaux de capteurs, quatre causes principales:
 - **Energie**: L'épuisement de la batterie cause l'arrêt du capteur. La consommation d'énergie est très importante pour déterminer la durée de vie d'un nœud capteur, et donc de tout le réseau;
 - **Sécurité**: La destruction physique accidentelle ou intentionnelle par un ennemi peut être une cause de panne. L'absence de sécurité dans les réseaux de capteurs augmente le risque des pannes de ce type;
 - **Transmission**: La nature vulnérable de transmission radio, la présence d'obstacles dans les environnements hostiles ainsi que les interférences électriques peuvent être la source d'une faute lors du transfert de données;
 - **Endommagement physique** (Khelladi. L et Badache. N, 2010).

3.3 Durée d'une panne

En basant sur la durée, la panne peut être classifiée en :

- **Transitoire**: Conséquence d'un impact environnemental temporaire, elle peut éventuellement disparaître sans aucune intervention. La radiation cosmique est un exemple de panne transitoire.

- **Intermittente:** Variante de la panne transitoire, elle se produit occasionnellement et de façon imprévisible. Elle est généralement due à l'instabilité de certaines caractéristiques matérielles ou à l'exécution du programme dans un espace particulier de l'environnement.
- **Permanente:** Continue et stable dans le temps, la panne permanente persiste tant qu'il n'y a pas d'intervention externe pour l'éliminer. Un changement physique dans un composant provoque une panne matérielle permanente.

3.4 Comportant résultant d'une panne

3.4.1 Pannes selon le comportement résultant (Challal. Y, 2008)

Après l'occurrence d'une panne, on distingue quatre différents comportements possibles du composant concerné:

- **Panne accidentelle (Crash):** Le composant soit, s'arrête complètement de fonctionner ou bien continue mais sans retourner à un état stable (valide) ;
- **Panne d'omission:** Le composant n'est plus capable d'améliorer son service (échec total);
- **Panne de synchronisation (Timing):** Le composant effectue son traitement mais fournit le résultat en retard;
- **Panne Byzantine :** Cette panne est de nature arbitraire; le comportement du composant est donc imprévisible. Du à des attaques très malicieuses, ce type de pannes est considéré le plus difficile à gérer.
 1. Les capteurs d'une région sont tombés en panne à cause d'un fort incident comme une explosion; les feux dans la forêt.
 2. La panne d'un seul nœud par exemple à cause de l'énergie.

3.4.2 Panne isolée

Dans le contexte de WSN (Wireless Sensor Network), une panne d'un capteur affecte la connectivité et la couverture du réseau. La défaillance peut être classée en deux catégories: la défaillance d'un nœud unique, et la défaillance multiples simultanée.

- **La défaillance multiple simultanée:** Les RCSFs opérant dans un environnement difficile, les capteurs peuvent être soumis à des dégâts qui peuvent être si importante dans une partie de la zone couverture que le réseau se partitionne en segments disjoints. Par exemple, dans un champ de combat, des parties de la zone de déploiement peuvent être bombardées, dont tous les capteurs déployés dans cette zone sont détruits. Les nœuds survivant sont partitionnées en segments disjoints en raison de la perte de connectivité. La défaillance simultanée de plusieurs nœuds est très difficile, non seulement au processus de la reprise, mais aussi dans la détermination de la portée de l'échec (Younis. B et Lee. S, 2014). La Figure 1.5 montre un exemple d'une panne massive.
- Dans cette Figure, les cercles pleins indiquent les capteurs échoués, les cercles blancs indiquent les capteurs qui sont opérationnels.

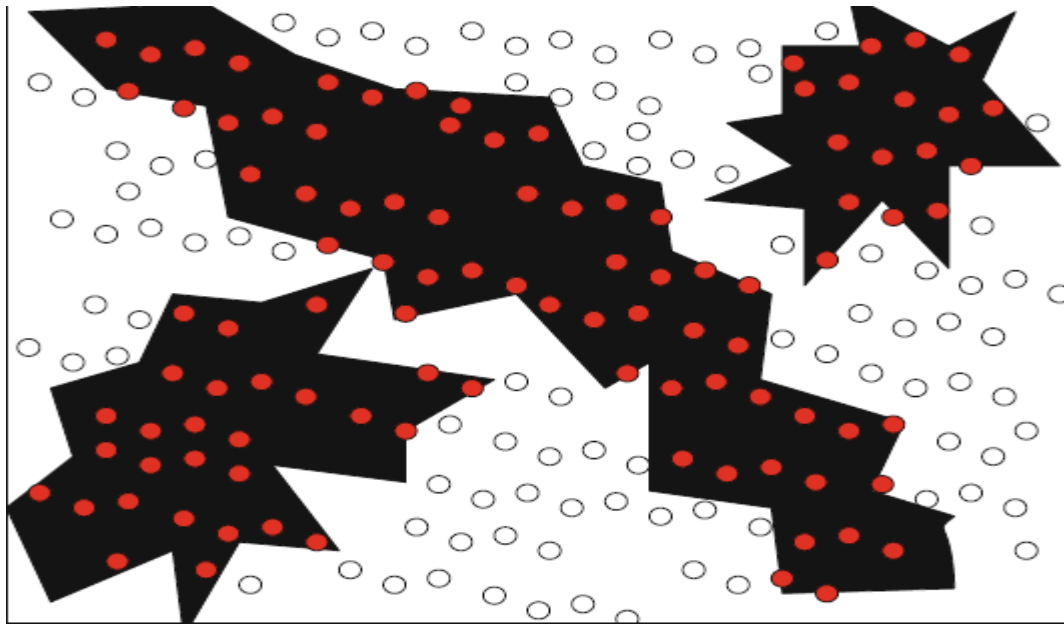


Figure 1.5: Représentation d'une panne massive (Younis. B et Lee. S, 2014).

- **La panne isolée (individuelle):** Selon Younis. B et Lee. S, (2014), la défaillance d'un seul nœud indique qu'un nœud échoue à la fois. Le type de défaillance peut être tout simplement détecté en utilisant des messages de battement de cœur local. Sauf s'il y a un chevauchement de la couverture, le nœud défaillant laissera une partie de la zone non surveillée comme le montre la Figure 1.6. D'autre part, la position du nœud dans la topologie de réseau détermine sa criticité de connectivité.

Dans la Figure 1.6, la panne de nœud S1 engendre un manque de couverture. Dans notre travail, on s'intéresse au traitement des pannes isolées.

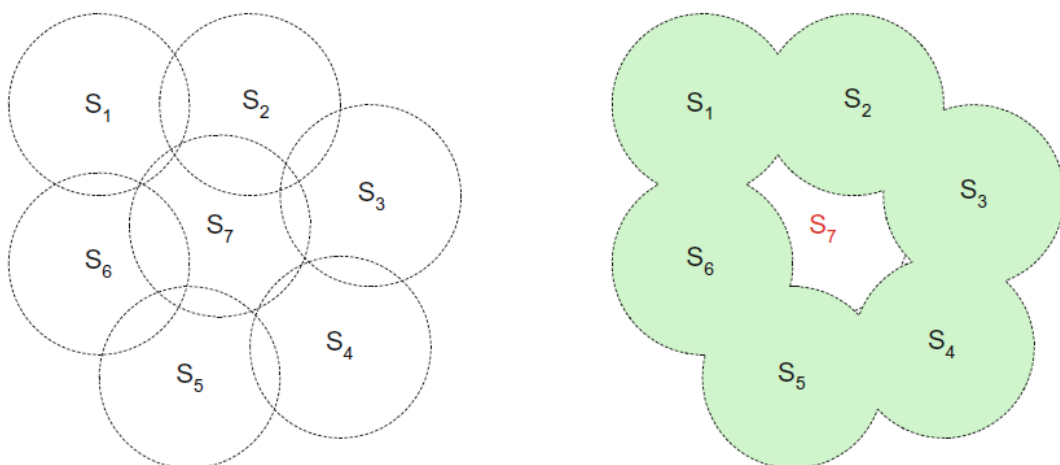


Figure 1.6: Exemple d'une panne isolée.

4. Les différentes phases de déploiement d'un RCSF

Les positions des nœuds, leurs distributions et leurs états de fonctionnement, dans la zone de surveillance, influent sur la couverture et la connectivité. Donc, elles influent, entre autre, sur la durée de vie de réseau.

4.1 Phase de déploiement

Les capteurs sont au préalable déployés sur une zone à surveiller. Pour satisfaire de nouvelles contraintes ou pour pallier des pannes, un déploiement de nœuds supplémentaires, dit itératif, peut être requis. Différents modes de déploiement sont envisageables et dépendent essentiellement de l'application de surveillance. Selon Gallais. A, (2008), le déploiement des capteurs peut être réalisé selon deux types :

- **Déploiement déterministe:** Lorsque l'environnement est accessible et connu, il est possible de placer avec précision les nœuds sur la zone. Par exemple : La mise en place les capteurs chargés de réguler la climatisation d'un immeuble ou de surveiller les constantes médicales de malades à distance.
- **Déploiement aléatoire:** L'utilisation des capteurs dans des zones inaccessibles ou sensibles rend impossible un déploiement déterministe, au cours duquel chaque objet serait placé à une position prédéterminée. Les nœuds peuvent alors être déployés à l'aide de moyens divers. Il a souvent été question d'un déploiement aléatoire des capteurs effectué comme un jeté de graines. Il semble pourtant difficile, vu la fragilité des capteurs existants, d'envisager un déploiement par avion par exemple. Néanmoins, un déploiement aléatoire peut être obtenu à partir d'une distribution de capteurs à des individus ou au placement de ces capteurs sur des objets mobiles.

Les caractéristiques de déploiement aléatoire dans le cas de fonctionnement autonome et la fréquence élevé de pannes, rendent la maintenance de la topologie d'un réseau de capteurs une tâche complexe (Khelladi. L et Badache. N, 2010). Dans ce qui suit, on présente les différentes étapes de déploiement due au changement de la topologie de réseau, à cause des pannes ou de la mobilité et quelques solutions qui peuvent réparer la topologie afin d'éviter ou de traiter les problèmes de manque de couverture ou de la connectivité.

4.2 Phase de pré-déploiement et de déploiement

Les nœuds de capteurs peuvent être éparpillés sur le champ de captage en masse ou placés d'une manière individuelle et ceci par le biais de plusieurs moyens tels que :

- Les jeter d'un avion
- Utiliser une artillerie, roquette ou missile, ou
- Les placer nœud par nœud d'une façon manuelle ou en utilisant des robots.

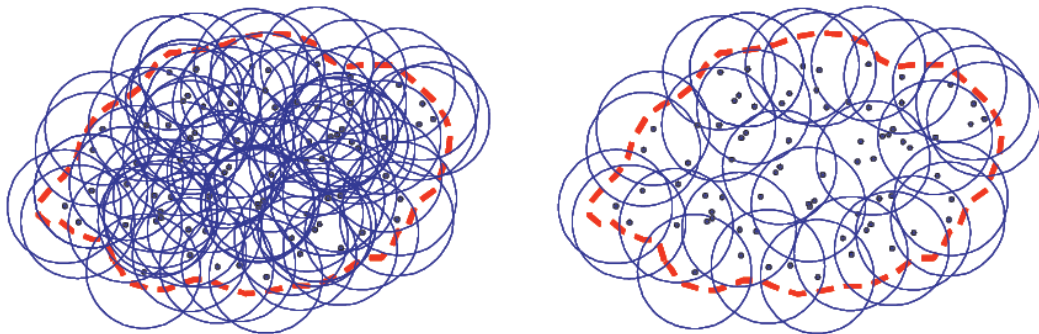
Le nombre important de nœuds utilisés dans un réseau de capteurs empêche leur déploiement suivant un plan soigneusement établi. Cependant, un schéma général pour le déploiement initial doit être conçu pour permettre (pré-déploiement) :

- De réduire les coûts d'installation
- Augmenter la flexibilité d'arrangement des nœuds

- Faciliter l'auto-organisation des nœuds et leur tolérance aux pannes (Khelladi. L et Badache. N, 2010).

Parmi les mécanismes de minimisation des effets des pannes dans cette phase, on cite :

- **Le déploiement dense:** La structure de déploiement des capteurs (lors de phase de déploiement) peut assurer la couverture et la connectivité malgré les pannes tant que le taux des pannes ne dépasse pas un certain seuil ou les capteurs se situent dans une zone K-couverte et K-connecté. Selon Khelladi. L et Badache. N, (2010), la surveillance d'un phénomène peut nécessiter le déploiement d'un nombre de nœuds qui sont de l'ordre de plusieurs milliers de capteurs. Suivant l'application, ce nombre peut encore augmenter jusqu'à des millions de capteurs, les nouveaux schémas doivent être capables de garantir un bon fonctionnement avec ce nombre de capteurs élevés, aussi bien d'exploiter la nature fortement dense des réseaux de capteurs. Cette densité peut varier entre quelques capteurs jusqu'à plusieurs centaines de capteurs dans une région de taille inférieure à 10 mètres de diamètre, elle dépend également de l'application. En général, la densité moyenne des nœuds dans un réseau de capteurs est estimée à 20 capteurs par m^3 .
- **L'ordonnancement des activités (Scheduling):** Les réseaux de capteurs sont généralement denses et redondants. En effet, suivant l'application, on déploiera plus ou moins de capteurs dans un souci d'allongement de la durée de vie de l'application. À tout moment, il existe donc des capteurs (actifs) qui observent la zone de déploiement et les autres capteurs sont en veilles (passifs). Cette redondance est exploitée par l'ordonnancement d'activité : Ordonnancer (voir Figure 1.7) l'activité dans un réseau de capteurs consiste à alterner les charges de façon à épuiser les nœuds équitablement. Pendant qu'une partie participe à l'application, les autres sont dans un mode passif, économisant ainsi leur énergie (Gallais. A, 2008).



(a) Un réseau de capteurs déployé. (b) Seule une partie des nœuds est nécessaire à la couverture de la zone.

Figure 1.7: Ordonnancement d'activités avec le critère de la couverture de surface.

La Figure 1.7(a) nous montre un réseau de capteurs déployé sur une zone à observer. Cette observation ne requiert pas une activité permanente de la part de tous

les nœuds comme illustré sur la Figure 1.7 (b) où seul un sous-ensemble des capteurs est suffisant pour une observation exhaustive de la zone (Gallais. A, 2008).

Le déploiement dense et l'ordonnancement des activités peuvent allonger la durée de vie du réseau malgré les pannes. Mais après une certaine période, le nombre des nœuds devient insuffisant pour garantir la couverture.

4.3 Phase de post-déploiement

Après la phase de déploiement, la topologie du réseau peut subir des changements dues aux :

- changement de position des nœuds
- accessibilité à cause du brouillage ou des obstacles en mouvements
- épuisement d'énergie
- mal fonctionnement des nœuds ou
- des besoins pour leur application.

En effet, bien que les nœuds d'un réseau de capteurs puissent être déployés d'une manière statique, la panne matérielle constitue un événement très commun à cause de l'épuisement d'énergie ou la destruction. Il est possible également d'avoir un réseau de capteurs avec des nœuds mobiles qui ont une mobilité très élevée. Par conséquent, la topologie du réseau de capteurs est exposée fréquemment aux changements après la phase de déploiement (Khelladi. L et Badache. N, 2010).

Parmi les méthodes de réparation de la couverture dans cette phase, on distingue :

- **La maintenance de la couverture en utilisant des nœuds mobiles:** L'article [15] a abordé le problème du remplacement des capteurs dans les réseaux sans fil pour les capteurs mobiles. Les capteurs mobiles sont utilisés pour maintenir la couverture totale et dans ce sens le concept du cluster dynamique est apparu. La solution de la réparation de la faute proposée n'assume pas l'information de la localisation qui est disponible. Les nœuds capteurs mobiles utilisent une opération géométrique simple et remplacent des nœuds mourants pour guérir ou augmenter la couverture et la connectivité existantes.

Le changement de topologie en utilisant des nœuds mobiles ou un robot permettra de garantir la couverture malgré la panne de quelques nœuds. Mais, si le nombre des nœuds descend en dessous d'un certain nombre, les capteurs existant ne permettent pas la couverture de toute la zone quelque que soit la topologie utilisée. En plus, l'équipement des capteurs par des unités de mobilisation augmente le coup de déploiement de réseau. Alors, il faut trouver d'autres mécanismes de remplacement des capteurs en panne et de réparation de réseau afin de garantir la couverture et la connectivité.

4.4 Phase de redéploiement des nouveaux nœuds

Les nœuds de capteurs additionnels peuvent être installés pour remplacer ceux qui sont en pannes ou bien pour répondre aux besoins des tâches assignées au réseau. Cette addition entraîne la réorganisation du réseau et le changement de sa topologie, (Khelladi. L et Badache. N, 2010).

Pour remplacer les nœuds en panne, il faut intervenir au niveau de la zone de surveillance pour remplacer ces derniers par des nouveaux capteurs fonctionnels. Parmi les méthodes d'intervention on cite:

- L'intervention des êtres humains : Dans cette solution, les humains se déplacent dans la zone de captage et remplacent les nœuds qui sont en pannes. Mais, cette solution possède beaucoup d'inconvénients. L'environnement est généralement dangereux pour la présence des êtres humains. Par exemple, si le champ de captage existe dans une autre planète où un être humain ne peut atteindre ou supporter ses conditions physiques très dure. L'utilisation des moyens de protection peut minimiser ces risques mais ils sont coûteux.
- Les réseaux de capteurs assistés par les robots Dans la démarche d'auto-déploiement, les capteurs sont déployés de manière autonome et intelligente et sont capables de changer leurs positions géographiques. Ce qui entraîne une répartition plus souhaitable. Alternativement, les robots mobiles peuvent transporter les capteurs statiques comme charge utile et les placer à des endroits prédéterminés. Cette approche est connue sous le nom déploiement des capteurs basé transport (*carrier-based*). Dans cette méthode de déploiement, les robots se déplacent à travers la zone de captage (ROI : *region of interest*) et placent les capteurs de manière intelligente, de sorte que ROI soit couverte par les capteurs. La zone sera entièrement couverte par les rayons de détection des capteurs à la fin du déploiement (Fletcher. G, 2010).

5. Les robots

Un robot est un dispositif mécatronique (alliant mécanique, électronique et informatique) accomplissant automatiquement soit des tâches qui sont généralement dangereuses, pénibles, répétitives ou impossibles pour les humains, soit des tâches plus simples mais en les réalisant mieux que ce que ferait un être humain¹.

5.1 Types de robots utilisés

Il est clair que l'aspect matériel des robots joue un rôle important dans leurs utilisations. Les robots utilisés doivent avoir la possibilité de :

- mouvement dans la zone de surveillance et supportent les contraintes de l'environnement et de l'application. Ils doivent aussi être capables de reconnaître leur environnement.
- détecter des obstacles,
- placer des capteurs dans la zone de surveillance (par exemple les robots peuvent avoir des bras ou des tournevis à visse pour le remplacement des capteurs dans les arbres).
- savoir leurs positions et celles des nœuds capteurs (exemple: être équipé d'un récepteur GPS où il peut calculer la position des nœuds capteurs en évaluant la distance entre le robot et ces nœuds (voir Chapitre 2 "la localisation des nœuds")

- posséder des capacités matérielles et logicielles pour évaluer la connectivité et la couverture.

5.2 Exemples de robots

5.2.1 Le robot "AVATAR"

La mobilité aérienne de ce type de robots (voir Figure 1.8) est souvent la raison principale qui pousse les chercheurs à le choisir pour les applications spécifiques. Ce mode de mobilité comme un oiseau dans l'environnement est souvent très utile pour résoudre certains problèmes. Corke.P et *al.*, (2006) proposent ce type de robot pour la réparation de la connectivité. Ces robots sont des hélicoptères autonomes qui portent des capteurs et qui possèdent la capacité de mobilité et de remplacement des capteurs en panne.

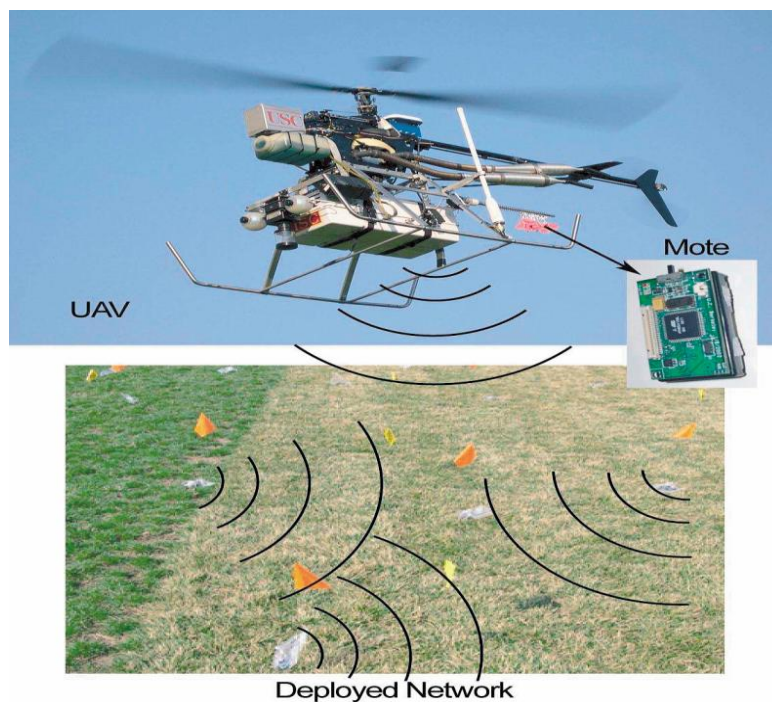


Figure 1.8.: AVATAR : hélicoptère autonome avec interface de capteurs pour le déploiement et le redéploiement des capteurs d'après Corke.P et *al.*, (2006)

5.2.2 Le robot "AMOUR" d'après Falcon Martinez. R-J, (2012)

C'est le premier scénario concernant un WSRN sous-marin, dans lequel les nœuds établissent des contacts optiques, acoustiques et utilisent la communication radio. Les capteurs (Aquanodes) dans la Figure 1.10 sont équipés d'un système de communication, d'un récepteur de captage et d'un module de traitement, et sont emballés dans des récipients étanches cylindriques. Les capteurs localisent les objets statiques et mobiles via leurs portées offertes par un modem acoustique. Ils mesurent la température de l'eau, la pression et les autres indicateurs chimiques. Les capteurs

sont ancrés avec des poids et forment un réseau de capteurs statique. Les nœuds mobiles sont de type de Robot (INTRIGUE AMOUREUSE) sous-marin, Optique, Modulaires, Autonomes. Ils ont des fonctionnalités des capteurs et, en plus, ils sont équipés d'une appareil-photo très avancée pour éviter les obstacles. Les unités de l'INTRIGUE AMOUREUSE visent à voyager autour et télécharger des données des capteurs (voir la Figure 1.9). En outre, si un événement d'intérêt est rapporté, les robots se déplacent à l'endroit de l'événement et entament un échantillonnage de l'environnement. Et, cela rehausse la capacité de l'échantillonnage dynamique du réseau. Les expériences qui ont été menées dans les océans, les rivières et les lacs ont données des résultats prometteurs.

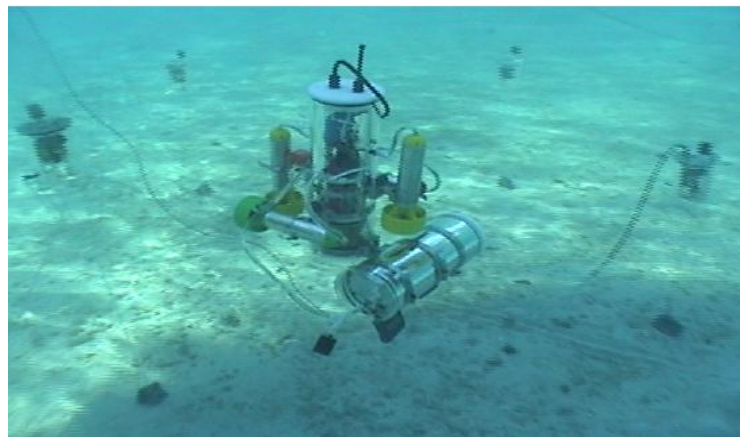


Figure 1.9: Un réseau déployé par le AMOUR.



Figure 1.10 : Aquanodes drying.

5.2.3 Robots de systèmes de jardinage

Le but sur le long terme est le développement d'une serre autonome où les robots mobiles exécutent des travaux comme les jardiniers en délivrant de l'eau et des éléments nutritifs aux plantes (voir Figure 1.11) et moissonner les fruits sur demande

(voir Figure 1.12). Les plantes sont, généralement, les tomates et les cerises implantées dans des pots dotés des capteurs du sol qui communiquent avec les robots. Les agents robotiques possèdent un système d'arrosage sur commande, équipé d'appareil-photo et de l'œil-dans-main et des logiciels d'un réseau maillé.

Pendant que les nœuds capteurs dirigent leur humidité du sol et demandent de l'arrosage par une requête, les robots sont capables d'offrir de l'eau et des éléments nutritifs pour une plante spécifique dans une localisation précise. Ils utilisent un modèle d'augmentation plante-spécifique encadré pour faire des prédictions sur l'état du fruit et enfin moissonner les tomates les plus mûres en réponse à une notification de l'utilisateur.



Figure 1.11: Robots tending to cherry tomato plants in the greenhouse.



Figure 1.12: Un robot qui récolte de la tomate ;

6. Le redéploiement utilisant les robots

Dans un environnement où les capteurs sont déployés, les capteurs peuvent être victimes de plusieurs types de pannes (panne d'énergie, incendie,.....). La panne d'un nœud peut provoquer un manque de couverture ou de connectivité. Si les capteurs sont statiques, ces trous ne seront pas comblés que si on redéploie un nœud fonctionnel au niveau de trou ou la panne. Ce nœud peut être un nœud redondant (relocation des nœuds) ou un nouveau nœud (redéploiement des nouveaux nœuds). Généralement après une certaine période de redéploiement, le réseau ne contiendra pas de nœuds redondants où la relocation ne deviendra pas utile, ce qui fait que

l'addition de nouveaux nœuds (i.e. redéploiement) sera nécessaire pour réparer les trous (voir Figure 1.13). Cette addition ne sera effectuée que par un acteur extérieur qui possède les capacités physiques nécessaires (la mobilité,...) qui lui permettent de redéployer le réseau.

A cause de développement intéressant et remarquable dans la robotique et les difficultés de déploiement dans certains environnements durs, ces unités (robots) peuvent agir comme des traiteurs de pannes dans le réseau. Mais, pour qu'ils puissent assurer leurs missions, plusieurs communications et traitements sont nécessaires. Parmi les étapes les plus importantes dans le redéploiement on cite : la détection et la réparation des pannes.

L'objectif de notre étude est de savoir comment les robots sont impliqués dans le RCSF pour qu'ils assurent le redéploiement du réseau? Notre étude est limitée à deux phases: la détection et la réparation.

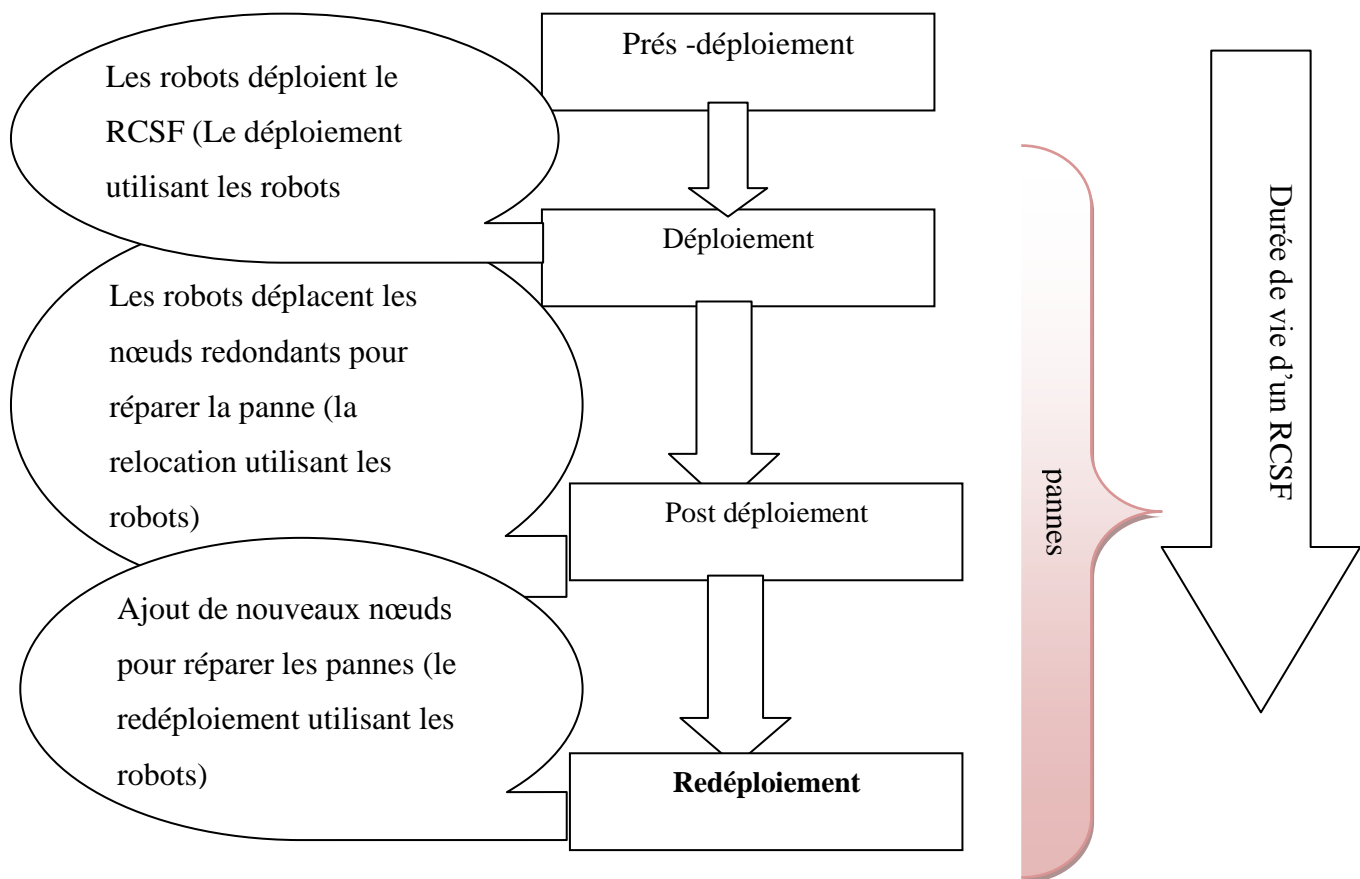


Figure 1.13: Phase de redéploiement utilisant les robots dans le processus de déploiement pendant la durée de vie de RCSF.

7. Conclusion

La panne isolée d'un capteur est la circonstance de plusieurs facteurs qui sont généralement dus aux caractéristiques physiques sensibles et faibles des capteurs. En cours de chaque phase de déploiement, des stratégies de prévention, de détection et de réparation des pannes peuvent être appliquées. Néanmoins, elles sont toujours insuffisantes pour assurer la maintenance pendant la durée de vie du réseau de capteurs. Dans un environnement dangereux où ces capteurs sont généralement déployés, l'utilisation des robots pour le déploiement, le post déploiement et le redéploiement peut être un bon choix. Donc, dans le cas où il n'existe pas de nœuds redondants, le rajout de nouveaux capteurs (redéploiement) en utilisant les robots devient l'ultime choix pour le traitement des pannes. Les robots se déplacent vers les positions des pannes et lancent un nouveau capteur dans l'emplacement de la panne.

Cependant, plusieurs questions se posent sur les mécanismes de traitement utilisant les robots: Comment les nœuds (capteurs ou robots) procèdent pour réaliser le traitement des pannes? Quelle sont les étapes nécessaires pour effectuer ce traitement? Et, quels sont les mécanismes déjà utilisées pour atteindre cet objectif.

Chapitre 2:

Classification des approches de traitement de pannes isolées utilisant les robots

1. Introduction

Dans un réseau de capteurs sans fil où les capteurs tombent en panne l'un après l'autre, on cherche à redéployer ce réseau afin d'assurer son bon fonctionnement et la continuation de ses services. D'après ce qui a été décrit dans le Chapitre 1, l'utilisation des robots pour le redéploiement est une solution idéale dans le cas d'un environnement de déploiement dangereux. Cependant, plusieurs questions concernant cette méthode de traitement vont se poser. Parmi elles, on cite : Comment le(s) robot(s) connaît les positions des pannes qui interviennent le long de la zone de déploiement (qui est généralement grande) ? Comment un robot arrive le plus rapidement possible à ces emplacements ? Comment le robot procède pour réparer un ensemble des pannes ? Plusieurs travaux de recherche ont traité cette problématique, chacun essaye de répondre à un sous ensemble de ces questions.

Le but de ce Chapitre est la présentation des approches de traitement des pannes isolées en se concentrant sur les approches de redéploiement classiques connues dans la littérature. Le Chapitre présente aussi notre propre méthode de classification de ces approches et la logique suivie pour faire cette classification.

