MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITE FERHAT ABBAS – SETIF
UFAS (ALGERIE)

Mémoire

Présenté à la Faculté des Sciences de l'Ingénieur
Département d'informatique
En vue de l'Obtention du Diplôme de

MAGISTER

ECOLE DOCTORALE D'INFORMATIQUE (STIC)
Option : ingénierie des systèmes informatiques (ISI)

Par
Mr **Ould Mohamedi Nedjib**

Thème :

Tolérance aux pannes d'une station de base (SINK) dans un réseau de capteurs sans fil.

Soutenu le : devant un Jury composé de :

REFOUFI Allaoua	M.C. à l'université de Sétif	Président
ALIOUAT Makhlouf	M.C. à l'université de Sétif	Rapporteur
BOUKERRAM Abdallah	M.C. à l'université de Sétif	Examinateur

Remerciements

Je remercie avant tout, DIEU tout puissant qui, sans Sa volonté, ce travail n'aurait jamais vu le jour. « الحمد و الشكر لله »

Je remercie mon directeur de projet, Mr le Dr Aliouat Makhlouf pour avoir bien voulu diriger et superviser ce projet ayant pour thème un domaine très instructif et intéressant, mais aussi, pour ses conseils avisés, ses encouragements, sa patience et la confiance qu'il m'a accordé.

Je remercie également le Dr A. Refoufi en tant que président du jury et le Dr A. Boukerram en tant qu'examinateur, pour m'avoir accordé l'honneur de bien vouloir juger ce modeste travail.

Mes remerciements les plus chaleureux vont enfin, à toute personne ayant contribué un temps sois peu, de près ou de loin, à l'accomplissement de ce modeste travail.

À ma chère mère, à mon cher père.

À mes chers frères et sœurs.

À tous ceux pour qui je compte.

Résumé

Un réseau de capteurs sans fil (RCSF) est un ensemble de nœuds de petites tailles, combinant à la fois des capteurs, une unité de traitement et un dispositif de communication sans fil, ayant pour mission la collecte et l'acheminement de données environnementales vers un utilisateur final, en passant par un nœud spécial appelé station de base ou SINK. La tolérance aux pannes dans de tels réseaux représente un facteur de conception capital du fait de la prédisposition des nœuds aux pannes, due entre autres, à la nature hostile des environnements de déploiement.

D'un autre coté, l'aspect centralisé de l'évacuation des données vers des réseaux extérieurs en passant par le nœud collecteur (SINK) d'un RCSF represente une véritable vulnérabilité, aussi, la tolérance aux pannes du SINK est primordiale et permet d'éviter un échec de la mission du RCSF, non toléré dans certaines applications,. Dans ce travail, nous proposons une stratégie permettant à un réseau de capteurs sans fil de recouvrir d'une panne de son SINK primaire, se basant sur le paradigme de communication Directed Diffusion et sur l'utilisation d'un second SINK de surveillance. Nous proposons un modèle UPPAAL pour ensuite en vérifier des propriétés importantes, validant le bon fonctionnement de la stratégie proposée.

Mots-clé: WSN; Fault tolerance; Directed diffusion; SINK

Abstract

A Wireless sensor network (WSN) is a set of tiny nodes that combine sensing, computation and wireless communication, whose mission is to collect and deliver environmental data to an end user via a base station (SINK). Fault tolerance in such networks is a crucial design matter due to the hostile environment nature that will increases the node's failure rate.

On the other hand, the failure of a WSN's SINK is an important issue that may leads to the WSN's mission fail, not allowed in some applications; it becomes obvious that a fault tolerance strategy is needed at this weakness point. In this work, we propose a strategy that enables a WSN to recover from a crash-type failure of its primary SINK, based on the Directed Diffusion communication paradigm and the use of a secondary watch-dog SINK. An UPPAAL model is proposed and some of its important properties are verified.

Keywords: WSN; Fault tolerance; Directed diffusion; SINK

Sommaire

Introduction générale

Les avancées technologiques dans des domaines tels que l'électronique, la télécommunication (en l'occurrence sans fil) ou encore la miniaturisation des composants électroniques ont donné naissance à une toute nouvelle classe de systèmes distribués à moindre coût de fabrication et de déploiement à large échelle, à savoir les Réseaux de Capteurs Sans Fil (RCSF). Ces systèmes présentent entre autres, la particularité d'être intimement liés aux environnements dans lesquels ils sont appelés à être déployés et ont pour mission la récolte de données environnementales de phénomènes physiques tel le degré de température, la pression, les niveaux acoustiques, l'humidité etc.

Un RCSF est composé d'un ensemble de nœuds autonomes dotés de capteurs, pouvant effectuer des traitements sur les données collectées et ayant la possibilité de communiquer entre eux à l'aide de dispositifs de communication sans fil. L'objectif étant d'acheminer les données obtenues sur le terrain vers une station de base du RCSF aussi appelé SINK (puits) qui va, à son tour, relayer les mesures reçues vers l'administrateur du RCSF. On peut donc remarquer l'aspect pluridisciplinaire impliqué dans les recherches menées sur les RCSF ; en effet, cela va du réseau à la gestion de base de données en passant par le traitement de signal et l'optimisation des ressources, sans oublier bien sûr, la tolérance aux pannes qui est le principal sujet de ce travail, et plus particulièrement, celle du nœud collecteur (SINK).

L'architecture du RCSF repose donc sur un point central de collecte de données qui est le SINK, ce qui confère à la survie de ce dernier, une importance capitale en dépit des pannes auxquelles il peut être soumis. La tolérance aux pannes de ce nœud spécial est donc primordiale puisqu'un crash de celui-ci signifie l'arrêt total de l'acheminement des données vers l'utilisateur final. Il est donc vital d'assurer la continuité de la mission d'un RCSF si le SINK venait à cesser de fonctionner.

Nous proposons dans ce mémoire, une stratégie de tolérance aux pannes du SINK dans un RCSF, reposant sur le constat suivant : les nœuds du RCSF ne peuvent assurer la mission du SINK, car celui-ci est considéré comme un nœud spécial tant du coté matériel que du coté logiciel, il faut donc avoir recours à un second SINK qui surveillera et relaiera le SINK primaire en cas de panne. Cette stratégie aura pour contexte de communication, le paradigme

Directed Diffusion, basé sur une diffusion d'intérêts, de données, et l'établissement de gradients pour l'acheminement des différents messages.

La stratégie à préconiser impliquera que les algorithmes de tolérance aux fautes à proposer devront être exécutés au niveau de chaque SINK, et en cas de défaillance du Sink primaire, le sink secondaire prendra immédiatement le relai. L'impact du mécanisme proposé, sur le temps nécessaire au recouvrement d'une panne ainsi que sur la consommation d'énergie engendrée en conséquence, sera discuté et examiné, une modélisation et vérification sera effectué dans le model-checker UPPAAL pour finir.

Le mémoire sera organisé comme suit :

Dans le premier chapitre (1), nous ferons un tour d'horizon sur les réseaux de capteurs sans fil, un état de l'art sur cette nouvelle famille de réseaux ad hoc.

Le second chapitre (2) sera consacré quant à lui, à la tolérance aux pannes dans les réseaux de capteurs sans fil de façon générale, et à celle d'une station de base au sein d'un RCSF en particulier. Nous exposerons pour cela, quelques uns des rares travaux traitant de cette problématique.

Nous détaillerons dans le troisième chapitre (3), le fonctionnement du paradigme de communication Directed Diffusion sur lequel sera basée la stratégie proposée.

Les deux derniers chapitres se pencheront d'avantage sur la stratégie en elle-même, en en détaillant le principe et les différents mécanismes, l'impact de celle-ci sur la consommation d'énergie et le temps de recouvrement d'une panne. Une modélisation sera par la suite réalisée à l'aide du model-checker UPPAAL afin de valider la stratégie proposée.

CHAPITRE 1 :

RESEAUX DE CAPTEURS SANS FIL

Les réseaux de capteurs sans fil offrent un panel très intéressant de possibilité d'exploitation, en effet, un réseau de dispositifs de petite taille communicant par ondes radio et pouvant effectuer des traitements sur les données récoltées peut avoir d'innombrables applications dans de nombreux domaines, ceci suscite l'enthousiasme de la communauté scientifique et permet d'avoir une quantité impressionnante de travaux de recherches traitant des RCSF.

1.1 Particularités des RCSF

D'une manière générale et non exhaustive, voici quelques caractéristiques propres aux RCSF [HW05, FS04, AFBI04, ASSC01] :

- Le nombre de nœuds déployés peut être considérablement élevé (on parle de millier de nœuds)
- Les RCSF (nœuds) ont des contraintes de ressources très strictes (batterie, puissance de calcul, mémoire.. etc.)
- Le déploiement du RCSF est effectué d'une façon aléatoire et la position des nœuds sur le terrain après le déploiement n'est pas prédéterminée.
- Le RCSF (protocoles et algorithmes) doit présenter une caractéristique d'auto organisation du fait de la difficulté, voir de l'impossibilité, d'une intervention externe post déploiement sur le RCSF.

1.1.1 RCSF : un type particulier de réseaux Mobile Ad hoc

Définition d'un réseau mobile ad hoc ou MANET d'après le groupe MANET de l'IETF

- Un réseau ad hoc comprend des plates-formes mobiles appelées nœuds qui sont libres de se déplacer sans contrainte. Un réseau ad hoc est donc un système autonome de nœuds mobiles. Ce système peut fonctionner d'une manière isolée ou s'interfacer à des réseaux fixes au travers de passerelles.

Cette définition générale permet de mettre en avant les caractéristiques qui différencient les réseaux mobiles ad hoc (Mobile Ad hoc Networks ou MANET) des réseaux classiques [HW05, PJ].

- ✓ *Mobilité* : La mobilité des nœuds constitue à l'évidence une caractéristique très spécifique des réseaux ad hoc. Cette mobilité est intrinsèque au fonctionnement du réseau. Elle se distingue de la nomadicité (mobilité des seuls nœuds terminaux) ou de l'itinérance (équipements statiques mais pouvant être déplacés). Dans un réseau ad hoc, la topologie du réseau peut changer rapidement, de façon aléatoire et non prédictible et les techniques de routage des réseaux classiques, basées sur des routes préétablies, ne peuvent plus fonctionner correctement.

- ✓ *Equivalence des nœuds du réseau* : Dans un réseau classique, il existe une distinction nette entre les nœuds terminaux (stations, hôtes) qui supportent les applications et les nœuds internes (routeurs par exemple) du réseau, en charge de l'acheminement des données. Cette différence n'existe pas dans les réseaux ad hoc car tous les nœuds peuvent être amenés à assurer des fonctions de routage.

- ✓ *Liaisons sans fil* : Les technologies de communication sans fil sont indispensables à la mise en place d'un réseau ad hoc. Malgré des progrès très importants, leurs performances restent et resteront en deçà de celles des technologies des réseaux filaires. La bande passante est moins importante, alors que le routage et la gestion de la mobilité génèrent davantage de flux de contrôle et de signalisation que dans une architecture de réseau filaire. Ces flux doivent être traités de façon prioritaire pour prendre en compte rapidement les modifications de topologie.

- ✓ *Autonomie des nœuds* : La consommation d'énergie constitue un problème important pour des équipements fonctionnant grâce à une alimentation électrique autonome. Ces équipements intègrent des modes de gestion d'énergie et il est important que les protocoles mis en place dans les réseaux ad hoc prennent en compte ce problème.

- ✓ *Vulnérabilité* : Les réseaux sans fil sont par nature plus sensibles aux problèmes de sécurité. Pour les réseaux ad hoc, le principal problème ne se situe pas tant au niveau

du support physique mais principalement dans le fait que tous les nœuds sont équivalents et potentiellement nécessaires au fonctionnement du réseau. Les possibilités de s'insérer dans le réseau sont plus grandes, la détection d'une intrusion ou d'un déni de service est plus délicate et l'absence de centralisation pose un problème de remontée de l'information de détection d'intrusions.

Il est évident que les RCSF sont assez similaires aux réseaux ad hoc, du fait de la structure générale du réseau, à savoir des nœuds distincts sans aucun lien physique communicant entre eux, tout en tolérant la suppression et l'ajout de nœuds au réseau à tout moment. Cependant, les algorithmes et protocoles existant pour les réseaux ad hoc se voient être inefficaces pour les RCSF sans des modifications et adaptations significatives. Il est donc intéressant d'évoquer les principales différences qui existent entre ces deux genres de systèmes.

1.1.2 RCSFs *VS* Réseaux Ad Hoc

Ce qui suit résume les principales divergences entre RCSFs et réseaux ad hoc [FS04, ASSC01] :

- Nombre de nœuds beaucoup plus élevé dans les RCSFs comparé au Réseaux ad hoc.
- Le déploiement des RCSFs est assez dense.
- Les RCSFs sont beaucoup plus sujets aux pannes.
- Le Changement de topologie est très fréquent dans les RCSFs
- Les nœuds d'un RCSF adoptent en général un paradigme de communication basé sur la diffusion (Broadcasting), alors que les réseaux Ad Hoc sont plus orientés vers la communication point à point.
- Les contraintes de ressources telles que la puissance de calcul ou alors l'énergie sont beaucoup plus strictes dans les RCSFs que dans les réseaux Ad Hoc.
- Les nœuds des RCSFs peuvent ne pas avoir d'identificateur global à cause de leur très grand nombre.

Le déploiement dense d'un RCSF implique une proximité évidente des nœuds, favorisant la communication multi-sauts qui vise à économiser l'énergie des nœuds et de ce fait la durée globale du RCSF.

Le principe de fonctionnement des RCSFs a ouvert la voie à d'innombrables possibilités d'exploitation et d'applications. Avant de se pencher plus en détails sur les RCSFs et sur les mécanismes propres à leur fonctionnement, nous présentons dans ce qui suit quelques applications intéressantes de ces réseaux.

1.2 Applications des RCSFs

Les nœuds des réseaux de capteurs peuvent comporter différents types de capteurs, tels que les capteurs séismiques, thermiques, visuels, infrarouges, acoustiques et radar, qui sont capables de surveiller une grande variété de phénomènes ambiants, notamment [ASSC01, EE] :

- Température
- Humidité
- Mouvement des véhicules
- Pression
- différents niveaux de bruits
- Présence ou absence de certains types d'objets
- niveaux de stress mécanique sur des objets attachés, ainsi que les caractéristiques courantes d'un objet, telles la vitesse, la direction ou encore le volume de l'objet donné.

Les nœuds capteurs peuvent être utilisés pour la surveillance continue d'un phénomène, la détection d'événements, l'identification d'évènements ou encore la surveillance d'une position ou emplacement donné.

Le concept de micro-capteurs interconnectés via un support sans fil ouvre la voie vers de nombreux et nouveaux domaines d'applications qui peuvent être catégorisés en des domaines touchant au militaire, à l'environnement, la santé, les milieux ambiants (maisons) ainsi qu'à

d'autres applications commerciales. Il existe aussi d'autres catégories telles que les RCSFs destinés à l'exploration spatiale, aux traitements chimiques ou aux secours en cas de désastres [ASSC01, SMZ07, YC08, EE].

1.2.1 Applications militaires

Les réseaux de capteurs sans fil peuvent être une partie intégrante dans les systèmes militaires de *Commandes*, *Contrôle*, *Communication*, *Calcul*, *Intelligence*, *Surveillance*, *Reconnaissance* et *Ciblage* (Targeting), ou systèmes « C4ISRT ». Le déploiement rapide, l'auto-organisation et la tolérance aux pannes des RCSFs en font une solution prometteuse dans un tel domaine.

