



HAL
open science

Tolérance aux pannes dans un réseau de capteurs sans fil multi-canal

Samira Chouikhi

► **To cite this version:**

Samira Chouikhi. Tolérance aux pannes dans un réseau de capteurs sans fil multi-canal. Informatique et langage [cs.CL]. Université Paris-Est; Université de la Manouba (Tunisie), 2016. Français. NNT : 2016PESC1031 . tel-01481987

HAL Id: tel-01481987

<https://pastel.hal.science/tel-01481987>

Submitted on 3 Mar 2017

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

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

Thèse en cotutelle

Présentée pour l'obtention du grade de DOCTEUR de

L'Université Paris-Est, France

L'École Nationale des Sciences de l'Informatique (ENSI), Tunisie

par

Samira CHOUIKHI

Tolérance aux pannes dans un réseau de capteurs sans fil multi-canal

Spécialité : Informatique

Soutenue le 02/06/2016 devant un jury composé de :

Président	Pr. Laurent George	(Université Paris-Est)
Rapporteurs	Pr. Paul Muhlethaler	(INRIA)
	Dr. Lila Boukhatem	(Université Paris-Sud)
Examinatrice	Pr. Pascale Minet	(INRIA)
Directeurs de thèse	Pr. Yacine Ghamri-Doudane	(Université La Rochelle)
	Pr. Leila Azouz Saidane	(ENSI)
Co-encadrante	Dr. Inès El Korbi	(ENSI)

Remerciements

C'est avec un grand plaisir que je réserve ces lignes en signe de gratitude et de reconnaissance à tous ceux qui ont contribué de près ou de loin à l'élaboration de ce travail. Ce travail, qui a été réalisé au sein du laboratoire CRISTAL, École Nationale des Sciences de l'Informatique (ENSI), Tunisie et du laboratoire LIGM, Université Paris-Est, France, m'a permis de rencontrer des personnes qui m'ont apporté beaucoup d'aide et de soutien dans mes recherches et que je souhaite aujourd'hui remercier.

Je tiens à remercier en premier lieu Pr. Leila AZOUZ SAIDANE et Pr. Yacine GHAMRI-DOUDANE, mes directeurs de thèse pour leur encouragement, leur disponibilité, leurs idées, leurs conseils et leur sympathie qui m'ont permis de mener à bien cette thèse. Leurs compétences, leur rigueur scientifique et leur clairvoyance m'ont beaucoup appris.

Je remercie tout particulièrement à ma co-encadrante Dr. Inès El Korbi qui m'a soutenu tout au long de cette thèse. Elle a toujours été disponible, à l'écoute de mes nombreuses questions, et s'est toujours intéressée à l'avancée de mes travaux. Les nombreuses discussions que nous avons eues ainsi que ses conseils sont pour beaucoup dans le résultat final de ce travail. Sa capacité d'analyse et son enthousiasme m'ont montré que le monde de la recherche pouvait être un univers passionnant. Enfin, ses nombreuses relectures et corrections de cette thèse ont été très appréciées. Cette thèse lui doit beaucoup.

Mes remerciements les plus sincères vont à Pr. Paul MUHLETHALER, Directeur de recherche à INRIA et Dr. Lila Boukhatem, Maître de conférence à l'Université Paris-Sud, d'avoir accepté d'être mes rapporteurs et d'avoir consacré une partie de leur temps à la lecture de mon manuscrit.

Mes remerciements s'adressent aussi à Pr. Pascale MINET, Chargée de recherche à INRIA et Pr. Laurent GEORGE, Professeur à l'Université Paris-Est Marne-la-Vallée, d'avoir accepté d'examiner ce travail et de participer au jury de cette thèse.

J'exprime aussi ma plus profonde gratitude à Mme Sylvie CACH, Responsable administrative de l'école doctorale MSTIC, Université Paris-EST et Mr. Nabil Morgham, Responsable de troisième cycle à l'ENSI, pour leur sympathie et leur efficacité dans l'organisation et la résolution des problèmes administratifs.

Je remercie également tous les thésards et les autres membres du pôle RAMSIS, laboratoire CRISTAL et laboratoire LIGM, pour la bonne ambiance de travail mais également

pour les nombreux bons moments passés ensemble. Je remercie aussi les membres du laboratoire L3i, Université La Rochelle, qui m'ont accueilli dans leur laboratoire et qui m'ont permis de travailler dans d'aussi bonnes conditions.

Je remercie tous mes ami(e)s et toutes les personnes formidables que j'ai rencontrées et qui m'ont aidé dans la réalisation de ce travail. Merci tous pour votre support et vos encouragements.

Enfin je ne trouverais sans doute pas les mots pour remercier assez les personnes qui me sont les plus chères : les membres de ma famille, pour leur sacrifices, leur patience, et tout ce qu'ils ont fait pour m'apporter le bonheur. Qu'ils sachent à travers ces quelques mots combien je leur suis reconnaissante et combien je sais tout ce que je leur dois.

Résumé

Le développement dans les micros systèmes électromécaniques (MEMS) combiné avec l'émergence des nouvelles technologies de l'information et de la communication a permis l'intégration des fonctionnalités de collecte, de traitement et de transmission des données dans un seul dispositif minuscule qui est le capteur sans fil. En voie de conséquence, les réseaux composés de ces capteurs offrent aujourd'hui une pléiade d'avantages par rapport aux réseaux traditionnels notamment en termes de simplicité et de coût de déploiement. Ceci a conduit au développement d'une gamme très variée d'applications des réseaux de capteurs sans fil dans les domaines de la santé, de l'environnement, de l'industrie, des infrastructures, des activités spatiales, ou encore des activités militaires et dans plusieurs autres domaines. Cependant, de nouveaux défis surgissent des caractéristiques particulières de ces réseaux. En réalité, de nombreuses applications de ces réseaux sont critiques et exigent qu'un fonctionnement correct du réseau soit maintenu le plus longtemps possible. Par contre, les environnements dans lesquels ces réseaux sont déployés rendent la mission de maintien en condition correcte de ces réseaux très compliquée et même parfois impossible ; d'où la nécessité d'intégrer des mécanismes d'auto-correction qui visent à surmonter les problèmes qui puissent surgir dans le réseau sans intervention humaine.

Dans ce contexte, nous avons, dans cette thèse, concentré notre étude sur les techniques et les mécanismes mis en œuvre pour améliorer la propriété de tolérance aux pannes dans les réseaux de capteurs sans fil. Tout d'abord, nous avons proposé des approches centralisées et distribuées pour l'auto-rétablissement de la connectivité et la réallocation des canaux dans un contexte de réseaux de capteurs sans fil reposant sur des communications multi-canal après la panne d'un nœud critique. Après la formulation du problème sous la forme d'un problème d'optimisation multi-objectif, nous avons proposé des algorithmes basés sur des heuristiques de coloration de graphes et d'arbre de Steiner, très connus dans la théorie de graphes pour la résolution de ce type de problèmes. Dans une deuxième partie de cette thèse, nous avons étudié un cas d'application particulier, l'agriculture de précision, et avons proposé une solution distribuée pour le rétablissement du fonctionnement du réseau de capteurs sans fil.

Mots-clé :

Réseaux de capteurs sans fil, tolérance aux pannes, restauration de connectivité, communication multi-canal

Abstract

The development in Micro Electro-Mechanic Systems (MEMS) combined with the emergence of new information and communication technologies allowed the integration of the data sensing, processing and transmission in a single tiny device which is the wireless sensor. Consequently, the networks formed by these sensors offer a lot advantages compared with the traditional networks, in particular in terms of the deployment simplicity and cost. This led to the development of a wide range of Wireless Sensor Networks' applications in the domains of health, environment, industry, infrastructures, spatial activities, or even military activities and in many other domains. However, new challenges appear from the particular characteristics of these networks. In fact, many applications of this type of networks are critical and require that the correct functioning of the network is maintained as long as possible. However, the environments in which these networks are deployed return the mission of network maintenance very complicated or even impossible; hence, the necessity of integrating mechanisms of self-correction which aim to overcome the appeared problems without a human intervention.

In this context, we focused our study on the techniques and mechanisms implemented to improve the property of fault tolerance in the wireless sensor networks. First, we proposed centralized and distributed approaches for the connectivity restoration and the channel reallocation in a multi-channel communication context after the failure of a critical node. After the formulation of the problem as a multi-objective optimization problem, we proposed some algorithms based on the heuristics of graphs coloring and Steiner tree, very known in the graph theory to solve this type of problems. In a second part in this thesis, we studied a particular application case, precision agriculture, and we proposed a distributed solution for the failure recovery in wireless sensor networks.

Keywords :

Wireless sensor networks, fault tolerance, connectivity restoration, multi-channel communications.

Table des matières

Table des figures	x
Liste des tableaux	xii
Introduction générale	1
1 État de l’art de la tolérance aux pannes dans les réseaux de capteurs sans fil	6
1.1 Introduction	6
1.2 Concepts fondamentaux de la tolérance aux pannes dans les RCSFs	7
1.2.1 Terminologie	7
1.2.2 Origines des pannes dans les RCSFs	7
1.2.3 Classification des techniques de tolérance aux pannes	10
1.3 Tolérance aux pannes dans les RCSFs à petite échelle	12
1.3.1 Gestion d’énergie et de flux dans les RCSFs à petite échelle	12
1.3.2 Gestion de données dans les RCSFs à petite échelle	15
1.3.3 Couverture et connectivité dans les RCSFs à petite échelle	19
1.4 Réseaux de capteurs sans fil à grande échelle	23
1.4.1 Gestion d’énergie et de flux dans les RCSFs à grande échelle	23
1.4.2 Gestion de données dans les RCSFs à grande échelle	27
1.4.3 Couverture et connectivité dans les RCSF à grande échelle	29
1.5 Conclusion	32
2 Proposition d’approches centralisées pour la restauration de la connec-	

tivité dans les RCSFs multi-canaux	37
2.1 Introduction	37
2.2 Communication multi-canal dans les RCSFs	38
2.2.1 Méthodes d'accès au médium	38
2.2.2 Méthodes d'allocation des canaux	39
2.2.3 Synthèse	42
2.3 Formulation du problème : Tolérance aux pannes dans les RCSFs multi-canaux	42
2.3.1 Modèle du réseau	43
2.3.2 Problème d'allocation des canaux	43
2.3.3 Problème de rétablissement de la connectivité	44
2.4 Résolution exacte du problème d'optimisation multi-objectif	47
2.5 Preventive Failure Recovery : PFR	49
2.5.1 Allocation initiale des canaux	51
2.5.2 Restauration de la connectivité dans le RCSF	53
2.5.3 Réattribution des canaux	56
2.5.4 Synthèse	58
2.6 Rotating Nodes based Failure Recovery : RNFR	58
2.6.1 Initialisation du réseau	59
2.6.2 Détection de la panne d'un nœud d'articulation	59
2.6.3 Rotation des nœuds frontaliers	60
2.6.4 Synthèse	63
2.7 Évaluation des performances	63
2.7.1 Métriques de performance	64
2.7.2 Étude des performances de PFR	64
2.7.3 Étude des performances de RNFR	67
2.7.4 Comparaison des approches PFR, RNFR et RIM	68
2.8 Conclusion	71
3 Proposition d'approches distribuées pour la restauration de la connectivité dans les RCSFs multi-canaux	72

3.1	Introduction	72
3.2	Communication multi-canal dans un contexte distribué	73
3.2.1	Allocation native des canaux	73
3.2.2	Allocation des canaux basée sur le routage	73
3.2.3	Synthèse	74
3.3	Connectivity Restoration for Multi-Channel WSNs : CR-MC	76
3.3.1	Formulation mathématique du problème	76
3.3.2	Phase d'initialisation du réseau	79
3.3.3	Réorganisation du RCSF	79
3.3.4	Réattribution des canaux	80
3.3.5	Synthèse	81
3.4	Connectivity Restoration for Routing based Multi-Channel WSNs : CR-RMC	82
3.4.1	Modèle du réseau	82
3.4.2	Formulation du problème	84
3.4.3	Stratégie de mise en veille	86
3.4.4	Allocation initiale des canaux	88
3.4.5	Réallocation des canaux	90
3.5	Étude de la complexité des approches CR-MC et CR-RMC	91
3.6	Évaluation des performances des approches CR-MC et CR-RMC	93
3.7	Conclusion	98
4	Proposition d'une solution de recouvrement de pannes simultanées dans les RCSFs à grande échelle	99
4.1	Introduction	99
4.2	Applications des RCSFs à grande échelle	100
4.2.1	Les grilles électriques intelligentes "Smart grids"	100
4.2.2	Surveillance sous-marine	101
4.2.3	Surveillance des forêts	101
4.2.4	Agriculture de précision	101
4.3	Proposition d'une stratégie de déploiement d'un RCSF pour l'agriculture de précision	103

4.3.1	Caractéristiques des nœuds déployés	103
4.3.2	Topologie du RCSF	104
4.3.3	Routage et transmission des données	104
4.4	Formulation du problème de recouvrement des pannes pour l'application d'agriculture de précision	106
4.4.1	Modèle du réseau	107
4.4.2	Formulation du problème	107
4.5	Simultaneous Failure Recovery based on Relay Node Relocation : SFR-RNR	109
4.5.1	Détection de pannes et découverte de voisinage	109
4.5.2	Déplacement des nœuds relais	111
4.5.3	Réattribution des canaux	112
4.6	Évaluation des performances de SFR-RNR	113
4.6.1	Paramètres de simulation	113
4.6.2	Étude des performances de SFR-RNR	113
4.7	Conclusion	118
	Conclusion et perspectives	119
	Liste des publications	123
	Bibliographie	125

Table des figures

1	État de l’art de la tolérance aux pannes dans les réseaux de capteurs sans fil	6
1.1	Le diagramme de transition d’états de CCP	20
1.2	Le placement des RNs dans MRP	21
1.3	Exécution du plan de recouvrement dans C^3R	22
1.4	Connexion de trois sommets par un SP	31
2	Proposition d’approches centralisées pour la restauration de la connectivité dans les RCSFs multi-canaux	37
2.1	Recherche des points de Steiner	54
2.2	Réorganisation d’un RCSF après une panne	56
2.3	Topologie d’un RCSF après la panne d’un AP	59
2.4	Recouvrement basé sur la rotation des nœuds	61
2.5	Détermination du seuil énergétique	65
2.6	Performances de la solution centralisée par rapport à la solution optimale .	66
2.7	Le nombre de nœuds recolorés et le nombre de conflits en fonction du nombre de nœuds	67
2.8	Le surcoût de la rotation des nœuds en fonction du nombre total de nœuds	68
2.9	Comparaison des approches PFR, RNFR et RIM	69
2.10	Comparaison de PFR et RNFR en termes de nombre de message et temps de recouvrement	70

3 Proposition d’approches distribuées pour la restauration de la connectivité dans les RCSFs multi-canaux	72
3.1 Configurations des canaux	75
3.2 Formation du polygone pour la détermination des SPs	80
3.3 Exemple des deux nœuds logiques interférents	84
3.4 Graphe des nœuds logiques	84
3.5 Stratégie de mise en veille dans CR-RMC	87
3.6 Coloration du graphe logique dans CR-RMC	90
3.7 Le nombre de liens conflictuels en fonction du nombre de nœuds	94
3.8 Le débit utile en fonction du nombre de nœuds	95
3.9 Nombre de commutations en fonction du nombre de nœuds	95
3.10 Le nombre de nœuds déplacés en fonction du nombre de nœuds	96
3.11 Le nombre de nœuds recolorés en fonction du nombre de nœuds	96
3.12 La stabilité des routes en fonction du nombre de nœuds	97
3.13 La distance parcourue en fonction du nombre de nœuds	98
4 Proposition d’une solution de recouvrement de pannes simultanées dans les RCSFs à grande échelle	99
4.1 Topologie de déploiement du RCSF	104
4.2 Affectation des canaux pour la communication capteur-RN	106
4.3 Topologie du RCSF après une panne	110
4.4 Topologie du RCSF après le déplacement des RNs	112
4.5 Nombre de RNs déplacés en fonction du nombre de RNs défaillants	114
4.6 Distance totale parcourue en fonction du nombre de RNs en panne	114
4.7 Nombre de RNs en conflit en fonction du nombre de RNs en panne	115
4.8 Couverture de la zone endommagée en fonction du nombre de RNs en panne	116
4.9 Nombre de capteurs réaffectés à un nouveau cluster	116
4.10 Nombre de RNs déplacés en fonction de la surface endommagée	117
4.11 Pourcentage de RNs déplacés par rapport au nombre de RNs fonctionnels .	117

Liste des tableaux

1	État de l’art de la tolérance aux pannes dans les réseaux de capteurs sans fil	6
1.1	Tableau récapitulatif des protocoles de routage dans les RCSFs à petite échelle	15
1.2	Tableau comparatif des protocoles de routage	34
1.3	Mécanismes de tolérance aux pannes pour les RCSFs à petite échelle . . .	35
1.4	Mécanismes de tolérance aux pannes pour les RCSFs à grande échelle . . .	36
2	Proposition d’approches centralisées pour la restauration de la connectivité dans les RCSFs multi-canaux	37
2.1	Tableau comparatif de PFR et de RNFR	63
3	Proposition d’approches distribuées pour la restauration de la connectivité dans les RCSFs multi-canaux	72
4	Proposition d’une solution de recouvrement de pannes simultanées dans les RCSFs à grande échelle	99
4.1	Paramètres de simulation	113

Introduction générale

Contexte et motivations

Ces dernières années, l'avancement connu dans les technologies des micro systèmes électromécaniques (MEMS), des communications sans fil et des systèmes électroniques numériques a permis la fabrication de petits nœuds capteurs multifonctionnels qui peuvent être connectés en utilisant une surface radio sans fil. Les capacités de ces nœuds capteurs minuscules en termes de capture, de traitement et de communication permettent le déploiement des réseaux de capteurs sans fil (RCSFs) autonomes[1]. Ces réseaux représentent un nouveau paradigme pour extraire les données de différents environnements, ce qui a permis le développement d'une vaste gamme d'applications comme le contrôle de l'environnement, la recherche biomédicale, l'e-Health, l'agriculture de précision, les activités militaires, *etc.* Pour de telles applications, les RCSFs permettent d'assurer diverses tâches telles que la surveillance, la détection d'événements et la collecte d'informations. Ces tâches ne peuvent être assurées correctement que si les RCSFs font preuve de robustesse, de tolérance aux pannes, d'autonomie d'énergie,

Or, la ressource énergétique limitée d'un nœud capteur, ainsi que les environnements hostiles et dangereux dans lesquels les RCSFs sont déployés, augmentent la probabilité de pannes et rendent la mission de maintien du bon fonctionnement du réseau plus difficile. En plus, pour certains environnements, le remplacement d'un nœud défaillant ou de sa batterie est rarement envisageable voire impossible. Par ailleurs, le déploiement dense des nœuds capteurs fait surgir d'autres problèmes tels que les interférences produites par les transmissions simultanées des nœuds voisins. Tous les problèmes cités précédemment représentent de vrais challenges que les RCSFs doivent relever. Pour pallier ces manquements, plusieurs techniques et mécanismes ont été proposés pour tenir compte des limites des RCSFs et améliorer leur autonomie, leur fiabilité et leur degré de tolérance aux pannes.

Considérant la tolérance aux pannes, plusieurs mécanismes, proposés dans la littérature, se sont penchés sur la résolution des problèmes tels que la pénurie de l'énergie,

les pannes des nœuds capteurs, *etc.* Néanmoins, ces solutions peuvent ne pas prendre en considération tous les paramètres réels du réseau à savoir les interférences entre les nœuds capteurs ou entre les composants coexistants, le bruit et les fautes.

C'est dans ce contexte que se situe notre travail de thèse de doctorat où nous nous intéressons à la proposition de nouvelles solutions de tolérance aux pannes, dans les RCSFs, qui tiennent compte de facteurs particuliers dans lesquels peuvent évoluer ces réseaux à savoir les interférences. En présence d'interférences dans les RCSFs, des solutions basées sur une communication multi-canal peuvent être préconisées. Ainsi, l'objectif de notre travail est de proposer des mécanismes de recouvrement des pannes dans un contexte multi-canal.

Contributions

Dans le cadre de cette thèse, nous avons utilisé le développement rapide de la technologie capteur sans fil qui permet aux capteurs d'intégrer des fonctionnalités évoluées (mobilité, communication multi-canal, ...). Avec ces fonctionnalités, les RCSFs deviennent plus à même de surmonter les problèmes auxquels ils sont confrontés. L'objectif de notre travail est de mettre en place une solution dédiée à la tolérance aux pannes dans les RCSFs qui peut :

- Réduire le taux d'interférences en utilisant une communication multi-canal.
- Recouvrir les pannes pour rétablir le fonctionnement normal du réseau.
- Réduire l'impact des pannes sur le reste du réseau.
- Respecter les caractéristiques des nœuds capteurs.

Ainsi, les contributions de cette thèse de doctorat peuvent être scindées en trois contributions décrites comme suit :

- **Contribution 1 : Proposition d'une solution centralisée pour le recouvrement d'une panne** : Afin de proposer de nouvelles solutions de tolérance aux pannes qui tiennent compte du contexte particulier d'un déploiement multi-canal, nous avons effectué une étude exhaustive des solutions de tolérance aux pannes dans les RCSFs. Ainsi, nous nous sommes penchés sur le problème de recouvrement d'une panne d'un nœud critique (nœud d'articulation) dans un contexte multi-canal. Cette panne mène à la division du réseau en plusieurs segments disjointes. La découverte de la panne et le processus de recouvrement sont entièrement exécutés au niveau du nœud puits qui en informe par la suite les nœuds concernés. Nous avons d'abord commencé par formuler notre problème de recouvrement

de pannes dans les RCSFs multi-canaux sous forme d'un problème d'optimisation multi-objectif. Ensuite, nous avons proposé des heuristiques centralisées pour la résolution de ce problème.

Comme notre solution est centralisée et le réseau est partitionné en plusieurs segments disjoints, il fallait trouver un moyen pour communiquer les informations de recouvrement de pannes aux segments isolés. Ainsi, nous avons proposé deux approches de recouvrement de pannes : une approche préventive PFR (*Preventive Failure Recovery*), exécutée avant l'occurrence de la panne, et une approche curative appelée RNFR (*Rotating Nodes based Failure Recovery*).

- **Contribution 2 : Proposition d'une solution distribuée pour la restauration de la connectivité dans les RCSFs** : Afin de pallier les insuffisances de la solution centralisée, nous avons opté pour une solution distribuée qui peut recouvrir une panne survenue dans un RCSF multi-canal d'une manière distribuée. Nous avons d'abord formulé le problème sous forme d'un problème d'optimisation multi-objectif. Ensuite, nous avons proposé deux approches distribuées pour la résolution du problème de recouvrement de pannes.

La première approche CR-MC (*Connectivity Restoration for Multi-Channel WSNs*) est une approche basique qui utilise les mêmes mécanismes que ceux utilisés dans la première contribution. Par contre, ces mécanismes sont exécutés d'une manière distribuée par les nœuds capteurs. La distribution des traitements nous a permis de surmonter le problème de partition du réseau en segments isolés.

Dans la deuxième approche CR-RMC (*Connectivity Restoration for Routing based Multi-Channel WSNs*), nous avons utilisé les arbres de routage pour mieux exploiter les performances de la communication multi-canal. Nous avons ainsi introduit la notion de nœud logique qui représente l'ensemble des nœuds capteurs ayant le même parent dans l'arbre de routage. Ceci a permis de réduire considérablement les interférences. L'exploitation des arbres de routage nous a permis aussi de minimiser les commutations radios, d'un canal à un autre, le long des routes vers le nœud puits pendant la transmission des données. Ceci réduit le délai de transmission de bout en bout et l'énergie consommée pendant les commutations radios entre les canaux.

- **Contribution 3 : Recouvrement de pannes simultanées dans un RCSF à grande échelle pour une application d'agriculture de précision** : Nous avons proposé un scénario de déploiement d'un RCSF à grande échelle pour le contrôle des cultures dans les champs. Dans ce scénario, nous avons utilisé deux types de nœuds : les nœuds capteurs qui collectent les informations et les nœuds

relais qui transmettent les données aux nœuds puits. Les nœuds relais utilisent deux interfaces radios : une interface pour communiquer avec les nœuds capteurs, et une interface pour communiquer entre eux. Le RCSF est déployé à grande échelle, c'est à dire que des milliers de nœuds capteurs et de nœuds relais sont déployés pour couvrir les champs agricoles. Les nœuds relais sont déployés selon un pavage hexagonal ce qui nous a permis de diviser la zone contrôlée en plusieurs régions. En plus du scénario de déploiement initial, nous avons intégré un mécanisme d'auto-réorganisation du réseau en cas de pannes simultanées d'un grand nombre de nœuds relais et de nœuds capteurs dans une même zone du réseau.

Organisation du rapport

Le présent rapport sera organisé comme suit :

- Dans le **premier chapitre**, nous présenterons les concepts fondamentaux de la tolérance aux pannes dans les réseaux de capteurs sans fil et l'état de l'art des techniques et mécanismes de tolérance aux pannes que nous classerons selon plusieurs critères. Dans la deuxième partie de ce chapitre, nous détaillerons ces mécanismes en révélant leurs avantages et leurs inconvénients par rapport à la tolérance aux pannes dans les RCSFs.
- Le **deuxième chapitre** sera consacré à notre première contribution. Tout d'abord, nous étudierons les techniques de la communication multi-canal proposées dans la littérature. Nous présenterons par la suite notre solution centralisée pour rétablir la connectivité suite à la panne d'un nœud critique dans les RCSFs multi-canaux. Nous formulerons le problème de restauration de la connectivité sous forme d'un problème d'optimisation multi-objectif. Ensuite, nous introduirons l'approche centralisée préventive PFR (*Preventive Failure Recovery*) qui propose des mécanismes pour la communication multi-canal et la réorganisation du réseau en se basant sur la théorie des graphes. Nous utiliserons des heuristiques de coloration de graphes pour l'allocation des canaux et une heuristique d'arbre de Steiner pour le recouvrement de pannes. Nous présenterons aussi l'approche curative RNFR (*Rotating Nodes based Failure Recovery*) qui définit une stratégie de déplacement en cascade pour transmettre l'information de recouvrement aux parties isolées du réseau. Enfin, nous évaluerons ces deux approches par simulation en les comparant entre elles et par rapport à la solution exacte et à une solution existante.
- Le **troisième chapitre** fera l'objet de notre deuxième contribution qui s'intéresse au recouvrement distribué de pannes dans les RCSFs. Tout d'abord, nous

étudierons les mécanismes de la communication multi-canal proposés dans un contexte distribué. Ensuite, nous proposerons les deux approches distribuées CR-MC (*Connectivity Restoration for Multi-Channel WSNs*) et CR-RMC (*Connectivity Restoration for Routing based Multi-Channel WSNs*) qui recouvrent la panne d'un nœud d'articulation dans un RCSF multi-canal. Pour chaque approche nous formulerons d'abord le problème de recouvrement de pannes sous forme d'un problème d'optimisation multi-objectif. Ensuite, nous proposerons des heuristiques pour la résolution de ces problèmes. Enfin, nous évaluerons les performances du CR-MC et du CR-RMC en les comparant à celles de la solution centralisée.

- Dans le **quatrième chapitre**, nous présenterons d'abord quelques applications des RCSFs à grande échelle. Ensuite, nous définirons un scénario de déploiement à grande échelle d'un RCSF multi-canal pour une application d'agriculture de précision. Après, nous proposerons une solution de tolérance aux pannes afin de recouvrir les pannes simultanées d'un grand nombre de nœuds relais et de nœuds capteurs. La dernière partie de ce chapitre sera consacrée à l'évaluation, par simulation, des performances de notre solution de recouvrement de pannes.
- Enfin, nous conclurons notre travail et proposerons des perspectives de recherche relatives à la tolérance aux pannes dans les RCSFs.

Chapitre 1

État de l'art de la tolérance aux pannes dans les réseaux de capteurs sans fil

1.1 Introduction

L'expansion des réseaux de capteurs sans fil n'a cessé de s'accroître ces dernières décennies. Ceci a permis de généraliser leur utilisation dans différents domaines applicatifs (médicales, environnementales, militaires, *etc.*). Malgré la diversité de ces applications et leurs exigences (nature du trafic, contraintes de débit, de délai,...), elles requièrent toutes que le réseau sous-jacent soit suffisamment robuste et fiable pour assurer les fonctions de base d'un RCSF à savoir la couverture et la connectivité. Il en résulte que quelle que soit l'application choisie, le RCSF en question est tenu d'assurer un degré de tolérance aux pannes selon la criticité de l'application afin de garantir son bon fonctionnement. Ainsi, plusieurs solutions et mécanismes ont été proposés dans la littérature pour assurer le recouvrement de pannes dans les RCSFs que nous proposons d'étudier dans le présent chapitre.

Nous entamons notre étude par l'identification de différentes catégories de pannes qui peuvent survenir au niveau d'un RCSF. Ensuite, nous présentons les classifications des techniques de tolérance aux pannes dans les RCSFs selon différents critères. Nous avons veillé tout au long du chapitre à différencier les techniques utilisées en petite échelle de celles utilisées en grande échelle du fait que la qualité de service offerte par le réseau sera différente selon qu'on soit en contexte petite ou grande échelle.

1.2 Concepts fondamentaux de la tolérance aux pannes dans les RCSFs

Avant de parler des mécanismes tolérants aux pannes, il serait préférable de comprendre les notions essentielles en relation avec la tolérance aux pannes.

1.2.1 Terminologie

Pour mieux comprendre la notion de la tolérance aux pannes, nous introduisons la terminologie suivante :

- Une **faute** est toute faiblesse (défaut) inhérente à un système qui mène à une erreur ;
- Une **erreur** est un état système incorrect ou non défini que peut mener à un échec ;
- Un **échec** (ou panne) est une déviation du bon fonctionnement du système ;
- La **tolérance aux pannes** est la capacité d'un système fonctionnel de continuer à exécuter son fonctionnement escompté en présence de fautes ou d'erreurs.
- La **détection des pannes** consiste à la détection de la fonctionnalité défectueuse dans un système par l'auto-diagnostic ou le diagnostic coopératif.
- Le **recouvrement des pannes** est la récupération du fonctionnement correct, après la détection de la faute, en réparant ou en remplaçant le composant responsable de la panne. Dans les RCSFs, les mécanismes de tolérance aux pannes exploitent les composantes redondantes ou dupliquées pour le recouvrement de la panne.

1.2.2 Origines des pannes dans les RCSFs

En général, un RCSF est déployé soit pour *collecter d'informations* d'une zone d'intérêt, soit pour *détecter un événement*. Selon l'application, une perturbation de l'une des tâches du réseau peut affecter la qualité du service livré. Ainsi, nous considérons toute faute ou erreur qui empêche un nœud du RCSF de capturer et/ou transmettre comme étant une panne. Dans cette partie, nous allons classifier les différentes pannes qui peuvent se produire au niveau d'un RCSF, en précisant leurs origines ainsi que l'impact que cela peut avoir sur le réseau. Deux classifications peuvent être appliquées sur les pannes des RCSFs : une première classification de point de vue composant du réseau et une deuxième classification de point de vue système. D'un point de vue composant du réseau, une panne peut se produire en raison des problèmes au niveau du *nœud capteur*, du *nœud puits* ou

du *réseau*; alors que d'un point de vue système, la panne est classée selon la couche où elle a eu lieu : *couche matérielle*, *couche logicielle*, *couche de communication* du réseau ou *couche application*.

1.2.2.1 Pannes de point de vue composant

Pannes du nœud capteur

La panne d'un nœud survient en raison de la défaillance du matériel (unité de capture, CPU, mémoire, interface réseau, batterie, *etc.*) ou du logiciel (routage, MAC et application). De plus, une panne du matériel peut mener à une défaillance logicielle. Par exemple, si l'énergie d'un nœud tombe en-dessous d'un certain seuil, l'unité de capture peut fournir des lectures incorrectes. Cependant, certains services du nœud peuvent toujours continuer à fonctionner correctement. À titre d'exemple, un nœud peut être toujours utilisé pour acheminer des données même si son unité de capture est défaillante. Comme l'épuisement de la ressource énergétique est une panne très fréquente dans les RCSFs, tout mécanisme qui minimise la consommation d'énergie et qui prolonge la durée de vie des nœuds est considéré comme une solution de tolérance aux pannes préventive dans les RCSFs.

Pannes du réseau

Les applications, qui utilisent de RCSFs, exigent la collecte d'informations des nœuds capteurs et la transmission des données vers un ou plusieurs nœuds puits. Cette tâche peut être perturbée par l'occurrence de certaines pannes au niveau des liens entre les nœuds du réseau. Dans les RCSFs, la nature des liens sans fil les rend propices aux erreurs en raison de beaucoup de facteurs comme les interférences entre les nœuds du même réseau ou avec des nœuds de réseaux différents ainsi que les collisions entre les paquets, ce qui mène à la perte des données transmises.

Comme une deuxième source d'erreurs, un mauvais choix des routes par le protocole de routage peut considérablement altérer le bon fonctionnement du réseau. Par exemple, si le trafic passe essentiellement par des routes où les liens sont de très mauvaise qualité alors le taux de perte des paquets sera très important et les performances du réseau seront dégradées. Par ailleurs, le protocole de routage doit respecter les exigences de l'application pendant la sélection des routes. Par exemple, une application de collecte périodique de données tolère le retard généré par la sélection d'une nouvelle route, tandis qu'une application d'alerte exige que les données atteignent le nœud puits dans un temps très court même en présence des pannes. De plus, le fait que les données doivent être toujours envoyées au nœud puits soulève l'importance de la connectivité dans les RCSFs.

Pannes du nœud puits

Le nœud puits est un composant essentiel dans un RCSF. Sa panne mène à la panne du réseau en entier si aucun mécanisme de tolérance aux pannes n'est mis en œuvre. Comme pour un nœud capteur, les erreurs au niveau du puits peuvent survenir aussi bien au niveau du matériel que du logiciel. Cependant, contrairement aux nœuds capteurs, le nœud puits n'a aucune contrainte énergétique.

1.2.2.2 Pannes de point de vue système

Pannes de la couche matérielle

Dans cette couche, les erreurs sont le résultat de la défaillance d'un ou de plusieurs composants du capteur, à savoir la mémoire, la batterie, l'unité de capture et la radio sans fil. Il y a trois raisons de base qui causent la panne : la qualité des composants, l'énergie limitée qui peut aboutir à des lectures incorrectes quand elle chute en-dessous d'un certain seuil ; et finalement, l'environnement hostile dans lequel le réseau est déployé.

Pannes de la couche logicielle

Cette couche est représentée par deux composants : le logiciel du système (le système d'exploitation) et le middleware (incluant communication, routage et agrégation des données). Les bogues sont la source d'erreurs principale dans cette couche. Une solution à envisager est d'implémenter des versions différentes de chaque programme. Par exemple, au niveau middleware, un grand nombre de protocoles sont conçus pour assurer la tolérance aux pannes.

Pannes de la couche communication du réseau

Cette couche est l'une des couches les plus critiques du RCSF en raison des liens de communication sans fil qui peuvent être enclins à la panne. Les fautes peuvent être causées par les environnements dangereux et par les interférences entre les nœuds capteurs eux-mêmes ou les réseaux coexistants. Une méthode pour surmonter ces problèmes et améliorer la performance de la communication est d'utiliser des méthodes de correction d'erreur, des mécanismes de retransmission et de la communication multi-canal.

Pannes de la couche application

Nous ne pouvons pas appliquer les mêmes techniques pour toutes les applications. En effet, chaque application spécifie son propre degré de tolérance aux pannes en se basant sur ses exigences qui peuvent être différentes des exigences des autres applications. La couverture et la connectivité représentent les deux exigences les plus importantes pour les applications. Leur importance diffère d'une application à une autre.

1.2.3 Classification des techniques de tolérance aux pannes

Les techniques de tolérance aux pannes peuvent être classifiées selon plusieurs critères. Les classifications les plus utilisées sont : la classification temporelle, la classification architecturale et la classification selon la taille du réseau.

1.2.3.1 Classification temporelle

Un des critères essentiels de classification est l'instant auquel la procédure de tolérance aux pannes est exécutée (avant ou après l'occurrence de la panne). Ainsi, selon cette classification temporelle, ces techniques sont classifiées comme *préventives* et *curatives*.

Techniques préventives

Cette catégorie inclut toutes les techniques basées sur l'évitement des erreurs dans le réseau. Ces techniques optimisent l'utilisation des ressources disponibles et mettent en œuvre de différentes alternatives de sorte qu'une erreur n'affecte pas la continuité du service. Nous pouvons distinguer deux niveaux d'application des techniques préventives : *niveau nœud* et *niveau réseau*. Au niveau nœud, la technique a pour but de prolonger la durée de vie du nœud. Au niveau réseau, par contre, la technique tolère la panne de certains nœuds, mais essaie de maximiser la durée de vie du réseau et son bon fonctionnement. Parmi les techniques préventives les plus connues, nous citons l'utilisation d'une stratégie de mise en veille, l'équilibrage de consommation d'énergie entre les nœuds et l'ordonnancement des tâches du réseau. De plus, la stratégie de déploiement des nœuds (utilisation des nœuds redondants) peut être aussi classée dans cette catégorie.

Techniques curatives

Les techniques appartenant à cette classe sont déclenchées après l'occurrence d'une panne dans le réseau afin de rétablir le bon fonctionnement du RCSF. Le rétablissement du fonctionnement est effectué en remplaçant des composants défectueux par de nouveaux composants. Cependant, ce remplacement ne signifie pas nécessairement l'ajout de nouveaux éléments dans le réseau, mais il peut être fait par activation des nœuds redondants qui étaient en veille ou par le déplacement de certains nœuds pour garantir la continuité du service.

1.2.3.2 Classification architecturale

La classification architecturale permet de classer les mécanismes de tolérance aux pannes en quatre classes principales :

- *Mécanismes de gestion de l'énergie* : L'objectif principal d'un mécanisme de gestion de l'énergie est d'éviter l'épuisement prématuré de la ressource énergétique et maximiser la durée de vie du réseau. Ces mécanismes se basent sur la minimisation et l'équilibrage de la consommation énergétique.
- *Mécanismes de gestion des flux* : Cette classe inclut tous les mécanismes qui essaient d'établir des routes fiables entre les nœuds capteurs et le collecteur de données (le nœud puits ou la station de base). Ces mécanismes assurent la transmission des données par la sélection de meilleures routes et par le recouvrement de toutes les pannes de routage. La tolérance aux pannes est assurée généralement par le routage multi-chemin et l'utilisation des routes alternatives.
- *Mécanismes de gestion des données* : Ces mécanismes ont pour objectif la réduction de la quantité de données transmises afin d'améliorer les performances du réseau. Ils assurent aussi la disponibilité de l'information en cas de panne de certains composants du RCSF. Cette classe inclut les mécanismes d'agrégation, de compression et de stockage des données.
- *Mécanismes de gestion de la couverture et de la connectivité* : L'occurrence des pannes dans un RCSF peut mener aux trous de couverture ou à la perte de connectivité dans le réseau. Les mécanismes qui traitent les problèmes de déploiement et de placement des nœuds capteurs sont inclus dans cette classe.

1.2.3.3 Classification selon la taille du réseau

Une autre classification des techniques de tolérance aux pannes peut être faite selon la taille du réseau. Ainsi, nous classons les mécanismes de tolérance aux pannes en deux classes : mécanismes pour les RCSFs à *petite échelle* et mécanismes pour les RCSFs à *grande échelle*. Dans les réseaux de petite taille composés d'un petit nombre de nœuds capteurs, chaque nœud a un rôle important dans la capture et la transmission des données. Ainsi, la panne de n'importe quel nœud influence considérablement la fiabilité du RCSF. Cependant, dans les réseaux de grande taille, un grand nombre de nœuds est déployé sur la zone à couvrir. Les RCSFs à grande échelle sont essentiellement conçus pour les applications de collecte de données, pour lesquelles l'intérêt est d'obtenir des informations sur la région contrôlée en tant que zones d'intérêts et non en tant que points d'intérêts. Comme la densité de déploiement est grande et la corrélation spatiale est très forte dans les RCSFs à grande échelle, la panne d'un nœud devient négligeable tant que les nœuds voisins continuent à fonctionner correctement.

Par ailleurs, la taille du réseau (petite ou grande taille) influence considérablement

le choix de la technique de tolérance aux pannes à appliquer (solution centralisée ou distribuée). En effet, une solution centralisée est privilégiée pour les RCSFs de petite taille où le nœud puits prend en charge le déroulement de l'algorithme de restauration et alerte les nœuds capteurs de nouveaux paramètres du réseau. Dans un réseau à grande échelle, par contre, une solution centralisée serait très peu avantageuse compte tenu de la surcharge causée par les messages envoyés afin d'alerter les nœuds capteurs de la nouvelle configuration du réseau. La solution distribuée serait alors privilégiée dans un tel contexte.

Dans les sections suivantes, nous présenterons les différentes solutions de tolérance aux pannes dans les RCSFs à petite et à grande échelle en mettant en évidence les classifications présentées.

1.3 Tolérance aux pannes dans les RCSFs à petite échelle

Les RCSFs à petite échelle sont composés de dizaines de capteurs dispersés sur une petite zone géographique. Ils peuvent être déployés pour différentes applications dans de nombreux domaines à citer la médecine à distance, la surveillance domestique, *etc.* Dans ce type de réseaux, chaque nœud capteur a un rôle très essentiel dans la capture et la transmission des données. Par conséquent, la panne de n'importe quel nœud capteur influence considérablement la fiabilité du réseau.