Ce Chapitre est structuré comme suit : On commence par la définition de quelques aspects sur la relocation dans les réseaux de capteurs utilisant les robots. Pour commencer notre classification, on a préféré éclairer la nature de décisions prises dans ces solutions et les étapes générales de résolution de ce type de problèmes en terminant par la classification des solutions. La présentation de ces solutions consiste à citer leurs objectifs et leurs principes pour finalement déceler leurs apports et leurs limitations.

2. Nature et acteurs des décisions pour le redéploiement des pannes isolées utilisant les robots

Le traitement de la panne passe par plusieurs étapes, à partir de sa détection jusqu'à sa réparation. Plusieurs décisions doivent être prises entre les nœuds du réseau (capteurs, robot(s)) afin de compléter le traitement. Un certain nombre de questions se posent alors.

- Décisions sur l'existence de la panne. L'évaluation de la panne est faite comment? Et par qui?
- Vers qui et par qui les rapports de pannes sont envoyés pour commencer la tâche de réparation de la panne?
- Décision sur le robot réparateur (dans le cas de plusieurs robots) : Quels sont les critères de choix du robot réparateur? Qui sont les responsables de choix de (s) robot(s) réparateurs?
- Décision sur la stratégie de réparation: Quelle est la méthode de réparation utilisée? qui est l'élément responsable de faire le traitement nécessaire pour trouver la stratégie de réparation.

En général, pour chaque étape on cherche à savoir quelle est la décision à prendre, le décideur et ses influences sur le système en général (voir Figure 2.1).

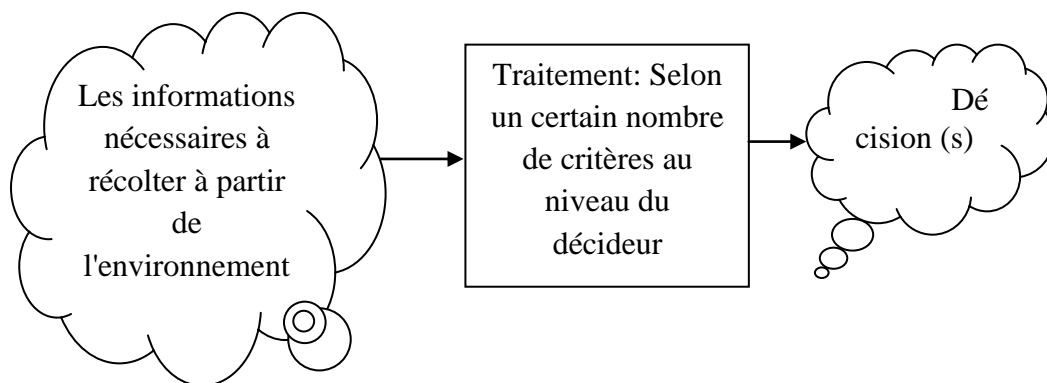


Figure 2.1: Prise de décision.

2.1 Etapes générales de traitement de pannes

A chaque étape, les nœuds (robots, capteurs) doivent prendre différentes décisions concernant l'état du réseau pour qu'ils puissent commencer le traitement adéquat. Pour mieux détecter la panne, il est nécessaire de connaître les symptômes d'une panne.

La décision de détection est prise soit par: les capteurs, le(s) robot(s), ou le sink. Ils doivent évaluer l'état du réseau à partir des informations récoltées à partir de l'environnement. Après l'étape de l'évaluation, les informations nécessaires sur l'identification de la panne doivent être calculées et envoyées au niveau de(s) nœud(s) responsable(s) de la réparation. Enfin, le calcul de la stratégie de réparation utilisant les informations requises de la détection permet de choisir le robot adéquat (selon certains critères) pour réparer la panne. Ce dernier se déplace au niveau de la panne et redéploie un nouveau capteur.

2.1.1 La détection de la panne

La détection est généralement la première étape du processus de traitement. La sémantique de la panne dépend de l'application ou de l'objectif du RCSF. Ainsi, on a plusieurs définitions :

- a) Le nœud est considéré non fonctionnel (panne matérielle): Le nœud est actif mais il n'échange pas des messages "hello " avec son voisinage (Mei. Y et *al*, 2007).
- b) Le nœud subit une attaque "DOS" (Denial of Service) (Hnin. et *al*, 2007).
- c) Le nœud atteint un niveau d'énergie très bas (Sheu. J-P et *al*, 2008).

Il existe un trou de couverture (Xu Li et *al*, 2012), (Hao. W et *al*, 2012), (Pengju. Si et *al*, 2011), (Yao. S et *al*, 2012).

- d) Un manque de connectivité: Si le nœud diffuse une information multi-saut, cette information n'arrive pas à atteindre tous les nœuds du réseau (Corke. P, 2008).
- e) Le nœud est changé soit parce qu'il atteint un niveau d'énergie très bas, ou il provoque un manque de couverture (la décision est prise selon le coup général de redéploiement) (Hao. W et *al*, 2012).

Le résultat de l'évaluation représente la matière première pour la réparation de la panne. Parmi ces informations, on cite:

- a) Les positions géographiques des nœuds en panne ou des centres des trous (Mei. Y et *al*, 2007), (Corke. P, 2008), (Hnin. et *al*, 2007), (Hao. W et *al*, 2012), (Pengju. Si et *al*, 2011), (Yao. S et *al*, 2012).
- b) Les identifiants des nœuds en panne avec les informations nécessaires pour accéder à ces nœuds (Sheu. J-P et *al*, 2008).

Selon les solutions vues, ces informations sont soit gardées et le nœud qui fait l'évaluation s'occupe de calculer la stratégie de réparation ou il doit transférer ces informations au nœud responsable de ce calcul (voir plus loin).

2.1.2 La réparation

L'objectif de la réparation est le redéploiement par le remplacement avec de nouveaux nœuds. Pour avoir de bons résultats au niveau des temps de réparation et des messages nécessaires, il faut réfléchir à une stratégie de réparation adéquate:

- En cas de plusieurs robots, la désignation de(s) robot(s) réparateur(s) le(s) plus adéquat(s) et leurs affecter des tâches. Parmi les critères de choix de robot réparateurs, on cite par exemple:
 - Le robot le plus proche.
 - Le robot responsable d'une zone.
 - Le robot qui possède un réservoir avec plus de capteurs.
- En cas de l'utilisation de plusieurs robots et le système détecte l'existence de plusieurs pannes, deux choix sont possibles:
 - Le système charge un seul robot pour réparer un ensemble des pannes (Boutina. A et *al*, 2011): Cette stratégie peut augmenter le temps d'attente maximal.

- Le système charge plusieurs robots pour un ensemble de pannes: Cette stratégie favorise la minimisation du temps d'attente maximal. Mais, s'il existe d'autres nouvelles pannes, elles doivent attendre l'arrivée de l'ensemble des robots.
- Si le robot est chargé de réparer plusieurs pannes à la fois, il est question de choisir un mécanisme qui peut être utilisé pour le redéploiement:
 - Réparer les pannes selon l'ordre de leurs interventions (garder les demandes dans une liste FIFO) (Mei. Y. et *al*, 2007).
 - Calculer de chemin de réparation le plus court qui permet la réparation de l'ensemble des pannes dans un délai raisonnable d'après Boutina. A et *al* (2011) et Yao. S et *al* (2012).

2.2 Défis existants

La réparation de la panne est une tâche critique qui doit être exécutée le plus tôt possible. Si la panne dépasse une certaine période, ses conséquences peuvent être catastrophiques. En général, le traitement de la panne pousse les nœuds à échanger un nombre important de messages, ce qui épuise l'énergie résiduelle des capteurs. Ce qui provoque la panne des capteurs. En effet, plusieurs défis apparaissent lorsqu'on veut développer une solution de traitement de pannes. Parmi eux, on cite :

- ✓ Temps de traitement de la panne: Temps maximal de réparation: C'est la borne maximale au bout de laquelle on peut avoir des conséquences néfastes.
- ✓ Le coup de réparation de la panne (le nombre de messages échangés, l'énergie perdue, le coup financier,...).
- ✓ Les caractéristiques du RCSF (les propriétés physiques des capteurs, la nature de la zone de déploiement du réseau (la surface, les obstacles, le nombre de nœuds, ...)).
- ✓ La nature des pannes isolées (le taux de pannes, leurs distributions,...).
- ✓ La centralisation ou la distribution des solutions : Selon Falcon Martinez. J-R (2012)14], quand le traitement est centralisé, le flux d'informations est concentré dans une unité centrale qui est la station de base, qui est responsable de la construction d'une vue globale du système. Le problème de cette catégorie est l'overhead des messages d'informations et de contrôle au niveau du gestionnaire, notamment à cause de la taille importante de la zone de captage. Ceci en plus du point de défaillance unique. Par contre, les algorithmes distribués permettent à chaque entité du réseau de former sa propre image de réseau en interagissant avec ses pairs. Les solutions distribuées sont plus fiables (pas de goulot d'étranglement (overhead) ou l'échec à un seul point) que les systèmes centralisés. Le problème est que les informations disponibles au niveau de chaque nœud sont limitées, par conséquent les opérations effectuées au niveau de ce dernier sont aussi limitées. Un protocole localisé est intrinsèquement distribué, mais n'a pas besoin ni d'une communication ni d'un calcul global car les deux opérations sont liées et limitées au voisinage approximatif de chaque nœud. L'avantage de ce type d'algorithme est qu'il entraîne une faible surcharge, ce qui assure l'évolutivité et l'adaptabilité de la solution.

3. Classification des approches de traitement de pannes isolées

Les solutions existantes dans la littérature qui traitent la problématique de panne isolées utilisent chacune des techniques et mécanismes différents, qui sont appliqués dans les différentes phases de la procédure de traitement de pannes. Dans cette Section, on présente quelques critères de classification, ainsi qu'un exemple de classification proposé dans la Thèse de Falcon. M, (2012) , A la fin, on présente ainsi notre modèle de classification proposé et un certain nombre de solutions dans chaque classe.

3.1 Critères de Classification

Il existe plusieurs paramètres de classification de ce type de solutions, parmi ces paramètres on cite:

- Le but de la maintenance (assurer la couverture, la connectivité, les deux)
- Le nombre des robots participants (un ou plusieurs).
- Equilibre de charges de solution : centralisée ou distribuée ou localisée
- La distribution et le mouvement des robots pour la détection et la correction des pannes
- le robot se déplace en cas de réception d'un message de panne.
- le robot est toujours en mouvement pour attendre des informations sur les pannes et les nœuds qui sont redondants.
- Les catégories des nœuds participants (exemple manager, maintenir)
- Les phases réalisées de la procédure de tolérance aux pannes : la détection, le recouvrement, le traitement.
- Le modèle de répartition 2D, 3D
- Le type de(s) robot(s) utilisé(s) (volant(s), roulant(s))

3.2 La classification selon l'équilibre de charge (Falcon. M, 2012)

Les algorithmes de redéploiement des réseaux de capteurs utilisant les robots sont classés selon l'aspect de centralisation des traitements.

3.2.1 Les algorithmes centralisés

Le flux d'informations du réseau est concentré dans une unité centrale (station de base), qui est responsable de la construction d'une vue globale du système.

Les capteurs transmettent les données demandées au contrôleur principal, éventuellement, par le biais de chemin multi hop. Le contrôleur principal est supposé être riche en ressources. Celui-ci doit alors, être en mesure d'effectuer des opérations enchevêtrées. Ses sorties peuvent être communiquées à un sous-ensemble (ou la totalité) des nœuds du réseau afin que des mesures complémentaires puissent être prises. Comme il est le cas de la solution de Sheu. J-P et *al*, 2008.

Le problème lors de la collection des données et la propagation des sorties est l'overhead généré en messages, qui défavorise la scalabilité. En plus, dans le cas de la panne du contrôleur, cet algorithme sera stop.

3.2.2 Les algorithmes distribués

Les algorithmes distribués permettent à chaque entité du réseau de former sa propre image de réseau en interagissant avec ses pairs. Il n'y a pas d'unité centrale chargée d'une tâche globale. Tout nœud peut requérir des données de chaque autre nœud (comme dans un protocole centralisé, mais maintenant l'algorithme fonctionne d'une façon répartie) ou de ses voisins immédiats (jusqu'à k sauts).

Un avantage est la distribution de la charge entre les nœuds de réseau. Les solutions distribuées sont plus fiables (pas de goulot d'étranglement (*overhead*) ou l'échec à un seul point) que les systèmes centralisés rencontrent. La surcharge provoquée par le passage de messages pourrait encore être significative.

Le problème de cette solution c'est que les ressources disponibles au niveau de chaque nœud sont limitées, par conséquent les opérations effectuées au niveau de ce dernier sont limitées. Il y a aussi une contrainte sur les ressources disponibles localement au niveau de nœuds de capteurs, ce qui limite l'étendue des opérations qui peuvent leur être déléguées. Parmi les solutions qui utilisent ce principe la solution de Mei Y. et *al*, (2007).

3.2.3 Les algorithmes localisés

Un protocole localisé est intrinsèquement distribué, mais n'a pas besoin ni d'une communication ni d'un calcul global car les deux opérations sont liées et limitées au voisinage approximatif de chaque nœud.

L'avantage de cet algorithme est qu'il entraîne une faible surcharge. Cela, assure l'évolutivité et l'adaptabilité de la solution. Tandis que les résultats ne reflètent pas la structure du groupe (, par exemple, un groupe de capteurs n'a pas les mêmes résultats, malgré que chaque capteur effectue les mêmes calculs). Ce compromis d'*overhead-précision* est souvent jugé en fonction des limites physiques des capteurs. Comme il est le cas de la solution de Grid R3S2(Xu. Li et *al* 2013).

3.3 Classification de solutions de traitement des pannes isolées

Il existe différentes méthodes pour le classement des approches de redéploiement. Dans notre étude, on a classé ces approches selon le mode de détection, le mode de réparation et le nombre de robots participants.

Les approches de redéploiement sont scindées en trois classes selon le responsable de détection :

- Les capteurs détectent les pannes.
- Le(s) sink(s) détectent les pannes.
- Le(s) robot(s) détectent les pannes.

3.3.1 Les capteurs détectent les pannes

Dans ce type des solutions, les capteurs détectent localement les pannes, cela peut être effectué généralement par l'envoi de messages « hello ». Après qu'ils détectent localement les pannes, ils ont deux choix, soit:

- Ils envoient les rapports des pannes au(x) robot (s): Chaque robot attend sans exploration un ordre ou un message help pour qu'il commence la réparation de la panne.
- Ils attendent l'arrivée d'un robot et l'informent de la situation locale du réseau (avec exploration).

Deux classes apparaissent alors, sans exploration et avec exploration.

3.3.1.1 Sans exploration

Dans ce mode, les capteurs prennent la responsabilité de l'évaluation de la panne, chaque capteur interagit avec ses voisins, un gestionnaire est distingué pour recevoir les rapports de pannes (voir Figure 2.2) :

La Figure 2.2 schématise d'une façon générale le processus de détection et de réparation de ce mode. Après l'occurrence de la panne, le(s) nœud(s) voisin(s) détecte(nt) la panne et envoie(nt) ensuite un rapport de pannes vers le gestionnaire via une transmission multi sauts. Le gestionnaire commence le processus de réparation selon sa stratégie de réparation et informe le robot pour réparer la panne. Dans le cas de plusieurs robots, il choisit le(s) robot(s) réparateur(s) approprié(s). Chaque robot réparateur se déplace vers l'emplacement de la panne et remplace un nouveau nœud.

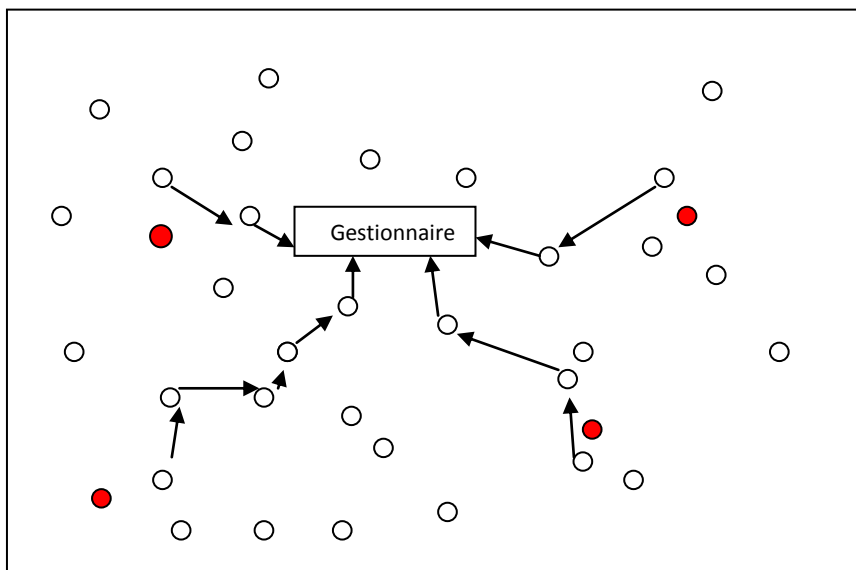


Figure 2.2: Processus général de traitement de pannes selon le mode "sans exploration".

Quand les capteurs détectent l'existence d'une panne, ils envoient les informations nécessaires (les rapports de pannes) soit vers:

- le sink,
- un robot manager,
- le robot réparateur,

d) un robot parmi l'ensemble des robots.

Les sections suivantes sont dédiées à l'étude et à la description de comportements possibles des systèmes de traitement de pannes selon chaque cas.

A) Les capteurs envoient les rapports vers le sink

Lors de l'occurrence de la panne, les capteurs envoient les rapports de pannes vers le sink, qui informe et ordonne le(s) robot(s) pour la réparation de l'ensemble des pannes. Cependant, on remarque la nature centralisée du traitement, ce qui permet au sink d'avoir une vue générale sur l'état global du réseau. En échange, il provoque d'autres problèmes comme l'overhead au niveau du sink, et ajoute un traitement supplémentaire à ce dernier.

La stratégie de réparation diffère selon le nombre de robots réparateurs, et leurs localisations, celui qui s'occupe du calcul de la stratégie de réparation (le sink ou le robot désigné pour la réparation), où la décision sera prise (par coopération entre les robots). De ce fait, des sous classes apparaissent selon les différents critères: le nombre de robots, les emplacements des robots réparateurs (au niveau du sink, loin du sink) ou selon le responsable du calcul de la stratégie de réparation.

Parmi les approches trouvées dans la littérature, on présente l'approche de Jang-Ping Sheu (Sheu, J-P et *al*, 2008) qui utilise ce mode de traitement. L'objectif de cette solution est le remplacement des nœuds à basse énergie par d'autres fonctionnels, sans avoir besoin d'utiliser de la position géographique. Cette solution fonctionne comme suit :

- La création de chemins de routage : Les capteurs sont initialement déployés dans la zone de captage d'une façon qui leur assure au moins la connectivité. Le sink diffuse à ses voisins la demande "RouteCreate" pour que chaque nœud du réseau construise un chemin de routage vers ce dernier (routing path). Les capteurs voisins du sink et lors de réceptions de ce message, rediffusent à leur tour leurs identifiants avec le nombre de sauts incrémenté, aux nœuds voisins, et gardent l'identifiant du sink comme "Uplink"(son prochain nœud vers le sink: Dans ce cas, c'est le sink). Les autres capteurs, et lors de réceptions de la demande "RouteCreate", gardent l'ID du nœud transmetteur pour qu'il soit leur "Uplink". Après, ils rediffusent leurs "ID" avec le nombre de sauts incrémenté. S'ils reçoivent plusieurs messages de plusieurs nœuds voisins, ils choisissent l'ID du nœud qui possède le nombre de sauts minimaux vers le sink (voir Figure 2.3).

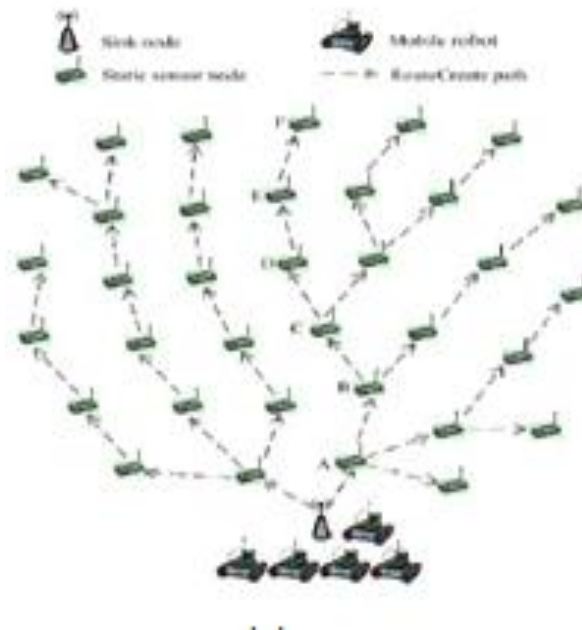


Figure 2.3: Création de chemins de routage.

- L'envoi de message help et la construction de chemin de navigation: Quand le nœud détecte que son niveau d'énergie est bas, il envoie un message "help" qui contient son identifiant à son "UpLink". Le nœud récepteur du message "help" ajoute son "ID" au chemin de navigation et renvoie à son tour ce message à son "UPLink", jusqu'à son arrivée au sink. Lors de réception de ce message ("help", chemin navigation) par le sink, ce dernier envoie le chemin de navigation au robot désigné.
- La réparation : Suivant le chemin de navigation, le robot commence sa mission de réparation. Pour qu'il puisse se diriger vers le nœud défaillant, les capteurs de chemin de navigation envoient des balises pour attirer le robot. La transmission des messages se fait par deux niveaux d'énergie, le niveau haut quand le robot est loin du capteur et le niveau bas quand le robot est proche du capteur.

- Dans le cas de détection de plusieurs pannes, le sink choisi entre deux situations: soit il attribue pour chaque panne un robot ou il utilise un seul robot pour réparer toutes les pannes (utilisation de la stratégie de réparation 1-TSP-SELPD (Falcon. R, 2010)). Les auteurs de cette solution discutent aussi l'utilisation de plusieurs robots pour la réparation de plusieurs pannes, où ils ont remarqué l'existence d'un compromis entre l'augmentation du nombre de robots, le temps de réparation et le taux d'interférences entre les signaux. A chaque fois que le nombre de robots augmente, le temps de réparation s'améliore. Par contre, si plusieurs robots prennent le même chemin, cela engendre une augmentation dans le nombre d'interférences entre les signaux.

Discussion

L'avantage de cette solution est que la période de traitement de la panne peut être très courte car le nœud sera remplacé avant qu'il expire toute son énergie. Les robots et les capteurs n'ont pas besoin d'utiliser la position géographique, ce qui minimise le coup de déploiement des capteurs et des robots. Le problème de cette approche est qu'elle traite un seul type de pannes, la panne d'énergie. Les autres types de pannes ne sont ni détectées ni traitées. De plus, elle nécessite une transmission énorme de messages, surtout quand le robot navigue pour réparer les pannes; même le nœud à basse énergie doit échanger des messages malgré son niveau bas d'énergie. Si ce nœud épuise toute son énergie avant qu'il puisse attirer le robot réparateur vers sa position, le robot n'arrive pas à atteindre la position de la panne. De plus, si ce dernier est loin, le nombre de nœuds qui participent augmente avec la distance et cela augmente l'énergie consommée par le réseau, ce qui représente pour la solution un handicap de passage à l'échelle. Les nœuds voisins du sink doivent prendre la responsabilité de transmission de toutes les requêtes. Cela engendre deux problèmes principaux: les interférences des signaux causés par la transmission des signaux surtout lorsque le nombre de pannes atteint un certain seuil, et l'épuisement d'énergie de ces nœuds à cause des transmissions des messages.

Synthèse

D'après la solution vue et le mode de détection, il ressort les avantages et les inconvénients suivants :

- ✓ Temps de réparation minimal, et calcul global de la stratégie de réparation
- ☒ Le traitement centralisé et la charge supplémentaire au sink.

B) Les capteurs envoient les rapports vers un robot manager

Dans ce cas, le robot manager traite les demandes et choisi le robot réparateur adéquat pour réparer la panne. Il est préférable que le gestionnaire soit fixe (ne change pas de position) pour éviter les problèmes de routage vers ce dernier, ou les problèmes de diffusions de sa position vers tous les nœuds de sa zone.

Parmi les solutions proposées par Mei. Y et al, (2007), l'algorithme centralisé suit ce mécanisme de traitement des pannes. Le but de cette solution est la détection et la réparation des pannes fonctionnelles des capteurs. Ces nœuds se trouvent dans un champ de captage 2D où les capteurs et les robots sont déployés. Chacun possède la possibilité d'être localisé.

- ✓ Initialement, chaque nœud (robot, capteur) identifie son rôle dans le réseau: premièrement le gestionnaire diffuse sa position vers tous les nœuds du réseau, les autres robots (mainteneurs) envoient leurs positions vers le manager via un routage géographique. Les capteurs à leurs tours diffusent leurs positions à leurs voisins. Chaque capteur choisi parmi ses voisins et selon les messages reçus de ces derniers, le capteur le plus proche comme un gardien et envoie à ce dernier un message de confirmation et de cette façon la relation gardien gardé est établie.
- ✓ Détection: Les capteurs diffusent périodiquement des balises, si un gardien n'a pas reçu de balise de son gardé pendant trois périodes, il déclare la panne de ce dernier et transmet un rapport de panne vers le gestionnaire via le routage géographique. Le gestionnaire choisi le mainteneur le plus proche pour réparer la panne, et envoie à ce dernier un ordre de réparation de la panne.
- ✓ Réparation: Le mainteneur à son tour se déplace vers la position de la panne. Après une certaine distance de son déplacement, ce robot s'arrête et envoie sa nouvelle position au gestionnaire Cette opération (mouvement- arrêt) sera répétée par ce robot jusqu'à son arrivée à la position de la panne. Cela va permettre au manager d'envoyer d'autres ordres qui seront traités par le robot mainteneur ordre après ordre (les ordres seront classés suivant une liste FIFO). Lors de son arrivée, il remplace le nœud défaillant par un nouveau nœud.

Discussion

Cette approche propose un traitement centralisé des demandes qui sera effectué par un robot, et cela minimise la charge de traitement des pannes par rapport à la première classe (traitement par sink). Le mécanisme de sélection de mainteneur se base sur sa position géographique ce qui permet de choisir le robot le plus proche de la panne, alors il y a moins de mouvements inutiles et le temps de réparation est très réduit. Selon Mei. Y et al, (2007), le problème cet algorithme est qu'un seul gestionnaire est chargé de traiter toutes les demandes de réseau, ce qui génère le problème d'overhead au niveau de ce dernier. Selon Fletcher.G (2010), le nombre de messages nécessaires pour la détection et la réparation de la panne est très important. Ce qui limite les capacités de cette solution lors de passage à l'échelle.

Synthèse

- ✓ Par rapport au mode où le sink est le gestionnaire: il y a moins de charge sur le sink et les capteurs voisins de ce dernier.
- ✓ Temps de réparation minimal, moins de mouvements inutiles.
- ✓ Traitement centralisé, nécessite un taux de message très élevé, moins de résistance lors de passage à l'échelle.

C) Les capteurs envoient les demandes de réparation vers le robot réparateur

Dans ce cas, les capteurs envoient les rapports de pannes vers le robot réparateur. Ce dernier traite toutes les demandes et choisi la stratégie de réparation adéquate pour réparer l'ensemble des pannes. Dans le cas de l'utilisation d'un seul robot, ce dernier doit recevoir toutes les demandes de réparations et les traiter. Même si le robot est en phase de réparation, il doit être informé des pannes qui interviennent durant cette phase, soit les capteurs attendent la fin de son traitement et envoient leurs rapports,

soit le robot après un certain seuil de déplacement (ce seuil peut être une distance ou une période), il informe les capteurs et attend un certain moment pour que les capteurs puissent envoyer d'autres rapports. Le problème est que le robot change sa position à chaque instant, et cela pose un problème lors de l'utilisation de routage géographique où il faut à chaque instant informer les capteurs de la nouvelle position du robot. L'utilisation de la diffusion pose aussi le problème de diffusion. La qualité de ces solutions dépend du routage utilisé. Il est à remarquer que le nombre de messages et mouvements sont très importants.

Dans le cas de l'utilisation de plusieurs robots, les capteurs doivent toujours envoyer les rapports de pannes vers le robot adéquat. Dans ce cas, les capteurs désignent leur gestionnaire ou désignent le robot le plus proche pour la panne pour assurer la réparation. Les capteurs et lors des réceptions des différentes informations sur les robots (exemple: leurs positions géographiques, la capacité de leurs réservoirs,...), ils envoient toujours les rapports de pannes au robot le plus proche. On peut avoir d'autres critères de choix comme la capacité de réservoir; l'essentiel est que les informations nécessaires pour le choix du robot seront diffusées par les robots, et les capteurs choisissent vers quel robot ils transmettent ces informations.

Dans l'algorithme distribué fixe de Mei Y. et *al*, (2007), les capteurs choisissent le robot le plus proche pour lui envoyer les rapports de pannes. La zone est subdivisée en sous zones égales, chaque robot s'occupe des traitements des pannes d'une sous zone. Le robot est au même temps un manager et un mainteneur. Initialement, les robots se déplacent vers le centre de leurs zones et diffusent vers ces centres. Les capteurs sélectionnent le robot le plus proche de sa zone comme "Myrobot", pour lui envoyer les rapports de pannes. La relation gardien gardé est attribuée entre les capteurs avec le même principe que la solution centralisée de Mei. Y et *al*, (2007). Quand une panne intervient, le gardien envoie son rapport à son myrobot (le robot de sa zone). Le robot se déplace vers l'emplacement de la panne et remplace le nœud défaillant. Lors de son déplacement, le myrobot rediffuse sa nouvelle position après une certaine distance de son déplacement (pour informer les capteurs de sa zone de sa nouvelle position). Le problème de cet algorithme est que les capteurs choisissent leurs robots réparateurs "myrobot" une seule fois dans la phase d'initialisation. Ce dernier change souvent sa position au niveau de sa sous zone, où on peut avoir le cas où ce robot n'est pas toujours le robot le plus proche pour réparer une panne. Cela provoque des mouvements inutiles et augmente le temps de réparation de la panne, comme il est montré dans l'exemple de la Figure 2.4.

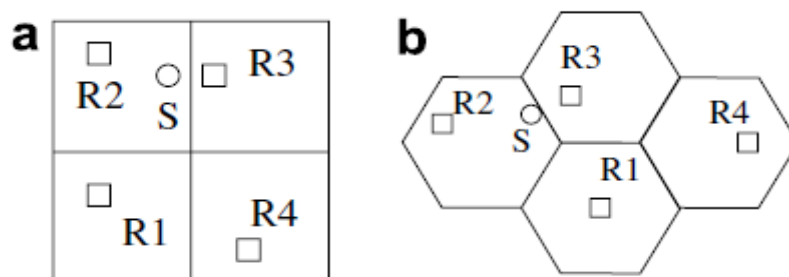


Figure 2.4: Subdivision de la zone de couverture en sous zones ;

Dans cette Figure, si le nœud S tombe en panne, le robot R2 doit se déplacer vers S alors que R3 est le plus proche, dans ce cas le choix de R2 fait perdre plus de temps que R3.

Mei. Y *et al*, (2007) proposent une autre solution qui règle ce problème, c'est l'algorithme distribué dynamique. Comme son nom l'indique, les limites entre les zones sont construites dynamiquement comme un graphe véronique (voir Figure 2.5).

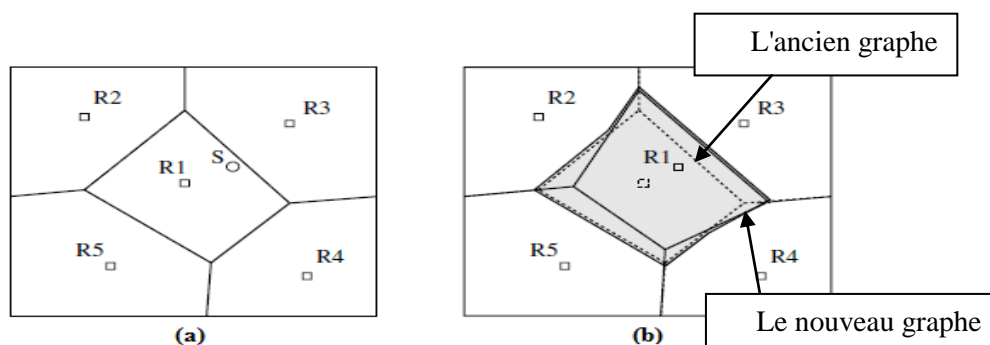


Figure 2.5: Un graphe véronique dynamique.

La Figure 2.5(a) montre un graphe véronique original. On suppose que le nœud S tombe en panne dans la sous zone de R1. La Figure 2.5(b) montre le déplacement de du robot R1 pour remplacer S. par conséquent, le graphe véronique de R1 change.