Le fait que les RCSFs soient basés sur le déploiement dense d'un grand nombre de nœuds capteurs jetables et à coût réduit, la destruction de certains nœuds n'affectera pas le déroulement d'une opération militaire, ce qui favorise l'utilisation des RCSFs sur les champs de bataille. Voici des exemples concrets d'applications militaires qui font appel aux réseaux de capteurs sans fil :

Contrôle et surveillance des forces, équipements et munitions

Les chefs de troupes et les commandants d'opérations militaires peuvent constamment surveiller l'état des troupes, les conditions et la disponibilité des équipements et des munitions en utilisant les réseaux de capteurs. Chaque troupe, véhicule, équipement ou munition critique peut porter plusieurs capteurs qui rapportent leurs états d'une manière régulière.

Ces rapports sont collectés au niveau d'un nœud puits (SINK) et envoyés aux chefs de troupes ou bien à des responsables hiérarchiquement supérieurs.

Reconnaissance et surveillance du champ de bataille

Les réseaux de capteurs, peuvent être utilisés également pour couvrir tous les terrains et emplacements critiques, les chemins et les détroits, afin de surveiller de près toutes les activités des forces ennemies. De plus, de nouveaux réseaux de capteurs peuvent être facilement déployés sur les terrains pour répondre à l'évolution des opérations et l'application des nouveaux plans.

Détection et reconnaissance des attaques nucléaires, biologiques et chimiques

Lors d'une guerre biologique ou chimique, la détection précise et instantanée d'agents chimiques pouvant être utilisés par les forces opposées est très importante. Les RCSFs jouent alors le rôle de systèmes d'alerte qui fournissant à leurs utilisateurs un temps de réaction et d'intervention critique permettant de diminuer le risque de danger induit par ce genre d'attaques.

1.2.2 Applications dans l'environnement

Parmi les applications des RCSFs dans l'environnement, on trouve :

- Pister le mouvement d'oiseaux, de petits animaux et d'insectes ;
- Surveillance et contrôle des aspects environnementaux qui peuvent affecter les récoltes et le bétail ;
- Détection chimique et biologique ;
- Détection d'incendies de forêts ;
- Travaux de recherche météorologiques et géophysiques ;
- Détection d'inondations ;
- Etudes de pollution et schématisation de la bio-complexité de l'environnement ; etc…

Détection d'incendies de forêts

Les nœuds capteurs peuvent être déployés d'une façon aléatoire et dense dans n'importe quel type de forêt, pouvant ainsi facilement détecter et rapporter l'origine de l'incendie à l'utilisateur avant qu'il ne se propage et devienne incontrôlable. Des millions de nœuds capteurs utilisant les fréquences radio ou la communication par voix optique peuvent être déployés ; ces nœuds doivent cependant pouvoir fonctionner d'une manière autonome pendant une longue durée qui peut aller jusqu'à des années, pour cela ils doivent être équipés par des systèmes de rechargement d'énergie efficaces (panneaux solaires par exemple).

Détection des inondations

Le système ALERT (Automated Local Evaluation in Real Time) est un système de détection d'inondations basé sur les RCSFs déployé aux U.S.A. Ce dernier englobe plusieurs types de capteurs hydrologiques qui détectent la pluie, le niveau d'eau, ainsi que d'autres capteurs météorologiques qui servent à la détection de la température, la pression, etc. Tous ces types de capteurs fournissent les informations nécessaires à une base de données centralisée via une communication radio ; un modèle de prévision des inondations est adopté pour analyser les données reçues et générer les éventuels avertissements.

Agriculture

Les réseaux de capteurs sont capables d'apporter des bénéfices considérables au domaine de l'agriculture, grâce à leur habilité à surveiller les taux de pesticides dans l'eau potable, le degré d'érosion du sol, le degré d'humidité et le niveau de pollution de l'air en temps réel.

1.2.3 Applications dans le domaine médical

Les réseaux de capteurs peuvent être très utiles dans le domaine médical :

- Fournir des interfaces d'aides aux handicapés ;
- Capteurs intégrés pour la surveillance de patients ;
- Diagnostique
- Administration des médicaments au sein de l'hôpital
- Surveillance des mouvements et processus internes d'insectes ou d'autres petits animaux.

Télésurveillance et contrôle de données physiologiques humaines

Les informations physiologiques collectées par un RCSF peuvent être stockées pendant une longue période, et utilisées par la suite pour des fins multiples, notamment les explorations médicales.

Les réseaux de capteurs installés peuvent également détecter et surveiller le comportement des personnes âgées et permettre une intervention rapide en cas de nécessité, ce qui donne aux sujets surveillés une meilleure liberté de mouvements et permet aux médecins traitants d'identifier plus rapidement certains symptômes prédéfinis. Ceci assure aux patients une meilleure qualité de vie comparée à celle qu'on trouve actuellement dans les centres hospitaliers.

1.2.4 Applications dans les maisons

Automatisation des maisons

Grâce aux avancée technologiques, des nœuds capteurs intelligents peuvent être intégrés dans les appareils électroménagers tel que les aspirateurs, fours micro-ondes, réfrigérateurs, et magnétoscopes. Ces nœuds capteurs peuvent interagir entre eux ainsi qu'avec les réseaux externes via Internet ou à travers les satellites pour permettre à l'utilisateur de contrôler plus aisément ces appareils d'une façon locale ou distante.

1.2.5 Autres applications commerciales

Les réseaux de capteurs possèdent également d'autres applications dans le domaine commercial telles que :

Surveillance de l'état du matériel,

- Gestion des inventaires,
- Contrôle de qualité des produits,
- Construction des espaces d'achat intelligents,
- Contrôle des robots dans les environnements de fabrications automatiques,
- Les musées interactifs,
- Diagnostique des machines,
- Détection et la surveillance des vols de voitures, etc.

Détection et surveillance des vols de voitures

Les nœuds capteurs peuvent être déployés pour identifier et détecter les menaces de vols dans une région géographique, et rapporter ces menaces à un utilisateur distant à travers Internet pour les analyser.

Gestion et contrôle de l'inventaire

Chaque article dans le magasin pourrait avoir un nœud capteur qui lui est attaché, dès lors, les utilisateurs pourront facilement localiser l'article et calculer la quantité exacte de chaque catégorie d'articles. Si des utilisateurs veulent insérer de nouveaux inventaires, ils n'auront qu'à attacher les capteurs appropriés à ces inventaires.

Il est clair que les réseaux de capteurs possèdent un potentiel d'exploitation et d'application gigantesque dans le monde réel, ce qui explique l'engouement de la communauté scientifique ainsi que l'importante activité de recherche autour de ce type de réseaux. Ces travaux de recherche ayant pour objectif l'élévation des RCSFs au rang de technologie mature et indispensable dans la majorité des domaines.

1.3 Facteurs de conception d'un RCSF

La conception d'un RCSF prend en compte différents facteurs, allant de la *tolérance aux pannes* à la *consommation d'énergie*, en passant par la *scalabilité*, le *coût de production*, la *topologie du RCSF* ainsi que les *contraintes matérielles* et *les supports de transmission.*

Bon nombre de recherches ont été effectuées sur chacun de ces facteurs, sans pour autant considérer l'ensemble de ceux-ci lors de la conception effective d'un RCSF, ce qui se répercute négativement sur le résultat obtenu. Nous exposant dans ce qui suit d'une manière non exhaustive, chacun de ces facteurs considérés comme repères lors de la conception d'algorithmes ou protocoles pour les RCSFs [ASSC01, YC08].

1.3.1 Tolérance aux pannes

La défaillance de nœuds d'un RCSF peut être causée par un épuisement des batteries d'énergie, des dommages physiques ou par des interférences liées à l'environnement, sans pour autant affecter la mission globale du RCSF, c'est ce que la tolérance aux pannes cherche à accomplir.

Cette propriété de fiabilité notée $R_k(t)$ est modélisée à l'aide d'une distribution de Poisson qui donne la probabilité de ne pas avoir de pannes dans un intervalle de temps (0, t) :

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

Où :

- λ_k est le taux de pannes d'un nœud k
- t une période de temps

Notons que chaque application des RCSFs requiert un niveau de tolérance aux pannes différent.

1.3.2 Scalabilité

Le nombre de nœuds capteurs déployés dans le but d'étudier un phénomène donné peut atteindre des valeurs importantes selon la nature de l'application. Il est donc nécessaire que les schémas développés pour les RCSFs soient capables de fonctionner avec un nombre très important de nœuds tout en tirant profit de la nature et des caractéristiques de ce type de réseaux, une très haute densité entre autre.

1.3.3 Coûts de production

Etant donné qu'un RCSF se compose de plusieurs nœuds capteurs, le coût d'un seul nœud est très important afin de justifier le coût global du réseau. Si le coût du déploiement de tels

réseaux excède celui de l'utilisation de capteurs ordinaires, faire appel aux RCSFs ne serait financièrement pas justifié. Il faut donc maintenir le coût du nœud capteur au plus bas possible.

1.3.4 Contraintes matérielles

Un nœud capteur se compose de quatre éléments de base (***figure 1.1***), à savoir :

1. Unité de captage
2. Unité de traitement
3. Unité de transmission
4. Unité de l'énergie du nœud

En fonction de la nature applicative du RCSF, d'autres unités peuvent être ajoutées aux unités de base citées précédemment, c'est le cas notamment d'unité de localisation du nœud, un générateur d'énergie ou encore un système assurant la mobilité du nœud.

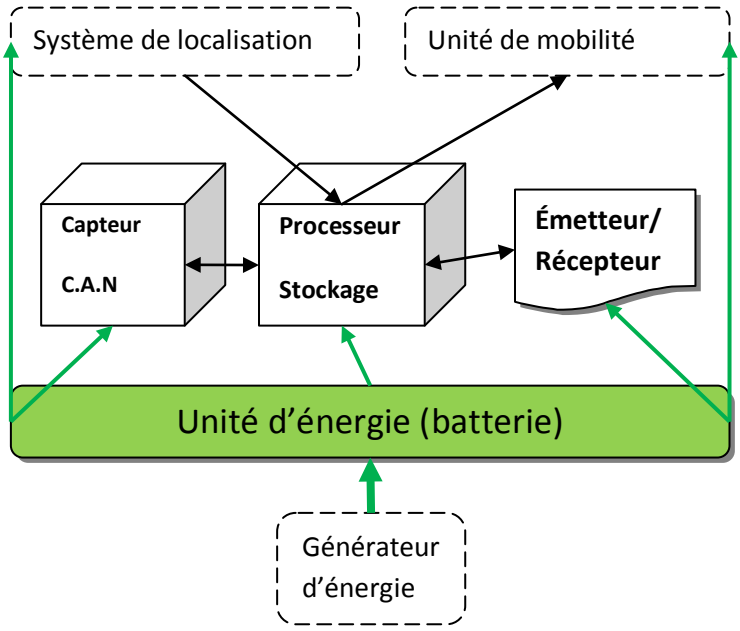

Figure 1.1 *Composants d'un nœud capteur*

L'unité de captage consiste généralement en deux sous-unités :

- un capteur qui mesure le phénomène observé produisant un signal analogique
- un Convertisseur de signaux Analogiques/Numériques

Le signal analogique obtenu par le capteur est converti en un signal numérique exploitable par l'unité de traitement.

L'unité de traitement, souvent associée à une petite unité de stockage de données, a pour mission la gestion des procédures permettant au nœud de collaborer avec les autres nœuds du RCSF pour mener à bien la mission assignée au réseau. L'unité de transmission quant à elle, connecte les nœuds au réseau. Enfin, l'unité de l'énergie assure le fonctionnement proprement dit du nœud.

Toutes ces unités doivent tenir dans un volume souvent réduit, pouvant même atteindre $1cm^3$.

En plus de cette contrainte de volume, un nœud doit :

- Consommer le moins d'énergie possible
- Fonctionner dans de grandes densités volumétriques
- Avoir un coût de production minimal
- Etre autonome, fonctionnant sans aucune assistance
- Avoir la capacité de s'adapter à l'environnement

1.3.5 Topologie du RCSF

Un grand nombre de nœuds inaccessibles et sans aucune assistance, déployés avec haute densité et exposés à des pannes imprévisibles rend la maintenance de la topologie du RCSF relativement délicate.

La mise en service d'un RCSF passe par le déploiement des nœuds constituant celui-ci, on distingue trois phases lors de cette opération :

- ***Phase de déploiement*** : les nœuds du RCSF peuvent être déployés de différentes façons selon un schéma de topologie préalablement étudié.
- ***Phase de post-déploiement*** : après le déploiement, la topologie du RCSF peut subir des changements assez fréquents causés entre autres, par des pannes, des mouvements de nœuds, etc.
- ***Phase de redéploiement*** : les changements qui peuvent affecter la topologie du RCSF lors de la phase précédente, peuvent conduire à la nécessité de redéployer un nombre de nœuds pour pallier aux pertes de nœuds dans le réseau.

1.3.6 Support de transmission

Dans un réseau de capteurs à communication multi-sauts, les nœuds sont liés via un support de communication sans fil utilisant des fréquences radio, des ondes infrarouges ou encore un support de communication optique. Cependant, il faut s'assurer de la disponibilité du moyen de transmission choisi dans l'environnement de captage, afin de permettre au réseau d'accomplir la totalité de ses tâches.

Fréquences radio : Pour les liens de communication basés sur des fréquences radio, les bandes ISM (Industrial Scientific Medical) peuvent être utilisées, ces bandes de fréquence sont employées pour assurer des communications libres de charge dans le domaine industriel, scientifique ou médical dans la plupart des pays du monde.

La table internationale d'allocation de fréquences spécifie certaines bandes de fréquences pouvant être considérées comme bandes « ISM », ces bandes sont listées dans la ***table 1.1***.

Certaines d'entre elles sont déjà utilisées dans les systèmes téléphoniques sans fil et les réseaux WLAN. 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. Certaines contraintes matérielles associés aux nœuds, ainsi que le compromis existant entre l'efficacité des antennes et la consommation d'énergie, limite le choix de la bande de fréquence utilisée sur les bandes à hautes fréquences.

L'avantage principal des bandes ISM est qu'elles sont libres de toute licence d'utilisation, offrent un large spectre de fréquences allouables et disponibles dans la plupart des pays du

monde. Ces bandes ne sont décrites par aucun standard, offrant ainsi plus de liberté pour l'implémentation des protocoles de communication spécifiques aux RCSFs. Toutefois, cette implémentation reste toujours limitée par d'autres contraintes telles que la consommation d'énergie et les interférences nuisibles aux autres applications utilisant les mêmes bandes de fréquences.

Bande de fréquences	Fréquence centrale
6765–6795 kHz	6780 kHz
13,553–13,567 kHz	13,560 kHz
26,957–27,283 kHz	27,120 kHz
40.66–40.70 MHz	40.68 MHz
433.05–434.79 MHz	433.92 MHz
902–928 MHz	915 MHz
2400–2500 MHz	2450 MHz
5725–5875 MHz	5800 MHz
24–24.25 GHz	24.125 GHz
61–61.5 GHz	61.25 GHz
122–123 GHz	122.5 GHz
244–246 GHz	245 GHz

Table 1.1 *Bandes de fréquences disponibles pour les applications ISM*

Ondes infrarouges : Les ondes infrarouges représentent un autre support possible pour la communication inter-nœuds dans un RCSF. Ce type de communication est également libre de

toute charge ou licence, il est robuste contre les interférences avec les appareils électriques, et les unités de transmission correspondantes sont moins chères sur le marché, et plus faciles à construire. Ceci peut expliquer l'existence de ports de communication infrarouge dans la plupart des ordinateurs, téléphones portables et PDAs.