1.3.1 Gestion d'énergie et de flux dans les RCSFs à petite échelle

Les mécanismes conçus pour la minimisation ou l'équilibrage de la consommation d'énergie sont considérés comme étant des techniques préventives de la tolérance aux pannes. Ainsi, ils peuvent prévenir la panne du nœud ou du réseau. En plus, de nombreux protocoles de routage (par exemple, EAR [2]) conçu pour la minimisation de la consommation d'énergie, intègrent des mécanismes de tolérance aux pannes. En général, les protocoles tolérants aux pannes se basent sur deux techniques : la *retransmission* et la *réplication*. Dans un protocole utilisant la retransmission, un nœud retransmet le paquet sur une route alternative si aucun acquittement (ACK) n'est reçu sur la route principale. L'inconvénient majeur de cette technique est qu'elle exige plus de ressources en termes d'énergie et de mémoire pour sauvegarder le paquet jusqu'à la réception d'un ACK. Dans la technique de réplication, plusieurs copies des mêmes données sont envoyées sur des routes différentes. Bien que cette technique tolère la panne d'un certain nombre de nœuds

ou de routes, elle engendre un surcoût très élevé dû au routage multi-chemin, ce qui augmente considérablement la consommation de l'énergie et de la bande passante. Cependant, l'utilisation des techniques de compression et de codage de données peut surmonter un tel problème.

1.3.1.1 Mécanismes basés sur la retransmission

Beaucoup de protocoles de routage optent pour le choix des routes les plus courtes en terme de nombre de nœuds traversés pour la transmission de données. Cependant, l'utilisation de la même route fréquemment mène à l'épuisement rapide de la ressource énergétique le long de cette route. Pour atténuer ce problème, la solution EAR-LEAHSN (*Energy-Aware Routing for Low Energy Ad Hoc Sensor Networks*) [3] utilise des routes différentes en se basant sur une métrique énergétique.

Cet algorithme diminue la probabilité de pannes des nœuds, causées par l'épuisement prématuré des ressources énergétiques, par l'utilisation des routes différentes vers le nœud puits. De plus, il tolère la panne d'une route en retransmettant les données sur une autre route fonctionnelle. Cependant, EAR-LEAHSN est coûteux en terme d'utilisation de la mémoire puisque chaque nœud doit maintenir une liste de toutes les routes vers la destination et leurs coûts.

Dans [4] et [5], les auteurs ont proposé deux approches permettant la transmission des données au nœud puits en utilisant un mécanisme de sélection rapide d'une nouvelle route après la panne de la route principale. Ainsi, les données peuvent être transmises sur la nouvelle route choisie avec un délai très court.

Dans [6], Hassanein *et al.* ont défini le protocole REAR (*Reliable Energy Aware Routing*) utilisant des mécanismes de conservation d'énergie au niveau de la couche réseau. REAR sélectionne deux chemins : un chemin de service qui est le chemin primaire et un chemin de secours qui va être utilisé en cas de panne du chemin de service.

L'avantage majeur de cet algorithme est que les données peuvent être retransmises immédiatement sur le chemin de secours sans aucun retard supplémentaire dû à la sélection d'un nouveau chemin. Cependant, les ressources réservées au niveau des nœuds capteurs pour la route de secours peuvent n'être jamais utilisées si la route primaire reste fonctionnelle ce qui mène à un gaspillage inutile des ressources.

Le protocole DD (*Directed Diffusion*) [7] était considéré comme le protocole de routage le plus utilisé pour les RCSFs. Il se base sur une description de la valeur d'un attribut pour demander des informations des nœuds capteurs. DD fonctionne en trois phases :

exploration, renforcement positif et maintenance des routes. Dans la première phase, le nœud puits explore tous les chemins possibles avec la source, tandis qu'à la deuxième phase, le chemin avec le délai d'arrivée des paquets le plus court sera sélectionné pour la transmission entre la source et le nœud puits. Enfin, si le nœud puits remarque un problème sur le chemin primaire, il sélectionne un autre chemin ou déclenche la phase d'exploration.

Dans [8], le protocole *Braided Multipath Routing* a été proposé. Ce protocole utilise les mêmes techniques que DD à la différence près qu'en phase de renforcement, deux messages sont transmis au lieu d'un. Le premier message renforce le meilleur voisin et le deuxième message renforce le deuxième meilleur voisin. Ce processus, répété au niveau de chaque nœud, permet de trouver plusieurs chemins partiellement disjoints ce qui permet de se remettre d'une panne rapidement.

1.3.1.2 Mécanismes basés sur la réplication

Deb *et al.* ont proposé le protocole ReInForm (*Reliable Information Forwarding using Multiple Paths in Sensor Networks*) [9], dans lequel le nœud puits émet périodiquement un message de mise à jour pour que chaque nœud puisse déterminer ses voisins et le nombre de sauts vers le nœud puits. Lorsqu'un nœud veut transmettre des données, il ajoute l'état dynamique du paquet (*Dynamic Packet State DPS*) à l'en-tête du message et envoie des copies multiples par des chemins différents. La DPS inclut les informations nécessaires (le nombre de sauts vers le nœud puits, le taux d'erreur du canal local et la fiabilité désirée) pour décider du nombre de copies qui vont être envoyées en se basant sur la fiabilité requise. Chaque nœud intermédiaire utilise les informations de la DPS pour déterminer le nombre de copies à transmettre et choisit les voisins les plus proches du nœud puits. Le processus est répété jusqu'à ce que tous les paquets transmis atteignent le nœud puits. Le principe de ce protocole permet de garantir un degré de fiabilité très élevé et assure la tolérance aux pannes en envoyant plusieurs copies du même paquet au nœud puits. Cependant, ces avantages sont très coûteux en termes de consommation d'énergie et d'utilisation de la bande passante dans le réseau.

Le tableau 1.1 récapitule les principales caractéristiques des protocoles de routage que nous avons présentés.

Protocole	Conservation d'énergie	Min. Messages	Agrégation	Mécanisme de Tolérance aux pannes
EAR-LEAHSN	Oui	Non	Non	Chemin alternatif
REAR	Oui	-	-	Chemin de secours
Directed Diffusion	Non	Oui	Oui	Renforcement d'un autre chemin
Braided Multi-path Routing	Oui	Non	Non	Chemins de secours tressés
ReInForm	Non	Non	Non	Routage multi-chemin

Tableau 1.1 : Tableau récapitulatif des protocoles de routage dans les RCSFs à petite échelle

1.3.2 Gestion de données dans les RCSFs à petite échelle

Dans cette partie, nous discutons les techniques les plus utilisées pour économiser la consommation d'énergie et garantir la disponibilité des données mêmes en cas de pannes. En général, trois techniques sont utilisées : *agrégation* de données, *compression* de données et *stockage* de données.

1.3.2.1 Agrégation de données

Cette technique permet d'obtenir une vue globale du domaine contrôlé au lieu de multiples vues individuelles (c'est à dire l'obtention de chaque valeur du nœud capteur séparément). Ceci réduit la quantité de données transmises par le réseau, et par conséquent l'énergie consommée par une telle transmission.

En général, l'agrégation est combinée avec la clusterisation (clustering) [10] [11][12] [13] pour atténuer la quantité d'informations transmises dans le réseau et le traitement des données. La clusterisation organise les nœuds dans des groupes (clusters) et sélectionne un nœud leader (cluste head) en tant que responsable du traitement et de la transmission des données. Plusieurs algorithmes de clusterisation ont été proposés, à savoir LEACH [14], LEASH-C [15], TEEN [16], APTEEN [17], PEGASIS [18], et TL-LEACH [19]. Certains algorithmes se basent sur un critère énergétique dans la sélection du nœud leader comme HEED [20], EEHC [21], EECS [22], EEUC [23], et MRPUC [24].

En général, les algorithmes d'agrégation utilisent deux techniques : une technique basée sur les arborescences (tree-based) et une technique basée sur le commérage (gossip-based). La première technique organise le réseau sous forme d'un arbre logique où les données se propagent des feuilles vers la racine (le nœud puits). Dans la deuxième technique, le réseau est divisé en petits groupes et les nœuds de chaque groupe communiquent

pour agréger leurs données. Les algorithmes utilisant cette technique convergent après un certain nombre d'itérations lorsque tous les nœuds du réseau ont la même information.

Dans [25], l'algorithme DRG (*Distributed Random Grouping*) a été conçu pour fonctionner en plusieurs tours. A chaque tour, un nœud devient un leader de groupe avec une probabilité p_g ; ses voisins rejoignent le groupe en envoyant leurs valeurs au leader qui calcule la valeur agrégée (SOMME, MOYENNE, MAX ...) et la renvoie à tous les membres du groupe. Les membres mettent à jour leurs valeurs et les remplacent par la nouvelle valeur. Cet algorithme converge quand tous les nœuds ont la même valeur agrégée, ce qui garantit la disponibilité et la cohérence des données dans le réseau en cas de panne de quelques nœuds.

Le protocole *Tiny AGgregation* (TAG) [26] est un exemple représentatif des algorithmes d'agrégation basés sur les arborescences. C'est un service principal intégré dans les capteurs utilisant Tiny OS, ce qui offre une interface utile pour les requêtes déclaratives. En fait, TAG n'a pas inventé un nouveau langage de requêtes, mais il imite la syntaxe des requêtes de style SQL sur une seule table (*sensors*). La procédure d'agrégation est divisée en deux phases : la phase de distribution, dans laquelle des requêtes d'agrégat (*MOY, SOM, COUNT, MÉDIANE, etc*) sont injectées dans le réseau. Un arbre est construit des nœuds capteurs vers le puits afin d'envoyer les données agrégées. La deuxième phase est la collecte de données où les valeurs agrégées sont acheminées de bas en haut jusqu'au nœud racine.

Le problème principal de cette technique est que la panne d'un nœud entraîne la panne d'un sous-arbre entier. Pour surmonter ce problème, quelques solutions sont proposées [26]. La première solution concerne les fils qui choisissent un autre parent quand ils ne reçoivent aucune donnée de leur parent initial. La deuxième solution concerne les parents qui maintiennent une cache pour enregistrer les valeurs partielles de leurs fils, lesquelles valeurs sont utilisées quand les parents ne reçoivent pas d'informations de leurs fils.

L'algorithme *Synopsis Diffusion* [27] utilise une technique de routage multi-chemin basée sur l'économie d'énergie afin de tolérer la panne de quelques nœuds et/ou quelques liens. Dans cet algorithme, les nœuds sont organisés en anneaux autour du nœud puits en se basant sur le nombre de sauts qui sépare les nœuds capteurs du nœud puits. Durant le routage de données, un nœud dans l'anneau R_i fusionne son synopsis (c'est à dire un résumé représentant les informations capturées) avec les synopsis des nœuds de l'anneau R_{i+1} et émet le nouveau synopsis vers l'anneau R_{i-1} . Les synopsis agrégés se propagent vers le nœud puits qui extrait la réponse de la requête.

L'utilisation de multiples chemins dans chaque transmission de synopsis, offre un méca-

nisme de tolérance aux pannes pour recouvrir la panne de quelques nœuds ou de quelques liens. Le problème de ce protocole est comment traiter les agrégats sensibles à la duplication (*SOMME*, *COUNT*, *etc*). Les auteurs ont suggéré l'utilisation des fonctions de hachage pour résoudre ce problème.

Un deuxième algorithme *Ride-Sharing* [28] tolère la perte d'un paquet, en raison d'une erreur du lien, en agrégeant le paquet perdu avec les données d'un autre nœud capteur qui a intercepté une copie du paquet perdu. Contrairement à *Synopsis Diffusion*, chaque valeur est agrégée au maximum une fois, donc il supporte directement les agrégats sensibles à la duplication. Chaque nœud choisit un parent primaire et quelques parents de secours. Les liens primaires forment un arbre de recouvrement pour la transmission des données jusqu'à ce qu'un problème ait lieu dans l'un des liens de communication primaires. Dans ce cas, les parents de secours sont utilisés pour récupérer ce paquet.

1.3.2.2 Compression de données

Contrairement à l'agrégation de données, la compression consiste à réduire de la quantité de données transmises par chaque nœud capteur sans perte d'information. En plus de la réduction de la taille des données transmises, la compression est une technique communément utilisée dans les protocoles de routage basés sur la réplication (voir section 1.3.1.2). Cette technique permet d'envoyer plusieurs copies des mêmes données sans augmenter la consommation des ressources disponibles (la bande passante et l'énergie).

L'une des techniques les plus utilisées liées à la compression de données est le codage DSC (*Distributed Source Coding*) [29] qui utilise la corrélation entre les informations fournies par les capteurs. Les algorithmes basés sur DSC exigent que le nœud puits et tous les nœuds capteurs soient conscients de la corrélation des données pour qu'ils puissent coder et extraire les informations correctement. Ces mécanismes exploitent le fait que les capteurs du même voisinage, dans les RCSFs, fournissent des informations fortement corrélées avec des valeurs redondantes. Par conséquent, si un nœud envoie sa valeur au nœud puits, l'information envoyée par son voisin sera limitée à quelques bits représentatifs. Par la suite, en se basant sur la corrélation entre les deux nœuds, le nœud puits peut déduire la valeur du deuxième nœud. Plusieurs mécanismes, basés sur DSC, ont été proposés et expérimentés pour différentes applications de RCSFs [30],[31].

1.3.2.3 Stockage de données

Le stockage de données est l'un des challenges les plus intéressants dans la gestion de données. Il est très stimulant d'améliorer la persistance et la récupération des données quand quelques nœuds tombent en panne. En général, les techniques de stockage des données sont basées sur des mécanismes de redondance.

Un mécanisme simple mais efficace a été proposé pour le stockage des données dans les RCSFs où tous les nœuds peuvent communiquer directement [32]. L'idée de ce mécanisme est de partager le stockage des données entre tous les nœuds pour que les données venant de n'importe quel capteur puissent être récupérées d'autres capteurs. Les capteurs stockent les données seulement quand le nœud puits est indisponible. Dans le cas contraire, les capteurs envoient leurs données directement au nœud puits et leurs mémoires restent vides.

Pour appliquer ce mécanisme, chaque capteur divise une copie de ses données sur les $(n-1)$ autres capteurs, tel que chaque capteur stocke une partie des données redondantes d'autres capteurs. Si à un moment donné un nœud tombe en panne, les autres nœuds coopèrent pour récupérer toutes les données de leurs mémoires redondantes. Cette récupération coopérative est exécutée seulement si le nœud puits n'est pas disponible. Dans le cas contraire, le nœud puits peut directement accéder à toutes les mémoires des capteurs et récupérer lui-même ces données. Malgré son efficacité, ce mécanisme ne tolère ni les pannes simultanées ni la panne d'un nœud entre le début et la fin du processus de récupération des données d'un autre nœud.

Dans [33], les auteurs ont présenté une nouvelle méthode nommée TinyDSM (Tiny Distributed Shared Memory) qui permet aux capteurs de partager leurs données d'une façon dépendante de l'application. En effet, le TinyDSM est un middleware fournissant une façon d'exécuter un stockage de données fiable et distribué. Le mécanisme fonctionne comme suit : le propriétaire de l'information émet ses données vers le reste du RCSF. Certains nœuds dans son voisinage deviennent des dupicateurs et maintiennent des répliques de données mises à jour par leur propriétaire après une demande du dupicateur. Si une requête, demandant des données spécifiques, est injectée dans le réseau, n'importe quel nœud ayant une réplique de ces données peut répondre et pas seulement le nœud propriétaire. En effet, les données seront toujours disponibles même si quelques nœuds sont en veille ou en panne. De plus, ce mécanisme permet à chaque nœud de prendre la décision de stockage d'une réplique des données indépendamment des autres nœuds. Il se base, alors, sur le nombre de répliques imposé, le nombre de sauts entre les nœuds, et l'espace mémoire encore disponible. Cependant, nous avons remarqué deux points faibles pour ce

mécanisme : le premier est que TinyDSM ne considère pas les contraintes énergétiques d'un nœud capteur ; le deuxième est la possibilité que le nœud propriétaire devienne un goulot d'étranglement si plusieurs dupicateurs demandent la mise à jour des données en même temps.

1.3.3 Couverture et connectivité dans les RCSFs à petite échelle

Dans la littérature, plusieurs algorithmes sont proposés pour garantir la couverture et la connectivité dans les RCSFs. Ces algorithmes sont classés généralement en deux classes principales : algorithmes préventifs et algorithmes curatifs.

1.3.3.1 Gestion préventive de la couverture et de la connectivité

Les algorithmes préventifs dépendent de la stratégie de déploiement et de maintien de la topologie du réseau. Ils permettent le remplacement des nœuds en panne avec d'autres nœuds en activant les nœuds redondants (nœuds en veille) ou en déplaçant quelques nœuds. Les mécanismes proposés dans [34] et [35] représentent deux méthodes utilisées dans cette catégorie d'approches préventives.

Le protocole CCP

Dans [35], le protocole de Configuration de Couverture (*Coverage Configuration Protocol* CCP) est proposé pour maintenir la couverture et la connectivité en utilisant le minimum possible de nœuds. Ce mécanisme active seulement un ensemble de nœuds de sorte que la zone entière soit couverte et que le réseau reste connecté. Les autres nœuds entrent en veille afin de prolonger la durée de vie du réseau. Un nœud peut être alors dans l'un des trois états : EN VEILLE, ACTIF ou EN ÉCOUTE. Les nœuds en veille seront activés si d'autres nœuds tombent en panne en raison de l'épuisement de la ressource énergétique ou de n'importe quel autre facteur. De plus, les nœuds utilisent des timers aléatoires pour éviter que deux voisins se retirent ou s'activent en même temps. La figure 1.1 montre les différentes règles de transition d'états dans CPP, où T_s , T_l , T_j , T_w désignent respectivement le temps de veille, le temps d'écoute, le temps de rejoindre le réseau et le temps de retrait.

CCP assure la connectivité seulement si $R_c \geq 2R_s$, où R_c et R_s représentent respectivement le rayon de communication et le rayon de couverture. Si ce n'est pas le cas, CCP est combiné avec l'algorithme SPAN [36] qui maintient un *backbone de communication* formé par les nœuds actifs.

Le mécanisme Monitoring and Self-Organization

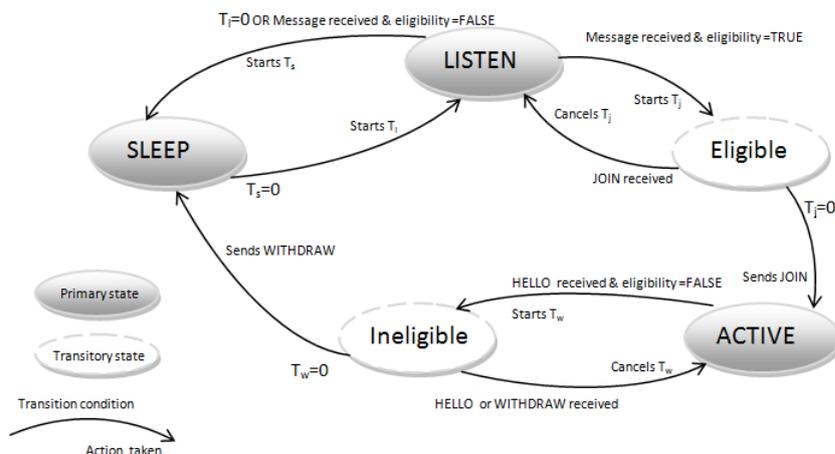


Figure 1.1 : Le diagramme de transition d'états de CCP

Benahmed *et al.* [34] ont proposé un mécanisme de contrôle et d'auto-organisation (*Monitoring and Self-Organization Mechanism*) pour améliorer la connectivité des nœuds critiques appelés points d'articulation APs (*Articulation Points*). En effet, un AP est un nœud capteur dont la défaillance divise le RCSF en des segments déconnectés. L'algorithme s'exécute en deux étapes : dans la première étape, les APs sont détectés ; tandis que dans la deuxième étape, un algorithme est déclenché pour réorganiser le réseau afin d'assurer plus de redondance autour de l'AP en réveillant ses voisins redondants. S'il n'y a aucun nœud redondant, l'auto-organisation recherche des nœuds redondants parmi ses voisins à deux sauts.

Le protocole MRP

En plus de la méthode qui se base sur les stratégies de déploiement des nœuds capteurs, nous pouvons utiliser une méthode de placement des nœuds relais RNs (*Relay Nodes*). Cette méthode a comme objectif le placement d'un ensemble de RNs pour améliorer la fiabilité et la tolérance aux pannes dans le réseau. Plusieurs travaux ont discuté le problème de placement optimal des RNs dans les RCSFs. Ce problème consiste à placer un nombre minimal de RNs pour couvrir un ensemble de capteurs, aléatoirement distribués, de sorte que chaque capteur soit couvert par au moins k RNs pour rendre le réseau k -connecté (2-connecté au moins pour garantir la tolérance aux pannes). Cependant, ce problème a été prouvé pour être NP-complet ; ainsi les solutions proposées sont des approximations de la solution optimale.

Dans [37], Liu *et al.* ont discuté le placement des nœuds relais et ont proposé l'approche *Minimum Relay-node Placement* (MRP) comme solution pour placer un ensemble minimal de nœuds relais en tant que backbone du réseau. Ce backbone assure la transmission des

données sans la participation de n'importe quel nœud capteur. MRP a pour but de rendre le réseau *2-connecté*, ce qui signifie que chaque capteur peut communiquer avec deux RNs et chaque RN communique au moins avec deux autres. Ce mécanisme s'exécute en deux étapes : la première étape est le placement des RNs pour rendre le réseau 1-connecté (*MRP-1*), alors que la deuxième étape consiste à ajouter quelques RNs pour rendre le RCSF 2-connecté (*MRP-2*).

La figure 1.2 nous montre la topologie du réseau durant le déroulement du MRP (les nœuds verts représentent les capteurs alors que les nœuds rouges représentent les RNs). Dans la première topologie, chaque capteur est connecté à un RN. Ces derniers ne sont pas connectés. Après la deuxième étape, le réseau devient *1-connecté*. Dans la dernière étape, MRP-2 garantit la tolérance aux pannes du réseau en ajoutant un ensemble de RNs.

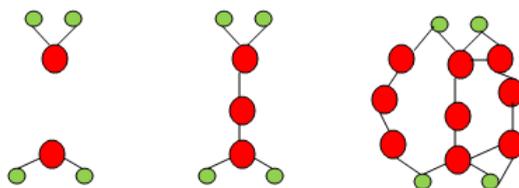


Figure 1.2 : Le placement des RNs dans MRP

1.3.3.2 Gestion curative de la couverture et de la connectivité

Les solutions proposées essaient, d'une manière curative, de réorganiser la topologie du réseau pour rétablir la connectivité et/ou la couverture. Comme pour la gestion préventive des pannes, plusieurs travaux peuvent être cités dans cette catégorie.

L'approche *RIM (Recovery through Inward Motion)* [38] déplace les nœuds voisins du nœud défaillant en direction du nœud en panne jusqu'à ce qu'ils puissent communiquer ensemble. Si à un moment donné un nœud i devient déconnecté d'un voisin j en raison du déplacement de j , le nœud i suit ce voisin j pour y être reconnecté. La procédure de déplacement s'exécute récursivement pour empêcher n'importe quel nœud de se déconnecter de ses voisins. Bien que *RIM* rétablisse la connectivité, cette approche devient très coûteuse en termes de nombre de nœuds déplacés et de distance parcourue par ces nœuds dans les réseaux denses.

L'algorithme *C³R (Coverage Conscious Connectivity Restoration)* [39] coordonne entre les nœuds dans un même voisinage pour qu'ils remplacent temporairement le nœud en

panne, en déplaçant chaque voisin à la position du nœud défaillant pendant une durée bien déterminée. Un coordinateur construit un plan de recouvrement de pannes dans lequel il précise la liste des nœuds participants, l'instant et la durée de remplacement. Le premier nœud dans le plan de recouvrement de la panne occupe la position du nœud défaillant. Après une certaine période, un autre nœud se déplace pour remplacer le nœud défaillant tandis que le premier nœud retourne à sa position initiale en notifiant ses voisins de son retour. Les nœuds se déplacent en va et vient, un à un, de leurs positions initiales à la position du nœud en panne. Si à un moment donné, l'énergie d'un nœud tombe en-dessous d'un certain seuil, le coordinateur actif propose un nouveau plan de recouvrement en excluant ce nœud.

La figure 1.3 montre un exemple de fonctionnement du C^3R dans un RCSF à huit nœuds capteurs, où le nœud rouge représente le nœud en panne ayant quatre voisins avec le même rayon de couverture. Ces voisins exécutent le plan de recouvrement un à un en se déplaçant de leur zone de couverture initiale.

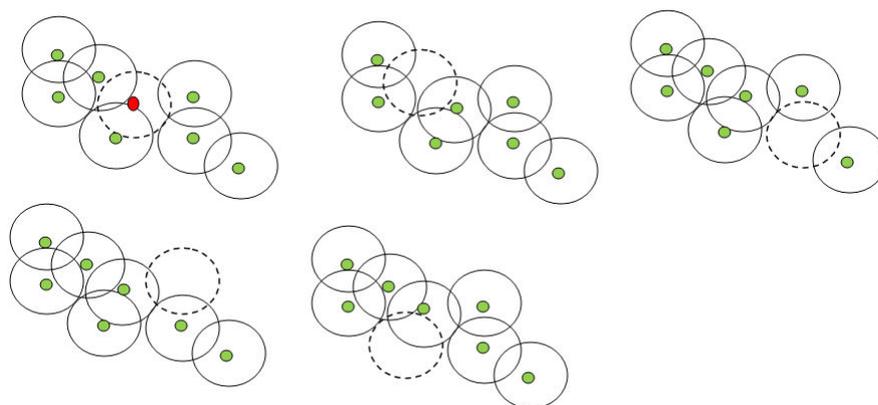


Figure 1.3 : Exécution du plan de recouvrement dans C^3R

Dans [40], les auteurs proposent deux techniques de tolérance aux pannes : la première technique (*migration des nœuds*) permet la restauration de la connectivité en déplaçant quelques nœuds ; tandis que la deuxième technique (*Minimum Movement Technique MMT*) s'exécute pour réduire les trous de couverture causés par la *migration des nœuds*.

Toutes les approches présentées dans cette partie garantissent d'une manière ou d'une autre la tolérance aux pannes dans les RCSFs. Il faut mentionner pourtant que ces approches s'appliquent dans le cas où un seul nœud capteur tombe en panne à la fois. Ainsi, elles ne peuvent pas traiter le cas de pannes simultanées.

Le tableau 1.3 récapitule les avantages et les inconvénients des mécanismes de tolérance aux pannes dans les réseaux de capteurs sans fil à petite échelle.

Après la présentation des travaux relatifs à la tolérance aux pannes dans les RCSF à petite échelle, le reste de ce chapitre sera consacré à l'étude des mécanismes de tolérance aux pannes dans les RCSFs à grande échelle.

1.4 Réseaux de capteurs sans fil à grande échelle

Les RCSFs à grande échelle (exemple d'applications : agriculture de précision, smart grids, surveillance des forêts, *etc.*) sont composés de milliers de nœuds capteurs, dispersés sur une large zone géographique, produisant une quantité très grande d'informations dans le réseau. Il en résulte que la nature des pannes qui peuvent survenir dans un tel contexte ainsi que les solutions qui peuvent être envisagées pour le recouvrement de ces pannes sont complètement différentes de celles proposées dans le cas des RCSFs à petite échelle.

1.4.1 Gestion d'énergie et de flux dans les RCSFs à grande échelle

Les algorithmes de gestion de flux se basent sur le routage par des chemins multiples ou par des chemins de secours. En général, ces algorithmes sont hiérarchiques et utilisent différentes techniques de clusterisation. Plusieurs protocoles sont conçus spécialement pour les RCSFs à grande échelle comme MRMS (*Multi-path Routing in large scale sensor networks with Multiple Sink nodes*) [41], MuMHR (*Multi-path, Multi-hop Hierarchical Routing*) [42], DECROP (*Distributed and Effective Cluster Routing Protocol*) [43], ONCP (*Off-Network Control Processing*) [44] [45] et ASN (*Alternative Sensor Nodes*) [46].

1.4.1.1 Le protocole MRMS

La solution MRMS [41] améliore l'efficacité énergétique en choisissant le chemin avec le coût énergétique le plus faible. Il introduit deux métriques énergétiques pour la sélection du chemin : la première métrique représente le coût du lien entre deux nœuds i et j , tandis que la seconde métrique est le coût du chemin en entier :

$$cost_{ij} = \alpha \times d^2 + \beta \quad (1.1)$$

$$path_cost = \sum cost_{ij} \times E_{RE}^\gamma \quad (1.2)$$

où d est la distance entre les deux nœuds, E_{RE}^γ est l'énergie résiduelle des nœuds

capteurs. L'approche MRMS fonctionne en trois phases : *découverte de la topologie*, *maintenance des clusters* et enfin *commutation des chemins*. La première phase permet de découvrir tous les chemins vers tous les nœuds puits. Ainsi, si une panne au niveau d'un nœud capteur, un CH (*Cluster Head*) ou un nœud puits est détectée, l'algorithme peut sélectionner soit un autre composant pour substituer le rôle du composant en panne, soit un autre chemin.

L'avantage majeur du MRMS est qu'il tolère la panne des CHs, des nœuds capteurs et même des nœuds puits par l'utilisation des chemins multiples vers les différents nœuds puits. Cependant, la sauvegarde de tous les chemins alternatifs nécessite une utilisation importante des ressources du réseau ce qui rend la solution MRMS très coûteuse et peu exploitable.

1.4.1.2 Le protocole ASN

Les auteurs de [46] ont proposé un mécanisme hiérarchique qui organise les nœuds en différentes couches en se basant sur le nombre de sauts vers la station de base BS (*Base Station*). Ce mécanisme fournit aussi la possibilité d'utiliser des nœuds capteurs alternatifs ASNs (*Alternative Sensor Nodes*) durant le routage des données. Un ASN est un nœud dont la distance est inférieure à $\frac{R}{2}$ (R est le rayon de communication) à un nœud appartenant à la route primaire, et son énergie résiduelle est supérieure à l'énergie résiduelle du nœud primaire. Le mécanisme de clusterisation utilisé dans ASN est le même que celui utilisé LEACH [14]. Selon ce mécanisme, chaque nœud choisit un nombre aléatoire entre 0 et 1. Si ce nombre est inférieur à un seuil $T(n)$, le nœud devient CH. Le seuil est calculé comme suit :

$$T(n) = \frac{P}{1 - P * (r \bmod \frac{1}{P})} \text{ si } n \in G, \text{ et } 0 \text{ sinon.} \quad (1.3)$$

où P est le pourcentage de CHs parmi tous les nœuds, r est le numéro de l'itération et G est l'ensemble de nœuds qui n'ont pas été élus CHs pour les derniers $\frac{1}{P}$ itérations. Les autres nœuds rejoignent le cluster ayant le signal le plus élevé en utilisant une communication à un saut. Si un nœud est loin de tous les CHs, il s'élit CH. Au niveau d'un cluster, les nœuds transmettent leurs données au CH en utilisant TDMA (*Time Division Medium Access*).

La route est choisie de telle façon à minimiser le nombre de sauts vers la station de base. Si plusieurs routes sont éligibles, l'algorithme examine pour chaque route i , E_{min}^i , la valeur minimale de l'énergie parmi tous les nœuds le long de i . Ainsi, la route j avec

$E_{min}^j = \max_i(E_{min}^i)$ est choisie. Si un nœud v possède un ASN, alors ce dernier exécute la tâche de routage des données du nœud v .

Le premier avantage d'ASN est l'équilibrage de la consommation d'énergie entre les CHs et les nœuds alternatifs. Le deuxième avantage est l'évitement des collisions par l'utilisation du TDMA au niveau de chaque cluster. De plus, chaque nœud peut envoyer les données immédiatement à un nœud alternatif si le nœud primaire tombe en panne. Cependant, ASN présente une faiblesse majeure en raison de la surcharge produite par la construction des clusters à chaque itération, la découverte de nœuds alternatifs et le remplacement des nœuds primaires.

1.4.1.3 Le protocole MuMHR

La solution *MuMHR* [42] fournit un mécanisme de communication qui réduit la consommation énergétique. Le nœud puits sélectionne 5 % des nœuds comme CHs d'une façon stochastique. Ensuite, le CH forme une liste de CHs de secours en ajoutant les nœuds capteurs ayant les niveaux d'énergie les plus élevés. Quand la durée du cluster actuel expire ou le CH tombe en panne, le premier nœud de la liste de secours substitue le rôle du CH actuel. Les membres de chaque cluster transmettent leurs informations au CH en utilisant TDMA. Le CH agrège toutes les données reçues et envoie le paquet agrégé au nœud puits en utilisant CDMA (*Code Division Multiple Access*) pour éviter les interférences avec les messages des autres groupes.

En se basant sur la description précédente, nous pouvons déduire que MuMHR réalise une économie de la ressource énergétique en multiplexant le trafic et en changeant le rôle du CH à chaque itération. En outre, ce protocole maintient un chemin alternatif et une liste de secours qui peuvent être utilisés en cas de panne.

1.4.1.4 Le protocole ONCP

L'approche ONCP [44] [45] utilise plusieurs nœuds puits connectés à un serveur ONCP par un réseau filaire. Au niveau de la couche application, un certain nombre d'utilisateurs sont connectés au serveur en utilisant une interface utilisateur graphique. La zone contrôlée est pré-divisée en différentes régions, dans lesquelles chaque nœud maintient un identifiant inchangé de la région (ID). Les IDs des régions ainsi que la liste des régions à traverser pour atteindre le puits sont calculés hors ligne au niveau d'un serveur ONCP afin de réduire les messages de contrôle.

Le protocole ONCP est un protocole orienté requête ; c'est à dire un utilisateur peut

obtenir des informations d'une région spécifique A en demandant au serveur ONCP de chercher le chemin global (liste des régions pour atteindre le puits). Ainsi, le serveur choisit la meilleure route minimisant la consommation d'énergie de bout en bout de cette région au nœud puits approprié en se basant sur l'énergie résiduelle de la région et la métrique de connectivité inter-régions. La route globale est alors calculée comme une séquence des régions d'une région source au nœud puits. Ensuite, la liste des régions participantes dans le routage est injectée dans le réseau à partir du nœud puits vers la région A . Une fois la liste arrivée, une route locale est construite pour assurer le routage vers la région suivante. Le chemin local sélectionné est celui qui minimise le nombre de sauts traversés vers le puits et l'énergie consommée.

Nous pouvons conclure qu'ONCP réduit considérablement les messages de contrôle dus à la construction des routes. En outre, il tolère la panne d'un nœud puits en utilisant de multiple puits dans le réseau. Cependant, ce protocole n'a pas mis en œuvre de mécanismes pour le recouvrement d'une panne d'un nœud dans la zone frontalière entre deux régions. De plus, la pré-configuration des IDs des régions dans chaque nœud est une tâche complexe et exige la pré-connaissance de la carte de déploiement.

1.4.1.5 Le protocole DECROP

Chen *et al.* ont proposé DECROP [43], un protocole de routage simple et efficace qui réduit le nombre de messages de contrôle et diminue le délai de bout en bout. Cet algorithme suit trois étapes : *l'initialisation et la clusterisation, la transmission de données* et finalement *la maintenance des routes*. Des clusters sont formés de manière à équilibrer le nombre de nœuds au niveau de chaque cluster. De plus, durant la phase d'initialisation, des routes alternatives sont construites. Ces routes sont utilisées en cas de panne de la route principale.

L'avantage principal de DECROP est que tous les chemins sont construits pendant la phase d'initialisation ce qui réduit le nombre de messages de contrôle. Par ailleurs, le nombre de clusters est réduit en y incluant les voisins à un saut et à deux sauts. Cependant, nous avons relevé deux inconvénients majeurs pour ce mécanisme : la consommation énergétique du CH qui augmente avec l'augmentation de la taille du cluster et la topologie en arbre adoptée pour cette approche qui peut mener à l'épuisement rapide de l'énergie des nœuds les plus proches du nœud puits.

Le tableau 1.2 compare les caractéristiques des protocoles de routage dans les RCSFs à grande échelle.

1.4.2 Gestion de données dans les RCSFs à grande échelle

La gestion de données dans les RCSFs à grande échelle devient une obligation et non un choix en raison de la grande quantité de données échangées ce qui affecte considérablement la durée de vie du réseau.

1.4.2.1 Agrégation de données

Les deux techniques d'agrégation utilisées dans les RCSFs à petite échelle (*tree-based* et *gossip-based*) ne peuvent pas être directement appliquées dans un contexte grande échelle. En effet, le délai de transmission des données et le temps de convergence de ces algorithmes augmentent considérablement avec la taille du réseau. L'idée est de combiner les deux techniques ensemble. Ainsi, les n nœuds du réseau sont rangés dans m groupes ou clusters avec un nœud leader pour chaque groupe [47]. Ensuite, dans chaque cluster, un algorithme *gossip-based* est utilisé pour calculer les valeurs agrégées. Enfin, les nœuds leaders de tous les clusters forment un arbre avec le nœud puits. Les données seront alors agrégées le long de cet arbre par un algorithme *tree-based*. Cependant, le nombre de groupes doit être soigneusement choisi étant donné qu'un petit nombre aboutit à l'augmentation de la taille de chaque groupe (temps de convergence devient plus important); mais réduit le délai le long de l'arbre vers le nœud puits et vice versa. Les avantages des deux algorithmes *tree-based* et *gossip-based* restent valables lorsqu'ils sont appliqués dans le contexte grande échelle.

1.4.2.2 Compression de données

La compression de données réduit considérablement la quantité de données transmises dans le réseau. De ce fait, plusieurs protocoles de compression de données sont proposés dans la littérature. Le mécanisme le plus connu est CDG (*Compressive Data Gathering*) [48] qui réduit le trafic de données en compressant les lectures des capteurs et en équilibrant la consommation d'énergie pour augmenter la durée de vie de réseau. L'idée de base de CDG est que chaque nœud transmet M messages ($M \ll N$, avec N le nombre de capteurs). Le nœud puits extrait les valeurs individuelles de chaque capteur à partir de ces messages. Le protocole exploite l'arbre virtuel formé par les chemins de routage et de ses sous-arbres. Chaque nœud dans le réseau doit connaître son état dans l'arbre de routage, c'est à dire si c'est une feuille ou un nœud intérieur de l'arbre. De plus, si c'est un nœud intérieur, il doit connaître le nombre de ses nœuds fils. Ces informations sont apprises par l'échange de messages d'inscription et de notification.

Avant la transmission des données, le nœud puits injecte un nombre aléatoire dans le réseau. Chaque capteur génère alors son propre nombre aléatoire en se basant sur le nombre émis par le nœud puits et son propre identifiant. Avec le générateur de nombres aléatoires préinstallé, le capteur produit une série de M coefficients ϕ_i (i de 1 à M). Chaque nœud multiplie ses lectures avec les coefficients appropriés. Si un nœud n'est pas une feuille de l'arbre virtuel, il ajoute toutes les valeurs de ses fils. Le nouveau message ainsi généré est alors transmis à son parent. Le processus est répété jusqu'à ce que les M messages atteignent le nœud puits. Il faut mentionner que la matrice des coefficients ne sera pas transmise au nœud puits. Ce dernier résout un système linéaire à M équations afin d'extraire les N données initiales à partir de M messages.

Un autre mécanisme intéressant, nommé CDP (*Compressive Data Persistence*) a été proposé dans [49]. Cette approche se base sur la compression des données lors de la capture des événements. Il en résulte que CDP minimise la consommation énergétique au niveau des nœuds capteurs. Par ailleurs, CDP garantit la disponibilité des données même si certains nœuds sources tombent en panne.

1.4.2.3 Stockage de données

Le stockage redondant de données garantit la disponibilité de l'information en cas de panne du nœud source. La meilleure façon de stocker les données est de coder les lectures des capteurs avant leur stockage ce qui réduit la taille des données stockées. Deux grandes classes de solutions de stockage de données sont utilisées pour les RCSFs : le stockage distribué basé sur les codes Fountain FCBDS (*Fountain Codes Based Distributed Storage*) [50] et le stockage distribué basé sur les codes Raptor RCBDS (*Raptor Codes Based Distributed Storage*) [51].

Dans [50], les auteurs ont présenté deux approches de stockage redondant des données basées sur FCBDS. Ils ont considéré un RCSF à grande échelle avec n nœuds parmi lesquels k capteurs ($k \ll n$) traitant les informations. L'idée principale de ces approches est que chaque nœud stocke un paquet codé tel que k paquets peuvent être récupérés de $(k + \epsilon)$ nœuds pour un petit $0 < \epsilon \ll 1$. Les deux approches utilisent simultanément les algorithmes *Random Walks* (RW) [52] et *Luby Transform* (LT) [53]. Dans la première approche, tous les nœuds doivent avoir les informations sur n (nombre de nœuds) et k (nombre de paquets). La deuxième approche, par contre, fonctionne sans ces informations.

Pour les deux algorithmes LTCDS (*LT-Codes based Distributed Storage*) I et II [50], les paquets sources sont disséminés dans le réseau par un simple algorithme *RW* où chaque nœud transmet le paquet à un voisin aléatoirement choisi parmi ses voisins à un saut.

Chaque paquet visite chaque nœud au moins une fois. Le nœud décide alors de l'accepter ou pas. Quand un paquet est accepté, il est ajouté au paquet codé en utilisant l'opérateur *XOR* et est expédié ensuite au nœud voisin. Après que tous les paquets aient visité un nœud donné, ce dernier déclare le paquet résultant comme étant le paquet à stocker. Les sources peuvent envoyer des mises à jour pour rafraîchir les données stockées.

