République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université A/Mira de Béjaïa
Faculté des Sciences Exactes
Département de Recherche Opérationnelle

Mémoire de Fin de cycle

En vue de l'obtention du diplôme Master en Recherche Opérationnelle

Spécialité : Modilisation Mathimatique et Evaluation de Performance Réseau

Thème

La tolérance aux pannes dans les réseaux de capteur sans fil

Réalisé par :

Mr MOKRANI Fahem.
Mr ZAGHAR Azzouz.

Devant le jury composé de :

Promotrice : Mme Rebouh Nadjette.
Présidente : Mme Ouyahia Samira.
Examinatrice : Mme Belkhiri Louiza.

Université de Béjaïa, 2016

Dédicaces

Je dédié ce modeste travail :

À ceux qui ont fait de moi ce que je suis, ceux grâce à qui tant d'années d'études ont été possibles, ceux envers qui j'ai une dette imprescriptible mes chères ***parents.***

Puisse ***Dieu****, le Très Haut, vous accorder santé, bonheur et longue vie et faire en sorte que jamais je ne vous déçoive.*

À mes chers ***frères*** *et* ***sœurs*** *et mes adorables* ***neveux*** *et* ***nièces***

À mes ***amis*** *sons exception.*

Et à tous ceux ou celles qui ont contribué de prés ou de loin à la réalisation de ce travail.

Azzouz

Je dédié ce modeste travail :
À mes chères ***parents****, Et à tous ceux ou celles qui ont contribué de près ou de loin à la réalisation de ce travail À mes chers* ***frères*** *et* ***sœurs****, À ma chère adorable* ***nièce Elyna***

À mes ***amis*** *sons exception.*

.

Fahem

Remerciements

Nous remercions le bon **Dieu** le tout puissant qui nous a accordé le courage, la patience et la force pour réaliser ce mémoire.
Tout d'abord nous tenons à remercier M^{me} S.Ouyahia et M^{lle} Rabouh pour leur encadrement et leurs précieux et judicieux conseils qu'elles n'ont cessé de nos prodiguer tout au long de ce projet, leurs confiance témoignée, sans oublier leurs qualités humaines.
Nos remerciements vont également aux membres de jury d'avoir accepte de juger notre travail.
Et à tous les enseignants et le personnel du département de Recherche Opérationnelle.

Liste des Acronymes

ACK	Acknowledgment (Acquittement)
ADCs	Analog-to-Digital Converters
ADV	ADVertisement
ADVT	ADVerTisement
AGRNTF	Aggregator Notification
AODV	Ad-Hoc On-Demand Distance Vector
CPEQ	Cluster-based PEQ
CTS	Clear To Send
DARPA	pour Defense Advanced Research Projects Agency
DMRF	Dynamical Jumping Real-Time Fault-Tolerant Routing Protocol for Wireless Sensor Networks
EAR	An Energy and Activity Aware Routing Protocol for Wireless Sensor Networks in S
ENFAT-AODV	ENhancedFAult-Tolerant AODV
FAT2D	Fault Tolerant Directed Diffusion for Wireless Sensor Networks
FATE-CSQ	FAult Tolerant Evaluation of Continuous Selection Queries
FCS	Set candidat Forwarding
FMS	Fixed Mobile Substitution
FCS	Set candidat Forwarding
HSEND	Hierarchal Sensor network Debugging
ISM	Industrial Scientific Medical bands
KAT-mobiliy	K-means And TSP-based mobility
MAC	Medium Access Control
MTBF	Mean Time Between Failure
MTTF	Mean Time To Failure
MTTR	Mean Time To Repair
PEGASIS	Power-Efficient Gathering in Sensor Information Systems
PEQ	Periodic, Event-driven, Query-based
RCSF	Réseau de Capteurs Sans Fil
REPEN	ReplyEnergy
REQEN	RequestEnergy

RERP	An Adaptive Fault Tolerant Routing Protocol with Error Reporting Scheme for Wireless Sensor Networks
RPT	RePorT
RREP	Route REPly
RREQ	RouteREQuest
RTS	Request To Send
SensIT	Sensor Information Technology
SETAGR	Set Aggregator
TDMA	Time Division Multiple Access
TFD	Timing Failure Detection
TSP	Traveling-Salesman Problem
TTL	time to live

Table des matières

Table des figures

Introduction générale

Les développements technologiques dans le domaine de la micro-électronique ont permis l'introduction d'un nouveau type de composants " les capteurs électroniques " qui sont constitués dans la majeure partie des cas d'un microprocesseur, d'une mémoire vive, d'une interface radio et d'une source d'énergie.

Un réseau de capteurs sans fil (RCSF) est composé généralement d'un grand nombre de capteurs, capables de s'auto-organiser et communiquent via des liens sans fil. Le but général d'un RCSF est la collecte d'un ensemble de paramètres de l'environnement entourant ces capteurs, telles que la température ou l'humidité et les acheminer vers des points de collecte distants appelés puits ou stations de base.

Vu le faible coût de ces capteurs, de la large gamme disponible sur le marché (allant du capteur de position, capteur de son, capteur météorologique jusqu'aux capteurs équipés de caméras, etc), de leur capacité de s'auto configréer, de se gérer sans qu'il y ait besoin d'interventions humaines, de leur facilité de déploiement même dans des zones hostiles ainsi que de leur tolérance aux pannes les RCSFs sont présents dans plusieurs domaines tel que : le domaine militaire pour surveiller les mouvements des forces ennemies, ou analyser le terrain avant d'y envoyer des troupes, dans le domaine de la surveillance que ce soit dans le milieu du bâtiment pour détecter par exemple les fissures ou les altérations de la structure, aussi bien qu'en guise de système d'alarme sécuritaire ainsi que dans les domaines environnemental, domestique ou sanitaire.

Dans les réseaux sans fil, on donne plus de l'importance à l'acheminement de l'information qui est assuré par des algorithmes de routage. De ce fait, il est judicieux de concevoir des protocoles de routage qui doivent prendre en considération les changements de la topologie du réseau, ainsi que d'autres caractéristiques comme la bande passante, le nombre de liens, la limitation d'énergie, etc. En outre, les capteurs sont des dispositifs fragiles sujets à des pannes multiples et dans certains cas il y aurait des capteurs défaillants qui sont impliqués dans l'acheminement des données à la station de base. Il résulte à cet effet que les données ne peuvent arriver correctement à cette dernière. Pour remédier à cette anomalie, plusieurs protocoles de routage tolérants aux pannes ont été proposés dans la littérature.

L'objectif de ce mémoire est de traiter le problème de tolérance aux pannes dans les réseaux de capteurs pour garantir un routage efficace, surtout ceux à taille importante. Le souci principal est d'assurer la livraison de données à la station de base tout en prolongeant la durés de vie du réseau.

Pour mener à bien notre travail, nous l'avons organisé en quatre chapitre sellon un plan méthodologique suivant :

Dans le premier chapitre nous introduisons des généralités sur les caractéristiques des réseaux de capteurs aussi que ses domaines d'application et les différents facteurs et contraintes liée à sa conception.

Dans le second chapitre, nous donnons des généralités sur la tolérance aux pannes, les sources de ces pannes et une classification des solutions.

Ensuite, nous présentons dans le troisième chapitre quelques protocoles de routage tolérants aux pannes et les différantes solutions appliquées, nous concluons ce chapitre avec une comparaison théorique de ces protocoles selon plusieurs critères.

Enfin le dernier chapitre consiste à présenter notre proposition pour l'amélioration de protocole de routage PEQ, cette nouvelle solution pour la tolérance aux pannes permet d'améliorer les performances de ce dérnier en minimisant sa consommation d'énergie, réduire le trafic sur le réseau et augmenter le taux de livraison des paquets, et nous terminons ce chapitre avec la presentation d'un simulateur capable d'introduire les principaux point de notre amélioration. Nous clôturons par une conclusion générale et quelque perspectives.

1

Généralité sur les réseaux de capteurs sans fil

1.1 Introduction

Les récentes avancées dans les domaines des technologies sans-fil et l'électroniques ont permis le développement à faible coût de minuscules appareils consommant peu d'énergie et qui peuvent communiquer entre eux via des liens radio, appelés capteurs. Ils coopèrent entre eux pour former une infrastructure de communication appelée réseau de capteurs.
Dans ce chapitre, nous allons présenter un ensemble de définitions sur les réseaux de capteurs sans fil, leurs caractéristiques, leurs architectures et leurs domaines d'application. Nous allons discuter également les principaux facteurs et contraintes qui influencent la conception des réseaux de capteurs sans fil.

1.2 Un nœud capteur

1.2.1 Qu'est ce qu'un capteur ?

C'est un système qui sert à détecter, sous forme de signal souvent électrique, un phénomène physique afin de le transformer en un signal utilisable.
Un capteur est un petit appareil doté de mécanismes lui permettant de relever des informations sur son environnement s'ils se trouvent à l'intérieur de leur rayon de perception.
La nature de ces informations varie très largement selon l'utilisation qui est faite du capteur : ce dernier peut tout aussi bien faire des relevés de température, d'humidité ou d'intensité lumineuse.

Un capteur possède également le matériel nécessaire pour effectuer des communications sans fil par ondes radio [1].

1.2.2 Architecture d'un nœud capteur

Un nœud capteur est composé, en général, des composants suivants : un matériel de perception, une mémoire, une batterie, un processeur embarqué et une unité de communication. Il peut avoir aussi d'autres composants dépendamment de l'application comme un système de localisation, un générateur d'énergie et un mobilisateur.
La figure 1.1 montre les différents composants d'un capteur.

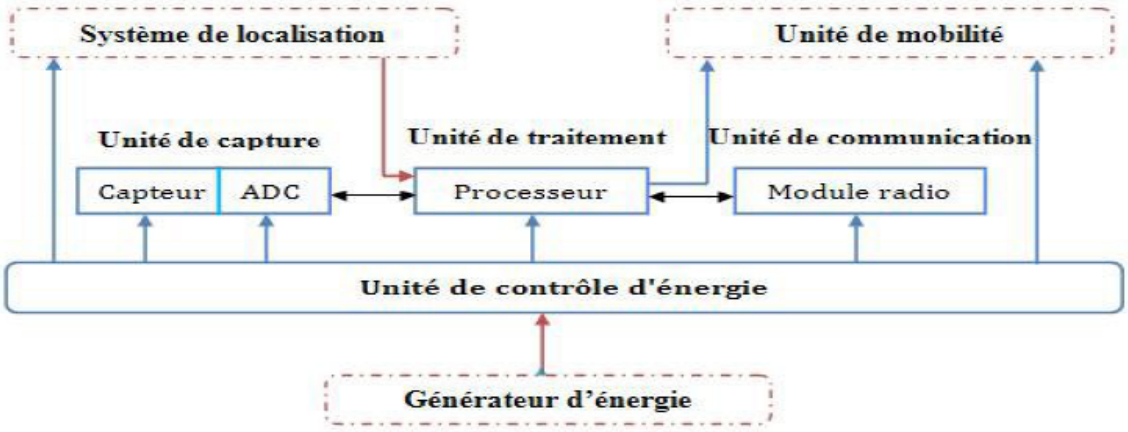

FIGURE 1.1 – Architecture de base d'un capteur.

- ***L'unité de capture*** : est généralement composée de deux sous-unités : le capteur et le convertisseur analogique-numérique (ADCs Analog-to-Digital Converters). Le capteur obtient des mesures numériques sur les paramètres environnementaux et les transforme en signaux analogiques. Le ADC convertit ces signaux analogiques en signaux numériques avant de les envoyer à l'unité de traitement.

- ***L'unité de traitement*** : elle est composée de deux interfaces qui sont une interface avec l'unité de captage et une autre avec le module de transmission. Elle contrôle les procédures permettant au nœud de collaborer avec les autres nœuds pour réaliser les tâches d'acquisition et stocker les données collectées.

- ***L'unité de communication***(transceiver) : elle est responsable de toutes les communications via un support de communication radio qui relie le nœud aux autres nœuds du réseau.

- ***L'unité d'énergie*** : est le composant le plus important, et qui fournit l'énergie nécessaire pour le fonctionnement du capteur [4].

1.3 Réseaux de capteurs sans fil

1.3.1 Définition

Un réseau de cateurs sans fil (RCSF) est composé d'un grand nombre de nœuds capteurs qui sont liés par un medium sans fil et qui sont déployés en masse soit dans le phénomène à observer ou à surveiller soit très près de lui. Les positions des nœuds capteurs n'ont pas besoin d'être prédéterminées. Ceci permet un déploiement aléatoire dans les terrains inaccessibles ou les opérations de secours lors d'un désastre. Les nœuds capteurs communiquent pour former une infrastructure de communication, rassemblent des mesures et envoient les résultats à une station de base [2].

1.3.2 Architecture

L'utilisateur accède à distance aux données capturées à travers un nœud appelé le nœud directeur de tâche. Le nœud directeur de tâche est relié à l'Internet ou au satellite à travers un nœud destinataire "station de base" (puits). Ce dernier agit en tant que passerelle pour le réseau de capteurs, c'est-à-dire il relie des réseaux de capteurs à d'autres réseaux.
Les nœuds capteurs sont habituellement dispersés dans une zone de capture appelée champ de captage. Ils rassemblent les données et les conduisent au destinataire.
Notons qu'un réseau de capteurs peut contenir plusieurs stations de base. De cette manière, les utilisateurs peuvent rechercher l'information dans les nœuds destinataires pour surveiller et commander l'environnement à distance [1].

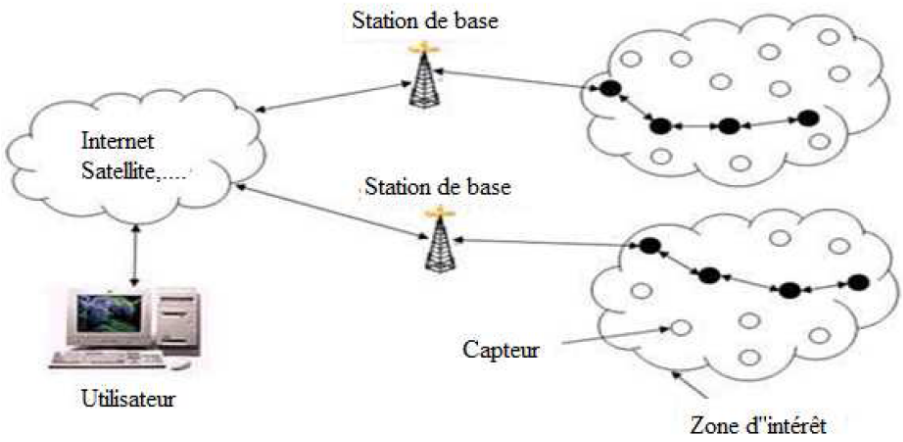

FIGURE 1.2 – Architecture d'un réseau de capteurs.

1.4 Domaines d'application

Plusieurs types d'applications peuvent être développés pour les réseaux de capteurs sans fil. Selon le mode de communication des données de mesure, on identifie quatre grands scénarios d'applications :

- ***Applications périodiques*** : les capteurs prennent des mesures dans des intervalles de temps réguliers et ils envoient les données vers la station de base de manière périodique.

- ***Applications à la demande***(On-Demand) : les capteurs attendent de recevoir un ordre de la station de base pour déclencher une mesure et l'envoyer.

- ***Applications événementielles***(Event-Driven) : dans ce type d'applications, l'envoi de données vers la station de base est déclenché lorsqu'un événement particulier est détecté.

- ***Applications hybrides*** : toute alliance des cas précédents.
  Les réseaux de capteurs sans fil ont trouvé un ensemble très vaste d'applications dans divers domaines, parmi lesquelles on peut citer [5] :

1.4.1 Applications militaires

Comme beaucoup d'autres technologies de l'information, les réseaux de capteurs sans fil proviennent principalement de la recherche militaire. Des réseaux de capteurs autonomes sont envisagés comme l'ingrédient essentiel dans cette lancée vers des systèmes de guerre centrés sur les réseaux. Ils peuvent être rapidement déployés et utilisés pour la surveillance des champs de bataille afin de fournir des renseignements concernant l'emplacement, le nombre, le mouvement,

et l'identité des soldats et des véhicules ou bien encore pour la détection des agents chimiques, biologiques et nucléaires.

Les applications militaires sont les premières et certainement les plus représentatives des applications trouvées actuellement dans le domaine des réseaux de capteurs sans fil. Une grande partie de cette croissance rapide a été apportée par des programmes financés par l'Agence américaine pour les Projets de Recherche Avancée de Défense (DARPA pour Defense Advanced Research Projects Agency), notamment grâce à un programme connu sous le nom de " SensIT " (Sensor Information Technology) de 1999 à 2002 [6].

1.4.2 Applications liées à la sécurité

L'application des réseaux de capteurs dans le domaine de la sécurité peut diminuer considérablement les dépenses financières consacrées à la sécurisation des lieux et des êtres humains. Ainsi, l'intégration des capteurs dans de grandes structures telles que les ponts ou les bâtiments aidera à détecter les fissures et les altérations dans la structure suite à un séisme ou au vieillissement de la structure [7].

1.4.3 Applications environnementales

Les études scientifiques des habitats écologiques (animaux, végétaux, micro-organismes) sont traditionnellement effectuées grâce à des activités sur le terrain par des enquêteurs. Un problème majeur dans ces études provient de ce qui est parfois appelé " l'effet de l'observateur". En effet, la présence et les activités potentiellement intrusives des enquêteurs sur le terrain peuvent affecter le comportement des organismes dans l'habitat supervisé et ainsi fausser les résultats des observations. Des réseaux de capteurs sans fil sans surveillance promettent une nouvelle approche écologique d'observation à distance pour la surveillance de l'habitat. En outre, les réseaux de capteurs, en raison de leur grande échelle potentielle et d'une haute densité spatio-temporelle, peuvent fournir des données expérimentales d'une richesse sans précédent [6].
D'autres applications environnementales sont destinées à la surveillance de certains phénomènes climatiques afin de détecter ou de prévoir certaines catastrophes naturelles telles que l'éruption des volcans, les inondations et les incendies de forêt.

1.4.4 Applications médicales

Dans le domaine de la médecine, les réseaux de capteurs peuvent être utilisés pour assurer une surveillance permanente des organes vitaux de l'être humain grâce à des micro-capteurs qui pourront être avalés ou implantés sous la peau (surveillance de la glycémie, détection de cancers, etc.). Ils peuvent aussi faciliter le diagnostic de quelques maladies en effectuant des mesures physiologiques telles que : la tension artérielle, battements du cœur, etc, à l'aide des capteurs ayant chacun une tâche bien particulière.

Les données physiologiques collectées par les capteurs peuvent être stockées pendant une longue durée pour le suivi d'un patient. D'autre part, ces réseaux peuvent détecter des comportements anormaux (chute d'un lit, choc, cri, etc.) chez les personnes dépendantes (handicapées ou âgées) [7].

1.4.5 Applications commerciales

Il est possible d'intégrer des capteurs au processus de stockage et de livraison dans le domaine commercial. Le réseau ainsi formé pourra être utilisé pour connaître la position, l'état et la direction d'un paquet. Il devient alors possible pour un client qui attend la réception d'un paquet, d'avoir un avis de livraison en temps réel et de connaître la localisation actuelle du paquet.
Pour les entreprises manufacturières, les réseaux de capteurs permettront de suivre le procédé de production à partir des matières premières jusqu'au produit final livré. Grâce aux réseaux de capteurs, les entreprises pourraient offrir une meilleure qualité de service tout en réduisant leurs coûts [7].