Discussion

Concernant la différence entre l'algorithme dynamique et l'algorithme fixe, les messages diffusés suivant l'algorithme par le robot lors de son déplacement ne concernent pas que les nœuds qui sont actuellement dans sa sous zone au moment de la diffusion. Mais, certains nœuds voisins de sa sous zone peuvent changer leurs MyRobots (sous zones) selon les nouvelles positions reçus des "myrobot" des sous zones voisines lors de déplacement de leurs robots. De cette façon, les nœuds mettent à jour leurs MyRobots dynamiquement à celui le plus proche.

Par rapport à l'algorithme fixe, l'algorithme dynamique peut avoir un overhead plus élevé en messages étant donné que quelques nœuds doivent faire le relai de ces messages (i.e les capteurs de relai perdent leurs énergies dans la transmission de ces messages). L'algorithme dynamique peut réduire de 10% les mouvements possibles par rapport à l'algorithme fixe. Mais, ce dernier s'adapte mieux aux pannes distribuées irrégulières (le robot le plus proche est toujours chargé par le traitement).

Synthèse

- ✓ Dans le traitement distribué, on a moins de mouvements inutiles, le temps de réparation minimal, il résiste plus au passage à l'échelle.
- ☒ Beaucoup de messages, une charge supplémentaire sur les capteurs qui possèdent moins de ressources pour décider du choix du robot réparateur.

D) Vers un ensemble de nœuds (capteurs, robots) qui coopèrent entre eux pour prendre les décisions sur la panne

Dans ce mode, le calcul de la stratégie de réparation nécessite une coopération entre l'ensemble des nœuds (capteurs, robots). Cette coopération sera effectuée soit entre les robots, ou les robots et le sink. L'objectif de coordination (coopération) peut être :

- ✓ La décision si la panne nécessite l'intervention du robot ou pas, comme l'exemple de l'approche de Hnin. Y et *al* (2008).
- ✓ La prise de décision sur le robot réparateur.
- ✓ La définition de la stratégie de réparation....

La coordination nécessite une étape de communication importante afin d'arriver à un accord, étant donné que les robots sont éparpillés dans la zone de captage. Cela peut engendrer des charges supplémentaires lors de transmissions multi sauts de ces message, la charge augmente avec le nombre de robots. Jusqu'à présent, on a pas trouvé dans la littérature des solutions qui utilisent la coordination entre les robots pour le choix de robot. La coopération peut se faire aussi sans transmission des messages multi sauts s'il existe des rencontres ou des entrevues entre les robots ou les robots se déplacent vers les nœuds responsables et récupèrent la réponse.

Hnin.Y et *al* (2008) proposent une solution de remplacement de nœuds défaillants par d'autres fonctionnels. Ce travail, s'adresse à une classe particulière d'attaques du DOS (Denial of Service). On remarque une coopération entre la station de base (sink) et les robots réparateurs afin que le robot décide s'il répare une panne ou non. Dans un champ à deux dimensions où les capteurs sont déployés, les robots sont distribués uniformément et aléatoirement. Chaque robot transporte un certain nombre des capteurs fonctionnels. Les robots voyagent avec une vitesse constante et possèdent la possibilité de se localiser. Le nombre de robots est très petit par rapport au nombre des capteurs et ces derniers communiquent entre eux via des liens sans fil. La zone de déploiement est subdivisée en sous zones selon le nombre de robots où chaque robot s'occupe de la réparation des pannes dans sa sous zone. Premièrement, les robots voyagent chacun au centre de sa sous-zone et diffuse son emplacement à tous les capteurs de sa sous-zone. Chaque capteur sélectionne le robot le plus proche comme *my Gardien* pour lui envoyer les rapports de pannes. Les capteurs diffusent aussi leurs emplacements à leurs nœuds voisins d'un saut dans leurs sous-zones. Chaque capteur recevant un message de chacun de ses voisins, choisi son voisin le plus proche comme son *nœud double*, puis il envoie un message de confirmation à ce dernier, (même principe que l'algorithme distribué fixe (Mei. Y et *al*, 2007)). Après initialisation, chaque capteur envoie périodiquement un message "hello" à son nœud double. Si un nœud ne reçoit pas de message de son nœud double pour un certain délai, il envoie un rapport de panne au robot de sa sous zone. Puis, il sélectionne un nouveau nœud double parmi ces voisins comme dans l'étape d'initialisation. Le robot renvoie le rapport d'échec vers la station de base et attend la décision de la station de base. Si la station de base et selon les paramètres du rapport, décide que c'est une attaque DOS, elle envoie son accord au robot pour qu'il commence la réparation. Le robot et lors de sa réception de l'accord, il se déplace à l'emplacement du nœud manqué et le remplace par un autre nœud de son réservoir. Quand le robot se déplace vers un nœud défaillant, il peut encore recevoir d'autres demandes de réparation de la

station de base. Le robot enregistre ces demandes dans une liste Fifo, et traite les demandes selon l'ordre d'interventions.

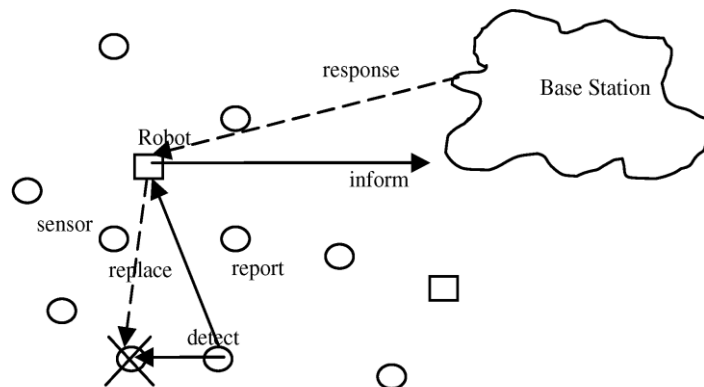


Figure 2.6: Stratégie de traitement des pannes dans l'approche de Hnin.Y et *al* (2008).

La Figure 2.6 présente une vue générale du mécanisme de traitement de la Figure 2.5 déjà expliqué. Les cercles représentent les capteurs et les carrés représentent des robots. Un nœud détecte l'échec d'un nœud double (montré par une croix sur le cercle) puis il transmet un rapport d'échec au robot. Un robot envoie le message de l'échec à la station de base.

Synthèse

- Cette solution engendre un traitement distribué, moins de mouvements inutiles, un temps de réparation minimisé, mais moins de résistance au passage à l'échelle par rapport à l'envoi de messages vers le robot réparateur.
- Beaucoup de messages, une charge supplémentaire sur les capteurs, des messages supplémentaires à cause de la coopération.

Comparaison entre les différentes sous classes du mode sans exploration

Dans le cas où les demandes sont envoyées vers des emplacements bien précis dans la zone de captage (vers le sink, vers le robot gestionnaire qui est généralement fixe), il s'agit de la gestion centralisée des pannes. En général, cette stratégie permet un calcul global de la stratégie de réparation mais un overhead au niveau du gestionnaire, moins de résistance au passage à l'échelle (plus de messages, plus d'énergie consommée, plus de temps de détection, un risque élevé de perte de messages). Dans le cas où le sink est un gestionnaire, cela ajoute des charges supplémentaires à ce dernier qui doit en général récolter les données observées, et des overheads supplémentaires à son niveau.

Concernant la gestion distribué, les décisions sont prises d'une manière distribuée et cela permet d'éviter le problème d'overhead au niveau du gestionnaire et une résistance meilleure lors de passage à l'échelle, quand la zone est subdivisée en sous zones selon le nombre de robots. Mais dans cette approche, les capteurs doivent envoyer les rapports à une unité mobile ou des unités mobiles (robots) pour minimiser le nombre de messages transmis. La reconnaissance de la position de robot est

nécessaire et cela augmente aussi le nombre de diffusions. En effet, le protocole de routage dans ce cas représente un grand déficit. La table 2.1 présente une comparaison entre les différentes sous classes dans la classe sans exploration.

| | Pourcentage de participation des capteurs dans le traitement | Mouvement inutile | Résistance à la scalabilité |
|----------------------------------|---|--|--|
| Vers le sink | 1 | Selon s'ils retournent toujours au niveau du sink ou non | Moins de résistance |
| Vers le robot manager | 2 | Plus | Bonne résistance |
| Vers e robot réparateur | 4 | Moins | Acceptable |
| Décision avec coopération | 3 | Moins | Moins de résistance à cause à la coopération |

Table 2.1: Comparaison entre les sous classes du mode sans exploration.

Résumé

Le mode sans exploration permet :

- Moins de mouvements inutiles.
- Un temps de réparation minimal.
- Plus de messages qui dépendent essentiellement du taux de pannes.
- Plus de charge sur les capteurs qui peut être élevée (consommation d'énergie,...).
- Moins de résistance au passage à l'échelle.

3.4.1.2 Avec exploration

Dans ce mode, les capteurs évaluent d'une manière locale l'état des pannes (commence la détection). Mais contrairement au premier mode (sans exploration), ils gardent ces informations à leurs niveaux jusqu'à l'arrivée d'un robot. Les robots sont toujours en exploration dans l'objectif de récupérer les rapports de pannes, et interviennent immédiatement pour les réparer.

Parmi les solutions qui utilisent ce mode, on distingue l'approche de Corke. P, et *al* (2006). Ce projet est le résultat de collaboration de trois laboratoires de robotique (CSIRO, USC, et Darmouth). Il représente une solution de déploiement, de mesure de connectivité et de réparation en cas de manque de connectivité utilisant un véhicule aérien automatique (AVATAR).

Le déploiement des capteurs se fait dans un champ 3D, où chaque nœud possède une position (X, Y, Z) bien déterminée par la station de base (déploiement déterministe). Après qu'Avatar termine le déploiement des capteurs, le robot mobile

mesure la connectivité. Deux méthodes sont proposées, l'une est appelée (ping_based) et l'autre est basée sur le passage d'un jeton (token_passing):

- Pour la première méthode (ping_based), un message multi-hop est diffusé vers les capteurs au début de chaque expérience. Chaque nœud diffuse un certain nombre de messages "pings" et calcule le nombre de réponses (qui doit être égal au nombre de pings envoyés).
- L'objectif de la deuxième méthode (token_passing) est le calcul des composantes connectées dans le réseau déployé. Chaque capteur lance un jeton incrémenté après chaque période (30 secondes, selon la solution). Si un nœud reçoit un autre jeton supérieur à celui qui se trouve à son niveau, il le garde. La taille des jetons envoyés par les capteurs dans la deuxième méthode est supérieure aux pings diffusés dans la première.

Pour la collecte des données, le robot Avatar interroge, pendant son balayage du champ de captage, les capteurs pour récupérer le nombre de réponses calculées au niveau de chaque nœud où il collecte les jetons des capteurs et identifie les groupes non connectés. Les positions des jetons collectés peuvent être utilisées pour déterminer les régions à réparer.

Cette solution demande la transmission de plusieurs messages échangés par les capteurs lors de détections des pannes. Les données sont collectées au lieu d'être envoyées via un protocole de routage par une transmission multi-sauts et cela minimise une charge considérable effectuée par les capteurs. Mais, il reste le problème des interrogations des capteurs effectuées par les capteurs malgré que la panne puisse ne pas exister. L'algorithme d'exploration et la structuration au niveau du réseau de capteurs jouent un rôle important pour l'amélioration de la durée d'exploration et les communications locales nécessaires entre les robots et les capteurs pour détecter une panne.

- **L'exploration**

L'exploration d'un environnement inconnu est un problème fondamental en robotique mobile. Elle permet, en particulier, de construire une carte utile dans de nombreuses applications (reconnaissance, sécurité, assistance, etc) (Boutina. A et *al* (2008), ou aussi dans la découverte des êtres humains, ou de survivants, ou dans l'exploration des autres planètes, ou dans la prise de vues aériennes.

Avant de passer aux solutions qui utilisent ce mode, on propose dans la section suivante quelques définitions concernant l'exploration.

Définition 1: L'exploration

Selon Maxim. A et *al* (2005), le problème d'exploration est la découverte de toute la zone traversable de l'environnement par le robot. Un robot est dit qu'il a exploré l'environnement complètement, si à un moment donné pendant son mouvement il ne reste aucun emplacement dans l'environnement qui n'était pas sous la portée de captage du robot. L'exploration peut être vue comme la phase initiale de la couverture dynamique.

Définition 2: La couverture statique et la couverture dynamique

Selon Maxim. A et *al* (2005) [2], le problème de couverture est la maximisation de la région totale couverte par les capteurs statiques. La couverture statique est le problème de déploiement des capteurs par le(s) robot(s) dans une configuration statique, tel que chaque point de l'environnement est sous l'ombre des capteurs du robot à chaque instant du temps. Autrement, la couverture dynamique est adressée par l'algorithme qui explore et non plus 'couvre' l'environnement avec un mouvement constant qui ne suit pas nécessairement une configuration particulière ni nécessairement un modèle particulier de traversée.

Définition 3: Couverture sur un graphe

Selon Maxim. A et *al* (2005), la couverture sur un graphe est le fait de visiter tous les sommets d'un graphe. La performance d'un algorithme de couverture est mesurée en utilisant le temps de couverture, métrique définie comme suit:

Définition 4: Temps de couverture

Le temps de couverture est mesuré par le nombre d'arêtes traversés de telle sorte que chaque sommet du graphe est visité au moins une fois, pour dire que le graphe est couvert (Maxim. A et *al*, 2005).

Définition 5: Exploration sur un graphique

L'exploration sur un graphique est l'acte de traverser chaque arête du graphe. Un algorithme d'exploration est évalué en utilisant la métrique de temps (Maxim. A et *al*, 2005).

Définition 6: Temps d'exploration

Le temps de l'exploration est mesuré en termes de nombre de nœuds (sommets) traversés tels que chaque sommet est traversé au moins une fois. Il résulte de ces définitions que l'exploration est un sur-ensemble de la couverture (Maxim. A et *al*, 2005).

- **L'exploration et le traitement des pannes**

Dans l'exploration, si le(s) robot(s) cherche(nt) à découvrir des pannes, l'algorithme d'exploration va lui permettre de contrôler chaque point (coin) dans son environnement en parcourant le moins de distance en moins de temps.

D'autre part, il est clair que l'exploration nécessite une communication locale au niveau de chaque sous zone. Dans la plupart des solutions étudiées, cette communication nécessite l'arrêt du robot.

L'objectif de l'exploration est la découverte de nouvelles pannes afin de les réparer. Dans ce cas, la répétition infinitive de ces algorithmes est nécessaire afin d'assurer le traitement des pannes au long de la durée de vie du réseau de capteurs. Si c'est le cas, plusieurs difficultés peuvent apparaître, parmi elles on cite :

- Quel est la nature du graphe parcouru par le(s) robot(s) et qui assure l'exploration de toute la zone?

- Quel est le mécanisme ou l'algorithme qui assure le parcours de toute la zone (le parcours de ce graphe) et dans un délai limité.
- Es ce que la répétition de l'algorithme d'exploration suivant un graphe ou le chemin donné assure que : la durée maximale d'attente entre deux visites successives d'un nœud (ou d'un groupe de nœuds) ne dépasse pas un certain seuil ?

Plusieurs questions peuvent se poser concernant l'exploration, parmi elles:

- 1) La nature de la trajectoire
- 2) La distance entre les arrêts (nécessaire pour la communication)
- 3) La stratégie de sélection d'une nouvelle direction
- 4) Si plusieurs robots sont supportés, la non terminaison de l'exécution de l'algorithme est à considérer, problème de blocage, résistance aux obstacles.

L'utilisation de plusieurs robots peut accélérer la découverte des pannes lors de l'exploration. Mais dans ce cas, les algorithmes d'exploration doivent assurer la distribution homogène des robots et la gestion de concurrence dans le cas de l'existence de plus d'un robot dans le même emplacement pour réparer une panne.

Parmi les trajectoires qui peuvent être suivie par le(s) robot(s), on cite :

- ✓ Trajectoire aléatoire,
- ✓ Trajectoire sous forme de serpent ou spirale,
- ✓ Trajectoire sous forme d'une grille.

A) Trajectoire aléatoire: Dans l'algorithme de relocation R3S2 (Xu. li et *al*, 2012), le choix de direction ne suit aucune règle, i.e. Les robots se déplacent tout au long de l'environnement et communiquent avec des capteurs à proximité (proxy) pour déterminer l'emplacement des capteurs passifs et les trous. Le même principe est suivi dans l'algorithme de relocation aléatoire proposé par (Mei. Y et *al* (2007). Selon Chang. C et *al* (2007), si la stratégie d'exploration aléatoire est utilisée pour le déploiement, le schéma de déploiement aléatoire est facile et simple à implémenter. Mais, pour assurer la couverture, le nombre de capteurs déployés doit être largement supérieur au nombre de capteurs requis. Le déploiement aléatoire des nœuds stationnaires peut donner un RCSF inefficace du fait que, quelques régions peuvent avoir une grande densité des capteurs tandis que d'autres possèdent une faible densité. Cette stratégie de recherche de pannes et de redéploiement du réseau peut engendrer le cas de l'existence des zones qui sont visitées plusieurs fois et d'autres moins visitées et cela provoque les problèmes suivants:

- Les zones de l'environnement n'ont pas la même chance d'être visités.
- Beaucoup de mouvements inutiles génèrent un temps de détection très grand et provoque aussi beaucoup de messages à cause des communications locales supplémentaires.

L'augmentation du nombre de robots peut accélérer le délai d'exploration, par contre elle peut aussi intensifier les communications locales supplémentaires.

Afin d'atténuer les effets de cette stratégie aléatoire, de minimiser le temps de traversée, et réduire les délais de réparation de la couverture, les auteurs de R3S2 proposent un autre algorithme de relocation Grid R3S2 (Xu. Li et *al*, 2012).

B) Trajectoire sous forme de serpent et spirale : Fletcher.G (2010) explique le principe d'exploration utilisé dans l'algorithme SLD de Chang. C et *al* (2007) et Batalin. A et *al* (2003). SLD (Snack-Like Deployment) est un algorithme de déploiement où l'exploration est sa première phase. Le but de l'exploration dans cet algorithme est la découverte des zones non couvertes. L'algorithme SLD effectue le déploiement déterministe des capteurs (voir Figure 2.7).

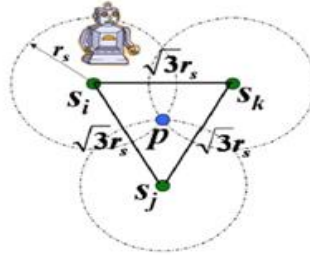


Figure 2.7: Déploiement déterministe des capteurs s_i, s_k, s_j qui assurent la couverture d'après Chang.C et *al* (2007).

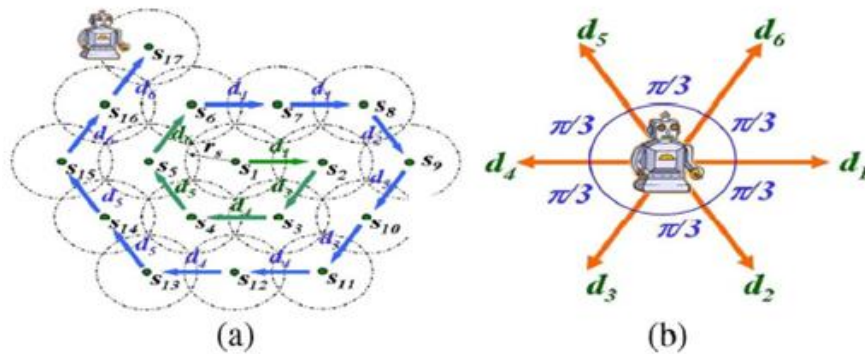


Figure 2.8: Mouvement spirale d'après Chang. C et *al* (2007).

Dans la trajectoire sous forme spirale, le robot choisi une direction parmi six directions proposées (voir Figure 2.8(b)). La distance entre les arrêtes étant de $\sqrt{3}r_s$ (r_s est le rayon de captage des capteurs) est considérée comme la distance optimale entre trois capteurs qui assurent la couverture. Le robot se dirige vers les six directions selon l'ordre, d_1 puis d_2, \dots, d_6 , si la direction suivante est couverte, il continue dans la même direction (voir Figure 2.8(a)). Selon Fletcher. G, (2010), cet algorithme ne supporte pas plusieurs robots, et il se termine avant qu'il assure la couverture complète du champ de captage, comme il ne résiste pas aux pannes. En effets, on peut déduire que cet algorithme et dans le cadre du traitement de pannes peut ne pas assurer la couverture dynamique du réseau.

C) Trajectoire sous forme d'une grille: Cette trajectoire est adoptée par Xu. Li et *al* (2012). Le but de cette solution d'exploration est le déploiement utilisant un seul

robot, mais elle peut être utilisée pour le redéploiement (assurer la couverture dynamique). Le robot se déplace en quatre directions Nord, Sud, Est, Ouest. Après chaque mouvement, ce robot diffuse localement des messages beacon pour informer le capteur le plus proche de sa présence. Le capteur à son tour, propose au robot la direction la moins récemment visitée.

La Figure 2.9 montre l'architecture de communication avec balise dans cette approche:

La chaîne T, C, E représente l'état de la direction où T: dénote si la direction est *open* ou *explored*, C: compte le temps à partir de la dernière exploration si T est dans l'état *explored*, E: est un extra champ de la propagation de l'information du réseau.

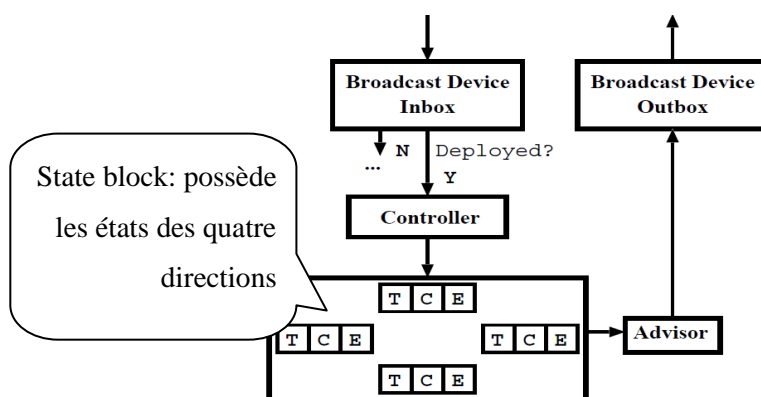


Figure 2.9: Architecture de communication balise dans l'approche Xu.Li. et *al*(2012).

L'Advisor (conseiller) calcule la meilleure position pour le robot s'il est dans le rayon de communication d'un capteur.

Toutes les directions *open* sont plus prioritaires à suivre que celles *explored* avec la valeur de C qui est la plus grande. Les balises de recommandation sont générées par le conseiller et envoyées par l'outil de diffusion vers les robots les plus proches. L'avantage de cette approche est que le robot choisi la direction proposée par les capteurs afin de se diriger vers la direction la moins visitées. Mais, dans le cas de l'utilisation de plusieurs robots, une direction (exemple le nord d'un capteur peut être le sud d'un autre capteur) est reconnait seulement par le capteur. Le capteur connaît la situation locale de ses directions, si un robot visite le nœud voisin il ne sait pas que cette direction a été déjà visité par un autre robot et le C de cette direction reste toujours le plus grand. Si un robot passe au niveau d'un capteur, il lui propose cette direction qui a été déjà visitée par un autre robot.

Batalin et Sukhatme ont proposé l'algorithme LRV (Least Recently Visited) qui dirige le robot dans son déplacement pour observer l'environnement. Il aborde le problème de couverture dynamique avec un seul robot (Maxim. A et *al*, 2002). Le robot se déplace et s'arrête selon une grille (voir Figure 2.10) au niveau de chaque point de la grille selon Xu. Li et *al* (2012), la longueur d'arrêt de la grille doit permettre la découverte des pannes, par exemple: elle ne dépasse pas le rayon de communication des capteurs). A ce niveau, il informe les capteurs voisins de sa

présence, chaque capteur répond par la transmission de la dernière date de visite du robot de son point de grille adjacent. Le robot choisit toujours la direction dont la date de visite est la plus petite. A chaque fois que le point de la grille est visité par le robot, le capteur adjacent de ce point incrémente le nombre de visites de la direction choisie. Selon Fletcher. G, (2010), le problème de LRV est qu'il ne termine pas et il ne supporte pas le cas de plusieurs robots.

Pour que l'algorithme d'exploration de LRV s'adapte avec le cas de l'utilisation de plusieurs robots dont le but est l'accélération du processus d'exploration, les auteurs de GRID_R3S2 (Xu. Li et al, 2012) ont proposé l'algorithme d'exploration LFV (Least Frequency Visited). Chaque robot maintient localement une suite croissante (seqno), qui sera utilisée conjointement avec son ID pour définir une visite distincte à un point de grille. Il incrémente ce numéro à chaque fois qu'il change son point de résidence (son dernier point de la grille déjà visité). Lors de la phase de découverte (l'arrêt), un robot diffuse un message de balise portant son (identité, SEQNO, et son emplacement actuel dans la grille: point de résidence). Les capteurs actifs adjacents à ce point de la grille déterminent si c'est une nouvelle visite ou non en comparant (l'ID robot, SEQNO) à celui déjà enregistré. S'ils sont différents,

- ils gardent le nouveau ID robot, SEQNO,
- ils incrémentent le nombre de visites de ce point adjacent, et
- ils proposent au robot la direction qui possède le nombre de visites le plus petit.

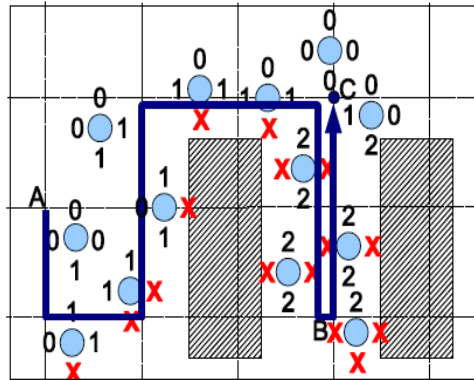


Figure 2.10: Politique de traversée utilisant LRV d'après Selon Fletcher. G, (2010).

Fletcher. G, (2010) propose l'algorithme BTM (Back-Tracking Deployment Algorithm). Le robot se déplace en quatre directions (Nord, Sud, Est, Ouest). Il déploie les capteurs un par un. Chaque capteur va avoir la couleur blanche s'il possède un voisinage vide, si non il prend la couleur noire. En plus, il enregistre à son niveau deux pointeurs (forward pointer, back pointer), le premier contient la position du successeur et le deuxième contient la position du prédécesseur. Le robot attribue à chaque capteur un numéro de séquence incrémenté à chaque placement d'un nouveau capteur jusqu'à son arrivée à un capteur "noir" (qui n'a pas de voisinage vide).

Suivant le back pointer, ce robot arrive jusqu'au premier nœud blanc (qui possède un voisinage vide) pour qu'il commence un nouveau déploiement.

Le but de l'exploration dans BTM est le déploiement. L'algorithme se termine quand tous les capteurs deviennent noirs. Concernant la découverte des pannes, l'algorithme doit se répéter sans arrêt. Dans ce cas, la couleur noire est remplacée par un numéro de tour. Quand un robot arrive à la situation où il n'existe pas un autre nœud non découvert dans une tournée, il incrémente ce numéro de tour et il recommence une nouvelle exploration. BTM supporte aussi le cas de l'utilisation de plusieurs robots, en ajoutant au numéro de séquence l'identifiant du robot.

Les approches de Elmaliach. Y, (2007), Chang. C et *al* (2007), Batalin. A, (2003) et Bautina. A et *al* (2011), utilisent aussi l'exploration.

- **Comparaison entre les algorithmes déjà vus**

Fletcher. G, (2010) a montré que SLD ne peut pas garantir une couverture complète avec un scénario très simple, contenant un seul obstacle. Cela implique que malgré l'utilisation d'un mécanisme de répétition, cet algorithme n'accède pas à toute la zone et il se bloque. Ils ont montré que LRV (Maxim A et *al*, 2005) n'est pas une solution pratique, car elle nécessite un nombre important de mouvements et de messages perdus. En outre, la solution ne supporte pas plusieurs robots, ce qui maximise le temps de découverte des pannes.

Selon Fletcher. G, (2010), BTM assure la couverture complète et une résistance aux obstacles et aux pannes concernant le déploiement mais il lui faut des modifications pour s'adapter à l'exploration pour la détection des pannes isolées.

3.4.1.3 Synthèse

Quand les capteurs envoient les rapports vers un gestionnaire, le protocole de routage utilisé possède une importance remarquable concernant le délai de parcours des messages et les transmissions multi sauts nécessaires. Mais, malgré les améliorations effectuées sur le routage, la transmission multi sauts nécessite la participation des capteurs qui possèdent déjà des ressources limitées et on peut avoir un overhead en cas de gestionnaire (sink, robot manager) fixe et si le gestionnaire est mobile le problème de routage devient un problème de routage vers une ou plusieurs cibles mobiles et cela nécessite un nombre supplémentaire de messages. L'avantage de ce mode est que les rapports seront envoyés dès qu'une panne intervienne, et cela permet la détection immédiate des pannes et une réparation dans un délai réduit et limité (dans la plupart des cas, le robot le plus proche est choisi pour la réparation). On remarque qu'à chaque fois que le taux de pannes est bas, les délais de réparations des pannes sont réduits, et même le nombre de messages de détection nécessaires au niveau du réseau.

Contrairement au premier mode, les robots se déplacent pour découvrir les pannes. A chaque phase d'arrêt, il(s) interroge(nt) les capteurs en effectuant des communications locales, alors le nombre de messages nécessaires pour que le robot détecte une panne dépend du nombre d'arrêts effectués avant la découverte d'une panne. Le temps de détection d'une panne augmente avec le chemin parcouru avant la découverte d'une panne et cela a une relation directe avec le temps d'exploration (si

l'algorithme d'exploration n'assure pas la couverture complète, le délai de réparation peut être maximale). L'amélioration des qualités de ces solutions dépend de l'algorithme d'exploration et le nombre d'arrêts.

La comparaison qualitatif entre ces sous classes, concernant le nombre de messages et le temps de réparation, nécessite une évaluation. Comme premières remarques, on peut dire que le temps d'information de robot dans le mode sans exploration est minimum par rapport au mode sans exploration. Par contre, le nombre de messages transmis dans le premier mode est très élevé par rapport au mode avec exploration. Le temps de réparation dépend du temps d'information du robot et la stratégie de réparation. Ces différents aspects seront étudiés plus loin dans ce Chapitre.

De ce qui précède, on peut déduire la Table 2.2.

| | Nombre de messages | Temps maximal de réparation d'une panne | Influence de la scalabilité |
|-------------------------|---------------------------|--|------------------------------------|
| Sans exploration | plus | moins | plus |
| Avec exploration | moins | plus | moins |

Table 2.2: Comparaison des modes sans exploration et avec exploration.

3.4.2 Le(s) sink(s) évalue(nt) les pannes

Dans ce mode, le sink exploite les données d'application récoltées et transmises par les capteurs pour détecter l'existence des pannes et définir leurs emplacements (ou leurs identifiants). Après la détection, le(s) sink(s) donne un ordre au(x) robot(s) pour réparer la panne, dans ce cas soit les robots prennent la responsabilité de calculer la stratégie de réparation ou le sink attribut et divise les tâches entre les robots, et ces derniers se déplacent suivant la stratégie calculée.

Une question peut se poser aussi pour ce mode, si le sink engage les robots disponibles à son niveau pour réparer les pannes, où le sink doit attendre la présence d'au moins un robot à son niveau, et cela augmente le temps de réparation des pannes. Une solution consiste à laisser les robots restent dans leurs zones de captage, et le sink transmettra les ordres via un protocole de routage, et cela nécessite la connaissance des informations sur les robots réparateurs et des transmissions multi sauts qui augmentent le taux de messages nécessaires et génère un overhead au niveau du sink.

Le but de l'approche de Hao. W et *al* (2012) (Robot Assisted Maintenance Strategy in Wireless Sensor Networks) avec un seul robot (RAMS) est le redéploiement pour la réparation des trous de couverture et de la connectivité: Ils considèrent que $R_s=2 \cdot R_c$ et les nœuds à faible énergie. Dans cette approche, chaque nœuds reporte les informations captées à la station de base dans une période fixe, cela permet à la station de base de connaître les facteurs de santé de ce nœud y compris son énergie résiduelle et sa position, utilisant une fonction de coup (cost). La station de base calcule l'énergie résiduelle et le taux de couverture du réseau et calcule les positions adéquates pour le remplacement de nouveaux capteurs. La liste des positions

candidates sera transmise vers le robot. Ce dernier se déplace vers les locations déterminées et déploie de nouveaux nœuds utilisant le chemin de localisation optimal des nouveaux nœuds, qui devient un problème NP-hard (i.e. Chercher le chemin optimal au robot pour réparer la liste des pannes).

Les auteurs de cette approche (Hao. W et *al* (2012) proposent une méthode de détection qui se base sur la modélisation probabiliste de la couverture. Cette méthode assure la connectivité et la couverture au même temps car les positions entre les capteurs doivent vérifier le critère suivant ($R_s=2*R_c$).