L'inconvénient majeur de ce type de communication est qu'il exige la disponibilité permanente d'une ligne de vue entre l'émetteur et le récepteur, contrainte peu souple ne favorisant pas l'utilisation de la communication infrarouge dans des RCSFs.

Support optique : Deux schémas de transmission sont possibles dans ce type de RCSF utilisant un support de communication optique entre les nœuds :

- Une transmission passive avec une consommation d'énergie optimale utilisant un rétro réflecteur appelé CCR (Corner Cube Retro-reflector), ce dernier est composé de trois miroirs qui utilisent une source de lumière externe pour pouvoir communiquer avec les autres nœuds en reflétant les rayons reçus, et ceci, avec un débit pouvant atteindre plusieurs Kbps et sur une portée de communication pouvant arriver jusqu'à 150m.
- Une transmission active est employée quand l'application l'exige, via une diode laser et des miroirs orientables servant à envoyer des rayons optiques réglables au récepteur destiné ; ce mode consomme relativement plus d'énergie et doit être utilisé pour de courtes périodes.

Les contraintes liées aux applications spécifiques des RCSFs rendent le choix du support de communication une étape critique pour la conception de ces réseaux. Par exemple, les applications liées au domaine maritime peuvent favoriser l'utilisation d'un support de transmission aqueux tel que les radiations à longueur d'onde élevée pouvant pénétrer la surface de l'eau. Tandis que les applications pour les terrains hostiles, tel que les champs de batailles peuvent être confrontées à un taux élevé d'erreurs et plus de brouillage et d'interférences, le choix du support de transmission doit, dans ce cas, prévoir des schémas de modulation beaucoup plus robustes.

1.3.7 Consommation d'énergie

Comme les nœuds capteurs sont des composants micro-électroniques, ils ne peuvent être équipés que par des sources limitées d'énergie (<0.5 Ampère-heure, 1.2 V). De plus, dans certaines applications, ces nœuds ne peuvent être dotés de mécanismes de rechargement d'énergie ; par conséquent, la durée de vie d'un nœud capteur dépend fortement de la durée de vie de la batterie associée.

Sachant que les réseaux de capteurs sont basés sur la communication multi-sauts, chaque nœud joue à la fois un rôle d'initiateur de données et également de routeur. Un dysfonctionnement d'un certain nombre de nœuds entraîne un changement significatif de la topologie globale du réseau, et peut nécessiter un routage de paquets différent et une réorganisation totale du réseau. C'est pour cela que le facteur de consommation d'énergie est d'une importance capitale dans les RCSFs. La majorité des travaux de recherche menés actuellement se concentrent sur ce problème afin de concevoir des algorithmes et protocoles spécifiques à ce genre de réseaux susceptibles de consommer un minimum d'énergie. En effet, dans les réseaux ad hoc classiques, la consommation d'énergie est un facteur important mais ne constitue pas la première considération pour les concepteurs, car les batteries sont supposées toujours remplaçables par l'utilisateur (hors exécution d'applications). Les chercheurs ont cependant concentré leurs efforts sur les facteurs de qualité de service dans ce type de réseau, tel que le débit de transmission et la tolérance aux pannes.

Par contre, dans les RCSFs, l'efficacité en consommation d'énergie représente une métrique de performance significative, qui influence directement la durée de vie du réseau entier. Pour cela, les concepteurs peuvent au moment du développement de protocoles négliger les autres métriques de performance telle que la durée de transmission et le débit, au détriment du facteur de consommation d'énergie.

Phases de consommation d'énergie

Détecter les évènements dans l'environnement, élaborer un traitement de données local et rapide, et transmettre les résultats à l'utilisateur sont les principales tâches d'un nœud dans un RCSF. Les étapes de consommation d'énergie par ce nœud peuvent être, dès lors, divisées en trois phases : *captage*, *communication* et *traitement de donnée*.

Captage : L'énergie consommée au moment du captage varie suivant la nature de l'application. Un captage sporadique consomme moins d'énergie qu'un contrôle d'événement constant. La complexité de l'événement à détecter joue également un rôle crucial pour déterminer la quantité d'énergie consommée.

Les environnements contenant un niveau de bruit élevé entraîne l'augmentation de l'énergie nécessaire pour cette phase.

Communication : Parmi les trois phases citées auparavant, la phase de communication (émission/réception) est celle qui consomme la plus grande quantité d'énergie. Il est démontré que pour les communications à courte portée, avec une faible puissance de radiation (~ 0 dbm), les coûts énergétiques pour l'émission et la réception de données sont pratiquement les mêmes.

Durant cette phase, il est important de considérer l'énergie nécessaire pour la mise en marche du circuit de communication, le temps de démarrage étant égal à plusieurs centaines de microsecondes rend l'énergie consommée durant cette période non négligeable. L'influence de cette étape sur la quantité globale d'énergie consommée par la communication augmente quand la taille des paquets transmis diminue.

Par conséquent, une bonne politique de consommation d'énergie passe obligatoirement par éviter au maximum le recours à la mise en marche et l'arrêt fréquents des circuits de communication.

Traitement de données : Comparée à la phase de communication, l'étape de traitement local des données consomme beaucoup moins d'énergie. En effet, le coût énergétique nécessaire pour transmettre 1 KB sur une portée de 100 m est approximativement égal à celui nécessaire pour exécuter 3 millions d'instructions à une vitesse de 100 millions instructions par seconde. Ce fait favorise largement le traitement local des données pour l'amélioration de la consommation d'énergie dans les réseaux de capteurs. Les nœuds capteurs doivent donc posséder des moyens de traitement local de données, tout en restant capable d'interagir avec les nœuds avoisinants.

Enfin il est à noter qu'un nœud peut contenir des circuits additionnels pour le codage/décodage des données, en plus de certain circuits spécifiques aux applications du

réseau. Dans tous ces cas, la conception d'algorithmes et protocoles du réseau est largement influencée par l'énergie consommée par ces circuits en plus de ceux évoqués précédemment.

1.4 Architecture d'un RCSF

Un RCSF est composé d'un grand nombre (des centaines voire des milliers) de nœuds (capteurs) densément déployés dans une région afin d'y surveiller les activités de son environnement et de collecter des informations pour d'éventuels traitements. Ces informations sont acheminées vers l'utilisateur du RCSF (ou observateur) en passant par un nœud appelé *station de base* ou encore SINK, ayant une bien meilleure portée de communication comparée aux autres nœuds du RCSF. L'organisation d'un RCSF peut être perçue comme suit (***Figure 1.2***) :

- **L'infrastructure:** il s'agit des nœuds capteurs et de leur actuel état de déploiement. plus précisément, cette infrastructure est influencée par les caractéristiques physiques des capteurs (capacité de capture et de calcul, taille de la mémoire, les caractéristiques de la batterie, et le mode de transmission...) et la stratégie de déploiement (la densité des capteurs, leur localisation, leur mobilité…).
- **les protocoles du réseau:** responsables de la création et du maintien de chemins de communication entre les capteurs et les observateurs.
- **Application/Observateur:** les données requise par l'observateur(s), d'un phénomène donné, sont collectées par les différents nœuds du RCSF et acheminé vers l'utilisateur. les requêtes de l'observateur peuvent être statiques (les capteurs sont préprogrammés de manière à collecter les données selon un modèle spécifique) ou dynamiques.

Notons que les nœuds capteurs peuvent participer à la synthèse et au traitement des données collectées (filtrage, fusion..).

Ainsi, un RCSF repose sur deux types de nœuds qui sont les acteurs importants du réseau :

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

- **SINK** : un nœud particulier du réseau. Il est chargé de la collecte des données issues des différents nœuds du réseau. Il doit être toujours actif puisque l'arrivée des informations est aléatoire. C'est pourquoi, idéalement, son énergie doit être illimitée. Dans un réseau de capteur sans fils plus ou moins large et à charge élevée, plusieurs SINK peuvent être déployés pour alléger la charge.

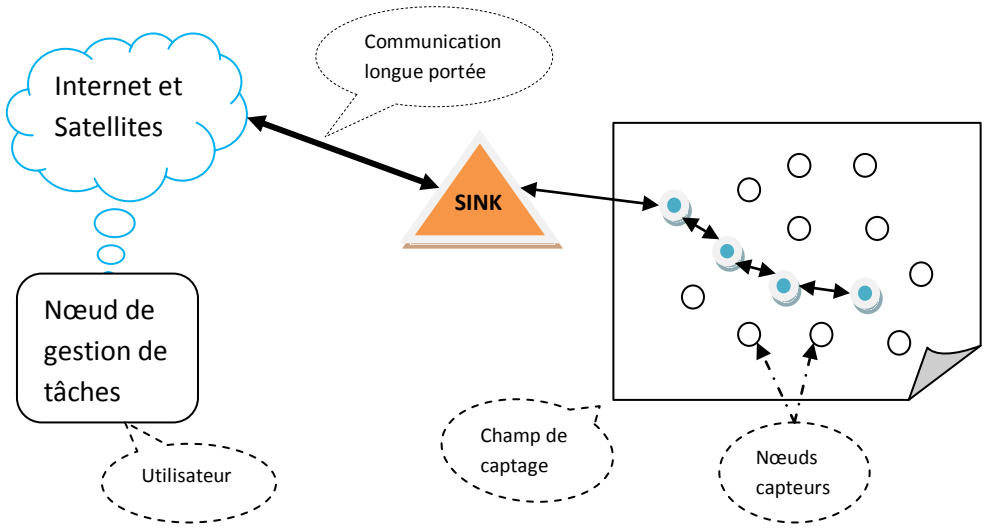

Figure 1.2 *Organisation et architecture globale d'un RCSF*

Il existe essentiellement trois types de SINK selon la nature de l'application et du RCSF [HW05] :

- Un nœud appartenant au réseau comme n'importe quel autre nœud.
- Une entité extérieure au réseau. Pour ce deuxième cas, le SINK peut être un dispositif extérieur, par exemple, un ordinateur portatif ou un PDA interagissant avec le réseau.
- Une passerelle vers un autre réseau tel qu'Internet, où la demande de l'information vient d'un centre de traitement distant.

1.4.1 Architecture de la communication dans un RCSF

La communication dans un RCSF entre les différents nœuds est basée sur la pile protocolaire de la ***figure 1.3***, constituée d'une couche *applicative*, couche de *transport*, couche *réseau*, une couche de *liaison de données* ainsi qu'une couche *physique* ; toutes ces couches projetées sur 3 plans, un plan de gestion de l'énergie, un plan de gestion de la mobilité ainsi qu'un plan de gestion des tâches [ASSC01].

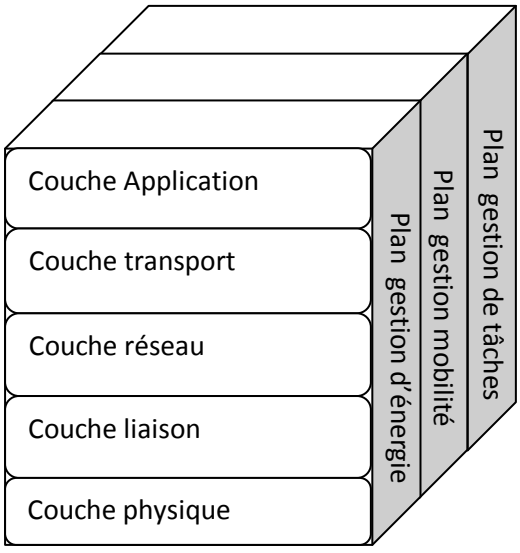

Figure 1.3 *Pile protocolaire des RCSF*

Différents types d'applications peuvent être construites selon la tâche assignée aux capteurs. La couche transport aide au maintien du flux de données si l'application du RCSF le nécessite. La couche réseau quant à elle, est responsable du routage des données fournies par la couche transport dans le RCSF. La couche liaison contenant le protocole MAC doit tenir compte de l'énergie et essayer de minimiser les collisions entre les diffusions voisines, dans un environnement contenant du bruit et l'éventualité d'une certaine mobilité des nœuds du RCSF. La couche physique doit permettre une transmission et une modulation des signaux à la fois simple et robuste.

Les fonctions intégrées au plan de gestion de l'énergie permettent de gérer l'énergie consommée par les capteurs, dès lors, un capteur peut par exemple éteindre son interface de réception dès qu'il reçoit un message d'un nœud voisin afin d'éviter la réception des messages dupliqués. De plus, quand un nœud possède un niveau d'énergie faible, il peut diffuser un message aux autres capteurs pour ne pas participer aux tâches de routage, et conserver l'énergie restante aux fonctionnalités de captage.

La gestion de la mobilité, détecte et enregistre tous les mouvements des nœuds capteurs, de manière à leur permettre de garder continuellement une route vers l'utilisateur final, et maintenir une image récente sur les nœuds voisins ; cette image est nécessaire pour pouvoir équilibrer l'exécution des tâches et la consommation d'énergie.

Lors d'une opération de captage dans une région donnée, les nœuds composant le réseau ne doivent obligatoirement pas travailler avec le même rythme, cela dépend essentiellement de la nature du capteur, son niveau d'énergie et la région dans laquelle il a été déployé. Pour cela, les fonctions du plan de gestion des tâches assurent l'équilibrage et la distribution des tâches sur les différents nœuds du réseau, afin d'avoir un travail coopératif et efficace en matière de consommation d'énergie, et par conséquent, prolonger la durée de vie du RCSF.

La communication au sein du RCSF est achevée grâce à cette pile protocolaire et à la coopération inter-nœuds, en tenant compte du niveau d'énergie du RCSF afin d'en prolonger la durée de vie.

1.4.2 Paradigmes de communication dans les RCSFs

Il existe différents niveaux de complexité dans les scénarios d'acheminement de données dans les RCSFs, allant de la simple communication source/destination, à la communication maillée impliquant plusieurs sources et destinations. Cet acheminement est effectué en assurant des tâches d'agrégation et en respectant des contraintes de qualité de service. C'est pourquoi, l'acheminement et l'agrégation des données (routage, requêtes, découverte ou exploration) doivent être adaptés à cette différence de complexité afin d'éviter une probable perte d'efficacité liée à l'aspect spécifique des applications dans la plupart des RCSF [DN,YC08].

Le **routage** des paquets dans un RCSF est d'une importance capitale, influant d'une façon directe sur la durée de vie du réseau entre autres, c'est pourquoi plusieurs protocoles de routage ont été proposés et étudiés dans la littérature afin d'aboutir à des solutions efficaces et spécifiques aux RCSFs.

La logique des **requêtes (**interrogations**)** émises dans un RCSF est très similaire à celle qu'on trouve dans les bases de données ; le flux de données généré par les capteurs du réseau peut être perçu comme étant une table interrogeable par de requêtes spatiales et temporelles, à l'aide de l'interface familière des bases de données classiques adaptée toutefois aux besoins spécifiques des RCSF. D'un autre coté, la liaison étroite entre l'émission de requêtes et le routage mènent à un schéma uniforme d'acheminement de données dans un RCSF.

La **découverte** ou l'**exploration** est un service important au sein d'un RCSF directement lié à l'autonomie de celui-ci ; les nœuds se découvrent mutuellement sans intervention extérieur tandis que l'administrateur du RCSF doit avoir connaissance de la topologie actuelle du réseau afin d'en vérifier la couverture. Les solutions mettant en œuvre ces opérations d'exploration doivent tenir en compte plus particulièrement des contraintes énergétiques.