1.5 Facteurs et contraintes conceptuels des RCSFs

La conception des RCSFs, leurs protocoles et algorithmes sont guidés par plusieurs facteurs, ci dessous une liste de quelques facteurs à considérer lors de la conception d'un réseau de capteurs :

1.5.1 Coût de production

Le coût de production d'un seul capteur est très important pour l'évaluation du coût global du réseau. Si ce dernier est supérieur à celui nécessaire pour le déploiement des capteurs classiques, l'utilisation de cette nouvelle technologie ne serait pas financièrement justifiée. Par conséquent, réduire le coût de production des nœuds est un objectif important pour la faisabilité de la solution des réseaux de capteurs sans fil [1].

1.5.2 Passage à l'échelle

Une des caractéristiques des RCSFs est qu'ils peuvent contenir des centaines voir des milliers de nœuds capteurs. Le réseau doit être capable de fonctionner avec ce nombre de capteurs tout en permettant l'augmentation de ce nombre et la concentration (densité) des nœuds dans une région [1].

1.5.3 La protection de l'information

Comme pour tout réseau sans fil, l'information circule sur une interface partagée et non dédiée. N'importe quel intrus peut alors soit récupérer l'information, soit la modifier ou la rendre

inexploitable. C'est pourquoi des mesures de sécurité doivent être mise en place pour protéger l'information.

1.5.4 Contraintes liées à l'application

Les nœuds peuvent être déployés pour relever des mesures pour une infinité de situations et dans des environnements très variables tout en ayant une concentration faible ou forte des capteurs. L'environnement de déploiement peut être à l'intérieur d'une grosse machine, au fond d'un océan, dans un lieu contaminé biologiquement ou chimiquement, dans un champ de bataille, dans une maison ou un immeuble, sur un animal, sur un véhicule, etc. Ces situations très variées engendrent des contraintes très fortes de l'environnement sur les nœuds capteurs.

1.5.5 La topologie du réseau

Le déploiement d'un grand nombre de nœuds nécessite une maintenance de la topologie. Cette maintenance consiste en trois phases [5] :

- **La phase de pré-déploiement et de déploiement** : les capteurs peuvent être soit jetés en masse ou bien placés un par un dans le champ de perception. Ils peuvent être déployés en les lançant d'un avion, délivrés d'un obus d'artillerie, fusée, ou missile et placés soit par un humain ou un robot.
- **La phase de post-déploiement** : le changement de la topologie est dû au changement de la position des nœuds, la portée, l'énergie disponible et le dysfonctionnement d'un ou plusieurs nœuds.
- **La phase de redéploiement de nœuds additionnels** : des capteurs additionnels peuvent être déployés à n'importe quel moment pour remplacer les nœuds défectueux [4].

1.5.6 Interaction avec l'environnement

Puisque ces réseaux interagissent avec l'environnement, les caractéristiques de leur trafic diffèrent de celui des autres réseaux sans fil. Une conséquence typique est que les réseaux de capteurs ont un taux de mesure de données faible, mais on peut avoir aussi des rafales de trafic dans certains scénarios, comme dans le cas d'une catastrophe ou d'un événement exceptionnel [4].

1.5.7 Media de transmission

Les nœuds communiquants sont reliés de manière sans fil. Ce lien peut être réalisé par radio ou un signal infrarouge. Il faut s'assurer de la disponibilité du moyen de transmission choisi dans l'environnement de capture afin de permettre au réseau d'accomplir la totalité de ses tâches. Pour les liens de communication via les fréquences radio, les bandes ISM (Industrial Scientific Medical bands) peuvent être utilisées.
Pour les réseaux de capteurs, les unités de transmission intégrées au niveau des nœuds doivent être

de petite taille et à faible consommation d'énergie. En effet, les contraintes matérielles associées aux nœuds, ainsi que le compromis existant entre l'efficacité des antennes et la consommation d'énergie, limitent le choix de la bande de fréquence utilisée sur les bandes à hautes fréquences [1].

1.5.8 La consommation d'énergie

Dans les réseaux de capteurs, la consommation d'énergie est une métrique de performance très importante puisque généralement les capteurs sont déployés dans des zones inaccessibles. Ainsi, il est difficile voire impossible de remplacer les batteries après leur épuisement. De ce fait, la consommation d'énergie au niveau des capteurs a une grande influence sur la durée de vie du réseau. L'énergie totale consommée par un nœud capteur a pour origine trois fonctions principales selon [2] :

- ***L'acquisition des données*** : cette tâche est effectuée par le composant d'acquisition des données qui traduit les phénomènes physiques en signal électrique et il peut être digital ou analogique. Il existe plusieurs types de ce composant qui mesurent les paramètres de l'environnement comme la température, le son, l'image, la pression, etc. Les sources de consommation d'énergie dans ces composants peuvent être : l'échantillonnage des signaux et la conversion des signaux physiques en signaux électriques et la conversion analogique-numérique [2].
- ***Les traitements*** : cette tâche inclut le contrôle des composants d'acquisition des données et l'exécution des protocoles de communication et des algorithmes de traitement de signaux sur les données collectées. Elle est effectuée par les microprocesseurs. Le choix de ces derniers est fonction du scénario de l'application, et il fait en général un compromis entre le niveau de performance et la consommation d'énergie. En plus, ces microprocesseurs supportent plusieurs modes d'opérations, incluant le mode actif, le mode libre ou "idle", et le mode en veille, pour la gestion de l'énergie. La quantité d'énergie consommée diffère d'un mode à un autre, mais les transitions entre ces modes consomment de l'énergie et du temps. Ainsi, le degré de consommation d'énergie des différents modes, le coût des transitions, et la durée que le microprocesseur passe dans chaque mode ont tous une influence significative sur la consommation totale de l'énergie [2,4].
- ***La communication*** : le support de transmission le plus utilisé est la radio. Plusieurs facteurs affectent les caractéristiques de la consommation d'énergie de la radio, tels que le type du système de modulation, le taux de données, la puissance de transmission (déterminée par la distance de transmission) et le cycle d'activité opérationnel . En général, les radios peuvent fonctionner dans quatre modes d'opération différents : transmission, réception, libre, et le mode veille. Quand la radio est en mode libre, elle consomme de l'énergie comme si elle est en mode réception, alors il est important d'éteindre complètement la radio si le nœud n'est pas en transmission ou en réception. Un autre facteur important est que les transitions entre les différents modes consomment de l'énergie et du temps [2,4].

1.5.9 Tolérance aux pannes

Certains nœuds capteurs peuvent être bloqués ou tomber en panne à cause d'un manque d'énergie, dommage physique ou d'une interférence. La panne d'un nœud capteur ne doit pas affecter la globalité de la tâche de réseau de capteurs. C'est le problème de fiabilité ou de tolérance aux pannes. La tolérance aux pannes est donc la capacité de maintenir les fonctionnalités du réseau sans interruption due à une panne d'un nœud capteur [8].
Dans le cadre de ce travail, nous nous intéressons exclusivement à la tolérance aux pannes dans les RCSFs.

1.6 La connectivité

La connectivité d'un réseau dépend principalement de l'existence des routes. Elle est affectée par les changements de topologie due à la mobilité et la défaillance des nœuds qui mènent au partitionnement du réseau et à l'isolement des nœuds. La connectivité de réseau peut être modélisé par un graphe $G(V, E)$ ou V représente l'ensemble des capteurs (nœuds) et E l'ensemble des liens.

1.6.1 1-Connectivité

Un réseau est totalement connecté si pour chaque pair de nœuds, il existe une route les reliant. En d'autres termes, chaque nœud du réseau peut communiquer avec n'importe quel autre nœud grâce au multi-sauts. Dans un réseau non connecté nous avons plusieurs sous réseaux connectés. Chaque sous réseaux est formé d'un ensemble de nœuds communiquant entre eux mais qui ne peuvent pas communiquer avec d'autre nœuds appartenant à d'autre sous réseaux [8].

1.6.2 k-Connectivité

Un réseau est dit k-connecté (k= 1, 2, 3,. . ., n) si pour chaque pair de nœuds il existe au moins k chemins différents les reliant [8].

1.7 La couverture

1.7.1 La couverture de zone dans les RCSFs

Les capteurs fonctionnent avec un modèle à seuil, c'est à dire qu'un capteur possède deux zones : une zone de perception (SR) et une zone de communication (CR) comme le montre la figure 1.3.

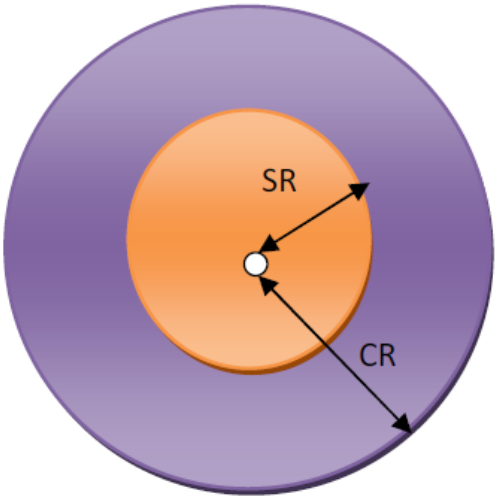

FIGURE 1.3 – La couverture dans une zone.

En influant sur le rapport entre le rayon du SR et le rayon du CR, on va modifier les contraintes. Ainsi, on va pouvoir minimiser le nombre de nœuds actifs et maximiser la durée de vie du réseau. Les zones CR et SR représentent la zone de couverture d'un capteur. Pour qu'une zone soit complètement couverte, il faut que la densité de capteurs soit suffisante.
Comme les capteurs sont généralement déployés aléatoirement sur une zone d'intérêt, il est nécessaire de disposer d'une densité importante de capteurs. Si la densité de capteurs est trop importante et que la zone que l'on veut surveiller est "trop" couverte, alors des capteurs vont fonctionner inutilement. De ce fait, il faut ordonnancer le mode d'activité des capteurs en mettant quelques capteurs en mode veille tout en assurant la couverture totale de la zone [8].

1.7.2 La k-couverture de surface dans les RCSFs

Pour assurer une couverture totale de la zone d'intérêt, un mécanisme de tolérance aux pannes basé sur la couverture multiple de tout point de la zone peut être utilisé. Ce mécanisme est appelé la k-couverture. Dans ce mécanisme tout point de la zone de déploiement est couvert par au moins k capteurs. Ce qui permet de tolérer la défaillance de (k-1) capteurs au niveau de chaque point de la zone [8].

1.8 Conclusion

Les réseaux de capteurs sans fil se sont développés pour être déployée dans plusieurs domaines. Ces RCSFs sont confrontés à des défaillances multiples. De ce fait, il est nécessaire de concevoir des protocoles qui tolèrent les pannes dans ces reseaux. Dans le chapitre qui suit, nous détaillons la notion de tolérance aux pannes dans les RCSFs.

2

La tolérance aux pannes dans les RCSFs

2.1 Introduction

Certains capteurs peuvent être bloqués ou tomber en panne à cause d'un manque d'énergie, d'un dégât matériel ou d'une interférence environnementale. La panne d'un capteur ne doit pas affecter le fonctionnement global de son réseau. C'est le problème de fiabilité ou de tolérance aux pannes. La tolérance aux pannes a pour objectif de maintenir les fonctionnalités du réseau sans interruption due à une panne d'un composant du RCSF.
Dans ce chapitre, nous présentons d'abord quelques généralités sur la tolérance aux pannes et les sources possibles des pannes, ensuite nous donnons un état de l'art sur l'ensemble de techniques existantes qui nous permettent de détecter des pannes et de restaurer le bon fonctionnement du système dans les réseaux de capteurs.

2.2 Sûreté de fonctionnement

La Sûreté de fonctionnement est définie par l'aptitude du réseau à maintenir ses fonctionnalités, en cas de panne ou anomalie. Elle vise donc à minimiser l'influence de ces pannes sur la tâche globale du réseau [8].
La figure 2.1 résume les notions associées à la sûreté de fonctionnement présentées ci-dessous.

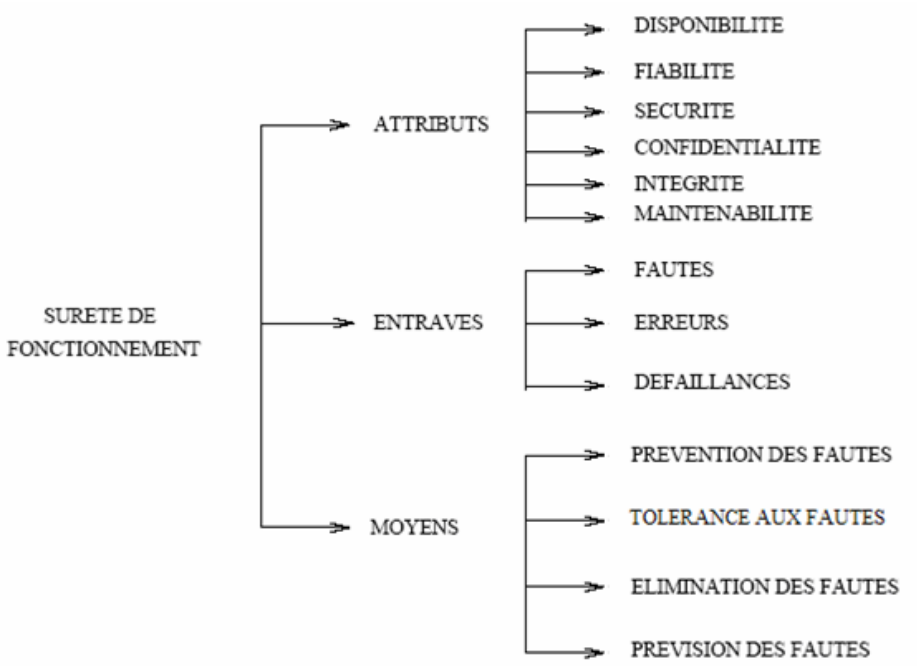

FIGURE 2.1 – L'arbre de la sûreté de fonctionnement.

2.2.1 Attributs de la sûreté de fonctionnement

Les attributs de la sûreté de fonctionnement d'un système mettent plus ou moins l'accent sur les propriétés que doit vérifier la sûreté de fonctionnement du système. Ces attributs permettent d'évaluer la qualité de service fournie par un système [11]. Les attributs de la sûreté de fonctionnement sont définis par :

- ***La fiabilité*** $R_k(t)$: est l'aptitude d'un système à accomplir une fonction requise dans des conditions données et ce pendant une durée donnée. Une mesure de cette fiabilité concerne le temps entre deux défaillances consécutives (MTBF : temps moyen entre deux défaillances). Pour une période de temps de durée t, la MTBF est liée à la fiabilité par une distribution de Poisson qui indique la probabilité de ne pas avoir un échec dans l'intervalle de temps $[0, t]$ [8] :

$$R_k(t) = \exp^{\lambda_k t} \tag{2.1}$$

  Où $\lambda_k = \frac{1}{MTTF}$ est le taux de défaillance du nœud capteur k, t c'est la période de temps et MTTF (temps moyen jusqu'à la défaillance) : ou temps moyen de bon fonctionnement.

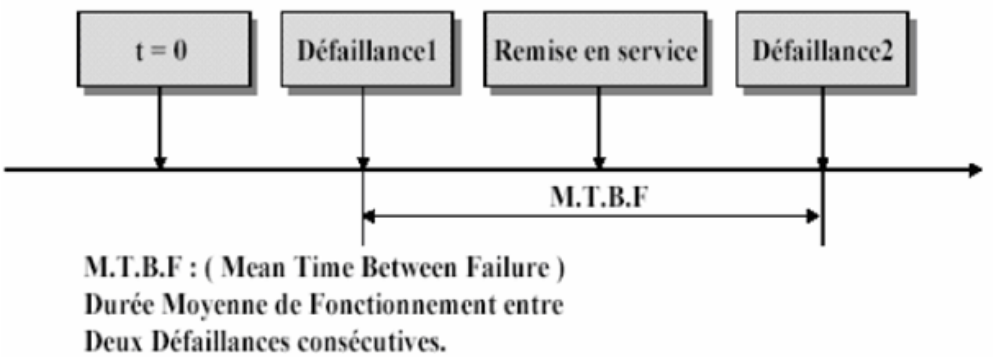

FIGURE 2.2 – Définition du MTBF.

- ***La disponibilité*** : est l'aptitude d'un système à être en état d'accomplir une mission dans des conditions données, à un instant donné et en supposant que la fourniture des moyens extérieurs nécessaires soit assurée. Elle caractérise les risques de dysfonctionnement d'un système et sa capacité à y faire face [10].
  La disponibilité regroupe les notions de fiabilité et de maintenabilité. La disponibilité d'un système est donc étroitement liée à la fiabilité, puisqu'elle est définie comme la probabilité pour laquelle le système fonctionne correctement à un moment donné. Elle est liée à la (MTBF : temps moyen entre deux fautes) et à la durée moyenne de réparation (MTTR : durée moyenne de reprise) par la relation suivante :

$$Disponibilite = MTBF/(MTTR + MTBF) \tag{2.2}$$

  $MTBF = MTTF + MTTR$
  Une disponibilité élevée, peut donc, être obtenue par un long MTTF ou par une durée moyenne de reprise MTTR courte. Cette situation est illustrée par la figure 2.3.

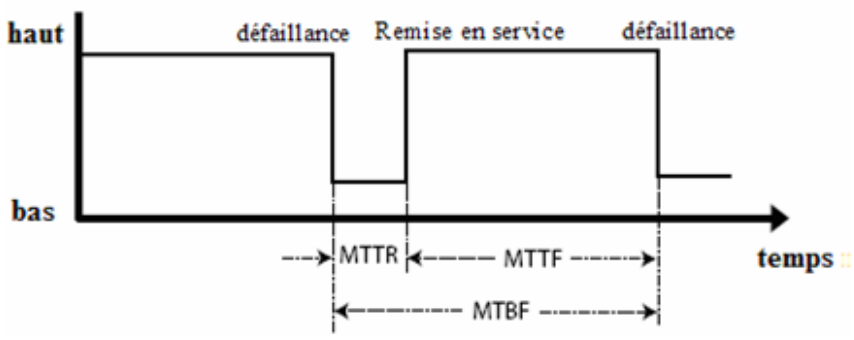

FIGURE 2.3 – Relation entre MTTF, MTTR et MTBF.

- ***La maintenabilité*** : est l'aptitude d'un système à être maintenu ou rétabli dans un mode de fonctionnement normal, lui permettant de fournir un service, lorsque la maintenance est

effectuée dans des conditions données et avec des moyens prescrits. C'est aussi la probabilité que la maintenance d'une entité E, assurée dans des conditions données avec des moyens et procédures prescrites, s'achève à un instant t2 pour un instant d'occurrence de la défaillance t1 [8].

M(t1,t2) = P[entité défaillante à t1 soit réparée à t2]

- ***La crédibilité*** : précise le comportement en cas d'anomalies du système. Elle repose sur la notion d'intégrité et de sécurité.