Le deuxième type de stockage [51] utilise un mécanisme semblable à celui décrit précédemment. Le stockage se fait en deux étapes : le pré-codage en utilisant LDPC (*Low-Density Parity-Check*) [54] et le codage LT (*LT-coding*). Dans la première étape, les k sources génèrent un certain nombre de copies de leurs paquets et utilisent RW pour les disséminer dans le réseau. Chaque nœud, autre que le nœud source, décide d'être un nœud redondant avec une certaine probabilité et stocke un paquet résultant de la combinaison d'un nombre aléatoire de paquets sources. Si un nœud choisit de stocker un paquet, ce dernier ne sera pas réexpédié aux voisins. Les nœuds sources et les nœuds redondants, considérés comme nœuds de pré-codage, servent comme nœuds sources pour la deuxième étape. La deuxième étape utilise les mêmes algorithmes que LTCDS.

Dans [55], les auteurs ont proposé le mécanisme GRDDS (*Greedy Replication-based Distributed Data Storage*) pour les RCSFs dans un contexte d'Internet des objets IoT (*Internet of Things*). Dans ce mécanisme, chaque nœud choisit un voisin pour être le nœud donateur (*donor*). Ce dernier est choisi selon la mémoire disponible pour garder une réplique des données à stocker. En plus, chaque donateur choisit le donateur suivant parmi ses voisins. Le processus de sélection se répète jusqu'à ce que le nombre maximal de répliques soit atteint ou qu'il n'y ait plus de donateurs disponibles. Si un nœud ne trouve pas d'espace disponible ni dans sa mémoire, ni dans les mémoires de ses voisins, il rejette les données.

Malgré sa faible complexité et sa propriété de tolérance aux pannes, l'approche GRDDS proposée présente quelques inconvénients. En l'occurrence, le nombre non contrôlé de répliques peut aboutir à l'inondation du réseau par des répliques des mêmes données, tandis que d'autres données significatives peuvent être supprimées en raison du manque de mémoire.

1.4.3 Couverture et connectivité dans les RCSF à grande échelle

Les algorithmes présentés dans cette partie se basent essentiellement sur la stratégie de déploiement des nœuds et le recouvrement des pannes dans le réseau.

1.4.3.1 L'algorithme OGDC

L'algorithme OGDC (*Optimal Geographical Density Control*) [56] utilise une stratégie de déploiement complètement décentralisée afin de maintenir la couverture de la zone contrôlée en activant un nombre minimal de nœuds capteurs. Il exploite la haute densité du réseau pour planifier l'activation des nœuds redondants. De plus, il suppose que chaque nœud est conscient de sa position et que tous les capteurs sont synchronisés. OGDC utilise trois états pour décrire un nœud : *UNDECIDED*, *ON* et *OFF*.

La solution OGDC commence par sélectionner aléatoirement un nombre de nœuds actifs qui sont désignés comme nœuds de départ. Si un nœud N est un nœud de départ, un de ses voisins, appelé M , placé à une distance approximative de $\sqrt{3}r$ (r est le rayon de couverture), sera sélectionné pour être actif à son tour. Ensuite, si un troisième nœud forme un triangle équilatéral de longueur de côté $\sqrt{3}r$ avec N et M , il sera aussi activé. Le processus se répète jusqu'à ce que tous les nœuds soient *ON* ou *OFF*. Les nœuds actifs changent à chaque itération ce qui garantit l'équilibrage de la consommation énergétique entre tous les nœuds.

1.4.3.2 L'algorithme CRR

L'algorithme CRR (*Connectivity Restoration through Rearrangement*) [57] rétablit la connectivité en réarrangeant quelques nœuds dans le voisinage du nœud défaillant. Le processus de recouvrement est localisé dans une petite partie du réseau pour éviter la réorganisation globale du réseau. De plus, le recouvrement est exécuté seulement si le nœud en panne est un nœud d'articulation.

Pour accomplir sa mission, CRR commence par trouver deux ensemble de voisins à un et à deux sauts du nœud défaillant appelés SN_{1-hop} et SN_{2-hop} . Ce dernier sera utilisé pour chercher les nouvelles positions des nœuds à déplacer appelées points de Steiner SPs (*Steiner Points*). Les SPs sont les nœuds d'un arbre de Steiner [58] qui est utilisé dans la théorie des graphes pour connecter un nombre de sommets ou de sous-graphes en minimisant le coût total. La figure 1.4 illustre le cas d'un SP qui relie trois sommets en minimisant la somme des distances entre les sommets. L'ensemble des voisins à un saut du nœud défaillant SN_{1-hop} sera utilisé pour restaurer la connectivité. Ensuite, CRR déplace quelques nœuds de SN_{1-hop} (et si nécessaire quelques nœuds de SN_{2-hop}) pour occuper les emplacements désignés par les SPs. Le déplacement des nœuds de SN_{2-hop} peut aboutir à la perte de la connectivité dans une autre partie du réseau. Pour pallier ce problème, CRR utilise l'algorithme DARA [59] qui remplace en cascade les nœuds déplacés par leurs

voisins.

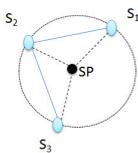


Figure 1.4 : Connexion de trois sommets par un SP

1.4.3.3 L'algorithme DORMS

Contrairement à CRR, DORMS (*Distributed algorithm for Optimized Relay node placement using Minimum Steiner tree*) [60] recouvre les pannes simultanées de plusieurs nœuds. Cet algorithme peut être appliqué dans le cas où les pannes survenues dans le réseau l'ont subdivisé en des segments complètement isolés. L'idée est d'utiliser des nœuds relais (RNs) mobiles pour reconnecter les différents segments isolés.

Avant l'application de DORMS, les nœuds doivent détecter la défaillance de plusieurs nœuds et la division du réseau en plusieurs segments disjoints. Une fois la panne détectée, les RNs de chaque segment se déplacent en cascade vers le centre de la zone contrôlée. Ces nœuds déplacés forment des arbres de Steiner entre les différents segments isolés afin de restaurer la connectivité. L'utilisation des arbres de Steiner permet à DORMS de recouvrir la panne en déplaçant un nombre minimal de RNs.

1.4.3.4 L'algorithme LDMR

Dans l'algorithme LDMR (*Least Distance Movement Recovery*) [61], quand un nœud M tombe en panne, chaque nœud N , voisin de M , détermine tous ses voisins qui ne sont pas points d'articulation et choisit le voisin le plus proche V . Après la sélection du nœud V , tous les nœuds N se déplacent vers la position du nœud M en panne. Chaque voisin N est remplacé à son tour par son voisin V . Tous les nœuds N en déplacement s'arrêtent quand ils arrivent à une distance $\frac{R}{2}$ de la position du nœud défaillant (R étant le rayon de communication) afin de restaurer la connectivité et recouvrir la panne.

1.4.3.5 L'algorithme EDCR

L'algorithme EDCR (*Energy centric Distributed partitioning Detection and Connectivity Restoration*) [62] désigne un nœud de secours pour chaque nœud critique (nœud d'articulation) parmi les voisins. Dès la détection de la panne de ce nœud critique, le

nœud de secours se déplace vers la position du nœud défaillant pour rétablir la connectivité. Il faut mentionner que la sélection des nœuds de secours se base sur quatre critères : l'état du capteur voisin (critique ou non), le degré du capteur (le nombre de fils), la distance inter-capteurs (la distance entre le voisin et le nœud primaire) et enfin l'énergie restante.

L'avantage principal de ECDR est que le recouvrement de la panne se fait d'une manière proactive et distribuée. Ceci permet de rétablir la connectivité dans le réseau rapidement avec un minimum de messages de contrôle. Cependant, EDCR suppose qu'un seul nœud tombe en panne à la fois et qu'aucun autre nœud ne tombe en panne pendant le processus de recouvrement.

En plus des algorithmes présentés précédemment, il existe un autre travail intéressant qui offre une solution complète au problème de la tolérance aux pannes dans les RCSFs combinant à la fois des mécanismes inhérents à l'agrégation des données, la clusterisation, la localisation, le routage, *etc.* [63]. D'autres travaux [64] et [65] ont proposé l'utilisation de robots pour déployer de nouveaux nœuds au lieu des nœuds en panne. Cette technique doit prédéfinir une stratégie pour le mouvement des robots afin de minimiser la distance parcourue par chaque robot et le temps de recouvrement de la panne.

Le tableau 1.4 récapitule les avantages et les inconvénients des mécanismes de tolérance aux pannes dans les réseaux de capteurs sans fil à grande échelle.

L'étude exhaustive des techniques et des mécanismes de tolérance aux pannes dans les RCSFs effectuée dans ce chapitre a fait l'objet d'un papier survey publié dans le journal Computer Communications (Comcom) d'Elsevier [66].

1.5 Conclusion

Dans la première partie de ce chapitre, nous avons présenté les concepts fondamentaux du problème de tolérance aux pannes dans les réseaux de capteurs sans fil. Ainsi, nous avons énuméré les origines des pannes dans les RCSFs et nous avons classifié les mécanismes de tolérance aux pannes dans les RCSFs selon plusieurs critères.

Dans la deuxième partie du chapitre, nous avons mis l'accent sur les principaux mécanismes proposés dans la littérature pour assurer ou améliorer la tolérance aux pannes dans les RCSFs. De plus nous avons distingué les mécanismes appliqués dans les RCSFs selon la taille des réseaux : réseaux à petite échelle et réseaux à grande échelle.

Toutes les solutions de tolérance aux pannes étudiées dans ce chapitre ne prennent

pas en compte un facteur important qui influence considérablement le comportement des RCSFs à savoir les interférences. En présence d'interférences, les performances du RCSF peuvent être dégradées compte tenu de la mauvaise qualité des liens et du taux de perte élevé dans le réseau. L'une des solutions pour pallier le problème d'interférences dans les RCSFs est d'utiliser la communication multi-canal. En particulier, le chapitre 2 sera consacré à la proposition de nouvelles solutions de recouvrement de pannes dans les RCSFs multi-canaux dans un contexte centralisé.

Protocole	Localisation	Agrégation	Clusterisation	Routage intra-cluster	Sélection de CH	Construction des routes	Technique de tolérance aux pannes
DECROP	Non	Oui	Proactive	Multi-saut	Degré du nœud	3 phases	Sélection de pré-hop
ASN	Non	Oui	Proactive	1-saut	Nombre de CHs exigé	Plusieurs rounds phases pour chaque round	Nœuds alternatifs
MuMHR	Non	Oui	Proactive	Multi-saut	Aléatoire	2 phases	Nœud de Backup et chemins alternatifs
ONCP	Non	Non	Réactive	1-saut	Énergie résiduelle	2 phases	Multiple nœuds puits
MRMS	Non	Oui	Proactive	1-saut	Aléatoire	3 phases	reconstruction du cluster Commutation de chemin

Tableau 1.2 : Tableau comparatif des protocoles de routage

Catégorie	Protocole	Avantages	Inconvénients
Gestion de flux	REAR	Chemins de secours Transmission fiable (ACKs)	Nombre de message de contrôle élevé
	Directed Diffusion	Maintenance des routes	Consommation d'énergie élevée
	EAR-LEASH	équilibrage de la consommation d'énergie	Gaspillage d'énergie
	Braided Multipath Routing	Chemins tressés alternés	Consommation d'énergie élevée
	ReInForm	Degré de fiabilité élevé	Gaspillage d'énergie et de bande passante
Gestion de données	DRG	Disponibilité des données en cas de panne d'un nœud	Augmentation du temps de convergence avec la taille du RCSF
	TAG	Service intégré dans le noyau du capteur	La panne d'un nœud entraîne la panne d'un sous-arbre
	Synopsis Diffusion	Transmission des Synopsis à travers plusieurs routes	Gaspillage d'énergie en raison de la transmission de plusieurs copies
	RidSharing	Recouvrement de la panne d'un lien	Complexité de la procédure
	DSC	Réduction de la quantité de données	La corrélation doit être connue en avance par tous les nœuds
	Fault recovery mechanism in single-hop sensor networks	Récupération des données en cas de panne	Les pannes simultanées ne sont pas considérées
	TinyDSM	Disponibilité des données en cas de panne	Pas d'économie d'énergie La source est un goulot d'étranglement
Gestion de couverture & de connectivité	CCP	Maximisation de la durée de vie du réseau Équilibrage de la consommation d'énergie	Exigence d'une densité élevée
	Monitoring and Self-organization Mechanism	Évitement de la division du réseau et maintien de la connectivité	Pas de recouvrement de panne
	MRP	Maintien de la connectivité	Pas de prise en compte de la couverture Nœuds relais manuellement déployés
	C^3R	Restauration de la couverture Réorganisation délocalisée	Délai de transmission Pas de prise en compte des pannes simultanées Mobilité des capteurs exigées

Tableau 1.3 : Mécanismes de tolérance aux pannes pour les RCSFs à petite échelle

Catégorie	Protocole	Avantages	Inconvénients
Gestion de flux	MRMS	Équilibrage de la consommation d'énergie Plusieurs nœuds puits	Gaspillage d'énergie
	ASN	Équilibrage de la consommation d'énergie Empêchement de collision (TDMA)	Surcharge de contrôle élevée
	MuMHR	Économie d'énergie Chemins alternatifs	Délai dû au back-off
	ONCP	Plusieurs nœuds puits Réduction de surcharge de contrôle	Les pannes après la détermination de la route globale non considérées
	DECROP	Construction des routes à l'initialisation Minimisation du nombre de clusters	Augmentation de la consommation d'énergie de CH Épuisement rapide de l'énergie des nœuds proches de BS
Gestion de données	Hybrid Aggregation Mechanism	Disponibilité des données	Compromis entre la taille du groupe et le délai de la transmission
	CDG	Réduction de la quantité de données	Complexité de la reconstruction des données
	FCBDS	Récupération des données Réduction de la quantité de données par le codage	Complexité du codage de données
	RCBDS	Amélioration du recouvrement des données Réduction de la quantité de données par le codage	Complexité du codage des données
Gestion de couverture & connectivité	OGDC	Couverture totale Maximisation de la durée de vie du RCSF	Densité élevée exigée
	CRR	Réorganisation délocalisé du réseau et restauration de la connectivité	La couverture n'est pas prise en compte Mobilité des capteurs exigée
	DORMS	Recouvrement de pannes simultanées	Mobilité des nœuds relais exigée Consommation d'énergie des nœuds relais
	LDMR	Restauration de la connectivité	Mobilité des capteurs exigée

Tableau 1.4 : Mécanismes de tolérance aux pannes pour les RCSFs à grande échelle

Chapitre 2

Proposition d'approches centralisées pour la restauration de la connectivité dans les RCSFs multi-canaux

2.1 Introduction

D'après l'étude que nous avons faite dans le chapitre précédent, nous avons remarqué que les mécanismes proposés pour la tolérance aux pannes dans les RCSFs ne considèrent pas le problème d'interférence qui peut dégrader considérablement les performances du réseau. Ces interférences peuvent être causées par plusieurs facteurs (présence d'autres composants n'appartenant pas au RCSF qui peuvent émettre sur la même bande de fréquence, transmissions simultanées des capteurs voisins,...).

L'une des solutions qui peuvent être proposées pour pallier le problème d'interférence dans les RCSFs est l'utilisation de la communication multi-canal. Ainsi, nous nous intéressons à la proposition de nouvelles solutions de tolérance aux pannes pour les RCSFs dans un contexte multi-canal. Ce chapitre sera consacré à la résolution d'un tel problème dans un contexte centralisé. Nous nous intéressons en particulier au problème de la restauration de la connectivité dans les RCSFs suite à la défaillance d'un nœud d'articulation AP (*Articulation Point*) tout en tenant compte du fait que la restauration de la panne doit minimiser les interférences entre les nœuds dans le réseau.

Pour ce faire, nous allons d'abord nous pencher sur l'étude de différents mécanismes

d'allocation multi-canal dans les RCSFs. Ensuite, nous essaierons de formuler notre problématique, qui consiste à assurer la tolérance aux pannes des nœuds d'articulation dans un RCSF, tout en garantissant une allocation minimisant les interférences, sous forme d'un problème d'optimisation multi-objectif. Puis, nous proposerons une approche préventive au problème sus-indiqué basée sur des solutions à base d'heuristiques et la comparerons à une solution exacte du problème. Enfin, nous proposerons une approche pour la mise en place de notre solution dans un environnement réel qui tienne compte de la division du réseau en plusieurs segments isolés. Ainsi, il faudrait faire en sorte que tous les nœuds du réseau soient conscients de la procédure de recouvrement avant ou après la production de la panne.

2.2 Communication multi-canal dans les RCSFs

Les interférences entre des transmissions simultanées peuvent aboutir à une dégradation rapide des performances des RCSFs. La réduction du taux d'interférence a été largement discutée dans la littérature et il est démontré que la communication multi-canal sont une des techniques les plus efficaces pour lutter contre ce problème. Cette technique se réfère à la division de la bande passante en plusieurs bandes de fréquence (respectivement le temps en plusieurs slots) pour permettre aux nœuds interférents de transmettre leurs données sans générer de conflits (interférences, collisions). En général, les interférences ne peuvent pas être complètement éliminées mais seulement réduites en raison de la limitation du nombre de canaux disponibles dans le réseau. Un RCSF dispose d'un nombre limité de canaux (16 canaux pour IEEE 802.15.4). En plus, ce nombre est réduit par les environnements dans lesquels les RCSFs peuvent évoluer. Depuis l'intégration de la communication multi-canal dans les RCSFs, de nombreuses recherches intéressantes ont été consacrées à l'exploitation optimale des avantages de ce type de communication dans les RCSFs [67–72]. Les mécanismes proposés aident à minimiser le taux d'interférence ce qui aboutit à une réduction du nombre de collisions et de retransmissions, et par conséquent, la minimisation de la consommation énergétique.

2.2.1 Méthodes d'accès au médium

Dans la littérature, on recense trois catégories de techniques d'accès au canal pour les RCSFs : l'accès *schedule-based*, l'accès *contention-based* et l'accès *hybride*.

2.2.1.1 Accès basé sur l'ordonnancement

Dans cette catégorie, les nœuds commutent leurs radios entre les différents canaux suivant un ordonnancement bien défini. La technique la plus connue dans cette catégorie est la technique d'accès TDMA (*Time Division Multiple Access*) qui divise le temps en slots et qui pour chaque nœud alloue un slot et un canal de transmission. Tous les nœuds doivent être synchronisés pour que l'émetteur et le récepteur commutent leurs radios vers le même canal en même temps. Cette méthode d'accès garantit une transmission des données sans collision ce qui réduit la consommation énergétique. Cependant, plusieurs facteurs limitent l'utilisation de cette méthode d'accès, notamment la mise à l'échelle et l'adaptation aux changements dans le réseau.

2.2.1.2 Accès basé sur la contention

Les nœuds concourent pour gagner l'accès au canal et transmettre leurs données. Dans cette méthode d'accès aucune synchronisation n'est nécessaire d'où une simplicité de déploiement et une meilleure adaptation aux changements qui peuvent survenir dans le réseau. Cependant, cette méthode ne garantit pas l'élimination des collisions surtout avec l'augmentation du nombre de nœuds émetteurs.

2.2.1.3 Accès hybride

Contrairement aux méthodes basées sur l'ordonnancement où tous les slots sont pré-assignés, avec les méthodes hybrides, le temps est divisé en slots mais à chaque slot, les nœuds concourent pour l'accès au canal. Le nœud ayant gagné la contention pour ce slot accède au canal pour transmettre ses données sans collision.

2.2.2 Méthodes d'allocation des canaux

L'allocation des canaux est l'un des défis majeurs de communication multi-canal. En effet, la stratégie utilisée pour assigner les canaux aux nœuds influence directement le poids des interférences dans le réseau. Ainsi, l'allocation doit être faite de sorte à minimiser le nombre des nœuds interférents auxquels sont assignés les mêmes canaux. Les mécanismes d'allocation des canaux peuvent être classés en trois catégories selon le type d'allocation : allocation fixe, allocation semi-dynamique et allocation dynamique.

2.2.2.1 Allocation fixe

Les canaux sont assignés à l'avance et restent inchangés pendant le processus de communication. Cette allocation peut être périodiquement rafraîchie pour tenir compte des changements dans la topologie du réseau. L'allocation fixe des canaux est appropriée pour les réseaux ayant peu de changements topologiques et où les nœuds sur des routes différentes ne doivent pas communiquer. Le protocole le plus connu dans cette catégorie est TMCP (*Tree-based Multi-Channel Protocol*) [67] conçu pour la collecte de données dans les RCSFs. L'idée principale de TMCP est de grouper les nœuds du réseau en sous-arbres disjoints, le nœud puits comme racine de tous les sous-arbres, et d'assigner un canal différent à chaque sous-arbre. La diffusion des messages entre des nœuds de sous-arbres différents n'est donc pas possible, ce qui remet en question l'efficacité d'un tel algorithme.

2.2.2.2 Allocation semi-dynamique

Dans cette catégorie, chaque nœud du réseau se verra attribuer un canal fixe dédié à la réception de données. Pour qu'un nœud puisse émettre des messages à un autre nœud, il doit commuter vers le canal de réception de ce dernier. Chaque nœud utilise alors un seul canal fixe pour la réception et plusieurs canaux pour l'émission. Le canal d'émission est celui du nœud récepteur pour la communication courante. Par analogie, il est possible de fixer le canal d'émission au lieu du canal de réception, auquel cas tous les récepteurs doivent commuter sur le canal de l'émetteur pour recevoir ses données. On peut aussi allouer un canal fixe pour chaque paire émetteur/récepteur. Cette méthode en revanche nécessite la synchronisation entre l'émetteur et le récepteur pour commuter en même temps vers le même canal. Le récepteur et l'émetteur peuvent être synchronisés sur le même canal soit en suivant un ordonnancement bien défini a priori, soit en utilisant un canal de contrôle. Le premier type de coordination est adéquat pour les applications de collecte périodique de données dans lesquelles chaque nœud connaît les instants de génération des paquets. Dans le deuxième type de synchronisation, le temps est divisé en périodes. Au début de chaque période tous les nœuds utilisent un canal de contrôle (en se basant sur la contention) pour prévenir leurs destinataires de l'existence des données à transmettre. Les nœuds ayant des données à recevoir commutent vers les canaux sur lesquels les paquets vont arriver.

Dans la littérature, plusieurs algorithmes sont proposés dans cette catégorie. Le protocole CMAC (*Convergent MAC*), proposé dans [68] utilise une radio avec une consommation très faible (*Low Power Wakeup Radio*) pour réveiller le récepteur et négocier le canal de transmission. Avec le protocole MC-LMAC (*Multi-channel Lightweight Medium Access*

Control) [69], le temps est divisé en slots et à chaque nœud est assigné un slot pour transmettre sur un canal particulier. MMSN (*Multifrequency Media access control for wireless Sensor Networks*) [70] est le premier protocole dédié aux RCSFs multi-canaux utilisant le *slotted CSMA* (*Carrier Sense Multiple Access*). Avec MMSN, les nœuds concourent pour l'accès au canal au début de chaque slot de temps. L'inconvénient majeur de cette méthode d'allocation est la nécessité de synchroniser l'émetteur et le récepteur pour éviter le problème de récepteur manquant. Ce problème se produit lorsque l'émetteur envoie ses données alors que la radio du récepteur est placée sur un autre canal.

2.2.2.3 Allocation dynamique

Dans ce type d'allocation, les nœuds utilisent un canal différent pour chaque transmission, ce qui nécessite la coordination entre les nœuds pour négocier le canal à utiliser à chaque transmission. Trois techniques sont utilisées dans l'allocation dynamique : *canal dédié au contrôle*, *phase de répartition* et *saut de fréquence*.

- **Canal dédié au contrôle (*dedicated control channel*)** : les nœuds se synchronisent en échangeant des messages sur un canal consacré au contrôle et négocient le canal qui va être utilisé pour la transmission. Une fois la négociation terminée, chaque nœud commute sa radio vers le canal choisi.
- **Phase de répartition (*split phase*)** : les nœuds accèdent au médium en deux phases. Pendant la phase de contrôle, tous les nœuds utilisent un même canal et négocient avec leurs destinataires les canaux qui vont être utilisés durant la phase de transmission. L'accès pendant la phase de contrôle est basé sur la contention. Le canal utilisé pour le contrôle est aussi utilisé pour la transmission des données.
- **Saut de fréquence (*frequency-hopping*)** : les nœuds commutent entre les différents canaux en utilisant le saut de fréquence. Une paire de nœuds arrête de commuter leurs radios lorsque les deux nœuds se mettent d'accord sur la fréquence à utiliser pour la transmission. Une fois la transmission terminée, les nœuds recommencent le saut de fréquence.

Cette méthode est très appropriée pour les RCSFs mobiles ou qui changent souvent leurs topologies. Elle est, par contre, très coûteuse pour les réseaux statiques en raison de la surcharge des messages de contrôle (pour la négociation de canal à utiliser). Comme exemple de protocoles utilisant l'allocation dynamique, nous citons le protocole Y-MAC (*energy-efficient multichannel MAC protocol*) [71] qui utilise une méthode d'accès hybride, dans laquelle à chaque récepteur est assigné un slot de temps et un canal. Des canaux différents peuvent être assignés au même récepteur à des slots de temps différents. Le

protocole Rainbow [72], conçu pour la collecte fiable des données dans les RCSFs, utilise TDMA en local et FHSS (*Frequency Hopping Spread Spectrum*). L'utilisation de FHSS diminue les collisions, améliore le débit et évite les interférences.

2.2.3 Synthèse

Toutes les solutions décrites précédemment s'intéressent principalement au problème d'allocation des canaux entre les nœuds capteurs sans mettre l'accent sur le fait que les pannes qui surviennent dans le réseau peuvent considérablement altérer le fonctionnement d'un tel réseau.

Dans un contexte multi-canal, l'occurrence d'une panne peut non seulement dégrader la connectivité et la couverture dans le réseau mais aussi rendre l'allocation optimale des canaux, initialement faite, complètement obsolète. Ainsi, nous proposons dans la suite du chapitre de nous intéresser au problème de la tolérance aux pannes dans les RCSFs dans un contexte multi-canal.

2.3 Formulation du problème : Tolérance aux pannes dans les RCSFs multi-canaux

Dans le cadre de cette thèse, nous nous intéressons au problème de la tolérance aux pannes dans les RCSFs multi-canaux. L'objectif de la communication multi-canal dans notre cas est de réduire les interférences entre les nœuds capteurs. Ainsi, l'allocation initiale des canaux doit être faite de sorte à ce qu'elle minimise le nombre d'interférences entre les nœuds. Dans le reste de cette thèse, nous utilisons une allocation semi-dynamique des canaux basée sur le canal de réception. Cette allocation est la plus simple à mettre en place et la plus utilisée. Par ailleurs, si une panne survient dans le réseau, son recouvrement doit être effectué de sorte à toujours obtenir une configuration du réseau qui minimise les interférences (allocation optimale des canaux dans le réseau). En l'occurrence, dans cette thèse nous ne considérons que les pannes des nœuds d'articulations (APs) qui sont responsables du maintien de la connectivité du réseau.

Pour ce faire, nous optons pour des techniques utilisées dans la théorie des graphes. En particulier, nous proposons d'utiliser des techniques de coloration des graphes pour l'allocation des canaux ainsi qu'une technique basée sur les points de Steiner [73] pour la restauration de la connectivité du réseau.

2.3.1 Modèle du réseau

Un RCSF peut être modélisé par un graphe d'interférence $G = (V, E_I)$, avec V l'ensemble des nœuds capteurs présents dans le RCSF dont le nombre est égal à n ($|V| = n$). Tous les capteurs sont mobiles et peuvent ainsi se déplacer pour recouvrir une panne. E_I représente l'ensemble des liens d'interférence entre les nœuds. Une arête $e_i, (u, w) \in E_I$, existe entre deux nœuds capteurs u et w si et seulement si ces deux capteurs sont interférents (la distance $d(u, v) \leq 2R_c$, avec R_c le rayon de communication). Chaque canal est représenté par une couleur différente. Nous définissons l'ensemble M comme étant l'ensemble des couleurs à utiliser pour colorer les nœuds du graphe. Le nombre de couleurs est égal à m ($|M| = m$).

A cette fin, nous définissons la fonction $c : V \rightarrow M$, où $c(v) = j$ signifie que le sommet v est coloré par la couleur j . Le vecteur $s = (V_1, V_2, \dots, V_m)$ est le vecteur décision qui minimise le nombre de sommets adjacents (interférents) ayant la même couleur (le nombre de conflits), où V_j désigne l'ensemble des nœuds colorés par la couleur j .

2.3.2 Problème d'allocation des canaux

Compte tenu du nombre limité de canaux disponibles dans le RCSF, faire une allocation optimale des canaux dans le réseau revient à réduire le nombre de nœuds interférents utilisant les mêmes canaux. Ce qui revient à minimiser le nombre de nœuds adjacents ayant la même couleur dans le graphe d'interférence. Pour ce faire, nous formulons le problème d'allocation des canaux sous forme d'un problème de coloration des sommets d'un graphe avec un nombre limité de couleurs.

Pour minimiser le nombre de conflits, nous définissons la fonction binaire pour un nœud $v \in V_j$ qui vérifie si ce nœud est en conflit avec un autre nœud interférent. Pour un nœud $v \in V_j$, cette fonction est donnée par :

$$\delta_j(v) = \begin{cases} 1 & \text{si } \exists u \in V_j \text{ tel que } (v, u) \in E_I \\ 0 & \text{sinon.} \end{cases} \quad (2.1)$$

Le nombre de conflits au niveau d'une couleur j peut être déterminé par :

$$\Delta_j = \sum_{v \in V_j} \delta_j(v). \quad (2.2)$$

Ainsi, nous pouvons définir f_1 , la fonction objectif qui vise à minimiser le nombre de sommets adjacents colorés avec la même couleur par :

$$f_1(s) = \sum_{j=1}^m \Delta_j \quad (2.3)$$

d'où le problème d'optimisation consistant à minimiser f_1 sous la contrainte que tous les sommets du graphe soient colorés, c'est à dire que tous les nœuds capteurs ont un canal alloué. Ceci se traduit par :

$$\text{minimiser } f_1(s) \quad (2.4)$$

$$c(v) \in M, \forall v \in V \quad (2.5)$$

Compte tenu de la fonction objectif f_1 pour une allocation optimale des canaux dans le réseau, nous garantissons que quelle que soit la disposition des nœuds capteurs dans le réseau (configuration initiale ou redéploiement partiel pour recouvrir une panne), les canaux sont allouées de sorte à minimiser les conflits entre les nœuds interférents.

2.3.3 Problème de rétablissement de la connectivité

Le problème de recouvrement de la panne d'un nœud d'articulation peut être présenté comme étant un problème d'optimisation qui vise à recouvrir la panne et restaurer la connectivité dans le réseau en minimisant le nombre de nœuds capteurs à déplacer pour assurer le recouvrement.

Un algorithme basé sur les points de Steiner (SPs) permet de déterminer les nouvelles positions des nœuds à déplacer. Un ensemble minimal des SPs doit par la suite être élu afin de rétablir la connectivité. Les nœuds voisins à un ou deux sauts du nœud défaillant peuvent alors être déplacés pour assurer la connectivité dans le réseau.

Ainsi, la restauration de la connectivité dans le réseau doit être faite de sorte à :

- Minimiser le nombre de nœuds déplacés pour recouvrir la panne du nœud point d'articulation (le nombre de SPs qui seront occupés par des nœuds) ;
- Minimiser la distance totale parcourue par les nœuds déplacés.

Si en plus nous tenons compte du contexte multi-canal, le recouvrement de la panne doit aussi :

- Minimiser le nombre de nœuds interférents ayant le même canal.
- Minimiser le nombre de nœuds auxquels sont assignés de nouveaux canaux après la restauration de la connectivité.

Ainsi, notre problème de recouvrement de pannes peut être défini comme un problème

d'optimisation multi-objectif où les objectifs à atteindre sont donnés comme suit :

1. Minimiser le nombre de nœuds interférents ayant le même canal ;
2. Minimiser le nombre de nœuds auxquels sont assignés de nouveaux canaux après la restauration de la connectivité ;
3. Minimiser le nombre de nœuds déplacés pour recouvrir la panne du nœud d'articulation ;
4. Minimiser la distance totale parcourue par les nœuds déplacés.

Pour formuler mathématiquement notre problème, nous définissons :

- SP_set : l'ensemble de tous les points de Steiner trouvés autour du nœud défaillant ($|SP_set| = k$).
- $H_1(AP_f)$: l'ensemble des voisins du nœud défaillant à un saut ($H_1(AP_f) = \{v \in V | d(v, AP_f) < R_c\}$).
- $H_2(AP_f)$: l'ensemble des voisins du nœud défaillant à deux sauts ($H_2(AP_f) = \{v \in V | R_c < d(v, AP_f) < 2R_c\}$).
- G' : le graphe coloré représentant le RCSF multi-canal après le recouvrement de la panne ($G' = (V', E'_I)$) et défini de la même façon que $G = (V, E)$ (section 2.3.1),
- s' : le vecteur solution du problème multi-objectif ($s' = (V'_0, V'_1, V'_2, \dots, V'_m)$).
- $c_s(v)$: la fonction qui donne la couleur du sommet v dans la solution s .
- V'_j : l'ensemble de sommets du graphe colorés par la couleur j ($V'_j = \{v \in V' | c_{s'}(v) = j\}$).

Nous introduisons une couleur fictive d'identifiant 0 et nous définissons V'_0 : l'ensemble des sommets de couleur 0 qui désignent des emplacements non occupés par des nœuds. Initialement, $V_0 = SP_set$; ensuite, V_0 contient les SPs non occupés par les nœuds capteurs remplaçants et les anciens emplacements de nœuds déplacés.

Chaque objectif i , $i = 1, \dots, 4$, décrit ci-dessus peut être défini par une fonction objectif f_i à optimiser. Si nous regardons l'objectif 1 qui consiste à minimiser le nombre de nœuds adjacents ayant la même couleur, cet objectif peut être atteint en minimisant la fonction f_1 (équation (2.3)) définie dans la section 2.3.2. Nous définissons d'abord pour chaque f_i , une fonction binaire au niveau d'un nœud, ensuite une fonction au niveau d'une couleur et finalement sur toutes les couleurs.

Pour déterminer le nombre de nœuds recolorés, nous vérifions tout d'abord si un nœud est coloré ou non par la fonction binaire :

$$\gamma_j(v) = \begin{cases} 1 & \text{si } v \in V'_j \wedge c_{s'}(v) \neq c_s(v) \\ 0 & \text{sinon.} \end{cases} \quad (2.6)$$

La fonction Γ_j qui calcule le nombre de nœuds recolorés dans une classe de couleur V'_j est donnée par :

$$\Gamma_j = \sum_{v \in V'_j} \gamma_j(v) \quad (2.7)$$

Enfin, la deuxième fonction objectif f_2 qui détermine le nombre total de nœuds recolorés dans le réseau est définie par :

$$f_2(s') = \sum_{j=1}^m \Gamma_j \quad (2.8)$$

De la même façon, pour déterminer les points de Steiner élus pour être occupés par les nœuds remplaçants dans le réseau, nous définissons $\varphi_j(v)$ qui vérifie si un SP v est occupé par un nœud capteur ou non :

$$\varphi_j(v) = \begin{cases} 1 & \text{si } v \in SP_set \cap V'_j \wedge j \neq 0 \\ 0 & \text{sinon.} \end{cases} \quad (2.9)$$

$$\Phi_j = \sum_{v \in V'_j} \varphi_j(v) \quad (2.10)$$

$$f_3(s') = \sum_{j=1}^m \Phi_j \quad (2.11)$$

Pour formuler la quatrième fonction objectif f_4 , nous définissons la fonction binaire d_{SP_i} , qui vérifie si un nœud est déplacé ou non, par :

$$d_{SP_i}(u) = \begin{cases} 1 & \text{si } u \text{ remplace } SP_i \\ 0 & \text{sinon} \end{cases} \quad (2.12)$$

La distance parcourue par un nœud capteur, éligible d'être déplacé, est donnée par :

$$D(SP_i) = \sum_{u \in H_1 \cup H_2} dist(SP_i, u) d_{SP_i}(u) \quad (2.13)$$

avec

$$dist(SP_i, u) = \sqrt{(x_{SP_i} - x_u)^2 + (y_{SP_i} - y_u)^2} \quad (2.14)$$

où (x_{SP_i}, y_{SP_i}) et (x_u, y_u) représentent les coordonnées de SP_i et de u .

La fonction f_4 qui représente la distance totale parcourue par tous les nœuds déplacés

est définie par :

$$f_4(s') = \sum_{SP_i \in SP_set} D(SP_i). \quad (2.15)$$

Ainsi, notre problème d'optimisation multi-objectif revient à minimiser la fonction objectif résultante f , avec $f(s') = [f_1(s'), f_2(s'), f_3(s'), f_4(s')]$, sous les contraintes que le graphe soit connecté, tous les nœuds soient colorés et le nombre des emplacements vides est égal au nombre initial de SPs. Ceci peut être traduit par :

$$\text{minimiser } f(s') \quad (2.16)$$

$$c_{s'}(v) \in M, \forall v \in V - H_1 \cup H_2 \quad (2.17)$$

$$|V'_0| = |SP_set| + 1 \quad (2.18)$$

$$\forall j \neq 0, \forall u \in H_2 \cup \{SP_set \cap V'_j\}, \exists v \in \{SP_set \cap V'_i\} | \text{dist}(u, v) \leq R_c \quad (2.19)$$

2.4 Résolution exacte du problème d'optimisation multi-objectif

Dans cette partie, nous allons proposer une résolution exacte du problème d'optimisation multi-objectif. Pour ce faire, nous définissons un certain nombre de concepts inhérents. Ces définitions sont tirées de [74].

Tout d'abord, un problème d'optimisation multi-objectif MOP (Multi-objective Optimization Problem) est tout problème qui s'écrit sous la forme :

$$\text{optimiser } F(x) \quad (2.20)$$

tel que :

$$F(x) = (f_1(x), \dots, f_k(x)) \quad (2.21)$$

Le vecteur décision **idéal** est le vecteur :

$$x_{0(i)} = [x_1^{0(i)}, x_2^{0(i)}, \dots, x_n^{0(i)}] \quad (2.22)$$

tel que :

- $f_i(x^{0(i)}) = \text{opt}(f_i(x)), x \in \Omega$, avec Ω l'ensemble des solutions possibles.
- $f^0 = [f_1^0, f_2^0, \dots, f_k^0]$, où f_i^0 désigne l'optimum de la $i^{\text{ème}}$ fonction objectif, est l'idéal pour le MOP.

Généralement, la solution idéale n'est pas réalisable. En ayant plusieurs fonctions objectifs, la notion d'optimum change parce que pour un MOP le but est de trouver de bons compromis qui permettent de satisfaire tous les objectifs à la fois. Pour ce faire, nous introduisons la notion d'*optimum de Pareto*.

Définition 1 (*Dominance Pareto*) : un vecteur $\mathbf{u} = (u_1, \dots, u_k)$ domine un autre vecteur $\mathbf{v} = (v_1, \dots, v_k)$ (noté $\mathbf{u} \preceq \mathbf{v}$) si et seulement si \mathbf{u} est partiellement inférieur à \mathbf{v} , c'est à dire $\forall i \in \{1, \dots, k\}, u_i \leq v_i \wedge \exists i \in \{1, \dots, k\}$ tel que $u_i < v_i$.

Définition 2 : Une solution $x \in \Omega$ est dite **Pareto Optimale** si et seulement s'il n'existe aucun $x' \in \Omega$ pour lequel $v = F(x') = (f_1(x'), \dots, f_k(x'))$, et que v domine u avec $u = F(x) = (f_1(x), \dots, f_k(x))$.

Définition 3 (*Ensemble optimal de Pareto*) : Pour un problème $F(x)$ donné de type MOP, le Pareto set Optimal est P^* tel que : $P^* := \{x \in \Omega \mid \neg \exists x' \in \Omega \text{ tel que } F(x') \preceq F(x)\}$.

En se basant sur ces définitions, plusieurs méthodes de résolution des MOPs sont proposées [74]. Parmi ces méthodes, nous optons pour la méthode **Min-Max**. L'idée d'appliquer l'optimum de min-max aux problèmes d'optimisation multi-objectifs s'inspire de la théorie des jeux qui traite la résolution des situations contradictoires. La méthode Min-Max compare les déviations relatives des minimums des fonctions objectifs accessibles séparément. Considérons la $i^{\text{ème}}$ fonction objectif, la déviation peut être définie comme suit :

$$z'_i(x) = \frac{|f_i(x) - f_i^0|}{|f_i^0|} \quad (2.23)$$

ou encore

$$z''_i(x) = \frac{|f_i(x) - f_i^0|}{|f_i(x)|} \quad (2.24)$$

Soit $z(x) = [z_1(x), z_2(x), \dots, z_k(x)]^T$ le vecteur des incréments relatifs, il peut être déterminé comme suit :

$$\forall i \in I, z_i(x) = \max\{z'_i(x), z''_i(x)\} \quad (2.25)$$

Ainsi, l'optimum Min-Max peut être défini comme suit :

Un point $x^* \in \Omega$ est un optimal Min-Max, si pour chaque $x \in \Omega$, la formule de récurrence suivante est satisfaite :

Étape 1 :

$$v_1(x^*) = \min_{x \in \Omega} \{z_{i_1}(x)\} \quad (2.26)$$

et $I_1 = \{i_1\}$, où i_1 est l'indice pour lequel la valeur de $z_1(x)$ est maximale.

S'il y a un ensemble des solutions $X_1 \subset \Omega$ qui satisfait l'étape 1, alors

Étape 2 :

$$v_2(x^*) = \min_{x \in X_1} \{\max_{i \in I, i \notin I_1} z_i(x)\} \quad (2.27)$$

et $I_2 = \{i_1, i_2\}$, avec i_2 l'indice pour lequel la valeur de $z_i(x)$ dans cette étape est maximale.