Ces trois aspects d'acheminement des données dans un RCSF définissent globalement les différents paradigmes de communication que nous exposons dans ce qui suit :

Node-Centric

Ce paradigme est celui employé dans les réseaux conventionnels, où les communications se basent sur l'identification des nœuds participants, qui se fait à l'aide d'adresses IP. Un des avantages d'un tel schéma d'organisation des nœuds, est la transparence du routage aux couches supérieures. Cependant, l'utilisation d'un tel adressage dans les RCSF est loin d'être une solution efficace, dû en grande partie aux cycles ON/OFF des nœuds ; de plus, l'architecture en couches des réseaux classiques, bien qu'indépendante du Node-Centric, se voit incompatible avec les RCSFs qui dépendent fortement de la nature de l'application. Il est donc clair que la philosophie d'adoption de couches séparées visant à normaliser l'architecture des réseaux classiques ne convient pas à des réseaux orientés application tels que les RCSFs qui pourraient être pénalisés en termes de performances. Des adaptations et modifications sont alors nécessaires.

Data-Centric

Dans un RCSF, la donnée est plus importante que le nœud lui même, ce qui rend son identification inutile. Dans le paradigme Data-Centric, les communicants sont identifiés par leurs données, et donc tout le système (routage, interrogation, etc.) doit être régi par cette propriété. Ainsi, le système peut être vu comme une base de données distribuée, où les nœuds forment des tables virtuelles, alimentées par les données captées. L'approche Data-Centric permet donc de répondre à des requêtes du type *«demande de données satisfaisant certaines conditions »*. L'importance de la donnée captée par rapport à l'identité des nœuds influe sur l'implémentation des différents aspects du paradigme (routage, exploration et interrogation) dans lesquelles l'identité des nœuds n'est jamais impliquée.

Position-Centric

Dans cette approche, les positions des nœuds représentent le moyen principal d'adressage et de routage. Dans certaines applications, il est plus intéressant d'interroger le système en utilisant les positions des nœuds, que leurs adresses IP. Dans ce cas, le routage s'effectue grâce à des techniques géométriques afin d'acheminer l'information d'une zone géographique vers une autre. Cette approche vient du simple fait que les données captées sont souvent requises avec la position de celles-ci pour plus de signification et d'intérêt.

1.4.3 Classification des protocoles de routage dans les RCSF

Après avoir vu les principaux paradigmes de communication dans les RCSFs, il est intéressant de passez en revue la classification des différents protocoles de routage et d'acheminement de données dans les RCSF (***Figure 1.4***), une classification qui se base sur le paradigme de communication entre autres :

Selon la Topologie du réseau

La topologie détermine l'organisation des nœuds capteurs dans le réseau. Il existe deux principales topologies dans les protocoles de routage pour les RCSFs.

- Topologie plate : dans une topologie plate, tous les nœuds possèdent le même rôle. Les nœuds sont semblables en termes de ressources.
- Topologie hiérarchique : afin d'augmenter la scalabilité du système, les topologies hiérarchiques ont été introduites en divisant les nœuds en plusieurs niveaux de responsabilité. L'une des méthodes les plus employées est le clustering, où le réseau est partitionné en groupes appelés "clusters". Un cluster est constitué d'un chef (cluster-head) et de ses membres.

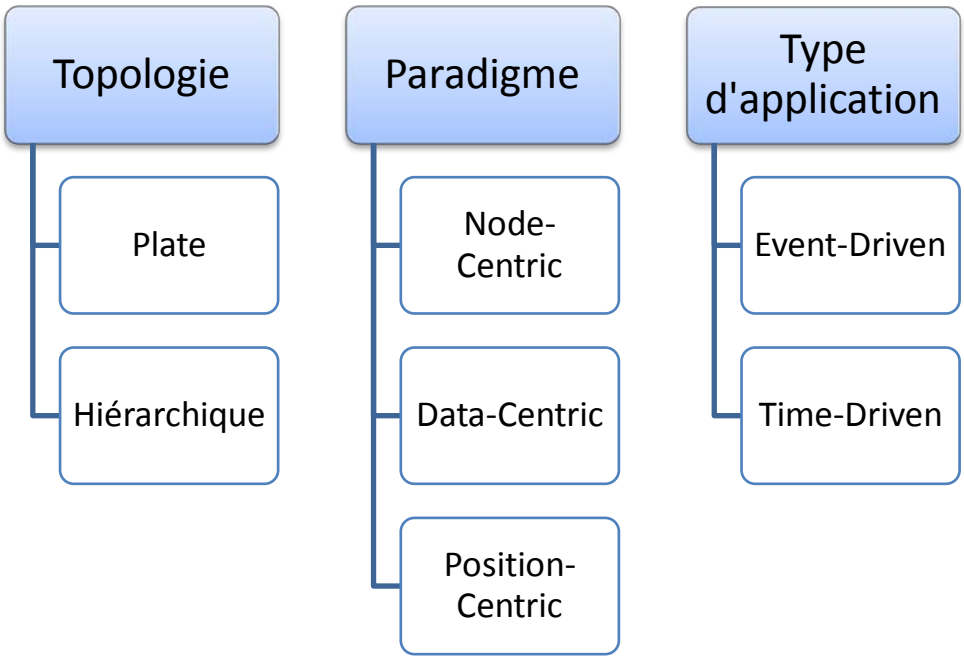

Figure 1.4 *classification des protocoles de routage dans les RCSF*

Selon le **Paradigme de communication**

Comme vu précédemment, il existe trois principaux paradigmes de communication dans les RCSFs :

- Node-Centric
- Data-Centric
- Position-Centric

Selon le Type d'application

La méthode de captage des données dans un RCSF dépend de l'application et de l'importance de la donnée. De ce fait, les RCSFs peuvent être catégorisés comme time-driven ou event-driven.

- Application time-driven : un réseau time-driven est approprié pour des applications qui nécessitent un prélèvement périodique des données. Par exemple, cela est utile dans des applications de monitoring (feu, météo) afin d'établir des rapports périodiques.
- Application event-driven : dans des applications temps réel, les capteurs doivent réagir immédiatement à des changements soudains des valeurs captées. Un prélèvement périodique des données est inadapté pour ce type de scénarii. Pour cela, le protocole doit être réactif et doit donner des réponses rapides à l'occurrence d'un certain nombre d'évènements.

1.5 Conclusion

Plusieurs généralités sur les RCSFs ont été exposées dans ce chapitre, en commençant par évoquer la légitime ressemblance avec les réseaux Ad Hoc ainsi que les principales différences. Nous avons aussi vu que l'un des aspects primordiaux de la conception des RCSFs était bien la tolérance aux pannes, le chapitre qui suit s'y intéresse plus particulièrement en essayant de montrer l'importance de celle-ci au sein d'un RCSF.

CHAPITRE 2 :

Tolérance aux pannes dans les RCSFs

Les réseaux de capteurs sans fil (RCSF) sont souvent déployés dans des environnements hostiles du fait de la nature de leurs applications réelles, ce qui conduit inévitablement à une augmentation significative de la probabilité de leurs altérations plus ou moins irréversibles durant l'accomplissement de leurs tâches ; ce qui pourrait conduire à une baisse considérable des performances du RCSF ou voire dans le pire des cas, à une panne générale du réseau : synonyme d'échec total de la mission qui lui était assignée. Un échec qui ne serait absolument pas tolérable en cas d'applications critiques.

Ces faits soulignent l'importance capitale et la nécessité d'avoir une politique de tolérance aux pannes au sein d'un RCSF capable d'assurer sa survie à d'éventuelles difficultés.

2.1 Importance de la tolérance aux pannes dans les RCSF

L'importance d'une stratégie de tolérance aux pannes nous permet de distinguer trois sortes de RCSF où celle-ci (tolérance aux pannes) est primordiale :

- ✓ RCSF **critiques**
- ✓ RCSF à environnement **hautement hostile**
- ✓ RCSF **critique** à environnement **hautement hostile**

2.1.1 RCSF critiques

Cette catégorie englobe tous les RCSFs qui sont déployés pour des applications de nature critique ne tolérant aucune interruption momentanée ou permanente du système dans des environnements peu, voir non hostile. Un arrêt de ce genre de RCSF peut avoir des conséquences souvent désastreuses, d'où la nécessité absolue d'avoir une très grande tolérance aux pannes. Nous pouvant citer comme exemple un RCSF pour la surveillance de l'état des patients dans un hôpital ou même chez eux.

2.1.2 RCSF à environnement hautement hostile

Le déploiement de ce genre de RCSF se fait dans des environnements présentant les particularités d'être instables et très risqués pour les nœuds qui composent ces RCSFs, ceci accroit considérablement la probabilité de pannes et de dysfonctionnements dus aux changements imprévisibles de l'environnement, avec souvent l'impossibilité d'interventions post-déploiement. La tolérance aux pannes permet donc la survie du RCSF et de la mission assignée à celui-ci dans de telles situations. Un exemple serait le déploiement d'un RCSF dans une forêt, au pôle nord ou dans un volcan pour la récolte de données scientifiques aidant à la compréhension d'un phénomène donné.

2.1.3 RCSF critiques à environnement hautement hostile

Il s'agit d'une combinaison des deux sortes de RCSF déjà évoqués ci-dessus, là aussi, la tolérance aux pannes s'avère être d'une importance capitale et ne doit plus que jamais être négligée. On peut trouver par exemple ce type de RCSF dans des applications de nature militaire comme la détection de cible et la surveillance dans un champ de bataille, ou encore la détection d'un feu de forêt.

Les genres de RCSF cités précédemment mettent en avant l'importance de la présence d'une tolérance aux pannes au même titre que les autres critères nécessaires à la conception d'un RCSF robuste et efficace (sécurité, durée de vie et consommation d'énergie, routage…etc.).

2.2 Généralités sur la tolérance aux pannes

Nous abordons dans ce qui suit, quelques notions liées à la tolérance aux pannes dans les réseaux en général et plus particulièrement dans les RCSFs.

Définition de la tolérance aux pannes dans un RCSF :

Capacité d'assurer la continuité des fonctionnalités du RCSF sans aucune interruption et sans affecter la tâche globale du réseau, malgré d'éventuelles pannes de nœuds du RSCF. [ASSC01]

Panne, erreur et faute :

Une ***panne*** ou ***défaillance*** est la déviation du système de son comportement spécifié, causé par un état non valide et incohérent du système ou ***erreur***, cette dernière concerne donc une situation (état) non tolérée par les spécifications du système. Une ***faute*** ou défaut (dans le système) est à l'origine de l'***erreur*** tant celle-ci représente la manifestation et le symptôme de la ***faute***. La ***faute*** est donc l'origine et la cause première de la ***panne***. [Tan02]

2.2.1 Classification des pannes

Différentes pannes peuvent se produire dans un RCSF, un déplacement d'un nœud peut causer une panne de lien, l'épuisement de la batterie conduit à une panne d'énergie synonyme d'arrêt du nœud, ou bien le nœud se met à envoyer des valeurs aléatoires du à une défaillance ou voire à une corruption du nœud en question [Tan02].

Il existe donc différents types de pannes qui peuvent se produire dans un système, un RCSF en l'occurrence, selon des critères donnés. Dans ce qui suit, une classification des pannes selon deux critères est exposée :

- Pannes selon la durée
- Pannes selon le comportement résultant

Pannes selon la durée

Cette catégorie englobe les types de pannes suivants :

Pannes transitoires : pannes qui éventuellement disparaissent après un temps très bref, sans une intervention extérieure.

Pannes permanentes : pannes qui ne disparaissent pas après un temps et qui nécessitent une intervention extérieure.

Pannes selon le comportement résultant

Crashs : panne qui se manifeste par un arrêt du fonctionnement du composant ou l'incapacité de celui-ci de retourner vers un état valide. Ceci se traduit par le fait que le composant ne fournit plus le service demandé.

Omission : le composant fournissant le service ne répond sporadiquement plus aux requêtes, ceci peut êtres par exemple causé par des interférences radio conduisant à une perte occasionnelle de messages. Il peut y avoir un degré d'omission défini qui détermine le nombre d'omissions nécessaires avant que la panne ne soit considérée comme crash et non plus comme omission.

Timing : le composant ne fournit plus son service dans les temps (résultat en retard ou trop tôt). Ceci se traduit par le fait que le composant fournit une valeur correcte, mais en dehors du délai spécifié par l'application ; ce genre de pannes est à considérer seulement si un délai dans la réception des résultats a déjà été défini dans l'application.

Valeur : le composant fournit un résultat erroné ou qui manque de précision. Un service (nœud) d'agrégation des données générées par les autres nœuds dans le réseau peut envoyer un résultat qui ne reflète pas les données acquises en entrée, ceci peut être causé par des défaillances logicielles et/ou matérielle, des messages corrompus ou bien par des nœuds malicieux générant des valeurs incorrectes.

Aléatoires ou Byzantines : des pannes d'une nature aléatoire, les plus difficiles à traiter.

2.3 Sources de pannes dans les RCSFs

Les RCSF sont souvent déployés dans des environnements hostiles et sont sujets à différentes pannes à tous les niveaux. La ***figure* 2.1** qui suit montre une classification par couche des composants du RCSF qui peuvent subir des pannes et met l'accent sur la notion de

propagation de la panne à partir des couches inférieures vers les couches supérieures [SVB, MCP]. En effet, une panne d'énergie d'un nœud se répercute sur l'acheminement des données qui transite par ce nœud défaillant et qui peut conduire à l'isolement complet d'une partie du réseau.