- ***L'intégrité*** : représente l'assurance du système à accomplir correctement la mission pour laquelle il a été conçu. Cette assurance implique bien entendu sa capacité à reconnaître le mode de fonctionnement dans lequel se trouve le système (normal ou anormal) et à signaler ce mode de fonctionnement.

- ***La confidentialité*** : cet attribut évalue la capacité du système à fonctionner en dépit de fautes intentionnelles et d'intrusions illégales.

- ***La sécurité*** : est l'aptitude du système à éviter de faire apparaître, dans des conditions données,des événements catastrophiques ou l'assurance du système à résister à des entrées non autorisées ou incorrectes et à pouvoir les signaler.

2.2.2 Entraves à la sûreté de fonctionnement

Uue entrave à la sûreté de fonctionnement peut être :

- ***La panne*** : il s'agit d'une sorte de dysfonctionnement matériel ou logiciel qui peut causer des erreurs, par exemple, un défaut matériel ou un bogue de programme [8].
- ***L'erreur*** : une erreur représente un état anormal du système qui peut éventuellement conduire le système à une faillite totale [11].
- ***La défaillance*** : la manifestation des erreurs qui se produit lorsque le système a dévié de sa spécification et ne peut plus livrer des fonctionnalités prévues [8].

2.2.3 Moyens d'assurer la sûreté de fonctionnement

La conception et le développement d'un système de fonctionnement sûr passe par l'utilisation combinée d'un ensemble de méthodes qui peuvent être classées en :

- ***La prévention des pannes*** : la propriété de prévention des pannes est définie par l'aptitude du système à maintenir ses moyens permettant d'éviter l'occurrence de pannes dans le système. Ce sont généralement les approches de vérification des modèles conceptuels.
- ***L'élimination des pannes*** : une fois une panne détectée et localisée, il faut que le système

réagisse rapidement. Il peut soit isoler la partie en panne, soit essayer de la réparer. Le but est de laisser le système continuer à fonctionner.
Une méthode qui est largement utilisée est la redondance. C'est-à-dire que les composants qui sont essentiels pour le bon fonctionnement du système sont répliqués pour augmenter sa fiabilité en cas de pannes sévères. Dans un réseau de capteur, les composants à répliquer sont souvent des nœuds assurant des services particuliers, par exemple les nœuds passerelles dans des protocoles de routage. Les nœuds ne sont pas capables de gérer des mécanismes très complexes, c'est aussi la raison pour laquelle les nœuds sont censés être déployés en masse, car on peut souvent trouver un grand nombre de nœuds qui sont disponibles. On peut les utiliser pour soit substituer les nœuds en panne, soit assurer simultanément une seule fonctionnalité.

- ***La prévision des pannes*** : théoriquement, il est impossible d'éviter complètement les pannes, et les anomalies qui surviennent durant le fonctionnement du réseau car ils ne sont pas toujours prévisibles. Par contre, il est possible de capturer un état erroné du système et de faire le nécessaire pour qu'il puisse continuer à fonctionner, même d'une manière réduite. Après le déploiement et la mise en service d'un réseau, les nœuds doivent effectuer périodiquement des opérations diagnostiques pendant la transmission et le traitement des données. Lorsqu'une erreur est détectée, ils la signalent immédiatement aux autres nœuds qui s'en occupent pour réagir le plus vite possible [10].

2.2.4 Tolérance aux pannes

La tolérance aux pannes vise à éviter les défaillances et elle est mise en œuvre par la détection des erreurs et le rétablissement du système.

2.2.4.1 Détection d'erreurs

La détection d'erreurs est une procédure cruciale sans laquelle le système ne peut pas choisir et prendre la réaction adaptée pour minimiser les impacts des défaillances et les services qu'il fournit risque d'être suspendus. Il s'agit généralement d'un diagnostic de fonctionnement d'un composant spécifique, parfois, on essaie aussi d'avoir une prédiction de pannes, c'est-à-dire une analyse des symptômes observés.
Dans un réseau de capteurs, les nœuds sont censés être déployés en masse et une maintenance manuelle n'est pas intéressante du tout, car trop coûteuse et compliquée. Dans cette section, nous présentons un ensemble de techniques qui nous permettent d'automatiser la procédure de détection de pannes dans les réseaux de capteurs.

- ***Diagnostic local*** : dans certains cas, un nœud est capable de détecter des erreurs en faisant un diagnostic local. Afin de tester les connexions avec ses voisins, une table de routage contenant les identifiants des voisins est crée en mémoire locale, le nœud diffuse périodiquement des paquets de test aux voisins et attend des réponses. Le nœud est soupçonné d'être isolé, si aucun voisin ne lui répond, ou ils répondent, mais avec une puissance de signal très faible [12].

- ***Diagnostic en groupe*** : il est aussi possible de détecter les erreurs du système causées par des valeurs erronées, si les nœuds disposent d'une valeur de référence. Une application qui est largement utilisée est la collection de données, on supposant que les nœuds qui se trouvent dans la même zone sont censés capturer des valeurs similaires. Si on trouve une valeur qui possède une grande différence par rapport aux autres, cela est souvent considéré comme un symptôme d'erreur [3].
- ***Diagnostic hybride*** : c'est en effet la combinaison des deux modes de diagnostic présentés ci-dessus. C'est-à-dire que les nœuds effectuent d'abord chacun un diagnostic local, mais il y a toujours des chances que les résultats ne soient pas assez précis. Ensuite, les nœuds échangent les résultats du diagnostic local pour faire un diagnostic de groupe [3].

2.2.4.2 Rétablissement du système

Suivant les moyens utilisés pour reconstruire un état correct, trois formes de rétablissement du système ont été identifiés : la reprise, la poursuite et la compensation d'erreur.

- ***Reprise*** : le système est ramené dans un état antérieur à partir duquel il reprend son fonctionnement (un exemple de la vie quotidienne consiste à rallumer son ordinateur après qu'il soit bloqué, et à travailler à partir de la dernière version sauvegardée de travail).
- ***Poursuite*** : le système est amené dans un nouvel état, comme à priori à partir duquel il peut continuer son fonctionnement (souvent de manière dégradée). Ce mode de recouvrement est souvent très dépendant de l'application.
- ***Compensation*** : la redondance continue dans l'état erroné. Il suffit d'éliminer l'erreur et de poursuivre l'exécution.

La technique la plus utilisée généralement pour le recouvrement d'une faute est la réplication ou la redondance des composants qui sont enclins pour être défectueux. Par exemple, les RCSF, sont habituellement utilisés pour surveiller périodiquement une région et envoyer les données captées à une station de base. Si quelques nœuds ne fournissent pas des données, la station de base reçoit toujours des données suffisantes. Le routage multi chemins est un autre exemple, dans le cas d'une seule route, une requête où les données ne peuvent pas être acheminées si quelque, nœud, ou liens le long de la route échouent. Garder un ensemble des chemins de réserve fournit la fiabilité élevée des routes pour le routage.

2.3 Tolérance aux pannes dans les réseaux de capteurs sans fil

Dans les réseaux de capteurs, on doit assurer la fidélité de détection c'est-à-dire tout point de la zone d'intérêt est couvert par au moins un capteur. Certains nœuds capteurs peuvent être bloqués ou tomber en panne à cause d'un manque d'énergie, d'un dégât matériel ou d'une interférence environnementale. La panne d'un nœud capteur ne doit pas affecter le fonctionnement global de

son réseau. C'est le problème de fiabilité ou de tolérance aux pannes. La tolérance aux pannes est donc la capacité de maintenir les fonctionnalités du réseau sans interruption due à une anomalie.

2.3.1 Sources des pannes dans les réseaux de capteurs

Une panne (faute) désigne une défaillance qu'elle soit matérielle ou logicielle, temporaire ou définitive, d'un ou plusieurs composants du système. Les défaillances dans un RCSF peuvent survenir dans trois niveaux.

2.3.1.1 pannes au niveau du nœud

Un nœud est composé de différents composants matériels et piloté par des logiciels, qui peuvent subir éventuellement des pannes tout au long de son fonctionnement. Par exemple le boitier du nœud risque d'être cassé suite à des impacts physiques et qui peuvent endommager des matériels encapsulés dedans. En outre, lorsque le niveau de pile d'un nœud devient faible, il y a des chances que ses composants ne puissent plus fonctionner correctement. On peut, donc, en déduire que la génération des données erronées est fortement corrélée à l'épuisement de la pile.

2.3.1.2 pannes au niveau du réseau

Le routage est une fonctionnalité essentielle des réseaux de capteurs, c'est une technique fondamentale dont nous avons besoin pour l'acheminent de données. Des défaillances au niveau de routage peuvent engendrer des pertes et le retard de messages. Généralement, dans un réseau de capteurs, les connexions entre les nœuds ne sont pas considérées comme très fiables, et le taux de délivrance des messages varie fortement selon les conditions ambiantes. Le protocole de routage est considéré comme la cause principale d'inefficacité du réseau, car il demande aux nœuds de choisir toujours le chemin le plus sûr. Il existe encore beaucoup d'autres causes possibles, par exemple la collision, comme les nœuds partagent le même médium pour la transmission de données, des collisions peuvent se produire lorsque plusieurs nœuds essayent d'émettre des signaux en même temps.

2.3.1.3 pannes au niveau de la station de base

Toujours dans une application de collecte de données, afin de faciliter et fluidifier la communication radio, l'ensemble d'informations collectées est agrégées au fur et à mesure durant leur transmission. Elles se propagent d'un nœud à l'autre pour aller jusqu'à leur destination finale, la station de base. Cette dernière peut aussi être victime des attaques extérieures et vulnérable aux différentes pannes. Si aucun mécanisme de protection n'est appliqué, le réseau risque d'être complètement isolé. La station de base est la seule interface entre les nœuds et les utilisateurs, lorsqu'elle tombe en panne, les utilisateurs ne pourront plus accéder aux services fournis par le réseau, et dans l'autre sens, les nœuds ne pourront plus envoyer des données ou recevoir des commandes des utilisateurs. Comme celles des nœuds déployés à distance, les pannes au niveau de la station de

base peuvent, aussi, être causées par des défaillances matérielles ou des bogues qui se cachent dans le programme exécuté.

2.3.2 Classification des solutions de tolérance aux pannes dans les RCSFs

Les solutions et les approches de tolérance aux pannes peuvent être vues de plusieurs angles différents. De ce fait, un ensemble de critères est défini pour les classifier. Des catégories de trois classifications distinctes peuvent être citée :

2.3.2.1 Classification selon la phase de traitement

Dans cette classification, on divise l'ensemble des algorithmes en deux principales catégories. Si le traitement est effectué avant la panne, on parle donc d'algorithmes préventifs sinon, les algorithmes sont dits curatifs.

- ***Algorithme préventif*** : ce type d'algorithmes implémente des techniques tolérantes aux pannes qui tentent de retarder ou éviter tout type d'erreur afin de garder le réseau fonctionnel le plus longtemps possible. La conservation d'énergie à titre d'exemple, permet de consommer moins d'énergie et évite donc une extinction prématurée de la batterie ce qui augmente la durée de vie des nœuds.
- ***Algorithme curatif*** : un algorithme curatif utilise une approche optimiste, où le mécanisme de tolérance aux pannes implémenté n'est exécuté qu'après la détection de pannes. Pour cela, plusieurs algorithmes de recouvrement après pannes sont proposés dans la littérature, par exemple : le recouvrement du chemin de routage, l'élection d'un nouvel agrégateur dans une architecture clustérisée, etc.

2.3.2.2 Classification architecturale

- ***Gestion de la batterie*** : cette catégorie est considérée comme une approche préventive, où les protocoles définissent une distribution uniforme pour la dissipation d'énergie entre les différents nœuds capteurs ; afin de mieux gérer la consommation d'énergie et augmentation la durée de vie de réseau. En outre, le mécanisme de mise en veille est une technique de gestion de batterie. En effet, les protocoles déterminent des délais de mise en veille des nœuds capteurs inactifs pour une meilleure conservation d'énergie [13].
- ***Gestion de flux*** : cette catégorie regroupe les techniques qui définissent des protocoles de gestion de transfert des données (routage, sélection de canal de transmission,etc.). On peut trouver des approches préventives ou curatives sur les différentes couches (réseau, liaison de données,etc.) telles que :

  – ***Routage multi-chemin*** : utilise un algorithme préventif pour déterminer plusieurs chemins depuis chaque capteur vers le nœud collecteur. Ceci garantit la présence de plus d'un chemin

fiable pour la transmission et offre une reprise rapide du transfert en cas de panne sur le premier chemin sélectionné (choisir un des chemins qui restent).

- ***Recouvrement de route*** : après détection de panne, une technique curative permet de créer un nouveau chemin plu fiable pour retransmettre les données.
- ***Allocation de canal*** : cette solution, implémentée au niveau MAC, effectue une allocation du canal de transmission d'une manière à diminuer les interférences entre les nœuds voisins et éviter les collisions durant le transfert.
- ***Mobilité*** : certains protocoles proposent comme solution tolérante aux pannes la sélection d'un ensemble de nœuds mobiles chargés de se déplacer entre les capteurs et collecter les données captées. Ceci réduira l'énergie consommée au niveau de chaque capteur en éliminant sa tâche de transmission. Un nœud mobile est généralement doté d'une batterie plus importante que celle d'un nœud capteur.

• ***Gestion des données*** : les protocoles classés dans cette catégorie offrent une meilleure gestion de données et de leur traitement. Deux principales sous-catégories sont déterminées :

- ***Agrégation*** : considérée comme approche préventive, l'opération d'agrégation effectue un traitement supplémentaire sur les données brutes captées depuis l'environnement. Un nœud agrégateur combine les données provenant de plusieurs nœuds en une information significative ; ce qui réduit considérablement la quantité de données transmises, demande moins d'énergie et augmente ainsi la durée de vie du réseau.
- ***Clustering*** : une des importantes approches pour traiter la structure d'un réseau de capteurs est le clustering. Il permet la formation d'un backbone virtuel qui améliore l'utilisation des ressources rares telles que la bande passante et l'énergie. Par ailleurs, le clustering aide à réaliser du multiplexage entre différents clusters. En outre, il améliore les performances des algorithmes de routage. Plusieurs protocoles utilisent cette approche préventive.

2.3.2.3 Classification selon le niveau d'implémentation

Une mauvaise gestion de transmission des données peut aboutir à plusieurs collisions aussi bien qu'à la congestion du réseau. D'où, il est impératif de concevoir des techniques de sélection de canal sur la couche liaison de données qui garantit la livraison des messages via des liens sans fil. Afin de garantir la fiabilité de transmission. La tolérance aux pannes est donc assurée par une phase de prévention de pannes avant transmission (en éjectant des délais d'attente, des mécanismes d'écoute de canal, etc.).

Le mécanisme de transmission des données qui ce fait grâce à des communications multi-sauts est efficace pour le problème de propagation et dégradation de signal, mais ce type de communication et confronté à des problèmes de choix des meilleurs chemins à emprunter pour garantir la livraison des données (qui soit de lien fiable et qui consomme le moins d'énergie). La tolérance aux pannes pour assurer une fiabilité de délivrance de paquets à la station de base est traitée au niveau de la couche réseau. Ces solutions sont classifiées en trois principales

catégories : agrégation, clustering et le routage qui va être le sujet du chapitre suivant.

2.4 Conclusion

Dans ce chapitre nous avons présenté des généralités sur la tolérance aux pannes dans les systèmes distribués et leur application dans les réseaux de capteurs. A cause de la limitation des ressources énergétique et la limitation de la puissance de calcul, les réseaux de capteurs sont généralement très vulnérables aux différentes pannes, et parfois le réseau risque d'être entièrement bloqué à cause d'un simple dysfonctionnement. C'est la raison principale pour laquelle actuallement les mécanismes de tolérance aux pannes sont considérés comme une partie indispensable pour les réseaux de capteurs et doivent être injectés à déférents niveaux (voir la section précédentes) du RCSF afin d'assurer une meilleure disponibilité et efficacité tout en préservant les spécificités de ce réseau.

3

Protocoles de routage tolérant aux pannes dans les RCSFs

3.1 Introduction

Dans les RCSF, les capteurs sont sujets à des pannes à cause de l'épuisement de leurs batteries, ou l'écrasement par des animaux, etc. Il résulte de l'occurrence des pannes une difficulté pour acheminer les données collectées à la station de base. Pour remédier à cette problématique des protocoles de routage tolérants aux pannes ont été proposés dans la littérature. La tolérance aux pannes pour assurer une fiabilité de délivrance de paquets à la station de base est traitée au niveau de la couche réseau.
Dans ce qui suit, nous présentons les fonctionnalités de certains protocoles de routage tolérants aux pannes et nous discutons leurs limites.

3.2 Solutions pour la tolérance aux pannes

3.2.1 Solutions de routage pour la tolérance aux pannes

Les protocoles de routage permettent de choisir les meilleurs chemins pour acheminer les données depuis les capteurs vers la station de base ou vers l'utilisateur final. Par ailleurs, ils permettent de sélectionner un chemin de remplacement en cas d'échec d'envoi sur la route initiale à cause d'une panne au niveau d'un ou plusieurs capteurs de cette route.

3.2.1.1 Protocole de routage dynamique tolérant aux pannes pour prolonger la durée de vie dans RCSF

PEGASIS [16],dans ce protocole, quand un nœud capteur est sur le point d'épuiser son énergie, il essaie de trouver un chemin alternatif pour établir une nouvelle connexion avec ses nœuds voisins pour gardé le réseau connecté. Ce chemin alternatif augmente la fiabilité de transmission de données entre les nœuds source et leurs voisins dans la direction de la station de base qui relaient les paquets envoyés par ces derniers. Ce protocole s'exécute en trois phases :

- ***Mise en œuvre et établissement de chemin*** : chaque nœud est caractérisé par un identifiant (N_j), le niveau (HC_j), nœud parent (P_j),un tableau (A_j) pour stocker les paquets de données jusqu'à ce qu'un accusé de réception soit reçu. La station de base est initialisée avec $HC = 0$ et $P = BS$, tandis que les nœuds ordinaires avec d'autres $HC_j = \infty$, $P_j = 1$. Une fois les nœuds sont déployés, la station de base diffuse un message d'avertissement ADVT(ADVerTisement) avec les paramètre (N_j, HC_j) pour découvrir les nœuds qui sont voisins à la station de base. Ces nœuds sont considérés comme des nœuds de niveau 1 puisqu'ils se trouvent à un saut de la station de base qui est considérée comme un nœud parent pour ces nœuds de niveau 1. Lorsqu'un nœud reçoit un message ADVT, son HC sera augmenté de un que de celui qui lui a envoyé le message ADVT et il est considéré comme un nœud de niveau N+1 si le nombre de sauts reçu est N. Ainsi, le message ADVT est utilisé pour hiérarchiser le réseau en des niveaux relativement à la station de base.
- ***Transmission de données*** : une fois que la hiérarchisation de niveaux est établie, la phase de transmission de données commence. De ce fait, lorsqu'un événement survient au niveau du nœud source. Ce dernier transmet le paquet de données relatif à l'événement au nœud parent et stocke une copie de ce paquet de données. Quand un parent reçoit le paquet de données émis, il envoie un accusé de réception (ACK) au nœud qui a transmis le paquet. De son côté le nœud source, une fois qu'il reçoit le paquet ACK, il supprime la copie du paquet de données correspondant. Cela continue jusqu'à ce que la station de base reçoive le paquet de données. Un numéro de séquence est attribué à chaque paquet de données transmis pour assurer la fiabilité et garantir sa livraison à la station de base. Si un paquet de données est perdu, il pourra être récupéré à partir du dernier nœud expéditeur.
- ***Rétablissement de chemin*** : si un nœud est sur le point d'épuiser son énergie, il envoie un message de notification à ses voisins fils en leur demandant de changer leurs nœuds parents pour maintenir la connectivité. Les nœuds fils qui reçoivent ce message, utilisent des paquets " Hello " pour découvrir les nouveaux parents dans leurs voisinages. Les nœuds fils modifient leurs paramètres N_j, HC_j en fonction de la réponse au message Hello. Si la réponse provient d'un nœud de niveau inférieur, les nœuds fils gardent leur H_j sinon c'est-à-dire le message provient d'un nœud voisin, les nœuds fils doivent incrémenter leurs niveaux de 1. La limitation de ce protocole est que le temps pris pour trouver un nouveau nœud voisin affecte la durée de livraison de données.