S'il y a un ensemble des solutions $X_{r-1} \subset \Omega$ qui satisfait l'étape $r - 1$, alors

Étape r :

$$v_r(x^*) = \min_{x \in X_{r-1}} \{\max_{i \in I, i \notin I_{r-1}} z_i(x)\} \quad (2.28)$$

et $I_r = \{I_{r-1}, i_r\}$, où i_r est l'indice pour lequel la valeur de $z_i(x)$ dans la $r^{\text{ième}}$ étape est maximale.

S'il y a un ensemble de solutions $X_{k-1} \subset \Omega$ qui satisfait l'étape $k - 1$, alors

Étape k :

$$v_k(x^*) = \min_{x \in X_{k-1}} z_i(x) \text{ for } i \in I \text{ et } i \notin I_{k-1} \quad (2.29)$$

où $\{v_1(x^*), \dots, v_k(x^*)\}$ est l'ensemble des valeurs optimales des déviations fractionnaires.

Cette méthode peut être implémentée sous MATLAB [75] ou CEPLEX et fournit un résultat exact. Malheureusement, elle n'est pas applicable dans les RCSFs en raison de son temps de calcul très élevé et sa complexité (NP-complet). Ainsi, nous avons opté pour l'utilisation des heuristiques afin de trouver une solution à notre problème d'optimisation en un temps raisonnable. Les heuristiques proposées simplifient la résolution du problème d'optimisation multi-objectif et peuvent être appliquées dans les RCSFs. Dans ce qui suit, nous proposons deux approches centralisées pour la résolution de notre problème : une approche préventive PFR (*Preventive Failure Recovery*) et une approche curative RNFR (*Rotating Nodes based Failure Recovery*).

2.5 Preventive Failure Recovery : PFR

Dans cette partie, nous proposons l'approche PFR (*Preventive Failure Recovery*) qui traite la panne d'un nœud d'articulation causée par l'épuisement de sa ressource énergé-

tique. Cette approche est préventive et elle est déclenchée juste avant la panne du nœud d'articulation.

Pour trouver une solution faisable de notre problème d'optimisation multi-objectif dans un temps raisonnable, nous décomposons le problème d'optimisation multi-objectif initial en sous-problèmes où chaque sous-problème est associé à un ou plusieurs objectifs à optimiser.

Ainsi, si nous voulons apporter une solution à notre problème de tolérance aux pannes dans les RCSFs multi-canaux, cette solution sera décomposée comme suit :

1. D'abord, trouver une configuration initiale optimale du réseau qui minimise le nombre de voisins adjacents qui utilisent le même canal. Cette solution peut être assurée en utilisant les algorithmes de coloration des graphes avec m couleurs représentant les canaux. Un nombre d'heuristiques proposées dans la littérature peuvent résoudre ce problème en un temps polynomial.
2. Ensuite, suite à l'occurrence d'une panne, il faut trouver l'ensemble minimal des nœuds voisins qui serviront à remplacer le nœud d'articulation défaillant afin de restaurer la connectivité dans le réseau. Le choix de cet ensemble doit se faire de sorte à minimiser la distance totale parcourue par les voisins. Pour ce faire, nous utilisons l'heuristique *k-LCA* (*k-restricted Loss-Contracting Algorithm*) [76] qui permet de trouver les points de Steiner utilisés pour déterminer les nouvelles positions des nœuds remplaçants.
3. Enfin, une fois la panne recouverte par le déplacement des nœuds, nous procédons à la réallocation des canaux dans le voisinage du nœud défaillant de sorte à minimiser le nombre de conflits (nombre de nœuds adjacents utilisant le même canal) et minimiser le nombre de nœuds recolorés (nombre de nœuds réassignés à de nouveaux canaux).

Nous introduisons tout d'abord les hypothèses suivantes :

- **Hypothèse 1** : Le RCSF étudié est déployé pour une application de collecte périodique de données non critiques.
- **Hypothèse 2** : Un seul nœud d'articulation tombe en panne à la fois.
- **Hypothèse 3** : Aucun autre nœud tombe en panne durant la procédure de recouvrement.
- **Hypothèse 4** : Initialement, tous les nœuds ont le même niveau d'énergie.

2.5.1 Allocation initiale des canaux

Le problème d’allocation des canaux dans le réseau peut être vu comme un problème de coloration de graphes à m couleurs. Nous présentons dans le paragraphe suivant, un certain nombre d’heuristiques qui ont été introduites dans la littérature pour résoudre un tel problème.

2.5.1.1 Heuristiques de coloration de graphes

Les heuristiques de coloration des graphes peuvent être classifiées en trois classes : *constructives*, de *recherche locale* et *évolutionnaires*. La première classe inclut les heuristiques construisant la solution pas à pas jusqu’à l’obtention d’un graphe coloré complet (ex : *l’algorithme glouton* et *DSATUR*). En revanche, les algorithmes de recherche locale commencent d’un graphe initialement coloré et essaient d’améliorer la solution itérativement (ex : le *recuit simulé* et *Tabucol* [77]). Enfin, les heuristiques évolutionnaires se basent sur un ensemble des individus qui s’adaptent individuellement et coopèrent entre eux par l’échange d’informations (ex : les *algorithmes génétiques* et les *algorithmes des colonies de fourmis*).

Dans les réseaux peu denses, les heuristiques de recherche locale sont très efficaces. Cependant, la qualité de la solution se dégrade avec la croissance du nombre de nœuds. Pour améliorer les performances des algorithmes de recherche locale, beaucoup de travaux proposent de les combiner avec un algorithme constructif pour réduire la taille du graphe avant l’application de la méthode de recherche locale. L’idée est que l’heuristique constructive colore le nombre maximal de sommets sans conflits appelés *classes de couleurs stables*. Ensuite, l’heuristique de la recherche locale colore les nœuds restants.

2.5.1.2 Proposition d’algorithmes de coloration centralisés

Compte tenues des heuristiques sus-indiquées, nous optons dans le cas de notre étude, pour une technique de coloration qui combine un algorithme constructif et un algorithme de recherche locale. L’allocation se fait d’une façon centralisée par le nœud puits, ce qui facilite la coloration puisque ce nœud peut avoir une vue globale du réseau. Pour commencer, l’heuristique constructive exploite les informations de voisinage rassemblées par le nœud puits pour construire les classes stables. L’algorithme fonctionne comme suit :

Initialisation

- Selon leurs degrés (nombre de sommets adjacents), les sommets sont mis dans une liste L en commençant par le sommet de degré le plus élevé.

- $C = \{C_1, \dots, C_m\}$ l'ensemble des sommets colorés, où la classe C_j inclut les sommets colorés par j . C est initialisé à \emptyset ($C = \emptyset$).
- $Arrêt=faux$

Tant que $Arrêt=faux$ **faire**

1. Prendre le premier sommet v dans L , le colorer avec la couleur disponible la plus petite j , le mettre dans C_j et l'enlever de la liste L .
2. Prendre le sommet suivant, s'il n'a aucun lien avec un sommet $u \in C_j$, le colorer avec j , l'enlever de L et l'ajouter au C_j . Sinon, passer au sommet suivant. Répéter ce processus jusqu'à la fin de la liste L .
3. Si $L = \emptyset$ ou aucune couleur n'est disponible, $Arrêt=vrai$.

Fin Tant que

Cet algorithme réduit le nombre de sommets non colorés, ce qui augmente l'efficacité de l'heuristique de recherche locale. Pour colorer les sommets restants, nous utilisons une heuristique basée sur l'algorithme *Tabucol* [77] qui est prouvé fournir de bons résultats.

Tabucol commence d'une solution initiale s et déplace, itérativement, un sommet v de C_i vers C_j pour obtenir une nouvelle solution s' . Le mouvement (v, i) devient *tabou*, ce qui signifie que v ne peut pas être coloré par la couleur i pour t itérations; t étant un paramètre. Le mouvement peut être enlevé de la liste *tabou* TL (*Tabu List*) en se basant sur une fonction d'*aspiration* Asp . Initialement :

$$Asp[z] = z - 1$$

où z représente le nombre de conflits dans une solution s . Ensuite, à chaque fois une solution s' est générée, tel que $f(s') \leq Asp[f(s)]$, nous mettons :

$$Asp[f(s)] = f(s') - 1$$

où $f(s)$ représente le nombre de sommets adjacents appartenant à la même classe colorée. Si le mouvement qui donne s' est *tabou*, nous enlevons le statut *tabou* de ce mouvement et nous le considérons comme étant possible. Le choix de la fonction d'aspiration évite que l'algorithme retourne vers une solution déjà visitée. Le nombre d'itérations peut être limité par un temps de calcul maximal prédéfini ou à un nombre maximal d'itérations consécutives sans amélioration de la solution optimale. Les étapes sont décrites en détail dans l'algorithme 1; où s , s' et s^* représentent respectivement la solution initiale, la solution actuelle et la solution optimale partielle.

Le choix du paramètre p utilisé dans l'algorithme 1, représentant le nombre de solutions

Algorithm 1 *Heuristique basée sur Tabucol*

Initialisation :

- $s = \{C_0, C_1, \dots, C_m\}$, $C_0 = L$ (ensemble des sommets incolores);
- $TL = \emptyset$, and $s^* = s$;
- Initialiser Asp et max_itr
- $Arrt = faux$, $itr = 1$;

Tant que $Arrt = faux$ **et** $itr < max_itr$

1. Générer $N'(s)$ de p (paramètre) solutions choisies de $N(s) = \{s' | s' \text{ obtenue de } s \text{ par le déplacement d'un sommet de } C_0 \text{ à } C_i, i = 1 \dots m, \text{ avec aucun mouvement tabou sauf si } f(s') \leq Asp[f(s)]\}$;
 2. $s' = argmin_{s' \in N'(s)} f(s')$, (la solution avec le nombre minimal de conflits);
 3. Si $f(s') < f(s^*)$, $s^* = s'$. si $f(s^*) = 0$, $Arrt = faux$;
 4. $s = s'$;
 5. Mettre à jour la liste tabou TL
 - Mettre le nouveau mouvement (v, θ) dans TL
 - Si $|TL| > t$, supprimer le plus ancien mouvement de TL
 - Mettre à jour Asp ; $itr = itr + 1$. Aller à 1.
-

choisies à chaque itération, influence la qualité de la solution. Il peut être choisi selon le nombre de nœuds non encore colorés et le nombre total de couleurs. Nous pouvons choisir de garder une solution pour chaque nœud déplacé à chaque classe. Dans ce cas, il y a $p = |C_0| \times m$ solutions au total; où $|C_0|$ représente le nombre de nœuds non colorés par l'algorithme constructif. Le paramètre max_itr (nombre maximal d'itérations) ainsi que la taille de la liste tabou influencent le temps d'exécution de l'algorithme et la qualité de la solution obtenue. Ils doivent donc être bien choisis afin de trouver un compromis entre le temps d'exécution et la diversité des solutions.

Une fois l'attribution des canaux de réception terminée, le nœud puits informe chaque nœud du réseau du "canal à utiliser". Ainsi, les nœuds commutent leurs radios vers leurs canaux et peuvent commencer la réception des données.

2.5.2 Restauration de la connectivité dans le RCSF

Après avoir terminé la phase de l'allocation initiale, nous allons dans cette partie nous intéresser au problème du recouvrement de panne afin de restaurer la connectivité dans le réseau tout en minimisant le nombre de capteurs à déplacer et la distance totale parcourue par ces capteurs. Pour ce faire, nous utilisons la technique de points de Steiner (SPs) pour trouver les nouveaux emplacements des nœuds suite à la détection de la panne d'un nœud d'articulation (AP).

Quand un nœud détecte que son niveau d'énergie tombe en-dessous d'un seuil prédéfini, il envoie un message *Failure_report* au puits. La procédure de recouvrement est déclenchée immédiatement si le nœud qui a envoyé le *Failure_report* est un AP.

Commençons par construire un polygone qui relie tous les voisins à deux sauts du nœud d'articulation défaillant (AP_f) comme illustré par la figure 2.1a où les nœuds bleus représentent les voisins à deux sauts. L'heuristique de détermination des SPs fonctionne comme suit :

1. Pour chaque paire de nœuds adjacents i et j du polygone, un triangle Δ_{ij} est formé avec AP_f , i et j comme sommets.
2. Pour chaque triangle Δ_{ij} ainsi construit, le SP est le point se trouvant à l'intérieur du Δ_{ij} et minimisant la distance

$$d = dist(SP, i) + dist(SP, j) + dist(SP, AP)$$

Pour déterminer le SP, nous utilisons l'algorithme d'approximation k -LCA [76] avec k égal à 3. La figure 2.1b montre comment le SP (nœud noir) est déterminé.

3. Parmi tous les SPs trouvés, nous supprimons les SPs ne couvrant aucun sommet du polygone (c'est à dire $\forall i$ sommet du polygone $dist(i, SP) > R_c$) et les SPs redondants (couvrants des sommets couverts par d'autres SPs).
4. S'il reste encore des sommets du polygone (voisins à deux sauts) non couverts, nous formons un nouveau polygone avec ces sommets et les SPs retenus. Nous appliquons les étapes 1 à 3 pour trouver les nouveaux SPs.
5. Ce processus est répété jusqu'à ce que tous les voisins à deux sauts soient couverts.

La figure 2.1c illustre les emplacements des tous les SPs (les points noirs), où les nœuds bleus représentent les voisins à deux sauts et les nœuds verts les voisins à un saut.

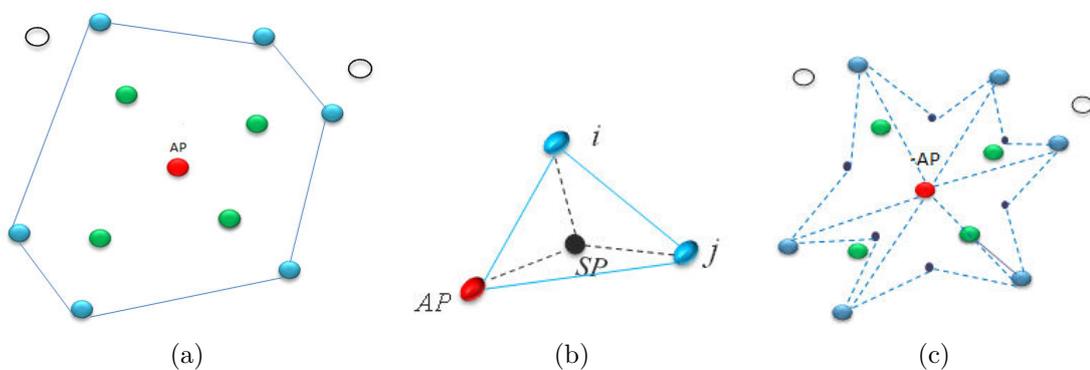


Figure 2.1 : Recherche des points de Steiner

Une fois l'ensemble minimal des SPs SP_set_{min} (donnant les positions des nœuds à déplacer) déterminé, nous passons à la deuxième étape de la phase de la restauration de la connectivité qui consiste à déplacer effectivement les nœuds vers leurs nouveaux emplacements. Dans cette étape, nous utilisons une autre heuristique pour élire les nœuds à déplacer de sorte à minimiser la distance totale parcourue par ces nœuds. Selon le nombre de SPs déterminés, deux scénarios peuvent avoir lieu :

- **Scénario 1** : $|SP_set_{min}| \leq |H_1(AP_f)|$, où $H_1(AP_f)$ représente les voisins à un saut de AP_f . Dans ce cas, chaque SP sera occupé par un voisin à un saut. L'élection des nœuds à déplacer se fait en utilisant l'heuristique suivante :

1. Déterminer d'abord le nœud $v_1 \in H_1(AP_f)$ et le point de Steiner $SP_1 \in SP_set_{min}$ qui donnent la valeur :

$$d(SP_1, v_1) = \max_{SP_i \in SP_set_{min}} (\min_{v_j \in H_1(AP_f)} \text{dist}(SP_i, v_j)) \quad (2.30)$$

2. Ajouter v_1 à l'ensemble des nœuds remplaçants R_set .
 3. Retirer v_1 de la liste des voisins éligibles pour le remplacement et SP_1 de l'ensemble SP_set_{min} .
 4. Reprendre les étapes de 1 à 3 sur l'ensemble $H_1(AP_f) \setminus v_1$.
 5. Répéter toutes les étapes jusqu'à ce que $SP_set_{min} = \emptyset$.
- **Scénario 2** : $|SP_set_{min}| > |H_1(AP_f)|$. Tous les voisins à un saut ainsi qu'un certain nombre de voisins à deux sauts, se déplacent vers les emplacements des SPs. Nous procédons de la même façon que le premier scénario sauf que nous incluons les nœuds à deux sauts. Nous vérifions à chaque étape que le nombre de nœuds à deux sauts déplacés reste inférieur à $|SP_set_{min}| - |H_1(AP_f)|$ qui représente le nombre maximal de nœuds à deux sauts à déplacer. Les nœuds à deux sauts déplacés seront remplacés en cascade en utilisant le protocole DARA [78].

La figure 2.2 illustre les étapes de réorganisation du réseau, où les nœuds verts représentent les voisins à un saut, les nœuds bleus les voisins à deux sauts et les nœuds gris les emplacements vides qui étaient occupés par des nœuds avant le recouvrement de la panne de l'AP.

La technique de déplacement proposée nous permet de connecter les différents segments disjoints du réseau en connectant tous les voisins à deux sauts du nœud défaillant. En plus, cette technique empêche l'apparition d'un nouveau nœud d'articulation en améliorant la connectivité dans le voisinage de la panne. Ainsi, les nœuds qui ont participé

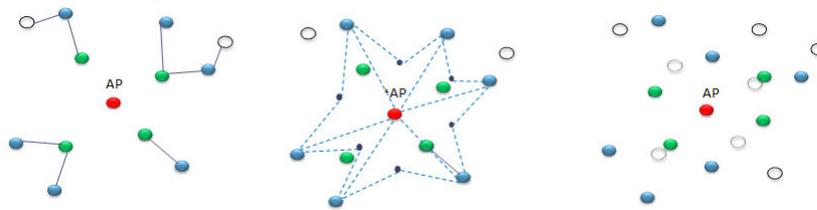


Figure 2.2 : Réorganisation d’un RCSF après une panne

au recouvrement de la panne, consommant plus d’énergie pour se déplacer, ne participeront pas à un autre recouvrement dans le futur proche. Ceci pour empêcher qu’un nœud capteur épuise son énergie en se déplaçant d’un emplacement à un autre.

2.5.3 Réattribution des canaux

Après le déroulement de la procédure de recouvrement décrite dans le paragraphe précédent, l’allocation optimisée des canaux n’est plus respectée. En effet, le déplacement des nœuds remplaçants induit une nouvelle topologie du réseau avec l’apparition de nouveaux liens interférents et la disparition d’autres liens. Une réallocation des canaux autour du nœud défaillant devient nécessaire afin de maintenir une allocation optimisée des canaux dans le RCSF. A cette fin, nous proposons une heuristique de recoloration basée sur l’algorithme *Tabucol* [77].

Soit f_0 le nombre initial de conflits (le nombre de nœuds interférents utilisant les mêmes canaux) avant la réorganisation du réseau. La procédure de la recoloration devrait garantir que le nombre de conflits ne dépasse pas le nombre initial f_0 . Cette procédure fonctionne comme suit :

- Recolorer les nœuds déplacés un à un et calculer à chaque fois le nombre de conflits :
 - Si ce nombre est inférieur ou égal à f_0 , alors fin de la recoloration.
 - Sinon, passer au nœud suivant,
- Si le nombre de conflits reste toujours supérieur à f_0 , recolorer les voisins à un saut des nœuds déplacés ;
- Répéter le processus jusqu’à atteindre le nombre maximal d’itérations.

Le processus de recoloration est décrit en détail dans l’Algorithme 2, où TL représente la liste tabou incluant les mouvements tabous, $r(s)$ le nombre de nœuds recolorés dans la solution actuelle s et s^* la solution de coloration optimisée.

Le nombre maximal d’itérations est égal à $\Delta \times n_m$, où Δ est le degré maximal de voisinage dans le réseau ($\Delta = \max_{v \in V} |H_1(v)|$) et n_m le nombre de nœuds déplacés. Un

Algorithm 2 Algorithme de recoloration des nœuds

Initialisation :

- $s = \{C_1, \dots, C_m\}$;
- Selon le degré de conflits, les nœuds réorganisés, sauf des nœuds dont le degré est 0, sont triés dans une liste L commençant avec le degré le plus haut.
- Initialiser max_itr
- $itr = 1, c(s) = 0$;

while $STOP = false$ **et** $itr < max_itr$ **do**

Générer $N'(s)$ de m solutions choisies de $N(s) = \{s' | s' \text{ est obtenue de } s \text{ en déplaçant le premier nœud } v \text{ de } L \text{ de } C_i \text{ à } C_j, j = 1, \dots, i - 1, i + 1, \dots, m\}$;

$s' = \underset{s' \in N'(s)}{argmin} [(f(s'), r(s'))]$;

if $f(s') < f_0$ **then**

 break;

end if

if $f(s') \leq f(s^*)$ **then**

$s^* = s'$;

else

 Ajouter le voisin $u \in C_i \setminus TL$ de v avec le degré de conflits le plus grand à la fin de L ;

$s = s'$;

 Si v n'est pas un nœud déplacé $r(s) = r(s) + 1$;

 Supprimer v de L et mettre à jour la liste tabou TL ;

$itr = itr + 1$;

end if

end while

tel choix permet à l'heuristique de trouver une solution acceptable où tous les voisins à un saut sont recolorés au moins une fois et limite la propagation du processus de recoloration.

À ce stade de notre étude, le problème d'optimisation multi-objectif est résolu d'une manière centralisée au niveau du puits et la décision concernant le processus de recouvrement de la panne est prise. Cependant, les nœuds concernés par la panne et son recouvrement restent inconscients de cette décision. Dans l'approche PFR, la décision de recouvrement de la panne est incluse dans un message *Recovery* et expédiée au réseau. Ce message contient essentiellement les informations concernant la liste de nœuds à déplacer et/ou à recolorer. Comme l'AP est encore fonctionnel, il communique le message à tous les segments du réseau.

2.5.4 Synthèse

L'approche préventive PFR permet de déplacer tous les nœuds capteurs avant la panne de l'AP. Ceci permet d'éviter la segmentation du réseau en renforçant la connectivité autour de l'AP. Cependant, cette approche ne peut être préconisée que si la panne de l'AP est causée par l'épuisement de la ressource énergétique. Pour les autres types de pannes, le problème reste soulevé. Considérons que le seul nœud qui relie les différents segments du réseau est le nœud d'articulation AP dont nous discutons la panne, la question qui se pose est comment le nœud puits peut communiquer les informations à tous les nœuds concernés par le recouvrement de la panne. La figure 2.3 montre l'exemple d'un RCSF divisé en trois segments après la panne de l'AP. Ainsi, nous sommes face à un problème de segmentation du réseau où le nœud puits ne peut pas atteindre les nœuds appartenant à un segment autre que le sien. Plusieurs alternatives peuvent alors être proposées pour résoudre ce problème. Nous avons opté pour une approche curative, appelée Rotating Nodes based Failure Recovery (RNFR), laquelle sélectionne un ensemble des nœuds pour transmettre l'information de recouvrement vers les nœuds impliqués dans la restauration de la connectivité.

2.6 Rotating Nodes based Failure Recovery : RNFR

Dans ce qui suit, nous présentons notre nouvelle approche RNFR (*Rotating Nodes based Failure Recovery*) dont l'objectif est de restaurer la connectivité dans le réseau après l'occurrence d'une panne et la segmentation du réseau en segments isolés. Ainsi, le réseau doit pouvoir détecter la défaillance des nœuds d'articulation, la segmentation du

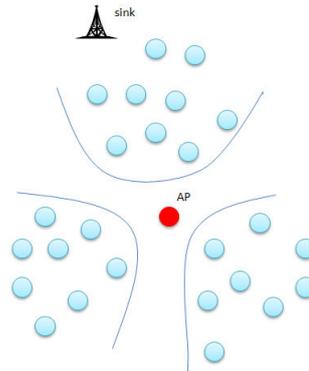


Figure 2.3 : Topologie d'un RCSF après la panne d'un AP

réseau et le déclenchement de la procédure de recouvrement de la panne.

2.6.1 Initialisation du réseau

La phase d'initialisation est la même que PFR. Dans cette phase, les tables de voisinage sont construites, l'ensemble des nœuds d'articulation est déterminé et l'allocation initiale des canaux est faite. Durant le fonctionnement normal, les nœuds collectent des informations dans leurs zones de couverture et envoient les données périodiquement au nœud puits. Après l'occurrence d'une panne au niveau d'un AP, le réseau se trouve divisé en plusieurs segments isolés. La détection et le recouvrement de la panne deviennent une exigence pour restaurer le bon fonctionnement du réseau.

2.6.2 Détection de la panne d'un nœud d'articulation

La première différence avec l'approche PFR est la détection de la panne d'un AP. Avec PFR, un AP s'autoproclame défaillant lorsque son niveau énergétique descend en-dessous d'un certain seuil. En revanche avec RNFR, la panne d'un AP est détectée par ses voisins à un saut. En effet, périodiquement le nœud puits met à jour les routes et chaque nœud vérifie si ses voisins sont toujours en vie en consultant sa table de voisinage construite pendant l'établissement des routes. Quand un nœud v ne reçoit pas le message de mise à jour de la route de l'un de ses voisins u , v envoie des paquets *heartbeat*. Si u ne répond pas, v déduit la défaillance de u et envoie un message *failure_report* au puits pour l'informer de cette panne. Si un rétablissement de la connectivité est nécessaire, tous les voisins à un saut et à deux sauts commutent leurs radios vers un canal de contrôle prédéfini. Une fois le message *failure_report* reçu, le nœud puits vérifie si le nœud défaillant est un nœud d'articulation ou pas. Si oui, il déclenche la procédure de la restauration de connectivité

décrite dans les sections 2.5.2 et 2.5.3.

Contrairement à l'approche PFR, un nouveau problème surgit. Suite à la panne, le réseau se retrouve diviser en segments disjoints. Comment informer les nœuds, concernés par le recouvrement de la panne de l'AP, de la décision du nœud puits alors qu'ils appartiennent à des segments déconnectés de ce dernier ? Comme solution, nous proposons de déplacer un certain nombre de nœuds de sorte que les informations puissent atteindre tous les segments disjoints du réseau.

2.6.3 Rotation des nœuds frontaliers

La restauration de la connectivité dans l'approche RNFR est basée sur l'idée de faire participer un ensemble des nœuds dans un mouvement rotatif. Un certain nombre de nœuds frontaliers de chaque zone primaire sont élus pour transmettre de proche en proche les données de restauration aux zones adjacentes déconnectées. La première étape concerne les nœuds frontaliers de la zone primaire (zone reliée au nœud puits). La figure 2.4 illustre les étapes de rotation et de déplacements des nœuds capteurs pour transmettre les informations de recouvrement et restaurer la connectivité du réseau.

En effet, les zones disjointes peuvent contenir des nœuds capteurs impliqués dans le processus de restauration et qui ne sont pas informés de la décision de recouvrement suite à la défaillance de l'AP. Ainsi, pour propager l'information de recouvrement d'une zone à une autre, le puits élit dans chaque zone i deux nœuds appelés $LBRN_i$ (Left Border Rotating Node) et $RBRN_i$ (Right Border Rotating Node), où $LBRN_i$ (respectivement $RBRN_i$) représente le voisin à un saut de l'AP défaillant se situant le plus à gauche (respectivement le plus à droite) de la zone i comme le montre la figure 2.4a.

Les nœuds $LBRN_i$ sont en charge de transmettre l'information de recouvrement aux nœuds voisins de la même zone en pivotant vers les emplacements des $RBRN_i$. Ces derniers se déplacent vers les positions des $LBRN_i$ des zones déconnectées pour leurs communiquer l'information de recouvrement. Le pivotement des nœuds se fait en cascade d'une zone à une autre dans le sens des aiguilles d'une montre. Tous les nœuds frontaliers $LBRN_i$ et $RBRN_i$ représentent l'ensemble des nœuds en rotation RN_set (Rotating Nodes set).

Si dans la zone i , il existe un certain nombre de nœuds $\in H_2(AP_f)$, voisins à deux sauts de l'AP défaillant, et qui ne sont pas couverts ni par $LBRN_i$ ni par $RBRN_i$, le puits élit dans la zone i un ensemble minimal de nœuds $\in H_1(AP_f)$ (MN_i) qui couvre les nœuds $H_2(AP_f)$. Les MN_i de toutes les zones représentent l'ensemble MN_set (Members

set). On définit l'ensemble IR_set (Information Recovery set) comme étant l'ensemble des nœuds responsables de la communication de l'information de recouvrement ($IR_set = RN_set \cup MN_set$).

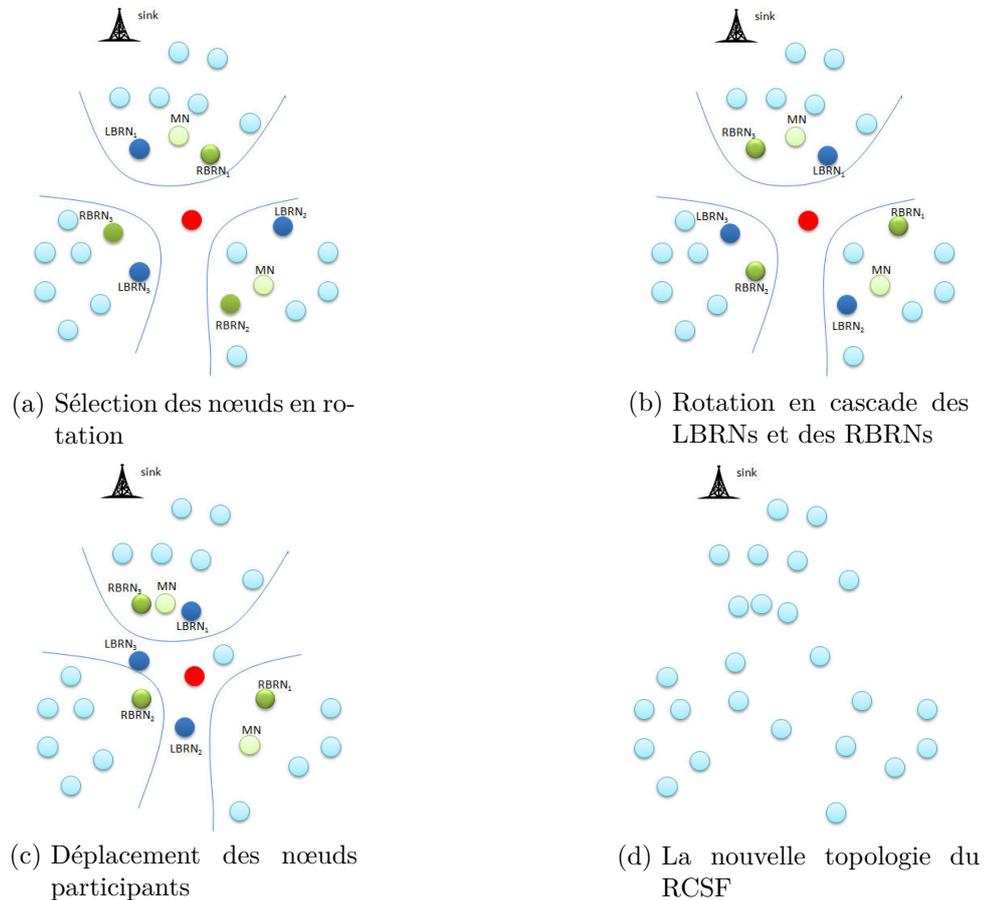


Figure 2.4 : Recouvrement basé sur la rotation des nœuds

Une fois le IR_set formé, la procédure de transmission de l'information de recouvrement vers les différentes zones isolées du réseau s'exécute comme suit :

- Le nœud puits envoie le IR_set et les informations de recouvrement au $LBRN_1$ de la zone primaire.
- Le $LBRN_1$ se déplace vers la position de $RBRN_1$ et transmet en se déplaçant les informations aux nœuds appartenant à MN_1 rencontrés sur son chemin ;
- Chaque nœud de MN_1 transmet les données reçues de $LBRN_1$ vers les voisins à deux sauts de l'AP défaillant ;
- Quand le $RBRN_1$ reçoit les informations, il se déplace vers le segment suivant et transmet le message au $LBRN_2$;
- Le processus se répète jusqu'à ce que le $RBRN_{ns}$ du dernier segment ns atteigne la

position du $LBRN_1$ du premier segment (comme l'illustre la figure 2.4b) ;

- Une fois tous les nœuds de l'ensemble RN_set propagent l'information de recouvrement à tous les segments et regagnent leurs positions finales, la restauration de la panne peut avoir effectivement lieu et tous les nœuds participants au recouvrement ($recovery_set$) se déplacent à leurs nouvelles positions (Les positions des SPs calculés tel que décrit dans le paragraphe 2.5.2). Cette étape est donnée par la figure 2.4c.

L'Algorithme 3 décrit en détail la procédure de la rotation des nœuds.

Algorithm 3 Algorithme de rotation des nœuds

```

if  $node\_id = sink$  then
     $i = 1$ ;
    sélectionner( $IR\_set$ );
    envoyer( $recovery\_information, IR\_set, LBRN_1$ );
end if
while  $i \leq ns$  (nombre de segments du réseau) do
    if  $node\_id = LBRN_i$  then
        passer_à( $RBRN_i$ );
    else if  $node\_id = RBRN_i$  then
        passer_à( $LBRN_{i+1}$ );
        diffuser( $recovery\_information, IR\_set$ );
    else if  $node\_id \in MN\_set$  et  $\exists v \in H_1(node\_id)$  then
        envoyer ( $recovery\_information, v$ );
    else
        supprimer ( $recovery\_information, IR\_set$ )
    end if
    if  $node\_id \in recovery\_set$  then
        déplacer_vers( $node\_id, SP_j$ ); ( $SP_j \in SP\_set_{min}$ )
    end if
     $i = i + 1$ ;
end while

```

La figure 2.4d représente la topologie du RCSF après la réorganisation du réseau.

Pour optimiser le processus de recouvrement de panne, nous pouvons prévoir que chaque nœud participant à la restauration de la connectivité se déplace vers son nouvel emplacement dès la réception des informations de recouvrement. Ainsi, avant la fin de la rotation en cascade des nœuds de RN_set , la majorité des nœuds participants au recouvrement auront atteint leurs nouvelles positions. Si le nœud en rotation participe au recouvrement, dès qu'il se trouve à une distance égale à R_c du nœud suivant (appartenant à l'ensemble des nœuds en rotation), il lui transmet l'information de recouvrement et se dirige vers son nouveau emplacement (position d'un SP). De plus, si $LBRN_1$ participe

Ap-proche	Type	Type de panne	Détection de panne	Transmission de l'information de recouvrement	Inconvénients
RNF	Centralisé	Épuisement d'énergie	L'AP avant de défaillir	L'AP lui même	Seulement la panne due à l'énergie est permise
RNFR	Centralisé	Tous les types de pannes	Les voisins du nœud défaillant	Nœuds en rotation	Temps de calcul et de rotation supplémentaires; Complexité due à la sélection des nœuds impliqués dans la rotation

Tableau 2.1 : Tableau comparatif de PFR et de RNFR

au recouvrement, $RBRN_{ns}$ se déplace directement vers la position finale supposée être occupée par $LBRN_1$. Cette optimisation impacte la distance totale à parcourir ainsi que la durée totale de recouvrement.

2.6.4 Synthèse

Les deux approches PFR et RNFR, proposées pour le recouvrement de la panne dans un RCSF d'une manière centralisée, sont préconisées pour le rétablissement de la connectivité dans un réseau segmenté suite à la défaillance d'un nœud d'articulation. Chacune des deux approches PFR et RNFR présentent des avantages et des inconvénients résumés dans le tableau 2.1.

Dans ce qui suit, nous évaluons les performances des approches PFR et RNFR selon des critères de performance définis de sorte à tenir compte à la fois du problème d'optimisation multi-objectif et des différents algorithmes mis en place pour le recouvrement de la panne.

2.7 Évaluation des performances

Les approches PFR et RNFR sont évaluées par simulation en utilisant le simulateur réseau Omnet++ [79]. Pour tous les scénarios, nous considérons des RCSFs déployés dans une zone de $1000m \times 1000m$. Le nombre de nœuds, aléatoirement dispersé dans la zone, varie entre 50 et 250 nœuds. Le rayon de communication est égal à 100 m.

2.7.1 Métriques de performance

Nous évaluons les performances des solutions de recouvrement de panne selon plusieurs critères. Ainsi, nous utilisons les quatre critères suivants pour évaluer la résolution du problème d'optimisation multi-objectif :

1. *Nombre de conflits* : indique le nombre de nœuds interférents qui utilisent les mêmes canaux.
2. *Nombre de nœuds recolorés* : les nœuds auxquels de nouveaux canaux ont été attribués après le recouvrement de la panne.
3. *Nombre de nœuds déplacés* : reflète l'influence de la procédure de rétablissement de la connectivité sur la topologie du RCSF.
4. *Distance totale parcourue* : Cette métrique dénote la distance parcourue par tous les nœuds déplacés pour recouvrir la panne. Cette distance influence alors le temps total de réparation de la panne.

Quatre autres métriques sont utilisées pour mesurer le coût de la stratégie de rotation utilisée pour transmettre les informations à chaque segment isolé du RCSF :

- *Nombre de nœuds en rotation* : indique le nombre de nœuds LBRNs et RBRNs impliqués dans le processus de rotation.
- *Distance totale de rotation* : cette métrique donne la distance parcourue par les nœuds en rotation.
- *Temps de restauration* : C'est le temps total depuis la détection de la panne jusqu'à la fin de la réorganisation du réseau.
- *Nombre de messages de contrôle* : cette métrique influence l'utilisation de la bande passante et l'énergie consommée lors du rétablissement de la connectivité dans le réseau.

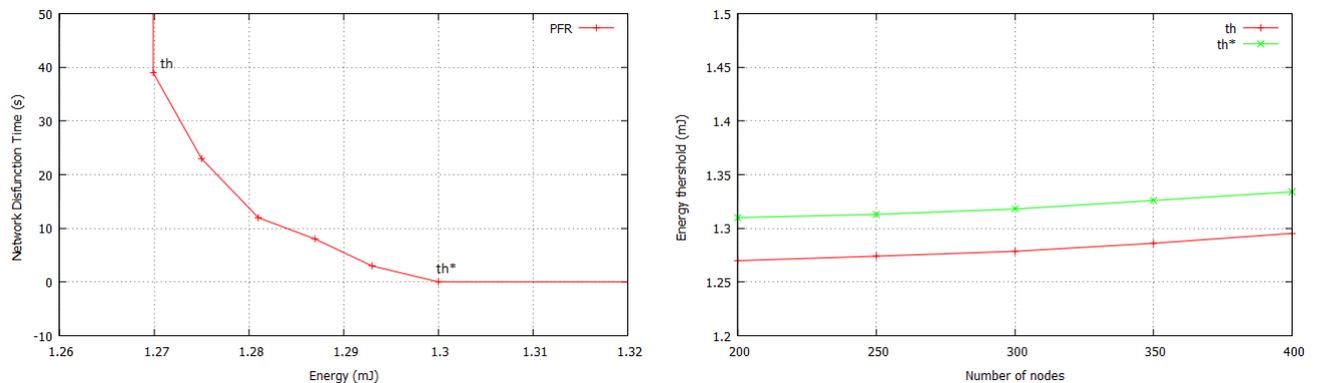
2.7.2 Étude des performances de PFR

Comme mentionné précédemment, le recouvrement de la panne dans PFR se déclenche suite à l'auto-détection du niveau énergétique du nœud d'articulation. Ainsi, nous essayons de déterminer le seuil au-dessous duquel l'AP transmet un message *failure_report* (figure 2.5).

La figure 2.5a montre le temps de dysfonctionnement du réseau en fonction de l'énergie résiduelle du nœud d'articulation. La première valeur th est le seuil qui permet au nœud d'articulation de transmettre le message *failure_report*, attendre la décision du nœud puits

puis communiquer cette décision aux nœuds concernés. Ce seuil est choisi de sorte que l’AP ne tombe en panne qu’après la transmission de la décision de recouvrement. Ainsi, le réseau peu devenir non fonctionnel en attendant que les nœuds déplacés atteignent leurs nouveaux emplacements. Une deuxième valeur th^* représente le seuil qui permet de s’assurer que le nœud d’articulation reste fonctionnel jusqu’à la fin de la réorganisation du réseau.

La figure 2.5b illustre la variation de ces deux seuils en fonction du nombre de nœuds dans le réseaux. Nous remarquons que les valeurs de ces deux seuils augmentent légèrement en augmentant le nombre de nœuds capteurs. Par exemple, dans un réseau à 200 nœuds, la valeur de th est égale à 1.2699 mJ et th^* est égal à 1.31 mJ . Ces deux valeurs deviennent respectivement 1.2953 mJ et 1.334 mJ pour un RCSF à 400 nœuds. Ces résultat sont dus au fait que le recouvrement implique toujours peu de nœuds même si la taille du réseau devient plus important. La petite augmentation des valeurs de seuils est due en général à l’augmentation de la taille de message incluant la décision de recouvrement et à la durée de propagation des messages de contrôle entre l’AP et le puits.