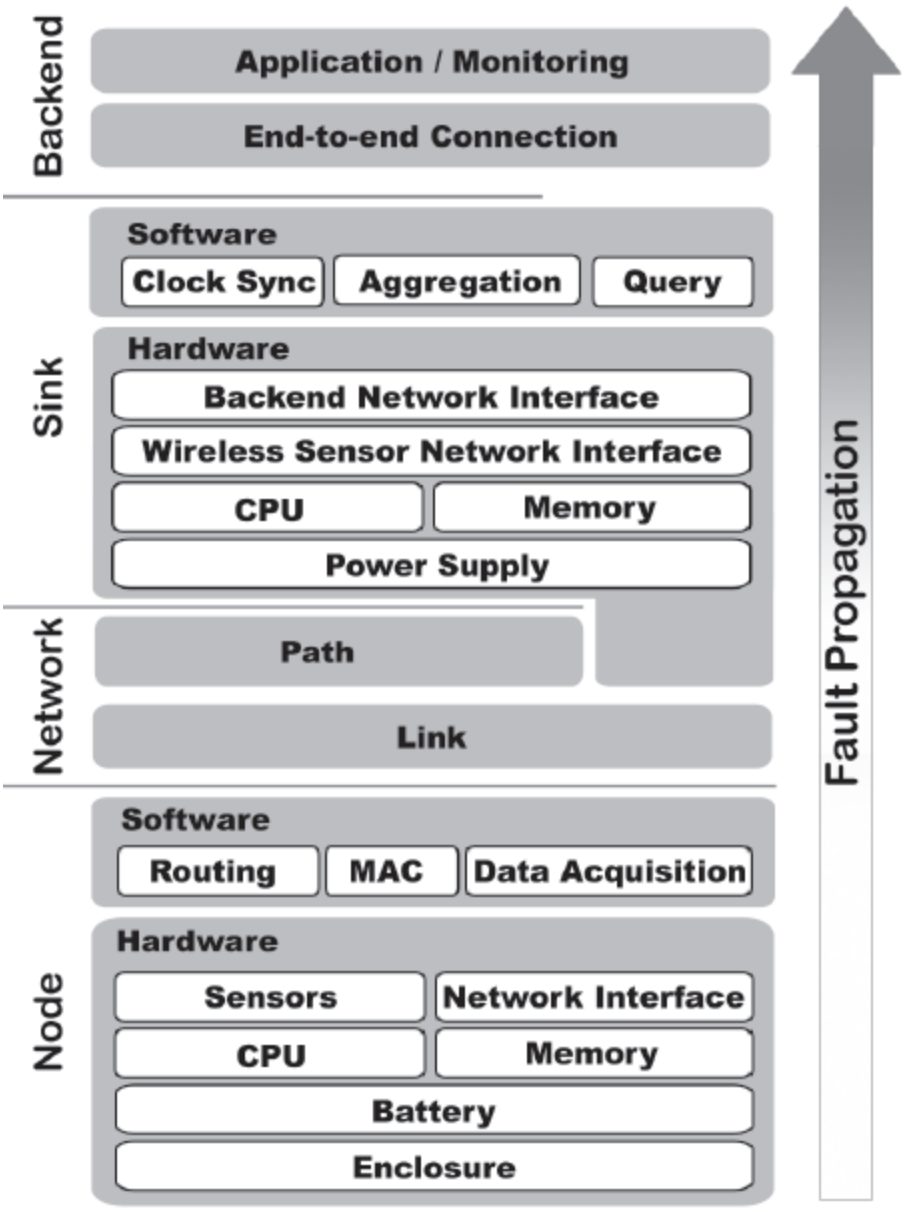

Figure 2.1 *Classification de pannes et propagation*

2.3.1 Pannes de nœuds

Plusieurs parties logicielles et matérielles composant un nœud peuvent produire des pannes pour différentes causes, comme par exemple :

- ✓ les impacts physiques qui peuvent endommager la protection extérieure du nœud qui rend du coup les composants internes du nœud vulnérables à l'environnement.
- ✓ le contact de liquides (l'eau par exemple) qui produit des courts circuits endommageant les circuits électroniques du nœud.
- ✓ le niveau de la batterie du nœud qui atteint un certain seuil (bas) peut conduire les capteurs du nœud en question à produire des résultats erronés.
- ✓ Batterie à plat est synonyme d'arrêt total du nœud
- ✓ Crash du processeur du nœud
- ✓ Bugs logiciels du nœud
- ✓ Capture du nœud et/ou corruption de celui-ci

Généralement, une panne matérielle conduit toujours à une panne logicielle, mais celle-ci peut ne pas affecter tous les services du nœud, ce dernier est toujours capable d'acheminer des messages sur le réseau même si l'un ou plusieurs de ces capteurs sont défaillants. Par contre, un ou plusieurs nœuds qui envoient des mesures erronées à la station de base affectent tout le système qui se voit traiter des valeurs ne reflétant absolument pas la réalité.

2.3.2 Pannes de réseaux

Le routage est l'un des plus importants aspects d'un RCSF, essentiel pour collecter les données mesurées, la configuration du réseau et pour la coordination inter-nœuds ; de plus, quelques applications sont orientées routage comme par exemple les applications de poursuite d'objets en mouvement. Une panne au niveau du routage peut donc conduire à une perte de messages et à des temps de réponse inacceptables.

Parmi les causes possibles de pannes de ce type :

- ✓ Instabilité des liens de communication
- ✓ Mobilité des nœuds qui conduit à leurs isolements.
- ✓ Interférences radio susceptibles d'être présentes dans l'environnement du RCSF
- ✓ Collisions inter-messages
- ✓ Présence de bugs dans la fonction de routage
- ✓ Attaques malicieuses contre le RCSF.

Notons qu'il est clair qu'une panne d'un nœud important entrant dans le mécanisme de routage conduira à une panne au niveau du réseau comme le suggère la notion de propagation mentionnée dans la ***Figure 2.1.***

2.3.3 Pannes du Sink

Le Sink est le nœud qui a pour mission de collecter toutes les données générées par le RCSF et de les acheminer à l'utilisateur final. Il peut donc être sujet à d'éventuelles pannes qui peuvent avoir des conséquences désastreuses sur toute l'application en l'absence d'une stratégie de tolérance aux pannes pour le Sink. En d'autres termes, une panne du Sink ou de l'un de ses composants conduit à une panne de tout le RCSF et à la perte de toutes les données collectées.

Parmi les causes possibles de pannes du Sink, en plus des causes citées dans les pannes de nœud étant donné que le Sink est un nœud spécial à la base, on peut y inclure :

- ✓ Déploiement du Sink dans des environnements sans aucune source d'énergie permanente, conduisant à son arrêt dû à l'épuisement des batteries ou d'une défaillance dans le système de rechargement de ces batteries (panneaux solaires par exemple).
- ✓ Environnement instable du Sink peut conduire à l'isolement de celui-ci par rapport à l'utilisateur final (ou bien par rapport au réseau de capteurs), une communication satellite entre le Sink et l'administrateur du système est, par exemple, impossible durant de fortes tempêtes
- ✓ Défaillance logicielle (bug) du processus responsable du stockage, traitement et de l'envoi des données collectées du Sink vers l'administrateur du système

La constatation de l'importance capitale du Sink dans un RCSF motive l'idée de nécessité d'avoir une stratégie efficace de tolérance aux pannes afin de pallier à d'éventuels problèmes touchant ce nœud primordial.

2.4 Gestion des pannes dans les RCSFs

Dans le cas d'une situation de panne et selon un schéma de gestion des pannes approprié, trois principales procédures doivent être effectuées [YMM] :

2.4.1 Détection de panne

Il s'agit de la première étape par laquelle le système du RCSF doit passer dans toute gestion de panne. Celle-ci doit être parfaitement identifiée afin de prendre les contre-mesures nécessaires.

Il existe deux types d'approches de détection de pannes dans les RCSF :

- ✓ Approche centralisée
- ✓ Approche distribuée

Approche centralisée

Il s'agit d'une solution courante pour identifier et localiser la cause des pannes ou les nœuds suspects dans un RCSF. Un nœud centralisé (d'un point de vue logique ou géographique) est habituellement responsable de la surveillance et l'identification des disfonctionnements et des pannes pouvant affectées les nœuds du RCSF. La plupart de ces approches suppose que le nœud central possède des ressources illimitées (comme l'énergie) et qu'il est capable d'assurer l'exécution d'une grande partie des procédures de gestion des pannes, mais aussi, que la durée de vie du RCSF peut être significativement améliorée si les taches complexes du réseau et les transmissions de messages sont confiées au nœud central.

Le nœud central adopte généralement un modèle de détection active pour obtenir l'état des performances du réseau et l'état des nœuds capteurs, ce model consiste à injecter périodiquement des requêtes dans le réseau et analyser les résultats de celles-ci afin de détecter et de localiser d'éventuelles pannes ou disfonctionnement des nœuds du RCSF.

L'approche centralisée est relativement précise et efficace pour identifier les pannes dans les RCSFs, mais présente toutefois des inconvénients dus au fait que ce genre de réseaux peut avoir de sérieuses contraintes de ressources, qui ne permettent pas toujours un prélèvement périodique des données mesurées par les capteurs et de leurs états d'une façon centralisée. Un autre inconvénient est le très grand trafic de données générées par ce type d'approche au niveau du nœud central qui représente un point unique de circulation étant donné qu'il est responsable de toutes les opérations de gestion des pannes, ce qui peut conduire à l'épuisement rapide de l'énergie dans certaines régions du réseau. Cette approche s'avère donc peu efficace et très coûteuse dès qu'il s'agisse d'un RCSF dense.

Approche distribuée

Cette approche privilégie la prise de décision locale au niveau de chaque nœud qui permet la distribution de la gestion des pannes sur tout le RCSF ; le nœud communique avec le nœud central seulement dans le cas de panne avérée, il en résulte alors moins de trafic dans le réseau.

Il existe plusieurs solutions développées dans ce sens :

Dans [HRR05] par exemple, un accéléromètre intégré dans le nœud permet de détecter si celui-ci a souffert d'impacts pouvant mener à des défaillances matérielles. Les nœuds peuvent analyser des informations relatives à l'épuisement de leurs batteries et prédire ainsi des pannes d'énergie. D'autres détections de pannes sont basées sur la coordination du voisinage, l'utilisation d'un chien de garde pour détecter des disfonctionnements de l'entourage ou encore des solutions basées sur l'utilisation de clusters pour répartir la détection de pannes sur le RCSF.

2.4.2 Diagnostic de panne

Cette étape permet d'identifier correctement les causes de la panne et de faire une distinction entre les vraies pannes et les fausses alertes de pannes. Il s'agit donc de déterminer la précision et l'exactitude de la détection afin de prendre éventuellement les mesures adéquates.

2.4.3 Recouvrement de panne

Il s'agit de la phase où le RCSF est restructuré et reconfiguré afin de limiter et de contenir les effets de la panne pour que celle-ci n'affecte pas les performances du réseau. La plupart des approches consistent à isoler les nœuds défaillant directement au niveau du routage, comme par exemple, sélectionner un nouveau voisin pour router les paquets si le nœud détecte que l'ancien voisin par qui les données transitaient est silencieux, ce genre de solution vient du faite de la redondance des nœuds dans le RCSF et peut être donc vue comme un mécanisme de réplication. Il existe deux types d'approches de recouvrement de pannes basées sur la réplication (redondance) pour les RCSF :

Redondance active

Toutes les requêtes du RCSF sont traitées par tous les composants redondants, un exemple naturel est la présence de plusieurs nœuds identiques dans le réseau ce qui permet d'obtenir les mesures d'un phénomène malgré la panne d'un ou plusieurs nœuds. Parmi les approches possible afin d'obtenir ce genre de redondance, on trouve le routage multi-routes qui permet d'avoir plusieurs routes possibles pour acheminer un message.

Redondance passive

Seule le composant original traite les requêtes, il faut maintenir donc une certaine cohérence entre les composants de secours et celui-ci en transférant l'état et les informations sur les requêtes du composant original vers les composants de secours d'une façon périodique ; ceci en tenant compte des contraintes strictes sur les ressources des RCSF. Il faut de ce fait minimiser l'impact de cet échange d'informations sur la performance du réseau en réduisant, voire en éliminant, la taille des informations sur l'état du composant. Un

recouvrement de panne dans un système de redondance passive dans un RCSF passe par trois étapes :

1. Détection de panne
2. Sélection du composant (nœud) relais
3. Distribution du service

La première étape a été vue précédemment, vient donc la deuxième étape dans laquelle une panne du nœud original est avérée ; il faut procéder dans ce cas à la sélection du nœud qui va prendre le relais. Il existe différentes approches de sélection :

Auto-sélection : chaque nœud exécute un algorithme qui lui permet de déterminer son rôle, les nœuds du RCSF vont donc constamment changer de rôle.

Groupe-sélection : la sélection du nœud de secours se fait par un consensus entre un groupe de nœud.

Sélection hiérarchique : la sélection du nœud de secours se fait par un coordinateur qui possède souvent une vue globale du réseau ou de la partie concernée du réseau.

Après avoir sélectionné le nœud de secours qui prendra le relais, celui-ci doit commencer à fournir le service en question, ceci peut être obtenu par une simple activation quand le service est déjà disponible sur le nœud, mais dans le cas où le service est indisponible sur le nœud faute de capacité de stockage, d'autres techniques sont sollicitées :

Pré-copie **:** le code de tous les services est disponible au niveau de chaque nœud avant le déploiement, ce qui permet à chaque nœud de changer de comportement en fonction de son rôle dans le RCSF.

Distribution du code : il s'agit d'une distribution (**migration ou envoi**) du **code** à travers le RCSF, utilisant des agents mobiles par exemple.

Exécution distante : vise à réduire le trafic généré par l'approche précédente, elle consiste en l'envoi des paramètres nécessaires à l'exécution vers le nœud qui possède le code, et à recevoir le résultat de cette exécution en retour.

Une approche dite hybride entre les deux dernières techniques consiste à copier le code d'un nœud dont le niveau d'énergie de la batterie atteint un seuil défini vers un autre nœud.

2.5 Exemple d'un RCSF tolérant aux pannes

Le problème de fusion dans un réseau de capteurs multimodal tolérant aux pannes utilisant des capteurs numériques binaires peut être modélisé par l'exemple illustré dans la ***Figure 2.2*** suivante. On considère un réseau de capteurs pour la reconnaissance de personnes déployé dans une société pour identifier ses employés. Six personnes nommées A, B, C, D, E et F travaillent dans cette société [KPS]. Le système de reconnaissance utilise deux types différents de capteurs :

1. capteur de taille (grandeur) ;
2. capteur pour la reconnaissance vocale qui demande à chaque entrant d'introduire une phrase secrète donnée à l'aide d'un microphone.

La figure ci-dessous montre les six personnes ainsi que leurs caractéristiques (taille et voix) représentées dans le graphe.

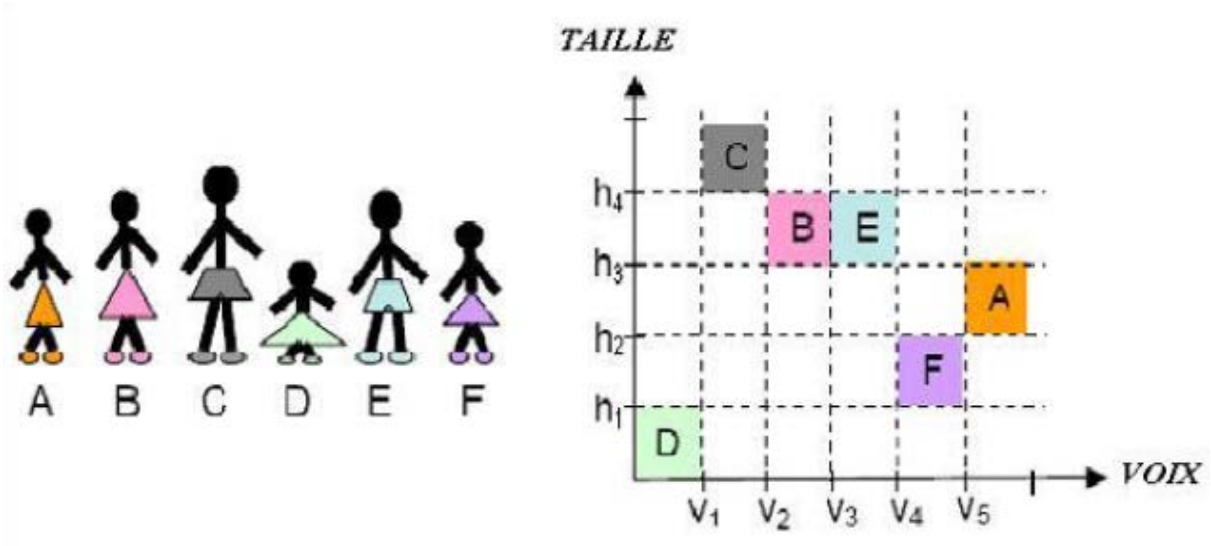

Figure 2.2 *Réseau de capteur multimodal*

Il est évident de constater que le système peut distinguer entre deux personnes P1 et P2 si elles sont représentées dans deux surfaces différentes sur le graphe. Selon l'exemple, si tous les capteurs fonctionnent correctement, chaque personne va occuper une surface différente.

En outre, dans la plupart des cas, et malgré la défaillance de l'un des capteurs de taille ou de voix, la reconnaissance de toutes les personnes est encore possible. Ceci grâce à la tolérance aux pannes hétérogène où le capteur en panne d'un certain type peut être remplacé par la fonctionnalité d'un capteur de l'autre type. Cependant, pour le cas des personnes B et E, qui ont la même taille, la voix est le seul critère pour les distinguer ; d'où, le système ne devrait avoir aucune tolérance aux pannes pour le capteur V3 qui distingue entre B et E. Si on exclut l'un de B ou E du personnel de la société, alors le système sera complètement tolérant aux pannes.