3.2.1.2 Protocole de routage tolérant aux pannes multi-niveaux (FMS)

Le protocole FMS [11] permet de maintenir la connectivité du réseau, même si un nœud est sur le point d'épuiser son énergie. Il permet aussi d'assurer la fiabilité et la rapidité de livraison des données à la station de base car il est conçu pour les applications orientées événement. Généralement, les capteurs sont déployés aléatoirement et en grand nombre. De ce fait, il y aura une redondance dans la livraison de données ce qui a une conséquence sur la durée de vie du réseau de capteurs. Pour remédier à cette limite, FMS permet un ordonnancement d'activité des capteurs en passant un certain nombre de capteurs en mode " veille " sans affecter la fiabilité de livraison de données. Ceci est dans le but d'économiser l'énergie. Dans FMS, on suppose que chaque nœud possède un identifiant unique, dénoté (Nr), et la communication entre les nœuds voisins est bidirectionnelle. En outre, on suppose que les nœuds sont contraints en termes de puissance de traitement, de stockage et de l'énergie, tandis que la station de base est considérée comme un nœud qui a plus de ressources pour effectuer des tâches ou de communiquer avec les autres nœuds. Le protocole FMS effectue deux opérations de base :

- ***Détermination des niveaux des nœuds et établissement de chemin*** : cette phase est analogue à celle du protocole cité précédemment.
- ***Ordonnancement d'activité des capteurs et transmission de données*** : l'ordonnancement d'activité des capteurs consiste à faire passer un certain nombre de nœuds périodiquement en mode veille. Au cours de cette période, les nœuds actifs transmettent les paquets de données. Avant qu'un nœud passe en mode veille, il devra informer ses nœuds fils afin qu'ils choisissent un autre nœud parent pour relayer les données. En outre, quand un nœud est en mode veille, il passera en mode actif que si son énergie est supérieure à une certaine valeur seuil. Le choix des nœuds actifs se fait aléatoirement et d'une manière périodique pour que le nœud n'épuise pas son énergie rapidement. Quand un nœud est en mode actif, il participe à l'opération de transmission de données à la station de base. De ce fait, la connectivité est toujours maintenue même si un nœud est mis en mode veille ou il est sur le point de perdre son énergie. Ainsi, FMS est considéré comme un protocole fiable et tolérant aux pannes.
  FMS présente les mêmes limitations que le protocole cité précédemment. Il est performant dans un environnement optimal mais ses performances se dégradent dans un environnement réel.

3.2.1.3 Protocole de routage adaptatif tolérant aux pannes (RERP)

Dans RERP [17], il est supposé que chaque nœud a au moins deux voisins dans la direction vers la station de base. Par conséquent, il aura au moins deux chemins alternatifs pour acheminer les données vers la station de base. Ainsi, la capacité d'un nœud tolérant aux pannes dépend du nombre des nœuds voisins actifs c'est-à-dire si un nœud a N voisins, il peut tolérer N-1 nœuds en panne. Le protocole RERP comporte deux tâches :

- ***Mise en place de RERP*** : cette tâche s'exécute en cinq phases :
  - ***Phase de publicité*** : dans cette phase, la station de base diffuse un paquet de publicité

à ses nœuds voisins pour indiquer qu'elle peut recevoir des paquets de données. Les nœuds qui reçoivent le paquet de publicité établissent une table de routage pour indiquer le chemin vers la station de base.

- ***Phase d'initialisation*** : dans cette phase, les nœuds qui n'ont pas de chemin direct vers la station de base diffusent une requête de découverte de routes (RREQ :Route REQuest) vers la station de base. Quand un concentrateur(le nœud qu'à une liaison direct vers la station de base) reçoit le paquet (RREQ), il diffuse une réponse (RREP : Route REPly). De même si ce nœud a déjà reçu la requête (RREQ), il diffuse un paquet (RREP) s'il existe un chemin entre lui et le concentrateur, sinon le paquet (RREQ) sera ignoré.
- ***Route de sélection*** : une table de routage est utilisée pour construire et entretenir les routes. Le choix de l'itinéraire des nœuds relais est basé sur l'énergie restante des nœuds.
- ***Phase de transfert de données*** : les nœuds capteurs génèrent des paquets de données à chaque fois qu'ils détectent toute nouvelle information. Cette information est transmise à la station de base en un mode multi-sauts.
- ***Table de sauvegarde*** : en plus du chemin principal, un chemin alternatif est prévu pour tous les nœuds du réseau. Chaque fois qu'un nœud reçoit un paquet RREP, s'il ne dispose pas de chemin direct vers la station de base, il stocke le chemin dans la table de routage, et il stocke les paquets (RREP) dans une table de sauvegarde. La table de routage de secours dispose de deux champs, l'identifiant du nœud ID et son énergie.

- ***Rapport d'erreurs*** : nous distinguons les messages d'erreurs suivants :
  - ***Echec des liens*** : le message d'échec de liens est généré dans deux cas. Le premier se produit quand un RTS (Request To Send) est envoyé mais aucun CTS (Clear To Send) correspondant n'est reçu et le nombre maximal de tentatives est dépassé. Le second se passe quand un paquet de données a été transmis, mais il n'a jamais reçu un ACK et le nombre maximal de tentatives est dépassé.
  - ***Message de batterie critique*** : ce message est généré lorsque le niveau de la batterie d'un nœud est inférieur à une valeur seuil dite critique. Ce message est envoyé au nœud source qui a envoyé les données et également aux voisins de ce nœud. Quand les autres nœuds reçoivent ce message ils suppriment l'identifiant du nœud défaillant de leurs tables de routage ou de leur table de voisins.
  - ***Message de destination inaccessible*** : ce message est généré lorsque le paquet de données est mis au rebut sans être transmis au nœud de destination en raison de l'indisponibilité du chemin vers la station de base.
  - ***Sélection du chemin de secours*** : chaque nœud possède une table de routage de secours dans laquelle il stocke un chemin de secours vers la destination. Quand un nœud échoue dans la transmission de paquet de données, alors son voisin consulte la table de secours pour trouver le chemin alternatif afin qu'il puisse transmettre le paquet de données. Dans RERP la communication entre les nœuds est réalisée par des messages Requête/Réponse. Ce type de messages est utilisé pour vérifier si le voisin est accessible et pour calculer le

temps de parcours.
RERP présente certaines limitations telles que la consommation d'énergie qui est assez grande lors de la diffusion des rapports d'erreurs.

3.2.1.4 Protocole de routage temps réel tolérant aux pannes (DMRF)

DMRF [18] fonctionne en deux modes de transmission de données : saut à saut et "Jumping". Chaque nœud utilise le temps restant pour transmettre un paquet à la station de base et l'ensemble des nœuds de transfert FCS (Set candidat Forwarding) pour choisir dynamiquement le prochain saut. Quand un nœud subit une défaillance, alors la congestion du réseau ou une région vide se produit. Le mode de transmission sera passé en mode "Jumping", ce qui peut réduire le délai de transmission, et assure la fiabilité de la livraison des paquets de données envoyés à la station de base dans un délai spécifié. Dans DMRF, le processus de transmission est divisé en cinq étapes :

- ***Phase d'initialisation*** : dans cette phase, DMRF initialise la liste de voisinage des nœuds, les informatrion de l'état du réseau (information sur la congestion d'un nœud, les zones vides, ... etc), la liste des candidats FCS, la table des probabilités de transition, et la voie de transmission initiale.
- ***Phase de transmission des données*** : dans cette phase, DMRF détecte la défaillance d'un nœud, la congestion du réseau ou une région vide. Le temps restant pour acheminer un paquet de données jusqu'à la station de base sera contrôlé. A partir de ce temps, le paquet sera transmis en mode "Jumping" ou non. Si aucune des conditions ci-dessus ne s'est produite, DMRF sélectionne dynamiquement un membre du FCS comme nœud relais. En outre, une fois les nœuds défaillants sont détectés, ou le temps restant est inférieur à un certain seuil, le mode de transmission "Jumping" sera utilisé.
- ***Phase de transmission "Jumping"*** : au cours de cette phase, chaque nœud ajuste dynamiquement le contenu de FCS et calcule la probabilité pour transiter par chacun de ces nœuds. Dans ce mode, le paquet de données peut utiliser un saut d'une grande portée pour éviter les nœuds défaillants. Cependant, il ne peut pas garantir le succès de la transmission. Donc, la phase d'ajustement des probabilités de transition est effectuée après chaque transmission "Jumping".
- ***La phase d'ajustement des probabilités*** : dans cette phase, DMRF ajuste la probabilité de saut en fonction du résultat de la transmission "Jumping" (succès ou l'échec) et renvoie l'information à son nœud en amont. Lorsque le paquet de données arrive au nœud récepteur, on considère que la transmission est terminée. DMRF présente certaines limitations en particulier dans le mode "Jumping" qui ne garantit pas la fiabilité de livraison de données et qui consomme plus d'énergie quand il utilise une grande portée.

3.2.1.5 AODV tolérant aux pannes (ENFAT-AODV)

ENFAT-AODV [19] est un protocole de routage qui tolère les pannes, l'auto-démarrage et le routage multi-sauts entre les nœuds. ENFAT-AODV établit les plus courts chemins entre les nœuds

en nombre de sauts. En outre, il permet aussi aux nœuds d'établir un chemin de secours quand le chemin principal subit des ruptures. Toutefois, dans ENFAT-AODV, il y a des champs additionnels comparativement à la version originale du "AODV". Certains champs sont ajoutés dans les paquets de contrôle tels que "BACKUP" (dans RREQ et RREP), "UPDATE" dans RREQ et "Distance" dans RREQ. ENFAT-AODV réduit également la complexité de mise en œuvre par suppression des paquets de contrôle inutiles dans le réseau et il s'exécute comme suit :

- ***Découverte du chemin principal*** : quand un chemin principal de livraison des données vers la station de base est nécessaire, le nœud source diffuse un message de découverte de chemin principal vers la station de base en utilisant des messages RREQ. Ainsi, chaque nœud intermédiaire recevant le message RREQ établit un chemin inverse vers le nœud source. Si le nœud reçoit le message RREQ pour la première fois et si ce dernier ne connaît pas la route principale menant à la destination, il transmettra le message RREQ à ses voisins. Si le nœud de réception est la destination ou bien s'il connaît la route principale menant à la destination, il va générer un itinéraire principal (RREP principal). Ensuite, ce RREP principal est envoyé en un saut à la source. Lorsque la source reçoit le message RREP, cette dernière enregistre la route principale menant à la destination dans sa table de routage.
- ***Construction du chemin de secours*** : au cours de la phase de découverte de chemin principal, les nœuds d'un chemin principal qui reçoivent un RREP principal créent un chemin de secours vers la station de base en diffusant un paquet de secours RREQ. Après la diffusion du RREQ de secours, le nœud attendra un paquet RREP de secours de la part de la destination ou d'un nœud intermédiaire qui peut satisfaire les conditions suivantes :
  - Il dispose d'une entrée de secours dans le chemin principal vers la station de base.
  - Il n'est pas un nœud sur le chemin principal.
  - Le nombre de sauts du chemin de secours à partir du nœud intermédiaire à la destination est inférieur aux autres nœuds.
- ***Entretien de la route*** : pendant la période de livraison des paquets de données quand le chemin principal n'est pas valide, le nœud utilise immédiatement sa route de secours pour relayer les prochains paquets de données. Par la suite, le nœud sur le nouveau chemin principal, qui utilise une route de secours, dirige un processus "Découverte de route de secours" visant à trouver un autre chemin. Par conséquent, il augmente la fiabilité et la disponibilité par rapport à la première version de "AODV". La limitation de ce protocole concerne essentiellement la consommation de l'énergie à cause de l'inondation des messages de contrôle.

3.2.1.6 Diffusion dirigée tolérant aux pannes (FaT2D)

FaT2D [20] est un protocole de routage tolérant aux pannes basée sur la diffusion.Cette tolérance aux pannes est assurée grâce à la construction de plusieurs chemins entre les nœuds et l'exploration périodique des routes pour la découverte d'éventuelles anomalies au niveau de ces routes. FaT2D définit une nouvelle technique qui permet de détecter rapidement une panne et la recouvrir quand il y a une collision entre les nœuds et des changements de topologie. Il s'exécute selon la démarche

qui suit :

- ***Détection de panne*** : FaT2D introduit un nouveau délai d'attente de détection de pannes, noté TFD. Il est défini afin de réduire le temps de recouvrement de la panne et par conséquent le remplacement des nœuds défaillants.
- ***Si TFD s'épuise*** : FaT2D transmet immédiatement un nouveau message appelé "Explore-Request" pour notifier l'événement de détection de la panne et demande une nouvelle exploration pour trouver un autre chemin fiable qui remplace le chemin défaillant. Par conséquent, tout nœud appartenant au chemin défaillant supprime le gradient correspondant pour éliminer la panne.
- ***Recouvrement du chemin*** : quand TFD est épuisé, il déclare la défaillance d'un nœud. De ce fait, FaT2D lance un processus pour réparer le chemin défectueux en envoyant un message de demande d'exploration appelé "ExploreRequest". Ce message contient les informations sur la route défectueuse et il est acheminé pour atteindre le nœud cible sans utiliser les transmissions bouclées ou rechercher les nœuds non adéquats. Quand le nœud cible reçoit le message "ExploreRequest", il ne le transmet pas, puis il lance une exploration par inondation comme dans "Direct Diffusion". Cela génère une phase d'exploration afin de trouver un nouveau chemin fiable.
- ***Elimination des pannes*** : pour chaque nœud intermédiaire recevant le message "Explore-Request", FaT2D vérifie si ce nœud appartient au chemin défectueux. Si c'est le cas, il aura un effet négatif pour renforcer son gradient. Ce dernier sera réélu par une exploration lancée par le nœud source du chemin correspondant. Ainsi, chaque nœud demande à ces voisins en amont de supprimer le chemin brisé et arrêter l'envoi de données sur ce chemin.

3.2.1.7 Algorithme PEQ

PEQ [21] combine la conservation d'énergie avec le routage multi-chemins en sélectionnant parmi tous les chemins disponibles ceux qui consomment moins d'énergie et le considère comme étant chemin principal et les autres des chemins secondaires. En plus de ce mécanisme préventif qui permet un routage fiable avant l'occurrence des pannes, un mécanisme de recouvrement de pannes est implémenté. Ce dernier remplace le chemin défaillant par un autre chemin qui a des liens fiables et consomme moins d'énergie. PEQ introduit le paradigme Publish/Subscribe comme montre la figure 3.1 pour l'interaction entre la station de base et les capteurs simples. En effet, les capteurs envoient des notifications d'événements à la station de base, qui va souscrire son intérêt pour certaines de ces informations. Les capteurs concernés publient par la suite l'information désirée.

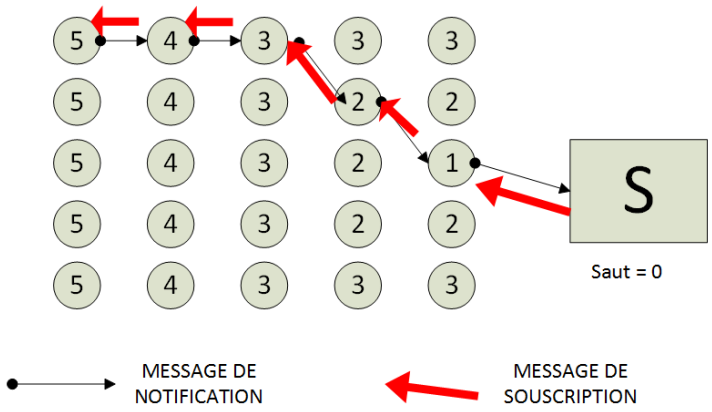

FIGURE 3.1 – Mécanisme Publication/Souscription.

PEQ s'exécute comme suit :

- ***Construction de l'arbre de routage*** : cet arbre permet de définir les différents chemins multi-sauts possibles pour acheminer les données de chaque nœud à la station de base. Cette dernière commence le processus en initialisant la variable " saut " à 0. Par la suite, chaque capteur prend la valeur du saut actuelle, l'incrémente puis l'envoie à tous ses voisins. Ainsi, la valeur au niveau de chaque capteur désigne le nombre nécessaire de sauts pour communiquer avec la station de base. A la fin de cette phase, seulement les meilleurs chemins sont enregistrés.
- ***Transmission de paquets de notification*** : chaque capteur envoie, selon sa table de routage et l'événement capté, une notification de l'information qu'il a à sa disposition. Pour cela, il utilise le chemin le plus court et le moins coûteux en terme d'énergie.
- ***Propagation des paquets de souscription*** : dans cette étape, après une souscription, par la station de base, des données à transmettre, chaque capteur achemine cette dernière jusqu'au capteur concerné.
- ***Mécanisme de recouvrement de route*** : le recouvrement est effectué après détection de pannes (figure 3.2). Un capteur envoie son paquet puis attend un acquittement ACK. S'il le reçoit, le message a été bien transmis ; sinon une panne est détectée au niveau du chemin de routage. On effectue donc une recherche "SEARCH" pour la sélection d'un autre capteur destination tout en minimisant le coût du nouveau chemin. Si aucun capteur n'est trouvé, le capteur devient isolé et doit donc augmenter son rayon de transmission radio pour atteindre les capteurs voisins lointains.

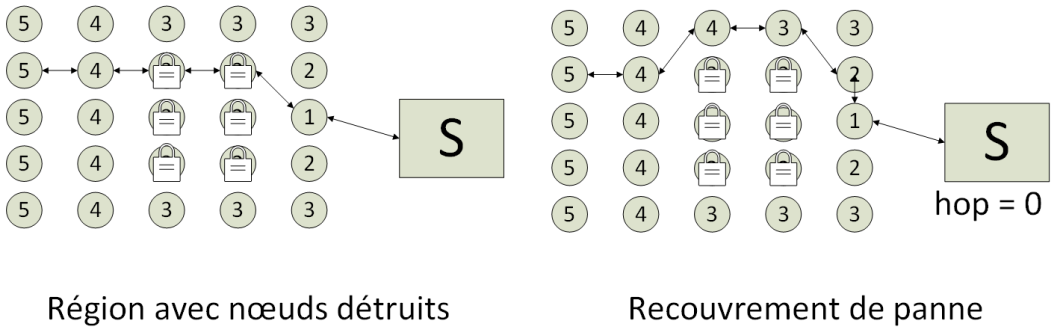

FIGURE 3.2 – Recouvrement de routes dans PEQ.