Dans cette approche, le traitement centralisé des demandes permet d'avoir une vue générale de l'état du réseau et de proposer les meilleurs solutions. Les auteurs proposent une modélisation de couverture avec la prise en compte des différents paramètres par exemple, le bruit au niveau de chaque nœud, et un poids pour chaque nœuds qui prend en considération plusieurs paramètres (la distance, l'énergie consommée pour chaque opération, ...), ceci permet une évaluation réelle du nombre de situations de trous au niveau du réseau. L'évaluation du taux de couverture avec la prise en considération du niveau d'énergie, permet d'éviter un certain nombre de pannes avant qu'elles interviennent. La méthode proposée ne sélectionne pas les nœuds qui sont déjà déployés pour les remplacer mais elle cherche la position (les coordonnées X, Y) la plus adéquate pour le placement d'un nouveau nœud. Cette méthode engendre une minimisation remarquable du taux de communication nécessaire pour la détection et la réparation. Aucune communication entre les capteurs et le robot n'est nécessaire. En plus, les informations nécessaires pour la détection et sont envoyés périodiquement par les informations de l'application, alors cette solution n'ajoute pas de charges supplémentaires sur l'application.

Inconvénients :

- La définition de la période de maintenance est considérée comme un défit? Selon Xu. Li et *al* (2012), si cette période est très courte, cela permet la découverte de nouveaux trous de couverture mais le coup de maintenance est très élevé, surtout, si le nombre de nouveaux capteurs est limité (capacité du réservoir limitée). Si elle est très longue, cela augmente énormément le temps de réparation d'une panne, ce qui engendre des conséquences catastrophiques. Une proposition stipule que la période de maintenance soit définie par l'application et sera inférieure au temps maximal de réparation = temps maximal d'absence d'information pour une zone.
- La maintenance ne commence qu'après une période de déploiement du réseau, et si une panne intervient dans cette période, elle doit attendre toute cette période pour être traitée, et cela peut augmenter énormément la période de réparation.
- La solution cherche à améliorer le taux moyen de couverture, mais elle ne permet pas une distribution homogène de la couverture. C'est-à-dire si une panne intervient et cette panne provoque un trou de couverture, l'utilisation de la méthode "particle swarm optimal method" permet une minimisation du nombre de la population à chercher. Si après 30 itérations et avant que les positions de trous et plus proches trous soient redéployés, on peut avoir le cas où le coup moyen sera amélioré, le RAMS termine et le trou qui peut être plus grave peut ne pas être traité.

- Si les capteurs envoient après chaque période les informations d'application, on peut avoir un overhead au niveau de la station de base. En plus, dans le mode de communication *event driven*, les nœuds n'envoient ces informations que dans le cas où un événement intervient et l'envoi des données après une période donnée ajoute un poids supplémentaire pour ces nœuds.

3.4.3 Le (s) robot(s) évalue(nt) les pannes

Dans ce mode, le robot fait un mouvement qui lui permet de contrôler chaque point observable de la zone et de calculer le taux de couverture et de connectivité ou de contrôler tous les nœuds du réseau. Par l'écoute des messages "hello" et le calcul des distances séparant le robot et les capteurs, le robot peut savoir la situation globale du réseau.

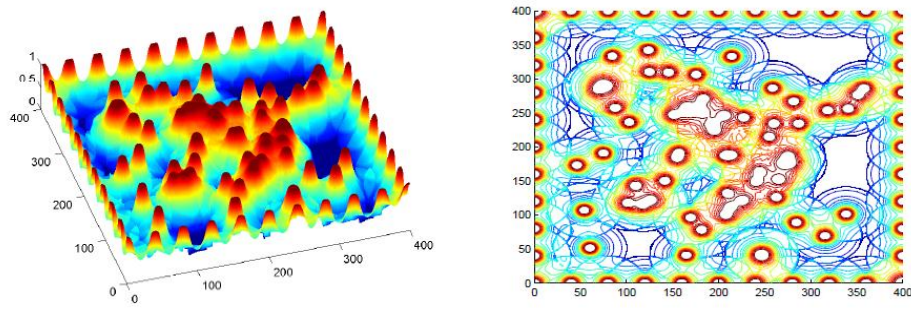
Il est à remarquer que ce mode ne nécessite pas de communications directes entre les capteurs et les robots, mais les capteurs doivent prouver leurs existences par diffusion périodique des messages "hello", par exemple. La réparation dans ce mode peut se faire :

- après la phase de découverte de toute la zone
- en cours de l'exploitation

A) Après la phase de découverte

a) Un seul robot

Solution de Pengju. Si et al (2012): L'objectif de l'approche de Pengju Si (*A Hole Detection and Redeployment Strategy in Wireless Sensor Network*) est le redéploiement de nouveaux nœuds afin de réparer la couverture. Elle se base sur le modèle probabiliste de couverture avec le taux de fausses alarmes (*the probabilistic detection model with false alarm rate*), comme l'approche précédente. Les capteurs sont distribués d'une manière aléatoire et uniforme, ils ont la possibilité de reconnaître leurs positions géographiques. Un seul robot (*Unmanned Aerial Vehicle, UAV*) est utilisé pour la détection de la surface et l'emplacement des trous et le remplacement de nouveaux nœuds au niveau de ces trous. Le robot commence son traitement par le parcours de toute la zone (exploration) afin de déterminer le taux de couverture de cette dernière (i.e. écoute les balises transmises et définit la distance entre lui et les capteurs). Ensuite, il calcule la carte appelée *iso_sensing* (voir Figure 2.11 pour un exemple) de la probabilité de détection et la carte de contour de la probabilité commune.



(a) Joint detection probability iso-sensing map (b) Joint detection probability contour map

Figure 2.11: Exemples de cartes iso-sensing et leurs contours d'après Pengju. Si et al (2012).

- Le robot explore la zone de déploiement où à chaque point de son graphe d'exploration et selon les messages beacon transmis par les capteurs (l'énergie de réception de chaque signal), il peut calculer la distance qui sépare le point d'exploration par rapport aux autres capteurs. Et, selon une fonction de couverture et la distance calculée, il peut déterminer la probabilité de détection d'un objet qui se trouve au niveau du point d'exploration par l'ensemble des capteurs.

PS est le seuil minimum de la probabilité de détection commune. Les points qui ont la probabilité de détection inférieure à P_s seront considérés comme des trous où le robot doit remplacer de nouveau nœuds.

$$C(P_j) = \begin{cases} C(P_j) & C(P_j) \geq P_s \\ 0 & C(P_j) \leq P_s \end{cases}$$

- Traitement d'image: Suivant un processus binaire, l'image de la carte de contour de probabilité de détection commune sera transformée en deux dimensions et en deux couleurs (blanc et noir).
- Définition des trous : Après l'élimination de bruits et des structures sans rapport, les formes des trous seront extraites utilisant la méthode de la boucle ouverte et la boucle fermée.

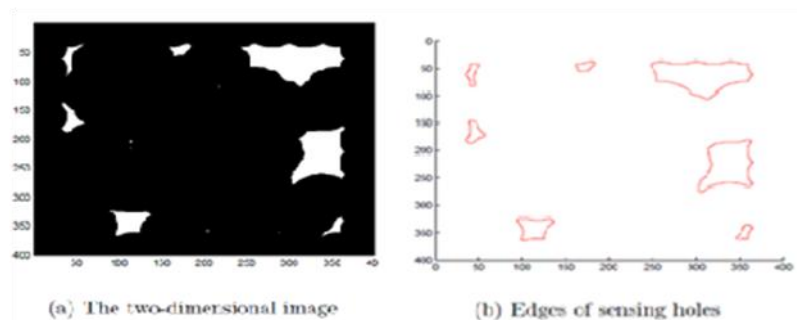


Figure 2.12: Exemple de trous de couverture extraits de Pengju. Si et al (2012).

Pour la réparation des trous (Voir exemple dans la Figure 2.12), le robot calcule le chemin de parcours pour la réparation de la panne le plus court en utilisant l'algorithme 1-TSP-SELPD (Xu. Li et al 2010).

Synthèse

Avantages

- Dans cette approche, il n'existe pas de communications directes entre les capteurs et le robot, car les capteurs transmettent périodiquement des beacons et le robot écoute seulement à ces beacons et évalue la couverture. Cela minimise le coup de consommation d'énergie de la solution
- Le robot s'occupe de tout le traitement et la recherche et cela minimise les charges supplémentaires sur le réseau.
- La solution propose une modélisation de couverture avec la prise en compte des différents paramètres, par exemple le bruit au niveau de chaque nœud, et donne un poids pour chaque nœud qui prend en considération plusieurs paramètres (la distance, l'énergie consommée pour chaque opération, ...) qui permettent une évaluation réelle du nombre de situation de trous au niveau du réseau.
- Cette méthode ne sélectionne pas les nœuds qui sont déjà déployés pour les remplacer mais elle cherche les positions (les coordonnées X, Y) les plus adéquats pour le placement d'un nouveau nœud.

Inconvénient

- Le temps de réparation de trous peut être très grand, il paraît que cette solution n'est pas scalable. Tant que la zone est grande, le robot doit parcourir toute cette zone pour qu'il puisse commencer le calcul de probabilités et cela augmente le temps de réparation et tant que la zone est grande le temps de réparation va augmenter.

Solution de Yao. S et al (2009): La solution proposée par Yao. S et al (2009) (*Holes Detection in Wireless Sensor Network and Redeploy Route Planning Based on Unmanned Aerial Vehicle*) possède le même but que l'approche précédente et le même principe général de détection et de réparation de couverture. Elle diffère dans le modèle de calcul de probabilité et dans la proposition de méthodes graphiques de la détection des trous de couverture, plus précisément dans la définition de la meilleure position de déploiement d'un nouveau nœud. Concernant la réparation, ce problème est transformé en un problème de recherche d'un plan de route pour UAV (le robot pour le redéploiement).

- **Evaluation de la couverture**

Connaissant le taux de couverture dans chaque point, cette solution peut définir la couverture du réseau. Après, il s'agit de calculer le taux de couverture de chaque nœuds, la valeur minimale aux dessus de laquelle un point ne sera pas capté par un capteur est définie par l'expression suivante:

$$C_{\text{node}}(s) = \min[CP(p)] \quad \forall p \in R(s)$$

P est un point aléatoire dans la zone de couverture de nœud s.

- **La détection des trous moyennant les méthodes graphiques**

Définition des trous (voir Figure 2.13): Supposons qu'il existe n nœuds, A_i est la section de la zone qui peut être captée par le nœud i . La zone de trou de couverture causée par la panne du nœud i est CH, elle peut être décrite comme:

$$CH = A_i - \sum_{j=0}^n A_i \cap A_j$$

La zone de détection est de deux types: zone couverte ou zone non couverte.

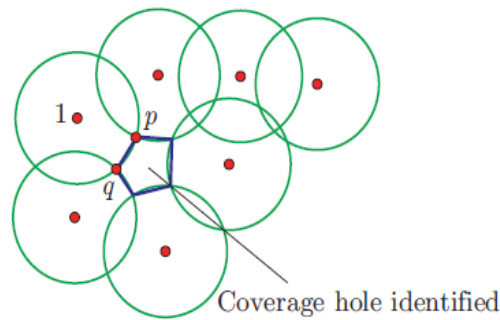


Figure 2.13: Identification de trou de couverture d'après Yao. S et *al* (2009).

- **L'algorithme de localisation des trous**

La position du nouveau nœud doit couvrir le trou le plus possible, de ce fait on doit déployer un nouveau nœud dans le centre du trou. Pour calculer ces positions, il existe deux méthodes.

- **Méthode 1: Basée sur le résultat de détection des trous graphiques** (le même principe de l'approche précédente, Figure 2.14).

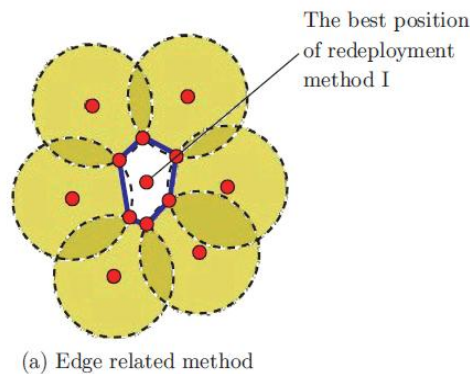


Figure 2.14: Position de redéploiement d'un nouveau capteur d'après Yao. S et *al* (2009).

- **Méthode 2: Basée sur l'auto positionnement des nœuds du RCSF**

Si les capteurs peuvent se localiser, il sera facile de savoir la position du trou de couverture à partir des coordonnées des nœuds reliant le trou (voir Figure 2.15).

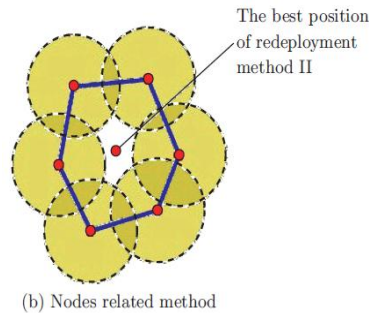


Figure 2.15: Position du trou à l'aide des nœuds de bordure d'après Yao. S et *al* (2009).

Cette formule est plus générale et facile à être appliquée pour localiser les trous. En utilisant le GPS, il est commode de savoir la latitude et la longitude du trou de couverture utilisant le module GPS.

Avantages

- Cette solution propose un positionnement optimal pour la réparation de trous de couverture.
- Le traitement centralisé calculé par UAV (gestionnaire mobile) permet d'éviter l'overhead au niveau du gestionnaire qui est un inconvénient majeur de traitement centralisé et permet d'avoir une vue générale sur le réseau avec un coup de communication négligeable.
- La méthode de réparation permet le traitement d'un ensemble des trous à la fois et d'une façon optimale et cela minimise le temps de réparation et les coûts en général.

Inconvénients

- Le temps de détection de la panne dépend de la surface parcourue par le robot, car le robot attend la terminaison de sa tournée et son traitement des positions optimales puis il commence la réparation.

B) En cours d'exploration

Dans ce mode, quand un robot détecte un trou ou une panne, il intervient directement pour réparer cette panne.

L'avantage de ce mode par rapport à la détection par les capteurs et avec exploration des robots, les capteurs ne transmettent pas de messages vers le robot et cela minimise les messages nécessaires, mais les arrêts lors de détection doivent être suffisants pour assurer la couverture dynamique de la zone.

C) Comparaison entre les modes en cours et après exploration

| | Nombre des messages | Temps de réparation |
|-------------------------------|----------------------|---------------------|
| Après l'exploration | Diffusion périodique | plus |
| Au cours d'exploration | Diffusion périodique | moins |

Table 2.3 : Comparaison entre modes en cours et après exploration.

4. Modèles de réparation

On appelle réparation, la phase où le robot est informé de(s) panne(s) et commence le redéploiement. Cette phase représente la dernière phase de traitement des pannes. Dans cette phase, on veut répondre à la question suivante: Si un robot est chargé de réparer un ensemble de pannes, quelle sera la meilleure stratégie? Parmi les critères qui peuvent être exigés, on cite:

- le délai de réparation moyen doit être minimal,
- le temps maximal d'attente de réparation doit être minimal,
- l'ordre d'apparition des pannes doit être respecté.

4.1 Liste FIFO (First In First Out)

Dans ce cas, le robot possède une liste de pannes ordonnée généralement selon l'ordre d'apparition, où il se déplace d'une position à une autre et il lance un nouveau capteur. Ce mode de réparation a été utilisé dans l'approche Mei. Y et al, (2007). L'avantage de ce mode est qu'il respecte l'ordre d'apparition de pannes, mais le robot fait beaucoup de mouvements lorsqu'il redéploie les capteurs, ce qui augmente le temps de réparation moyen. Ce mode peut être efficace si la zone où se trouve l'ensemble de pannes est de petite taille (traitement distribué, ou localisé).

4.2 Recherche du plus court chemin

Un exemple de solution qui utilise cette stratégie est l'approche de réparation centralisée "*One commodity travelling salesman problem with selective pickup and delivery (1-TSP-SELPD)*" utilisée dans l'algorithme Yao. S et al (2009) et Pengju Si et al (2012):

- **Généralités sur l'approche de Xu. Li et al (2010)**

- Fletcher. G, (2010) modélise la relocation des capteurs en utilisant un robot comme un nouveau problème d'optimisation combinatoire, nommé "*one commodity travelling salesman problem with selective pickup and delivery (1-TSP-SELPD)*".
- Le problème est le suivant : Commencant à partir de la station de base, nous voulons trouver le tour le plus court de tel sorte que toutes les demandes de livraison soient traitées (tous les trous de captage sont réparés). Toujours les endroits de partenaire rentable sont visités avant de retourner vers la station de base par la suite.
- La tournée ne doit inclure aucun nœud dupliqué (à part la station de base comme le premier et le dernier point).
- Ne pas permettre au robot de transporter plus que la limite prédéfinie en nombre de capteurs dans n'importe quel point dans la tournée.
- Dans la Figure 2.16, le nombre maximum de points (capteurs portés par un robot) sont égales à trois.

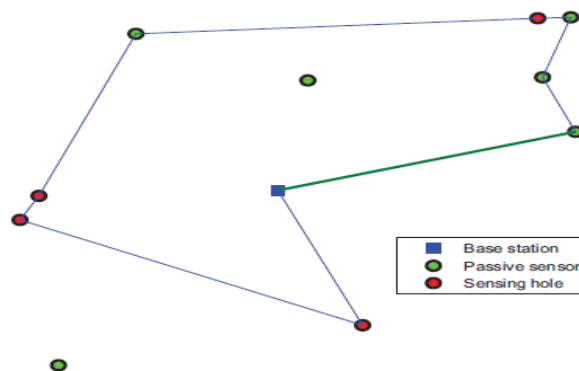


Figure 2.16: Exemple de mouvement d'un robot dans 1-TSP-SELPD avec le nombre maximum de points = 3.

1-VRP-SELPD (One Vehicle Routing Problem with selective pickup and delivery)

Le problème de tournées de véhicules (voir Figure 2.17) est une classe de problèmes de recherche opérationnelle et d'optimisation combinatoire. Il s'agit de déterminer les tournées d'un flot de véhicules afin de livrer une liste de clients ou de réaliser des tournées d'interventions (maintenance, réparation, contrôle) ou de visites (visites médicales, commerciales, etc.). Le but est de minimiser le coût de livraison des biens. Ce problème est une extension classique du problème du voyageur de commerce, et ainsi fait partie de la classe des problèmes NP-Comple¹.

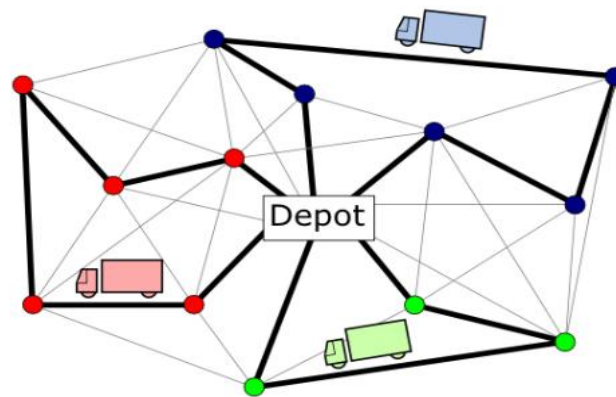


Figure 2.17 : Le tour d'un véhicule pour attendre tous les points¹.

- Il possède le même principe de calcul mais, dans ce cas, la station de base choisie N tours pour N véhicules où les nœuds de réseau seront visités contrairement à 1-TSP-SELPD (Yao. S et *al*, 2012).
- Inconvénient de cette approche: Il est dû à la nature centralisée de l'algorithme et les mises à jour nécessaires fréquentes, pour permettre à la station de base d'avoir une vue correcte sur le nombre de nœuds redondants, et les trous de captage. Il est coûteux en termes de coût en messages et en bande passante

5. Discussion

D'après les solutions décrites dans ce Chapitre, la stratégie et les phases de traitement de pannes diffèrent généralement selon le mode de détection dont on distingue trois classes : le cas où le robot détecte les pannes, le sink détecte les pannes, les capteurs détectent les pannes.

Dans le premier mode, le robot explore la zone de couverture et détecte les trous et les répare soit en cours de son exploration où il va gagner du temps mais il n'aura pas une vision globale des pannes, soit après l'exploration où il explore toute la zone puis il commence la réparation, où il va perdre du temps mais il peut avoir une vision générale des pannes, qui lui permet de prendre les meilleures décisions. Dans ce cas, la phase d'information du robot n'est pas nécessaire. Cette stratégie de détection est moins coûteuse en termes de charges supplémentaires sur les capteurs, Mais, le temps de réparation peut être très grand, la coopération entre l'ensemble des robots peut régler le problème, mais elle est difficile dans ce modèle.

Concernant la détection par le sink, où le traitement des pannes possède une nature centralisée, le sink et selon les informations reçues périodiquement à partir des capteurs, peut déterminer les pannes de ces derniers. L'avantage de cette stratégie est le nombre de messages minimisé, si la période de transmission des informations est grande. Mais, ce nombre s'agrandit aussi si la taille de la zone est grande, Cependant, le temps de détection peut être grand, ce qui influe sur le temps de réparation.

Concernant la détection par les capteurs, il existe deux modes avec exploration et sans exploration. Dans le premier, le nombre de messages est important mais le

temps de réparation est moyen. Dans le second, le temps de traitement peut être grand mais, le nombre de messages est minimisé. On a remarqué que la plupart des traitements effectués dans le mode détection par les capteurs sont distribués ou localisés, ce qui permet un temps de réparation faible.

Si on compare les trois stratégies, l'utilisation de l'exploration pour la découverte des pannes peut être un bon moyen pour minimiser le nombre de messages transmis par les capteurs. En échange, le temps de réparation reste un défi dans ce cas. Par contre, l'envoi de messages pour informer les robots est une stratégie qui assure une rapidité d'information des robots, surtout lorsque le traitement est distribué ou localisé. Par contre, ce mécanisme ajoute une charge supplémentaire au capteurs en termes d'énergie liés aux transmissions.

Concernant les modèles de réparation, lorsqu'un robot suit la liste Fifo pour réparer les pannes, il va respecter l'ordre d'apparition des pannes, mais on peut avoir des mouvements inutiles. Par contre, s'il cherche le chemin le plus court pour réparer l'ensemble des pannes, il peut minimiser le temps de réparation moyen, mais il pénalise l'ordre de réparation des pannes. Par conséquent, le temps maximal de réparation peut être grand.

6. Conclusion

Le traitement des pannes isolées utilisant les robots est un problème qui dépend des stratégies de détection et de réparations des pannes. Quand les capteurs s'occupent de la phase de détection, cela représente une charge supplémentaire sur les capteurs mais assure une détection dans des délais raisonnables. Si le robot s'occupe de tout le traitement dépendamment de la stratégie d'exploration on peut avoir des délais de réparation illimités mais moins de charge sur les robots.

Après la détection, et selon les approches vues, une bonne affectation des tâches entre les robots et un cours chemin de réparation et le respect de l'ordre de réparation, influent sur le temps de traitement de la panne et la messagerie nécessaire.

Chapitre 3:

Directing Robots to Failures using Lists (DRFL)

1. Introduction

Le rôle principal d'un réseau de capteurs est l'observation de sa zone de déploiement. Malheureusement, les capteurs peuvent être victimes de plusieurs types de pannes, dues à l'environnement et à leurs caractéristiques physiques sensibles. Les pannes influent sur le comportement d'un réseau, ce qui génère un manque de couverture et de connectivité. Selon la taille de la zone infectée, les pannes sont classées en deux types : les pannes massives et les pannes isolées. On dit qu'une panne est massive, si la réparation d'un trou généré par cette panne nécessite le déploiement de plusieurs nœuds. La cause de ce type de panne peut être un bombardement, un feu ... etc. Par contre, la réparation de trou de la panne isolée (individuelle) nécessite seulement un seul capteur, la cause de cette panne est généralement l'épuisement de l'énergie du capteur ou la panne matérielle. Dans notre recherche, on est intéressé au traitement des pannes isolées.

Les pannes isolées peuvent intervenir à n'importe quel moment, et à n'importe quel emplacement. Au bout d'un certain temps, leur nombre devient plus important. Par conséquent, le traitement des pannes devient vital pour allonger la durée de vie du réseau. Mais, dans un environnement dur ou inaccessible, il est primordial d'utiliser des mécanismes de réparation des pannes. Parmi les solutions vues dans la littérature, on distingue la réparation utilisant les capteurs mobiles, où les capteurs redondants se déplacent vers les emplacements des trous afin de les réparer [60]. Malheureusement, ces solutions sont coûteuses, et si le réseau ne contient pas des capteurs redondants, la réparation termine.

Grace aux recherches scientifiques très avancées en robotique, les robots sont utilisés dans plusieurs domaines. Par conséquent, des robots qui possèdent les capacités physiques et techniques pour accéder à des lieux très durs sont produits. Donc, plusieurs chercheurs ont pensé à l'utilisation des robots au traitement des pannes isolées. Le Chapitre précédent a été dédié à la présentation et la classification de quelques approches de traitement des pannes isolées utilisant les robots. Notre recherche est concentrée sur les solutions de redéploiement des nouveaux capteurs utilisant les robots.

En général, et d'après les solutions vues dans la littérature, le redéploiement des capteurs utilisant les robots suit le scénario suivant : la détection des emplacements des pannes, l'information de(s) robot(s) des pannes, la réparation : le(s) robot(s) se déplace(nt) vers ces emplacements des pannes et place(nt) un nouveau

nœud fonctionnel au niveau de chacune. Dans chaque étape, plusieurs techniques sont utilisées dès la détection jusqu'à la réparation afin d'améliorer le temps de traitement des pannes et minimiser le coup en terme d'énergie consommée par les capteurs. Il existe trois modes de détection: la détection par les robots, par le sink, ou par les capteurs. Dans la détection par les robots, les robots s'occupent de la détection et la réparation des pannes (Sheo. J-P et *al* (2007)). L'avantage de cette stratégie est le non ajout de charges supplémentaires aux capteurs (moins de communication), mais le temps de réparation est grand et s'agrandit de plus à chaque fois que la zone est grande. Par contre, dans la détection par le sink (traitement centralisé), dans le cas de l'approche de (Hao. W et *al* (2012)), les capteurs doivent après une certaine période, envoyer des messages d'information qui indiquent son niveau d'énergie actuel. Le sink utilise les données récoltées et calcule les emplacements de toutes les pannes existantes dans la zone de captage et désigne un robot pour réparer l'ensemble des pannes. Le temps de réparation et le nombre de messages générés dépendent d'une manière directe de la période d'envoi des messages d'information et de la taille de la zone. Si cette période est petite, le nombre de messages augmente. Si elle est grande, la période de réparation d'une panne s'agrandie.

Concernant la détection utilisant les capteurs (Mey. Y et *al* 2007),les capteurs détectent localement la panne. Après la détection, soit ils attendent l'arrivée de robot qui est en exploration de la zone, soit ils envoient leurs rapports de panne vers un gestionnaire. L'envoi des rapports de pannes vers un gestionnaire fixe ou mobile (, exemple de l'approche basée cluster (Mey. Y et *al* 2007) ; nécessite beaucoup de transmissions multi sauts, ce qui augmente le nombre de messages mais le robot est informé dans un délai acceptable (le temps de réparation est minimisé).

Quand les capteurs attendent l'arrivée de robot, qui est généralement en exploration pour l'informer de l'état local des pannes, l'information du robot nécessite moins de transmissions multi sauts donc moins de messages, mais le temps de réparation des pannes dépend d'une manière générale de l'algorithme d'exploration utilisé.

Dans notre contribution, on a choisi l'utilisation de la détection par les capteurs, et que ces derniers attendent l'arrivée de robot pour l'informer de l'état local d'une sous zone. Certains capteurs d'une sous zone communiquent avec les robots qui visitent cette sous zone. Ces capteurs et selon ces informations locales guident les robots vers les emplacements les plus probables de la panne. Si un capteur détecte que la durée d'attente de réparation d'une panne dépasse un certain seuil, il lance des messages d'urgence et attire les robots à proximité vers ces pannes. Cette stratégie nous permet de minimiser le nombre de messages transmis, l'énergie consommée des capteurs et le temps de traitement d'une panne.

Concernant l'ordre de réparation, en général il existe deux types (voir le Chapitre précédent): La réparation par une liste FIFO (First in First Out) , où le robot répare les pannes selon l'ordre d'intervention de chaque panne comme il est le cas de l'approche basée cluster (Mey. Y et *al* 2007) Dans l'autre type, l'ordre de réparation devient la solution d'un problème combinatoire comme l'exemple de (1-TSP-SELPD) (Xu. LI et *al* 2010). Dans notre solution, on a choisi l'utilisation des listes FIFO, afin de conserver l'ordre de réparation des pannes.

Ce chapitre est dédié à la présentation de notre approche de réparation. Il permet d'expliquer les différents aspects de la solution en commençant par la spécification de notre objectif, suivi par la description de notre environnement. Une description générale de la solution, structuration et rôles des capteurs, les apports de notre approche, les données manipulés et techniques de base sont aussi explicités. Une description détaillée de notre solution est aussi apportée. Ensuite, on explique les mécanismes de gestion de quelques cas d'exceptions. Le Chapitre termine par une discussion suivie par une conclusion.

2. Problématique et objectifs

Les capteurs sont des unités de traitement qui utilisent la batterie dans leurs fonctionnements. Une grande partie de leur énergie est consommée lors d'émissions ou de réception des messages. Ceci représente un grand déficit lorsqu'on conçoit une solution où chaque communication supplémentaire effectuée par les capteurs influe sur l'énergie résiduelle de ces derniers et peut provoquer leurs pannes. Autrement, lorsqu'une panne intervient, elle peut causer un manque de couverture ou de connectivité, les robots doivent alors réparer cette dernière le plus rapidement possible. Donc, quel est la stratégie qui nous assure un traitement des pannes moins coûteuse et dans un délai minimale?

Parmi les stratégies citées dans le Chapitre précédent, on a choisi celle où les capteurs détectent localement les pannes et attendent un des robots qui sont en exploration pour l'informer de l'ensemble des pannes pour qu'il puisse les réparer. L'exploration de la zone de couverture doit répondre à quelques critères. On cite par exemple : le non blocage et la non terminaison (il ne doit pas y avoir de situation où le robot n'arrive pas à continuer son mouvement), la couverture dynamique de toute la zone (les robots doivent être informés de toutes les pannes du réseau qui interviennent à n'importe quel moment et à n'importe quel emplacement), Le temps de réparation d'une panne ne doit pas être très grand (le temps entre la détection d'une panne et sa réparation). Dans cette situation, on pose la question : Comment les capteurs peuvent attirer les robots vers les emplacements des pannes? D'une autre manière, quelle est l'algorithme de redirection qui permet aux capteurs de diriger les robots vers les pannes?

3. Notre solution

Afin de traiter les pannes dans un délai acceptable et avec le moindre coup (en terme d'énergie), le scénario général choisi pour le traitement des pannes isolées est le suivant: Les capteurs et selon leurs états locaux détectent la panne d'un de leurs voisins. En parallèle, les robots explorent la zone de captage suivant une grille virtuelle pour découvrir les pannes. Lors de l'arrivée d'un robot à proximité des capteurs, ces capteurs l'informent des pannes existantes et le dirigent vers d'autres zones (les emplacements des pannes les plus probables). Chaque robot et après la réparation des pannes existantes continue son mouvement vers le point de la grille proposé par les capteurs. Ce scénario est le même que l'approche de relocation Grid

R3S2 (Xu, Li et *al* 2013) est une approche de relocation à base de l'exploration, dont le but est la découverte des nœuds redondants et le redéploiement de ces nœuds dans les endroits des trous. Nous nous sommes inspiré de leur principe d'exploration et de réparation de pannes.

Le problème de l'exploration est que les capteurs attendent l'arrivée d'un robot qui est en exploration pour réparer les pannes. Ce qui implique que le temps de détection d'une panne dépend directement du temps d'exploration d'une zone. Le temps d'exploration d'une zone dépend de l'algorithme d'exploration, du nombre de robots et de la taille de la zone de déploiement. Dans notre travail, on prévoit l'amélioration de la qualité de l'exploration afin d'attirer les robots vers les emplacements des pannes, et les pannes dont la durée d'attente de réparation dépasse un certain seuil deviennent prioritaires. En plus, on cherche aussi à minimiser le nombre des arrêts effectués par les robots en augmentant le pas de déplacement effectué par ces derniers.

Dans cette section, on présente en détail notre approche en commençant par une description de l'environnement de déploiement, une description générale de la solution, les nouveautés sont aussi explicitées. Ensuite, les différents mécanismes utilisés et les données manipulées sont spécifiés. L'algorithme général qui explique les différentes étapes de notre solution est finalement donné.