2.6 Classification des solutions de tolérance aux pannes dans les RCSF

Les protocoles tolérants aux pannes peuvent être vus de plusieurs angles différents. De ce fait, un ensemble de critères est défini pour les classifier. Deux catégories de classification sont exposées :

2.6.1 Classification selon la nature de l'algorithme/protocole

Dans cette classification, nous divisons 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 [YC08].

- ***Algorithme préventif*** : 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 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*** : 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 nouveau nœud agrégation...etc.

2.6.2 Classification architecturale

Cette classification traite les différents types de gestion des composants, soit au niveau du capteur individuellement ou bien sur tout le réseau. Nous distinguons trois catégories principales :

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 augmenter ainsi la durée de vie de tout le 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.

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.). Nous pouvons 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 qui soit le plus 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 des 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 d'agrégation 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 approches importantes 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 (parfois considérée comme approche curative).

2.7 Tolérance aux pannes d'une station de base (Sink)

Nous avons vu que le Sink peut être affecté par plusieurs et différents facteurs pouvant mener à son disfonctionnement, voire dans le pire des cas à son arrêt total. Ainsi, un arrêt de l'unique Sink d'un RCSF ne signifie pas moins que la perte totale des données récoltées par le réseau, d'où l'importance capitale d'avoir un mécanisme de tolérance aux pannes garantissant la survie du Sink.

D'un autre coté, étant donné le rôle du Sink et en considérant le pire scenario qu'est le crash permanant du Sink, la nécessité d'avoir un second Sink qui assure la continuité de la mission de tout le RCSF se fait fortement sentir et s'impose comme l'une des principales solutions contre une pannes d'un Sink. En effet, dans le cas de l'absence d'un autre point

d'évacuation des données récoltées par le RCSF vers l'utilisateur, le réseau se voit complètement isolé et les données totalement perdues.

Faire appel à d'autres Sinks dans un RCSF n'est pas pour le moins une idée nouvelle et il existe quelques travaux ([XIA], [ILK], [CIC]) traitant de l'utilisation de deux ou plusieurs Sink. Néanmoins, l'utilisation de plusieurs Sink dans le RCSF n'a pas forcement un objectif de tolérance aux pannes car elle peut, par exemple, avoir un objectif de maximisation de la durée de vie du RCSF, l'amélioration de la latence du RCSF ou encore la répartition des charges dans celui-ci.

Aussi, les travaux de recherche concernant l'utilisation de plusieurs Sink se penchent beaucoup plus sur le routage efficace et optimal vers plusieurs Sinks (avec la possibilité d'intégrer la notion de mobilité d'un Sink dans le RCSF) ainsi que le placement et la localisation optimal de ceux-ci dans le RCSF afin d'en prolonger la durée de vie en minimisant la consommation d'énergie.

Ayant vu qu'il est nécessaire d'avoir un Sink secondaire pouvant prendre le relais en cas de pannes du Sink primaire, une question importante s'impose :

Les nœuds du RCSF peuvent-ils jouer le rôle du SINK dans le cas d'une panne de celui-ci ?

Un Sink est généralement considéré comme un super nœud ayant des capacités (transmission, unité de traitement, énergie, stockage) supérieures aux autres nœuds capteurs du réseau à cause de la nature délicate de sa mission au sein du RCSF ; sa production est donc plus coûteuse et il est difficile d'imaginer un RCSF dont chacun de ses nœuds peut relayer le Sink en cas de pannes.

Nous présentons dans ce qui et d'une façon non exhaustive, quelque uns des rares travaux traitant de la problématique de la tolérance aux pannes d'une station de base dans un RCSF.

Dans [SALEH], les auteurs se sont penchés sur la question d'une éventuelle panne du Sink, en considérant un model de RCSF où tous les nœuds du RCSF sont ***homogènes***, cela veut dire que n'importe quel nœud ayant suffisamment d'énergie peut jouer le rôle du Sink

dans le réseau. La mission du Sink serait la collecte des informations reçues des nœuds sources et la leur acheminement vers l'utilisateur final.

Initialement, l'utilisateur choisit un nœud du RCSF comme premier Sink *S1*, si celui-ci tombe en panne, un autre nœud avec suffisamment d'énergie est élu comme nouveau Sink *S2*. Ainsi et après l'application successive (m fois) de ce mécanisme, le Sink sera dénoté S_m et le nœud S_{m-1} représentera le nœud de *checkpoint*. Le chemin engendré par le changement du Sink de S1 à S_m est un *chemin de check point.* Le Sink sauvegardera périodiquement son état dans le nœud *check point* afin de garantir une reprise cohérente lors du changement de Sink.

Le nœud de *check point* S_{m-1} est responsable de la détection d'une panne du Sink courant S_m en se basant sur l'information du niveau d'énergie du Sink qui lui est envoyée régulièrement avec la sauvegarde de l'état du Sink, ainsi qu'une estimation du temps de transmission périodique de l'état du Sink. Une fois la panne détectée, le nœud *check point* prend le relais et devient le nouveau Sink, ou bien choisit un autre nœud pour prendre la succession du Sink défaillant si le nœud *check point* est dans l'incapacité de le faire à cause d'un manque d'énergie.

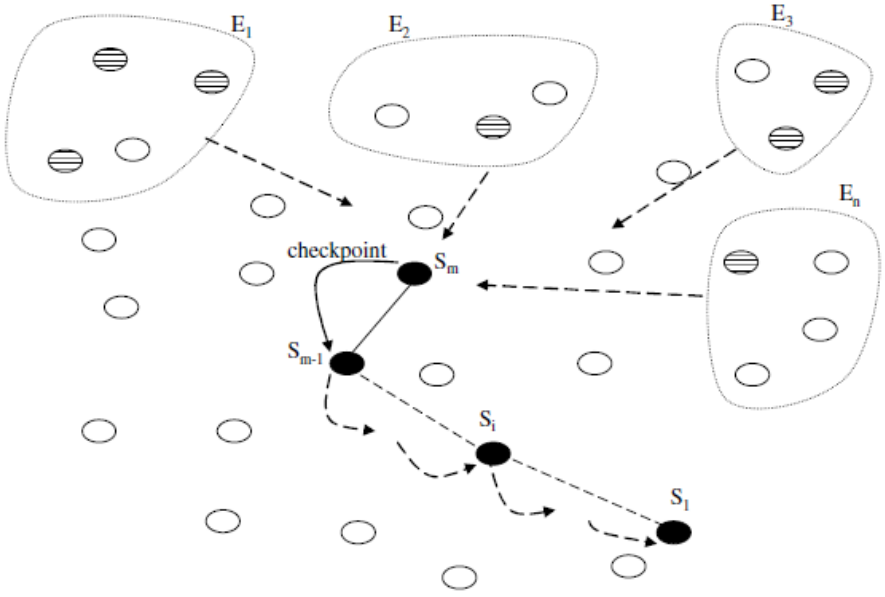

Figure 2.3 changement de SINK et checkpointing

D'autres travaux, notamment dans [DHM03], ont abordé la problématique de la tolérance aux pannes dans un Sink mais d'une façon secondaire. En effet, il s'agit plutôt d'une conséquence d'un mécanisme d'amélioration de la sécurité d'une station de base dans un RCSF. Les auteurs ont proposé dans cette étude, trois stratégies visant à augmenter la sécurité du Sink dans un RCSF (***Figure 2.4***) :

1. Un routage multi-chemin vers plusieurs Sinks
2. Brouillage des adresses et des champs d'identification dans les entêtes des paquets
3. Replacement du Sink dans le RCSF par une mobilité de celui-ci qui vise à améliorer la robustesse du réseau.

Les résultats obtenus dans cette étude démontrent que l'utilisation de plusieurs Sinks avec un routage multi-chemin vers ces derniers augmente considérablement la robustesse du réseau. En effet, une panne d'une ou plusieurs des stations de base du RCSF n'entraine que l'isolement d'un certain nombre de nœuds permettant au reste du réseau de continuer sa mission tant qu'il a à sa disposition un Sink opérationnel.

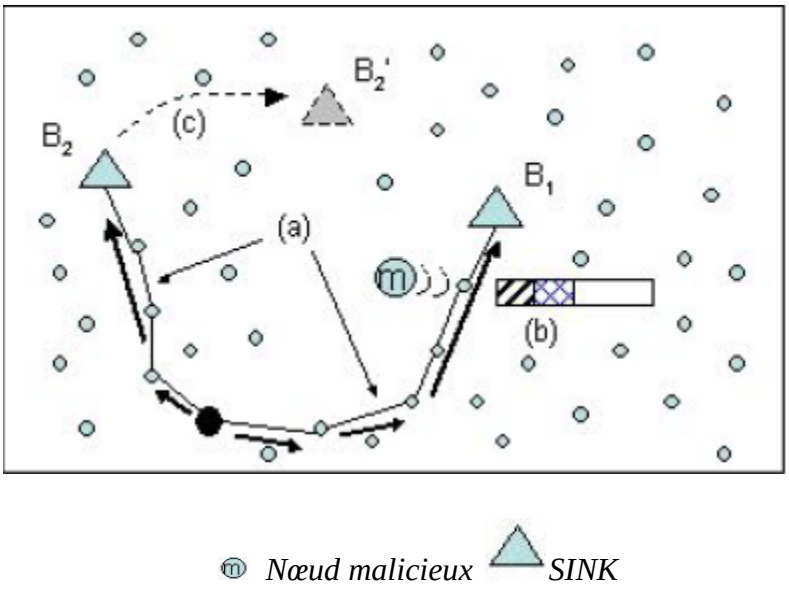

Figure 2.4 *(*a) routage multi-chemin (b) déguiser l'entête des champs adresse (c) replacement du SINK

2.8 Conclusion

Les travaux traitant de la problématique de la tolérance aux pannes d'une station de base sont peu nombreux, sans doute dû à la relative jeunesse du domaine et à l'absence de normes dans les RCSFs au niveau de la communication entre autres. Notre travail qui consiste en une proposition d'une stratégie de tolérance aux pannes pour un Sink s'oriente donc vers un paradigme de communication très répandu dans les RCSF, à savoir le Data-Centric duquel découle une implémentation qui est le protocole Directed Diffusion (DD). La stratégie proposée aura donc pour contexte de communication, le paradigme DD, et se basera en partie sur les mécanismes propres à ce paradigme pour offrir une continuité de la mission du RCSF dans le cas d'une panne du Sink.

Nous présentons dans le chapitre qui suit, le paradigme de communication Data-Centric DD, en exposant ses différents principes de fonctionnement.

CHAPITRE 3 :

Diffusion Dirigée (Directed Diffusion)

Dans ce qui suit, nous expliquons le fonctionnement et les mécanismes utilisés dans le paradigme Directed Diffusion. Pour illustrer ceci, nous prenons comme exemple, un réseau de capteurs sans fil chargé de détecter des cibles dans une région donnée. L'utilisateur de ce RCSF lui délègue la tâche suivante : "Fournir, durant les T prochaines secondes et toutes les 20 ms, la position de n'importe quel véhicule roulant se trouvant dans la sous-région R du champ de captage".

Nous détaillerons, ci-après, les étapes du fonctionnement du paradigme Directed Diffusion pour la dissémination des données [INT02, IGEHS, IGE].

3.1 Propagation des intérêts et établissement des gradients

Un intérêt représente une tâche assigné par l'utilisateur au RCSF à travers le SINK, celui-ci procède à la diffusion de l'intérêt reçu sur le RCSF. En effet, le SINK commence par envoyer périodiquement un message d'intérêt. Cette tâche (intérêt) est définie par la paire "attribut-valeur". Une description simplifiée de l'exemple précédent pourrait être de la forme suivante :

```
Type = véhicule roulant          // le type de la cible à détecter
Intervalle = 20 ms               // envoyer des événements toutes les 20 ms
Durée = 10 minutes               // durant les prochaines 10 minutes
rect = [0, 100, 200, 400]        // par les nœuds de cette région
```

Le message ainsi décrit permet de spécifier les données par lesquelles l'utilisateur est intéressé. Pour cette raison, il est appelé intérêt. Il peut contenir différents attributs. L'exemple précédent spécifie : le type de cible détectée, le débit (l'intervalle) avec lequel les réponses doivent être envoyées, la durée de validité de l'intérêt, et la sous-région de captage concernée par l'intérêt.

Au départ, l'intérêt est diffusé par le nœud collecteur à tous ses voisins ***(Figure 3.1)***. Le débit demandé initialement par l'intérêt envoyé est plus faible que celui demandé par l'utilisateur (Par exemple : intervalle = 1s, au lieu de : intervalle = 10ms).

Chaque nœud du réseau maintient un cache qui a pour but de garder trace des intérêts reçus. Chaque entrée du cache correspond à un intérêt distinct et contient plusieurs champs, parmi lesquels, une liste de gradients qui identifient les voisins depuis lesquels l'intérêt a été reçu. Un gradient spécifie une valeur et une direction dans laquelle les données répondant à l'intérêt seront envoyées. Le choix de la sémantique de la valeur associée au gradient dépend de l'application du RCSF. Dans l'exemple cité précédemment, cette valeur représente le débit avec lequel les données seront envoyées. Dans d'autres applications, cette valeur peut, par exemple, être une probabilité p avec laquelle une donnée est envoyée vers le voisin dans un souci d'équilibrage de charge.

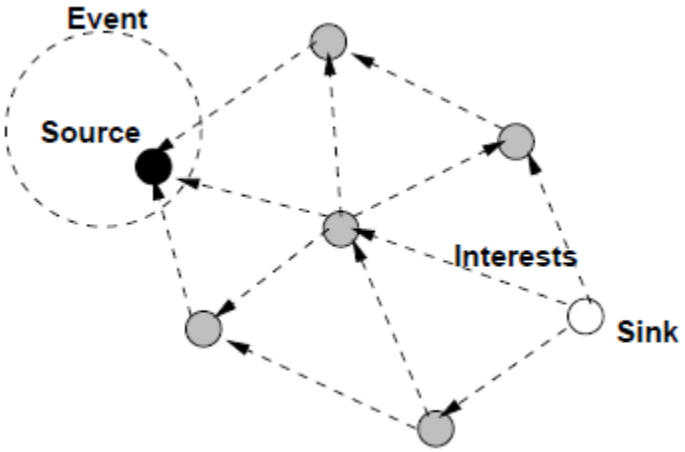

Figure 3.1 Dissémination des intérêts

Chaque gradient possède une certaine durée de vie qui représente la durée de vie de l'intérêt associé. Quand un gradient expire, il est retiré de l'entrée de l'intérêt correspondant. Quand tous les gradients d'une entrée d'intérêt expirent, l'entrée elle-même (donc l'intérêt), est supprimée du cache.

Quand un nœud reçoit un intérêt, il vérifie son cache d'intérêts. Si aucune entrée correspondante à l'intérêt reçu n'existe (il est différent des intérêts du cache), le nœud crée

une nouvelle entrée dans son cache, dans laquelle il va définir un gradient vers le voisin à partir duquel l'intérêt a été reçu. S'il existe une entrée correspondante, mais aucun gradient pour l'émetteur de l'intérêt, le nœud ajoute un gradient vers le voisin émetteur. Finalement, s'il existe déjà une entrée et un gradient pour l'émetteur, le nœud les met simplement à jour.
Par la suite, chaque nœud ayant reçu un intérêt diffuse ce dernier à tous ses voisins. Bien que l'intérêt soit initialement généré par le nœud collecteur, chaque nœud émetteur semble être l'origine de cet intérêt auprès de ses voisins récepteurs.