(a) Temps de dysfonctionnement du RCSF en fonction de l’énergie résiduelle de l’AP (nombre de nœuds=200) (b) Seuils énergétiques en fonction du nombre de nœuds

Figure 2.5 : Détermination du seuil énergétique

La figure 2.6 montre les performances des mécanismes de restauration de la connectivité, utilisés dans les approches PFR et RNFR, comparées à une résolution exacte du problème d’optimisation multi-objectif utilisant la méthode Min-Max [74]. Ces performances sont déterminées en fonction du nombre de nœuds dans le réseau. Cette figure dépeint que les heuristiques proposées donnent des résultats proches aux résultats optimaux en particulier dans les RCSFs à petite taille (même nombre de nœuds en conflit pour les réseaux à 15 et à 20 nœuds). Les résultats dévient lentement de l’optimum lorsque la taille du réseau devient plus importante. Cependant, ils suivent le même comportement

que les résultats de Min-Max pour les quatre métriques.

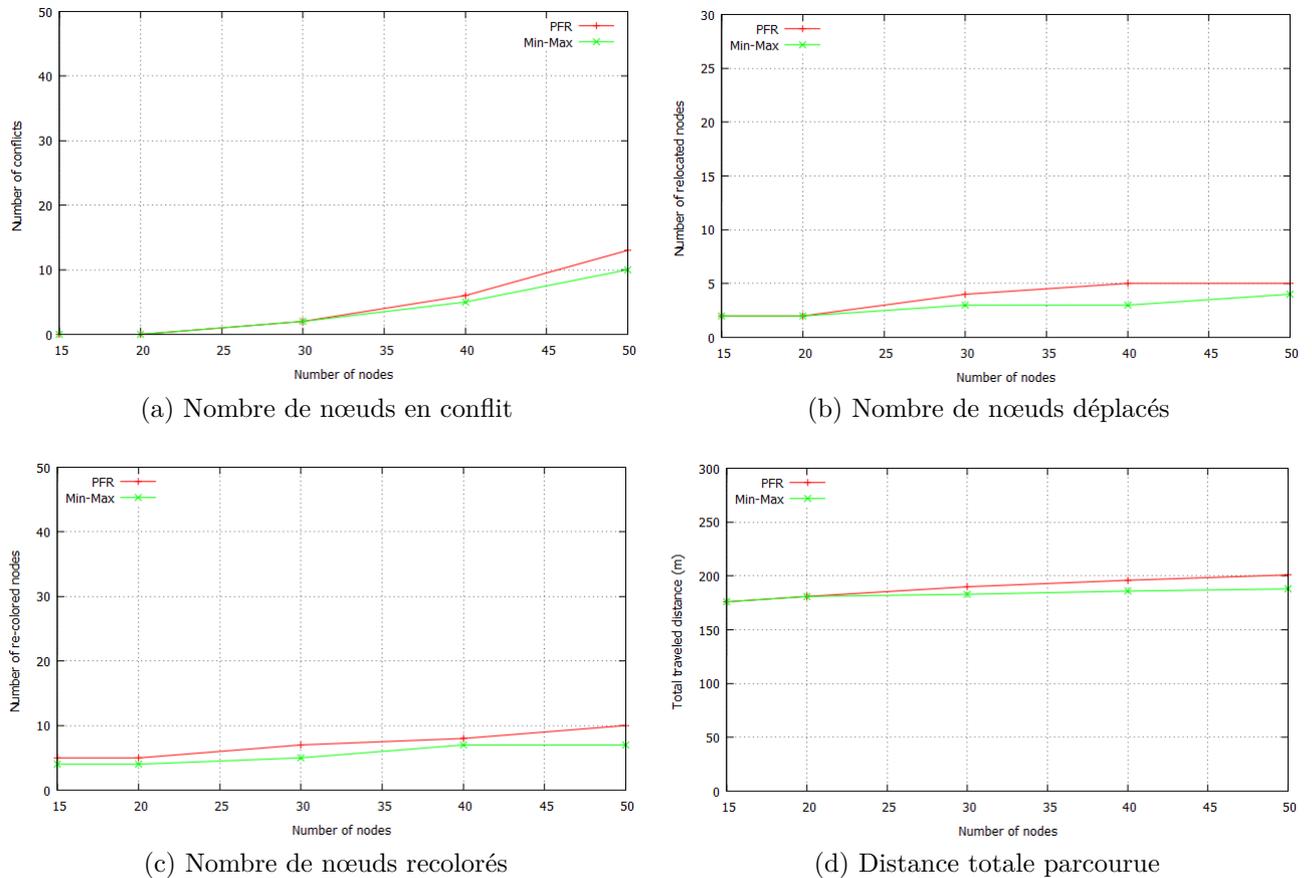


Figure 2.6 : Performances de la solution centralisée par rapport à la solution optimale

Ainsi, la figure 2.6b montre que le nombre de nœuds déplacés devient plus important avec l'augmentation de la taille du RCSF. Ce résultat est dû au fait que le nombre de voisins à deux sauts, utilisés pour déterminer les nouveaux emplacements des nœuds à déplacer, augmente avec l'augmentation de la densité du réseau. Comme une conséquence directe, la distance parcourue totale devient plus importante comme l'illustre la figure 2.6d. Cependant, la distance moyenne parcourue par chaque nœud diminue de 59m dans un réseau de 15 nœuds à 19m dans un réseau à 50 nœuds.

Les figures 2.6a et 2.6c illustrent que même si nous recolorons plus de nœuds, le nombre de nœuds en conflit augmente avec l'augmentation du nombre de nœuds dans le réseau. Ceci s'explique par le fait que plus de nœuds peuvent interférer quand la densité du réseau devient plus importante. Ainsi, les canaux disponibles deviennent moins efficaces pour la réduction des interférences.

La figure 2.7 montre l'impact de l'augmentation du nombre de nœuds dans le réseau

sur le nombre de nœuds en conflit et le nombre de nœuds réaffectés à d’autres canaux après le recouvrement de la panne. Les résultats mis en évidence par la figure 2.7b reflètent le fait que plus le nombre de voisins est grand, plus le nombre de nœuds à recolorer devient important. De plus, quand le nombre de canaux disponibles diminue, nous pouvons recolorer seulement quelques nœuds pour minimiser le nombre de nœuds interférents. Nous observons que si nous utilisons une allocation à 2 canaux, nous recolorons seulement 55 nœuds dans un RCSF à 250 nœuds, tandis qu’une allocation à 4 canaux génère 89 nœuds recolorés. Cependant, le nombre de nœuds en conflit est bien évidemment considérablement inférieur en utilisant 4 canaux (figure 2.7a).

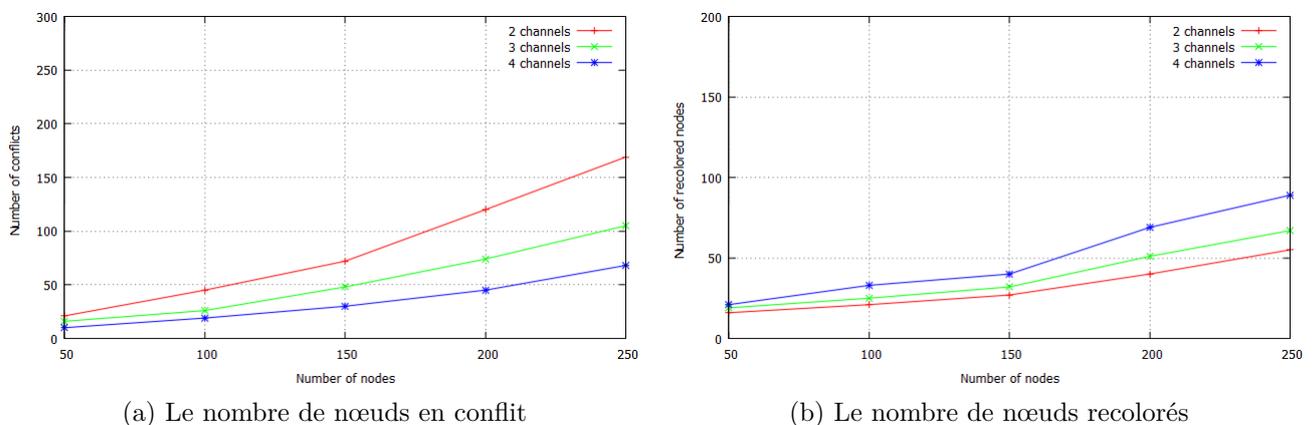


Figure 2.7 : Le nombre de nœuds recolorés et le nombre de conflits en fonction du nombre de nœuds

2.7.3 Étude des performances de RNFR

L’approche RNFR fournit les mêmes résultats que l’approche PFR en termes de nombre de conflits et nombre de nœuds réassignés à de nouveaux canaux après le recouvrement de la panne. Cependant, le nombre de nœuds déplacés ainsi que la distance totale parcourue par ces nœuds diffèrent de ceux générés par PFR en raison de la technique de rotation utilisée par RNFR.

La figure 2.8 illustre le surcoût engendré par le mécanisme de rotation des nœuds dans l’approche RNFR. Ce surcoût est exprimé en termes de nombre de nœuds en rotation et distance totale parcourue par ces nœuds en fonction du nombre de nœuds présents dans le RCSF.

Comme nous pouvons voir sur la figure 2.8a, le nombre de nœuds en rotation diminue avec l’augmentation du nombre total de nœuds. Nous pouvons expliquer ce comportement

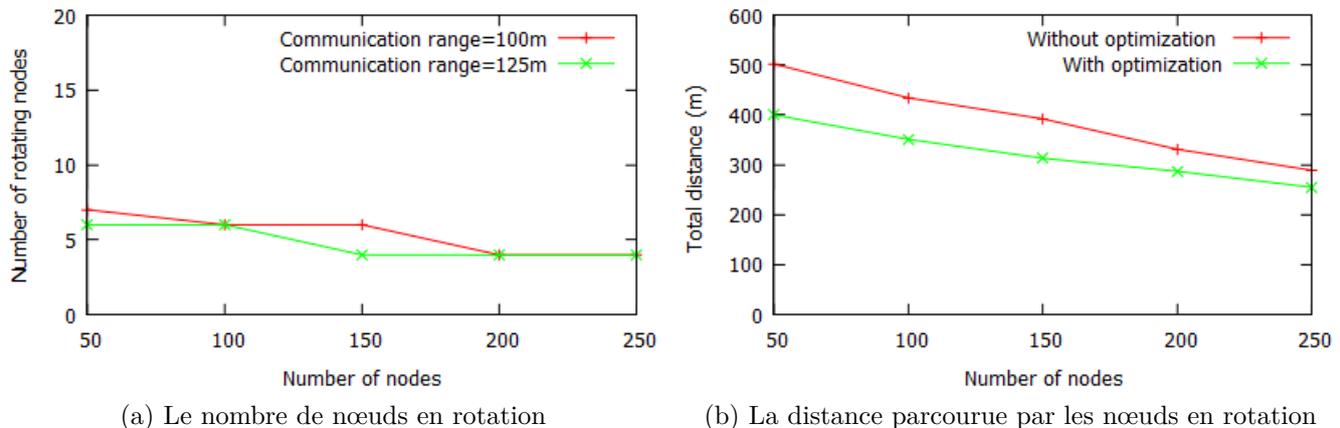


Figure 2.8 : Le surcoût de la rotation des nœuds en fonction du nombre total de nœuds

par le fait que le nombre de segments disjoints devient plus petit quand le RCSF devient plus dense. Il en résulte que peu de nœuds sont nécessaires pour communiquer les informations à tous les segments (7 nœuds en rotation dans un RCSF peu dense comparé à 4 nœuds dans un réseau plus dense). De plus, l'augmentation du rayon de communication (de 100 m à 125 m) peut diminuer le nombre de nœuds en rotation puisque le nombre de segments isolés dépend de la densité du réseau ainsi que de la portée de chaque nœud.

La figure 2.8b présente la distance totale engendrée par le processus de rotation. Cette métrique influence directement la durée totale du processus de recouvrement de la panne. Par ailleurs, le déplacement optimisé réduit considérablement la distance parcourue pendant le processus de rotation des nœuds. Avec la rotation optimisée, la distance diminue de plus de 100 m par rapport à la rotation non optimisée (de 502 m à 400m) dans un RCSF de 50 nœuds. L'optimisation a, en revanche, moins d'influence sur les réseaux denses (la distance parcourue diminue de 289 m à 255 m avec seulement 34 m de différence dans un RCSF à 250 nœuds).

2.7.4 Comparaison des approches PFR, RNFR et RIM

Pour évaluer les performances de nos approches proposées, nous les comparons avec l'approche RIM [38] présentée dans le paragraphe 1.3.3.2.

La figure 2.9 montre les performances des trois approches en termes de nombre de nœuds déplacés et de distance totale parcourue en fonction du nombre total de nœuds capteurs.

Comme indiqué par la figure 2.9a, PFR et RNFR offrent de meilleures performances

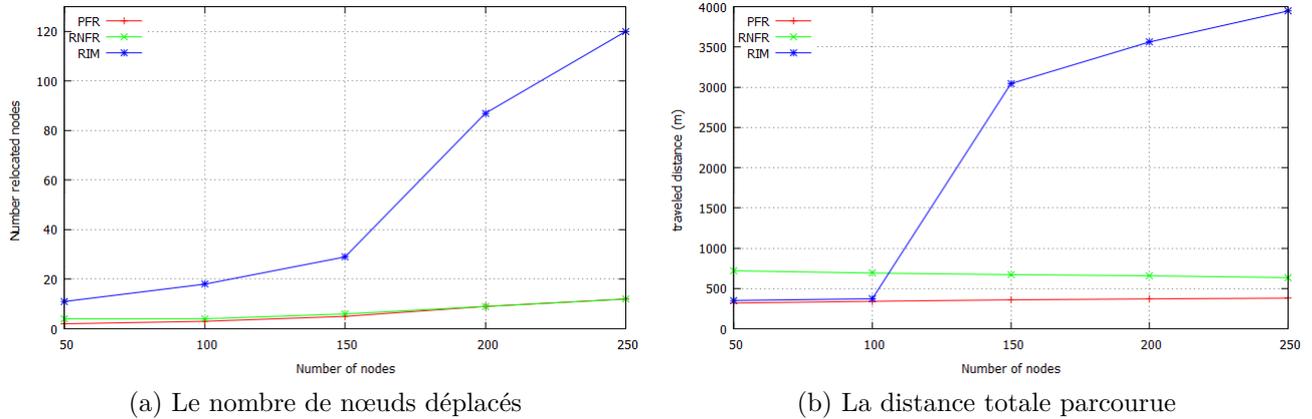


Figure 2.9 : Comparaison des approches PFR, RNFR et RIM

que RIM en terme de nombre de nœuds impliqués dans le processus de déplacement. Il est clair que les deux premières approches restreignent le processus de réorganisation dans le voisinage du nœud défaillant et empêchent alors des déplacements en cascade même dans les réseaux denses. Par contre, RIM implique plus de nœuds en particulier dans les réseaux denses où le degré de voisinage devient plus important. Nous remarquons qu'au delà de 150 nœuds, le nombre de nœuds déplacés augmente d'une manière très rapide. Ceci est dû au fait que le degré de connectivité du réseau devient plus important, alors RIM a besoin de déplacer plus de nœuds pour maintenir ce degré. Par exemple, le nombre de nœuds déplacés dans un réseau à 250 nœuds est égal à 12 nœuds pour PFR et RNFR (4.8% de nœuds), alors que ce nombre atteint 123 nœuds pour RIM (49.2%). L'approche PFR outrepassa aussi RNFR qui doit déplacer plus de nœuds. Ceux nœuds déplacés supplémentaires sont les nœuds en rotation utilisés pour communiquer les informations de recouvrement aux nœuds capteurs. Cependant, les deux approches déplacent le même nombre de nœuds dans les réseaux denses. Ceci s'explique par le fait que tous les nœuds en rotation participent au processus de la restauration de la connectivité.

La figure 2.9b rapporte la distance totale parcourue par les nœuds déplacés pour recouvrir la panne d'un nœud d'articulation. Comme indiqué sur la figure, PFR produit moins de distance que RNFR et RIM. Cette dernière approche produit une distance très importante dans des réseaux denses. Ceci est dû au nombre important de nœuds déplacés qui génèrent une distance plus importante.

La figure 2.10 compare les performances de nos approches PFR et RNFR en termes de nombre de message de contrôle et de temps total de recouvrement en fonction du nombre de nœuds. Le nombre de messages émis prend en considération tous les messages de contrôle émis depuis la détection de la panne jusqu'au rétablissement du fonctionnement

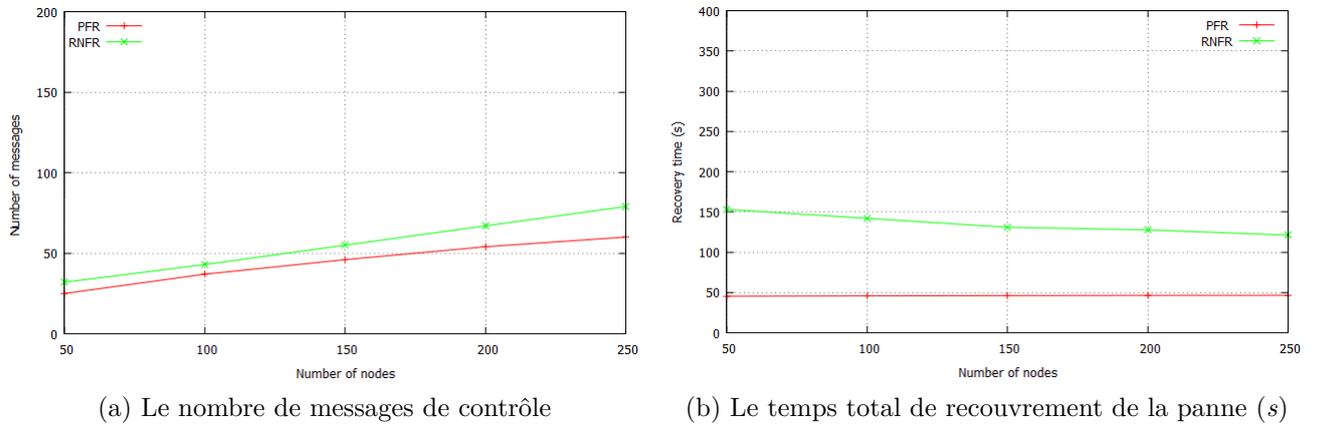


Figure 2.10 : Comparaison de PFR et RNFR en termes de nombre de message et temps de recouvrement

normal du réseau. La figure 2.10a montre que RNFR génère plus de messages que PFR. Cette différence est due à la nécessité de coopération entre les nœuds pour la détection de la panne et la communication des informations entre les différents segments du réseau. Cette différence devient plus importante avec l'augmentation de la taille du réseau. Ceci s'explique par le fait que le processus de recouvrement implique plus de nœuds dans les RCSFs denses. Ainsi, RNFR génère 32 messages dans un RCSF à 50 nœuds et 79 messages dans un réseau à 250 nœuds. Ces nombres sont réduits respectivement à 25 et 60 messages en utilisant l'approche PFR.

La figure 2.10b dépeint le temps nécessaire pour terminer le recouvrement de la panne pour les deux approches PFR et RNFR. Ce temps est calculé à partir de l'instant de détection de la panne jusqu'à ce que les nœuds remplaçants atteignent leurs positions finales. Tous les nœuds se déplacent à une vitesse de $1.5m/s$. Nous remarquons sur la figure 2.10b, que RNFR prend plus de temps pour converger. Ce temps est dû au déplacement supplémentaire (en cascade) effectué par les nœuds en rotation. Pour PFR, le temps de recouvrement reste presque stable avec l'augmentation du nombre de nœuds. Par contre, ce temps diminue pour RNFR. Ce résultat est escompté étant donné que dans les réseaux denses il y a plus de nœuds à déplacer mais ils parcourent moins de distance surtout qu'ils commencent à se déplacer simultanément pour PFR, et sans attendre la fin de la rotation des nœuds pour RNFR.

La comparaison des approches proposées montre bien que PFR offre de meilleures performances que RNFR. Cependant, cette approche n'est applicable qu'en cas de pannes dues à l'épuisement de l'énergie.

Pour exploiter les avantages des approches PFR et RNFR, nous pouvons les combiner

ensemble pour améliorer les performances de notre solution de recouvrement de pannes. Ainsi, nous nous proposons d'appliquer PFR pour les pannes causées par l'épuisement de l'énergie et RNFR pour les autres types de pannes. Ceci est fait dans le but de réduire le surcoût de recouvrement de pannes en termes de temps de recouvrement et de nombre de messages de contrôle échangés.

2.8 Conclusion

Dans ce chapitre, nous nous sommes concentrés sur la proposition d'une solution centralisée de rétablissement de la connectivité dans les RCSFs multi-canaux suite à la panne d'un nœud d'articulation qui a pour conséquence la division du RCSF en plusieurs segments disjoints. Il s'agit d'un problème d'optimisation multi-objectif. Pour le résoudre, nous l'avons d'abord formulé mathématiquement en utilisant la théorie des graphes, en particulier les graphes colorés et les points de Steiner. Dans un contexte centralisé, nous avons proposé une approche préventive PFR (*Preventive Failure Recovery*) et une approche curative RNFR (*Rotating Nodes based Failure Recovery*) pour résoudre ce problème en un temps polynomial. Nous avons enfin évalué les performances des deux approches proposées en utilisant un certain nombre de métriques de performance.

Bien que la solution centralisée soit efficace en termes de calcul et de nombre de messages échangés entre les nœuds capteurs, elle présente un inconvénient majeur dû à la segmentation du réseau en plusieurs segments déconnectés. Cette segmentation a abouti à un surcoût de rotation des nœuds pour transmettre les informations de recouvrement de la panne vers tous les segments disjoints. Ainsi, pour surmonter ce genre de problèmes, nous proposons d'investiguer la possibilité d'une solution distribuée. En effet, en distribué, seuls les nœuds participants au recouvrement de la panne seront concernés par tous les traitements inhérents à la restauration de la connectivité (calcul des SPs, réallocation, *etc*).

Ainsi, dans le chapitre suivant, nous proposerons deux nouvelles approches distribuées pour la résolution du problème de perte de la connectivité dans les RCSFs multi-canaux. Ces approches auront pour objectif le recouvrement de la panne d'un nœud d'articulation tout en palliant les insuffisances dues aux traitements centralisés.

Chapitre 3

Proposition d'approches distribuées pour la restauration de la connectivité dans les RCSFs multi-canaux

3.1 Introduction

Bien que la solution centralisée ait des avantages, comme nous l'avons montré dans le chapitre précédent, elle devient peu efficace avec l'augmentation du nombre de nœuds capteurs dans le réseau. Une telle solution devient donc peu envisageable pour les RCSFs avec un grand nombre de nœuds capteurs. Pour pallier ces insuffisances, nous optons pour la distribution de tous les traitements inhérents au recouvrement de la panne sur les différents nœuds capteurs impliqués.

Nous allons nous focaliser, dans ce chapitre, sur les solutions distribuées pour le recouvrement des panne dans les RCSFs multi-canaux. Nous allons tout d'abord nous situer dans un contexte multi-canal natif (c'est à dire que le canal est alloué indépendamment du trafic échangé sur le réseau). Ensuite, nous nous situerons dans un cas qui tient compte des routes suivies par les paquets de sorte à minimiser les interférences entre deux nœuds voisins dans l'arbre de routage. Enfin, nous évaluerons nos approches et nous les comparerons à la solution centralisée proposée dans le chapitre précédent.

3.2 Communication multi-canal dans un contexte distribué

Dans cette partie, nous nous focalisons sur les protocoles distribués conçus pour la communication multi-canal. Ainsi, nous différencions entre les protocoles d'allocation des canaux qui utilisent seulement les informations de voisinage (solutions natives) de ceux qui utilisent aussi les informations de routage pour mieux attribuer les canaux aux nœuds capteurs (solutions basées sur le routage).

3.2.1 Allocation native des canaux

Dans [80], les auteurs ont proposé trois techniques d'allocation des canaux d'une manière distribuée. Deux des trois protocoles proposés visent à minimiser le nombre de canaux nécessaires pour éliminer complètement les interférences dans le réseau. Le premier protocole attribue un canal à chaque récepteur ; tandis que le deuxième attribue un canal pour chaque lien de transmission, c'est à dire un canal est alloué à chaque paire émetteur-récepteur. Par contre, le troisième protocole MinMax, lui aussi distribué et proposé dans le même article, s'applique dans le cas où le nombre de canaux disponibles est limité. L'objectif principal de MinMax est d'attribuer un canal à chaque lien de transmission tout en minimisant le nombre maximal d'interférences se produisant sur n'importe quel lien. Les auteurs de [80] ont également proposé une technique distribuée basée sur TDMA pour l'ordonnancement des liens afin d'éviter les interférences entre les liens utilisant le même canal dans MinMax.

L'avantage majeur de MinMax est d'atténuer les goulots d'étranglement dans le RCSF où un nœud ou un lien génère un nombre important d'interférences. Cependant, MinMax traite les voisins appartenant au même parent et ceux appartenant à des parents différents de la même façon. Ainsi, il peut attribuer le même canal à deux liens de transmission ayant des émetteurs et des récepteurs totalement différents. Il peut, par contre, donner des canaux différents à deux liens avec le même récepteur alors que ces deux liens ne peuvent jamais transmettre simultanément. Un autre point faible pour ces protocoles est qu'ils n'offrent pas un mécanisme pour la réallocation des canaux en cas de pannes.

3.2.2 Allocation des canaux basée sur le routage

Le protocole le plus connu dans cette catégorie est TMCP (*Tree-based Multi-Channel protocol*)[67], qui est proposé essentiellement pour la collecte de données dans les RCSFs.

L'idée principale de TMCP est de considérer les arbres de routage comme étant des groupes disjoints. Ainsi, il assigne un canal différent aux nœuds de chaque sous-arbre. Ceci permet de router les données des nœuds sources au nœud puits sans commuter les radio le long des routes vers le puits. Cependant, la communication entre les nœuds de deux arbres différents est impossible ce qui met en question l'efficacité de cet algorithme en cas de panne d'un nœud ou de changement du chemin de transmission.

En 2014, le protocole MCC (*Multi-Channel Collection*) [81] est proposé pour la collecte de données dans les RCSFs. MCC incorpore des mécanismes sophistiqués pour la formation d'arbres de routage équilibrés, l'allocation multifréquence des canaux et la planification TDMA synchronisée. Pour l'attribution des canaux, MCC utilise un mécanisme de coloration de graphes pour colorer le graphe de conflits. Ce dernier est construit en mettant un lien entre deux nœuds si l'un d'eux interfère avec le fils de l'autre nœud. Ensuite, si un lien existe entre les deux nœuds, le mécanisme de coloration donne deux couleurs différentes pour ces deux nœuds. Cependant, bien que MCC maximise le débit en utilisant la communication multi-canal et la planification de la transmission, il suppose que la topologie du réseau et la qualité des canaux ne changent pas au cours du temps. Par conséquent, il ne prend pas en compte la défaillance d'un nœud et n'intègre aucun mécanisme pour résoudre le problème dans ce cas.

Le protocole MCRT (*Multi-Channel Real-Time communication*) [82] est proposé dans le but de minimiser le délai de bout en bout dans les RCSFs temps réel. MCRT utilise une stratégie d'allocation des canaux basée sur les flux pour optimiser le nombre de commutations radios afin de minimiser la consommation énergétique et le délai de transmission. Ainsi, il cherche les routes disjointes pour les flux de données entre chaque nœud source et sa destination. Une fois les routes sélectionnées, MCRT essaye d'allouer un canal différent pour chaque flux de données sur une route disjointe pour éliminer ou réduire les interférences entre les différents flux. Le point faible de ce protocole est que chaque nœud (sauf la destination) ne peut appartenir qu'à une seule route.

3.2.3 Synthèse

Tous les mécanismes d'allocation des canaux proposés ne prennent pas en considération le rétablissement de la connectivité après l'occurrence d'une panne dans le réseau. Ceci peut affecter les performances du RCSF même avec l'utilisation la communication multi-canal d'où la nécessité d'utiliser un mécanisme qui combine à la fois la communication multi-canal et le recouvrement de pannes. Ainsi, nous proposons deux approches distribuées pour la restauration de la connectivité dans un RCSF multi-canal. Dans la

première approche, appelée CR-MC (*Connectivity Restoration for Multi-Channel WSNs*), nous adoptons un mécanisme d'allocation des canaux où seule l'information de voisinage est utilisée pour minimiser les interférences entre les nœuds voisins. Dans la deuxième approche, appelée CR-RMC (*Connectivity Restoration for Routing based Multi-Channel WSNs*), nous exploitons les arbres de routage pour améliorer l'efficacité de notre algorithme d'allocation des canaux et ainsi minimiser le taux d'interférences dans le réseau.

Néanmoins, nous optons pour une technique d'allocation de canaux qui minimise le nombre de liens conflictuels pour chaque nœud au lieu de minimiser le nombre total de nœuds en conflit dans le réseau. Un lien entre deux nœuds i et j est dit conflictuel si $dist(i, j) \leq R_c$ et les deux nœuds sont sur le même canal. L'idée est de permettre à chaque nœud du réseau d'avoir un nombre minimal de nœuds voisins avec lesquels il est en conflit plutôt que de minimiser le nombre total de nœuds en conflit dans tout le réseau.

En effet, si nous considérons un scénario d'un réseau à six nœuds et deux canaux (chaque couleur représente un canal), une configuration de canaux qui minimise le nombre de nœuds en conflit dans ce réseau est donnée par la figure 3.1a. Dans cette figure le nœud 2 est en conflit avec trois de ces voisins qui émettent sur le même canal, alors que le nœud 1 et 6 ne sont en conflit avec aucun voisin. Le nombre total de nœuds en conflit dans tout le réseau est égal à quatre.

Si maintenant nous optons pour l'affectation des canaux présentée dans la figure 3.1b, nous nous trouvons dans le cas où les nœuds 2 et 6 auront deux liens en conflit et le nœud 1 aura deux liens en conflit, alors que le nombre de nœuds en conflit dans tout le réseau est égal à six. Ainsi, nous avons essayé de minimiser le nombre de conflits pour chaque nœud du réseau plutôt que se trouver dans un cas de figure où certains nœuds ont très peu de liens conflictuels et d'autres nœuds se trouvent en conflit avec un grand nombre de voisins.



Figure 3.1 : Configurations des canaux

Il en résulte que dans un contexte distribué, le problème d'allocation des canaux devient un problème d'optimisation du nombre de liens conflictuels par nœud au lieu d'un problème d'optimisation du nombre total de nœuds en conflit comme c'est le cas centralisé.

3.3 Connectivity Restoration for Multi-Channel WSNs : CR-MC

Dans cette approche, nous nous plaçons dans un contexte multi-canal où l'allocation des canaux se fait d'une manière distribuée. L'objectif principal de notre approche est de restaurer le bon fonctionnement du réseau tout en limitant la procédure de recouvrement autour du nœud défaillant. Si nous nous intéressons au problème de recouvrement de pannes dans un RCSF multi-canal dans un contexte distribué, ce problème peut être énoncé comme suit :

1. Minimiser le nombre de liens conflictuels pour chaque nœud capteur.
2. Minimiser le nombre de nœuds à déplacer.
3. Minimiser le nombre de nœuds à allouer un autre canal.
4. Minimiser la distance parcourue par les nœuds déplacés.

Les quatre objectifs seront formulés mathématiquement dans la section suivante.

3.3.1 Formulation mathématique du problème

Pour formuler notre problème d'optimisation multi-objectif décrit ci-haut, nous modélisons d'abord notre RCSF par un graphe d'interférence coloré dont les couleurs utilisées représentent les canaux assignés aux nœuds capteurs. Nous définissons :

- $G_0 = (V_0, E_0)$: le graphe coloré représentant le réseau multi-canal initial.
- E_0 : l'ensemble des liens d'interférence.
- V_0 : l'ensemble des nœuds capteurs.
- $G'_0 = (V'_0, E'_0)$: le graphe représentant le RCSF après le recouvrement et défini de la même façon que G_0 .
- $c_G(v)$: la fonction qui donne la couleur d'un sommet v dans un graphe G .
- $pos_G : V \mapsto \mathbb{R} \times \mathbb{R}$: la fonction qui retourne les coordonnées d'un sommet dans un graphe G .

Pour formuler notre premier objectif f_1 représentant le nombre de liens conflictuels pour chaque nœud du réseau dans un contexte distribué, nous vérifions d'abord si un lien est conflictuel ou pas par l'introduction de la fonction $\delta : V'_0 \times V'_0 \mapsto [0, 1]$ définie par :

$$\delta(v, u) = \begin{cases} 1 & \text{si } e_i(v, u) \in E'_0 \wedge c_{G'_0}(v) = c_{G'_0}(u) \\ 0 & \text{sinon.} \end{cases} \quad (3.1)$$

Le nombre de liens conflictuels pour un nœud v est déterminé en calculant le nombre de liens conflictuels entre ce nœud et tous ses voisins :

$$f_1^v = \sum_{\forall u \in V'_0} \delta(v, u). \quad (3.2)$$

Ainsi, f_1 est définie par le vecteur de fonctions à optimiser $[f_1^v]$ comme suit :

$$f_1 = [f_1^v], \forall v \in V'_0 \quad (3.3)$$

Pour ce qui est du deuxième objectif, nous définissons la fonction $\varphi(v)$ pour déterminer si un nœud est déplacé ou pas :

$$\varphi(v) = \begin{cases} 1 & \text{si } pos_{G'_0}(v) \neq pos_{G_0}(v) \\ 0 & \text{sinon.} \end{cases} \quad (3.4)$$

D'où, le nombre total de nœuds déplacés est déterminé par :

$$f_2 = \sum_{\forall v \in V'_0} \varphi(v). \quad (3.5)$$

De la même façon, nous définissons $\gamma(v) : V'_0 \times V'_0 \mapsto [0, 1]$, la fonction qui indique si un nœud est recoloré ou pas :

$$\gamma(v) = \begin{cases} 1 & \text{si } c_{G'_0}(v) \neq c_{G_0}(v) \\ 0 & \text{sinon.} \end{cases} \quad (3.6)$$

Le nombre total de nœuds recolorés est défini par la fonction objectif f_3 comme suit :

$$f_3 = \sum_{\forall v \in V'_0} \gamma(v). \quad (3.7)$$

Le dernier objectif dans notre problème f_4 consiste à minimiser la distance totale parcourue par les nœuds déplacés. Soient $pos_{G_0}(v) = (x, y)$ et $pos_{G'_0}(v) = (x', y')$ les positions respectives du nœud v avant et après la réorganisation du réseau. La distance parcourue par ce nœud est calculée par :

$$dist(v) = \sqrt{(x' - x)^2 + (y' - y)^2} \quad (3.8)$$

La fonction objectif f_4 est alors donnée par :

$$f_4 = \sum_{\forall v \in V'_0} \varphi(v) dist(v) \quad (3.9)$$

Après la définition des quatre fonctions objectifs, la fonction multi-objectif f est définie comme suit :

$$f = [f_1, f_2, f_3, f_4]^T \quad (3.10)$$

Le problème d'optimisation multi-objectif revient à minimiser f sous les contraintes que le graphe soit connecté et que tous les nœuds soient colorés. Ce problème sera alors énoncé par :

$$minimiser f \quad (3.11)$$

$$G'_0 \text{ connecté} \quad (3.12)$$

$$1 \leq c_{G'_0}(v) \leq m, \forall v \in V'_0 \quad (3.13)$$

Comme nous l'avons expliqué dans le chapitre 2, la résolution exacte ne peut être appliquée dans le contexte d'un RCSF réel compte tenues des ressources limitées des capteurs et de la taille du réseau qui peut atteindre des centaines, voire des milliers de nœuds. Ainsi, nous présentons dans ce qui suit notre approche CR-MC (*Connectivity Restoration for Multi-Channel WSNs*), qui se base sur des heuristiques relatives à la réorganisation du réseau et la réallocation des canaux, pour restaurer la connectivité dans un RCSF où l'allocation des canaux se fait d'une manière distribuée.

L'approche CR-MC utilise les mêmes hypothèses introduites dans le paragraphe 2.5 du chapitre 2.

3.3.2 Phase d'initialisation du réseau

Pendant la phase d'initialisation, les nœuds construisent leurs tables de voisinage en échangeant des messages *hello*. Ces messages contiennent des informations sur les identifiants et les positions des nœuds. Pour l'approche CR-MC chaque nœud exécute l'algorithme distribué proposé dans [83] afin de déterminer s'il est ou non un nœud d'articulation. Si c'est le cas, il en informe ses voisins.

Les nœuds doivent aussi sélectionner leurs canaux de réception. La technique utilisée pour l'allocation initiale des canaux se base sur une heuristique simple de coloration de graphes où chaque couleur représente un canal. La procédure de coloration s'exécute comme suit :

- A l'initialisation du processus de coloration, une couleur fictive ayant l'identifiant 0 (c_0) est affectée à tous les nœuds ;
- Le nœud i ayant l'identifiant (ID) le plus petit dans son voisinage prend la plus petite couleur (ayant l'indice le plus petit) non encore utilisée. Les autres nœuds ne changent pas leurs couleurs ;
 - Si aucune couleur n'est disponible, i choisit la couleur la moins utilisée par ses voisins ;
- Le nœud i informe ses voisins de sa nouvelle couleur.
- Le processus se répète jusqu'à affecter une couleur ($\neq c_0$) à chaque nœud ;

Une fois l'allocation des canaux terminée, les capteurs commencent la collecte et l'envoi des données vers le nœud puits, en utilisant les canaux déjà affectés, jusqu'à l'occurrence d'une panne.

Contrairement au cas centralisé, où la panne est annoncée par le puits, dans l'approche CR-MC la détection de la panne se fait d'une manière distribuée. En effet, périodiquement, les nœuds vérifient l'état de leurs voisins par l'envoi de plusieurs messages *heartbeat*. Si un nœud tombe en panne, tout voisin u à un saut peut détecter sa défaillance. En outre, si le nœud défaillant v est un nœud d'articulation, u informe ses propres voisins pour qu'ils commutent leurs radios vers un canal de contrôle prédéfini et déclenchent le processus de recouvrement de panne.

3.3.3 Réorganisation du RCSF

L'approche CR-MC exploite les mêmes techniques utilisées par la solution centralisée pour la recherche de nouveaux emplacements à occuper par les nœuds remplaçants (Steiner Points SPs), mais d'une manière distribuée.

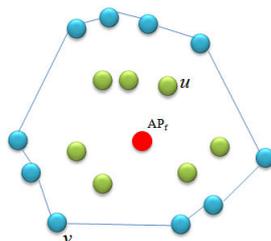


Figure 3.2 : Formation du polygone pour la détermination des SPs

Les SPs sont déterminés par tous les voisins à un saut du nœud d'articulation défaillant AP_f en construisant un même polygone avec tous les voisins à deux sauts de l'AP en question comme le montre la figure 3.2, où les nœuds verts représentent les nœuds à un saut et les nœuds bleus les voisins à deux sauts. Tous les nœuds à un saut ont besoin donc des positions de tous les nœuds à deux sauts pour construire ce polygone. Il est clair sur la figure 3.2 que le nœud u voisin à un saut de AP_f ($u \in H_1(AP_f)$) est voisin à trois sauts du nœud v appartenant au polygone ($v \in H_2(AP_f)$). Il en résulte que pour calculer correctement les SPs dans un contexte distribué, tous les voisins à un saut de l'AP en panne auront besoin des informations de leurs voisins à trois sauts.

Compte tenu de la segmentation du RCSF en raison de la panne de l'AP, les informations de voisinage ne peuvent être échangées après l'occurrence de la panne. Ainsi, les positions des voisins à trois sauts doivent être envoyées pendant la phase d'initialisation du réseau.

Une fois que tous les voisins à un saut de l'AP ont calculé les SPs et identifié les nœuds remplaçants, ils notifient leurs voisins (voisins à deux sauts de l'AP) de leurs nouvelles positions.

3.3.4 Réattribution des canaux

Après la fin du processus de réorganisation des nœuds capteurs, l'étape suivante consiste à réattribuer les canaux aux nœuds déplacés. Ce processus a pour objectif l'attribution d'un canal à chaque nœud en minimisant les conflits avec ses voisins tout en minimisant le nombre total de changements de canaux. Comme CR-MC est une approche distribuée, chaque nœud négocie son nouveau canal avec ses voisins à un et à deux sauts. L'algorithme de réattribution des canaux fonctionne alors comme suit :

- Chaque nœud déplacé v diffuse un message *hello* ;
- Chaque voisin u de v répond par un message contenant son ID, son canal et le nombre de ses liens conflictuels ;

- Si un canal est toujours disponible, v le sélectionne ;
- Sinon, v calcule le nombre de ses liens conflictuels pour chaque canal et diffuse un message *conflicts* ;
- Chaque voisin u vérifie s'il peut réduire le nombre de liens conflictuels en changeant son canal. Si oui, il envoie un message *suggestion* à v avec son ID et le numéro de son canal ;
- A la réception des messages *suggestion*, v choisit son canal de sorte à minimiser le nombre de liens conflictuels ;
- v diffuse un message *decision* pour informer ses voisins du canal choisi ;
- Si un voisin u a envoyé une proposition et v a choisi le même canal que u , u change son canal et informe ses voisins.

La procédure de réallocation des canaux est décrite dans l'algorithme 4.