3.2.1.8 Protocole EAR

EAR [22] propose une solution hybride pour la tolerance aux pannes. Il permet une meilleure conservation d'énergie et définit plusieurs chemins de routage afin de garantir une fiabilité de livraison de données. En outre, un mécanisme de recouvrement de pannes est implémenté. Le protocole EAR supporte des réseaux de capteurs à plusieur stations de base. Chaque capteur génère un paquet RPT (RePorT) contenant des informations pour les préférences de l'utilisateur. Les paquets RPT peuvent être envoyés vers n'importe quel station de base. Cependant, pour chaque capteur intermédiaire le protocole de routage choisit le meilleur chemin qui réduit la consommation d'énergie et la latence. EAR s'exécute selon les étapes suivantes :

- ***Phase d'initialisation*** : cette phase permet la construction de l'arbre de routage contenant tous les chemins possibles pour la dissémination des données. Chaque station de base diffuse un message d'avertissement ADV (ADVertisement) demandant des paquets RPT. Seuls les capteurs voisins du station de base qui reçoivent le message ADV, enregistrent le chemin dans leur table de routage ; sans qu'ils propagent le message ADV vers les autres capteurs, comme le montrent les étapes a et b de la figure 3.3. Les autres capteurs envoient une demande RREQ (Route Request) cherchant un chemin vers la station de base (étape c). Si un capteur ayant déjà une route stockée dans sa table, reçoit RREQ, il envoie un paquet RREP (Route Reply) à son capteur voisin concerné par la demande (étapes d, e). Le processus d'initialisation se termine quand chaque capteur reçoit une réponse RREP suite à sa requête RREQ ; puis enregistre le chemin dans sa table de routage.
- ***Phase de gestion de route*** : les micro-capteurs, avec leur mémoire de taille réduite, ne peuvent pas garder tous les chemins possibles dans leurs tables de routage. Pour cela, et afin d'assurer une bonne tolérance aux pannes, on devrait garder que les meilleurs chemins. Le protocole EAR définit donc deux métriques pour la sélection des meilleurs chemins à mémoriser. La première métrique est le nombre de sauts dans une route. Ceci permet de choisir le chemin le plus court. Cependant, la qualité des liens n'est pas prise en considération ; dans ce cas, le plus court chemin n'assure pas forcément la fiabilité de transmission. En effet,

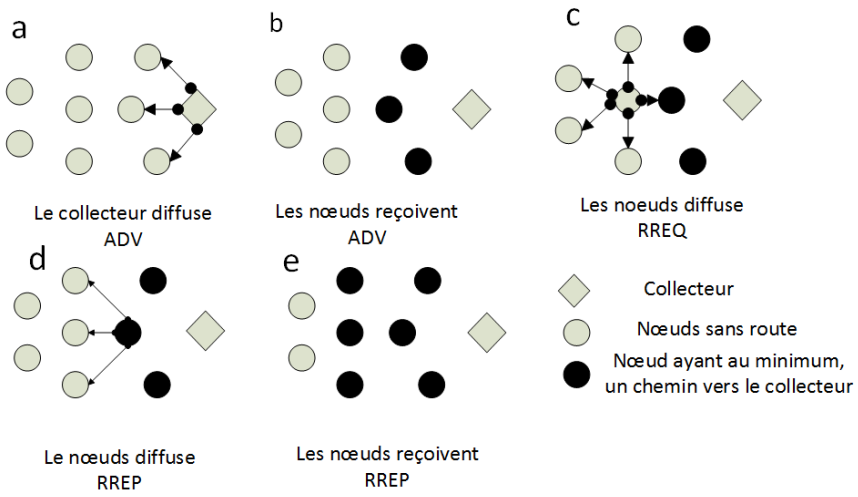

FIGURE 3.3 – Fonctionnement du protocole EAR.

si un chemin échoue à transmettre N paquets consécutifs, il sera mis dans une "liste noire" l'écartant ainsi d'une future utilisation. La deuxième métrique, appelée Score de route, est définie comme suit :

$$RS = PE \times WE + PT \times WT \tag{3.1}$$

- PE : niveau de l'énergie du capteur du prochain saut .
- WE : poids assigné à PE dans l'intervalle $[0, 1]$.
- PT : taux de succès dans la transmission .
- WT : poids assigné à PT dans $[0, 1]$ tel que $(WT + WE = 1)$.

- ***Phase de dissémination de données*** : après les deux premières étapes, chaque capteur aura au moins un chemin vers la station de base. Les capteurs commencent donc à générer des paquets RPT, et le routage des données utilise la métrique "score de route" pour définir le meilleur chemin à emprunter. En cas où ce dernier présente une panne au niveau d'un ou plusieurs de ses capteurs, un mécanisme de recouvrement de route est exécuté, afin d'élire un second chemin fiable pour transmettre les données depuis le capteur vers la station de base. Par ailleurs, au moment de sa durée d'inactivité, chaque capteur est mis en veille afin d'épargner davantage son énergie et augmenter ainsi la durée de vie de tout le réseau.

3.2.1.9 FATE-CSQ

Le protocole FATE-CSQ [23] offre une solution tolérante aux pannes pour l'évaluation des requêtes continues de type sélection dans les réseaux de capteurs sans fils. Il vise à prévenir les pannes par une réorganisation occasionnelle de la topologie du réseau, à détecter les pannes par un des mécanismes d'acquittement intelligents avec " feedback " et à les traiter par des mécanismes de retransmission des réponses perdues. FATE-CSQ garantit à l'utilisateur un niveau de qualité spécifié par la requête lancée. Si l'utilisateur a lancé une requête demandant " s'il y a une fuite de gaz toxique " dans un réseau de 1000 capteurs et s'il a reçu 100 réponses positives à sa requête, l'utilisateur ne

connaît pas ce qui se passe pour les 900 capteurs restants. Est-ce qu'ils ont une réponse négative à la requête (pas de fuite) ? Ou leurs réponses sont perdues (mais il y a une fuite) ? Si l'utilisateur peut savoir que 400 capteurs envoient des réponses négatives, alors il obtiendra une meilleure précision sur la réalité parce qu'il a reçu 100 réponses positives parmi 600 réponses potentielles et non pas parmi 1000. Le niveau de qualité est mesuré par le rapport suivant :

$$Q = \frac{|A|}{|S| - |N|} \tag{3.2}$$

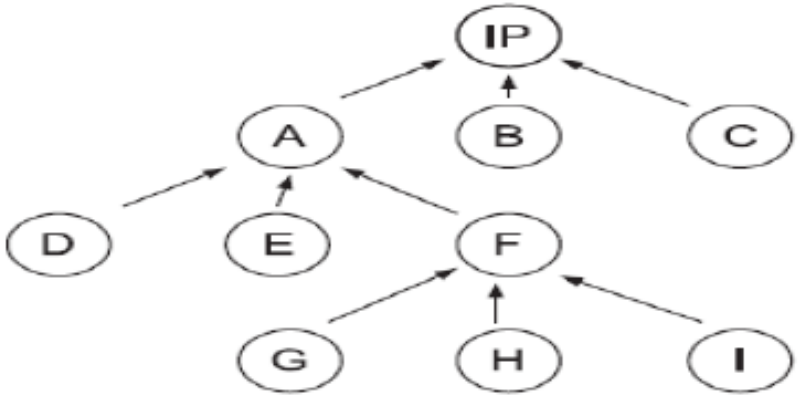

FIGURE 3.4 – Arbre de routage.

Ou |A| : nombre des réponses positives, |S| : nombre total des capteurs, N : nombre des réponses négatives.

Exemple : considérons l'exemple suivant : |S|=1000 capteurs et |A|=100 et N=500 alors Q=20 (pour cent). L'utilisateur a reçu seulement 20 pour cent des réponses potentielles à cause des défaillances. La connaissance du rapport de garantie aide l'utilisateur à prendre des décisions plus précises. L'algorithme doit compter alors le nombre des réponses négatives pour satisfaire la requête de l'utilisateur.

L'algorithme commence par la diffusion de la requête à travers le réseau permettant de construire un arbre de routage. Chaque nœud maintient l'ID de son parent et les ID de ses enfants. Puis, chaque nœud envoie à son parent sa réponse s'il est positif, sinon il envoie un message de négation. Si le nœud a un sous arbre, il renvoie les messages positifs et le nombre de messages de négation de ses nœuds fils. Le processus d'évaluation se répète jusqu'à la fin de la période ou la satisfaction de l'exigence de qualité. A la fin de chaque tour, un nœud parent diffuse un vecteur binaire d'acquittements à son sous arbre. Chaque nœud fils vérifie le vecteur et retransmet la réponse en cas de perte (bit 0). C'est ce qu'on appelle " Direct Feedback " entre le nœud parent et les nœuds fils directs. De même, pour la perte des réponses provenant des descendants des nœuds fils, par

exemple le cas où la réponse du nœud H (Figure 3.4) se perd sur le lien F-A, l'algorithme introduit un autre mécanisme d'acquittement appelé " Forwarding Feedback ". Le but de ce mécanisme est de retransmettre les réponses perdues provenant des nœuds plus lointains que le nœud fils direct. La retransmission se fait du nœud F et non pas du nœud d'origine H, ce qui réduit le temps de latence et la surcharge de communication. La topologie du réseau peut subir, pendant la phase d'évaluation de la requête, des modifications dues à la coupure des liens et/ou défaillances des nœuds dans le réseau. C'est pourquoi un nœud parent peut décider de déclencher la phase de restructuration de la topologie avant de commencer la période suivante. Il diffuse un message " Restructure " permettant de rétablir la relation entre un nœud parent et les nœuds descendants. Le protocole FATE-CSQ rationalise la consommation d'énergie de plusieurs façons :

- Il évite la collision lors de la transmission FATE-CSQ utilise le mécanisme TDMA (couche MAC) pour allouer le temps de transmission pour chaque nœud fils, ce qui évite la communication simultanée des nœuds fils.
- Il profite de la nature de communication dans les réseaux sans fils. En effet, lorsqu'un capteur envoie un message, celui-ci sera écouté par l'ensemble des capteurs qui sont dans la même portée de la communication. Ainsi, le feedback est envoyé dans un seul message pour tous les fils.
- La réponse d'un fils est transmise à son parent au plus une seule fois ;
- Il réduit au maximum le temps d'écoute passive en transmettant le feedback directement à la fin de la transmission des réponses.
- Le mécanisme de fiabilité (Feedback MTV) est du type saut par saut et non de bout en bout, ce qui permet de retransmettre le message à partir du lien où il est perdu et non pas à partir de l'émetteur d'origine.
- Le capteur entre dans l'état de sommeil lorsqu'il a accompli son rôle de transmission (envoie sa réponse et
  les réponses de ses fils à son parent) avant la fin de la période de transmission. Par contre, la phase de reconstruction de l'arbre introduit une surcharge considérable due aux messages de diffusion transmis.

3.2.2 Solutions basées sur le clustering pour la tolérance aux pannes

3.2.2.1 Protocole CPEQ

En plus des mécanismes de tolérance aux pannes implémentés dans PEQ, la variante CPEQ (Cluster-based PEQ) [24] utilise l'approche de clustering pour offrir une meilleure gestion de routage. En effet, les capteurs ayant le plus d'énergie résiduelle sont sélectionnés comme des nœuds agrégateurs (cluster heads). Dans un cluster, les nœuds membres à ce dernier envoient leurs données au cluster head qui effectue d'éventuel traitement sur les données avant de les acheminer vers la station de base. Chaque capteur du réseau peut devenir cluster head pendant une certaine période de temps selon son niveau de batterie. Le but principal de CPEQ est de distribuer d'une manière

uniforme la dissipation d'énergie entre les capteurs, et de réduire la latence et le trafic de données dans le réseau. Le protocole CPEQ s'exécute en cinq étapes :

- ***Configuration initiale*** : cette phase est exécutée de la même manière que dans l'algorithme PEQ ; où chaque capteur commence par un mécanisme de diffusion pour connaître par la suite le nombre de sauts nécessaires pour atteindre la station de base le plus proche. En outre, CPEQ introduit un champ additionnel contenant le pourcentage des capteurs qui deviendront agrégateurs.
- ***Sélection d'agrégateur*** : c'est la phase d'élection des cluster heads. Après la configuration initiale, chaque capteur peut devenir agrégateur avec un pourcentage donné. En effet, chaque capteur génère un nombre aléatoire entre 0 et 1. Si ce nombre est inférieur à une probabilité p (probabilité pour devenir agrégateur), le capteur demande à tous ses voisins directs leur niveau de batterie en envoyant un paquet REQ_EN (RequestEnergy). Chaque voisin répond par un message REP_EN (ReplyEnergy) contenant son ID et la quantité d'énergie. Le capteur choisit le voisin ayant le maximum d'énergie et diffuse un SET_AGR (Set Aggregator) pour informer tous les capteurs du nouvel agrégateur. Les trois étapes de cette phase sont illustrées dans la figure 3.5.

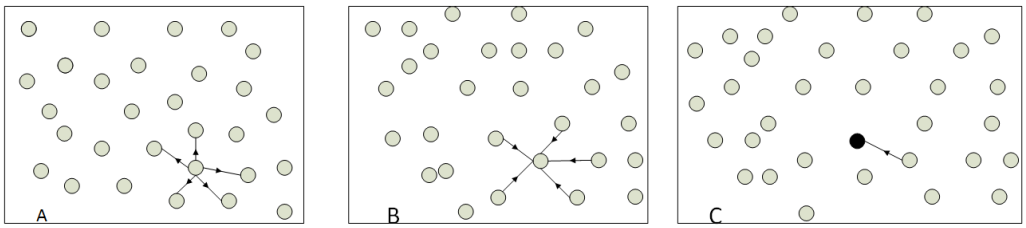

FIGURE 3.5 – Processus d'élection des cluster heads.

- ***Configuration de clusters*** : au cours de cette phase on assiste à la formation des clusters. Le nouveau capteur agrégateur sélectionné doit aviser ses voisins de son rôle d'agrégateur. De ce fait, chaque agrégateur construit son cluster. La configuration des clusters est réalisée à l'aide des messages AGR_NTF (Aggregator Notification) avec un champ TTL(time to live) pour limiter la propagation du paquet sur les capteurs se trouvant à une distance inférieure ou égale au TTL. Chaque fois qu'un capteur reçoit ce message, il enregistre l'ID du capteur émetteur dans sa table de routage pour déterminer le chemin vers l'agrégateur. Si un capteur reçoit plusieurs messages AGR_NTF ; il choisit l'agrégateur avec le moindre nombre de sauts. La figure 3.6 illustre la configuration de clusters avec un TTL=2.
- ***Transmission de données à l'agrégateur*** : chaque capteur utilise sa table de routage pour envoyer la donnée vers son agrégateur. Dans CPEQ, l'agrégateur peut être considéré

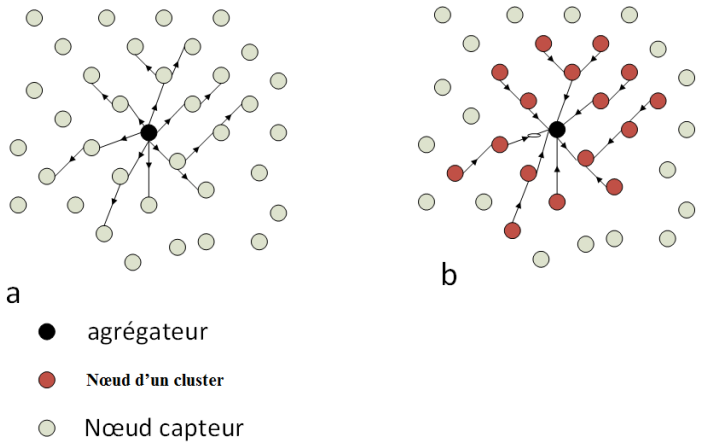

FIGURE 3.6 – Formation des clusters.

comme un nœud puits. Le mécanisme de recouvrement de chemin est aussi hérité du protocole PEQ.

- ***Transmission de données a la station de base*** : après réception des données depuis les capteurs de son cluster, l'agrégateur doit acheminer ces données vers la station de base. CPEQ utilise une communication multi-sauts entre l'agrégateur et la station de base comme montre la figure 3.7.

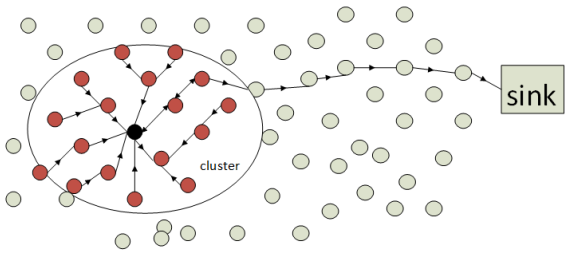

FIGURE 3.7 – Transmission des données vers la station de base.

3.2.2.2 KAT-Mobility

Dans KAT-mobiliy [25], en plus du concept de clustering, le concept de mobilité est implémenté au niveau de la station de base. Ces deux mécanismes, définissent une technique préventive hybride tolérante aux pannes qui offre une meilleure gestion d'énergie et augmente donc la durée de vie du réseau. Après réorganisation du réseau en clusters, la méthode proposée pilote la sation de base mobile pour se déplacer à travers les centres des clusters en prenant le chemin optimal. La station de base mobile récupère donc les données depuis les capteurs des clusters visités. Le principe de KAT-mobility se résume en deux procédures : clustering et optimisation du routage. La figure 3.8 illustre le principe de fonctionnement de KAT-mobility.

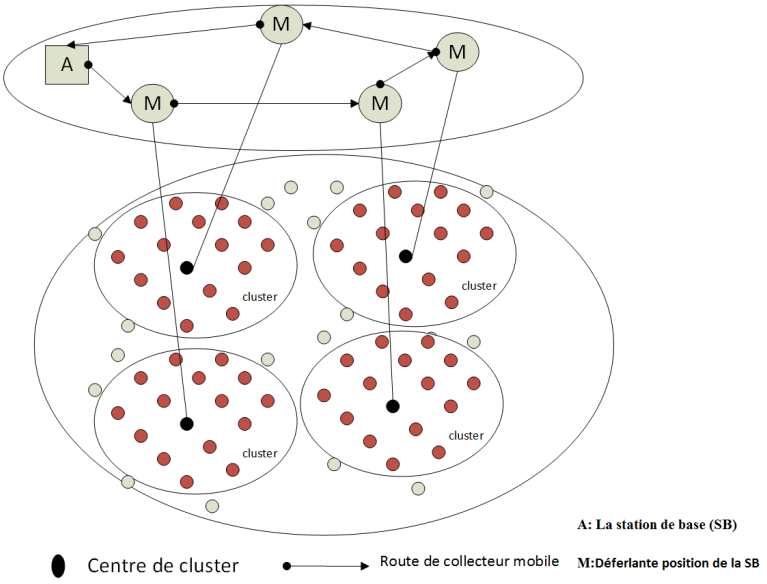

FIGURE 3.8 – Fonctionnement de KAT-Mobility.