De cette manière, l'intérêt est diffusé dans tout le réseau par inondation. Cette technique peut causer une forte dépense en énergie. Cependant, l'inondation de tout le réseau est inévitable en l'absence d'informations sur les nœuds capteurs susceptibles de satisfaire l'intérêt.

Ainsi, un intérêt est disséminé à partir du nœud collecteur sur tout le réseau. Cette dissémination installe des gradients dans chaque nœud du réseau pour orienter les données de réponses à l'intérêt diffusé, vers le nœud collecteur (***Figure 3.2***).

Notons que le collecteur rediffuse périodiquement chaque intérêt. Ceci est nécessaire pour rétablir les gradients en cas de changements topologiques dans le réseau (défaillances de liens, mobilité, pannes de nœuds, etc.). Le taux de rafraîchissement d'intérêt est un paramètre de conception du protocole soumis à un compromis : charge du réseau / fiabilité de transmission.

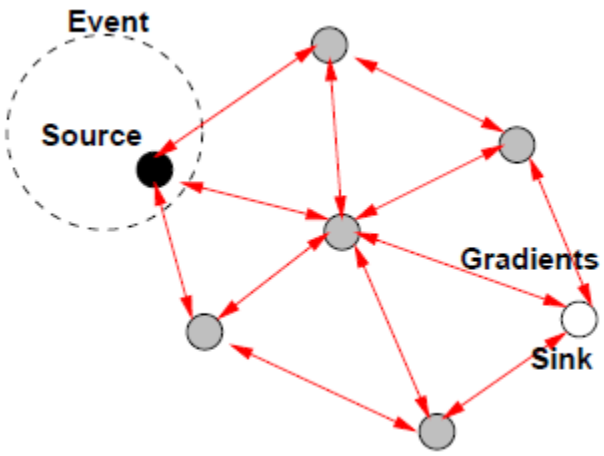

Figure 3.2 *Etablissement des gradients*

3.2 Propagation des données

Quand un nœud capte un évènement dans son champ de déploiement, il consulte son cache d'intérêts pour rechercher un enregistrement correspondant aux données captées. Si l'intérêt est trouvé, le nœud (qui devient désormais source de données pour cet intérêt) commence à envoyer l'évènement capté à tous ses voisins (constitués par les gradients établis dans la phase de diffusion d'intérêt) avec un faible débit appelé débit exploratoire. Suivant l'intérêt donné dans l'exemple adopté, l'évènement suivant forme une réponse pour le SINK :

Type = véhicule roulant	// type du véhicule détecté.
Instance = fourgon	// instance de ce type.
Position = [125, 220]	// position du nœud.
Intensité = 0.6	// amplitude de signal capté.
Confiance = 0.85	// degrés de confiance de la correspondance.
Date = 01 :20 :40	// heure de génération de la donnée.

Au départ, les données sont considérées comme exploratoires, et sont envoyées à tous les voisins, pour lesquels un gradient a été établi, avec un faible débit (dans l'exemple adopté, le débit est égal à une donnée par seconde). Ces données exploratoires sont acheminées vers le nœud collecteur dans le sens inverse de celui de la propagation des intérêts. Elles sont prévues pour la construction et la réparation des chemins qui seront adoptés par la suite.

Un nœud intermédiaire qui vient de recevoir un message de donnée exploratoire d'un nœud voisin, cherche dans son cache d'intérêts une entrée correspondante à l'évènement capté. Si aucune entrée n'est trouvée, le message de données reçu est simplement ignoré. Sinon, le nœud vérifie alors son cache de données (associé à l'entrée de l'intérêt). Ce dernier garde trace des données récemment envoyées et permet, entre autres, l'élimination des boucles. Si le message reçu correspond à une entrée du cache de données, il est alors ignoré car il a déjà été envoyé. Dans le cas contraire, le message est stocké dans le cache puis diffusé aux voisins pour lesquels un gradient a été établi.

Ainsi, chaque nœud intermédiaire exécute le même mécanisme en diffusant les données exploratoires vers tous ses voisins afin d'atteindre le nœud collecteur. Cette étape est caractérisée par la propagation des données exploratoires (appelée aussi phase d'exploration).

3.3 Renforcement positif

La phase de propagation des données exploratoires a pour but d'explorer les chemins existants entre la source et le SINK. Le renforcement positif sert à choisir une route (selon un certain critère) parmi celles découvertes, afin d'obtenir les données à haut débit appelées données renforcées. Pour cela, quand les données exploratoires atteignent le nœud collecteur, ce dernier choisit l'un de ses voisins appropriés et lui envoie un message de renforcement positif. Dans l'exemple adopté, un message de renforcement positif est identique à l'intérêt initial, sauf que le débit des données demandées est plus élevé (la valeur de l'attribut intervalle est plus petite. Exemple : intervalle = 10ms). Un nœud intermédiaire qui reçoit ce message de renforcement, renforce son gradient vers l'émetteur. Dans l'exemple, un gradient est considéré comme étant renforcé si sa valeur (c'est-à-dire le débit de données demandé) est supérieure à 1 donnée / sec. Par la suite, le nœud doit, à son tour, envoyer le message de renforcement à l'un de ses voisins (en suivant le même critère de sélection). De cette manière, l'un des chemins explorés est récursivement renforcé (***Figure 3.3***). Pour le choix du chemin à renforcer, le nœud collecteur, et par la suite les nœuds intermédiaires, appliquent localement la règle de renforcement positif.

Grâce au cache de données, un nœud choisit le premier voisin à partir duquel il a reçu une donnée exploratoire correspondant à l'intérêt. Par conséquent, un chemin ayant la plus faible latence est établi entre la source et le collecteur. D'autres règles peuvent être appliquées. Par exemple, un nœud peut choisir le voisin à partir duquel il a reçu le plus grand nombre de messages de données. Ainsi, le chemin le plus fiable est élu.

Une fois qu'un chemin reliant le nœud source au nœud collecteur est renforcé, les données générées par la source sont envoyées avec un débit plus élevé à travers ce chemin (***Figure 3.3***). Un nœud qui vient de recevoir une donnée renforcée enverra celle-ci uniquement aux voisins pour lesquelles il possède un gradient renforcé.

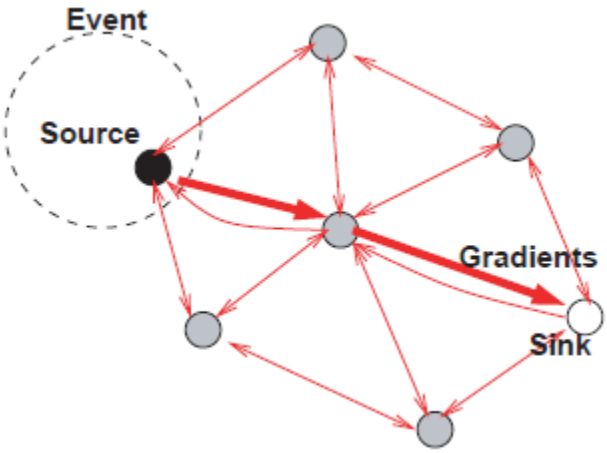

Figure 3.3 *Renforcement d'un chemin*

Cependant, même en présence d'un chemin renforcé, le nœud source rediffuse périodiquement des données exploratoires dans le but de renforcer ultérieurement un meilleur chemin vers le nœud collecteur. Dans notre exemple, les données exploratoires sont envoyées chaque seconde.

3.4 Etablissement de chemins pour de multiples sources et Sink

Les mécanismes de propagation de données et d'intérêts fonctionnent également pour un scénario incluant plusieurs sources. Initialement, tous les gradients sont exploratoires, et les données arrivent au Sink à travers les nœuds C et D, si l'un des nœuds (C par exemple) achemine plus rapidement les données vers le Sink, seul le chemin incluant ce nœud C est renforcé (***Figure 3.4***).

D'une manière similaire, si deux Sinks expriment un intérêt identique, la propagation des intérêts, des données et les règles de renforcement fonctionnent tous correctement. Si le Sink Y a déjà renforcé un chemin de qualité vers le nœud source, les autres nœuds du RCSF continuent quand même à recevoir des événements exploratoires (données et intérêts), de cette façon, quand l'utilisateur confie une tache au Sink X avec le même intérêt, X consulte son cache de données et renforce immédiatement un chemin vers le nœud source (***Figure 3.5***).

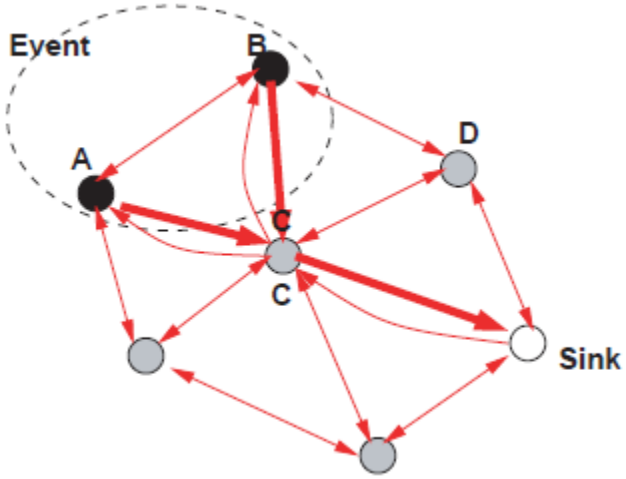

Figure 3.4 *Gradients après renforcement, plusieurs sources*

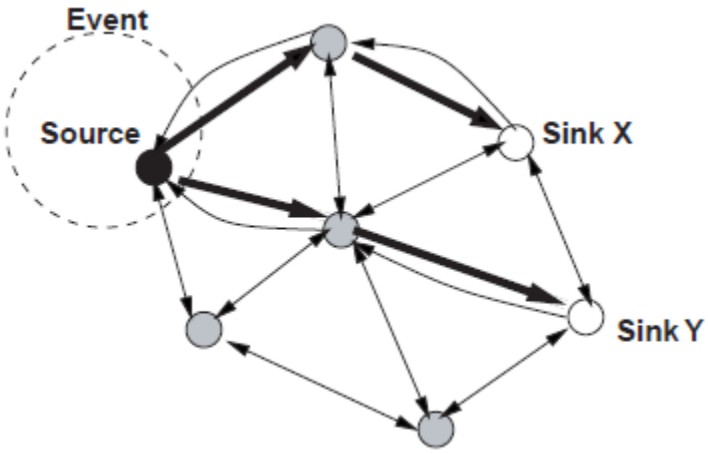

Figure 3.5 *Gradients après renforcement, plusieurs SINK*

3.5 Renforcement négatif

Le renforcement négatif est utilisé pour supprimer les chemins inutiles. La règle locale appliquée au niveau de chaque nœud dépend de la règle choisie pour le renforcement positif. Ce renforcement négatif permet par exemple de ***Réparer localement des chemins défaillants***.

En effet, Les nœuds intermédiaires peuvent appliquer les règles de renforcement pour permettre la réparation locale des chemins défaillants.
Les communications entre chaque paire de nœuds voisins d'un même chemin renforcé peuvent subir certaines dégradations ou défaillances dues à l'épuisement de la batterie d'un nœud, ou à des facteurs environnementaux (obstacles).

Considérons l'exemple de la ***Figure 3.6***, dans lequel la qualité du lien entre la source et le nœud C se dégrade et les données sont fréquemment perdues. Quand C détecte cette dégradation (soit en se rendant compte que d'autres voisins transmettent de nouvelles données et avec un débit plus important, ou en notant que le débit de réception des données depuis le nœud source est plus faible) il peut appliquer dés lors les règles de renforcement pour découvrir un nouveau chemin vers la source. Par la suite, C renforce négativement le lien direct à la source.

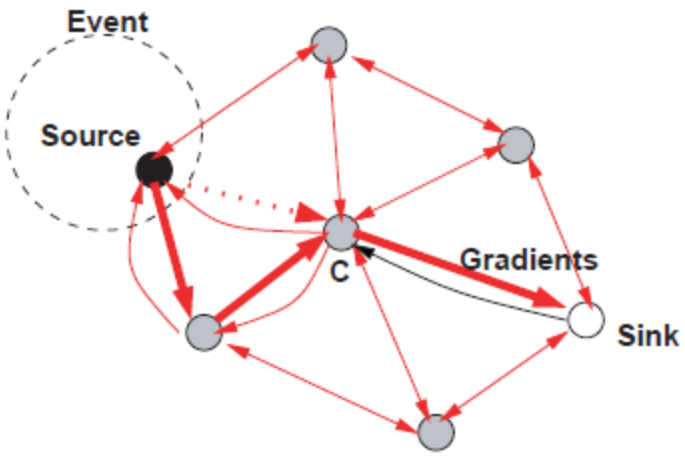

Figure 3.6 *Réparation des chemins défaillants*

Un nœud peut décider des voisins qui seront renforcés négativement selon plusieurs critères. Il peut choisir le voisin à partir duquel aucun nouvel événement n'a été reçu (c'est-à-dire que d'autres voisins ont régulièrement envoyé des événements avant lui) dans une fenêtre de N données ou de temps T. D'autres variantes consistent à renforcer négativement le voisin à partir duquel peu de nouvelles données ont été reçues.

3.6 Conclusion

Le paradigme de communication Directed Diffusion offre plusieurs possibilités de conceptions et ne limite donc pas le concepteur à une manière particulière quant à la propagation des intérêts/données et à l'établissement des gradients par exemple. Le tableau suivant (***Table 3.1***) montre un espace partiel de conception (différents choix de conception) pour quelques éléments du paradigme Directed diffusion.

Elément	Choix de conception
Propagation des intérêts	▪ Inondation ▪ Inondation directionnelle basée localisation (position) ▪ Propagation directionnelle basée sur les données en cache
Propagation des données	▪ Renforcement d'un unique chemin d'acheminement ▪ Acheminement multi-chemins avec qualité sélective le long des différents chemins ▪ Acheminement multi-chemins basée sur des probabilités
Renforcement	▪ Règles pour décider de quand renforcer ▪ Règles pour déterminer le nombre de voisins à renforcer ▪ Mécanisme et règles de renforcement négatif

Table 3.1 *Espace partiel de conception pour DD*

Directed diffusion expose donc une certaine souplesse dans les choix de conception de ce paradigme, ce qui renforce l'efficacité de la communication au sein d'un RCSF.

Dans le chapitre suivant, nous décrirons en détails la stratégie de tolérance aux pannes du SINK proposée dans le contexte de communication DD, pour ensuite proposer une modélisation des différents éléments du système permettant la vérification de quelque propriétés qui viendrons valider notre solution .

CHAPITRE 4 :

Stratégie de Tolérance aux pannes d'un Sink

Après avoir vu le fonctionnement du paradigme de communication Directed Diffusion, nous proposons dans ce qui suit une stratégie de tolérance aux pannes pour le nœud collecteur (SINK) dans le cadre de ce paradigme, nous commençons par établir quelques hypothèses nécessaire à ce contexte :

1. Le SINK est un nœud spécial du RCSF et les autres nœuds ne peuvent en aucun cas jouer le rôle du SINK.
2. La panne traitée dans cette stratégie est celle du crash total du SINK
3. La panne du SINK ***secondaire*** est exclue
4. La communication dans le RCSF est basée sur Directed Diffusion
5. Les SINK peuvent communiquer d'une façon directe

4.1 Un second SINK dans le RCSF

Comme précédemment évoqué, il est nécessaire d'avoir un SINK ***secondaire*** dans le RCSF, en prévision d'une éventuelle panne du SINK ***primaire***. Une panne est synonyme d'isolement total du RCSF et d'une perte complète des données récoltée à cause de l'aspect centralisé de la récolte et l'acheminement des données vers l'utilisateur final.