Algorithm 4 Algorithme de recoloration des nœuds

```

ch=0
envoyer(msg_hello,id);
recevoir(msg_hello, id, c, nb_cf(c));
if  $\exists i \in M | C_i = \emptyset$  then
    ch=i;
else
    Pour tout  $i \in M$ 
        calculer (nb_cf(i)); /* calculer le nombre de conflits pour chaque canal  $i^*$ /
        envoyer (msg_conflicts);
        recevoir(msg_suggestion, id, c,);
        ch=best_suggestion(suggestion_list);
        if suggestion_list= $\emptyset$  then
            ch= $\text{argmin}_{i \in M}$  (nb_cf(i));
        end if
    end if
    envoyer(msg_Decision);

```

3.3.5 Synthèse

Bien que la solution CR-MC assure la restauration de la connectivité et la réattribution des canaux d'une manière localisée (autour de l'AP défaillant), elle présente certains inconvénients. En effet, CR-MC ne tient pas compte des routes suivies par les données acheminées vers le nœud puits. Avec CR-MC des canaux différents peuvent être attribués à deux fils ayant le même parent dans l'arbre de routage, alors qu'un même canal peut être assigné à deux voisins ayant deux parents différents dans l'arbre de routage. Ainsi,

nous proposons, dans la section suivante, une nouvelle approche CR-RMC qui utilise les informations de routage pour l'allocation des canaux.

3.4 Connectivity Restoration for Routing based Multi-Channel WSNs : CR-RMC

Dans cette section, nous proposons l'approche CR-RMC (*Connectivity Restoration for Routing based Multi-Channel WSNs*) qui prend en compte les informations de routage dans son fonctionnement. L'utilisation des arbres de routage permet une meilleure affectation des canaux disponibles par l'attribution du même canal à tous les fils d'un même parent dans l'arbre de routage. Cette stratégie d'allocation va minimiser le nombre de commutations des canaux le long des routes vers le nœud puits. Afin d'optimiser l'utilisation de la ressource énergétique des nœuds, l'approche CR-RMC utilise une stratégie de mise en veille des nœuds capteurs qui dépend du nombre de sauts vers le nœud puits.

3.4.1 Modèle du réseau

Dans l'approche CR-RMC, l'allocation des canaux se fait en exploitant l'information de routage. L'idée est de dire si deux nœuds ont un même parent au niveau de l'arbre de routage, mieux vaut leur attribuer un même canal (le canal de réception du nœud père). Ces nœuds utilisent une technique d'accès déterministe de type TDMA pour transmettre leurs données à leurs ascendants dans l'arbre de routage. Cette idée repose sur le fait que ces deux nœuds ne peuvent pas transmettre à leur parent simultanément. Si, par contre, deux nœuds voisins ont deux parents différents, il vaut mieux leur attribuer deux canaux différents de sorte à maximiser le nombre de transmissions simultanées à un niveau donné de l'arbre de routage. Par conséquent, nous utilisons pour CR-RMC une nouvelle modélisation du réseau que nous décrivons dans ce qui suit.

Pour modéliser le réseau, nous introduisons, en plus du lien d'interférence, un lien de parenté $e_p(u, v)$ comme étant le lien entre le nœud u et son parent v dans l'arbre de routage. Nous rappelons qu'un lien d'interférence $e(u, w)$ entre deux nœuds u et w existe si la distance $d(u, w) \leq 2R_c$ (R_c étant le rayon de communication). Étant donné qu'un nœud capteur ne peut recevoir que d'un seul fils à la fois, deux nœuds u et w , ayant le même parent i , ne peuvent pas transmettre les données simultanément à leur parent i même s'ils utilisent des canaux différents. Il en résulte que le lien $e(u, w)$ entre ces deux nœuds ne peut être considéré comme un lien conflictuel. Compte tenu de ce qui précède, nous

introduisons le concept d'un nœud logique LN (Logical Node) comme étant l'ensemble des fils d'un même parent $i : N_i = \{u | p(u) = i\}$, où $p(u)$ désigne le parent du nœud u . Deux nœuds logiques N_i et N_j sont interférents (un lien d'interférence existe entre eux) si et seulement si l'une des conditions suivantes est vérifiée :

- Des fils de i et de j sont interférents (\exists un lien d'interférence $e(u, v) | u \in N_i$ et $v \in N_j$)
- Un des parents interfère avec des fils de l'autre parent (\exists un lien d'interférence $e(v, i)$ (respectivement $e(u, j)$) $| v \in N_j$ (respectivement $u \in N_i$)).

Nous introduisons un coût pour le lien d'interférence d'un nœud logique N_i vers un autre nœud N_j . Pour déterminer ce coût, nous définissons tout d'abord la fonction α qui vérifie si un lien d'interférence existe entre deux nœuds capteurs. Cette fonction est donnée par :

$$\alpha(u, v) = \begin{cases} 1 & \text{si } e(u, v) \in E_0 \\ 0 & \text{sinon.} \end{cases} \quad (3.14)$$

Si deux nœuds u et j interfèrent, u ne peut pas transmettre simultanément avec tous les fils de j sur le même canal et vice versa. Ainsi, nous définissons le coût d'un lien d'interférence d'un nœud capteur u vers le nœud logique N_j comme étant :

$$c(u, N_j) = \alpha(u, j) |N_j| \quad (3.15)$$

Le coût total d'un lien d'interférence d'un nœud logique N_i vers un autre nœud N_j inclut les coûts de tous les liens entre les capteurs de N_i et le nœud j (parent de tous les capteurs de N_j). En plus, nous ajoutons les liens d'interférence entre les capteurs de N_i , non interférents avec le nœud j , et les capteurs de N_j . Ce coût est alors déterminé par :

$$c(N_i, N_j) = \sum_{u \in N_i} c(u, N_j) + \sum_{w \in N_i \wedge w \neq u, v \in N_j} \alpha(w, v) \quad (3.16)$$

La figure 3.3 montre un exemple de deux nœuds logiques interférents N_i et N_j . Un lien d'interférence existe entre le nœud capteur u ($\in N_i$) et le nœud capteur j (parent de N_j). Ainsi, u ne peut pas transmettre simultanément avec les trois fils de j sur le même canal. De plus, le capteur w ($\in N_i$) interfère avec le capteur v ($\in N_j$). Ainsi, le coût de lien d'interférence de N_i vers N_j est égal alors à 4 (3+1).

Nous pouvons ainsi, extraire le graphe logique $G = (V, E)$ à partir du graphe initial G_0 (défini dans la section 3.3.1), où V est l'ensemble des LNs et E l'ensemble des liens d'interférence entre les LNs. Nous définissons $G' = (V', E')$ le graphe logique représentant

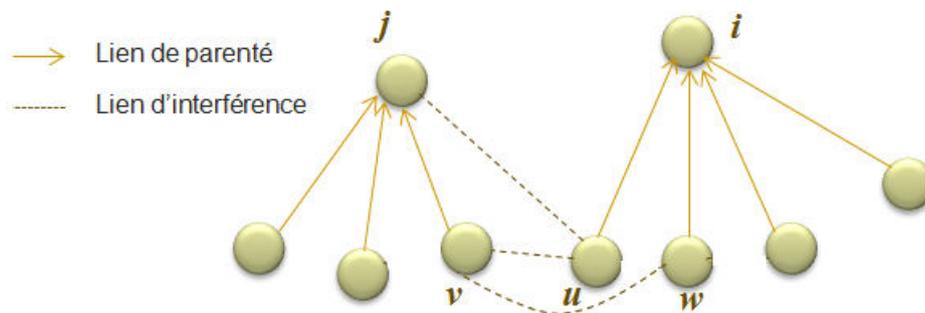


Figure 3.3 : Exemple des deux nœuds logiques interférents

le RCSF après le recouvrement de la panne. Le passage du graphe G_0 au graphe G est illustré par la figure 3.4.

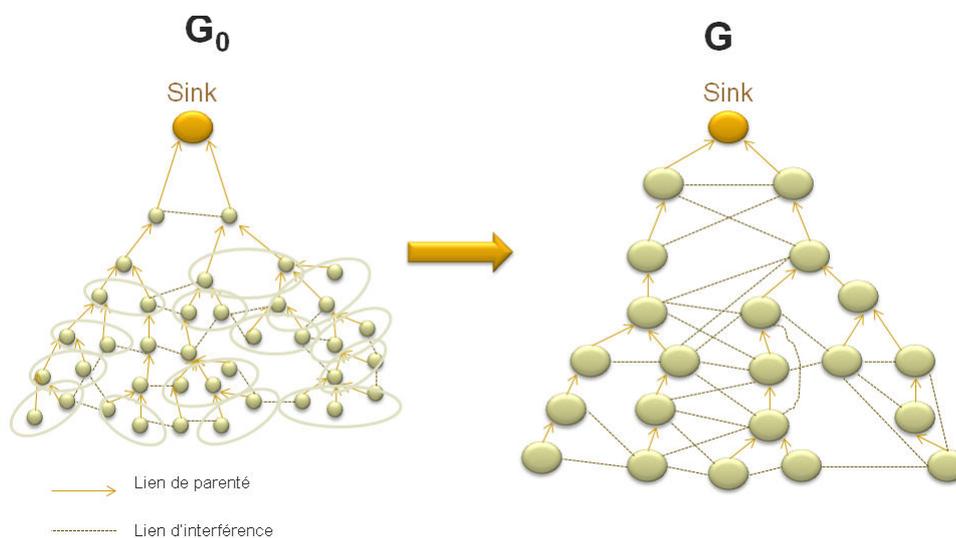


Figure 3.4 : Graphe des nœuds logiques

En exploitant le nouveau modèle du réseau, nous présentons dans la section suivante une formulation du problème de recouvrement de pannes dans un contexte multi-canal basé sur le routage.

3.4.2 Formulation du problème

Contrairement à CR-MC, CR-RMC a pour objectif de minimiser le nombre de liens conflictuels au niveau de chaque LN au lieu de minimiser le nombre de liens conflictuels au niveau de chaque nœud capteur. Ainsi, notre problème d'optimisation multi-objectif peut être énoncé comme suit :

$$\text{minimiser } f \tag{3.17}$$

sous les contraintes :

$$G' \text{ connecté} \tag{3.18}$$

et tous les nœuds soient colorés ; ce qui se traduit par :

$$c_{G'}(v) \in M, \forall v \in V' \tag{3.19}$$

avec

$$f = [f_1, f_2, f_3, f_4, f_5, f_6]^T \tag{3.20}$$

où f_2 , f_3 et f_4 sont données respectivement par les équations (3.5), (3.7) et (3.9) ; et où f_1 , f_5 et f_6 seront explicitées dans ce qui suit.

La fonction f_1 , quant à elle est obtenue en redéfinissant la fonction δ donnée par (3.1) :

$$\delta(N_i, N_j) = \begin{cases} c(N_i, N_j) & \text{si } e(N_i, N_j) \in E' \wedge c_{G'}(i) = c_G(j) \\ 0 & \text{sinon.} \end{cases} \tag{3.21}$$

L'équation (3.2) devient alors :

$$f_1^{N_i} = \sum_{\forall N_j \in V'_0} \delta(N_i, N_j). \tag{3.22}$$

Enfin, la fonction objectif f_1 donnée par (3.3) est redéfinie comme suit :

$$f_1 = [f_1^{N_1}, \dots, f_1^{N_{n_l}}], \text{ où } n_l = |V| \tag{3.23}$$

La fonction f_5 s'intéresse aux nœuds capteurs déplacés après la restauration de la connectivité. En effet, le déplacement d'un nœud dans le réseau peut avoir un impact direct sur le nombre de sauts (appelé niveau) qui le séparent du nœud puits. Ainsi, si un nœud parent v est déplacé pour des besoins de recouvrement, ses fils et leurs descendants devront peut-être changer de niveau dans l'arbre de routage et une réattribution des canaux devra alors être faite pour tous les sous-arbres de v . Nous définissons alors la fonction $h_G : V \mapsto L$ donnant le niveau d'un nœud capteur dans un graphe G , où L représente l'ensemble des niveaux dans l'arbre de routage du réseau représenté par ce graphe. Nous introduisons la fonction $\lambda(v)$ qui vérifie si un nœud appartient au même

niveau avant et après la réorganisation du réseau :

$$\lambda(v) = \begin{cases} 1 & \text{si } h_G(v) \neq h_{G'}(v) \\ 0 & \text{sinon.} \end{cases} \quad (3.24)$$

Ainsi la fonction f_5 peut être définie comme étant la fonction qui calcule le nombre total de nœuds capteurs qui changent leurs niveaux dans le réseau :

$$f_5 = \sum_{v \in V'_0} \lambda(v) \quad (3.25)$$

La fonction f_6 donnée dans (3.20) détermine le nombre de nœuds qui ont besoin de commuter leurs radios pour transmettre les données à leurs parents. Ainsi, nous définissons la fonction ω par :

$$\omega(v) = \begin{cases} 1 & \text{si } c_{G'}(v) = c_G(p(v)) \\ 0 & \text{sinon.} \end{cases} \quad (3.26)$$

Le nombre total de nœuds qui n'ont pas le même canal que leurs parents est donné par :

$$f_6 = \sum_{v \in V'_0} \omega(v) \quad (3.27)$$

Dans ce qui suit, nous détaillons les différentes étapes de l'approche CR-RMC proposée pour la résolution de ce problème.

3.4.3 Stratégie de mise en veille

L'approche CR-RMC que nous décrivons ici, se base sur l'information de routage pour l'allocation des canaux. Ainsi, le mécanisme d'allocation doit tenir compte des données réelles sur les différentes routes de l'arbre à un instant donné. En effet, la transmission simultanée des nœuds d'un niveau i de l'arbre de routage avec leurs descendants du niveau $i+1$ ou leurs parents du niveau $i-1$ ne peut être efficace quelle que soit la configuration optimale choisie pour l'allocation des canaux. Ceci est dû aux collisions et aux interférences causées par les transmissions des nœuds de niveaux adjacents. Il en résulte une dégradation importante des performances du réseau en terme de débit utile et ceci quelle que soit la distribution des canaux pour laquelle nous avons opté.

Ainsi, nous proposons dans notre approche CR-RMC d'utiliser une transmission sé-

quentielle des nœuds de niveau i vers leurs parents de niveau $i-1$. Cette transmission se fera de proche en proche jusqu'au nœud puits. Cela garantira qu'à un instant donné seuls les nœuds de niveau i enverront leurs données à leurs parents de niveau $i-1$. Tous les nœuds des autres niveaux attendront la fin d'une telle transmission.

Cette transmission permettra aussi de faire une allocation optimale des canaux aux nœuds de niveau i . En effet, tous les canaux disponibles peuvent être alloués à l'instant t_i (instant de transmission des nœuds de niveau i) au n_i nœuds logiques de niveau i . Afin d'optimiser cette transmission et cette allocation séquentielle des canaux, un algorithme de synchronisation entre les différents niveaux est nécessaire pour déterminer les instants auxquels les capteurs doivent commuter leurs radios. Par ailleurs, nous proposons d'associer à cet algorithme de synchronisation une stratégie de mise en veille des nœuds capteurs de niveau i une fois que ces derniers ont terminé leur transmission. Cette stratégie de mise en veille assurera une meilleure utilisation de la ressource énergétique au niveau des nœuds capteurs si ces derniers n'ont rien à transmettre.

La figure 3.5 illustre la stratégie de mise en veille que nous adoptons. En effet, nous divisons le temps en trames. Une trame est choisie de sorte que tous les nœuds logiques des L niveaux de l'arbre de routage (niveau 0 étant le puits) transmettent leurs données aux niveaux supérieurs d'une manière séquentielle. Ainsi, la trame est divisée en slots de temps pour la diffusion (*Broad*), la réception (*Receive*), le traitement (*Pr*) et l'envoi des données (*Send*).

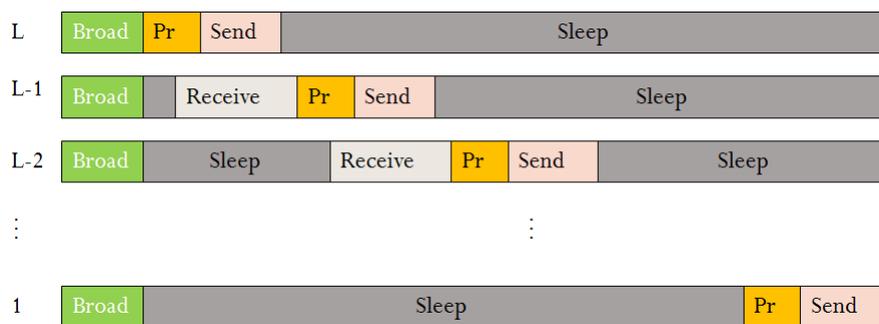


Figure 3.5 : Stratégie de mise en veille dans CR-RMC

Un premier slot *Broad*, pendant lequel tous les nœuds capteurs sont actifs, est réservé pour la diffusion des messages de contrôle. Ensuite, tous les capteurs se mettent en veille sauf ceux du dernier niveau L qui restent actifs pour traiter les données (slot *Pr*). Avant leur slot d'envoi (*Send*), les nœuds capteurs du niveau supérieur $L-1$ se réveillent pour recevoir (slot *Receive*) les données du niveau L , les traiter (slot *Pr*) et les envoyer (slot *Send*) à leurs parents du niveau $L-2$. Ce processus se répète pour chaque niveau l de

l'arbre de routage. Par conséquent et selon cette stratégie, deux nœuds appartenant à deux niveaux différents ne transmettent jamais leurs données simultanément.

Nous optons pour cette stratégie car nous nous intéressons au cas où la durée de la période de collecte de données est plus importante que le temps nécessaire pour que les données atteignent le nœud puits. Dans le cas contraire, nous pouvons avoir recours au parallélisme pour réduire le délai de transmission, c'est à dire que les nœuds de deux niveaux non interférents (éloignés au moins de deux sauts) peuvent transmettre simultanément sur les mêmes canaux.

3.4.4 Allocation initiale des canaux

En se basant sur la stratégie de synchronisation et de mise en veille présentée dans la section 3.4.3, nous proposons une technique d'allocation initiale des canaux qui minimise les interférences entre les nœuds logiques d'un niveau l de l'arbre de routage ($1 \leq l \leq L$). Une heuristique de coloration de graphes est utilisée pour minimiser le nombre de liens conflictuels pour les nœuds logiques d'un même niveau.

Ainsi, une allocation initiale des canaux se fera pour chaque niveau de l'arbre de routage d'une manière indépendante. Un niveau l de l'arbre peut alors être modélisé par le graphe réduit $G_l = (V_l, E_l)$. Les m couleurs, représentant les m canaux disponibles, peuvent alors être allouées à tous les sommets du graphe G_l .

La procédure de coloration du graphe G_l est décrite par l'algorithme 5. Nous commençons d'abord par colorer le sommet (le nœud logique) N_i ayant le degré $c(N_i)$ le plus élevé (avec $c(N_i) = \sum_{\forall N_j} c(N_i, N_j)$ représentant le nombre de liens d'interférence de N_i avec ses voisins). Ensuite, tous les sommets N_j qui ne sont pas voisins que N_i sont colorés par la même couleur de N_i du fait qu'ils ne génèrent aucun conflit avec N_i . Puis, l'algorithme essaie d'assigner une couleur différente à tous les sommets ayant un lien d'interférence avec les sommets déjà colorés. Si à un moment donné, aucune couleur n'est disponible, les sommets restants sont colorés un par un par la couleur générant le plus petit nombre de liens conflictuels. En passant du niveau l vers le niveau $l-1$, le nœud N_i sera coloré par la couleur la plus utilisée par ses fils du niveau l .

Les figures 3.6a, 3.6b, 3.6c, et 3.6d illustrent les étapes de coloration des nœuds logiques des différents niveaux de l'arbre de routage. Nous remarquons sur la figure 3.6d que le mécanisme de coloration essaie d'assigner la même couleur aux nœuds logiques appartenant à la même route vers le nœud puits. Ceci aura pour conséquence de minimiser les interférences entre les routes vers le puits.

Algorithm 5 Allocation initiale basée sur le routage

```

for  $\forall i \in V_l$  do
     $Lt = Tri(N_i, c(N_i))$ 
end for
 $C = \emptyset$ ; /*  $C$  ensemble des sommets déjà colorés */
for  $\forall j \in M$  do
     $C_j = \emptyset$ ; /*  $C_j$  ensemble des sommets colorés par la couleur  $j$  */
end for
 $k=1$ ; /* couleur */
 $l=L$ ; /* niveau */
while  $l \geq 2$  do
     $C_k = \{N_i | c(N_i) = \max_j(c(N_j)) \wedge |N_i| = \max_j(|N_j|), \forall N_j \in Lt\}$ ;
    Supprimer  $N_i$  de  $Lt$ ;
     $C_k = C_k \cup \{N_j | c(N_j, N_i) = 0, \forall N_j \in Lt \wedge \forall N_i \in C\}$ ;
    Supprimer  $N_j$  de  $Lt$ ;
    repeat
         $k = k + 1$ ;
         $C_k = \{N_j = \operatorname{argmax}_{N_i \in C}(c(N_j, N_i)) | (N_j, N_i) \in E_l\}$ ;
        Supprimer  $N_j$  de  $Lt$ ;
         $C = C \cup C_k$ 
    until  $Lt = \emptyset$  or  $k = L + 1$ 
    if  $Lt \neq \emptyset$  then
        repeat
             $m = \{ch(N_i) | N_i = \operatorname{argmin}_{N_j \in C}(\Delta(N_j))\}$ ;
             $C_m = C_m \cup \{N_j\}$ ;
        until  $Lt = \emptyset$ 
        end if
         $l = l - 1$ ;
    end while

```

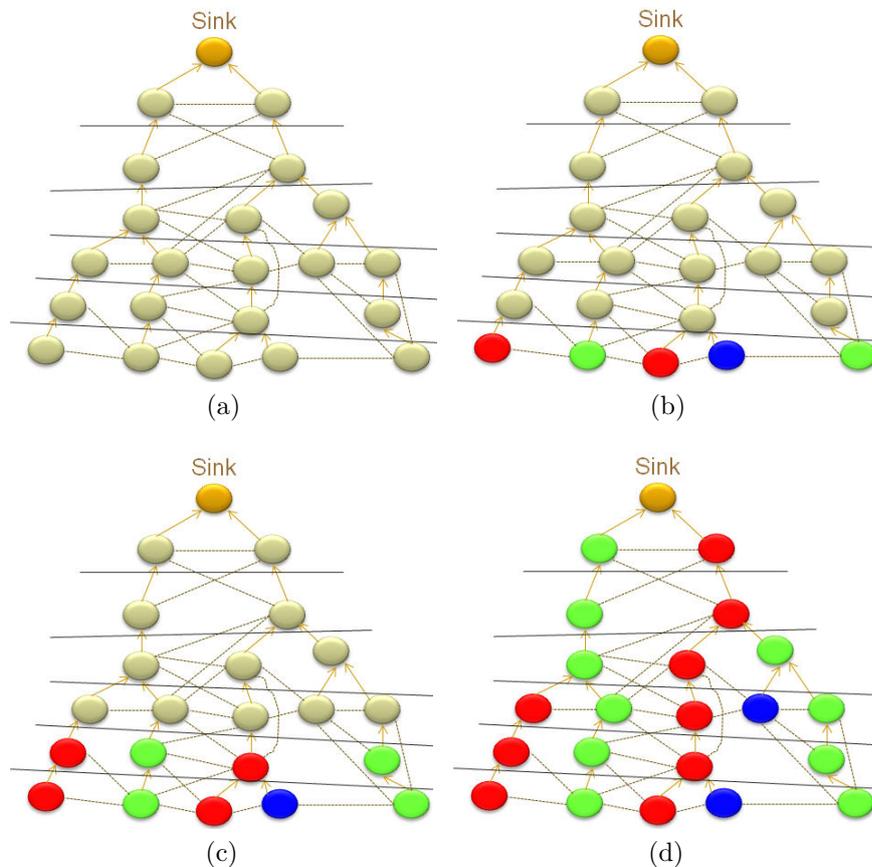


Figure 3.6 : Coloration du graphe logique dans CR-RMC

Une fois l'allocation initiale terminée, le RCSF continue son fonctionnement normal jusqu'à l'occurrence d'une panne. L'approche CR-RMC déclenche alors les mêmes mécanismes de détection et de recouvrement de pannes, basés sur les points de Steiner, décrits dans l'approche CR-MC. Seule la réattribution des canaux sera différente pour CR-RMC.

3.4.5 Réallocation des canaux

Après le déplacement des nœuds remplaçants vers leurs nouveaux emplacements, ces nœuds ainsi que leurs fils essaieront de trouver de nouveaux parents. Une réallocation des canaux peut s'avérer nécessaire. Notre objectif est de réduire l'impact du recouvrement de la panne sur la totalité du réseau en limitant les changements par rapport à la configuration initiale du réseau. Ces changements incluent le choix des parents (nœuds logiques), des niveaux et des canaux utilisés. En effet, un nœud capteur v du réseau peut être amené à changer de couleur (changer de canal) si :

- Il est directement impliqué dans le recouvrement de la panne (élu pour être déplacé

vers un SP).

- ou bien, son parent dans l'arbre de routage est directement impliqué dans le recouvrement de la panne.

Ainsi, il devra retrouver un nouveau parent dans l'arbre de routage de sorte à introduire le moins de changements possibles au niveau de l'arbre déjà construit. La recherche d'un nouveau parent s'exécute comme suit :

- Le nœud v met à jour sa table de voisinage par l'échange de messages *hello*.
- Le nœud v déclenche un timer T_0 et diffuse un message *parent_request* avec son ID, son niveau et ses voisins.
- Lorsqu'un nœud u , ayant déjà un parent, reçoit le message *parent_request*, il calcule le nombre de conflits générés par la demande de v , u envoie un message *parent_candidate* à v avec son ID, son niveau, son canal et le nombre de conflits générés.
- Si u lui-même n'a pas de parent, il sauvegarde le message et attend la réponse à son propre message *parent_request*.
- Si v reçoit un message *parent_candidate*, alors il vérifie le niveau dans l'arbre de routage et le nombre de conflits reçus :
 - Si le niveau est le même que son ancien parent et le nombre de conflits est égal à zéro, v annule immédiatement T_0 .
 - Sinon, v attend l'écoulement de T_0 et sélectionne le meilleur candidat pour être son parent. La sélection se base essentiellement sur le niveau du candidat puis sur le nombre de conflits.
- Le nœud v retourne par un message *parent_join* au candidat sélectionné.
- Le nœud v doit aussi répondre à tous les messages *parent_request* déjà sauvegardés.
- Si le niveau est changé, v doit informer les nœuds de ses propres sous-arbres pour qu'ils mettent à jour leurs niveaux et leurs slots d'activité.

Si le nombre de niveaux dans l'arbre de routage a augmenté, il faudrait mettre à jour la taille de la trame à tous les niveaux de l'arbre.

3.5 Étude de la complexité des approches CR-MC et CR-RMC

Dans cette partie, nous analysons les approches CR-MC et CR-RMC pour comparer leurs performances en termes de complexité algorithmique et de nombre de messages de contrôle. Pour ce faire, nous nous concentrons sur le nombre de messages de contrôle à échanger (NM) et le nombre d'opérations à exécuter (NO). Le fait que les approches

CR-MC et CR-RMC soient distribuées implique que les mécanismes de recouvrement de panne sont déployés autour de la zone affectée. Ainsi, la complexité des solutions CR-MC et CR-RMC est insensible à la taille du réseau (en terme de nombre de nœuds) mais forcément affectée par sa topologie et sa densité.

Pour commencer, nous allons calculer le nombre de messages émis pendant la phase de la réorganisation du réseau. Chaque voisin à un saut déclenche les algorithmes de réorganisation indépendamment des autres nœuds. Les nœuds n'ont pas besoin d'échanger de messages à cette étape. Cependant, les voisins à deux sauts doivent être informés de la décision prise par les nœuds à un saut du nœud défaillant. Si nous supposons que $H_1(AP_f)$ et $H_2(AP_f)$ sont les ensembles des voisins à un et à deux sauts du nœud défaillant et ND le nombre de nœuds à déplacer, alors le nombre de messages émis pendant la phase de réorganisation est :

- $NM(restauration) = |H_1(AP_f)|$ si $ND \leq |H_1(AP_f)|$.
- $NM(restauration) = ND$ si $ND > |H_1(AP_f)|$.

De plus, il a été démontré dans [76] que le nombre de points de Steiner (ND) ne peut pas dépasser $|H_2(AP_f)| - 2$. Comme chaque nœud déplacé informe ses voisins de son déplacement, dans les deux cas, le nombre de messages est borné ($NM(restauration) \leq \max(|H_1(AP_f)|, |H_2(AP_f)| - 2)$). Si nous désignons par Δ le degré de voisinage maximal d'un nœud alors le nombre de messages émis est borné par $\Delta^2 - 2$. Ceci est vrai dans tous les cas où $\Delta > 1$ (le degré maximal de voisinage d'un nœud est supérieur à 1). Par conséquent, le nombre de messages échangés pendant la phase de réorganisation du RCSF est de l'ordre de $O(\Delta^2)$.

La deuxième étape du recouvrement de la panne est consacrée à la réallocation des canaux. Nous commençons par l'approche CR-MC où chaque nœud déplacé i envoie 3 messages (*hello*, *conflicts* et *decision*) et reçoit au pire des cas $2\Delta_i + 1$ messages (*hello* et *suggestion*), avec Δ_i le degré (nombre de voisins) du nœud i . Il en résulte un nombre total de messages émis par tous les nœuds déplacés (V_d) donné par :

$$NM(réallocation) = \sum_{\forall i \in V_d} (2\Delta_i + 4) \quad (3.28)$$

Si nous suivons le même principe que la première partie, nous aurons :

$$NM(réallocation) \leq ND(2\Delta_i + 4) \quad (3.29)$$

Puisque $ND < \Delta^2 - 2$ et $\Delta_i \leq \Delta$, l'inéquation (3.29) peut s'écrire :

$$NM(\text{réallocation}) < (\Delta^2 - 2)(2\Delta + 4) \quad (3.30)$$

D'où :

$$NM(\text{réallocation}) < 2(\Delta^3 + 2\Delta^2 - 4\Delta - 4) \quad (3.31)$$

Par conséquent, le nombre de messages échangés durant la phase de réattribution des canaux est de l'ordre de $O(\Delta^3)$ pour CR-MC. Le nombre total de messages émis ($NM = NM(\text{restauration}) + NM(\text{réallocation})$) est de l'ordre de $O(\Delta^3)$. De la même façon, nous pouvons montrer que CR-RMC émet aussi un nombre de messages de l'ordre $O(\Delta^3)$.

Passons maintenant au calcul du nombre d'opérations effectuées NO par chaque nœud pendant la procédure de la réattribution des canaux. Tout d'abord, pour CR-MC, un nœud i reçoit Δ_i messages *hello* et doit vérifier parmi ces messages s'il reste encore un canal disponible. Au pire des cas, le nœud i doit chercher la meilleure proposition en faisant au plus Δ_i comparaisons. Suivant la procédure de réallocation, le nœud i peut aussi recevoir au plus Δ_i propositions de ses voisins pour choisir son canal. Ainsi, i effectue encore Δ_i opérations de comparaison. Il en résulte un nombre total d'opérations inférieur à $2\Delta_i$. Ceci implique que $NO \leq 2\Delta_i$. Si Δ représente le degré maximal de voisinage d'un nœud, alors la complexité de l'algorithme au niveau de chaque nœud est $O(\Delta)$.

De la même manière, si nous considérons l'approche CR-RMC, chaque nœud j reçoit au plus Δ_j messages *parent-candidate*. Le nœud j doit faire deux comparaisons pour chaque message reçu pour vérifier le niveau dans le réseau et le nombre de liens conflictuels avant de choisir son parent. Au total, le nœud j effectue $2\Delta_j$ opérations de comparaison, ce qui donne une complexité algorithmique de $O(\Delta)$ pour CR-RMC. Nous pouvons conclure donc que CR-MC et CR-RMC ont la même complexité algorithmique.

3.6 Évaluation des performances des approches CR-MC et CR-RMC

Nous évaluons dans cette section, les approches distribuées CR-MC et CR-RMC en utilisant les mêmes paramètres de simulation utilisés dans le chapitre 2 (section 2.7) et en les comparant avec l'approche centralisée RNFR. Les métriques de performance utilisées dans le présent chapitre sont :

- Le nombre de liens conflictuels.
- Le débit utile.
- Le nombre de commutations de canaux pendant la transmission. Ce critère est très important parce qu'il a un impact direct sur le délai de transmission de bout en bout et sur la consommation énergétique.
- Le nombre de nœuds déplacés.
- Le nombre de changements de canaux (nombre de nœuds recolorés).
- La stabilité des routes : Cette métrique définit les changements subis par les chemins de routage après le recouvrement de la panne. Elle reflète les changements au niveau des canaux utilisés par les capteurs et leurs niveaux dans l'arbre de routage (nombre de sauts vers le puits).
- La distance totale parcourue.

La figure 3.7 évalue le nombre de liens conflictuels en fonction du nombre de nœuds dans le réseau. Cette figure illustre l'apport de l'approche CR-RMC par rapport aux approches RNFR et CR-MC. En effet, l'introduction de la notion de nœud logique a permis une meilleure exploitation des canaux disponibles en attribuant le même canal à tous les fils du même parent. Par contre, CR-MC génère un nombre de conflits plus élevé puisque chaque nœud n'a qu'une vue locale de son voisinage par rapport à RNFR dans lequel le nœud puits possède une vue globale du réseau.

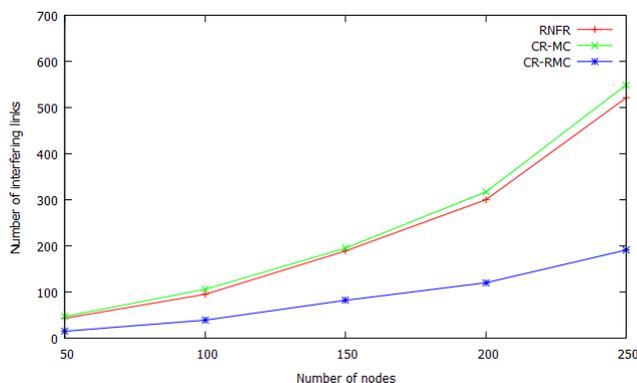


Figure 3.7 : Le nombre de liens conflictuels en fonction du nombre de nœuds

La figure 3.8 permet de comparer le débit utile atteint par les trois approches en fonction du nombre de nœuds capteurs. Cette figure montre que l'approche CR-RMC offre un meilleur débit que les approches RNFR et CR-MC. Ceci s'explique par le fait que le nombre de liens conflictuels générés par cette approche est toujours inférieur à celui généré par RNFR et CR-MC. En effet, le débit atteint par CR-RMC reste toujours important (>92%) quel que soit le nombre de nœuds capteurs présents dans le réseau. Ce débit, par contre, décroît rapidement pour les approches CR-MC et RNFR avec l'augmentation du

nombre de nœuds (avec un réseau à 250 nœuds, nous avons un débit de 78% pour RNFR et 76% pour CR-MC).

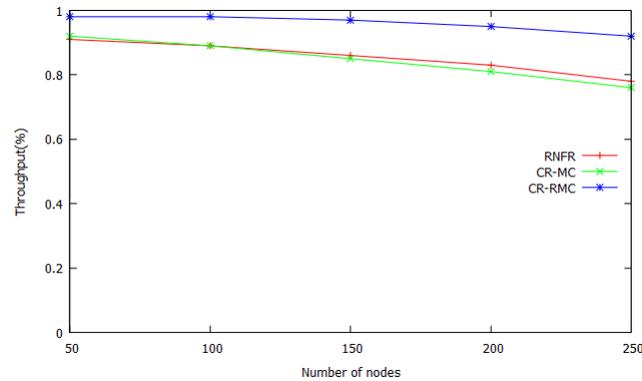


Figure 3.8 : Le débit utile en fonction du nombre de nœuds

La figure 3.9 présente le nombre de commutations de canaux dans tout le réseau pendant la transmission en fonction du nombre de nœuds présents dans le réseau. En général, le capteur a besoin de $200\mu s$ pour commuter sa radio à un autre canal. Pour les réseaux à grande échelle, la commutation de canaux le long d'une route peut avoir un impact sérieux sur le délai de transmission total. CR-RMC génère seulement 150 commutations dans un RCSF à 250 nœuds comparées à 593 et 653 commutations respectivement pour CR-MC et RNFR. Si nous considérons ces valeurs en terme de temps total de commutations, nous obtenons respectivement $30 ms$, $118.6 ms$ et $130.6 ms$ pour les approches CR-RMC, CR-MC et RNFR. Ceci montre l'importance de la prise en compte de l'information de routage dans l'allocation des canaux.

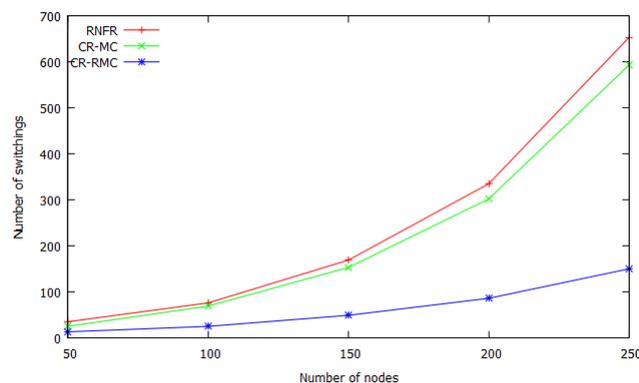


Figure 3.9 : Nombre de commutations en fonction du nombre de nœuds

La figure 3.10 dépeint le nombre de nœuds déplacés en fonction du nombre total de nœuds capteurs. Il est clair que les deux approches distribuées déplacent moins de nœuds que l'approche RNFR. Les nœuds supplémentaires déplacés dans RNFR sont dus à la

rotation en cascade utilisée pour reconnecter les différentes parties du réseau segmenté. Cependant, quand la densité du réseau devient importante, les trois approches déplacent le même nombre de nœuds. Ceci s’explique par le fait qu’à forte densité (cas des réseaux à 200 et à 250 nœuds dans la figure 3.10) le réseau sera divisé en un petit nombre de segments. Ce qui réduira le nombre de nœuds en rotation pour l’approche RNFR.

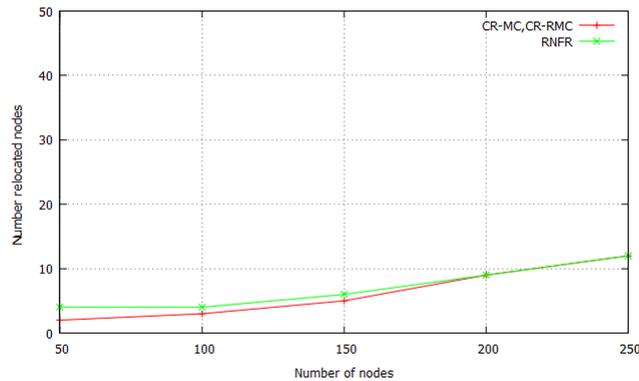


Figure 3.10 : Le nombre de nœuds déplacés en fonction du nombre de nœuds

La figure 3.11 montre le nombre de nœuds réassignés à d’autres canaux après le recouvrement de la panne en fonction du nombre de capteurs. Nous remarquons que, de nouveau, CR-RMC a montré son efficacité par rapport aux autres approches en recolorant moins de nœuds. Cette efficacité est d’autant plus remarquable pour les réseaux denses. Cela se traduit par la recoloration de 45 nœuds seulement pour CR-RMC, dans un réseau à 250 nœuds, comparés à 75 nœuds recolorés pour CR-MC et 89 nœuds pour RNFR pour le même nombre de nœuds.

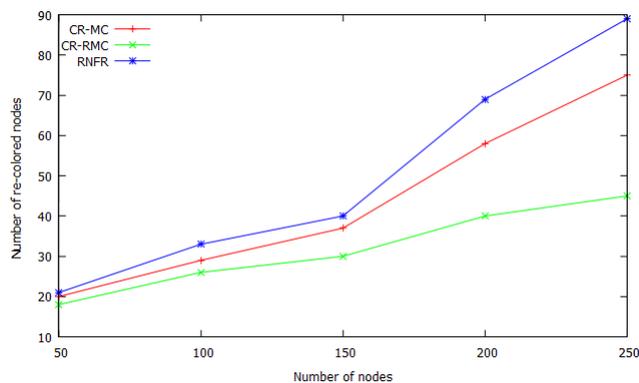


Figure 3.11 : Le nombre de nœuds recolorés en fonction du nombre de nœuds

La figure 3.12 montre la stabilité des routes après le recouvrement de la panne en fonction du nombre total de nœuds présents dans le réseau. Cette figure illustre bien que

l'approche CR-RMC offre un degré de stabilité très élevé par rapport aux approches CR-MC et RNFR en particulier dans les réseaux denses. Ceci s'explique par le fait que cette approche favorise les parents ayant le même nombre de sauts vers le puits et le même canal que l'ancien parent lors de la sélection d'un nouveau parent d'un nœud capteur. Par contre, CR-MC et RNFR ne considèrent que le nombre de conflits dans le choix d'un nouveau canal ; alors que les routes sont mises à jour par le protocole de routage utilisé. En plus, les routes deviennent plus stables lorsque la densité du réseau augmente. Cette stabilité est due à l'augmentation de la probabilité de trouver un parent avec le même niveau et le même canal que l'ancien parent. Par exemple, pour CR-RMC les routes sont stables à 85% dans un réseau à 50 nœuds (comparé à 62% pour CR-MC et RNFR). Ce pourcentage passe à 98% dans un réseau à 250 capteurs (80% pour CR-MC et 81% pour RNFR).