3.1 Description de l'environnement de déploiement

Nous considérons un champ de captage à 2D dont les frontières sont définies et exemptes des obstacles. Des capteurs statiques sont supposés être déjà déployés et d'une façon aléatoire et peuvent être denses dans un environnement hostile. Plusieurs événements peuvent être à l'origine de la panne individuelle de ces capteurs sensibles. Les robots se déplacent le long de cette zone pour réparer les pannes intervenant tout au long de la durée de vie du réseau. On suppose que les nœuds déployés dans le réseau (robots, capteurs) connaissent leurs positions.

Les capteurs sont considérés comme des unités physiquement sensibles aux pannes. Par contre, on suppose que les robots possèdent les capacités physiques d'être mobiles, ne tombent jamais en panne, équipés d'un réservoir qui porte un nombre suffisant de capteurs.

On suppose que les capteurs sont connus par leurs positions géographiques et un identifiant. Par contre, les robots sont connus par leurs identifiants seulement. Le rayon de communication des robots est beaucoup plus important que celui des capteurs. On suppose aussi que le rayon de communication des capteurs est supérieur à au moins deux fois leurs rayons de captage (afin d'assurer que, si le réseau est couvert alors il est connecté).

3.2 Description générale

Dans notre solution, les robots explorent successivement la zone de déploiement des capteurs suivant une grille virtuelle où ils se dirigent seulement vers les quatre directions : SUD, NORD, EST, OUEST. Au niveau de chaque point de la grille, ils communiquent avec les capteurs les plus proches, afin d'être informés de la situation

des pannes dans une sous zone et de la prochaine direction à suivre. La Figure 3.1 représente le modèle de la grille parcourue par les robots.

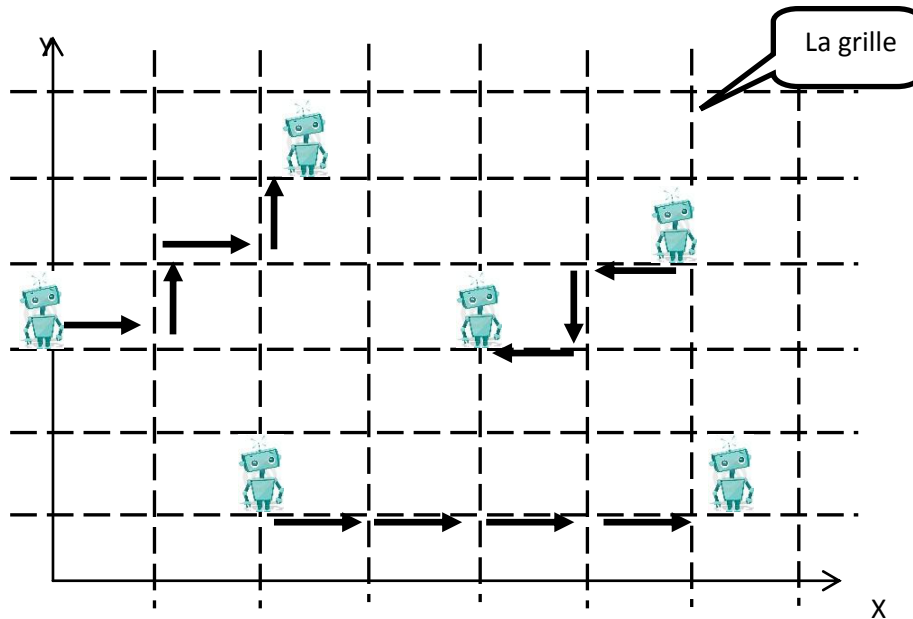


Figure 3.1: Modèle de déplacement des robots.

Un point de la grille est l'intersection de deux droites de la grille, où il représente la position d'arrêt d'un robot. Le rayon de communication de robot doit atteindre les points de la grille voisins de la position actuelle d'un robot afin de permettre aux capteurs à proximité de ces points de faire les mises à jour nécessaires afin de rediriger les autres robots vers les points de grille les moins récemment visités (voir Figure 3.2).

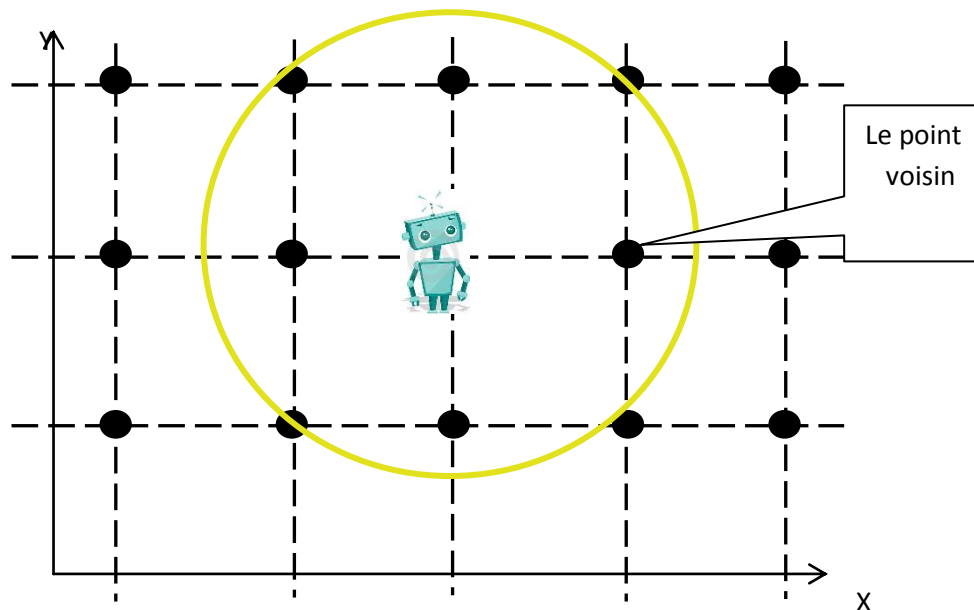


Figure 3.2: Rayon de communication d'un robot.

Comme on a montré dans la Figure 3.3, la zone de couverture est subdivisée en sous zones carrées selon la grille de déplacement des robots, le coté d'une sous zone est égale au pas de déplacement des robots. Chaque point de la grille devient le centre d'une sous zone. Chaque sous zone est représentée par son point de la grille qui représente l'emplacement où les capteurs doivent envoyer leurs rapports de pannes, et aussi la position d'arrêt d'un robot quand il visite cette sous zone. Le calcul de point de la grille est définie dans la section "Mécanismes de base".

La division de la zone de captage (déploiement) en sous zones est adoptée par d'autres approches déjà vues dans la littérature. La zone de déploiement dans l'algorithme distribué fixe de l'approche basée cluster est divisée en sous zones où le nombre de ces dernières est égale au nombre de robots.

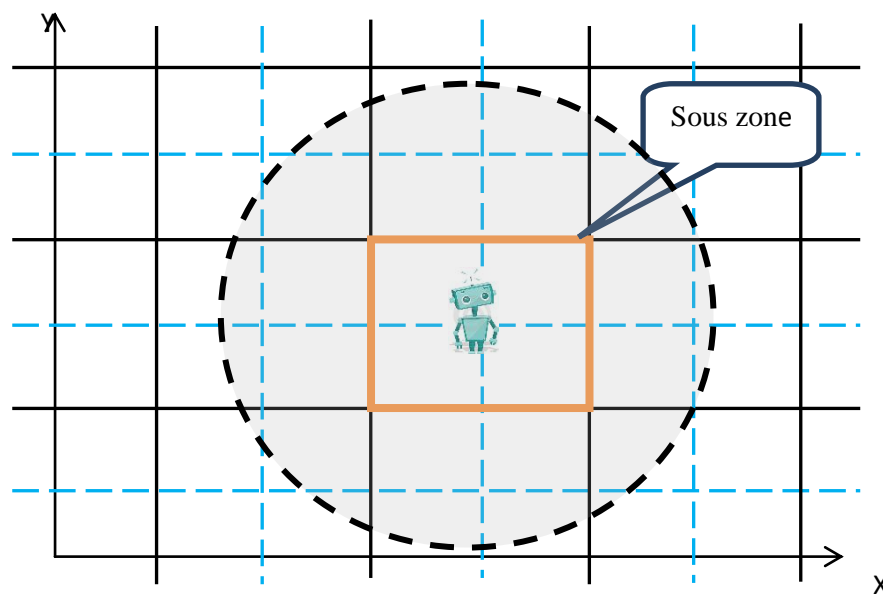


Figure 3.3: Division de la zone d'intérêt en sous zone selon la grille de déplacement des robots.

Afin de garantir l'échange des informations entre les capteurs et les robots, seuls les capteurs les plus proches du point de la grille peuvent envoyer des messages au robot. Selon la distance entre les capteurs et leur point de la grille calculé, un capteur peut avoir initialement un des deux rôles "normal" ou "responsable". Si cette distance calculée localement par un capteur est inférieure au rayon de communication des capteurs, ce capteur est considéré comme un responsable de communication "responsable". Si non le capteur est de type "normal". Le calcul de la distance entre le capteur et le point de la grille est expliqué dans la section 'Mécanismes de base'. La Figure 3.4 montre la position des capteurs responsables au niveau de leur sous zone.

Chaque capteur d'une sous zone est attaché à un point de la grille. S'il détecte une panne, il envoie son rapport de panne vers son point. Les capteurs responsables (les capteurs les plus proches de ce point et qui peuvent communiquer avec le robot) récupèrent les rapports des pannes envoyés, ensuite, ils attendent l'arrivée d'un robot pour l'informer des pannes.

Lorsque la durée d'attente de réparation d'une panne dépasse un certain seuil, les capteurs doivent informer les sous zones voisines afin de rediriger les robots qui visitent ces zones vers la position de la panne qui devient urgente. Pour éviter que tous les capteurs responsables envoient le message d'urgence, ils choisiront parmi eux le capteur qui possède plus d'énergie résiduelle pour envoyer ce message, ce capteur appelé "Grid head". L'idée de Grid head est utilisée dans plusieurs applications par exemple dans le protocole de routage cité dans [62]. La méthode de sélection de grid head est expliquée dans la section "Mécanismes de base".

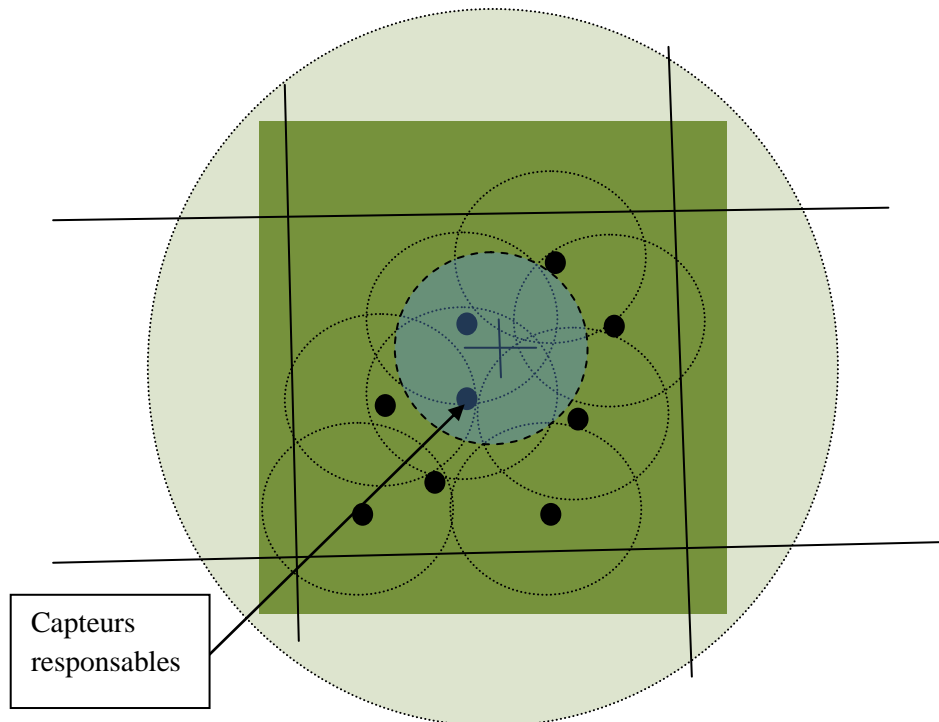


Figure 3.4: Attribution des rôles aux capteurs.

Un capteur peut avoir alors un des trois rôles: normal, responsable ou Grid head. Le rôle des capteurs normaux est la détection des pannes et l'envoi des rapports de pannes vers les nœuds responsables (leur point de grille). Par contre, les capteurs responsables possèdent les fonctionnalités des capteurs normaux, de plus, ils communiquent avec les robots afin de les informer des pannes de leur sous zone, et les dirigent vers les positions les moins récemment visités. Ils entretiennent à leurs niveaux les structures de données nécessaires pour assurer ces tâches. Ces structures sont : la liste des directions, la liste des pannes et la liste des directions d'urgence. Le *Grid head* possède les mêmes responsabilités que les capteurs responsables, en plus, il d'envoie les messages des pannes d'urgence aux capteurs responsables des sous zones voisines.

Quand un robot est informé de l'existence d'une ou de plusieurs pannes, il se déplace vers les emplacements des pannes et place un nouveau capteur au niveau de chacune. Après la réparation des pannes indiquées, ce robot revient au point de la grille courant et continue son mouvement vers la sous zone prochaine suivant la direction proposée par les capteurs. La Figure 3.5 montre un exemple de déplacement pour la réparation des pannes d'une sous zone.

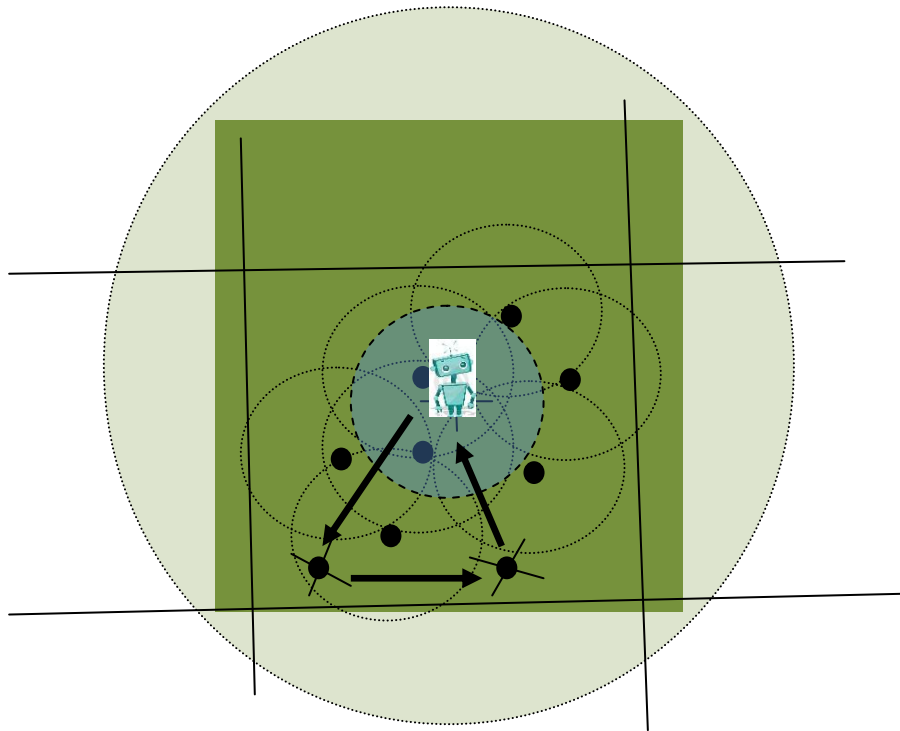


Figure 3.5: Chemin de réparation parcouru par un robot.

3.3 Apports de notre solution

Afin de minimiser le temps moyen de traitement d'une panne et la durée maximale d'attente de réparation d'une panne et la consommation d'énergie des capteurs due à leur participation au traitement des pannes, on a cherché à définir des stratégies qui minimisent le nombre d'arrêts des robots. La méthode consiste à agrandir la distance de déplacement et à rediriger les robots vers les emplacements les moins récemment visités, ainsi que, vers les pannes qui ont une durée d'attente de réparation dépassant un certain seuil.

3.3.1 Le pas de déplacement

Afin de minimiser le nombre de messages échangés entre les robots et les capteurs, et de minimiser le temps de réparation d'une panne, on a tenté de minimiser le nombre d'arrêts effectués par les robots. Concernant l'approche (Grid R3S2(Xu. Li et *al* 2013)), le pas de déplacement des robots est inférieur ou égale au rayon de communication des capteurs. Cette distance permet aux robots de communiquer avec tous les capteurs de la zone de déploiement. Mais, généralement le rayon de communication des capteurs est petit, ce qui augmente sensiblement la fréquence d'arrêt des robots. De plus, à chaque arrêt, les robots communiquent avec les capteurs. Par contre, dans notre contribution, et pour minimiser les communications entre les capteurs et les robots, et améliorer le temps de réparation d'une panne, on a cherché à minimiser le nombre d'arrêts effectués par les robots. À chaque fois que la distance de déplacement est grande, le nombre d'arrêts est minimum. Le pas de déplacement définie dépend de la différence des capacités physiques entre les capteurs et les robots

de telle sorte qu'à chaque fois que le rayon de communication d'un robot est grand ce pas s'agrandi aussi.

La taille du pas de déplacement est la distance parcourue par le robot pour atteindre le centre d'une sous zone voisine. Lorsque le robot s'arrête au centre d'une sous zone (sur un point de la grille), il doit informer les capteurs responsables des sous zones voisines de sa position actuelle. Le pas de déplacement doit assurer que le rayon de communication de robot atteigne tous les capteurs responsables de toutes les sous zones voisines (voir la Figure 3.6). Mathématiquement, le R_c (rayon de communication de robot) est :

$$R_c = D + r_c \quad (1)$$

où r_c est le rayon de communication d'un capteur, et D est le pas de déplacement d'un robot. Selon (1), on déduit que la distance de déplacement d'un robot est égale au rayon de communication de robot moins le rayon de communication d'un capteur.

$$D = R_c - r_c \quad (2)$$

Selon (2), à chaque fois que la différence entre le rayon de communication de robot et le rayon de communication d'un capteur est grande (les robots sont plus puissants par rapport aux capteurs), le pas de déplacement est grand. Par conséquent, les robots font moins d'arrêts. La Figure 3.6 montre le pas de déplacement par rapport au rayon de communication de robot et de capteur.

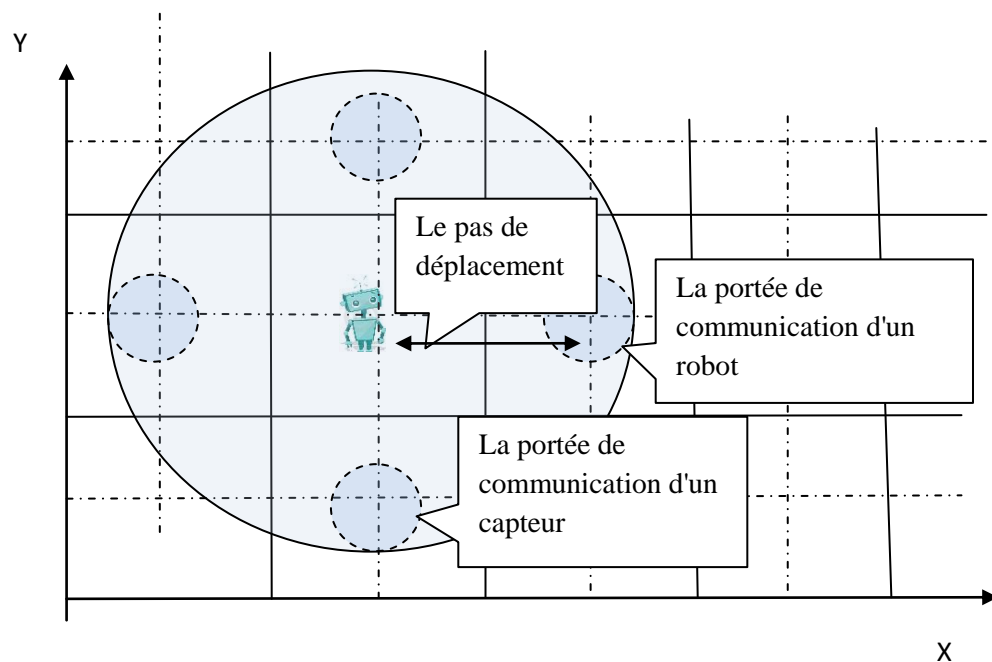


Figure 3.6: Le pas de déplacement.

3.3.2 Utilisation des listes pour la redirection des robots

Le but de l'exploration est la découverte des nouvelles pannes qui sont distribuées le long de la zone de captage. Pour se faire, les robots doivent être dirigés vers les

emplacements probables des pannes. L'approche Grid R3S2(Xu. Li et *al* 2013) inspire son algorithme d'exploration à partir de l'algorithme LRV (Least Recently Visited) (Maxim. A et *al* 2005). Les capteurs actifs attribuent pour chaque direction un numéro qui est incrémenté à chaque fois que cette direction est visitée. L'exemple suivant montre que cette méthode n'exprime pas vraiment le sens de moins récemment visité: Considérons le scénario suivant: on suppose les quatre directions, NORD, SUD, EST, OUEST. Initialement, chaque numéro de visite de chaque direction est zéro. Si un robot arrive au point OUEST, après un certain moment un autre visite le point NORD, puis un autre visite le point EST, ensuite un autre visite le point SUD. Dans ce cas, le nombre de visites de chaque direction est égal à "1". Quand un autre robot communique avec ce capteur, il trouve que toutes les directions possèdent la même priorité. Mais, réellement, la première direction qui à été visitée est la direction OUEST. Mais avec l'utilisation de numéro de séquence, le capteur dirige le robot vers une position aléatoire ou vers une position prédéfinie. Donc, le numéro de séquence ne suffit pas pour définir la direction la moins récemment visitée dans le cas où deux directions ou plus possèdent le même numéro de séquence. Pour cette raison, on a cherché à garder l'ordre de visite des directions sans pénaliser leurs priorités. Dans notre solution, on a choisi d'ordonner les directions dans une liste des directions. À chaque fois qu'une direction est visitée, elle sera dégradée à la queue de la liste, (voir plus de détail pour les mises à jour de la liste de directions dans la section suivante "Techniques de base").

3.3.3 Utilisation des listes d'urgence

Le problème d'exploration dans le cas de traitement des pannes isolées est l'aggravation des temps d'attente de réparation de quelques pannes. Pour cette raison, nous avons introduits la stratégie des messages d'urgences et les listes des directions d'urgence. Cette stratégie permet aux capteurs de diriger les robots vers les sous zones où il existe des pannes dont les durées dépassent un certain seuil.

3.4 Structure de données utilisées

L'algorithme fonctionne d'une manière distribuée localisée, dont la majorité des décisions sont prises selon l'état actuel des données locales. Ces données sont généralement les listes des pannes, et les listes utilisées pour la redirection des robots. D'après les différents messages reçus, les nœuds peuvent manipuler l'état de ces données locales et prendre les bonnes décisions.

3.4.1 Messages

Les messages sont échangés généralement entre les capteurs responsables et les capteurs normaux, entre les capteurs responsables et les robots, et localement entre les capteurs responsables. Ci-dessous, on représente quelques messages échangés.

- **Hello** : Chaque capteur diffuse périodiquement un message de signalisation qui contient sa position, en plus son niveau actuel d'énergie dans le cas des capteurs responsables.
- **Rapport de panne** : Quand un capteur détecte la(es) panne(s) d'un ou de plusieurs de ses voisins, il envoie à son point de la grille un message qui contient la(es) position(s) géographique(s) de ces dernières.

- **Message de signalisation d'arrivée d'un robot :** Quand un robot arrive à un point de la grille, il informe les capteurs de la sous zone correspondante à ce point et les sous zones voisines de sa présence. Ce message de signalisation contient l'identifiant du robot et sa position actuelle, un numéro de séquence (incrémenté à chaque fois que ce robot visite un nouveau point, il permet de différencier entre les différentes visites). La manipulation du numéro de séquence suit le même principe que celui utilisé dans l'algorithme Grid R3S2(Xu. Li et *al* 2013).
- **Message de la liste des pannes et de direction :** Quand un capteur responsable détecte l'arrivée d'un nouveau robot au niveau de sa zone (sa position actuelle est le point de la grille de sa sous zone), il envoie une réponse qui contient: la liste des positions des pannes existantes, la direction proposée (NORD, SUD, EST, OUEST), l'identifiant de robot concernée.
- **Message d'urgence:** Si le temps d'attente de réparation d'une panne dépasse un certain seuil, le grid head d'une sous zone envoie ce message vers les sous zones voisines. Ce message contient essentiellement les informations sur la sous zone de la panne d'urgence.
- **Message "Grid head":** Il est diffusé périodiquement à deux saut au maximum par un capteur responsable, afin d'informer les autres capteurs responsables de son niveau d'énergie.

3.4.2 Les différentes listes utilisées

Elle consiste en les structures de données qui se trouvent au niveau des capteurs responsables :

- **La liste des directions:** Elle représente les directions les moins récemment visitées. Le nombre d'éléments de cette liste est égale au nombre des voisins (directions permises). Une direction est représentée essentiellement par un indicateur (NORD, SUD, EST, OUEST) et la position géographique d'une sous zone atteinte par un robot s'il suit cette direction, ainsi que d'autres informations supplémentaires.
- **La liste des messages d'urgences :** Cette liste n'est créée ni manipulée que si le capteur reçoit un message d'urgence. Elle contient les directions d'où proviennent les messages d'urgences. Les directions d'urgences sont ordonnées dans cette liste selon l'ordre d'arrivée de ces messages.
- **La liste des pannes :** Elle se trouve au niveau des capteurs responsables, elle contient les positions des pannes avec le temps d'attente de réparation de chaque panne.

3.5 Mécanismes de base

Dès la détection d'une panne jusqu'à sa réparation, notre algorithme applique différents mécanismes et techniques de calcul, afin de, réussir les différentes étapes de traitement des pannes isolées. Certaines techniques ont été inspirées des solutions qui existent dans la littérature, comme la détection locale [63][10], la méthode de réparation et le routage géographique utilisé dans [1]. On a introduit certaines d'autres, comme la méthode de calcul de point de la grille, le calcul des directions permises, l'attribution des rôles, la mise à jour des listes. Ci-dessous sont données les descriptions de chaque technique utilisée.

3.5.1 Calcul de point de la grille

Lorsque un capteur détecte une panne, il envoie un rapport de panne via un protocole de routage (routage géographique "GPSR [64]") vers les capteurs responsables approximatifs de son point de la grille virtuelle. Lors de l'initialisation, en utilisant le rayon de captage des robots, le rayon de captage des capteurs et les limites de la zone de couverture (x_{min} , x_{max} , y_{min} , y_{max}), la position géographique de capteur/robot, chaque capteur et chaque robot défini dans quelle sous zone il est positionné. Ci-dessous, on représente la position géographique d'un nœud par $P_c (X_c, Y_c)$ et la position géographique de point de la grille par $P_g (X_g, Y_g)$.

Le calcul de la position géographique du centre d'une sous zone (point de la grille) effectué par les nœuds (robots ou capteurs), selon le processus suivant:

1. Définir le pas de déplacement. $D = R_c - r_c$ (2). R_c : rayon de communication de robot, r_c : rayon de communication d'un capteur.
2. Calculer le numéro de la sous zone pour l'axe des "X", puis pour l'axe des "Y" en utilisant l'équation (3) et (3'). Dans un plan à deux dimensions composé de deux axes (voir Figure 3.7): l'axe des X et l'axe des Y, le numéro d'une sous zone selon l'axe des X(Y) est le résultat entier de la division de la différence entre X_{min} (Y_{min}) de la zone de captage et X_c (Y_c) de nœud par le pas de déplacement D .

$$\text{NuméroX} = (X_c - X_{min}) / D \quad (3)$$

$$\text{NuméroY} = (Y_c - Y_{min}) / D \quad (3')$$

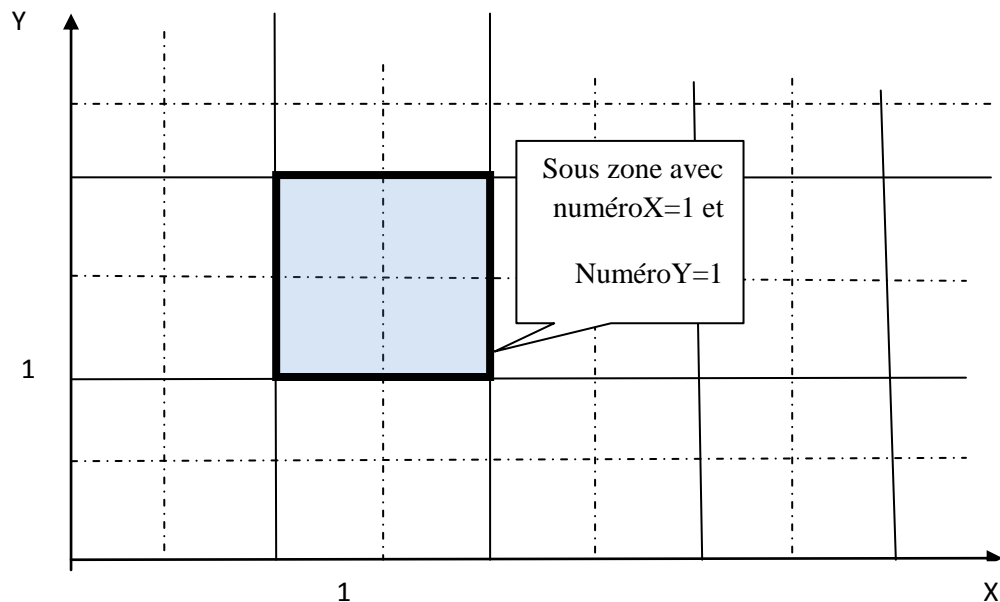


Figure 3.7: Calcul des numéros d'une sous zone selon l'axe X et l'axe Y.

3. Chaque capteur peut donc calculer les coordonnées de son point de la grille, qui est le centre d'une sous zone selon l'équation (4).

$$X_g = X_{min} + (D \times \text{numéro}X) + (D \times 1/2) \quad (4)$$

X_g doit être inférieur à X_{max}

$$Y_g = Y_{min} + (D \times \text{numéro}Y) + (D \times 1/2) \quad (4')$$

Y_g doit être inférieur à Y_{max}

4. Si les coordonnées d'un point de la grille calculées par un nœud dépassent les frontières de la zone de déploiement ($X_g > X_{max}$ ou $Y_g > Y_{max}$), ce dernier choisit le point de la grille de la sous zone la plus proche de lui et qui se trouve à l'intérieur de la zone de déploiement. Donc, le calcul de numéro de la sous zone est effectué selon les différents cas suivants:

1. Si $X_g > X_{max}$ et $Y_g > Y_{max}$

$$\text{Nouveau numéro}X = \text{numéro}X - 1.$$

$$\text{Nouveau numéro}Y = \text{numéro}Y - 1$$

2. Si $X_g > X_{max}$ et $Y_g < Y_{max}$

$$\text{Nouveau numéro}X = \text{numéro}X - 1.$$

$$\text{Nouveau numéro}Y = \text{numéro}Y$$

3. Si $X_g < X_{max}$ et $Y_g > Y_{max}$

$$\text{Nouveau numéro}X = \text{numéro}X.$$

$$\text{Nouveau numéro}Y = \text{numéro}Y - 1$$

Exemple explicatif (voir Figure 3.8)

On suppose l'exemple de la Figure 3.8. On pose $X_{min}=Y_{min}= 0$, $X_{max}=8,5$ et $Y_{max}=7$, $D=2$, $P_c(0,5,2,7)$. Le numéro de la case de capteur A est :

$$\text{Numéro}X=0 \quad \text{et} \quad \text{Numéro}Y=1$$

Le point de la grille du capteur A,

$$X_g = 0 + (2 \times 0) + (2 \times 1/2) = 1 \quad \text{et} \quad Y_g = 0 + (2 \times 1) + (2 \times 1/2) = 3$$

Alors les coordonnées du point de la grille du capteur A est (1,3).

Le capteur B (8,2, 5) est situé dans la sous zone de numéro:

$$\text{Numéro}X=4 \quad \text{et} \quad \text{Numéro}Y=2$$

Le point de la grille de B est

$$X_g = 0 + (2 \times 4) + (2 \times 1/2) = 9 > X_{max} \quad \text{et} \quad Y_g = 0 + (2 \times 2) + (2 \times 1/2) = 5$$

$9 > X_{max}$, par conséquent le nouveau numéro de la sous zone de B est $4-1=3$, le nouveau point de la grille de B est alors (7,5).