- ***Algorithme de clustering*** : cette procédure divise l'ensemble des N capteurs en k clusters $C_1, C_2, ..., C_k$. Le coût du cluster est évalué par l'erreur approximative entre la station de base et les nœuds capteurs. Soit $d(x, y_i)$ cette erreur, où x est un capteur et y_i est la station de base $(i = 1, 2, ..., k)$. $d(x, y_i)$ est définie par la distance euclidienne entre le capteur et la station de base. Le but est donc, d'affecter chaque capteur au cluster le plus proche.
- ***Optimisation du routage*** : cette phase consiste à criée un chemin optimal pour le nœud mobile comme dans le cas du voyageur de commerce (TSP). Ainsi ; la station de base représente le voyageur, et les centres des clusters définissent les villes. L'optimisation de la route de la station de base mobile consiste à visiter tous les centres des clusters une et une seule fois.

  Les résultats de simulation ont montré que KAT-mobility peut fournir une meilleure conservation d'énergie aussi bien qu'une bonne tolérance aux pannes en cas où certains capteurs cessent à fonctionner.

3.2.2.3 H-SEND

H-SEND [22] est une approche semi-automatique pour la détection des pannes. Dans cette approche, le programmeur spécifie l'exactitude des propriétés du protocole appelées " invariants ". Ceux-ci sont associés aux conditions (les variables observées) que ce soit au niveau du nœud ou au niveau du réseau. Le compilateur insère automatiquement les codes de vérification pour assurer que les variables observées satisfont les invariants. La panne est détectée à la violation de ces invariants. Dans ce cas, le programmeur procède au transfert des nouveaux programmes ou des patches afin de traiter la panne. La détection peut être locale ou globale. Elle est locale lorsqu'elle dépend des invariants locaux au nœud. Elle est globale si elle dépend des variables et conditions associées à plusieurs nœuds du réseau. La vérification des variables globales engendre une surcharge d'énergie pour communiquer les informations de débogage au cluster head. Afin d'économiser la consommation d'énergie, l'approche H-SEND adopte le mécanisme de marquage où les variables requises au débogage sont superposées avec les données du réseau. De plus, la simulation a montré que la détection hiérarchique d'erreurs (nœud-CH-puits) a réduit la surcharge de (7 pour cent) par rapport à l'approche centralisée qui consiste à envoyer les informations de débogage au puits.

3.3 Comparaison des approches

Tenant compte des protocoles traités dans ce chapitre ainsi que de l'importance d'étudier l'impact de certains critères sur la validation d'une solution globale de détection des fautes adaptée aux RCSF, nous avons comparé les approches présentées selon sixe critères,(Tableau 3.9). Etablissement des routes, mécanisme de tolérance aux pannes, mécanisme de détection des pannes, type d'application, les critères qui augmente la consommation d'énergies et type de hiérarchie.

PROTOCOLES	Etablissement des routes	Mécanisme de tolérance aux pannes	Mécanisme de détection des pannes	Type d'application	Critère augmente la consommation d'énergies	Type de hiérarchie
PEGASSIS	Proactif	Préventif	Par les ACK	Périodique	Les ACK et recouvrement de chemin	Chaine
FMS	Proactif	Préventif	Par les ACK	Even-driven	Les ACK	Chaine
RERP	Proactif	Préventif	Par les ACK	Even-driven	La diffusion des raports d'erreur	Chaine
DMRF	Proactif	Préventif		Even-driven ou periodique	Le mode Jamping	Chaine
ENFAT-AODV	Proactif	Préventif	Par les ACK	Even-driven	L'inondation avec messages de contrôle	Chaine
FaT2D	Proactif	Curatif	Exploration périodique		L'inondation avec messages de contrôle	Chaine
PEQ	Proactif	Préventif	Par les ACK	Hybride	Les ACK et recouvrement de chemin	Chaine
EAR	Proactif	Préventif		A la demande		Chaine
H-SEND		Curatif	Comparisons de données		Les raports d'erreur	Cluster
CPEQ	Hybride	Préventif	Par les ACK	Hybride	Les ACK et recouvrement de chemin	Cluster
KAT-Mobility	Proactif	Préventif		Hybride		Cluster
FATE-CSQ		Curatif	Exploration périodique			Chaine

FIGURE 3.9 – Comparison entre les protocole.

3.4 Conclusion

Dans ce chapitre, on a présenté une classification sur la tolérance aux pannes, et les différentes approches proposées dans la littérature. Puis, on a fait une comparaison entre les différons protocoles.

4

Proposition d'un protocole de routage tolérant aux pannes EPEQ

4.1 Introduction

Dans les RCSFs, la défaillance se manifeste à plusieurs niveaux (nœud, réseau, données, et application) et peut parfois se propager dans l'ensemble du système en dégradant la qualité des services fournie à l'utilisateur final. Par exemple, l'épuisement de la batterie d'un nœud provoque sa défaillance et par conséquence, elle provoque la perte des messages des nœuds descendants de l'arbre de routage. De la même façon, si la couche finale (la couche applicative du système) présente des fautes logicielles et/ou matérielles, l'ensemble du système sera en panne. Pour remédier à ces problèmes, plusieurs protocoles tolérant aux pannes ont été proposés dans la littérature. Un ensemble de ces protocoles ont été étudiés dans le chapitre précédent et cette étude nous a permis de noter que le protocole PEQ est parmi les meilleurs protocoles proposés. Néanmoins, nous avons remarqué que PEQ souffre de quelques lacunes que nous pouvons améliorer.

Dans ce chapitre, nous avons présenté, dans une première partie, notre proposition nommée EPEQ pour "Enhanced PEQ" qui vise à remédier aux lacunes de PEQ. Dans une deuxième partie, nous avons présenté le simulateur MNSim [26] qui implémente PEQ. Nous avons bien étudié le simulateur mais nous n'avions pas eu le temps nécessaire pour implémenter et évaluer notre proposition EPEQ.

4.2 Description du fonctionnement de EPEQ

L'intérêt major de PEQ se présente dans le fait qu'il répond à plusieurs exigences des RCSFs à la fois, à savoir un latence réduite, la fiabilité, le recouvrement rapide de routes en cas de défaillance et la conservation d'énergie. En plus, PEQ, contrairement à d'autres solutions qui répondent à plusieurs exigences des RCSFs, n'utilise pas de nœud spécial, tous les nœuds sont ordinaires et ne nécessitent ni un matériel spécial ni des calculs compliqués. Néanmoins, PEQ souffre de quelques lacunes auxquelles nous proposons des solutions. Dans la suite de cette section, nous allons présenter notre solution EPEQ (Enhanced PEQ) en soulignons les différentes améliorations que nous avons apporté à PEQ.

EPEQ, comme PEQ, est un protocole de routage qui est réalisé en trois phases. Le protocole commence par la construction d'un arbre par saut. Puis, c'est l'étape de propagation des souscriptions de la station de base vers les nœuds capteurs. La dernière étape est la transmission des paquets de données des nœuds sources vers la station de base. Cette étape implémente un mécanisme de recouvrement de routes en cas de défaillance d'un nœud.

4.2.1 La construction de l'arbre par saut

Dans le réseau de capteurs sans fil considéré ici, un nœud n'a pas une connaissance de la topologie du réseau, à savoir un nœud ne connaît qu'une petite quantité d'informations sur ses plus proches voisins (ceux qui sont à sa portée de transmission). L'algorithme pour la construction de l'arbre par saut est basé sur les inondations de réseau, à partir de la Station de Base (SB), avec une valeur de saut initialisée à zéro '0'. Cette valeur est mémorisée incrémentée et transmis par chacun des voisins de la SB à ses voisins. A leurs tour, ces nœuds voisins stockent la valeur de saut reçue, l'incrémentent et la transmettent à ses voisins, et ainsi de suite jusqu'à la configuration du réseau avec différents niveaux par saut.

Parce que la communication entre les nœuds du réseau est par fréquence radio, tous les voisins d'un nœud reçoivent la transmission. Ainsi, un nœud qui a déjà transmis, peut recevoir la transmission de son voisin, générant une boucle. Pour éviter ces transmissions inutiles qui provoquent le gaspillage d'énergie, les auteurs de PEQ ont défini un ensemble de règles pour la diffusion du niveau par saut. La plus importantes des règles établies consiste à ce que lorsqu'un nœud reçoit un saut de son voisin, il compare cette valeur avec sa valeur de saut locale. Si cette dernière est supérieure à la valeur reçue, le nœud met à jour son saut, incrémente cette nouvelle valeur et la retransmet vers ses voisins. Dans le cas contraire, le nœud ne fait rien.

Comme nous pouvons le remarquer, avec cette méthode, on peut avoir des retransmissions inutiles qui gaspillent beaucoup l'énergie des nœuds. Pour éviter ceci, nous proposons dans EPEQ une nouvelle règle. Un nœud, dans EPEQ, qui reçoit le premier niveau de saut d'un de ses voisins définit un délai d'attente jusqu'à recevoir la plupart des niveaux de saut de ses voisins. Ainsi, une fois le délai est expiré, le nœud choisit le plus petit niveau de saut parmi ceux reçus, l'incrémente et le retransmet. De cette façon, nous réduisons la probabilité qu'un nœud reçoit un niveau de saut

inférieur au sien après l'avoir établi et diffusé vers ses voisins contrairement à PEQ où ce cas est plus fréquent. Ainsi, notre solution évite le gaspillage d'énergie par les retransmissions.

La figure 4.1 montre une configuration initiale d'un réseau maillé après la construction de l'arbre.

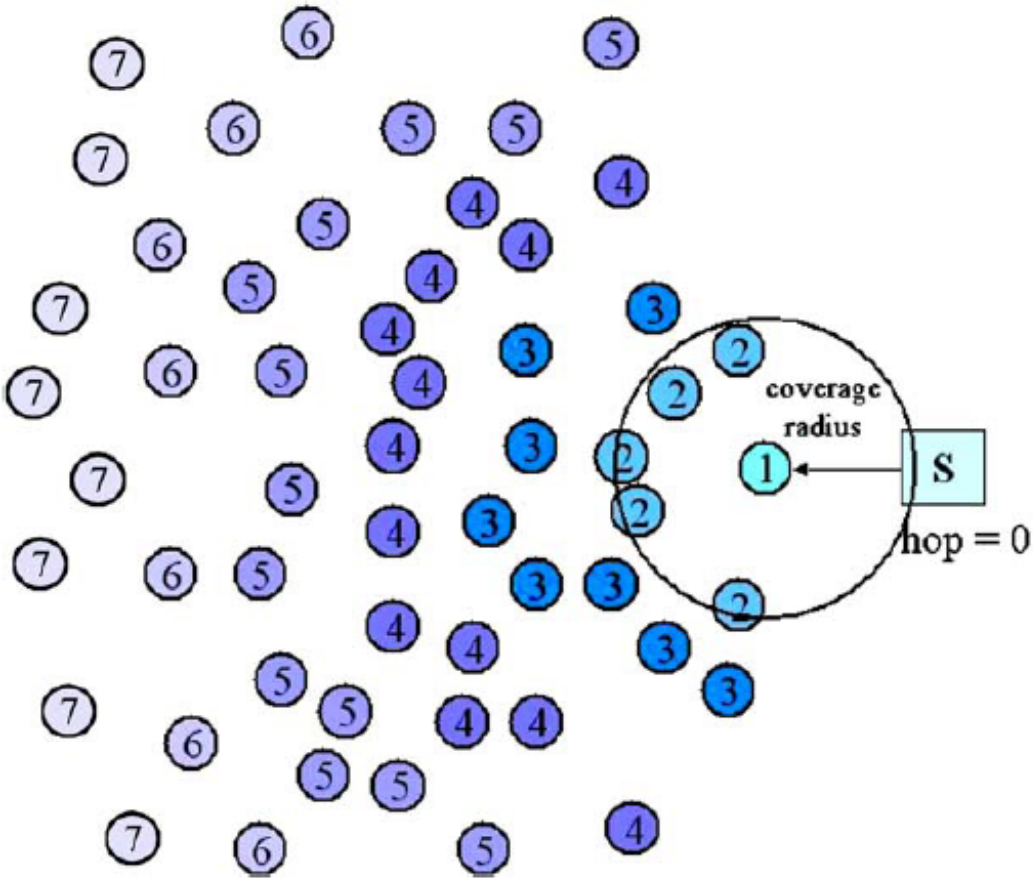

FIGURE 4.1 – La configuration initiale d'un réseau maillé.

La structure de données utilisée dans l'algorithme comporte trois tableaux : configTable, routingTable et subscriptionTable. Le configTable maintient les paramètres de configuration associés à une SB. Un nœud utilise le routingTable pour transférer des paquets à ses voisins. Enfin, le subscriptionTable est utilisé pour stocker les souscriptions d'un nœud. Le routingTable à l'origine comporte quatre champs : sinkID, SenderId, DESTID1 et coordonnées. Dans notre amélioration, nous proposons d'ajouter un chemin de secoure et alors un cinquième champs dans routingTable que nous appelons DESTID2. Les coordonnées de l'attribut sont utilisées pour indiquer la position du nœud, de sorte qu'une application peut savoir d'où les lectures viennent, et la SB peut envoyer une souscription à une région délimitée par des coordonnées, au lieu d'envoyer à des sources spécifiques.

4.2.2 Propagation des paquets de souscription

PEQ utilise le paradigme de notification / souscription, la SB diffuse des demandes à travers le réseau pour exprimer son intérêt à certaines informations captées de l'environnement physique par un ou plusieurs nœuds en définissant un ou plusieurs critères (température > 60 °C, présence de fumée, etc.) qui doivent être adaptés avant d'envoyer tout paquet d'événement. Et on envoi des paquets d'événements seulement quand ils correspondent à un critère, cela réduit le trafic dans le réseau, minimise la consommation d'énergie et prolonge la vie du réseau de capteurs. Après la configuration initiale du réseau, chaque nœud contient juste l'information sur son propre niveau par saut établi dans la première étape. Cette information n'est pas suffisante pour la propagation des souscriptions, et dans l'absence d'autres informations sur les nœuds du réseau qui peuvent satisfaire un intérêt de la SB, il reste une façon de propager la souscription initiale qui est d'inonder le réseau avec cet intérêt. Chaque nœud du réseau conserve une petite table de souscription et une table de routage. Chaque enregistrement de la table de souscription représente une nouvelle souscription. Au cours de la propagation de paquets de souscription, lorsqu'un nœud reçoit ce paquet, il compare les coordonnées attribuées à ses propres coordonnées. Si elles sont identiques, elle sera stockée dans sa table de souscription. Sinon, le nœud transmet la souscription. Au cours de la propagation de la souscription, lorsqu'un nœud reçoit une transmission, il fixe dans sa table de routage les champs DESTID1 et DESTID2 qui correspondent aux identificateurs des deux nœuds qui ont transmis la souscription et le champ sinkID qui correspond à l'identificateur de la SB qui a envoyé la souscription. Lorsqu'un nœud a besoin de transmettre des données à la station de base, il vérifie sa table de routage et transmet le paquet au DESTID1 correspondant à la sinkID, et grâce à notre amélioration si ce dernier ne confirme pas la transmission il retransmis au DESTID2.

4.2.3 Propagation des paquets de notification

Lorsque l'information est captée à partir de l'environnement physique par un capteur, il vérifie la liste de souscriptions pour déterminer s'il y a un intérêt enregistré pour les données. Si un critère est rempli, le nœud vérifie le champs SenderID du nœud qui a transmis la souscription. Après cela, le nœud assemble un paquet de notification d'événements qui contient les attributs suivants : type, la valeur, les coordonnées et sinkID et les envoi à ses voisins. Lorsque chaque nœud voisin reçoit le paquet, il compare le champs destID du paquet reçu avec son propre ID. Si le résultat est vrai, le nœud stock les coordonnées et senderID dans sa table de routage, obtient le DESTID1 de sa table de routage et il lui envoi le paquet et chaque nœud répète l'algorithme jusqu'à ce que la notification atteint la SB.

Chaque nœud traite uniquement les paquets parvenus des nœuds qui sont d'un niveau de saut précédent.

Pour mieux expliquer ces deux dernières étapes nous présentons un exemple de propagation de souscription et de paquet. Supposons qu'une SB S envoie une souscription au réseau, et considérons que le nœud qui est en haut et à gauche dans la grille est le nœud qui produit l'événement qui

répond aux critères de la souscription de la SB S. Le chemin qui est créé vers la SB pour l'envoi du paquet de notification est présenté dans la figure 4.2.

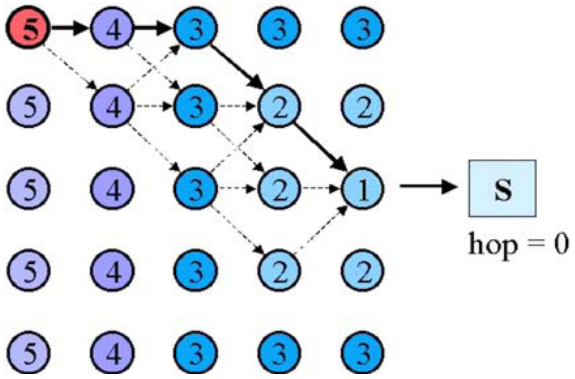

FIGURE 4.2 – Les chemins de paquet de notification.

Noter que les flèches indiquent les liens qui pourraient constituer des voies alternatives qui dépondent seulement du choix de chaque nœud du nœud voisin qui a délivré le message de souscription plus rapidement. Lorsqu'un nœud reçoit une transmission à partir d'un nœud voisin, il ne retransmet le paquet que si le nœud a un plus grand nombre de sauts (une unité plus). Par exemple, seuls les nœuds avec hop = 4 Retransmettre l'information reçu de nœuds avec hop = 5, et ainsi de suite. Le chemin d'accès utilisé pour les données vers l'avant à partir du nœud source au SB peut également être utilisé pour transmettre la souscription au nœud source, tel que représenté sur la figure 4.3.

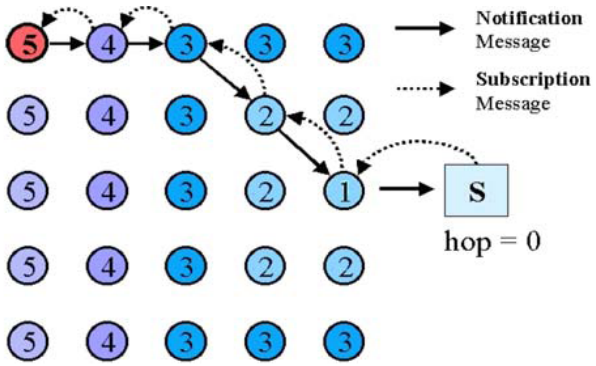

FIGURE 4.3 – Mécanisme de notification/souscription.

4.2.4 Mécanisme de réparation du chemin

Notre proposition EPEQ, comme PEQ, offre un mécanisme de réparation de chemin à base de ACK. Cette réparation se compose de deux parties : La détection de la panne et la sélection d'une nouvelle destination.

Pour la détection des pannes EPEQ fonctionne comme suit. Lorsqu'un nœud a besoin de transmettre des données jusqu'à sa destination, il envoie simplement le paquet "Hello" et définit un délai d'attente et attend l'accusé de réception du voisin, contrairement à PEQ qui envoie directement le paquet de données. Si le nœud émetteur reçoit l'ACK de son voisin, il peut en déduire que le voisin est vivant. Si le nœud émetteur ne reçoit pas le paquet ACK, un problème doit avoir eu lieu avec le voisin et un autre nœud doit être sélectionné comme nouvelle cible dans la deuxième étape.