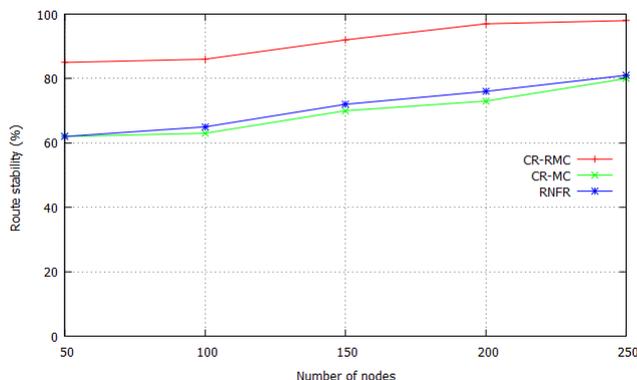


Figure 3.12 : La stabilité des routes en fonction du nombre de nœuds

Pour terminer, nous examinons la distance totale parcourue en fonction du nombre de nœuds dans le réseau (figure 3.13). Nous observons que les deux approches CR-MC et CR-RMC génèrent la même distance puisqu'elles utilisent le même mécanisme pour le choix des nœuds à déplacer et leurs futurs emplacements. Cependant, ces nœuds parcourent une distance plus grande avec RNFR. Cette distance supplémentaire est due à la rotation des voisins à un saut du nœud d'articulation défaillant pour informer les régions segmentées du recouvrement de la panne.

En se basant sur les résultats précédents, nous pouvons conclure que l'approche CR-RMC présente une meilleure efficacité que les approches CR-MC et RNFR lors du recouvrement de la panne d'un nœud d'articulation dans le RCSF. Cette efficacité se résume en termes de nombre de liens interférents, débit utile, nombre de nœuds déplacés, nombre de nœuds recolorés, nombre de commutations et distance totale parcourue.

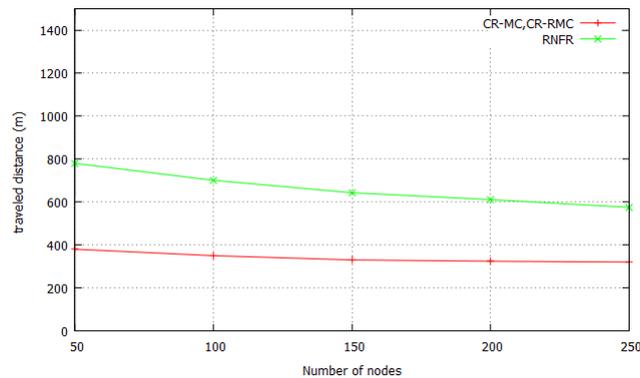


Figure 3.13 : La distance parcourue en fonction du nombre de nœuds

3.7 Conclusion

Dans ce chapitre, nous avons proposé deux approches distribuées CR-MC et CR-RMC pour le rétablissement de la connectivité dans les RCSFs multi-canaux après la panne d'un nœud critique. CR-MC est une approche native qui essaie de restaurer la connectivité du réseau en utilisant la technique de points de Steiner avant de réattribuer les canaux par une simple négociation entre les voisins à un saut du nœud défaillant. Dans l'approche CR-RMC, par contre, nous avons essayé de minimiser le nombre de liens conflictuels en exploitant les informations de routage. Nous avons par ailleurs proposé une stratégie de mise en veille pour minimiser la consommation énergétique au niveau des capteurs qui n'ont pas des données à envoyer.

L'évaluation des performances des deux approches, a montré que l'approche CR-RMC est plus efficace que CR-MC. De plus, une comparaison avec l'approche centralisée RNFR décrite dans le chapitre 2 a été effectuée et a montré que les performances de CR-RMC outrepassent celles de l'approche centralisée RNFR.

Pour toutes les solutions que nous avons décrites jusque là dans les chapitres 2 et 3, nous nous sommes intéressés à la défaillance d'un seul nœud d'articulation à la fois. Or dans les scénarios réels, des segments entiers du RCSF peuvent être endommagés à cause d'une panne. Le recouvrement de la panne impose alors une restauration de la connectivité et de la couverture dans toute la zone endommagée.

Ainsi, dans le prochain chapitre, nous allons nous intéresser aux mécanismes de tolérance aux pannes dans les RCSFs à grande échelle où la défaillance de plusieurs nœuds dans le réseau peut altérer la connectivité et la couverture dans des régions entières. La restauration de la panne revient alors à rétablir la connectivité et la couverture à l'intérieur de ces régions.

Chapitre 4

Proposition d'une solution de recouvrement de pannes simultanées dans les RCSFs à grande échelle

4.1 Introduction

Dans les deux chapitres précédents, nous avons proposé des approches pour le rétablissement de la connectivité dans les RCSFs multi-canaux après l'occurrence d'une panne au niveau d'un nœud d'articulation. Ces solutions ne traitent que le cas où un seul nœud tombe en panne à la fois. Dans un contexte grande échelle, la probabilité que plusieurs nœuds tombent en panne simultanément devient importante. En effet, le déploiement de ce type de réseaux est très répandu dans le cas d'applications telles que l'agriculture de précision, les grilles électriques intelligentes, les installations de distribution de l'eau potable, *etc.* Ces applications se caractérisent par des environnements de déploiement hostiles où les problèmes d'interférences et de pannes sont très fréquents.

Dans ce contexte, nous nous focalisons sur la tolérance aux pannes pour les RCSFs à grande échelle. Le scénario de pannes auquel nous allons nous intéresser concerne la défaillance d'une zone entière du réseau. En particulier, nous nous intéresserons à une application typique de RCSF à grande échelle : l'application d'agriculture de précision pour laquelle nous allons proposer une approche de recouvrement de pannes suite à l'occurrence de plusieurs pannes simultanées dans une région du réseau.

Pour ce faire, nous exposerons d'abord, des exemples d'applications existantes pour les RCSFs à grande échelle et en particulier l'application d'agriculture de précision. Ensuite,

nous détaillerons un scénario de déploiement d'un RCSF dans une application d'agriculture de précision. Puis, nous proposerons une solution, appelé SFR-RNR (*Simultaneous Failure Recovery based on Relay Node Relocation*), de recouvrement de la panne d'une zone entière du réseau. Dans la dernière partie du chapitre, nous évaluerons les performances de la solution proposée.

4.2 Applications des RCSFs à grande échelle

De nombreuses applications nécessitent le déploiement des RCSFs à grande échelle. Dans ces applications, des milliers de capteurs sont dispersés sur une zone géographique très étendue.

4.2.1 Les grilles électriques intelligentes “Smart grids”

Les smart grids sont des réseaux électriques qui intègrent des technologies d'information et de communication pour mieux gérer les ressources et améliorer les services fournis [84–86]. Ces technologies permettent une coordination entre tous les participants dans le processus de génération, de distribution, de transmission et de consommation de l'électricité (fournisseurs, consommateurs, équipements, *etc.*). Les RCSFs jouent un rôle primordial dans le contrôle et la surveillance de tous les composants de la grille. Ainsi, ils peuvent être déployés dans les trois parties principales d'un réseau électrique :

- **La génération de l'énergie** : Dans les réseaux conventionnels, les générateurs d'énergie sont contrôlés par un nombre limité de capteurs câblés à coût élevé déployés dans quelques emplacements critiques. Un des objectifs principaux des réseaux intelligents est l'utilisation de sources d'énergie renouvelable (le vent, les cellules photovoltaïques, *etc.*) et leur intégration facile dans le processus de génération de l'électricité. Ces sources de génération sont généralement placées dans des zones éloignées et difficiles d'accès. Ainsi, les RCSFs représentent la technologie idéale pour contrôler de tels générateurs d'énergie.
- **La transmission & la distribution de l'énergie** : La panne du système de transmission (les tours, les lignes à haute tension) et/ou le réseau de distribution (les sous-stations, les transformateurs et raccordements aux consommateurs) peut causer des coupures d'électricité et peut même présenter un danger pour la sécurité publique. Ces composants doivent ainsi être contrôlés en temps quasi réel afin d'intervenir pour régler n'importe quel problème. Encore une fois, les RCSFs fournissent des solutions prometteuses pour le contrôle et la sécurité des réseaux

électriques.

- **Coté consommation** : Dans les réseaux intelligents, un flux bilatéral d'électricité et d'informations entre le fournisseur et le consommateur devient possible avec l'utilisation des métreurs (meters) intelligents et d'une infrastructure de mesure avancée (*Advancing Metering Infrastructure* AMI). Il devient alors possible de contrôler la consommation d'énergie de chaque consommateur. Ainsi, chaque appareil dans la maison moderne peut être équipé de quelques capteurs pour déterminer le coût de consommation d'énergie et le communiquer à un centre de contrôle local.

4.2.2 Surveillance sous-marine

L'objectif principal de cette application [87, 88] est d'acquérir et d'analyser des informations sur des facteurs environnementaux comme la profondeur de la mer, la température, la luminosité des fonds marins, la pollution, *etc.* En utilisant un RCSF, il devient possible de réaliser la surveillance continue des environnements sous-marins. Cependant, ces mêmes facteurs environnementaux que nous voulons analyser (vent, courants marins, marées,...) peuvent affecter les performances des RCSFs en endommageant les nœuds capteurs. Ainsi, des mécanismes de tolérance aux pannes doivent être intégrés pour ce type de réseaux.

4.2.3 Surveillance des forêts

Le déploiement des RCSFs dans les forêts permet de détecter les incendies qui détruisent des milliers d'hectares de surfaces plantées chaque année [89]. En effet, la détection rapide d'incendies aide à l'intervention efficace des pompiers avant que le feu se propage. Il permet aussi la détection des activités illégales [90] qui peuvent détruire la faune et la flore. Cependant, l'environnement ouvert des forêts rend le maintien du fonctionnement correct et continu des RCSFs un peu difficile. Par conséquent, plusieurs pannes peuvent se produire ce qui dégrade la qualité de service fourni par le RCSF déployé. Nous devons alors intégrer des mécanismes pour le redéploiement des capteurs afin d'améliorer la couverture et rétablir la connectivité si nécessaire.

4.2.4 Agriculture de précision

L'agriculture de précision [91–94] se réfère à l'utilisation d'un système d'information pour la gestion des cultures dans les champs agricoles. Autrement dit, c'est la gestion de la

variabilité spatiale et temporelle des cultures afin d'améliorer les rendements économiques et réduire l'impact environnemental. Ceci donne aux agronomes le potentiel d'appliquer de nouvelles technologies pour améliorer la productivité.

Dans l'application d'agriculture de précision, les RCSFs permettent de gérer l'irrigation, de contrôler les quantités d'engrais utilisés, de définir les caractéristiques des semences à utiliser selon les conditions des champs, etc. Les avantages des RCSFs pour cette application peuvent être résumés comme suit :

- Le contrôle d'état des cultures.
- Le contrôle d'état des champs et la capacité d'intervenir en cas d'urgence.
- La prise de décision à distance.
- Le stockage analytique d'informations pour créer un rapport de cas pour les cultures des champs.
- L'évaluation de nouvelles méthodes et techniques de culture.

Cependant, le RCSF déployé dans une telle application doit répondre à de nouvelles exigences à savoir la capacité du réseau à bien fonctionner pendant une longue période de temps sans aucune intervention humaine. De plus, le RCSF doit supporter des requêtes synchrones et asynchrones. Un autre défi important, que le RCFS doit relever, est l'auto-organisation du réseau en cas de changement de topologie en raison des pannes qui peuvent engendrer des trous de couverture et une perte de connectivité.

Dans ce chapitre, nous allons étudier les RCSFs à grande échelle dans le cadre d'une application d'agriculture de précision qui est une application prometteuse. En effet, l'engouement pour l'agriculture de précision n'a cessé de s'accroître compte tenues des limitations auxquelles est confrontée l'agriculture traditionnelle. Parmi ces limitations, nous pouvons citer les changements climatiques (effet de serre), la rareté de l'eau et l'expansion des zones urbaines par rapport aux zones agricoles. Ceci affecte la qualité et la quantité des produits agricoles, ce qui mène aux phénomènes de famine connus dans de nombreux pays. Un autre problème est l'utilisation excessive des produits chimiques (engrais chimiques, pesticides, etc.) ce qui nuit à la santé et détruit l'environnement. Ainsi, l'agriculture de précision est proposée comme une solution pour résoudre ces problèmes par l'utilisation de nouvelles technologies (communication sans fil, capteurs, drones, robots, etc.) en agriculture dans le but de mieux exploiter les ressources disponibles et augmenter la productivité des champs tout en respectant l'environnement.

4.3 Proposition d'une stratégie de déploiement d'un RCSF pour l'agriculture de précision

Dans l'agriculture de précision, la particularité de gestion des cultures réside dans le fait que le champ doit être géré selon plusieurs critères (humidité du sol, niveau de salinité, type de culture, *etc.*), ce qui révèle le besoin de diviser la zone contrôlée en des régions diverses. Chaque région peut être traitée différemment.

En se basant sur la description ci-dessus, nous proposons d'utiliser un RCSF dans lequel nous différencions deux types de nœuds :

- Des nœuds capteurs statiques de bas prix répartis sur toute la zone contrôlée (des milliers des nœuds). Ces capteurs collectent les informations de leurs voisinages et les envoient vers plusieurs nœuds puits.
- Des nœuds relais RNs (Relay Nodes) mobiles qui jouent le rôle de CHs (Cluster Heads). Les informations collectées par les nœuds capteurs doivent être envoyées aux nœuds relais pour être agrégées ; ensuite, transmises vers les nœuds puits à travers un routage multi-sauts assuré par les RNs.

4.3.1 Caractéristiques des nœuds déployés

Les caractéristiques des nœuds capteurs et des nœuds relais sont différentes vues les tâches auxquelles ils sont dédiés.

- Ainsi, un nœud capteur est :
 - Stationnaire.
 - D'un bas prix.
 - Son énergie est limitée.
 - Sa capacité de calcul est réduite.
 - Son interface radio est unique.
 - Son rayon de communication est r_c .
 - Son rayon de couverture est r_s .
- Par contre, un nœud relais est caractérisé par :
 - Sa mobilité.
 - Son énergie illimitée.
 - Sa capacité de calcul élevée.
 - Ses deux interfaces radios.
 - Deux rayons de communication : r_c et R_c (variable selon la puissance d'émission)

- Des capteurs intégrés avec un rayon de couverture r_s .

4.3.2 Topologie du RCSF

Nous divisons notre réseau en deux parties : *réseau d'acquisition de données* composé de nœuds capteurs et *réseau de transmission de données* constitué de nœuds relais. Le réseau d'acquisition assure aussi la transmission des données vers les RNs. En plus, les deux réseaux utilisent une communication multi-canal et un routage multi-saut pour assurer la propagation des données vers les nœuds puits. Afin de minimiser l'utilisation des ressources des nœuds capteurs, les RNs sont déployés d'une manière déterministe sur tout le champ de culture. Nous optons pour une stratégie de déploiement qui divise le champ à surveiller en régions hexagonales où les RNs se situent au niveau des sommets de chaque hexagone. Le pavage hexagonal est jugée optimal pour assurer la couverture totale d'un RCSF avec un minimum de RNs.. La figure 4.1 montre comment les capteurs (petits cercles) et les RNs (grands cercles) sont déployés dans la zone à contrôler.

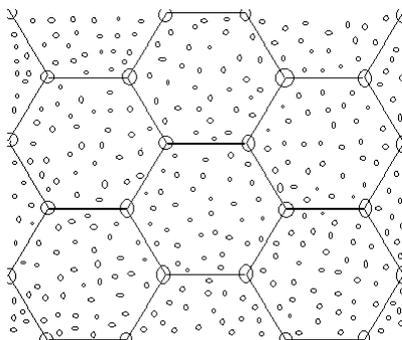


Figure 4.1 : Topologie de déploiement du RCSF

4.3.3 Routage et transmission des données

Selon l'architecture du RCSF décrite dans la figure 4.1, le champ de culture est divisé en plusieurs régions ayant une forme hexagonale ayant les RNs comme sommets. Les capteurs collectent les informations de leurs voisinages et les envoient aux RNs. Par ailleurs, la forme hexagonale que nous avons choisie permet de sélectionner un nombre minimal de RNs égal à deux pour collecter les données d'une région. Ainsi, les nœuds capteurs de chaque région peuvent se regrouper en deux clusters avec ces deux RNs comme CHs. Nous appliquons un mécanisme de clusterisation simple dans lequel chaque capteur s'associe avec le CH le plus proche dans sa région. Il en résulte que parmi les six RNs de chaque région, deux seulement sont élus pour cette région. Une fois que les données atteignent les CHs,

ces derniers appliquent des algorithmes de compression et d'agrégation pour réduire la quantité de données qu'ils vont transmettre aux nœuds puits. Ensuite, ils acheminent les paquets vers le nœud puits approprié.

Pour acheminer les données des nœuds relais vers les nœuds puits, un routage hiérarchique est utilisé. Un routage global permet de déterminer la liste des régions qui vont être traversées vers le nœud puits approprié et un routage local qui cherche la séquence des RNs à traverser. De plus, nous proposons que les routes globales soient choisies par un serveur central qui détermine le nœud puits associé à chaque région et les régions à traverser vers ce nœud puits. Ces choix se basent sur le nombre de régions à traverser, la charge de chaque région et la charge du nœud puits. Le serveur détermine aussi un nœud puits de secours au cas où le nœud puits primaire tombe en panne. Une fois la route déterminée, les informations concernant le routage global sont envoyées vers toutes les régions du réseau. Cette stratégie de routage globale est inspirée du protocole de routage ONCP [44] déployé pour les RCSFs à grande échelle.

Connaissant la région suivante, chaque RN cherche le RN suivant vers lequel les paquets vont être transmis. La procédure de sélection des routes locales fonctionne comme suit :

- Le RN u diffuse un message "*next_RN*" contenant son ID et l'ID de sa région R_i et un compteur *hop-count* (à l'initialisation *hop-count*=0).
- Si un RN v de la même région R_i ou de la région suivante R_{i+1} , mais qui n'est pas un CH pour R_{i+1} , reçoit le message "*next_RN*", il incrémente *hop-count* et rediffuse ce message.
- Si le RN v recevant le message "*next_RN*" est un CH de la région R_{i+1} , il répond par un message "*RN_candidate*" sur le chemin inverse. Ce message contient l'ID du CH, le nombre de sauts, et le nombre de paquets routés par le CH.
- Selon le nombre de sauts et le nombre de paquets routés, le RN u sélectionne la meilleure route et envoie un message "*path_confirm*" pour informer les RNs de la route choisie afin qu'ils mettent à jour leurs informations de routage.
- Le processus se répète à chaque région jusqu'à atteindre le nœud puits approprié.

De plus, afin de réduire les interférences entre les communications RN-RN et les communications RN-capteur ou capteur-capteur, nous choisissons d'utiliser deux interfaces radios au niveau des RNs. Cela permettra d'utiliser une première interface (IEEE 802.11) pour les communications inter RNs et une deuxième interface (typiquement IEEE 802.15.4) pour les communications RN-capteur ou inter capteurs. Les communications sur les deux interfaces peuvent se faire d'une manière simultanée. Sur chacune des interfaces,

nous utilisons une communication multi-canal afin de réduire les interférences sur chaque interface. La figure 4.2 montre un exemple d'affectation des canaux de communication capteur-RN, où chaque couleur désigne un canal de communication.

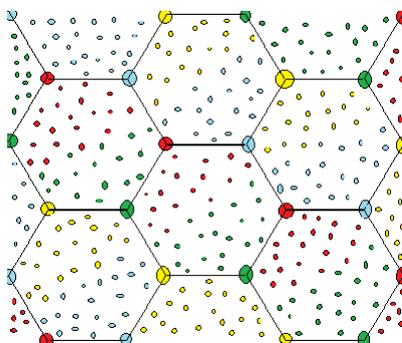


Figure 4.2 : Affectation des canaux pour la communication capteur-RN

4.4 Formulation du problème de recouvrement des pannes pour l'application d'agriculture de précision

Dans cette section, nous nous intéressons au cas où la panne concerne une région entière, où un nombre important de RNs et de capteurs tombent en panne simultanément. Ceci engendre un trou de couverture et/ou une perte de connectivité dans toute la région. Dans le cas de l'application d'agriculture de précision que nous considérons dans ce chapitre, nous pouvons prendre comme exemple de pannes l'occurrence d'un incendie ou d'une inondation. Ceci aura pour conséquence d'endommager la récolte dans une partie du champ de culture. Dans ce cas, le rétablissement de la couverture n'est plus une priorité puisque la zone affectée n'est plus exploitable pour le moment. Néanmoins, la connectivité doit être rétablie pour maintenir le bon fonctionnement du RCSF. En conséquence, nous proposons de réorganiser la zone endommagée par le déplacement de certains nœuds relais afin d'assurer la restauration de la connectivité dans cette zone. Nous avons toujours besoin de certaines informations utiles au niveau de cette zone afin de savoir à quel moment, il faudrait par exemple la ré-exploiter de nouveau, la cultiver, *etc.* Ainsi, une restauration partielle de la couverture est aussi nécessaire dans la zone endommagée. Les RNs étant équipés eux-mêmes de capteurs, les redéployer dans la zone concernée sera suffisant pour assurer à la fois couverture et connectivité dans cette zone.

Par ailleurs, nous devons veiller, dans la solution que nous proposons, à ce que la

procédure de recouvrement de la panne induise un minimum de changements dans le reste du réseau. Ainsi, la solution que nous proposons devra :

- Minimiser le nombre de RNs déplacés.
- Minimiser le nombre de capteurs réaffectés à un autre cluster.
- Minimiser la distance totale parcourue par les RNs.
- Minimiser le nombre de capteurs voisins appartenant à des clusters différents utilisant le même canal.
- Minimiser le nombre de RNs interférents ayant le même canal.

Comme nous l'avons fait dans les chapitres précédents, nous formulons le problème ci-haut sous forme d'un problème d'optimisation multi-objectif.

4.4.1 Modèle du réseau

Nous modélisons notre réseau après le recouvrement de la panne par deux graphes colorés $G = (V, E)$ et $G_0 = (V_0, E_0)$ où V désigne l'ensemble des RNs et V_0 l'ensemble des nœuds capteurs, E (respectivement E_0) désigne l'ensemble des liens d'interférence entre les nœuds relais (respectivement capteurs). Le réseau, avant la panne, est représenté par les deux graphes $G' = (V', E')$ et $G'_0 = (V'_0, E'_0)$. Nous définissons les fonctions suivantes :

- $Rn_G : V_0 \mapsto V$, donne le CH d'un capteur dans un graphe G .
- $c : V_0 \mapsto M_0$, retourne le canal d'un capteur.
- $ch : V \mapsto M$, retourne le canal d'un nœud relais.
- $pos_G : V \mapsto \mathbb{R} \times \mathbb{R}$, donne la position d'un RN dans un graphe G .

4.4.2 Formulation du problème

Nous commençons tout d'abord, par déterminer le nombre de nœuds relais déplacés vers une autre position. Ce nombre est donné par la fonction objectif f_1 comme suit :

$$f_1 = \sum_{\forall v \in V} \varphi(v) \tag{4.1}$$

où

$$\varphi(v) = \begin{cases} 1 & \text{si } pos_{G'}(v) \neq pos_G(v) \\ 0 & \text{sinon.} \end{cases} \tag{4.2}$$

Par la suite, nous calculons le nombre de capteurs qui seront affectés à un autre CH après la réorganisation du réseau. Ainsi, la fonction γ qui vérifie si un capteur donné a

changé de cluster est définie comme suit :

$$\gamma(i) = \begin{cases} 1 & \text{si } Rn_{G_0}(i) \neq Rn_{G'_0}(i) \\ 0 & \text{sinon.} \end{cases} \quad (4.3)$$

La fonction objectif f_2 , désignant le nombre total de nœuds capteurs réaffectés à un nouveau cluster est donné par :

$$f_2 = \sum_{\forall i \in V_0} \gamma(i) \quad (4.4)$$

La distance parcourue par un nœud relais est définie par :

$$dist(v) = \sqrt{(x - x')^2 + (y - y')^2} \quad (4.5)$$

où (x', y') et (x, y) sont les coordonnées d'un RN avant et après la réorganisation. Ainsi, la distance totale donné par f_3 est calculée comme suit :

$$f_3 = \sum_{\forall v \in V} dist(v)\varphi(v) \quad (4.6)$$

Ensuite, nous définissons f_4 , la fonction objectif correspondant au nombre de capteurs en conflit avec les capteurs d'un autre cluster, par :

$$f_4 = \sum_{i \in V_0} \delta_i \quad (4.7)$$

où

$$\delta_i = \begin{cases} 1 & \text{si } \exists j \in V_0 | e(i, j) \in E_0 \wedge c(i) = c(j) \wedge Rn_{G_0}(i) \neq Rn_{G'_0}(j) \\ 0 & \text{sinon.} \end{cases} \quad (4.8)$$

Enfin, nous définissons la fonction f_5 représentant le nombre de nœuds relais en conflit durant la communication RN-RN. Cette fonction sera calculée comme suit :

$$f_5 = \sum_{\forall u \in V} \lambda_u \quad (4.9)$$

avec

$$\lambda_u = \begin{cases} 1 & \text{si } \exists v \in V | e(u, v) \in E \wedge ch(u) = ch(v) \\ 0 & \text{sinon.} \end{cases} \quad (4.10)$$

Ainsi, le problème d'optimisation multi-objectif que nous cherchons à résoudre dans

le cadre de ce chapitre revient à minimiser la fonction objectif f :

$$\text{minimiser } f = [f_i]_{i=1..5} \tag{4.11}$$

Une solution exacte à un tel problème sera peu envisageable surtout dans le contexte grande échelle que nous considérons. Nous proposons la solution SFR-RNR (*Simultaneous Failure Recovery based on Relay Node Relocation*) qui vise à résoudre le problème d'optimisation donné par l'équation (4.11) en introduisant un certain nombre d'heuristiques.

4.5 Simultaneous Failure Recovery based on Relay Node Relocation : SFR-RNR

L'approche SFR-RNR permettra le redéploiement de certains RNs pour couvrir la zone endommagée et y restaurer ou améliorer la connectivité. Les algorithmes de recouvrement de pannes s'exécutent d'une manière distribuée au niveau des RNs frontaliers de la zone en panne. Un RN est dit frontalière de la zone en panne si l'un de ses voisins RNs est tombé en panne.

L'idée est de diviser la zone endommagée en régions hexagonales comme nous l'avons appliqué pour le champ initial. Cependant, la taille de nouveaux hexagones sera plus importante que celle des hexagones initiaux afin de minimiser le nombre de RNs redéployés. Cela suppose une augmentation systématique du rayon de communication au niveau des RNs. Ces RNs assureront la couverture de la surface les entourant dans la limite de leur rayon de couverture (r_s). Ceci nous donnera une couverture partielle de la zone endommagée.

Ainsi, le choix de la taille de nouveaux hexagones (déterminés par le nouveau rayon de communication R'_c) doit se faire de sorte à trouver le meilleur compromis entre le nombre de RNs déplacés, le nombre de nœuds en conflits et le taux de couverture.

4.5.1 Détection de pannes et découverte de voisinage

La détection de la panne de la zone endommagée se fait par ses nœuds relais frontaliers. En effet, périodiquement chaque RN_i de notre réseau vérifie l'état de ses RNs voisins par l'envoi de messages "heartbeat" sur sa première interface (communication RN-RN). Si l'un de ses voisins RN_j ne répond pas, RN_i fait augmenter son rayon de communication à $\sqrt{3}R_c$ et envoie des messages "heartbeat" pour vérifier l'état des ses voisins à deux sauts. En effet,

le fait que les RNs augmentent leur rayon de communication à $\sqrt{3}R_c$ leur permet de joindre leurs voisins à deux sauts sur les hexagones initiaux. Si aucun autre voisin n'est en panne, RN_i baisse son rayon de communication à R_c et reprend son fonctionnement normal. Dans le cas contraire, RN_i envoie des messages "heartbeat" sur sa deuxième interface (communication RN-capteur) pour vérifier l'état des capteurs membres de son cluster. Une fois, les réponses de ses membres reçues, RN_i diffuse un message "failure_state" dans lequel il met son ID, sa position, le nombre de ses voisins (RNs) et de ses membres (nœuds capteurs) en panne. En recevant les messages "failure_state" émis par les autres RNs, le nœud RN_i peut déterminer sa position par rapport à la zone endommagée et son état par rapport à ses voisins (nombre de capteurs fonctionnels et nombre de RNs voisins en panne). Ces informations seront utilisées par la suite dans la prise de décision de nœuds à déplacer.

Une fois la détection de la panne et la découverte du voisinage terminées, les RNs frontaliers de la zone z affectée (désormais appartenant à l'ensemble $H_1(z)$) et leurs voisins (appartenant à l'ensemble $H_2(z)$) augmentent leur rayon de communication à $R'_c > R_c$. Ce nouveau rayon de communication va servir pour la construction de nouveaux hexagones. Un exemple de topologie d'un RCSF après l'occurrence d'une panne est donné par la figure 4.3, où les points rouges désignent les RNs en pannes, les nœuds verts représentent les RNs frontaliers ($H_1(z)$) et les nœuds bleus les nœuds de l'ensemble $H_2(z)$.

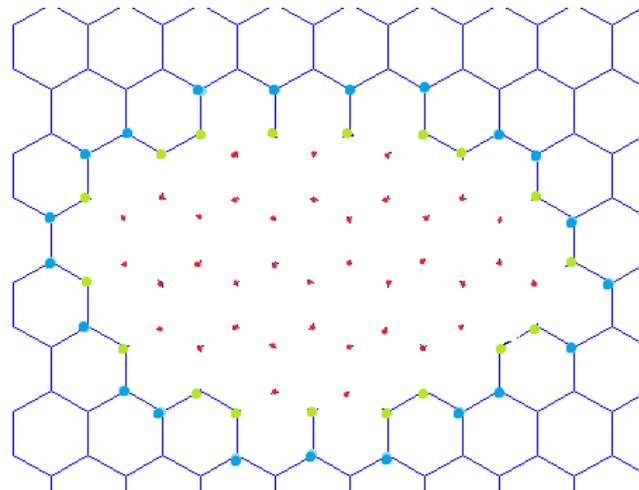


Figure 4.3 : Topologie du RCSF après une panne

4.5.2 Déplacement des nœuds relais

Le déplacement des nœuds relais s'effectue en plusieurs étapes allant des frontières de la zone endommagée jusqu'au centre. La procédure de réorganisation du réseau s'exécute comme suit :

- Le nœud relais RN_i , ayant le nombre maximal de voisins en panne et de capteurs endommagés, déclenche la réorganisation du réseau dans son voisinage.
- Le nœud RN_i se dirige vers le sommet le plus proche du nouvel hexagone construit lors de la phase de découverte de voisinage. Les RNs se déplacent toujours dans la direction du centre de la zone endommagée.
- Lorsque RN_i atteint sa nouvelle position, il diffuse un message “*discovery*” avec son ID, sa position et la distance parcourue.
- A la réception du message “*discovery*”, tous les voisins de RN_i situés à une distance R'_c doivent répondre par un message *ACK*.
- Si RN_i ne reçoit pas l'*ACK* de l'un de ses voisins, il déduit qu'il reste au moins un sommet du nouvel hexagone non encore occupé par un RN. Pour tous les sommets du nouvel hexagone, voisins de RN_i , non encore occupés, RN_i va essayer de trouver des remplaçants parmi les RNs frontaliers.
- S'il existe un voisin RN_j appartenant à $H_1(z)$ qui n'est pas déplacé, RN_i demande à RN_j de se déplacer vers l'un des sommets inoccupés.
- Si les nœuds frontaliers de la zone z ($H_1(z)$) ne sont pas suffisants pour occuper les sommets des tous les hexagones, nous passons aux nœuds de $H_2(z)$ après $H_3(z)$ et ainsi de suite.
- Si un nœud RN_j reçoit une demande de déplacement, il informe ses voisins à un saut de son déplacement et se dirige vers la position demandée. Ces derniers doivent également augmenter leurs rayon de communication à R'_c .
- Si à un moment donné, deux RNs se déplacent vers la même position, le nœud ayant parcouru la plus grande distance sera prioritaire pour occuper le sommet. Grâce aux messages “*discovery*”, il déterminera le sommet non encore occupé du nouvel hexagone et en informera le RN en déplacement et ayant parcourue une plus petite distance.
- En plus, avant de se déplacer, chaque RN jouant le rôle d'un CH doit diffuser à ses anciens membres l'ID et le canal du nouveau CH avec lequel ils seront réassociés. Ce nouveau CH sera choisi parmi les voisins de l'ancien CH de sorte à minimiser le nombre de capteurs qui lui sont associés, le nombre d'interférences, etc.

La nouvelle topologie du RCSF est donnée par la figure 4.4 qui montre les trajets

parcourus par les RNs déplacés vers leurs nouvelles positions. Les anciens emplacements sont représentés par la couleur grise.

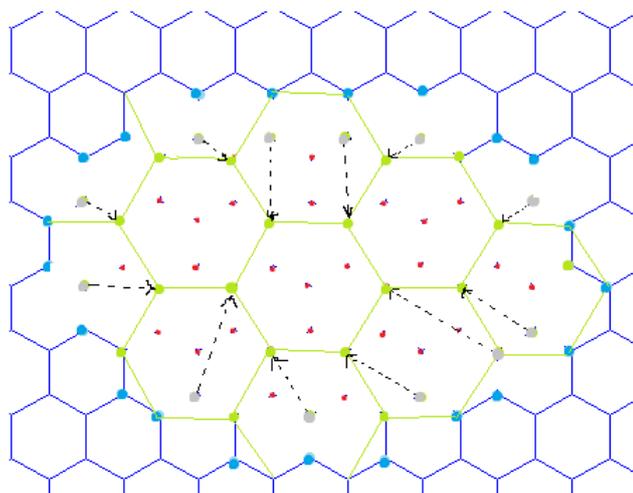


Figure 4.4 : Topologie du RCSF après le déplacement des RNs

4.5.3 Réattribution des canaux

Une fois le réseau réorganisé, les RNs déplacés doivent négocier leurs nouveaux canaux. La réallocation des canaux s'exécute de la frontière de la zone endommagée vers son centre. Les nouveaux nœuds frontaliers auront des voisins situés à une distance R'_c et des voisins situés à une distance R_c . Par contre, les nouveaux RNs situés à l'intérieur de la zone endommagée auront seulement trois voisins se situant à une distance R'_c . La réallocation des canaux suit la procédure suivante :

- Le nœud frontalier RN_i ayant le plus grand nombre de capteurs membres choisit le canal, non encore affecté, ayant le plus petit identifiant.
- Si aucun canal n'est disponible, RN_i envoie un message *“negotiate”* à ses voisins pour voir si l'un d'entre eux accepte de changer de canal.
- Chaque nœud RN_j recevant le message *“negotiate”*, vérifie si son degré d'interférence est plus petit que celui du nœud RN_i . Si tel est le cas, alors RN_j propose le changement de son canal au nœud RN_i en lui envoyant un message *“offer”*.
- Lorsque RN_i reçoit des messages *“offer”* de ses voisins, il choisit la meilleure proposition (générant le plus petit nombre de conflits) et informe le voisin ayant proposé ce canal par un message *“accept”*.
- Si aucune proposition n'est reçue, RN_i choisit le canal avec le plus petit nombre de conflits parmi les canaux disponibles.

Paramètre	Valeur
Champ	$10000m \times 10000m$
Nombre de capteurs	40000
Nombre de RNs	2400
Nombre de canaux	3
Nombre de RNs en panne	[40,100]
Surface endommagée	[5%,70%]
r_c	100m
r_s	50m
R_c	200m
R'_c	$\sqrt{3}R_c, 2R_c$

Tableau 4.1 : Paramètres de simulation

- Une fois les nœuds frontaliers ont choisi leurs canaux, les RNs suivants en direction du centre de la zone z vont réitérer la même procédure jusqu'à ce que tous les RNs déplacés vers la zone endommagée aient choisi leurs nouveaux canaux de sorte à minimiser les interférences.

4.6 Évaluation des performances de SFR-RNR

Les performances de notre approche SFR-RNR sont évaluées par simulation sous MATLAB.

4.6.1 Paramètres de simulation

Notre RCSF est créé en déployant des nœuds capteurs et des nœuds relais dans un champ étendu conformément au scénario décrit dans le paragraphe 4.3. La couverture et la connectivité sont supposées établies au niveau du champ à surveiller. Les paramètres de simulation sont présentés dans le tableau 4.1.

4.6.2 Étude des performances de SFR-RNR

Nous rappelons que nous évaluons l'approche SFR-RNR par rapport aux critères suivants :

- Le nombre de RNs déplacés.
- La distance totale parcourue par les RNs.
- Le nombre de RNs en conflit.
- Le pourcentage de couverture de la zone endommagée.

- Le nombre de capteurs réaffectés à un autre cluster.

Nous évaluons tout d'abord notre approche SFR-RNR dans le cas où la panne se produit dans une petite région de la zone contrôlée (nombre de RNs défaillant inférieur à 5% du nombre total de RNs).

Les figures 4.5 et 4.6 donnent le nombre de RNs déplacés et la distance parcourue par ces nœuds en fonction du nombre de RNs défaillants. La première figure met en évidence l'augmentation du nombre de RNs nécessaires pour remplacer les RNs en panne quand le nombre de ces derniers augmente. La deuxième figure montre que la distance parcourue par les nœuds déplacés augmente en raison de l'étendue de la zone endommagée. Néanmoins, la distance parcourue décroît avec le rayon de communication.

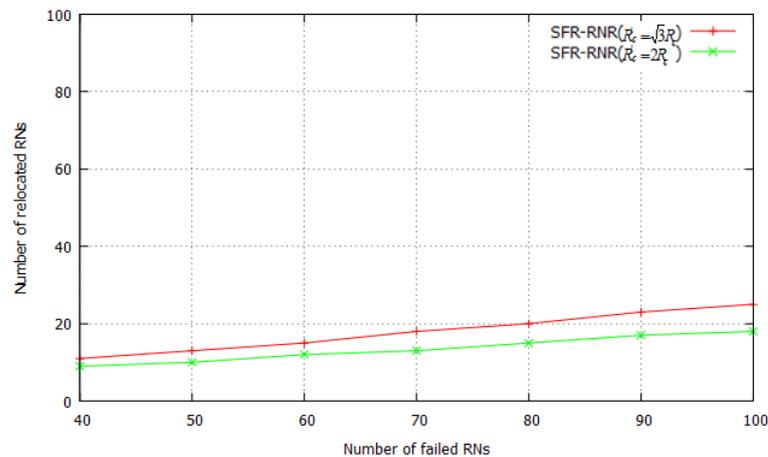


Figure 4.5 : Nombre de RNs déplacés en fonction du nombre de RNs défaillants

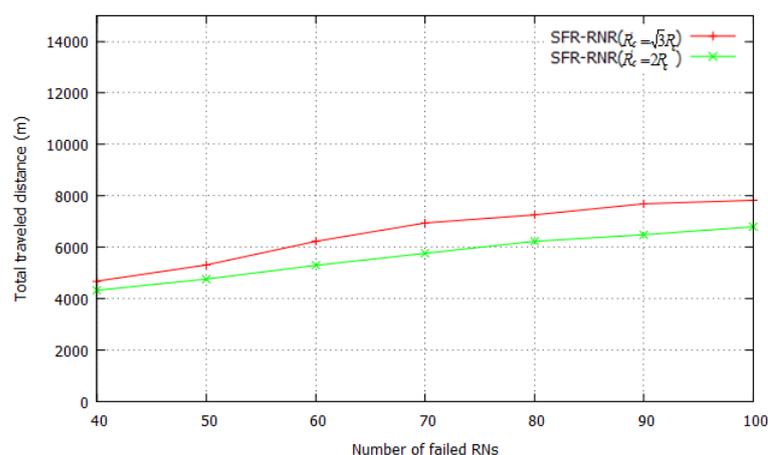


Figure 4.6 : Distance totale parcourue en fonction du nombre de RNs en panne

La figure 4.7 montre que le nombre de RNs en conflit devient plus important lorsque l'étendue de la zone endommagée est plus grande (nombre de RNs en panne plus impor-

tant). Ce résultat peut être expliqué par le fait que le nombre de RNs qui se trouvent à la périphérie de la zone endommagée devient plus important. Comme ces RNs ont déjà augmenté leur rayon de communication, il en résulte un nombre de nœuds en conflits plus grand. La variation du rayon de communication affecte aussi le nombre de RNs interférents. Nous remarquons ainsi, que l'augmentation d'un tel rayon implique un nombre d'interférences plus important entre les RNs.

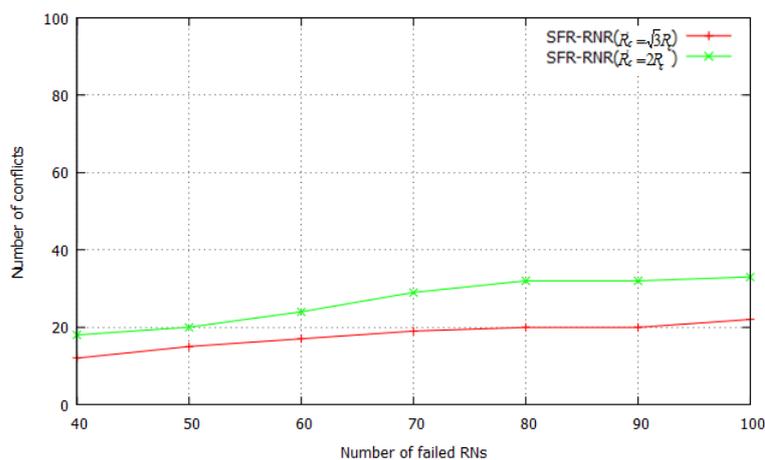


Figure 4.7 : Nombre de RNs en conflit en fonction du nombre de RNs en panne

Dans la figure 4.8, nous nous intéressons à l'évaluation des performances de la solution SFR-RNR de point de vue restauration partielle de la couverture en fonction du nombre de RNs défectueux. Avant l'occurrence de la panne, la couverture du champ de récolte est assurée à 100% par les nœuds capteurs. Après l'occurrence des pannes simultanées, la zone endommagée n'est plus de tout couverte. L'approche SFR-RNR propose d'activer les capteurs intégrés dans les nœuds relais déplacés pour assurer un minimum de couverture de la dite zone. Ainsi, le recouvrement partiel de la couverture dépend directement du nombre de RNs déplacés. Cette figure montre que la couverture diminue avec l'augmentation du nombre de RNs en panne et avec l'augmentation du rayon de communication. Par exemple, dans le cas de panne de 100 RNs, la zone endommagée sera recouverte à 10.01% avec un nouveau rayon de communication égal à $\sqrt{3}R_c$, alors que la couverture est réduite à 9.11% avec un rayon égal à $2R_c$.