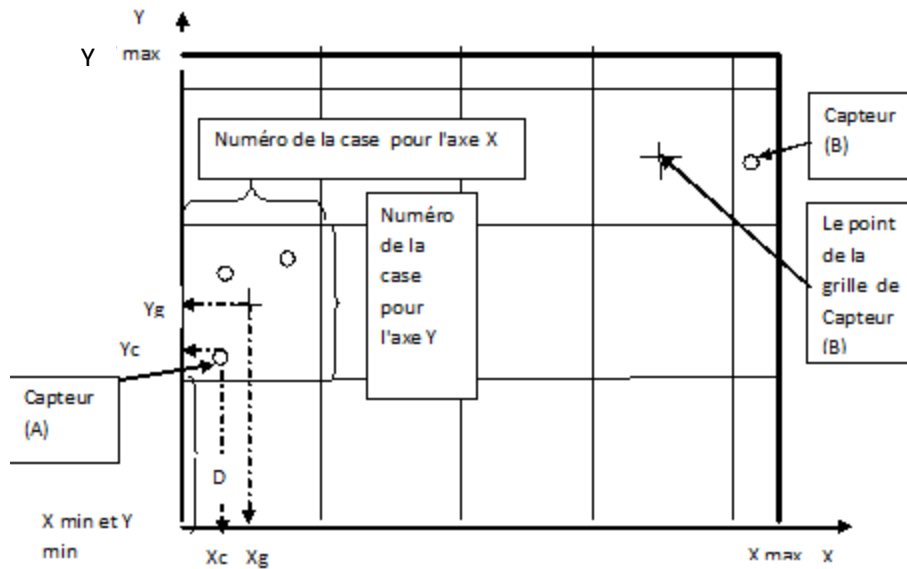


Figure 3.8: Exemple de calcul d'un point de la grille ;

3.5.2 Attribution des rôles

Comme on a déjà expliqué dans la section "Structuration et rôles", un capteur peut définir son rôle qui est soit normal, responsable ou Grid head. Initialement et après le calcul du point de la grille de capteur, il calcule la distance cartésienne qui le sépare de ce dernier selon la formule (5):

$$\text{distance} = \sqrt{|Xc - Xg|^2 + |Yc - Yg|^2} \quad (5)$$

Si cette distance est inférieure au rayon de communication d'un capteur, alors ce capteur est un capteur responsable (Voir la formule (6))

$$\text{Rôle} = \begin{cases} \text{si distance} < rc & (\text{responsable}) \\ \text{si non} & (\text{normal}) \end{cases} \quad (6)$$

Exemple explicatif : On suppose l'exemple cité dans la section précédente "calcul de point de la grille". La distance qui sépare le capteur A (0.5, 2.7) et son point de la grille Pg(1,3) et avec un rayon de communication égale à 1 est: Distance = $\sqrt{|0.5 - 1|^2 + |2.7 - 3|^2} = \sqrt{|0.5 - 1|^2 + |2.7 - 3|^2} = 0.59 < 1$, alors A est un capteur responsable.

Concernant la désignation de Grid Head, les capteurs responsables diffusent périodiquement des messages "Grid head" qui contiennent leur niveau d'énergie actuelle. Chaque capteur responsable compare son niveau d'énergie avec celui des autres capteurs responsables, si son niveau d'énergie est le plus grand, il devient le Grid head et il prend la responsabilité d'envoi des messages d'urgence aux sous zones voisines. Lorsque le Grid head détecte que le temps d'attente de réparation d'une panne dépasse un certain délai, il envoie un message d'urgence vers les sous zones voisines. Ce mécanisme de choix de Grid Head est déjà utilisé dans d'autres

approches comme la méthode de calcul de l'ensemble CDS (clustering) utilisé dans l'approche Cluster R3S2 et Cluster Grid R3S2 (Xu. Li et *al* 2013) .

3.5.3 Calcul des directions permises

Afin de bien rediriger les robots, les capteurs responsables calculent les points de la grille voisins qui représentent les sous zones voisines. En général, lorsqu'un robot désire visiter une autre sous zone, les capteurs responsables peuvent le diriger vers une des quatre directions : Nord, Sud, Est, Ouest. Pour éviter que le robot se dirige vers une position hors de la zone de déploiement, les capteurs responsables calculent et vérifient la validité de chaque direction calculée, afin de construire la liste des directions permises. La liste des directions permise possède au maximum quatre éléments et au minimum deux (voir Figure 3.9). Ce calcul est effectué lors de l'initialisation après le calcul de point de la grille et l'attribution des rôles.

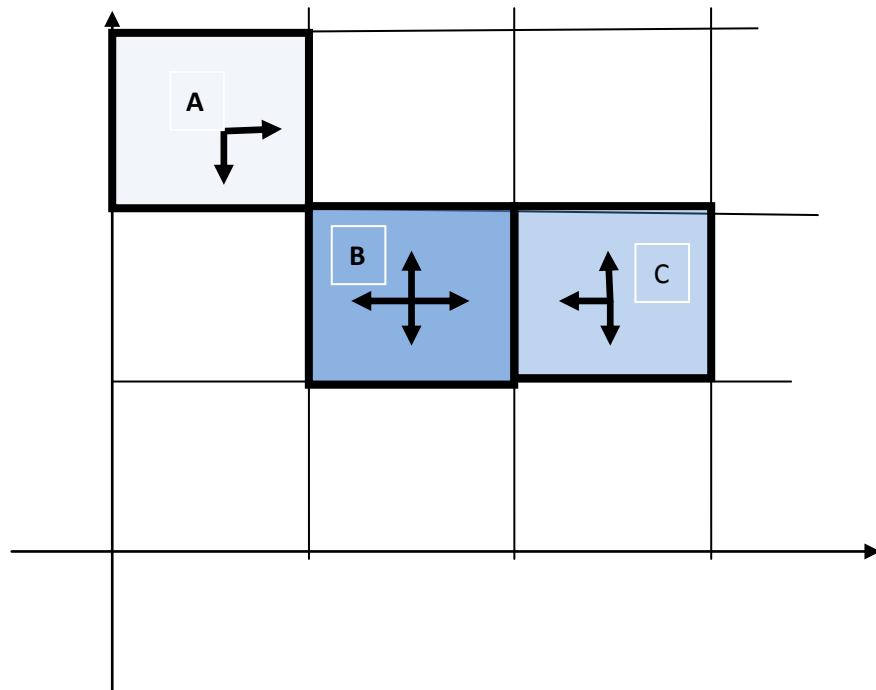


Figure 3.9: Différence entre les directions permises selon la position de la sous zone.

Dans la Figure 3.9, la sous zone A possède deux directions permises et la sous zone C possède trois directions, par contre, la sous zone B possède quatre directions.

La Table 3.1 montre la méthode de calcul et de vérification de chaque direction. Chaque direction valide sera ajoutée à la liste des directions permises correspondantes à chaque sous zone.

Le résultat de ce calcul est représenté par une liste qui possède l'ordre initial suivant: NORD, SUD, EST, OUEST. La Figure 3.10 présente un exemple de calcul d'une direction d'une sous zone.

| Direction | Equation de calcul des positions des points de la grille voisin. | Condition de refus d'une direction |
|--------------|--|------------------------------------|
| NORD | $X_{Nord}=X_g$ $Y_{Nord}= Y_g + D$ | $Y_g + D > y_{max}$ |
| SUD | $X_{SUD}=X_g$ $Y_{SUD} = Y_g - D$ | $Y_g - D < y_{min}$ |
| EST | $X_{EST}=X_g + D$ $Y_{EST} = Y_g$ | $X_g + D > X_{max}$ |
| OUEST | $X_{OUEST}= X_g - D$ $Y_{OUEST}= Y_g$ | $X_g - D < X_{min}$ |

Table 3.1: Méthode de calcul des directions permises.

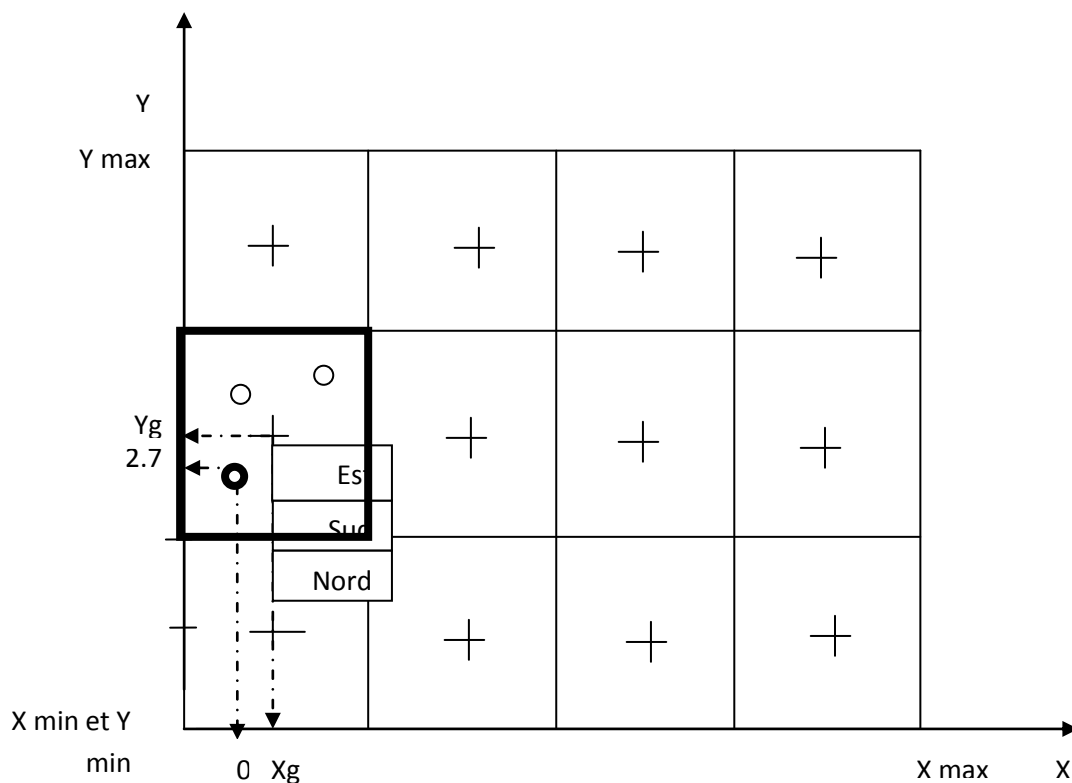


Figure 3.10: Exemple de calcul des directions permises.

Pour le capteur A (0.5, 2.7) qui se trouve dans la sous zone de point de la grille (1, 3) avec un rayon de communication de 1, la direction OUEST calculée n'est pas

permise pour ce capteur (le capteur ne propose jamais cette direction au robot visitant).

$$\text{OUEST} = (1 - 2, 3) = (-1, 3)$$

$-1, 3 < 0$ la direction OUEST n'existe pas.

Cette méthode nous permet aussi d'introduire la notion d'obstacle si ces derniers ont la forme d'un rectangle et si tous les capteurs connaissant les coordonnées des obstacles, ils peuvent éliminer les directions qui passent par des obstacles.

3.5.4 Détection locale de la panne

Selon un mécanisme similaire à celui de Grid R3S2(Xu. Li et *al* 2013) chaque capteur envoie périodiquement un message "hello" qui contient sa position afin d'informer ses voisins de sa présence. Un capteur détecte la panne d'un de ses voisins en exécutant les étapes suivantes:

1. Selon les positions de ses voisins actifs, chaque capteur calcule les arcs non couverts.
2. Si l'angle d'un arc non couvert est supérieur ou égale à π , cet arc sera divisé en deux (le placement d'un nouveau capteur dans ce trou ne suffira pas pour le remplir).
3. Pour chaque arc non couvert (un de ses voisins est en panne), chaque capteur calcule la position de la panne (où placer un nouveau nœud) qui est la position de la symétrie du centre du nœud par rapport à la corde de l'arc.
4. La position du nouveau nœud est décalée au capteur suivant la médiatrice jusqu'à ce que la distance entre ces deux nœuds soit inférieure ou égale au rayon de communication d'un capteur. Ce décalage permet d'assurer la connectivité entre le capteur et le nouveau nœud calculé.

La Figure 3.11 montre un exemple de détection et de calcul des arcs non couverts. Dans cette Figure, le capteur S détecte deux arcs non couverts $\widehat{a_1b_1}$ et $\widehat{a_2b_2}$, il détermine la position probable des trous par P_1 et P_2 , mais si un nouveau capteur est déployé dans la position de P_1 , il ne sera pas connecté avec S, alors la position calculée P_1 est décalée au centre de S, et remplacée par P_1' .

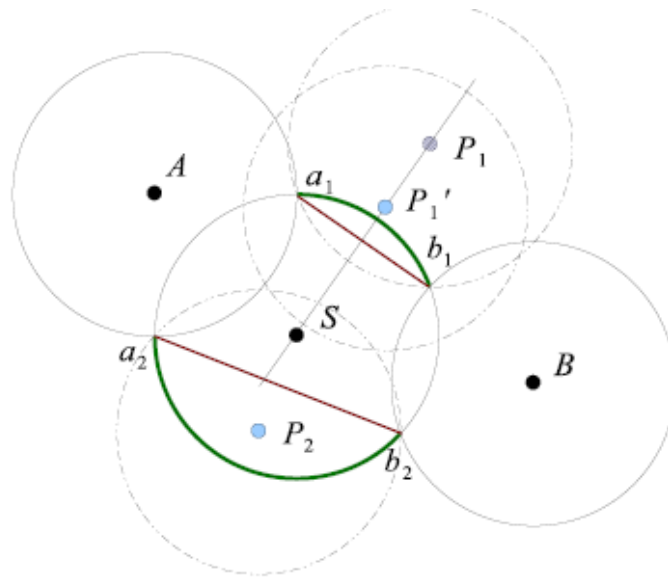


Figure 3.11: Identification des trous locaux Grid R3S2 (Xu. Li et *al* 2013).

Cette méthode permet la détection du manque de couverture, tandis que, cette méthode de calcul de la position de trou assure la connectivité seulement entre le capteur détecteur de panne et le nouveau capteur à déployer. Mais, à cause de traitement localisé des positions des pannes, deux ou plusieurs positions des pannes calculées par les capteurs peuvent représentées le même trou (problème de la redondance). Pour régler ce problème, les auteurs de R3S2 ont proposé un autre algorithme Grid R3S2 (Xu. Li et *al* 2013) où ils ont introduit l'algorithme VFA (Virtual Force Algorithm). Dans notre solution, on a utilisé cette méthode avec ce problème de redondance afin d'éviter les transmissions supplémentaires générées lors de l'application de l'algorithme VFA et le clustering. On prévoit aussi d'autres améliorations sur notre solution pour éliminer ce problème.

3.5.5 Routage géographique

Les capteurs qui détectent la panne envoient leurs rapports, qui contiennent les positions des pannes, vers le point de la grille afin d'informer les capteurs responsables. Le routage géographique va nous permettre d'envoyer les messages de panne vers une position géographique, ce qui évite beaucoup de transmissions supplémentaires. Pour notre solution, on a choisi le protocole de routage géographique GPRS¹.

GPSR¹ est un protocole de routage géographique qui garanti le passage à l'échelle et utilise seulement les informations locales stockées. Dans cet algorithme, le capteur envoie les messages de données à son voisin le plus proche de la destination (le mode Greedy Forwarding) . Lorsque GPSR se trouve face à une zone de vide, il passe du mode *greedy* au mode *perimeter*.

¹ http://www.memoireonline.com/04/10/3394/m_Greedy-perimeter-stateless-routing-sur-omnet2.html, 2/5/2014

Ce faisant, il va choisir le prochain saut par rapport à la règle de la main droite. Ce choix a un caractère arbitraire et ne tient pas compte de la possibilité qu'il y ait plus d'opportunités d'acheminement d'un côté ou d'un autre. Pour augmenter le taux de réussite d'acheminement des paquets, GPSR utilise les deux algorithmes (*greedy*, *perimeter*) en fonction de la densité locale du réseau.

Le protocole GPSR nécessite dans son fonctionnement la position et l'identifiant du destinataire. Mais, dans notre contribution, les capteurs envoient leurs rapports de pannes vers une position virtuelle (point de la grille) afin d'informer les capteurs responsables adjacents de ce point des pannes existantes, où il n'existe pas nécessairement un capteur qui se trouve exactement à la position de point de la grille. Pour éviter la perte des paquets, on a adapté le mécanisme suivant, le premier capteur responsable adjacent de ce point qui reçoit le paquet stop le routage et diffuse ce paquet vers ses voisins.

L'approche décrite dans (Yongguo. M, 2007) a aussi utilisé un des protocoles de routage géographique, dont les capteurs transmettent leurs rapports de panne vers le robot de leur zone. Mais, cela suppose que le robot diffuse après chaque distance de son déplacement sa position géographique, ce qui augmente le nombre des messages. En échange, dans notre solution les capteurs envoient leurs rapports de panne vers une position virtuelle fixe déjà calculée par les capteurs.

3.5.6 Méthode de mise à jour des listes

Si un capteur responsable reçoit un message de signalisation de l'arrivée d'un robot et ce message indique que la position actuelle du robot est l'une des sous zones voisines, il met à jour ses listes locales afin de diminuer la priorité de cette sous zone:

1. il met cette direction à la queue de la liste des directions,
2. si cette direction existe dans la liste des messages d'urgence, il la supprime de cette liste.

3.6 Algorithme générale

On considère que le réseau de capteurs est initialement couvert et connecté. Les capteurs et les robots coopèrent entre eux afin de traiter toutes les pannes isolées qui peuvent intervenir dans ce réseau. D'après la structuration déjà expliquée dans la section "Structuration et rôle", on a quatre types d'acteurs : capteur normal, capteur responsable, capteur Grid head, robot. Chaque acteur participe dans une ou plusieurs phases de la solution qui est composée de cinq phases suivantes :

1. Initialisation: Elle consiste en, la structuration et l'attribution des rôles aux capteurs, le positionnement initial des robots, l'initialisation des données locales.
2. Détection: Calcul des positions des pannes. Et si le temps d'attente de réparation d'une panne dépasse un certain délai, le grid head déclenche le traitement des pannes d'urgence
3. Communication entre les robots et les capteurs: Il s'agit d'informer le robot des pannes existantes et la redirection de ce dernier.
4. Réparation: Le robot se déplace vers la position de chaque panne et place un nouveau capteur.

5. Reprise: Le nouveau capteur déployé exécute l'initialisation et commence le traitement des pannes.

Ci-dessous, on présente les différentes étapes de notre algorithme où on spécifie, pour chacune, les mécanismes et les stratégies utilisées.

3.6.1 Initialisation

Cette étape est exécutée après le déploiement des capteurs et des robots. Le but de l'initialisation pour les capteurs est la reconnaissance de la sous zone de ce capteur ainsi que son rôle (Normal, Responsable). Concernant les robots, dans la phase d'initialisation, ils doivent reconnaître le point de la grille (le centre de la sous zone) où ils doivent se déplacer pour commencer l'exploration successive de la zone.

Chaque capteur et lors de l'initialisation suit les étapes suivantes:

1. Il calcule le point de la grille (les coordonnées du centre de sa sous zone).
2. Il détermine son rôle dans le réseau.
3. Si ce capteur est un capteur responsable:
 - a) Il calcule les directions permises, et les points des grilles voisines
 - b) Il crée à son niveau une liste qui contient les directions permises. On suppose que l'ordre initial des directions est : Nord, Sud, Est, Ouest. La liste des directions d'urgence initialisée à vide.

Concernant les robots, chacun et selon sa position procède comme suit:

1. Il calcule le point de la grille le plus proche (détermine la sous zone où il se positionne).
2. Il se déplace vers la position de ce point afin de commencer l'exploration.

La méthode de calcul de point de la grille (la sous zone d'un capteur ou robot) et la l'attribution des rôles sont déjà détaillés dans la section "Mécanismes de base".

3.6.2 Détection

Rappelons que les capteurs détectent localement les pannes et préparent un rapport de panne. Ce rapport contient la position de chaque panne détectée. Chaque capteur envoie ses rapports vers son point de la grille virtuelle, afin d'informer les capteurs responsables à proximité. Les capteurs responsables à leurs tour, ajoutent la panne à la liste des pannes avec un temps d'attente de réparation égale à zéro. Après chaque période, les capteurs responsables incrémentent le temps d'attente de réparation de chaque panne, s'ils trouvent que le temps de réparation d'une panne dépasse un délai "T", ils déclenchent le processus de gestion des pannes d'urgence.

3.6.2.1 Détection des pannes

La détection des pannes est une opération effectuée localement par les capteurs afin de déterminer les positions des pannes. En général, le processus de détection d'une panne est le suivant:

1. Après chaque période, chaque capteur calcule les arcs non couverts et détermine la panne d'un de ses voisins.

2. S'il trouve une panne, il envoie un rapport de panne qui contient la position calculée de la panne vers son point de la grille (voir l'exemple présenté dans la Figure 3.12).

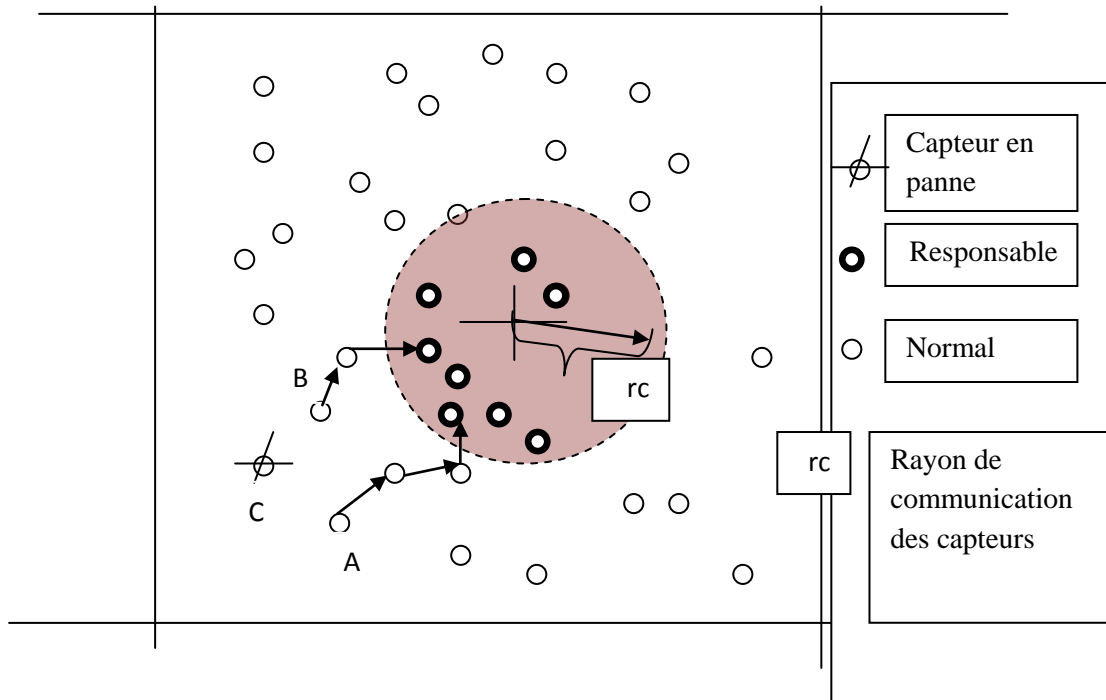


Figure 3.12: Les capteurs A et B détectent la panne du capteur C et envoient les rapports de panne vers les capteurs responsables via le routage géographique.

Quand un capteur responsable reçoit un rapport de panne, il procède comme suit:

1. Il diffuse ce rapport vers ses voisins avec un nombre des sauts qui est égal à deux. La diffusion permet aux capteurs responsables de la même sous zone d'avoir les mêmes informations sur les pannes. Si un de ces capteurs tombe en panne, ses informations reçues sur les pannes ne se perdent pas. Mais, le problème de cette stratégie est l'augmentation du nombre de messages transmis pour chaque panne, ce qui épuise l'énergie des capteurs responsables qui jouent un rôle vital.
2. Si la position d'une panne détectée n'existe pas dans la liste des pannes (nouvelle panne), il ajoute cette panne à la liste des pannes avec un temps d'attente de réparation de la panne égale à zéro.

3.6.2.2 Gestion des pannes d'urgence

Après chaque période, les capteurs responsables incrémentent la durée d'attente des pannes reçus. Si cette durée dépasse un seuil T, la panne est classée comme une panne d'urgence et les capteurs exécutent le processus suivant:

1. Grace aux messages "Grid head" diffusés périodiquement à deux sauts, les capteurs responsables choisissent parmi eux un capteur grid head selon la méthode déjà expliquée dans la section "Mécanismes de base".
2. Le grid head envoie un message d'urgence vers ses sous zones voisines, ce message contient les informations suivantes: le point de la grille courant (le point de la grille définie la sous zone de ce capteur). Les sous zones voisines de ce capteur sont calculées lors de l'initialisation et enregistrées dans la liste des directions permises. L'envoi de ce message est effectué en utilisant le routage géographique.

Si un capteur responsable reçoit un message d'urgence d'un voisin, il exécute les étapes suivantes:

1. Il ajoute la sous zone de la source de message (la sous zone de capteur d'où vient le message) à la liste des directions d'urgences.
2. Il diffuse le message vers les autres capteurs responsables, afin de leurs permettre d'effectuer les mises à jour nécessaires sur ces listes des directions d'urgence.

Dans l'exemple représenté dans la Figure 3.13, les capteurs responsables envoient un message d'urgence, la liste des messages d'urgence redirige les robots vers la sous zone où se trouve la panne d'urgence.

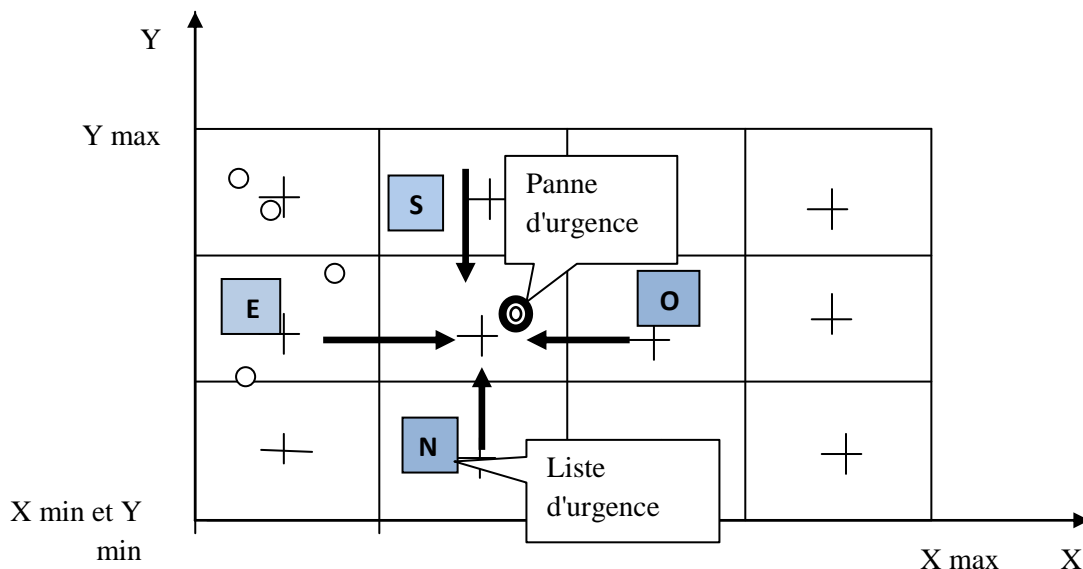


Figure 3.13: Etats des listes après l'envoi d'un message d'urgence.

3.6.3 Information et redirection des robots

Quand un robot arrive à une sous zone, il communique avec les capteurs responsables de cette sous zone, pour être informé des pannes existantes. La communication entre les capteurs et ce robot se déclenche lorsque ce dernier diffuse un message de signalisation de son arrivée. Un message de signalisation de d'arrivée de robot contient essentiellement les informations suivantes (l'identifiant robot, numéro de séquence, position courante de robot).

Lorsqu'un capteur responsable reçoit un message de signalisation dont l'identifiant de robot et le numéro de séquence sont différents de celle déjà enregistrée, et si la position de robot est la même position que le centre de sa sous zone (son point de la grille), alors, il effectue le traitement suivant:

1. Il envoie le message de la liste des pannes qui contient les positions ordonnées des trous, et la prochaine direction de robot. Rappelons que la liste des pannes contient les positions des pannes classées (selon l'ordre d'intervention de chacune) + DIRECTION + l'identifiant de robot concerné.
2. La direction proposée sera choisie selon l'état de la liste de directions et la liste des messages d'urgence.
 - a) Si la liste des messages d'urgence est vide, alors, la direction proposée est la première direction de la liste des directions permises.
 - b) Sinon la nouvelle direction est le premier élément de la liste des directions d'urgence.
 - c) Il remplace l'identifiant de robot qui existe à son niveau par le nouveau.

Dans le cas où la position indiquée dans le message de signalisation d'arrivée de robot est une sous zone voisine de la sous zone de capteur responsable, il dégrade la direction courante de robot à la queue de sa liste des directions permises et il supprime cette direction de la liste des messages d'urgence si elle existe.

La Figure 3.14 montre un aperçu sur l'état des listes des directions d'urgence et les listes de directions avant que le robot diffuse son message de signalisation. La Figure 3.15 représente l'état des listes après la diffusion de message de signalisation par le robot. Les capteurs proposent la direction Sud qui se trouve au bout de la liste des directions d'urgences et la liste des directions. Les capteurs responsables des directions voisines mettent à jour les listes des directions qui se trouvent à leurs niveaux (voir la Figure 3.13).

Dans l'exemple de la Figure 3.14 et 3.15, le robot se déplace vers la direction SUD avec une distance "D", et il incrémente son numéro de séquence. Quand il arrive à la sous zone voisine qui est située au sud de sa sous zone actuelle, il s'arrête et commence la communication avec les capteurs, afin de l'informer des nouvelles pannes qui peuvent exister dans cette nouvelle sous zone.

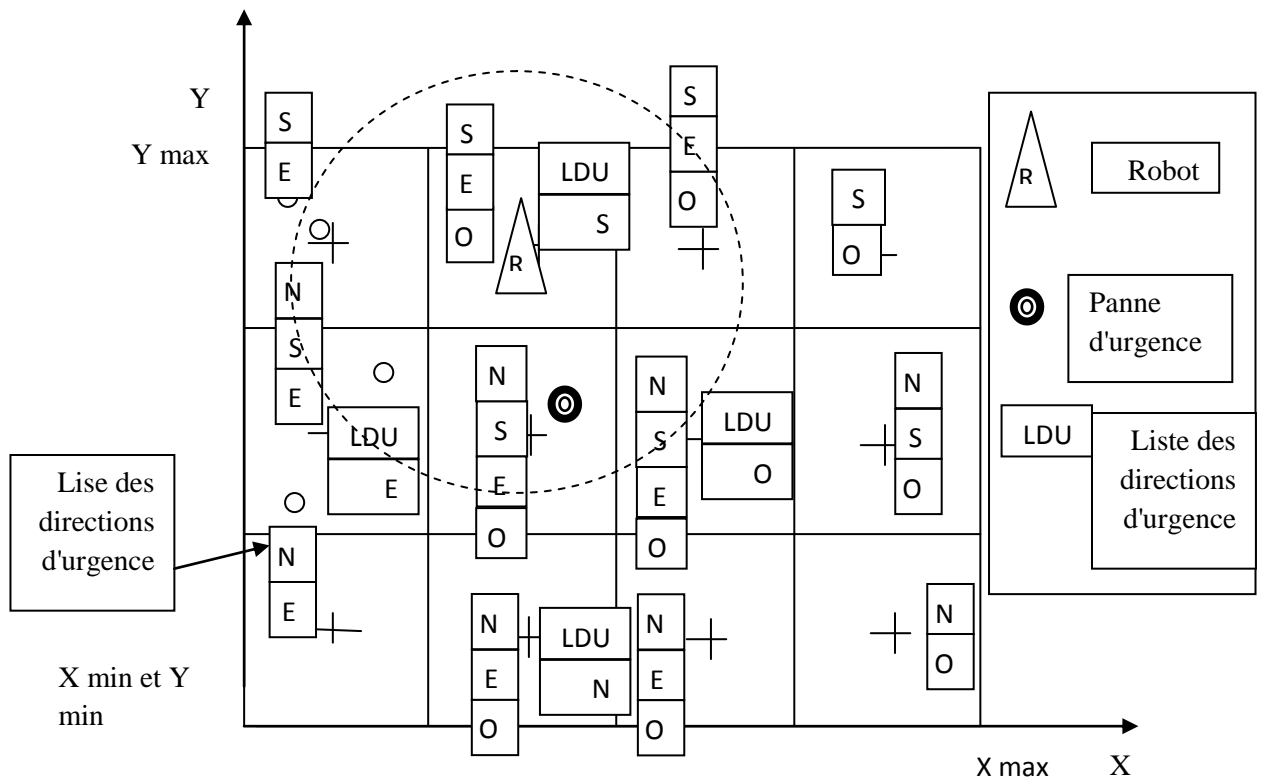


Figure 3.14: Etat du système avant que le robot diffuse son premier message de signalisation.