En plus de ce mécanisme basé sur ACK, nous proposons d'utiliser des messages de niveau d'énergie et d'isolation. Les premiers messages sont envoyés par les nœuds qui ont un niveau d'énergie inférieur à un seuil donné, pour dire qu'ils ne peuvent plus participer à la fonction de routage et que les voisins qui les utilisent comme relai doivent chercher d'autres relai. Les messages d'isolation sont envoyés par les nœuds qui ont détecté que leurs deux routes vers la SB sont défaillantes et qu'ils n'ont plus de chemin fiable pour router les données. Ces deux messages vont permettre d'accélérer le recouvrement des routes défaillantes et de trouver un chemin de secoure parmi les plus courts.

Une fois une défaillance est détectée, il faut passer à la deuxième étape qui consiste à chercher une nouvelle route. Dans cette étape, nous proposons d'utiliser la route de secoure que nous avons sauvegardé dans l'étape de propagation des souscriptions, contrairement à PEQ qui ne sauvegarde qu'une seule route et qui doit lancer une procédure de recherche d'une nouvelle route.

Dans la figure 4.4, nous présentons un exemple de défaillance de route. Dans PEQ, nous avons une seule route du nœud source de niveau 5 vers la SB et il est représenté par les flèches pleines. Nous supposons qu'une défaillance survient au niveau 2de la route, alors le nœud du niveau 3 détecte la défaillance et lance la recherche d'une nouvelle route. En supposant que le deuxième voisin du niveau 2 est aussi en panne, le nœud doit router à travers un nœud du même niveau que lui comme illustré dans la figure 4.5.

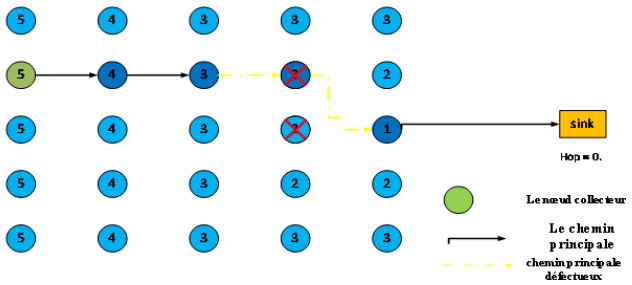

FIGURE 4.4 – Chemin défectueux.

Dans notre proposition EPEQ, nous avons sauvegardé deux routes, une principale qui est la même que dans PEQ et une de secoure qui est représenté en pointillé sur la figure 4.5. Alors, dans notre cas, quand le nœud du niveau 3 détecte que ses deux plus courts chemins sont défaillants, il envoie au nœud du niveau 4 un message d'isolation pour qu'il route ses données par la route de secoure. Et comme on peut le remarquer sur la figure 4.6, dans notre solution, la nouvelle route est de 5 sauts comme les routes défaillantes mais dans PEQ la nouvelle route est de 6 sauts.

Enfin, nous gardons toujours la solution de PEQ pour le cas où un nœud est complètement isolé

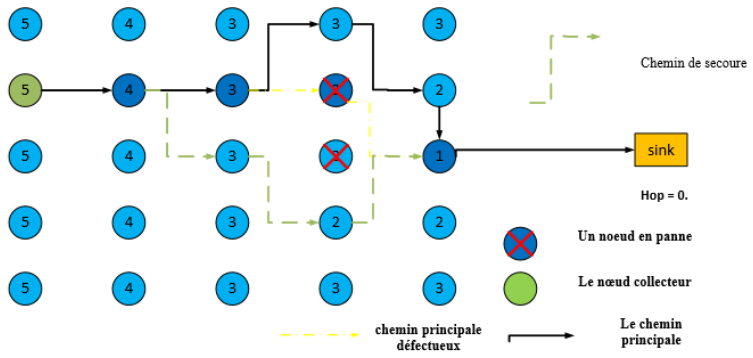

FIGURE 4.5 – Recovrement de chemin dans PEQ.

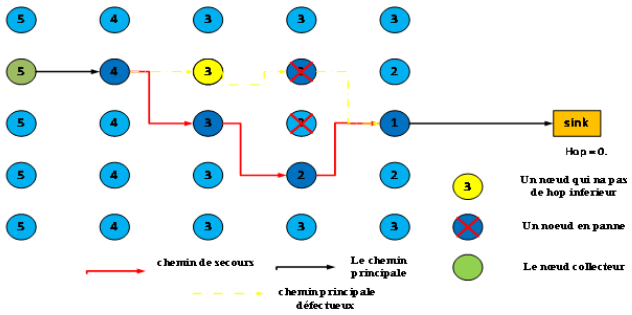

FIGURE 4.6 – Recouvrement de route dans EPEQ.

qui consiste à augmenter la puissance de transmission pour augmenter le rayon de transmission et avoir ainsi de nouveau voisins pour établir une nouvelle route vers la SB.

4.3 Présentation de l'outil de simulation

Pour évaluer les performances de notre solution et les comparer à celles de PEQ, nous avons choisi un simulateur qui implémente PEQ. Ce simulateur est MNSim (pour Modular Network Simulator) qui est développé par Thomas E. Tamayo dans[26]. Nous avons bien étudié et compris le fonctionnement de MNSim mais malheureusement nous n'avion pas eu le temps d'implémenter et d'évaluer notre solution. Ainsi, nous nous contenterons de présenter, dans ce qui suit, MNSim.

MNSim est développé avec le langage de programmation *C♯* "C sharp" et contient plusieurs interfaces.

4.3.1 Langage de programmation

Le langage de programmation *C♯* "C sharp" est l'un des langages intermédiaires qu'utilisent les programmeurs pour créer des programmes exécutables. *C♯*, est un langage dérivé du C++. Il reprend certaines caractéristiques des langages apparus ces dernières années et en particulier de

Java (qui reprenait déjà à son compte des concepts introduits par Smalltalk quinze ans plus tôt). *C♯* peut être utilisé pour créer, avec une facilité incomparable, des applications Windows et Web. *C♯* devient le langage de prédilection d'ASP.NET qui permet de créer des pages Web dynamiques avec programmation côté serveur. *C♯* s'inscrit parfaitement dans la lignée $C \longrightarrow C++ \longrightarrow C\sharp$: le langage C++ a ajouté les techniques de programmation orientée objet au langage C, mais la réutilisabilité promise par C++ ne l'a jamais été qu'au niveau source. Le langage *C♯* ajoute au C++ les techniques de construction de programmes sur base de composants prêts à l'emploi avec propriétés et événements, rendant ainsi le développement de programmes nettement plus aisé. La notion de briques logicielles aisément réutilisables devient réalité.

4.3.2 L'interface ILocation

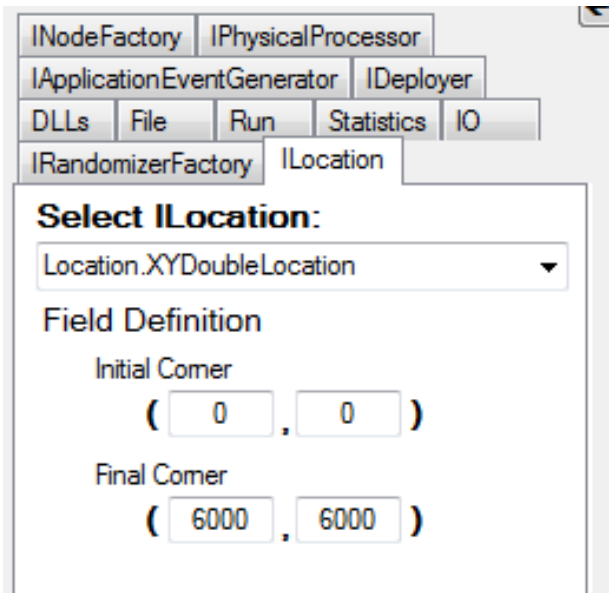

FIGURE 4.7 – L'interface ILocation.

L'onglet ILocation, représenté sur la figure 4.7, définit la surface ou le terrain où nous pouvons déployer les capteurs, comme il définit le type de coordonnées, qui sont actuellement limité aux coordonnées cartésiennes. Le champ est par défaut une zone carrée allant de 0 à 6000 mètres.

4.3.3 L'interface INodeFactory

Grâce à cette interface, l'utilisateur peut régler certains paramètres d'nœud capteur, la figure 4.8 représente des paramètres enregistrés par défaut sur la classe de nœud.

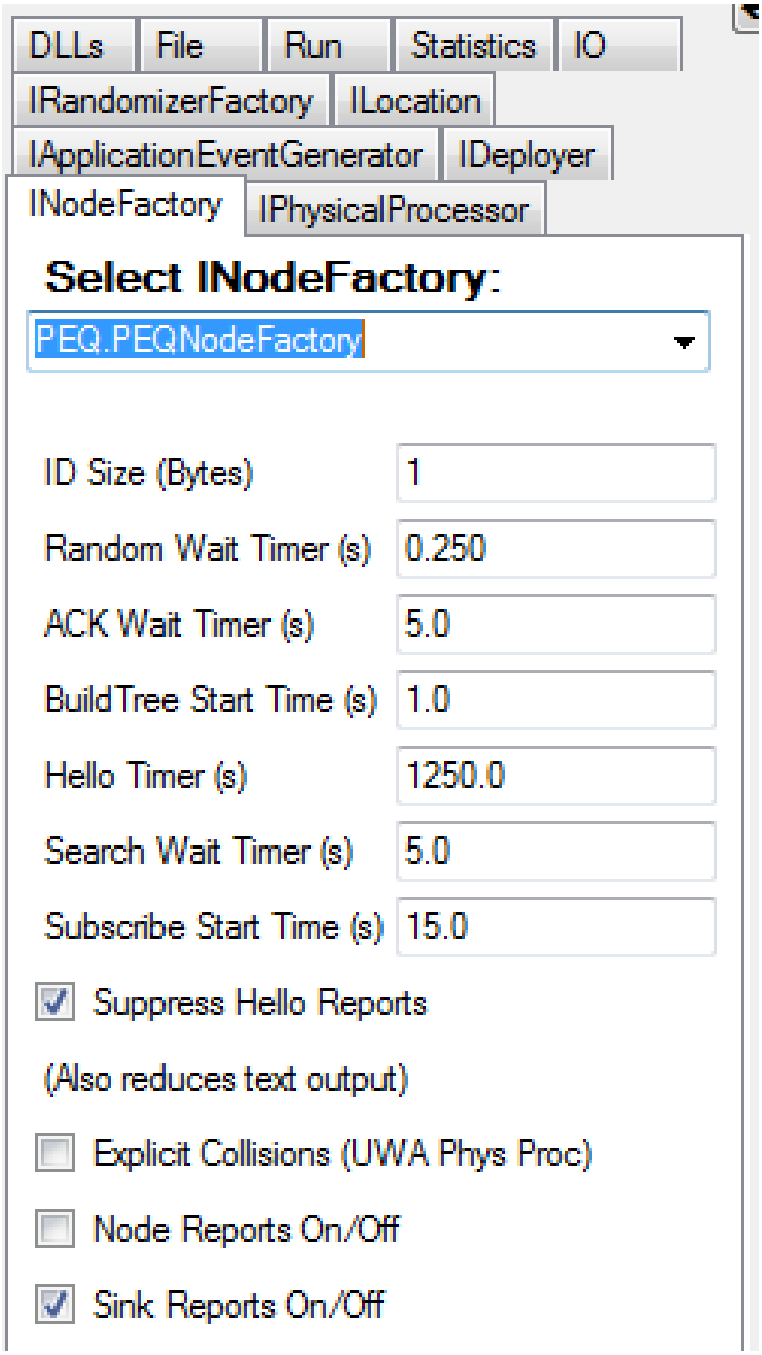

FIGURE 4.8 – L'interface INodeFactory.

4.3.4 L'interface IDeployer

L'interface IDeployer permet à l'utilisateur de déterminer comment les nœuds sont placés dans le champs de la simulation. Quatre options de déploiement sont disponibles. Deux options comprennent le déploiement de nœuds au long d'une grille fixe (GridDeployer) et aléatoire (RandomDeploye). La troisième option représentée sur la figure4.9, montre une spirale Archimédien.

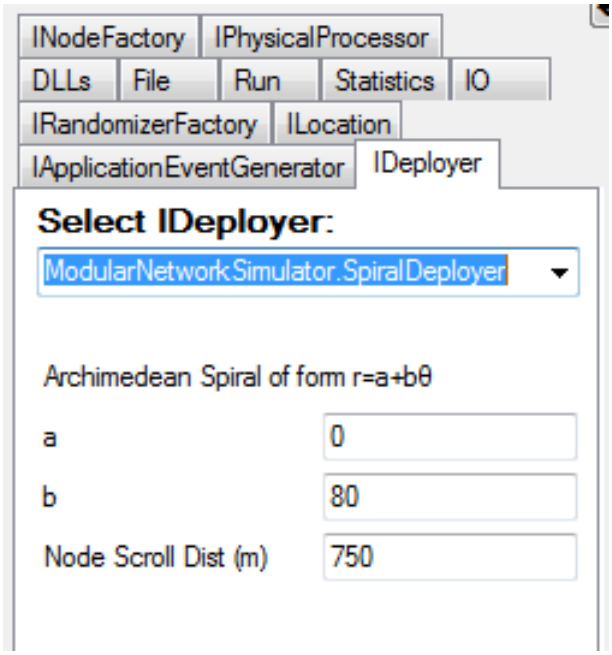

FIGURE 4.9 – IDeployer Interface.

L'objectif était d'avoir les nœuds à une distance déterminée. Une approximation est dérivée de la séparation, qui est constante dans la spirale d'Archimède. Il serait également avantageux d'avoir une distribution ainsi qu'une distance aléatoire entre les nœuds au long de la spirale. Un quatrième déploiement, le PEQTestDeployer, Ceci est équivalent à laRandomDeployer à l'exception des cinq nœuds supplémentaires situés à des distances égales le long du côté gauche du terrain et un nœud supplémentaire fixé au centre du côté droit de champ. Le nœud de droite est configuré pour être le nœud SB. Les nœuds de la gauche fonctionnent en liaison avec l'PEQTestApplication pour forcer les événements à se produire de leurs endroits. Ces cinq nœuds deviennent alors sources et lanceront cinq événements simultanés via l'PEQTestApplication.

4.3.5 L'interface IApplicationEventGenerator

Le générateur de base des événements d'application crée une zone où un événement censé se produit. Un nœud quelconque à l'intérieur de la zone concernée reçoit un message de la couche application. Ces événements peuvent être configurés pour se reproduire à des intervalles spécifiques. Actuellement, les événements créés parPEQTestApplication sont statiquse.

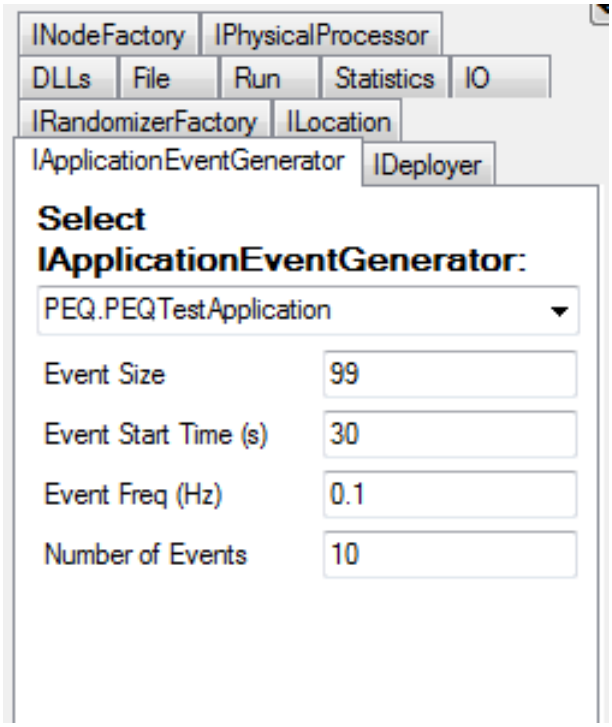

FIGURE 4.10 – L'interface IApplicationEventGenerator.

4.4 Fonctionnement de PEQ sur MNSim

Le simulateur implémente les trois étapes de PEQ : construction de l'arbre par saut, l'envoi de données, et la construction de chemin de réparation.

4.4.1 La construction de l'arbre

La classe de construction de l'arbre comporte deux parties : la construction d'un arbre, et la propagation des souscriptions de la SB. La construction de l'arbre commence par la SB, puis le message se propage pour inonder tous le réseau, comme le montre la figure 4.11.

Dans la figure suivante (4.12), le point vert représente la SB et les autres points bleus les nœuds capteurs, les nœuds avec un cercle vert représente le niveau 1 et les flèches représentent la direction de plus court chemin de chaque nœud. Donc à chaque fois qu'un niveau et construis ce dernier cherche son niveau supérieur et en arête quand tous les nœuds sont marqués comme le représente la figure suivante.

Cette figure (4.12) représente l'étape finale de construction de l'arbre, on remarque que dans cette figure tous les capteurs sont marqués par une flèche verte dons la direction d'un nœud de niveau inférieur jusqu'à la SB, ce marquage représente le plus court chemin vers la SB.

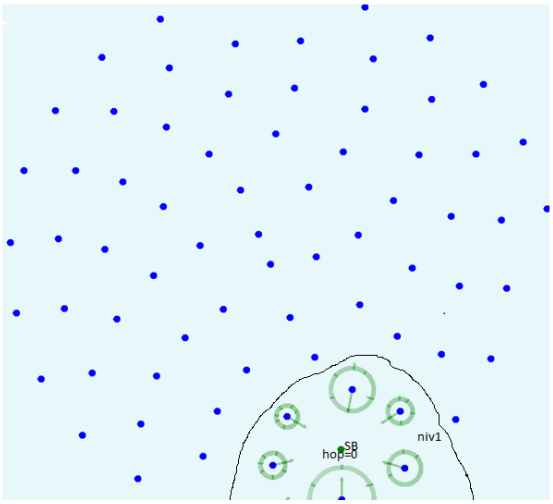

FIGURE 4.11 – Propagation des messages de construction de l'arbre.

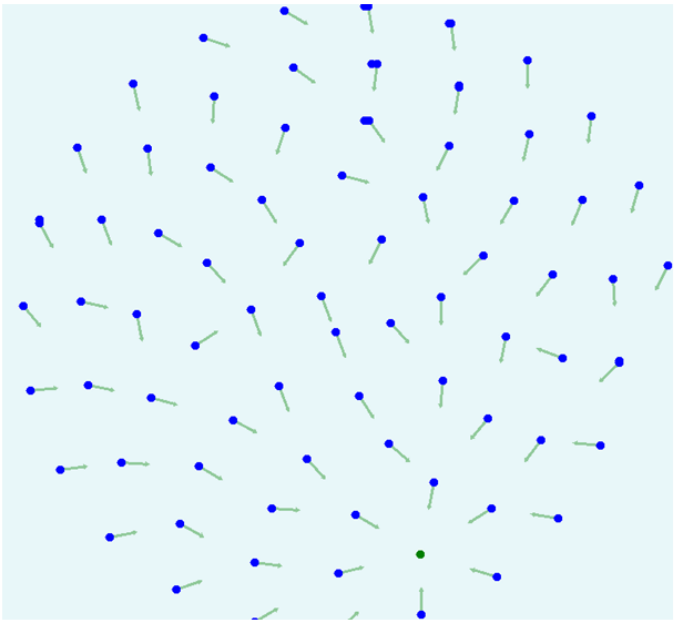

FIGURE 4.12 – Fin de la construction de l'arbre.