Le critère de performance suivant, auquel nous nous intéressons pour le recouvrement de la panne, est le nombre de nœuds capteurs réaffectés à un nouveau cluster après ce recouvrement. En effet, ce nombre inclut les capteurs qui ont perdu leurs CHs (RNs) en raison d'une panne ou d'un déplacement. Comme illustré par la figure 4.9, ce nombre est influencé par le nombre de RNs en panne et le nombre de RNs déplacés. Ainsi, le nombre de capteurs réaffectés à un nouveau cluster devient plus important avec l'augmentation

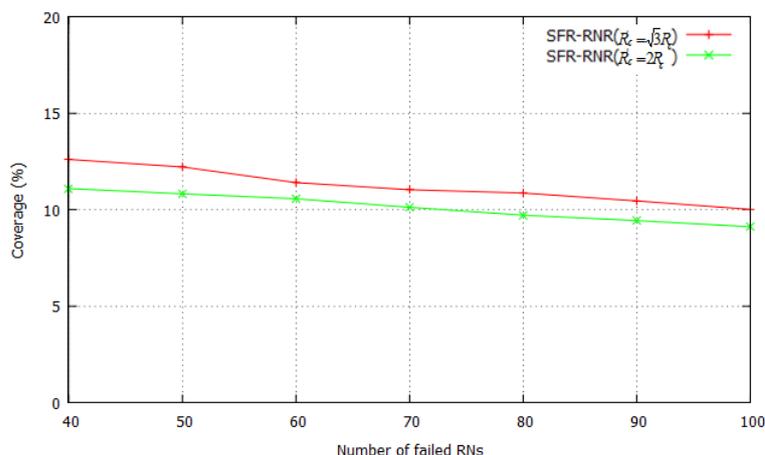


Figure 4.8 : Couverture de la zone endommagée en fonction du nombre de RNs en panne du nombre de RNs en panne. En plus, ce nombre est moins important pour un rayon de communication plus élevé (le nombre de RNs déplacés dans ce cas est plus petit).

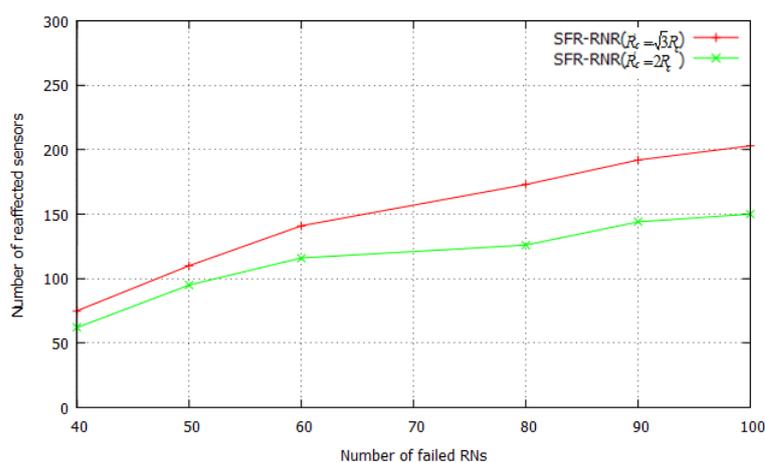


Figure 4.9 : Nombre de capteurs réaffectés à un nouveau cluster

L'évaluation des performances de l'approche SFR-RNR, faite jusque là, s'est limité au cas où l'étendue de la zone endommagée est très faible par rapport à la surface totale de la zone surveillée. Par la suite, nous évaluons notre solution dans le cas où l'étendue géographique de la zone endommagée est importante.

Les figures 4.10 et 4.11 montrent l'influence de la surface de la région endommagée sur les performances de SFR-RNR en termes de nombre de RNs déplacés et de pourcentage de ces nœuds par rapport au nombre de RNs fonctionnels.

Nous remarquons que lorsque la panne implique une région d'une surface supérieure à 40% de la surface totale de la zone contrôlée, l'approche SFR-RNR déplace 436 RNs

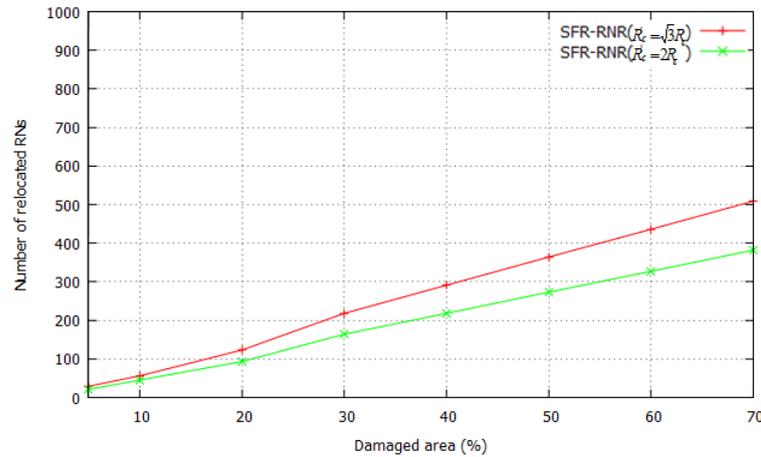


Figure 4.10 : Nombre de RNs déplacés en fonction de la surface endommagée

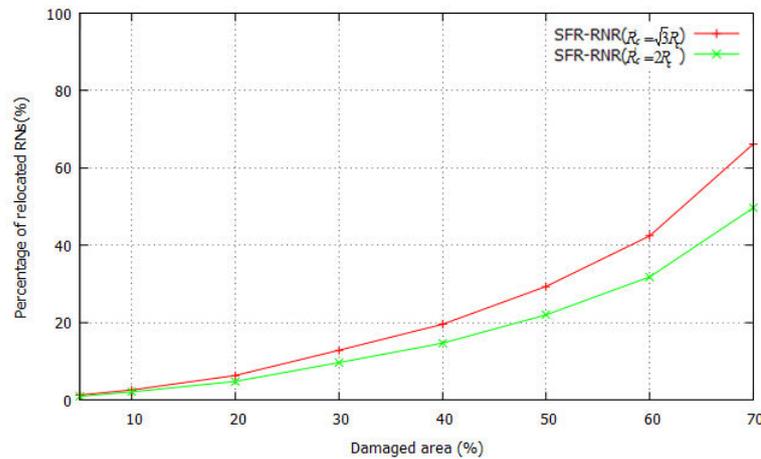


Figure 4.11 : Pourcentage de RNs déplacés par rapport au nombre de RNs fonctionnels

des 1487 RNs fonctionnels (figure 4.10). Ceci signifie que la procédure de redéploiement concerne plus de 20% des RNs fonctionnels pour un rayon de communication R'_c égal à $\sqrt{3}R_c$ (figure 4.11). Étant donné que la procédure de réorganisation implique aussi les capteurs appartenant aux clusters des RNs déplacés, il vaut mieux, dans ce cas, déplacer un nombre réduit de RNs pour restaurer ou améliorer la connectivité que d'appliquer SFR-RNR. En effet, il ne serait plus intéressant de perturber le fonctionnement de plus de 20% du réseau et altérer la couverture du champ de récolte afin de couvrir une région éventuellement inexploitable.

Nous pouvons alors conclure que notre approche SFR-RNR n'est efficace que pour le cas d'une panne limitée à une région restreinte de la zone totale surveillée (étendue de la zone endommagée inférieure à 30% de la surface totale du champ de récolte).

4.7 Conclusion

Dans ce chapitre, nous avons proposé une solution de tolérance aux pannes pour les réseaux de capteurs sans fil à grande échelle dans le contexte d'une application d'agriculture de précision. Nous avons d'abord proposé un scénario de déploiement des RCSFs pour cette application en utilisant des milliers de nœuds capteurs et de nœuds relais. Ces derniers sont déployés de sorte à former des hexagones réguliers. Ensuite, nous avons formulé le problème de recouvrement de pannes d'une zone entière du réseau sous forme d'un problème d'optimisation multi-objectif. Puis, nous avons proposé la solution SFR-RNR (*Simultaneous Failure Recovery based on Relay Node Relocation*) pour la résolution de ce problème. Cette solution utilise une stratégie de redéploiement d'un certain nombre de nœuds relais dans la zone endommagée de sorte à rétablir la connectivité et partiellement la couverture. Enfin, nous avons évalué les performances de notre approche proposée par simulation.

Les résultats de performances associés à notre solution grande échelle montrent que le nombre de nœuds relais redéployés suite à l'occurrence d'une panne est légèrement influencé par l'étendue de la zone endommagée. Cela montre que notre solution s'adapte très bien aux réseaux de grande taille.

Un récapitulatif des différents travaux effectués dans cette thèse ainsi que des perspectives à notre travail seront présentés dans le chapitre suivant intitulé "Conclusion et perspectives".

Conclusion et perspectives

Les caractéristiques des réseaux de capteurs sans fil ont permis de les déployer en masse dans différents domaines et différents environnements. Par conséquent, les réseaux de capteurs sans fil composés de ces dispositifs minuscules deviennent de plus en plus utilisés dans des applications critiques et essentielles. Cependant, ces mêmes caractéristiques rendent les RCSFs très vulnérables face à bon nombre de problèmes comme la pénurie d'énergie au niveau des nœuds capteurs et le déploiement des nœuds capteurs dans des environnements hostiles ce qui peut causer leur défaillance prématurée et rend difficile voire impossible leur remplacement. Par conséquent, il est essentiel d'intégrer des mécanismes qui permettent l'auto-rétablissement du réseau après l'occurrence d'une panne. Cette propriété de continuer à fonctionner même en présence de pannes s'appelle tolérance aux pannes.

Dans cette thèse, nous nous sommes intéressés à la problématique de la tolérance aux pannes dans les réseaux de capteurs sans fil. En particulier, nous nous sommes focalisés sur le problème d'interférences et le problème de restauration de la connectivité après l'occurrence d'une panne. Ainsi, nous nous sommes proposés d'utiliser une communication multi-canal afin de réduire les interférences dans le voisinage d'un nœud capteur. De plus, nous avons proposé des mécanismes d'auto-organisation pour le recouvrement des pannes des nœuds critiques dans le réseau, appelés nœuds d'articulation. La panne de tels nœuds entraîne la division du réseau en plusieurs segments disjoints. Ceci a pour conséquences de biaiser le comportement du RCSF qui ne sera plus en mesure de récupérer les données des segments déconnectés.

Le travail effectué dans cette thèse a porté sur trois contributions outre l'état de l'art. Nous avons tout d'abord étudié la tolérance aux pannes dans les RCSFs. En effet, nous avons présenté les concepts de base inhérents à la tolérance aux pannes dans les RCSFs. Nous avons ensuite présenté les classifications des techniques de tolérance aux pannes existantes dans la littérature.

Au niveau de la première contribution, proposée dans le chapitre 2, nous nous sommes

penchés sur l'étude de solutions centralisées pour restaurer la connectivité dans un RCSF multi-canal suite à la panne d'un nœud d'articulation. Ainsi, le recouvrement de la panne revient à rétablir la connectivité dans le réseau tout en minimisant les changements à apporter à savoir le déplacement des nœuds remplaçants, la réallocation des canaux, *etc.* Nous avons formulé notre problème de tolérance aux pannes comme un problème d'optimisation multi-objectif. Pour résoudre un tel problème, nous avons utilisé des heuristiques de coloration de graphes pour l'allocation initiale et la réallocation des canaux. En plus, nous avons utilisé les points de Steiner pour déterminer les nouveaux emplacements des nœuds à déplacer. Les mécanismes que nous avons proposés ont donné lieu à deux approches : PFR (*Preventive Failure Recovery*) et RNFR (*Rotating Nodes based Failure Recovery*). La première approche préventive est déclenchée lorsque l'énergie du nœud d'articulation tombe en dessous d'un certain seuil ce qui permet de restaurer la connectivité dans le réseau avant que la panne n'ait effectivement lieu. La deuxième approche curative se base sur la rotation de certains nœuds au voisinage du nœud d'articulation en panne afin de communiquer les informations de recouvrement entre les segments déconnectés. Une étude comparative de PFR et de RNFR a montré que PFR offre de meilleures performances. Cependant, cette approche s'applique seulement pour les pannes causées par l'épuisement de la batterie. Ce travail a fait l'objet de deux publications dans des conférences internationales et un article journal [95], [96] et [97].

Pour la deuxième contribution, qui a fait l'objet du chapitre 3, nous nous sommes intéressés à l'étude du problème de recouvrement de la panne d'un nœud d'articulation d'une manière distribuée. En effet, dans un contexte de déploiement réel, il vaut mieux que le recouvrement de la panne se fasse d'une manière locale dans le voisinage du nœud défaillant au lieu de le traiter au niveau du nœud puits. Ceci introduira un retard pour le traitement de la panne et un certain nombre de messages de contrôle. Ainsi, comme pour le cas centralisé, nous nous sommes proposés dans ce chapitre de formuler le problème de tolérance aux pannes sous forme d'un problème d'optimisation multi-objectif. En utilisant les heuristiques de coloration et de redéploiement des nœuds, nous avons proposé une première approche CR-MC (*Connectivity Restoration for Multi-Channel WSNs*). CR-MC utilise les mêmes mécanismes utilisés dans la solution centralisée mais d'une façon distribuée.

Une deuxième approche distribuée a été proposée dans la suite de troisième chapitre appelée CR-RMC (*Connectivity Restoration for Routing based Multi-Channel WSNs*). Cette approche exploite l'information de routage disponible au niveau de chaque nœud afin de réduire les interférences entre les communications simultanées. L'allocation des canaux a été effectuée de sorte à minimiser les interférences entre les nœuds n'ayant

pas un même parent dans l'arbre de routage. Nous avons aussi proposé une stratégie de mise en veille des nœuds capteurs de sorte à économiser leur énergie lorsqu'ils ne sont pas en mode de transmission. Une comparaison des performances de CR-RMC avec les approches distribuée CR-MC et centralisée RNFR a montré que CR-RMC offre de meilleures performances. Cette contribution a donné lieu à deux publications dans des conférences internationales [98] et [99].

Dans le dernier chapitre de notre thèse, nous nous sommes intéressés à la tolérance aux pannes simultanées dans les RCSFs dans un contexte grande échelle. Par contexte grande échelle, nous désignons le déploiement d'un grand nombre de capteurs sur une large zone. En effet, dans un tel contexte, la panne peut concerner non pas un seul nœud du réseau, mais une région entière.

Nous avons considéré le cas d'une application de RCSF grande échelle à savoir l'application d'agriculture de précision. Nous avons, d'abord, présenté un scénario de déploiement d'un réseau de capteurs sans fil avec des nœuds capteurs et des nœuds relais. Les nœuds capteurs collectent les informations, tandis que les nœuds relais transmettent les données vers les nœuds puits. Dans ce scénario, tous les nœuds utilisent une communication multi-canal pour réduire les interférences dans le réseau. Nous avons par la suite proposé une approche de recouvrement des pannes simultanées dans le réseau, appelée SFR-RNR (*Simultaneous Failure Recovery based on Relay Node Relocation*), basée sur le redéploiement de certains nœuds relais. Cette approche vise à restaurer partiellement la couverture et à rétablir la connectivité dans la zone endommagée. Les résultats de simulation ont montré que l'efficacité de l'approche SFR-RNR dépend de l'étendue de la zone endommagée et qu'elle devient non efficace lorsque la taille de cette zone devient importante.

Dans les différentes solutions que nous avons proposées dans le cadre de cette thèse, nous nous sommes focalisés sur des applications de collecte périodique de données non critiques. Dans la réalité, plusieurs cas de déploiement de réseaux de capteurs peuvent imposer que le trafic remonté vers le puits ne soit pas périodique (smart grids, alerte critique,...). Ce trafic doit être traité différemment du trafic périodique. Ainsi, les caractéristiques temps réel et priorité des données critiques doivent être prises en compte lors de la transmission des données. Par conséquent, il devient impossible d'appliquer les techniques proposées précédemment dans cette thèse directement. Nous pouvons, par exemple, proposer l'utilisation d'un canal dédié à la transmission des données critiques et le déplacement provisoire d'un nœud remplaçant vers l'emplacement du nœud d'articulation défaillant le temps que la procédure de recouvrement de panne habituelle ait lieu.

Comme deuxième perspective à notre travail, nous pouvons considérer le cas de réseaux en présence d'obstacles. En effet, le déploiement des nœuds capteurs dans des environnements hostiles rend la présence d'obstacles dans un tel contexte de plus en plus probable. Ainsi, le déplacement des nœuds remplaçants vers leurs nouveaux emplacements après l'occurrence d'une panne doit prendre en compte le fait que leurs trajectoires vers les nouvelles positions peuvent changer à la rencontre d'un obstacle. Les nœuds capteurs peuvent alors collaborer ensemble pour déterminer leurs nouvelles trajectoires en présence d'obstacles. Des solutions inspirées de la théorie des jeux ou de l'intelligence artificielle (colonies de fourmis, d'abeilles, etc.) peuvent être utilisées dans un tel contexte.

Comme dernière perspective, il serait intéressant de discuter la tolérance aux pannes pour les réseaux de capteurs sans fil à grande échelle pour des applications autre que l'application d'agriculture de précision à savoir les smart grids. En effet, l'architecture des réseaux électriques est sous forme quasi linéaire (une grande partie de la grille électrique se présente sous forme de lignes de transmission d'électricité à haute tension traversant les zones rurales) ce qui signifie que le RCSF déployé dans cette application se présente aussi sous forme quasi linéaire (les capteurs déployés sur les lignes à haute tension afin de contrôler par exemple la transmission de l'électricité). Il devient alors primordial de proposer des mécanismes tolérants aux pannes qui respectent une telle spécificité type du réseau.

Liste des publications

Articles acceptés

Reuves internationales

1. S. Chouikhi, I. El Korbi, Y. Ghamri-Doudane and L. Azouz Saidane, “A Survey on Fault Tolerance in Small and Large Scale Wireless Sensor Networks”, *Computer Communications Journal*, vol. 69, pp. 22–37, September 2015.
2. S. Chouikhi, I. El Korbi, Y. Ghamri-Doudane, and L. Azouz Saidane, “A Comparison of wireless sensor network Co-Simulation Platforms for Smart Grid Applications”, *International Journal of Digital Information and Wireless Communications (IJDIWC)*, vol.3 N.1, pp.93-102, 2013.

Conférences internationales

1. S. Chouikhi, I. El Korbi, Y. Ghamri-Doudane, and L. Azouz Saidane, “Articulation Node Failure Recovery for Multi-Channel Wireless Sensor Networks”, *Global Communications Conference GLOBECOM*, San Diego, CA, USA, 2015.
2. S. Chouikhi, I. El Korbi, Y. Ghamri-Doudane, and L. Azouz Saidane, “Connectivity Restoration in Multi-Channel Wireless Sensor Networks”, *International conference on Protocol Engineering and International Conference on New Technologies of Distributed Systems CFIP-NOTERE*, pp. 1-6, Paris, France, 2015.
3. S. Chouikhi, I. El Korbi, Y. Ghamri-Doudane, and L. Azouz Saidane, “Routing-based multi-channel allocation with fault recovery for wireless sensor networks”, *Proceedings IEEE International Conference on Communications ICC*, pp. 6424–6430, London, UK, 2015.
4. S. Chouikhi, I. El Korbi, Y. Ghamri-Doudane, and L. Azouz Saidane, “Centralized proactive and reactive recovery for node articulation failure in wireless sensor

- networks”, Proceedings International conference on Performance Evaluation and Modeling in Wireless Networks PEMWN, pp.49-53, Sousse, Tunisia, 2014.
5. S. Chouikhi, I. El Korbi, Y. Ghamri-Doudane, and L. Azouz Saidane, “Fault tolerant multi-channel scheme for wireless sensor networks”, Proceedings IEEE Wireless Communications and Networking Conference WCNC, pp.2474-2479, Istanbul, Turkey, 2014.

Articles soumis

1. S. Chouikhi, I. El Korbi, Y. Ghamri-Doudane, and L. Azouz Saidane, “Distributed Connectivity Restoration in Multi-Channel WSNs”, à soumettre à IEEE Transactions on Parallel and Distributed Systems.
2. S. Chouikhi, I. El Korbi, Y. Ghamri-Doudane, and L. Azouz Saidane, “Centralized Recovery from an Articulation Node Failure in Multi-Channel WSNs”, Ad Hoc Journal, Elsevier.
3. S. Chouikhi, I. El Korbi, Y. Ghamri-Doudane, and L. Azouz Saidane, “Robust Deployment of a Wireless Sensor Network in an Application of Precision Agriculture”, IEEE international symposium on Personal, Indoor and Mobile Radios Communications (PIMRC'16), Valencia, Spain, 2016.

Bibliographie

- [1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. Wireless sensor networks : a survey. *Computer Networks*, 38 :393–422, 2002.
- [2] R. Vidhyapriya and P.T. Vanathi. Energy aware routing for wireless sensor networks. *International Conference on Signal Processing, Communications and Networking*, pages 545 – 550, 2007.
- [3] Rahul C. Shah and Jan M. Rabaey. Energy aware routing for low energy ad hoc sensor networks. *IEEE Wireless Communications and Networking Conference WCNC*, 1 :350 – 355, 2002.
- [4] Y. M. Lu and V. W. S. Wong. An energy-efficient multipath routing protocol for wireless sensor networks. *Int. J. Commun. Syst.*, 20 :747 – 766, 2007.
- [5] A. Boukerche, R.W.N. Pazzi, and R.B. Araujo. Hpeq - a hierarchical periodic, event-driven and query-based wireless sensor network protocol. *In Proceedings of the IEEE Conference on Local Computer Networks*, pages 560 – 567, 2005.
- [6] H. Hassanein and J. Luo. Reliable energy aware routing in wireless sensor networks. *Proceedings of the Second IEEE Workshop on Dependability and Security in Sensor Networks and Systems DSSNS*, pages 54 – 64, 2006.
- [7] C. Intanagonwiwat, R. Govindan, and D. Estrin. Directed diffusion : A scalable and robust communication paradigm for sensor networks. *In Proceedings of the 6th Annual International Conference on Mobile Computing and Networking MobiCom, Boston, MA, USA*, pages 56–67, 2000.
- [8] D. Ganesan, R. Govindan, S. Shenker, and D. Estrin. Highly-resilient, energy-efficient multipath routing in wireless sensor networks. *Mobile Comput. Commun. Rev*, 5 :11–25, 2001.

-
- [9] B. Deb, S. Bhatnagar, and B. Nath. Reinform : Reliable information forwarding using multiple paths in sensor networks. *Proceed of the 28th IEEE Int'1 Conf. on Local Computer Networks*, pages 406–415, 2003.
- [10] A. A. Abbasi and M. Younis. A survey on clustering algorithms for wireless sensor networks. *Journal Computer Communications*, 30 :2826–2841, 2007.
- [11] O. Boyinbode, A. Mbogho H. Le, M. Takizawa, and R. Poliah. A survey on clustering algorithms for wireless sensor networks. *In Proceeding of International Conference on Network-Based Information Systems NBiS*, pages 358–364, 2010.
- [12] G. Gupta and M. Younis. Fault-tolerant clustering of wireless sensor networks. *In Proceedings of the IEEE Wireless Communication and Networks Conference WCNC*, 3 :1579 – 1584, 2003.
- [13] M. Demirbas, A. Arora, and V. Mittal. Floc : a fast local clustering service for wireless sensor networks. *In Proceedings of Workshop on Dependability Issues in Wireless Ad Hoc Networks and Sensor Networks DIWANS*, 2004.
- [14] W. R. Heinzelman and A. Chandrakasan, , and H. Balakrishnan. Energy-efficient communication protocol for wireless micro-sensor networks. *In Proceeding of HICSS 2000*, 8 :8020, 2000.
- [15] W. B. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan. An application-specific protocol architecture for wireless micro-sensor networks. *IEEE Transactions on Wireless Communications*, 1 :660 –670, 2002.
- [16] A. Manjeshwar and D. Agrawal. Teen : a routing protocol for enhanced efficiency in wireless sensor networks. *In Proceedings of 15th International Parallel and Distributed Processing Symposium IPDPS*, pages 2009–2015, 2001.
- [17] D. Agrawal A. Manjeshwar. Apteen : a hybrid protocol for efficient routing and comprehensive information retrieval in wireless sensor networks. *In Proceedings of International Parallel and Distributed Processing Symposium IPDPS*, pages 195– 202, 2002.
- [18] S. Lindsey and C. Raghavendra. Pegasus : power-efficient gathering in sensor information systems. *In Proceedings of IEEE Aerospace Conference*, 3 :1125–1130, 2002.
- [19] V. Loscri, G. Morabito, and S. Marano. A two-level hierarchy for low-energy adaptive clustering hierarchy. *Proceedings of Vehicular Technology Conference*, 3 :1809–1813, 2005.

-
- [20] O. Younis and S. Fahmy. Heed : A hybrid energy-efficient distributed clustering approach for ad hoc sensor networks. *IEEE Transactions on Mobile Computing*, 3 :366 – 379, 2004.
- [21] S. Bandyopadhyay and E. Coyle. An energy efficient hierarchical clustering algorithm for wireless sensor networks. *In Proceedings of the 22nd Annual Joint Conference of the IEEE Computer and Communications Societies INFOCOM*, 3 :1713–1723, 2003.
- [22] M. Ye, C. Li, G. Chen, and J. Wu. Eecs : An energy efficient clustering scheme in wireless sensor networks. *IEEE International Performance, Computing, and Communications Conference*, pages 535 – 540, 2005.
- [23] C. Li, M. Ye, G. Chen, and J. Wu. An energy-efficient unequal clustering mechanism for wireless sensor networks. *In Proceedings of IEEE International Conference on Mobile Adhoc and Sensor Systems Conference MASS05*, pages 604–611, 2005.
- [24] B. Gong, L. Li, S. Wang, and X. Zhou. Multi-hop routing protocol with unequal clustering for wireless sensor networks. *International Colloquium on Computing, Communication, Control, and Management ISECS*, pages 552–556, 2008.
- [25] D. Kempe, A. Dobra, and J. Gehrke. Gossip-based computation of aggregate information. *In 44th Annual IEEE Symposium on Foundations of Computer Science*, pages 482 – 491, 2003.
- [26] S. Madden, M. J. Franklin, J. Hellerstein, and W. Hong. Tag : a tiny aggregation service for ad-hoc sensor networks. *Proceedings of the Fifth Symposium on Operating Systems Design and implementation OSDI*, pages 131–146, 2002.
- [27] S. Nathy, P.B. Gibbons, S. Seshany, and Z.R. Anderson. Synopsis diffusion for robust aggregation in sensor networks. *SenSys'04*, pages 250–262, 2004.
- [28] S. Gabriel, S. Khattab, D. Mosse, J. Brustoloni, and R. Melhem. Ridesharing : Fault tolerant aggregation in sensor networks using corrective actions. *Third Annual IEEE Communications Society on Sensor and Ad Hoc Communications and Networks*, 2 :595–604, 2006.
- [29] S. S. Pradhan, J. Kusuma, and K. Ramchandran. Distributed compression in a dense microsensor network. *IEEE Signal Processing Magazine*, 19 :51–60, 2002.
- [30] J. Chou, D. Petrovic, , and K. Ramchandran. A distributed and adaptive signal processing approach to reducing energy consumption in sensor networks. *Proc. INFOCOM*, 2 :1054 – 1062, 2003.

-
- [31] V. Stankovic, A. Liveris, Z. Xiong, and C. Georghiadis. Code design for lossless multiterminal networks. *Proceedings International Symposium on Information Theory ISIT*, 2004.
- [32] S. Chessa and P. Maestrini. Fault recovery mechanism in single-hop sensor networks. *Computer Communications*, 28 :1877–1886, 2005.
- [33] K. Piotrowski, P. Langendoerfer, and S. Peter. tinydsm : A highly reliable cooperative data storage for wireless sensor networks. *International Symposium on Collaborative Technologies and Systems*, pages 225–232, 2009.
- [34] K. Benahmed, H. Haffaf, M. Merabti, and D. Llewellyn-Jones. Monitoring connectivity in wireless sensor networks. *International Journal of Future Generation Communication and Networking*, 2 :507 – 512, 2009.
- [35] X.Wang, G. Xing, Y. Zhang, C. Lu, R. Pless, , and C. Gill. Integrated coverage and connectivity configuration in wireless sensor networks. *In Proc. of SENSYS*, pages 28–39, 2003.
- [36] B. Chen, K. Jamieson, H. Balakrishnan, and R. Morris. Span : An energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks. *Wireless Networks Journal*, 8 :481–494, 2002.
- [37] H. Liu, P.J.Wan, and X. Jia. On optimal placement of relay nodes for reliable connectivity in wireless sensor networks. *Journal of Combinatorial Optimization*, 11 :249–260, 2006.
- [38] M. Younis, S. Lee, S. Gupta, and K. Fisher. A localized self-healing algorithm for networks of moveable sensor nodes. *Proc. of the IEEE Global Communications Conference GLOBECOM*, 2008.
- [39] N. Tamboli and M. Younis. Coverage-aware connectivity restoration in mobile sensor networks. *IEEE International Conference on Communications ICC*, pages 1–5, 2009.
- [40] K.I. Gandhi, P. Narayanasamy, and P. Ushapreethi. Coverage holes reduction in movable sensor networks using minimum movement technique. *IEEE International Conference on Recent Trends in Information Technology, ICRTIT*, pages 275 – 280, 2011.
- [41] Y. Chen, E. Chan, and S. Han. Energy efficient multipath routing in large scale sensor networks with multiple sink nodes. *In Proc. APPT*, pages 390–399, 2005.

-
- [42] M. Hammoudeh, A. Kurz, and E. Gaura. Mumhr : Multi-path, multi-hop hierarchical routing. *In Proceedings of International Conference on Sensor Technologies and Applications*, pages 140–145, 2007.
- [43] J. Chen, Z. Yin, D. Li, , and T. Sun. A distributed and effective cluster routing protocol of sensor networks. *In Proceedings of 1st International Conference on Intelligent Networks and Intelligent Systems*, pages 271–275, 2008.
- [44] T. Wu and S. Biswas. Off-network control for scalable routing in very large sensor networks. *In Proceedings of 2007 IEEE International Conference on Communications ICC*, pages 3357 – 3363, 2007.
- [45] T. Wu, F. Yu, and S. Biswas. Scalable hybrid routing in very large sensor networks. *In Proceedings of 8th International Conference on Mobile Data Management*, pages 366 – 370, 2007.
- [46] H. Lu, J. Li, , and G. Wang. A novel energy efficient routing algorithm for hierarchically clustered wireless sensor networks. *In Proceedings of 4th International Conference on Frontier of Computer Science and Technology*, pages 565–570, 2009.
- [47] L. Chitnis, A. Dobra, and S. Ranka. Aggregation methods for large-scale sensor networks. *ACM Transactions on Sensor Networks TOSN*, 4, 2008.
- [48] C. Luo, F. Wu, J. Sun, and C. W. Chen. Compressive data gathering for large-scale wireless sensor networks. *MOBICOM*, pages 145–156, 2009.
- [49] M. Lin, C. Luo, F. Liu, and F. Wu. Compressive data persistence in large-scale wireless sensor networks. *In Proceedings of IEEE GLOBECOM*, pages 1–5, 2010.
- [50] S. A. Aly, Z. Kong, and E. Soljanin. Fountain codes based distributed storage algorithms for large-scale wireless sensor networks. *International Conference on Information Processing in Sensor Networks*, pages 171 – 182, 2008.
- [51] S. A. Aly, Z. Kong, and E. Soljanin. Raptor codes based distributed storage algorithms for wireless sensor networks. *In IEEE International Symposium on Information Theory*, pages 2051 – 2055, 2008.
- [52] K. Pearson. The problem of the random wal. *Nature*, page 294, 1905.
- [53] M. Luby. Lt codes. *In Proc. 43rd Symposium on Foundations of Computer Science*, pages 271 – 280, 2002.

-
- [54] A. Shokrollahi. Raptor codes. *IEEE Transactions on Information Theory*, 52 :2551–2567, 2006.
- [55] P. Gonizzi, G. Ferrari, and V. Gay. Data dissemination scheme for distributed storage for iot observation systems at large scale. *Journal of Information Fusion*, 22 :16–25, 2015.
- [56] H. Zhang and J. C. Hou. Maintaining sensing coverage and connectivity in large sensor networks. *Wireless Ad Hoc and Sensor Networks*, 1 :89–123, 2005.
- [57] M. Younis and R. Waknis. Connectivity restoration in wireless sensor networks using steiner tree approximations. *GLOBECOM*, pages 1–5, 2010.
- [58] F.K. Hwang, D.S. Richards, and P. Winter. The steiner tree problem. *Elsevier*, 1992.
- [59] A. Abbasi, M. Younis, and K. Akkaya. Movement-assisted connectivity restoration in wireless sensor and actor networks. *IEEE Trans. on Parallel and Dist. Sys*, 20 :1366–1379, 2009.
- [60] S. Lee and M. Younis. Recovery from multiple simultaneous failures in wireless sensor networks using minimum steiner tree. *J. Parallel and Dist. Sys*, 70 :525–536, 2010.
- [61] A. Alfadhly, U. Baroudi, and M. Younis. Least distance movement recovery approach for large scale wireless sensor and actor networks. *In Wireless Communications and Mobile Computing Conference IWCMC*, pages 2058 – 2063, 2011.
- [62] M. Jahanshahi, M. Maddah, and N. Najafzadegan. Energy aware distributed partitioning detection and connectivity restoration algorithm in wireless sensor. *International Journal of Mathematical Modelling & Computations Networks*, 3 :71– 82, 2013.
- [63] H. S. AbdelSalam and S. Olariu. Bees : Bioinspired backbone selection in wireless sensor networks. *IEEE Transactions on Parallel and Distributed Systems*, 23 :44 – 51, 2012.
- [64] T. T. Truong, K. N. Brown, and C. J. Sreenan. Autonomous discovery and repair of damage in wireless sensor networks. *38th Annual IEEE Conference on Local Computer Networks*, 2013.
- [65] S. Azzaz and L. A. Saidane. Cmsa : a centralized maintenance strategy with anticipation for wireless sensor networks. *International Journal of Digital Information and Wireless Communications IJDIWC*, pages 97–105, 2013.

- [66] S. Chouikhi, I. El Korbi, Y. Ghamri-Doudanec, and L. Azouz Saidane. A survey on fault tolerance in small and large scale wireless sensor networks. *Computer Communications, Elsevier*, 69 :22–37, 2015.
- [67] Y. Wu, J. Stankovic, T. He, and S. Lin. Realistic and efficient multi-channel communications in wireless sensor networks. *In Proceedings INFOCOM*, pages 1867–1875, 2008.
- [68] K. R. Chowdhury, N. Nandiraju, D. P. Agrawal P. Chanda, and Q. A. Zing. Channel allocation and medium access control for wireless sensor networks. *Ad Hoc Networks*, 7 :307–321, 2009.
- [69] O. D. Incel, L. Van Hoesel, P. Jansen, and P. Havinga. Mc-lmac : A multi-channel mac protocol for wireless sensor networks. *Ad Hoc Networks*, 7 :73–94, 2011.
- [70] G. Zhou, C. Huang, T. He T. Yan, J. A. Stankovic, and T. F. Abdelzaher. Mmsn : Multi-frequency media access control for wireless sensor networks. *In Proceedings INFOCOM*, 2006.
- [71] Y. Kim, H. Shin, and H. Cha. Y-mac : An energy-efficient multi-channel mac protocol for dense wireless sensor networks. *Proceedings IPSN*, pages 53–63, 2008.
- [72] Y. Yang and W. Yi. Rainbow : reliable data collecting mac protocol for wireless sensor networks. *In Proceedings WCNC*, 2010.
- [73] G. Robins and A. Zelikovsky. Tighter bounds for graph steiner tree approximation. *SIAM Journal on Discrete Math.*, 2005.
- [74] C. A. Coello, G. B. Lamont, and D. A. Van Veldhuizen. Evolutionary algorithms for solving multi-objective problems. *Springer*, 2007.
- [75] Matlab : The language of technical computing. *Online <http://www.mathworks.com/products/matlab/>*.
- [76] G. Robins and A. Zelikovsky. Tighter bounds for graph steiner tree approximation. *SIAM J. on Discrete M Math.*, 19 :122–134, 2005.
- [77] A. Hertz and D. De Werra. Using tabu search technique for graph coloring. *Computing Journal*, pages 345–351, 1987.
- [78] A. Abbasi, M. Younis, and K. Akkaya. Movement-assisted connectivity restoration in wireless sensor and actor networks. *IEEE Trans. on Parallel and Dist. Sys*, pages 1366–1379, 2009.

- [79] Discrete event simulator omnet++. <https://omnetpp.org/>.
- [80] A. Saifullah, Y. Xu, C. Lu, and Y. Chen. Distributed channel allocation protocols for wireless sensor networks. *IEEE Transactions on Parallel and Distributed Systems*, 2, 2014.
- [81] Y. Chen, P. H. Gomes, and B. Krishnamachari. Multi-channel data collection for throughput maximization in wireless sensor networks. *In proceedings international conference on Mobile Ad Hoc and Sensor Systems MASS*, pages 443 – 451, 2014.
- [82] X. Wang, X. Wang, X. Fu, G. Xing, and N. Jha. Flow-based real-time communication in multi-channel wireless sensor networks. *Wireless Sensor Networks, Lecture Notes in Computer Science, Springer*, 5432 :33–52, 2009.
- [83] S. Xiong and J. Li. An efficient algorithm for cut vertex detection in wireless sensor networks. *IEEE International Conference on Distributed Computing Systems ICDCS*, pages 1366–1379, 2010.
- [84] NIST. Framework and roadmap 4 for smart grid interoperability standards, release 2.0. *Draft NIST*, 2011.
- [85] W. Wang, Y. Xu, and M. Khanna. A survey on the communication architectures in smart grid. *Comput. Networks*, 55 :3604–3629, 2009.
- [86] S.L. Ullo and A. Vaccaro. Gossip sensor networks for power quality monitoring in smart grids. *International Conference on Renewable Energies and Power Quality ICREPQ*, 2012.
- [87] M. Ayaz, I. Baig, A. Abdullah, and I. Faye. A survey on routing techniques in underwater wireless sensor networks. *Journal of Network and Computer Applications*, 34 :1908–1927, 2011.
- [88] J. Heidemann, W. Ye, J. Wills, A. Syed, and Y. Li. Research challenges and applications for underwater sensor networking. *Wireless Communications and Networking Conference WCNC*, pages 228 – 235, 2006.
- [89] Y. E. Aslan, I. Korpeoglu, and Ö. Ulusoy. A framework for use of wireless sensor networks in forest fire detection and monitoring. *Computers, Environment and Urban Systems, Elsevier*, 36 :614–625, 2012.

-
- [90] J. Papán, M. Jurecka, and J. Púchyová. Wsn for forest monitoring to prevent illegal logging. *Proceedings of the Federated Conference on Computer Science and Information Systems*, pages 809 – 812, 2012.
- [91] D.D.Chaudhary, S.P.Nayse, and L.M.Waghmare. Application of wireless sensor networks for greenhouse parameter control in precision agriculture. *International Journal of Wireless and Mobile Networks IJWMN*, 5, 2011.
- [92] K. Konstantinos, X. Apostolos, and K. Panagiotis. Topology optimization in wireless sensor networks for precision agriculture applications. *International conference on Sensor Technologies and Applications SensorComm*, pages 526 – 530, 2007.
- [93] H. Sahota, R. Kumar, and A. Kamal. A wireless sensor network for precision agriculture and its performance. *Wireless Communication and Mobile Computing, Wiley Online Library*, 11 :1628–1645, 2011.
- [94] M. Keshtgari and A. Deljoo. A wireless sensor network solution for precision agriculture based on zigbee technology. *Wireless Sensor Network*, 4 :25–30, 2012.
- [95] S. Chouikhi, I. EL Korbi, Y.Ghamri-Doudane, and L. A. Saidane. Fault tolerant multi-channel allocation scheme for wireless sensor networks. *in proceedings of Wireless Communication and Networks Conference WCNC*, pages 2438–2443, 2014.
- [96] S. Chouikhi, I. El Korbi, Y. Ghamri-Doudanec, and L. Azouz Saidane. Centralized proactive and reactive recovery for node articulation failure in wireless sensor networks. *The International Conference on Performance Evaluation and Modeling in Wired and Wireless Networks PEMWN*, 2014.
- [97] S. Chouikhi, I. El Korbi, Y. Ghamri-Doudanec, and L. Azouz Saidane. Articulation node failure recovery for multi-channel wireless sensor networks. *Globecom*, 2015.
- [98] S. Chouikhi, I. El Korbi, Y. Ghamri-Doudanec, and L. Azouz Saidane. Connectivity restoration in multi-channel wireless sensor networks. *CFIP-NOTERE*, 2015.
- [99] S. Chouikhi, I. El Korbi, Y. Ghamri-Doudanec, and L. Azouz Saidane. Routing-based multi-channel allocation with fault recovery for wireless sensor networks. *Proceedings International Conference on Communications ICC*, pages 6424–6430, 2015.