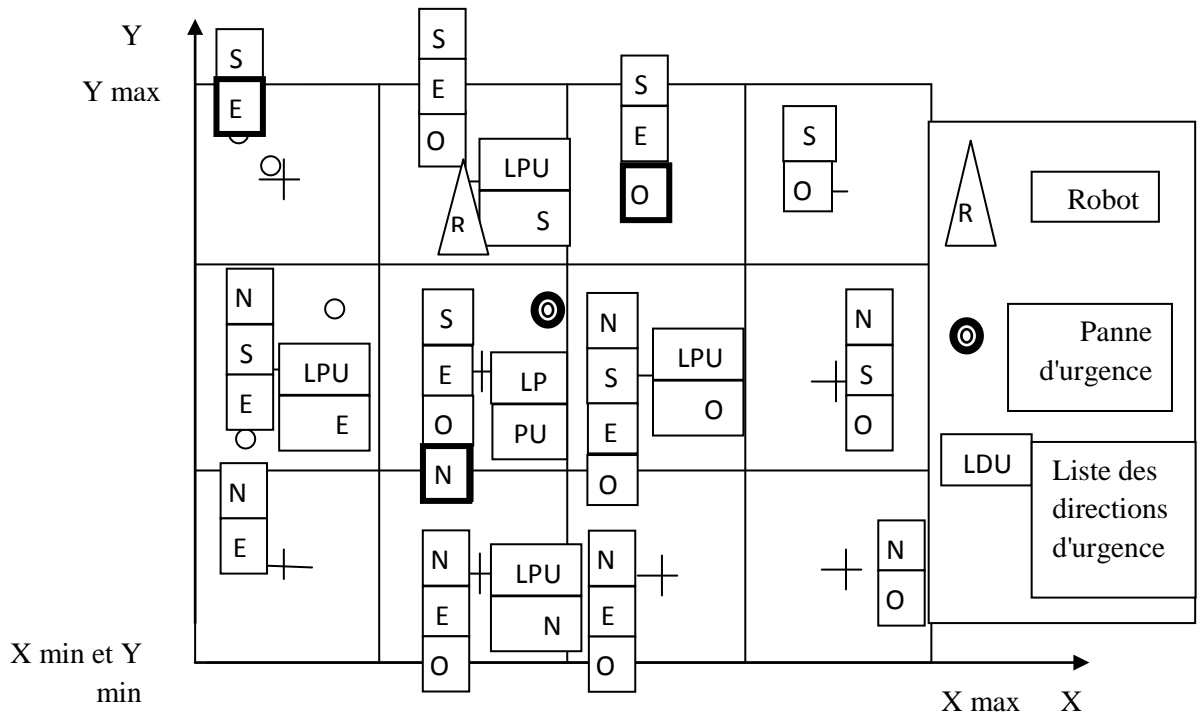


Figure 3.15: Etat du système après que le robot diffuse son premier message de signalisation.

3.6.4 Réparation

A la fin de l'étape de communication, le robot reçoit la liste des pannes qui interviennent dans sa sous zone actuelle. Suivant le même ordre indiqué dans cette liste, il se déplace d'une position de panne vers une autre. Quand il arrive à la position d'une panne, il redéploie (lance) un nouveau capteur. Quand il termine la réparation de toutes les pannes, il revient au centre de sa sous zone actuelle (point de la grille actuelle), ensuite il continue son mouvement vers une prochaine sous zone suivant sa nouvelle direction proposée par les capteurs. Rappelons qu'un robot ne se déplace pas vers un autre point de la grille que s'il répare toutes les pannes de sa sous zone.

3.6.5 Reprise

Le capteur nouvellement déployé exécute une phase d'initialisation. S'il est un capteur responsable, il envoie un message "new" à ses voisins (reconnue par l'écoute des messages "hello"). Quand un capteur responsable reçoit ce message, il envoie à ce nouveau capteur un message qui contient: la liste des directions, la liste des directions d'urgence, le dernier robot visitant de sa sous zone. Cette opération assure que les capteurs responsables possèdent toujours la même information malgré les pannes.

4. La gestion des cas d'exceptions

Plusieurs cas d'exceptions peuvent intervenir, on cite ci-dessous quelques cas, et les mécanismes utilisés pour les gérer.

Cas 1: Le mécanisme de la gestion des pannes d'urgence peut attirer plusieurs robots vers un point de la grille (celui qui contient la panne d'urgence). Par conséquent, plusieurs robots peuvent se retrouver dans la même sous zone. Quand un capteur détecte l'arrivée de plusieurs robots à sa sous zone (il reçoit des messages de signalisations dont les identifiants des robots sont différents), il procède comme suit:

1. Il choisit toujours pour la réparation, le robot dont l'identifiant est le plus petit (il diffuse un message de panne qui contient cet identifiant).
2. Selon les directions disponibles et suivant l'ordre indiqué dans les listes, il redirige les autres robots. Si le nombre de robots dépasse le nombre de directions, le reste des robots choisissent leurs prochaines directions aléatoirement.

Cas 2: Lorsqu'un capteur responsable reçoit deux ou plusieurs messages d'urgence venant des sous zones voisines différentes, sa liste des messages d'urgence peut alors contenir plusieurs directions. Dans cette situation, les capteurs responsables proposent au robot la première direction sans l'informer des autres sous zones d'urgence. Dans ce cas, le robot peut s'éloigner des sous zones des pannes d'urgence. On peut éviter ce problème si les capteurs informent les robots de toutes la liste des directions des pannes d'urgence, et lorsqu'il répare les pannes d'urgence de la première direction, et si dans cette sous zone, le robot reçoit une liste des directions d'urgence vide, il revient à la sous zone où il a été informé des pannes d'urgence, et il communique avec ses capteurs responsable pour le diriger vers les pannes d'urgence restantes, et nécessairement qui n'ont pas été réparées par un autre robot.

5. Discussions

Le but principal de notre solution est le traitement des pannes isolées en dirigeant les robots vers les emplacements des pannes, notamment, les pannes urgentes. Pour assurer cette tâche, plusieurs stratégies et mécanismes sont exploités pour que les capteurs responsables dirigent les robots vers les positions les moins récemment visitées qui sont classées selon la liste des directions permises, ainsi que, vers les directions des pannes d'urgences, en utilisant la liste des directions d'urgence. Mais plusieurs questions peuvent se poser concernant notre algorithme d'exploration : est-ce qu'il garanti le non blocage des robots? Est-ce qu'il garanti que le temps d'exploration soit acceptable? Est-ce qu'il peut résister aux contraintes des réseaux de capteurs? Toutes ces questions auront leurs réponses dans les mesures de simulation.

Dans les sections précédentes, on n'a pas spécifié la valeur de quelques paramètres comme la distance de déplacement, le nombre des sauts pour le message d'urgence. La fixation de leurs valeurs se fait dans l'étape de simulation, afin d'obtenir les meilleures valeurs de ces paramètres et qui peuvent nous permettre un temps de réparation et une consommation d'énergie minimale.

Plusieurs problèmes peuvent exister dans notre solution, on cite par exemple:

- La panne des capteurs responsables: Les capteurs responsables jouent un rôle vital dans notre solution, mais, afin d'accomplir leur mission ils dépensent beaucoup d'énergie. L'épuisement de l'énergie des capteurs responsables cause la panne de ces derniers, ce qui influe directement sur la redirection des robots et l'information de ces derniers des pannes existantes.
- La stratégie d'urgence peut attirer plusieurs robots vers la même sous zone.
- La stratégie d'urgence attire des robots vers une panne alors que cette dernière est réparée par un autre.
- La stratégie de détection des pannes effectuée par les capteurs génère le lancement de plusieurs messages avec des différentes positions alors qu'il existe un seul trou.

Afin d'éviter ces problèmes, plusieurs améliorations peuvent être apportées, on cite par exemple:

- Si on considère que la panne d'un capteur responsable est une panne urgente: la réparation de cette panne est plus prioritaire, alors le nombre des capteurs responsables reste suffisant pour assurer le service demandé.

6. Conclusion

Dans ce chapitre, nous avons décrit une nouvelle solution de réparation des pannes isolées des capteurs moyennant l'utilisation des robots mobiles. L'approche adoptée se base sur l'exploration des robots pour découvrir les pannes déjà détectées par les capteurs et gérées par les capteurs responsables. Dans cette solution, on a estimé d'améliorer plusieurs aspects comme la minimisation du nombre d'arrêts, la direction des robots vers les pannes d'urgence et les directions moins récemment visitées. Les mécanismes introduits favorisent la minimisation du temps de réparation et

l'optimisation du nombre de messages transmis, tout cela afin de conserver l'énergie des capteurs et éviter les conséquences catastrophiques d'une panne dans le réseau de capteurs sans fil.

L'exécution de notre algorithme passe par cinq étapes, qui sont l'initialisation, la détection, l'information des robots, la réparation, la reprise après panne. Dans chaque étapes, plusieurs techniques ont été utilisées, dont certaines ont été inspirées de quelques solutions trouvées dans la littérature et d'autres sont nouvellement introduites, telles que: la redirection des robots en utilisant les listes des directions permises et les listes des messages d'urgence, la gestion des pannes urgente, et la structuration suivant le rayon de communication des robots.

Chapitre 4:

Evaluation des performances du protocole DRFL.

1. Introduction

Pour évaluer les performances d'une solution de traitement des pannes isolées dans le réseau de capteurs et de robots, plusieurs méthodes de modélisation et simulation peuvent être appliquées, mais à cause de la complexité de notre système et de notre besoin d'améliorer quelques métriques on a choisi la simulation comme une méthode d'évaluation. La simulation informatique¹ ou numérique désigne l'exécution d'un programme informatique sur un ordinateur ou réseau en vue de simuler un phénomène physique réel et complexe.

Dans ce chapitre, nous présentons les résultats de l'évaluation des performances de DRFL, où on a choisi la simulation comme une méthode d'évaluation et le logiciel de simulation réseaux OMNET (OPEN SIMULATOR) comme outil. Ce logiciel est un simulateur d'événements discrets, il possède une architecture modulaire où on a pu implémenter une partie de notre solution et effectuer les calculs nécessaires. Ce logiciel est accompagné d'une gamme importante de plateformes qui peuvent être appliquées dans les différents projets comme: INET, MIXIM, etc. Ceux là peuvent être adaptés au réseau de capteurs sans et portent les différents protocoles que nous avons utilisé dans nos implémentations.

Pour se rendre compte des performances de notre solution relativement à celles connues dans la littérature, nous avons réalisé des comparaisons avec un travail représentatif, qui admet des performances connues notamment en temps moyen de réparation; il s'agit de la solution Grid R3S2(Xu. Li et *al* 2013).

Ce Chapitre est structuré comme suit: Il commence par la présentation de l'outil de simulation utilisé. Par la suite, une description des différents paramètres et mesures de simulation utilisés a été effectuée. La partie principale de ce Chapitre étant la présentation des différents résultats des mesures réalisées ainsi que leurs interprétations a été ensuite explicitée. Une discussion sur certains aspects liés aux performances de notre solution en est suivie. Le Chapitre est clôturé par une conclusion.

2. Outils de simulation

2.1 Le simulateur Omnet

Omnet est un simulateur modulaire orienté objet des réseaux un simulateur d'événements discrets, il possède une architecture générique et peut être utilisé dans divers domaines et problématiques comme :

- la modélisation des réseaux de communication filaires et sans fil,
- la modélisation de protocoles,
- la modélisation de réseaux de files d'attente,
- la modélisation de multiprocesseurs et d'autres systèmes et matériel distribués,
- la validation des architectures matérielles.

Les objets dans OMNET ++ sont structurés sous formes de modules qui communiquent entre eux par le passage de messages. Ces modules sont écrits en C++ et utilisent les bibliothèques de class de simulation. La figure 4.1 représente l'architecture modulaire qui représente les modules simples et les modules composés (compound modules). Dans nos différentes simulations, nous avons utilisé, OMNET 4.3.1.

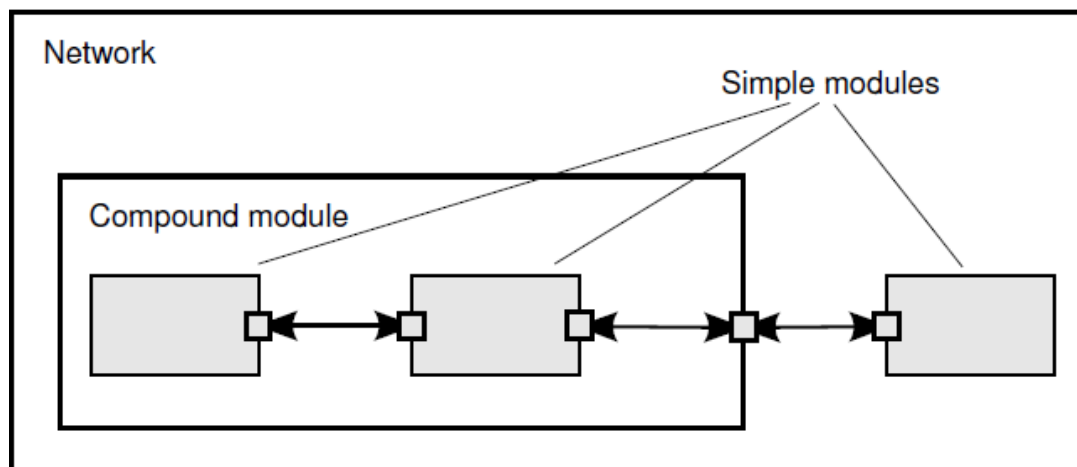


Figure 4.1: Simple and compound modules¹.

Une comparaison entre les performances des simulateurs OMNET, NS2 et NS3 a été effectuée par (Attar Rehman Khan et *al.* 2013), les résultats obtenus en terme de temps d'exécution et d'utilisation d'espace mémoire montrent qu'OMNET engendre des valeurs moyennement acceptables par rapport aux autres simulateurs.

¹ www.omnetpp.org/doc/omnetpp/manual/usman.html . 1/11/2014

2.2 INET 2.4.01

INET est une plateforme de librairie construite sur OMNET, elle utilise le aussi le concept de modules, qui communiquent entre eux par le passage des messages. Des dispositifs comme les ordinateurs, routeurs, Switch, et d'autres composants réseau sont représentés par les modules composés OMNET. Les modules composés sont assemblés par des modules simples qui représentent les protocoles, les applications, et d'autres modules. Les interfaces externes d'un module sont représentées par le fichier NED qui décrit les paramètres et les ports (connecteurs)¹.

INET présente une gamme importante de protocoles comme IPv4, IPv6, TCP, SCTP, UDP, tous ceux là nous servent dans l'implémentation du protocole de routage géographique GPSR¹.

Avec la plateforme INET, on peut utiliser les protocoles de la couche physique implémentés dans d'autres plateformes, comme MIXIM, soit par l'utilisation des modèles implémentés dans la plateforme MIXINET² ou par l'utilisation de la dernière version (MIXIM 2.3) afin d'implémenter les caractéristiques radios spécifiques aux réseaux de capteurs sans fil. MIXIM est une plateforme qui offre les modèles détaillés de la propagation d'un signal radio, l'estimation d'interférence, l'énergie de transmission d'un signal radio, la consommation d'énergie, les protocoles MAC sans fil (e.g. Zigbee) qui est la norme de couche MAC adaptée aux réseaux de capteurs sans fil. Nous avons pu implémenter notre réseau de capteurs et de robots sans d'être contraints de limiter le rayon de communication, ce qui nous a permet de l'augmenter à notre grée¹.

3. Méthodologie de simulation

Pour réaliser la simulation et récupérer ses résultats on suit la méthode suivante :

- Implémenter l'environnement de déploiement, y compris les nœuds et les algorithmes à simuler.
- Pour chaque test effectué, on a défini les paramètres de simulations, et on a exécuté les algorithmes pendant une période fixe afin d'obtenir les résultats. Les résultats obtenus concernent l'impact de la variation de certains paramètres sur les métriques de simulation afin de déterminer les qualités et les points faibles de DRFL par rapport à Grid R3S2(Xu. Li et *al* 2013).

Dans les sections qui suivent immédiatement, on présente les différentes phases de l'implémentation de notre solution, les différents paramètres de notre système et aussi les métriques qu'on a calculées.

¹. <http://inet.omnetpp.org/>, 1/11/ 2014.

². <http://mixim.sourceforge.net/> 1/11/2014.

3.1 Implémentation de la méthode

L'implémentation de l'environnement de déploiement, des nœuds et les différentes étapes de l'exécution des algorithmes sont une partie importante dans le processus de simulation. L'implémentation d'un environnement hétérogène qui possède deux types de nœuds où chaque type possède un comportement différent rend la mission d'implémentation un peut difficile.

3.1.1 Implémentation des nœuds

Afin de réaliser les différents aspects de notre solution, on a défini une architecture modulaire compatible avec chaque type de nœuds (capteurs, robots) ; en respectant les différences entre ces deux types dans la phase d'initialisation, dans la nature du mouvement, et dans le mécanisme de transmission des paquets (protocole de routage) ainsi que dans la couche application.

a) Capteurs

Dans notre environnement de déploiement, des capteurs statiques sont distribués d'une manière aléatoire, à condition que leur distribution lors de l'initialisation assure la couverture et la connectivité du réseau. Le mode de transmission des messages diffère selon la situation du capteur: Si un capteur détecte une panne ou transmet un rapport de panne, il utilise le protocole de routage géographique GPSR. Par contre, il utilise la diffusion quand il s'agit de la communication avec les robots.

L'architecture des capteurs est l'extension d'une des architectures existantes au niveau d'INET qui est celle des nœuds « router GPSR », ce qui nous a amené à modifier l'implémentation du module de mobilité, d'application, de routage ; de plus, on a ajouté un module « dist » (ce module est chargé de l'adaptation des points de la grille virtuelle à l'espace d'adressage utilisé dans le protocole de routage. La Figure 4.2 présente le modèle d'un capteur. Ci-dessous, on présente une explication des caractéristiques de chaque module.

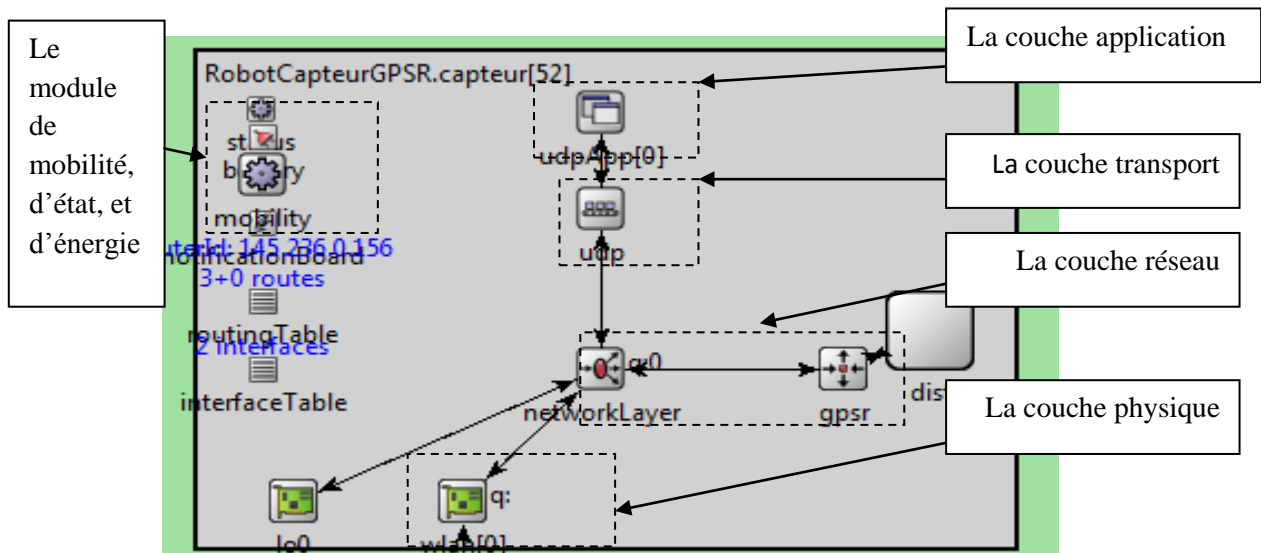


Figure 4.2: Architecture d'un capteur.

- **La couche application:** Comme le montre la Figure 4.2, le module UDPApp [0] représente cette couche dans le modèle des nœuds, qu'on a modifié afin de pouvoir réaliser les communications spécifiques de notre application puisqu'il effectue la majorité des communications par l'envoi et la réception des paquets UDP. En plus, il est responsable de la récupération des informations calculées par le module de mobilité, la préparation des listes des pannes et les mises à jour des listes des directions, la transmission des informations requises vers les robots, et la simulation de la durée de vie de capteur.
- **La couche transport:** On a utilisé le même module que celui des nœuds GPSR.
- **La couche réseau:** La partie intitulée network layer dans la Figure 4.2 est représentée par les deux modules IPV4 network layer et le module GPSR. Mais la version existante de GPSR ne permet la transmission des messages vers une position virtuelle (elle nécessite l'adresse IP du destinataire), i.e. le point de la grille. Par conséquent, on a modifié ce protocole afin de l'adapter à notre besoin.
- **La couche liaison et la couche physique:** Pour cette couche, on a gardé la norme 802.11. Concernant la couche physique, les capteurs utilisent le modèle (le protocole) « IdealWirelessNic».

Concernant les modules de gestion, on a trois modules essentiels : l'énergie, state, et mobilité. Le module de mobilité a été adapté et modifié afin de réaliser: la distribution initiale des capteurs, le calcul des sous zones, l'attribution de rôle aux capteurs et le calcul des directions permises.

b) Les robots

L'architecture d'un robot est différente de celle d'un capteur dans la couche application, la couche réseau, et la gestion de mobilité. La Figure 4.3 montre le modèle des nœuds utilisés. Ce module n'a pas de module de gestion d'énergie ni le module d'état (state).

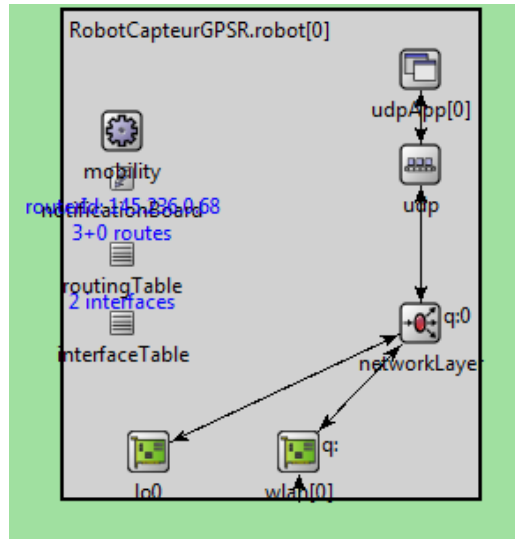


Figure 4.3: Architecture d'un robot.

La figure 4.4 montre les différents modules utilisés. Ci-dessous, on expliquera les caractéristiques de chaque couche, ainsi les caractéristiques des modules de gestion.

- **La couche physique et la couche liaison:** Elles sont les mêmes que celles des capteurs, la seule différence est dans la portée de communication spécifiée dans le module NIC (IDEAL CHANEL NIC).
- **La couche réseau :** Elle est représentée seulement par le module de IPV4 network layer, dont les robots effectuent seulement les diffusions et ne participent pas au routage géographique.
- **La couche application:** Elle gère les communications qui sont effectuées entre les robots et les capteurs, elle ordonne le module de mobilité afin de rediriger le nœud vers la direction désirée.
- **Le module de mobilité:** Selon les collaborations effectuées entre ce module et le module d'application, ce dernier change la position de robots vers une position cible en respectant une certaine vitesse.

3.1.2 Etapes d'exécution de DRFL

Initialement, l'ensemble des capteurs est distribué le long de la zone de déploiement tout en assurant la connectivité et la couverture, et chaque robot est positionné au centre d'une sous zone afin de commencer l'exploration.

La figure 4.4 montre un exemple de distribution initiale des nœuds dans une zone de 600 m sur 600 m. Le rayon de communication d'un robot est égal à 250 m, et le rayon de communication d'un capteur est égal à 50 m. Dans cet exemple, le pas de déplacement d'un robot est fixé à 200 m. Le nombre de points de la grille est calculé de la façon suivante :

$$600/(250-50) * 600/ (250-50)=3*3=9.$$

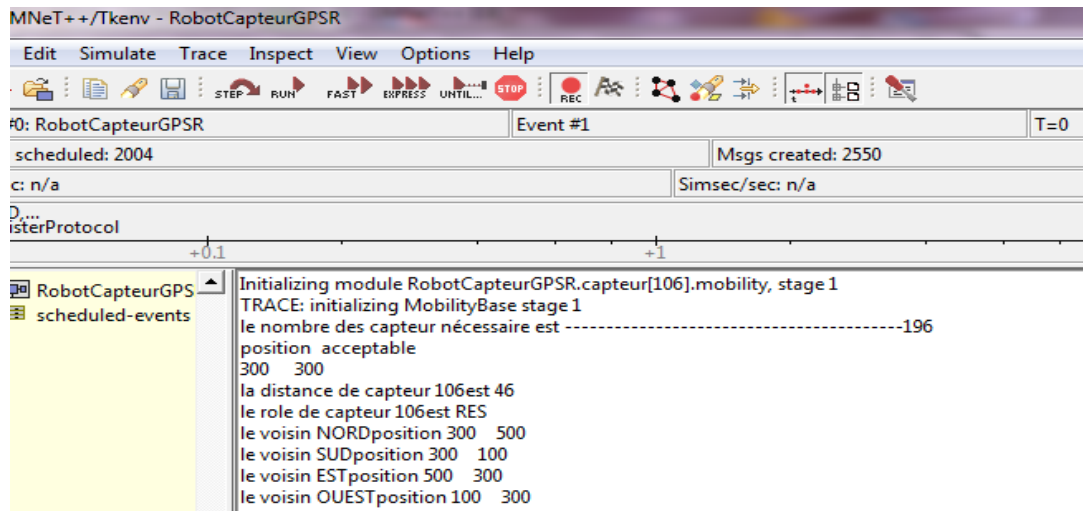


Figure 4.5: Calcul des directions permises par le capteur 106.

3.1.2 Génération des pannes et envois des rapports de pannes

Afin de simuler les pannes des capteurs, chacun d'eux calcule sa durée de vie en utilisant la fonction inverse de la fonction $R(t)$. Selon (Khelladi. L et Badache. N, 2010), $R(t)$ donne la probabilité de ne pas avoir une panne pour un capteur pendant l'intervalle de temps $[0, t]$.

$$R(t) = \exp(-\lambda k t)$$

La fonction inverse de cette fonction donne la durée de vie d'un capteur.

$$T = -1/\lambda k * \ln(U)$$

où U est un nombre aléatoire entre $[0, 1]$, λk est le taux de panne du capteur k (dans notre réseau on suppose que tous les capteurs possèdent la même valeur de λ) qu'on a fixé à 0.02. La Figure 4.6 montre l'état du réseau qui n'a pas été réparé, où tous ses capteurs sont en panne. D'après cette Figure, on remarque la gravité des pannes isolées sur un réseau.

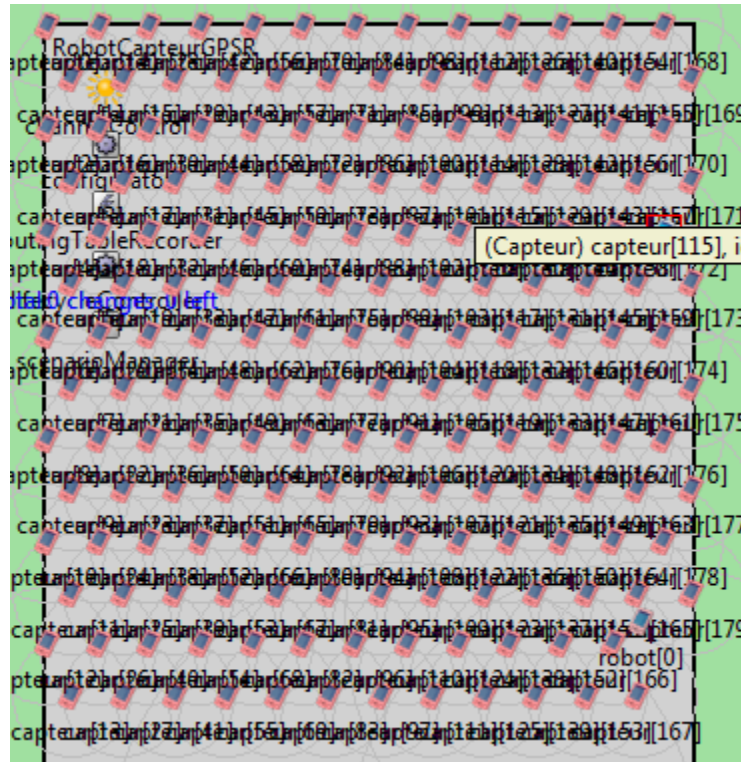


Figure 4.6: Etat du réseau dans T=235.710290203104.

Dans la Figure 4.7, le capteur responsable 124 reçoit un rapport de panne.

```

sending, changing RadioState to TRANSMIT
** Event #45817 T=6.415379567785 RobotCapteurGPSR.capteur[124].udpApp[0] (UDPBasicApp, id=2926), on 'UDPBasicAppData-3' (cPacket, id=54481)
l'adresse ip de point de la grille 145.236.0.189 qui a la position(300, 500, 0)
j'ais un rapport de panne qui a la position389.711 519.615
cette existe
1 101.127. 700 9 042772. 770
    
```

Figure 4.7: Réception d'un rapport de panne.

3.1.3 L'information du robot et la mise à jour des listes

Quand un robot visite une sous zone, il diffuse son message de signalisation. Quand les capteurs responsables reçoivent ce message, ils dégradent la direction de cette sous zone à la queue de leurs listes des directions. La figure 4.8 montre le nouvel état des listes des capteurs 124 et 176 qui se positionnent dans des sous zones différentes. Le capteur 124 déduit que la position du robot est dans la sous zone EST et le capteur 176 déduit que le robot se trouve dans la sous zone NORD.

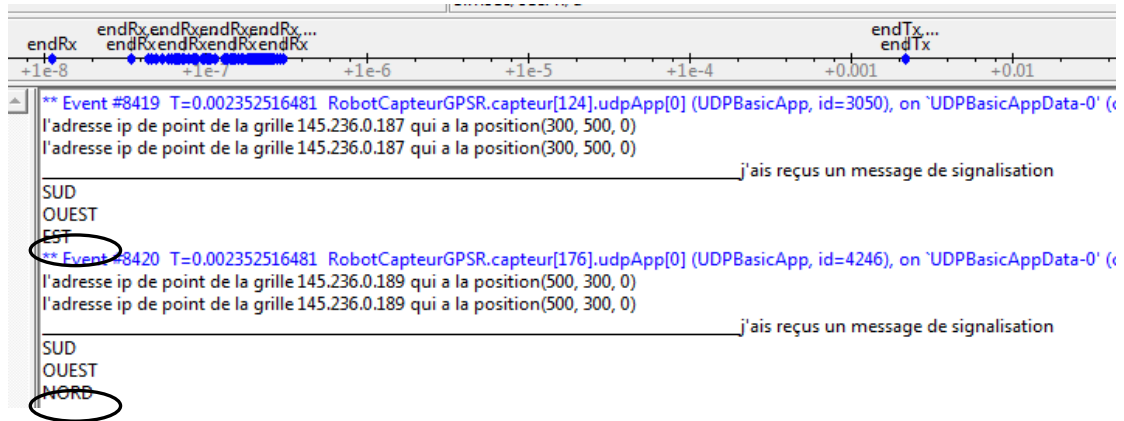


Figure 4.8: Mises à jour des listes des directions.

3.2 Paramètres de simulation utilisés

Dans le processus d'évaluation d'une application, on fixe les différentes valeurs entrant dans le processus de simulation (i.e. paramètres) pour chaque expérience, qui sont de deux types: fixes et variables. Dans nos expériences, on a étudié l'impact de la variation de nombre des robots et de rayon de communication des robots (RCR) sur les qualités de notre algorithme. La Table 4.1, montre les différents paramètres utilisés et leurs valeurs.

Un autre paramètre très important utilisé est le nombre de pannes générés dans une période. Ce paramètre dépend du nombre de capteurs et de la probabilité de la panne d'un capteur.

| | |
|--|----------------------------------|
| Dimension de la surface | 600 m X 600 m ou 1000 m X 1000 m |
| Nombre des robots | De 1 jusqu'à 8 |
| Nombre des capteurs | 180 |
| RCR | De 50 jusqu'à 400 |
| Rayon de communication des capteurs | Egale à 50 ou 25 |

| | |
|---------------------------------|--|
| Rayon de couverture (rc) | Inférieur à 25m |
| Couche MAC | IEEE 802.11 |
| Couche radio | IdealWirelessNic |
| Canal de transmission | WirelessChannel |
| Chanel control | IdealChannelModel |
| Network Configurator | IPv4NetworkConfigurator avec adresse sous réseau address='145.236.x.x' et un mask réseau address='255.255.x.x' |

Table 4.1: Paramètres de simulations.