4.4.2 La propagation de la souscription

Le message de souscription est représenté comme un anneau rouge, comme le montre la figure 4.13.

Comme la souscription se propage, chaque nœud qui intercepte ce message stocke l'ensemble des informations dans sa table de souscription puis il le transmit à ses voisins. Pour différencier l'arbre et les messages de souscription, la flèche de la route passe au rouge si le nœud enregistre une souscription. Ceci est illustré sur la figure 4.14. Cela signifie également que la voie peut changer, soit en raison d'un meilleur nombre de sauts ou du même nombre de sauts reçu d'un autre nœud.

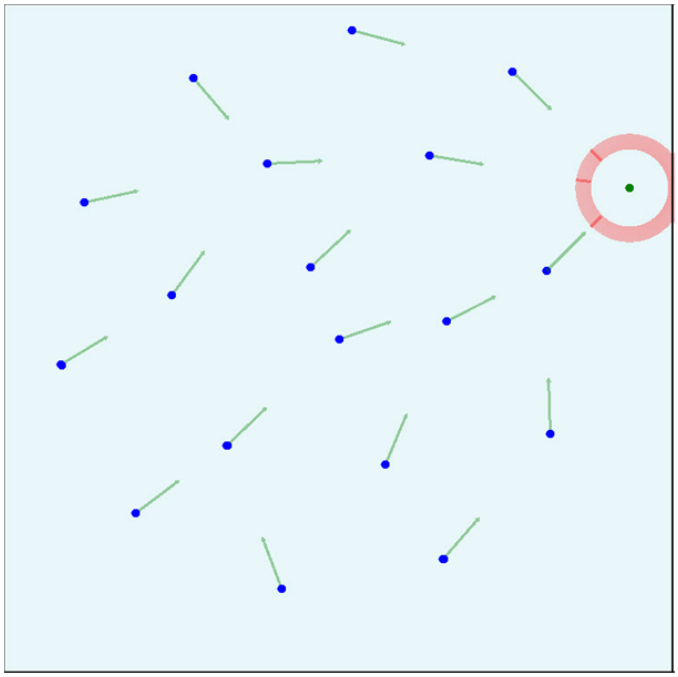

FIGURE 4.13 – La SB envoie un message de souscription.

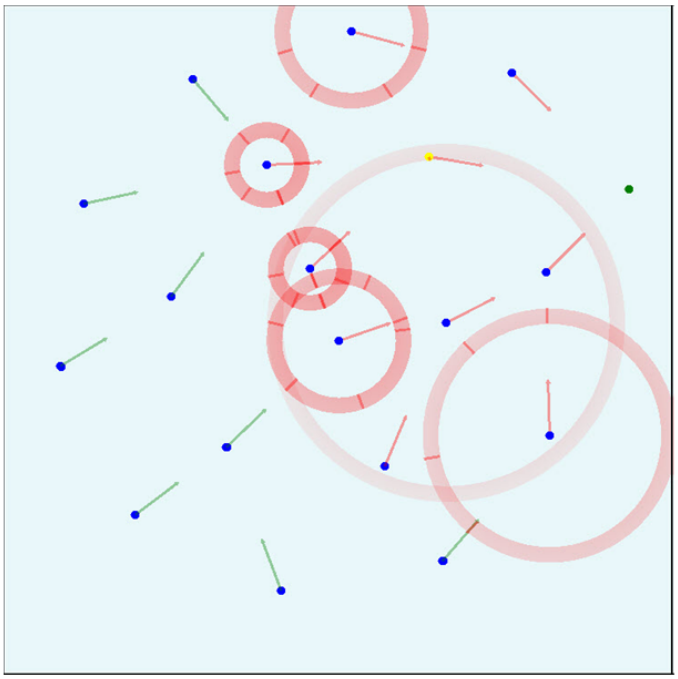

FIGURE 4.14 – La propagation de la souscription.

4.4.3 Envoi des données vers la station de base

Si un événement se produit, le nœud envoie un message qui contient ces informations, tous les nœuds voisins entendent le message mais seul le nœud qui a transmit la souscription qui le reçoit,

puis il envoie un ACK si la transmission s'est bien passé, ainsi de suite jusqu'à ce que le message arrive à la station de base.

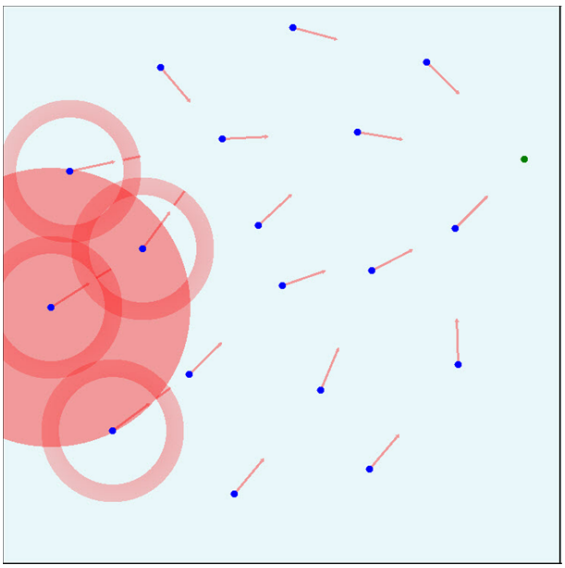

FIGURE 4.15 – L'occurrence d'un évènement dans le réseau.

4.4.4 La réparation de chemin de transmission

La figure 4.16 représente une collision entre un ACK et une autre information ce qui va provoquer une perte de la route pour l'envoi des données,

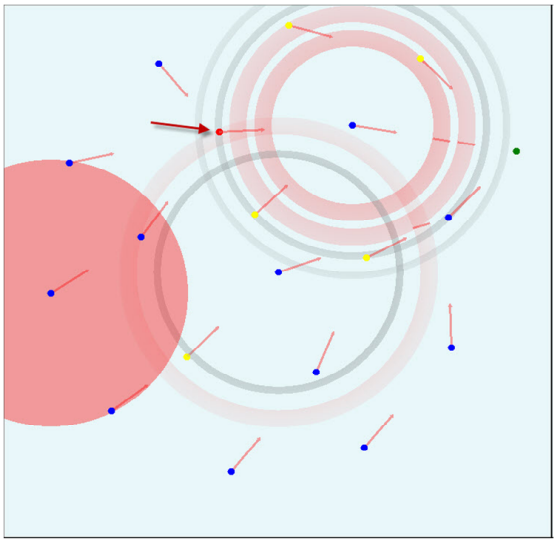

FIGURE 4.16 – La collision entre un ACK et une autre donnée.

Une fois un nœud perd sa route, il envoie un message de recherche demandant à tous ses voisins

leur nombre de sauts vers la station de base. La figure 4.17 montre ce message, visualisées comme un anneau orange, diffusé par le nœud qui a perdu sa route.

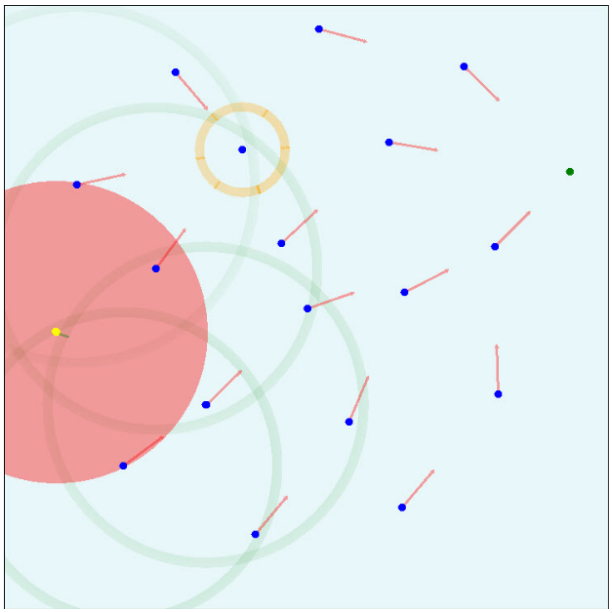

FIGURE 4.17 – Recherche des nœuds voisins.

Les messages de réponse sont envoyés par les nœuds voisins qui ont entendu le message de recherche, illustré sur la figure 4.18. Les messages de réponse sont visualisés comme des anneaux verts.

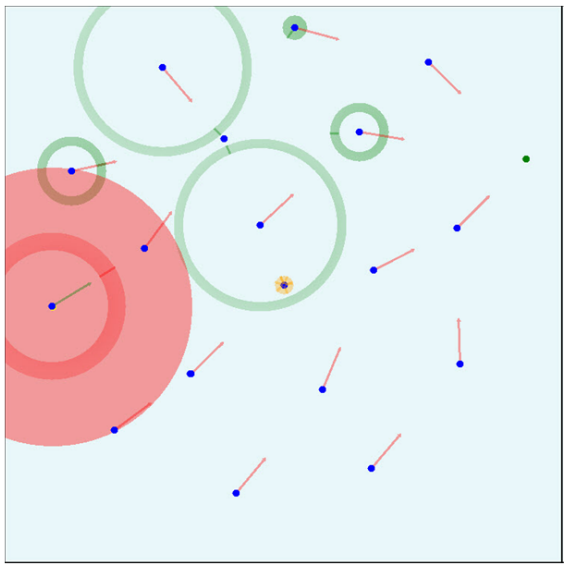

FIGURE 4.18 – Les messages de réponse.

Il y a deux choses à noter également sur cette figure. Tout d'abord, un autre nœud a perdu sa route et commence le processus de recherche. Deuxièmement, la recherche / réponse produises

beaucoup de trafic concentré dans une zone sur un certain délai de temps.

4.5 Conclusion

Dans ce chapitre nous avons présenté notre proposition EPEQ qui est une amélioration de PEQ. EPEQ est un protocole de routage tolérant aux pannes. Pour l'évaluation de performance de EPEQ, nous avons installé et étudier le simulateur MNSim dans lequel PEQ est déjà implémenté mais faite de temps nous n'avons pas peu implémentés nos amélioration et évaluer EPEQ.

Conclusion générale et Perspectives

Les réseaux de capteurs sans fil ont un large potentiel et constituent un sujet de recherche innovant ainsi qu'un outil convoité par plusieurs domaines.
C'est sans aucun doute, une technologie qui va nous accompagner pour les prochaines années et ainsi faire partie de notre vie quotidienne. Cependant, il y a encore beaucoup de problèmes qui doivent être abordés pour un fonctionnement efficace de ces réseaux dans des applications réelles.
La tolérance aux pannes est l'un des problèmes fondamentaux dans ces réseaux
Une panne au niveau d'un capteur peut se produire à cause d'une perte de connexions sans fil due à l'extinction du capteur suite à l'épuisement de sa batterie, ou tout simplement à une destruction physique accidentelle ou intentionnelle par un ennemi. Par conséquence, il faut faire face à ces pannes en proposant des protocoles tolérants aux pannes.
Dans ce mémoire, nous avons réalisé une étude pour atteindre un routage efficace avec tolérance aux pannes dans les réseaux de capteurs sans fil. Cet aspect est fondamental pour ce genre de réseau où le routage se réalise en collaboration avec les différents nœuds du réseau. De ce fait, un protocole de routage doit prendre en compte les contraintes matérielles d'un capteur : une énergie limitée, une capacité de stockage modeste, une bande passante faible, etc. En guise de perspective, Nous proposons pour la suite de ce travail de développer l'implémentation de notre protocole EPEQ, et prendre en charge tous les critères, comme l'énergie, les chemins de secoure, etc, dans le but de faire une comparaison entre notre améliorations et PEQ ainsi que d'autre protocole de routage similaire. Comme nous vison à ajouter à notre amélioration des clusters qui vas réduire surement la consommation d'énergie. Bien que notre projet nés pas en grand complait et il est besoin surement a une grande amélioration, mais tout ça nous empêche pas d'être fières et reconnaissons pour tous les connaissances qui nous a permis d'acquérir soit qui concerne le routage dans les réseaux de capteur sans fil, soit dans la programmation et la simulation.

Bibliographie

[1] M. L. Messai, "Sécurité dans les réseaux de capteurs sans-fil", mémoire de magistère, Université de Abderrahmane Mira de Bejaia, 2008.

[2] S. Yessad, " Sécurité dans les réseaux de capteurs sans-fil", thèse de Doctorat Université de Abderrahmane Mira de Bejaia, 2015.

[3] L. I. NIAR, "Analyse Graphique pour la surveillance dans un réseau de capteurs sans fils (RCSF)", Mémoire Magister, Université d'Oran Es-senia, 2012.

[4] S. Yessad, "Couche MAC avec contrainte d'énergie et équité dans les réseaux de capteurs", mémoire de magistère, Université de Abderrahmane Mira de Bejaia, 2006.

[5] C. Duran-Faundez, "Transmission d'images sur les réseaux de capteurs sans fil sous la contrainte de l'énergie" ,thèse de Doctorat, Université Henri Poincaré, Nancy 1, 2009.

[6] K. Rahim, "Techniques de conservation d'énergie pour les réseaux de capteurs sans fil", thèse de Doctorat Institut National Polytechnique de Toulouse, 2009.

[7] K.Beydoun, "Conception d'un protocole de routage hiérarchique pour les réseaux de capteurs", thèse de Doctorat, Université, de Franche-Compte, 2009.

[8] k. Benahmed, "Approche théorie des graphes pour la surveillance d'un réseau de capteurs sans fil", mémoire de magistère, Université d'Oran Es-senia,2007.

[9] Y. Shou, "Cryptographie sur les courbes elliptiques et tolérance aux pannes dans les réseaux de capteurs", thèse de Doctorat,, 2014.

[10] T. Carpentier, "Placement de Capteurs Pour la Surveillance Des Processus Complexes", thèse de Doctorat, Université des Sciences et Technologies de LILLE, 1999.

[11] S. Ray, S. Dash, N. Tarasia, A. Ajay, AR. Swain, " Fault tolerant multilevel routing protocol with sleep scheduling (fms) for wireless sensor networks", European Journal of Scientific Research, Vol. 55(1) : pp97-108, 2011.

[12] A.Zemmar,"tolérance aux défaillances dans les réseaux de capteurs sans fil",mémoire de magistère, Université de Abderrahmane Mira de Bejaia, 2008

[13] D. Hamdan, "Détection et diagnostic des fautes dans des systèmes à base de réseaux de capteurs sans fils", thèse de Doctorat, Université de Grenoble, 2013.

[14] J. N. Al-Karaki et A. E. Kamal, "Routing Techniques in Wireless Sensor Networks : A Survey", IEEE Wireless Comm.,vol. 11, pp6-21, 2004.

[15] F. Theoleyre, "Une auto-organisation et ses applications pour les réseaux ad hoc et hybrides", These de Doctorat, 2006.

[16] S. Ray, S. Dash, N. Tarasia, A. Ajay, AR. Swain, "A Dynamic Fault Tolerant Routing Protocol for Prolonging the Lifetime of Wireless Sensor Networks", Journal (IJCSIT), Vol. 2 (2), pp727-734, 2011.

[17] K. Kulothungan, J.Angel Arul Jothi, A. Kannan, "An Adaptive Fault Tolerant Routing Protocol with Error Reporting Scheme for Wireless Sensor Networks", European Journal of Scientific Research ISSN 1450-216X N ° 1 Vol.60 2011, pp 19-32.

[18] G. Wu, C. Lin, F. Xia, L. Yao, H. Zhang, B. Liu ,"Dynamical Jumping Real-Time Fault-Tolerant Routing Protocol for Wireless Sensor Networks", de la Fondation nationale des sciences naturelles de Chine par la concession numéro 60703101 et n ° 60903153 2010.

[19] Z. Che-Aron, W. Al-Khateeb, et F. Anwar, "The Enhanced Fault-Tolerance Mechanism of AODV Routing Protocol for Wireless Sensor Network", IJCSNS International Journal of Computer Science et de sécurité réseau, Vol.10 No.6, Juin 2010.

[20] F.Z. Benhamida, Y. Challal, "FaT2D : Fault Tolerant Directed Diffusion for wireless sensor networks", International Conference on Availability, Reliability, and Security, 2010 (ARES '10), pp112-118, Krakow, Poland.

[21] A.Boukerche et al, "A Fast and Reliable Protocol for Wireless Sensor Networks in Critical Conditions Monitoring Applications", International Conference (MSWiM'04), Canada, 2004.

[22] N. Pais, B. K. Cetin, N. Pratas, F.J. Velez, N. R. Prasad et R. Prasad, "Cost-Benefit Aware Routing Protocol for Wireless Sensor Networks with Hybrid Energy Storage System", Journal of Green Engineering, 189–208, 2011

[23] D. Herbert, Y. Lu, S. Bagchi and Z. Li," Detection and Repair of Software Errors in Hierarchical Sensor Networks", IEEE International Conference on Sensor Networks, Ubiquitous and Trustworthy Computing, June 2006, Taichung, Taiwan, pp. 403 - 410.

[24] A. Boukerche et al, "Fault-tolerant wireless sensor network routing protocols for the supervision of context-aware physical environments", Journal of Parallel and Distributed Computing, Vol. 66 no. 4, pp586-599, 2006.

[25] Fei. Dai et Jie. Wu, "On constructing k-connected k-dominating set in wireless ad hoc and sensor networks", Journal of Parallel and Distributed Computing, Vol. 66 no. 7, pp947-958, 2006.

[26] Thomas E. Tamayo,"A modular,visual simulator of underwater sensor networks", mémoire de master, Université de RHODE ISLAND 2011.

Résumé

Les réseaux de capteurs constituent un axe de recherche très fertile ces dernières années. Cette technique se développe dans différents domaines comme l'environnement, l'industrie, le commerce, la médecine, l'armée, etc. Les réseaux de capteurs sont difficiles à concevoir parce qu'ils sont fortement contraints en énergie et que tous les éléments ont potentiellement une influence sur la durée de vie du système. Dans ce projet, nous avons visé les protocoles de routage tolérants aux pannes pour garantir la fiabilité de livraison des données à la station de base. Dans cette optique, nous avons essayés d'évalués les performances d'un protocole de routage à base hiérarchique nommé PEQ dans l'objectif de prolonger la durée de vie du réseau et assurer une bonne connectivité entre les nœuds capteurs. Comme nous avons développé un simulateur capable de simuler presque la totalité de notre évaluation.
Mots clés : Réseaux de capteurs, tolérance aux pannes, évaluation des performances, simulation.

Abstract

Sensor networks are a very fertile axis of research in recent years. This technique is developed in various fields such as environment, industry, commerce, medicine and the military, etc. Sensor networks are difficult to design because they are highly constrained by energy and all the elements potentially influence the lifetime of the system. In this project we aimed tolerant routing protocols fault to ensure reliable delivery of data to the base station. In this light, we tried to evaluate the performance of a basic hierarchical routing protocol named PEQ with the aim to extend the network lifetime and ensure proper connectivity between the sensor nodes. As we have developed a simulator that can simulate almost all of our evaluation.
Keywords : sensor networks, fault tolerance, performance evaluation, simulation.