3.3. Métriques de performance

Pour éviter les conséquences catastrophiques d'une panne isolée, le délai de réparation doit être le plus petit possible, et la conservation de l'énergie des capteurs doit être considérée. Par conséquent, parmi les métriques qui doivent être considérées dans un tel type de solution, on cite: le temps de réparation moyen d'une panne, le temps de réparation maximale d'une panne, le nombre d'arrêts d'un robot, le nombre de messages supplémentaires, l'énergie consommée par les capteurs. Pour le moment, on est intéressé à calculer uniquement trois métriques : le temps de réparation moyen pour chaque panne, le nombre de points de la grille et le nombre de messages supplémentaires.

3.3.1 Nombre des points de la grille pour une zone de déploiement

Au niveau de chaque sous zone (chaque point de la grille ou point arrêt), le robot s'arrête et communique avec les capteurs responsables, pour que le robot arrive à une sous zone qui contient une panne, il peut faire beaucoup d'arrêts, alors dans le but de minimiser le temps de réparation moyen et le nombre de communications entre les robots et les capteurs, on cherche à minimiser le nombre de points de la grille dans le but de minimiser le nombre d'arrêts effectués par un robot pour découvrir les pannes. Alors, le nombre de points de la grille (i.e. nombre de sous zones) est une métrique importante, elle possède une influence directe sur le temps de réparation moyen et le nombre de messages transmis par les capteurs. Cette métrique dépend directement du RCR (rayon de communication d'un robot) et du RCC (rayon de communication d'un capteur)

Selon DRFL, le nombre de points de la grille est calculé par la formule suivant :

Nombre point = *nombre de points selon*(X) × *nombre de points selon* (Y).

Nombre des points selon $X = \frac{\text{taille de la zone selon } X}{(RCR - RCC)}$,

Nombre de points selon $Y = \frac{\text{taille de la zone selon } Y}{(RCR - RCC)}$,

Remarque : Cette division peut donner un nombre réel. Si le chiffre après la virgule de ce nombre est supérieur à 5, le nombre de points prend la valeur du nombre immédiatement supérieur à la partie entière de cette valeur. Sinon, il prend la valeur du nombre entier de la valeur calculée. Exemple : (2.5 → 3; 2,4 → 2)

où RCC est le rayon de communication d'un capteur et RCR est le rayon de communication d'un robot.

3.3.2 Temps de réparation moyen

Dans DRFL, on a à profité de la portée de communication importante qui peut être générée par le signal d'un robot (l'énergie de transmission du signal de robot peut être augmentée sans aucune limite à condition que cette valeur ne dépasse pas les capacités du canal de transmission) afin de rediriger les robots vers les positions des pannes le plus rapidement possible. Pour cette raison, on a évalué le temps de réparation moyen lorsqu'on augmente le rayon de communication des robots (RCR). On a aussi étudié le taux d'amélioration du temps de réparation par rapport au nombre de robots.

Le calcul du temps moyen de réparation est effectué lors de simulation de la réparation :

- Le chemin du robot, lors de réparation, est une boucle qui commence du centre d'une sous zone et arrive au même point. Selon la liste des pannes et la distance qui sépare un point d'un autre, le temps de réparation de chaque panne est calculé.
- Pendant une période de simulation, et selon le temps de réparation de chaque panne, une moyenne est calculée par la formule suivante:

$$T = \frac{\sum_{i=1}^{\text{Nombre panne}} \text{Temps de réparation de panne } i}{\text{nombre de pannes}}$$

Le nombre de pannes dépend de la fonction de calcul de la durée de vie. Pour les pannes qui n'ont pas été réparées dans cette période, on a fixé le temps de réparation de ces dernières à une valeur donnée.

3.3.3 Nombre de messages supplémentaires

Quand les capteurs échangent des messages, ils épuisent leurs énergies ce qui influe sur le nombre de pannes. Afin de minimiser l'effet de la consommation d'énergie des

capteurs, on a calculé pour un intervalle de temps, le nombre de messages générés par chaque panne en utilisant la formule suivante :

Nombre des messages supplémentaires

$$= \sum_{i=0}^{\text{nombre panne}-1} \text{nombre de messages générés par une panne } i$$

4. Résultats et interprétations

Dans les évaluations effectuées, on a concentré nos efforts sur la minimisation du temps de réparation moyen afin d'étudier l'impact de la minimisation du nombre de points de la grille (le nombre d'arrêts), et la redirection des robots vers les directions les moins récemment visitées. En raison du temps, on a étudié seulement les effets de la variation du nombre de robots et le rayon de communication d'un capteur sur le temps de réparation moyen, ainsi que sur le nombre de messages supplémentaires générés pour chaque panne. Ci-dessous, on présente les différents tests qu'on a effectué et les différents résultats obtenus. Dans le reste du présent Chapitre, on note le rayon de communication d'un robot par RCR.

4.1 Impact du rayon de communication d'un robot sur le nombre de points de la grille

Le but de ce test est de se rendre compte des variations du nombre de points de la grille en fonction du rayon de communication d'un robot. La surface choisie du réseau est de 1000 m sur 1000 m, avec un rayon de communication d'un capteur qui est égale à 25 m. Cette mesure est réalisée en comparaison avec le protocole Grid R3S2. Dans la Figure 4.9, on présente le graphe qui montre les résultats de ce test.

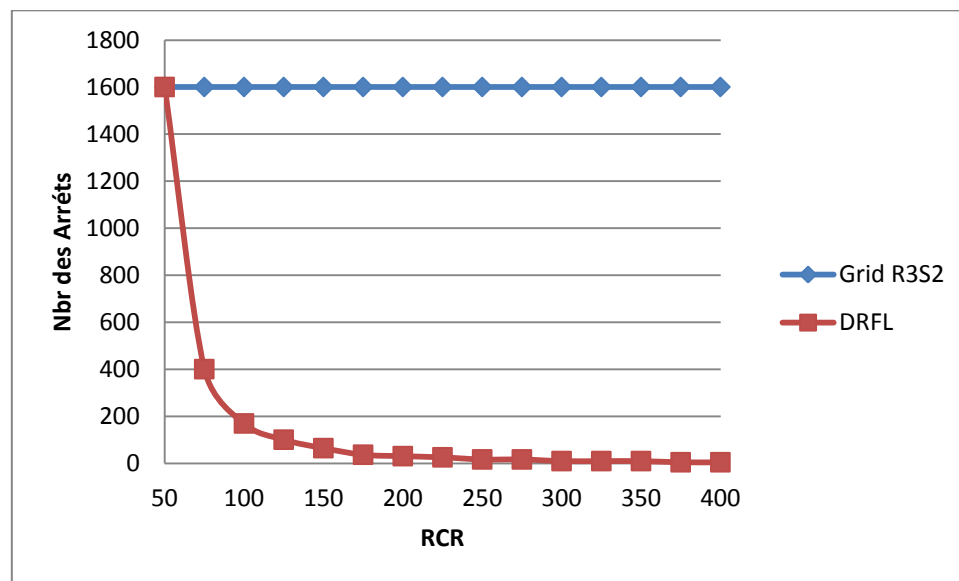


Figure 4.9: Impact de RCR sur le nombre de points dans la grille.

Selon le graphe représenté dans la Figure 4.9, on observe que le nombre de points de la grille dans l'algorithme DRFL diminue à chaque fois que le rayon de communication d'un robot augmente, contrairement à l'algorithme Grid R3S2 qui possède un nombre stable et très élevé de points. Cette stabilité est due au pas de déplacement de robot dans cette approche qui dépend du rayon de communication des capteurs qui est fixe et égale à 25 m. Mais pour DRFL, on remarque une certaine stabilité du nombre de points de la grille pour des valeurs grandes de RCR, qui convergent vers 1. L'explication est évidente, pour des valeurs grandes du pas de déplacement, la surface de la zone (1000m x 1000m) ne peut être divisée. Dans cette situation, le traitement des pannes dans l'algorithme DRFL ne sera plus distribué, ce qui lui donne un caractère centralisé. Dans ce cas, on peut dire que l'augmentation de la portée de communication d'un robot minimise le nombre de points de la grille, mais on doit toujours limiter la valeur de RCR afin d'assurer le caractère distribué de notre solution.

4.2 Impact de RCR sur le temps de réparation moyen d'une panne

Dans une zone de 600 m sur 600 m, avec un nombre de robots égale à trois et un rayon de communication de capteur égale à 50, on a mesuré pour chaque valeur de RCR le temps de réparation moyen en fonction du rayon de communication d'un robot (en commençant par la valeur 100 m). Pour chaque valeur de RCR, on calcule le temps de réparation moyen d'une panne après une certaine période d'exécution du protocole. La figure 4.10 montre les résultats obtenus pour chaque test.

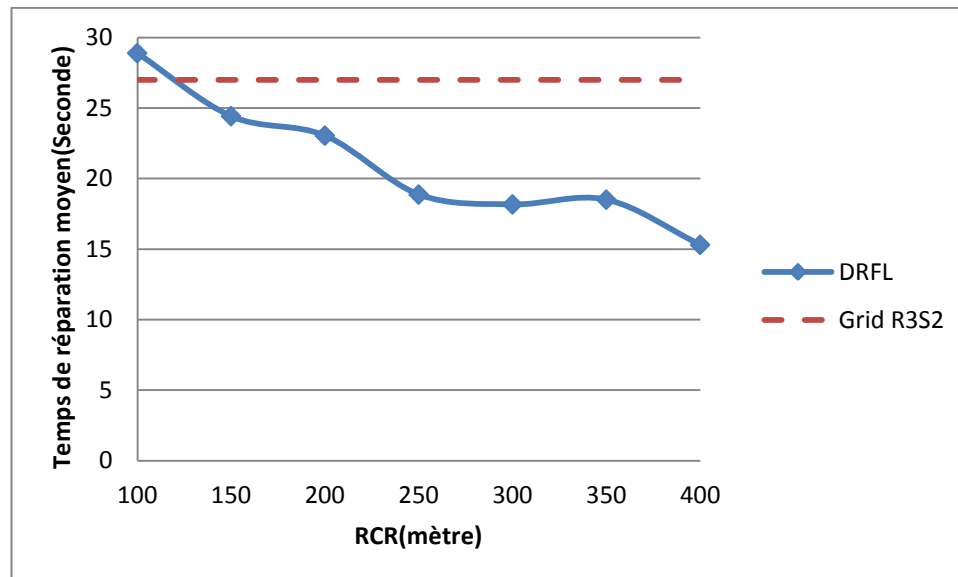


Figure 4.10: Impact de RCR sur le temps de réparation.

Dans la Figure 4.10, on remarque une diminution du temps de réparation moyen avec l'augmentation de RCR. Par contre, ce temps est stable à 28,1358717 s pour l'algorithme Grid R3S2. Dans DRFL, lorsqu'un robot visite une sous zone, il

sera informé d'un pourcentage important de pannes de l'ensemble des pannes qui peuvent intervenir dans la zone de déploiement, ce qui accélère le temps d'information de robot des pannes. Ceci qui permet un temps minimal de réparation de l'ensemble des pannes ; par contre, la taille de la sous zone visitée par un robot selon l'algorithme R3S2 est petite.

4.3 Impact du nombre de robots sur le temps de réparation moyen

En apparence, l'utilisation de multiples robots (plus de 1) peut améliorer le temps de réparation moyen d'une panne. Afin d'étudier le taux d'amélioration du temps de réparation en fonction du nombre de robots, on utilise un réseau de 600 m X 600 m avec un rayon de communication de robot $RCR=250$ m. Le temps moyen de réparation est calculé pour chaque valeur du nombre de robots. Les résultats de l'évaluation obtenue pour ce test sont représentés dans la Figure 4.11.

D'après les résultats représentés dans la Figure 4.11, on remarque que le temps de réparation dans les deux algorithmes s'améliore quand on ajoute plus de robot, l'amélioration dans l'algorithme DRFL est meilleure par rapport Grid R3S2. En effet, avec le même nombre de robots, la réparation dans DRFL est plus rapide. Cela est dû au mécanisme de redirection et à la minimisation du nombre d'arrêts effectués par les robots. L'avantage de DRFL est qu'avec un nombre petit de robots il peut donner un temps de réparation minimal, ce qui minimise les coups de déploiement du réseau WSRN en termes de nombre de robots qui sont généralement très coûteux.

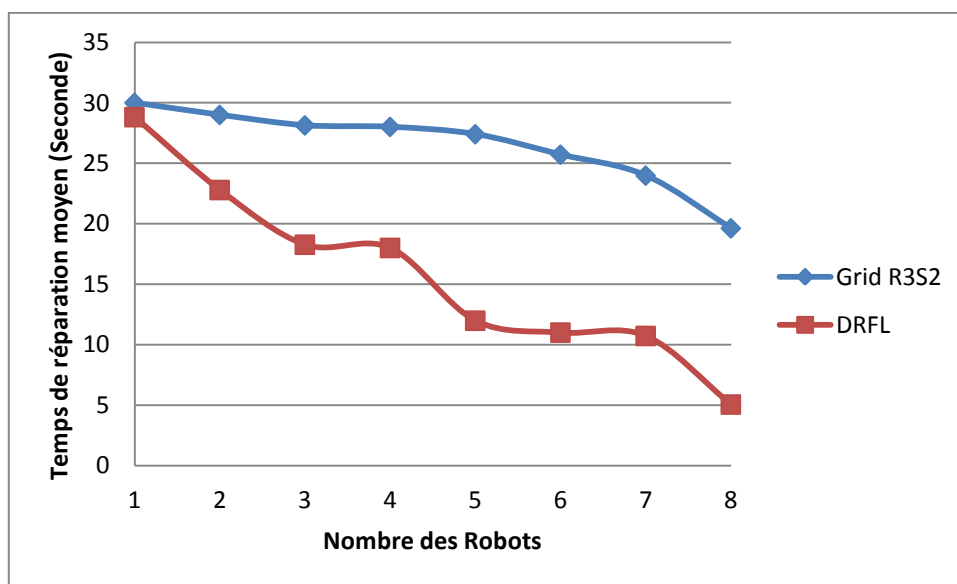


Figure 4.11: Impact du nombre de robots sur le temps de réparation moyen.

4.4 Messages supplémentaires

On remarque à travers les différentes simulations concernant DRFL qu'une partie importante de pannes est générée lors des détections des pannes. Il semble que la

source de ce phénomène est le nombre de messages de diffusions entre les capteurs responsables pour qu'ils puissent construire la même liste de pannes. Ces diffusions dépendent directement du nombre de pannes. Dans la Figure 4.12, on présente les résultats obtenus concernant le nombre de messages générés en fonction du nombre de pannes.

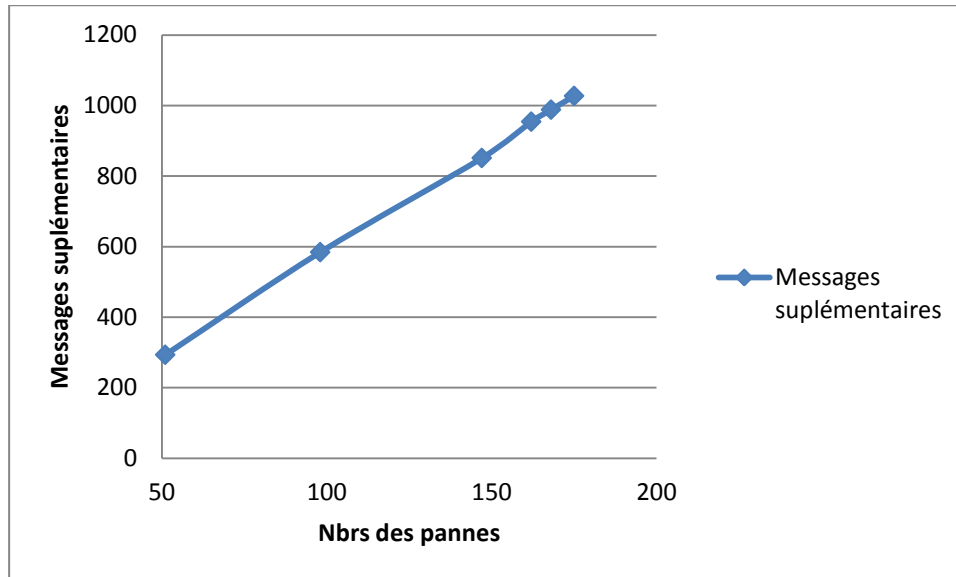


Figure 4.12: Messages supplémentaires.

Dans ce graphe, on remarque une augmentation importante du nombre de messages avec l'augmentation du nombre de pannes. Cela est dû aux diffusions locales générées aux niveaux des capteurs responsables ce qui peut épuiser l'énergie de ces derniers. Alors, il faut réfléchir à d'autres stratégies de construction de la liste des pannes dans le but d'éliminer ou de réduire ces diffusions locales.

5. Discussion

D'après les résultats obtenus, on a remarqué que pour DRFL, l'augmentation du nombre de robots et du rayon de communication d'un robot ont une influence positive sur le temps de réparation qui est plus important que celui de Grid R3S2. Donc, les robots dans DRFL traitent rapidement les pannes. Mais, pour une certaine limite, l'augmentation du rayon de communication des robots peut influencer négativement les qualités de DRFL. L'augmentation du nombre de robots a aussi ses effets positifs, mais il n'est pas nécessaire dans DRFL, d'augmenter ce nombre au point d'égaliser le nombre de points de la grille. En échange, si on regarde le côté financier, DRFL avec un nombre petit de robots traite les pannes plus rapidement relativement à Grid R3S2.

Grid R3S2 est connu être parmi les méthodes de traitement de pannes qui génère le moins de messages. Cette minimisation est due à son caractère localisé. On

a gardé les mêmes principes de détections locales, et on a utilisé le routage des données pour informer les robots des pannes, le problème dans ce cas c'est qu'on transforme une solution à caractère localisé (Grid R3S2) vers une solution à caractère distribué (DRFL), ce qui influe sur le nombre de messages transmis par les capteurs.

Afin de se rapprocher des qualités de Grid R3S2, en termes du nombre de messages, on compte réviser la méthode de détection locale, pour éliminer les diffusions aux niveaux des capteurs responsables, et définir le pas de déplacement des robots qui nous permet une réparation rapide et un nombre de messages réduit.

6. Conclusion

L'évaluation de l'algorithme DRFL a montré quelques points positifs et quelques points négatifs de cet algorithme. La stratégie d'informations des robots appliquée permet une rapidité remarquable dans le temps de réparation des pannes. En échange, nous avons remarqué que le nombre de messages échangés augmente avec l'augmentation du nombre de pannes.

Les résultats obtenus nous encouragent à poursuivre l'évaluation de notre algorithme afin d'étudier l'impact d'autres paramètres comme la résistance au passage à l'échelle, et à réaliser d'autres métriques, comme la consommation d'énergie. Tout cela, pour nous permettre de fixer le pas optimal de déplacement du robot pour un traitement rapide des pannes et un cout minimisé en nombre de messages.

Conclusion générale

Le travail de recherche que nous avons entrepris dès l'état de l'art jusqu'à l'évaluation des performances, nous a permis de faire un premier pas dans le domaine de traitement des pannes dans le réseau de capteurs utilisant les robots. Ce travail nous a permis de découvrir la diversité et la richesse de ce domaine, ce qui nous a permis finalement d'enrichir nos connaissances dans divers domaines. Malgré les efforts qu'on a consentis, il s'avère que beaucoup d'aspects restent à découvrir et à explorer.

En profitant des expériences d'un nombre important d'auteurs qui ont traité la même problématique, nous avons pu faire une contribution quant à la conception d'une nouvelle solution de détection et réparation de pannes isolées dans les réseaux de capteurs moyennant l'utilisation de robots mobiles. Cette solution inspire certains de ses mécanismes des algorithmes connus dans la littérature mais aussi propose de nouveaux autres et tout cela dans le but d'améliorer les performances de ces solutions. Ces performances concernent notamment l'accélération du processus d'information des robots des pannes existantes, ce qui améliore le temps de réparation moyen d'une panne.

Le protocole conçu adopte une approche qui structure la zone d'intérêt en une grille et permet aux capteurs de détecter les pannes des capteurs à proximité et d'informer et d'orienter vers ces pannes les robots en nombre déterminé qui circulent de manière permanente dans la région d'intérêt. L'avantage en est l'économie d'énergie des robots et aussi des capteurs grâce aux transmissions limitées d'informations. Un certain nombre de mécanismes ont été utilisés dans le but d'optimiser les déplacements des robots et les communications dans le réseau. Nous citons : Structure de la zone en grille, utilisation du routage géographique, affectation de différents rôles aux capteurs, réparation de pannes dans l'ordre d'apparitions, introduction des listes de pannes prioritaires (urgentes), introduction des listes de directions des robots pour attirer les robots vers les zones de pannes, mécanismes favorisant la minimisation du nombre d'arrêts de robots pour diminuer les communications, etc.

Les différents tests primaires d'évaluation des performances de notre solution ont montré l'apport de nos différents mécanismes. Notamment, en ce qui concerne la minimisation du temps de réparation par rapport à l'algorithme Grid R3S2. En échange on a remarqué que le mécanisme de détection local et de transmissions des rapports de pannes génère un nombre de messages moyennement important. Ceci montre la nécessité de la révision de ces mécanismes.

On compte dans les futurs travaux améliorer l'algorithme DRFL et faire d'autres tests de simulations avec plus de paramètres et de mesures d'évaluations, afin de rattraper les faiblesses de cet algorithme. On cherche aussi à trouver le pas de

déplacement idéal qui permet à DRFL de minimiser le nombre de messages et de réparer les pannes le plus tôt possible. Ainsi, on compte aussi évaluer la stratégie de gestion des messages d'urgence.

On projette aussi d'adapter notre solution pour résister aux contraintes d'un environnement réel. Par exemple, tenir compte du cas de pannes de robots, considérer un réservoir de capacité limitée, le canal de transmission non fiable, le cas des capteurs mobiles, le cas de pannes massives, l'environnement à trois dimensions, tenir compte des obstacles, etc. On peut aussi penser à une approche sans coopération entre les capteurs et les robots, ce qui élimine une grande partie de charge sur les capteurs. Ceci nécessite une coopération étroite entre les robots pour traiter les pannes.

Bibliographie

Antoine Gallais, Ingelrest Jean Carle, David Simplot-Ryl, "*Maintien de la couverture de surface dans les réseaux de capteurs avec une couche physique non idéale*", CFIP, Colloque Francophone sur l'Ingénierie des Protocoles. 2006.

Akyildiz I.F., W. Su, Y. Sankarasubramaniam, E. Cayirci, "*Wireless sensor networks: a survey*", 2001 Broadband and Wireless Networking Laboratory, School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta GA 30332, Computer Networks, 2002, vol 38, pp 393–422

Attaur Rehman Khana, Sardar M. Bilalb, Mazliza Othmana, "*A Performance Comparison of Network Simulators for Wireless Networks*", LP 50603, Kuala Lumpur, Malaysia. University Carlos III de Madrid, 30 E-28911 Leganés, Madrid, Spain.

Batalin.M-A and G. S. Sukhatme, "*Efficient exploration without localization*", Proceedings IEEE Int. Conf. Robot. Autom. (ICRA), Taipei, Taiwan, Sep. 2003. pp 2714-2719.

Bautina.A, O. Simonina, F. Charpilleta, "*Stratégie d'exploration multi-robot fondée sur les champs de potentiels artificiels*", publié dans la Journées Francophones sur les Systèmes Multi-Agents (JFSMA), 2011.

Carle.J, "*Couverture de zone de surveillance dans les réseaux de capteurs*", presentation of Master Recherche Informatique. Option informatique mobile 2008/2009.

Challal.Y, "*Réseau de capteurs sans fils*" document de centre national de la recherche scientifique, Université de Technologie Compiègne, 17 nov 2008. 70 pages

Corke.P, S. Hrabar, R. Peterson, D. Rus, S. Saripalli, G. Sukhatme, "*Deployment and Connectivity Repair of a Sensor Net with a Flying Robot*", Springer Tracts in Advanced Robotics Vol 21, 2006, pp 333-343

Chih-Yung Chang, Jang-Ping Sheu, *Senior Member, IEEE*, Yu-Chieh Chen, and Sheng-Wen Chang, "*An Obstacle-Free and Power-Efficient Deployment Algorithm for Wireless Sensor Networks*", proceedings of IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS—PART A: SYSTEMS AND HUMANS, Vol. 39, NO. 4, JULY 2009. pp. 795-806.

David Martins, "*Sécurité dans les réseaux de capteurs sans fil Stéganographie et réseaux de confiance*", thèse Doctorat en informatique, L'U.F.R des sciences et technique de l'université de Franche-Comté, 2010.

Falcon Martinez.R-J, "*Towards Fault Reactiveness in Wireless Sensor Networks with Mobile Carrier Robots*", theses of doctoral in School of Electrical Engineering and Computer Science. Faculty of Engineering University of Ottawa, Ottawa, Canada, 2012.

Falcon.R, Xu Li, Amiya Nayak and Ivan Stojmenovic, *The One-Commodity Traveling Salesman Problem with Selective, Pickup and Delivery: an Ant Colony Approach*, IEEE Congress on Evolutionary Computation (CEC), 2010, Date of Conference:18-23 July 2010. pp. 1 – 8.

Fletcher.G, "*Sensor Placement and Relocation by a Robot Team*", thesis master of Computer Science degree, Ottawa-Carleton Institute for computer science of School of Information Technology and Engineering(SITE), Ontario, Canada K1N 6N5 , 2010.

Gallais.A,"*Ordonnancement d'activité dans les réseaux de capteurs : l'exemple de la couverture de surface*", thèse doctorat en Sciences mathématiques (spécialité informatique). Université des Sciences et Technologie de Lille. France. 2007.

Greg Fletcher, Xu Li, Amiya Nayak, Ivan Stojmenovic, "*Randomized Robot-assisted Relocation of Sensors for Coverage Repair in Wireless Sensor Networks* ", IEEE 72nd Vehicular Technology Conference Fall (VTC 2010-Fall), 2010, pp 1-5.

Hao Wu, Long Cheng, Chengdong Wu, Li Chen, "*Robot Assisted Maintenance Strategy in Wireless Sensor Networks*", Proceeding in the 7th International Conference on Computer Science & Education (ICCSE). Melbourne, Australia, 14-17July, 2012.

Hnin Yu Shwe1, Bong Jae Lee, and Jang-See Lee, Thulasiraman et al ,"*DoS Attack Mining in Sensor Node Replacement*", published in (Eds.): ISPA Workshops, Springer Verlag Berlin Heidelberg LNCS 4743, 2007, pp. 11–19.

Jang-Ping Sheu, Kun-ying hsueh and Po-wen cheng, *Design and Implementation of Mobile Robot for Nodes Replacement in Wireless Sensor Networks*, Journal of information science and engineering No 24, 2008, pp. 393-410

Jennifer Yick, Biswanath Mukherjee, Dipak Ghos al," *Wireless sensor network survey*", Computer Network ,Volume 52, Issue 12, 22 August 2008, Pages 2292–2330.

Jung-Eun Kim, Man-Ki Yoon, Junghee Han, and Chang-Gun Lee, "*Sensor Placement for 3-Coverage with Minimum Separation Requirements*", in proceedings of the 4th IEEE International Conference on Distributed Computing in Sensor Systems, DCOSS Santorini Island, Greece, June 11-14, 2008, pp 266-281.

Kacimi.R,"*Techniques de conservation d'énergie pour les réseaux de capteurs sans fil*". Thèse doctorat en réseaux et télécommunications, Ecole doctorale Mathématiques, Informatique et Télécommunication, Unité de recherche : IRIT, l'Institut National Polytechnique de Toulouse, 2009.

Khelladi. L, Badache. N, *Les réseaux de capteurs : état de l'art*, département de l'informatique, Faculté électronique & informatique Université de science et technologie Bab Ezzouar, Alger, Algérie .N° LSI-TR0304, Février 2004

Marc T. Kouakou, Shinya Yamamoto, Keiichi Yasumoto and Minoru Ito, "*Cost-Efficient Deployment for Full-Coverage and Connectivity in Indoor 3D WSNs*", 1Nara

Institute of Science and Technology,2 Tokyo University of Science Yamaguchi 2010.
16 pages

Maxim A. Batalina and Gaurav S. Sukhatmea, "*Sensor Coverage using Mobile Robots and Stationary Nodes*", In *Proceedings of the SPIE*, Boston, MA, University of Southern California, Robotics Research Lab, Los Angeles, CA 90089, USA. Vol 4868, August 2002, pp. 269-276.

Moad.S, "*Optimisation de la consommation d'énergie dans les réseaux de capteurs sans fil*" Rapport de stage Master Recherche en Informatique, Laboratoire de recherche DYONISOS - IRISA, Université IFSIC-Rennes 1,2008.

Maxim A. Batalin and Gaurav S. Sukhatme ,"*The Analysis of an Efficient Algorithm for Robot Coverage and Exploration based on Sensor Network Deployment*" , Proceedings of IEEE International Conference on Robotics and Automation Barcelona, Spain, 2005. pp. 3478-3485

Mei.Y, Changjiu Xian, Saumitra Das, Y. Charlie Hu *, Yung-Hsiang Lu, "*Sensor replacement using mobile robots School of Electrical and Computer Engineering*", Purdue University, West Lafayette, IN 47907, USA, 27 June 2007, page 1.

Naskath.J, K. G. Srinivasagan, S. Pratheema, "*Coverage Maintenance using Mobile Nodes in Clustered Wireless Sensor Networks*", Proceedings of International Journal of Computer Applications, may 2011.

Robert Szewczyk, Joseph Polastre, Alan Mainwaring, and David Culler," *Lessons from a Sensor Network Expedition* ", EuropeanWorkshop, EWSN 2004, Berlin, Germany, January 19-21, 2004. Proceedings pp. 307-322.

Tilak.S , Nael B. Abu-Ghazaleh , Wendi Heinzelman, "*A Taxonomy of Wireless Micro-Sensor Network Models*",Computer System Research Laboratory, Dept. of CS, Binghamton University. Electrical and Computer Engineering, University of Rochester, Rochester, 2002. 5 page

Pengju Si, Chengdong Wu, Yunzhou Zhang, Zhijia Xia, Long Cheng, *A Hole Detection and Redeployment Strategy in Wireless, Sensor Network*, Journal of Information & Computational Science 8: No 13 ,2011, pp. 2577–2585,

Watteyne.T, "*Proposition et validation formelle d'un protocole MAC temps réel pour réseaux de capteurs linéaires sans fils*", Laboratoire CITI Mémoire de Master Recherche Informatique Spécialité Réseaux, Télécommunications et Services, 2004/2005.

Wendi B. Heinzelman, Amy L. Murphy, Hervaldo S. Carvalho, and Mark A.Perillo, "*Middleware to Support Sensor Network Applications*", IEEE network, January/February 2004, pp 6-13

Xu Li a, Greg Fletcher b, Amiya Nayak b, Ivan Stojmenovic b, "*Randomized carrier-based sensor relocation in wireless sensor and robot networks*", Ad Hoc Networks.. Volume 11, Issue 7, September 2013, PP 1951–1962.

Xu Li, Amiya Nayak and Ivan Stojmenovic ,Rafael Falcon,"*The One-Commodity Traveling Salesman Problem with Selective Pickup and Delivery: an Ant Colony Approach*", published in 2010 IEEE.

Yaser.Y ," *Routage pour la Gestion de l'Energie dans les Réseaux de Capteurs Sans Fil*", Thèse de Doctorat en informatique, Université de Haute Alsace, 08 Jul 2010.

Yao Sun,Chengdong Wu, Yunzhou Zhang, Nan Hu , "*Holes Detection in Wireless Sensor Network and Redeploy Route Planning Based on Unmanned Aerial Vehicle*", in *Journal of Information & Computational Science* , 2012.

Yehuda Elmaliach, Noa Agmon and Gal A. Kaminka, "*Multi-Robot Area Patrol under Frequency Constraints*", proceedings of 2007 IEEE, International Conference on Robotics and Automation Roma, Italy, 10-14 April 2007. pp 385- 390.

Younis.M , S. Lee, " *Topology management techniques for tolerating node failures in wireless sensor networks: A survey*", in Springer-Verlag Berlin Heidelberg, University of Maryland at Baltimore County, Baltimore, USA, 2014.

Yongguo Mei, Changjiu Xian, Saumitra Das, Y. Charlie Hu , Yung-Hsiang Lu, *Sensor replacement using mobile robots School of Electrical and Computer Engineering*, Computer Communications No 30 ,2007, pp. 2615–2626.

Sites web consultés:

<http://inet.omnetpp.org/>,1/11/ 2014.

<http://mixim.sourceforge.net/> 1/11/2014.

www.omnetpp.org/doc/omnetpp/manual/usman.html . 1/11/2014

<http://www.omnetpp.org/component/content/article/9-software/3676-mixim-20-released> , 1/11/2014.

<http://www.balluff.com/balluff/MCA/fr/news/6414.jsp>, 1/11/2014.

http://www.memoireonline.com/04/10/3394/m_Greedy-perimeter-stateless-routing-sur-omnet2.html 2/5/2014.