4.2 Objectifs de la stratégie proposée

Le fonctionnement correct d'un RCSF implique l'acheminement des données vers le SINK, ainsi que le bon fonctionnement du traitement effectué par le SINK sur ces données récoltées avant qu'elles ne soit envoyées à l'utilisateur final, autrement dit le bon déroulement de l'ensemble des processus exécutés sur le SINK. Lors d'une panne du SINK, ce fonctionnement correct du RCSF est altéré, il est donc clair que les deux principaux objectifs du mécanisme de tolérance aux pannes proposé soient :

- ✓ Le rétablissement de la communication au sein du RCSF d'un coté; c'est-à-dire entre les nœuds du réseau et un autre SINK éventuellement ; et avec l'utilisateur final de l'autre.
- ✓ La reprise cohérente de la part du nouveau SINK des processus anciennement exécutés sur le SINK en panne.

4.3 Rôle du SINK secondaire

Le SINK secondaire jouera un rôle passif quand le SINK ***primaire*** est opérationnel, ne participant pas aux tâches du RCSF (routage, agrégation, captage ou traitement de données…), La principale mission du SINK ***secondaire*** dans le RCSF va consister en :

1. La détection de la panne du SINK ***primaire***.
2. Le lancement de la procédure de recouvrement de la panne du SINK ***primaire*** qui permettra au SINK ***secondaire*** de prendre le relai.

4.4 Principe général du mécanisme proposé

Etant donné un RCSF avec une communication basée sur DD, nous dénotons le SINK primaire par $\mathbf{S_P}$ et le SINK secondaire par $\mathbf{S_S}$. P_P dénote le processus exécuté sur le SINK primaire, P_S est le processus présent sur le SINK secondaire. Il est évident que les deux SINK sont identiques du coté matériel ainsi que du coté logiciel, ceci est nécessaire afin que le SINK secondaire puisse continuer exactement la même mission du SINK primaire en panne.

Lorsque S_P diffuse un intérêt sur le RCSF, S_s le reçoit également et garde une trace de tous les intérêts diffusés par S_P , une fois la panne du SINK primaire détectée à l'aide d'un mécanisme d'envoi périodique (de la part de S_P) de check-point jouant également le rôle de *heartbeats*, le SINK *secondaire* diffuse alors la liste des intérêts encore actifs de S_P dans le RCSF afin d'établir de nouveaux gradients qui permettent d'acheminer les données collectées vers le nouveau SINK du RCSF, à savoir S_S (Notons que les messages d'intérêt ne comportent aucune information sur le SINK concerné dans DD). L'envoi de check-point permet la reprise de l'exécution du processus de S_P sur S_S.
Ce qui suit détaillera encore plus le principe général évoqué ci-dessus.

4.4.1 Gestion des intérêts diffusés

Lors d'un fonctionnement normal du RCSF, le SINK primaire diffuse ses intérêts sur le réseau afin d'obtenir les données dont il a besoin, ceci suivant le principe de la diffusion dirigée. Un intérêt comporte entre autres une durée qui indique le temps de validité de l'intérêt, ainsi qu'un intervalle d'envoi de données récoltées, le message d'intérêt ne contient aucun identificateur du SINK propriétaire de l'intérêt.

Le SINK primaire envoi un message contenant une copie de chaque message d'intérêt qu'il diffuse, au SINK secondaire, ce dernier garde une liste de tous les intérêts diffusés par S_P dans le réseau (***Figure 4.1***).

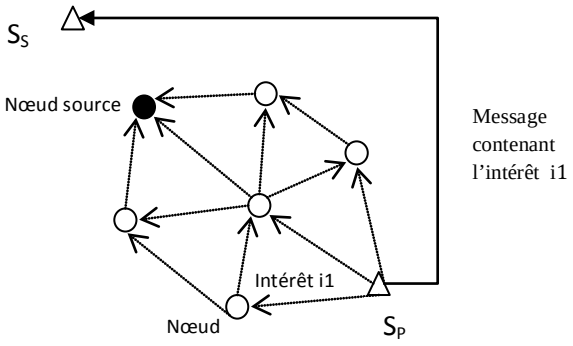

Figure 4.1 *S_S sauvegarde l'intérêt diffusé par S_P*

4.4.2 Surveillance et détection de pannes du SINK primaire

L'une des deux principales missions du SINK *secondaire* est la détection de la panne du SINK *primaire*, pour ce faire, nous utilisons un mécanisme de ***heartbeat*** qui permet au SINK primaire de signaler son bon fonctionnement au SINK secondaire à l'aide de messages « Alive », ces messages sont en réalité des check-point du processus P_P s'exécutant sur le SINK *primaire* permettant la reprise cohérente de celui-ci en cas de panne de S_P. S_P envoie un check-point à intervalle régulier I, permettant à S_S de conclure à une panne du SINK *primaire* s'il ne reçoit aucun message de S_P après l'écoulement d'un délai de garde T_G (***Figure 4.2***); ce délai dépend fortement de l'importance et de l'aspect critique des données récoltées mais tient aussi compte du taux de perte de données constaté dans les communications sans fil du réseau. Pour confirmer la panne du SINK primaire, le SINK secondaire tente de contacter le SINK primaire après l'écoulement du temps de garde, si aucune réponse n'est reçue après **m** tentatives, S_S conclut à une panne de S_P.

Le délai T_G et l'intervalle d'envoi des check-point « Alive » I doivent avoir une relation directe avec l'intérêt du SINK primaire qui possède le plus petit intervalle d'envoi de données, ceci permet de minimiser la perte de données engendrée par la latence du recouvrement de panne de S_P. En outre, plus le temps entre la panne et la détection effective

de celle-ci est petit, plus le taux de perte des données est moindre. Tout ceci en tenant toujours compte d'un autre coté, du taux de perte lié aux communications sans fil du réseau.

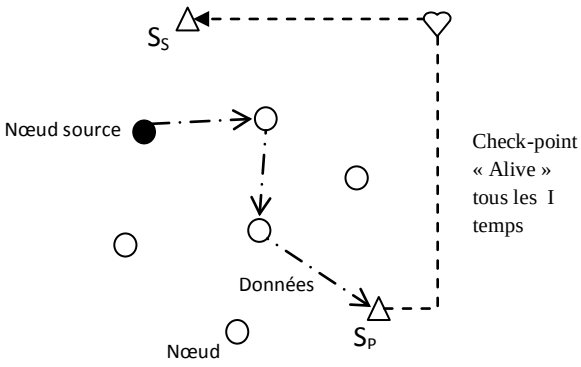

Figure 4.2 *surveillance du SINK primaire*

Lorsque le SINK primaire tombe en panne, le SINK secondaire ne reçoit plus de check-point « Alive », il détecte alors une panne de S_P après l'écoulement du délai de garde T_G et des **m** tentatives de communication avec S_P, et initie la procédure lui permettant de prendre le relais, qui est la diffusion des intérêts encore actif du SINK *primaire* en plus de la restauration du contexte d'exécution de S_P à partir des check-point reçus.

4.4.3 Initiation de la procédure de recouvrement de panne

Une fois la panne du SINK primaire avérée, le SINK secondaire consulte la liste des intérêts du SINK primaire qu'il possède (***Figure 4.3***).

Un intérêt est dit valide, si sa date d'expiration n'a pas encore été atteinte (l'attribut durée), S_S diffuse alors chaque intérêt encore valide sur le RCSF (les intérêts expirés sont ignorés et supprimés), ceci permet d'établir de nouveaux gradients pour un intérêt donné ; ce qui aura pour effet d'acheminer les données vers le nouveau SINK du réseau et non plus vers le SINK en panne. La diffusion d'intérêts a pour but dans un premier temps de recevoir des données exploratoires, pour que le SINK S_S puisse renforcer par la suite (avec un débit égal au débit avec lequel S_P recevait les données) un chemin parmi les chemins disponibles doit être choisi selon un critère donné.

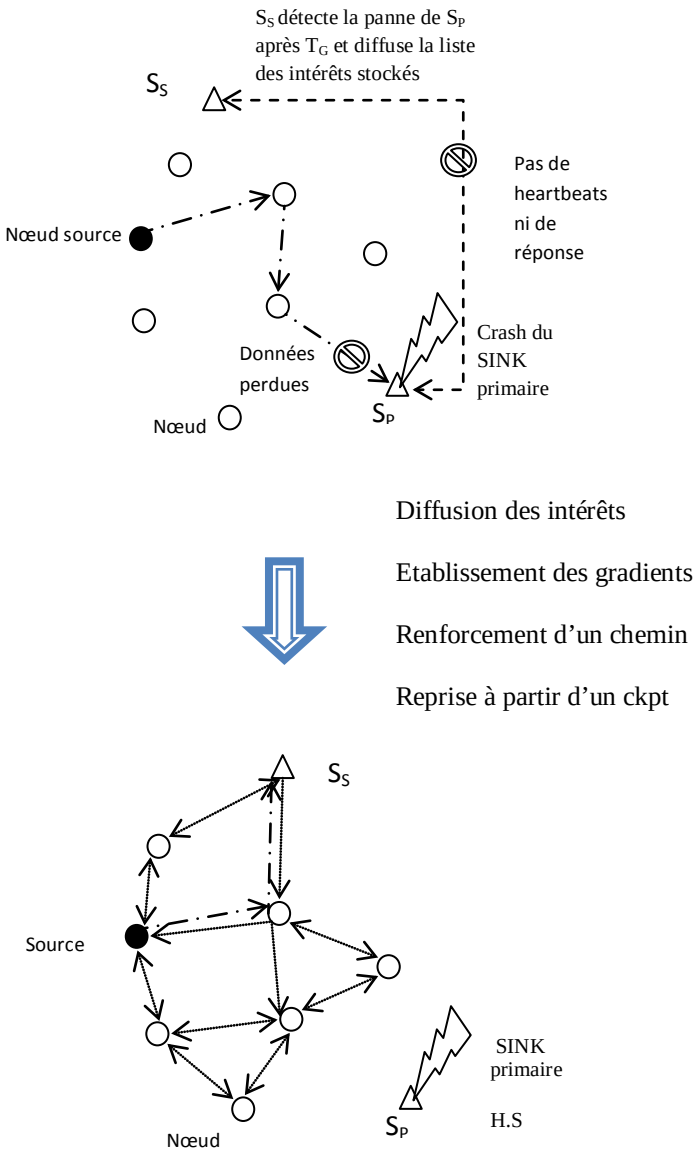

Figure 4.3 *Recouvrement de panne après le crash de S_P*

4.5 Algorithmes de la stratégie proposée

Nous décrivons dans ce qui suit, au niveau de chaque acteur du RCSF, l'algorithme qui permet d'assurer la stratégie proposée de tolérance aux pannes du SINK.

4.5.1 Au niveau du SINK primaire :

Réception et diffusion des intérêts et des données, envoi de check-points :

```
Début

Faire en parallèle {

        Sur_reception (Intérêt  i) :

                Envoi_S(i) ;
                Ajouter_int (i) ;
                Diffuser (i) ;

        Sur_reception (Donnée d) :

                Envoie_User (Traitement (d)) ;

        Tant que non_vide (liste_int)  faire

                Rafraichir (Liste liste_int) ;

        Envoyer_ckpt (checkpoint ckpt, Timer x) ;  }

Fin.
```

Figure 4.4 *au niveau du SINK primaire*

Les opérations sont exécutées en parallèle dans le SINK primaire (***Figure 4.4***), à la réception d'un intérêt i en provenance de l'utilisateur, le SINK primaire commence par l'envoyer en priorité vers le SINK secondaire, c'est l'opération exécutée en premier lieu et l'envoi est considéré comme étant fiable dans ce cas. L'intérêt est ensuite stocké dans une liste d'intérêts, puis diffusé sur le réseau de capteur sans fil.

Après la diffusion de l'intérêt sur le RCSF, les données commencent à arriver au niveau de S_P, dès leur réception, elles sont traitées par le SINK (le traitement effectué dépend de la nature de l'application du RCSF) puis sont acheminées vers l'utilisateur final.
Les intérêts actifs dans la liste d'intérêts doivent subir un rafraichissement périodique, cette opération est faite tant que cette liste n'est pas vide. Ce rafraichissement s'agit en fait d'une simple réémission de l'intérêt suivant un intervalle qui peut lui être propre.

La dernière tâche exécuté sur le SINK primaire et pas des moindres, est l'envoi périodique de check-point du contexte d'exécution vers le SINK secondaire ; S_P sauvegarde donc son contexte, et l'envoie d'une manière fiable au SINK secondaire, ceci à intervalle régulière I. Le

but étant de permettre au SINK secondaire de détecter une éventuelle panne de S_P en plus d'avoir un point de reprise d'exécution lors de la prise de relai.
Après avoir vu l'algorithme des tâches devant être accomplies dans le SINK principal, nous verrons dans ce qui suit, les opérations assurées par le SINK secondaire dans le cadre de la stratégie proposée.

4.5.2 Au niveau du SINK secondaire

Le SINK secondaire va jouer un rôle de Watch-dog vis-à-vis du SINK primaire (***Figure 4.5***), se basant sur un délai de garde prédéfini, fortement dépendant de l'aspect critique de l'application du RCSF et en relation avec l'intérêt ayant le plus petit intervalle de transfert de données ainsi qu'avec le taux de perte de données constaté dans les communications sans fil du réseau. On peut imaginer que $\mathbf{T_G}$ et **I** peuvent être modifiés durant la vie du RCSF pour plus d'efficacité.

A la réception d'un intérêt provenant de S_P, le SINK secondaire le stocke dans une liste d'intérêts afin d'avoir une trace de tous les intérêts diffusés par S_P sur le réseau ; il faut noter cependant que cette liste d'intérêts est périodiquement purgée afin de ne garder que les intérêts actifs à un moment donnée. La fréquence de cette opération dépend de la capacité de stockage des deux SINK.

D'un autre coté, lorsque le SINK $\mathbf{S_S}$ reçoit un check-point, il sauvegarde celui-ci et réinitialise immédiatement le délai de garde $\mathbf{T_G}$.
La réception des check-points et des intérêts diffusés par S_P sont les deux principales missions du SINK secondaire et sont donc exécutées tant que le délai de garde n'est pas dépassé. S'il y a dépassement du délai de garde, le SINK secondaire tente une dernière fois de contacter le SINK primaire à l'aide messages directement envoyés à S_P, si aucune réponse n'est reçue, S_S conclut à une panne de S_P.

Le SINK secondaire commence alors par restaurer le contexte d'exécution du SINK en panne à partir du dernier check-point reçu s'il existe (le SINK principal peut se crasher avant qu'il ne puisse envoyer le premier check-point), S_S diffuse ensuite tous les intérêts encore actifs à partir de la liste d'intérêts reçu préalablement du SINK en panne. Il s'agit la de la procédure de recouvrement de panne devant être respectée afin de rétablir la communication et le contexte d'exécution.

Le SINK secondaire ayant pris le relai assure la même mission qu'assurait S_P à une exception près, il n'est plus nécessaire d'envoyer des check-points « Alive » à un autre SINK dans le cas de notre scénario, mais une solution avec une chaine de plusieurs SINK reste envisageable et nécessite une coordination plus poussée entre ceux-ci.

```
Début

Tant que non_depass( Garde g)  faire {

        Faire en parallèle {

                Sur_reception ( Intérêt  i ) :

                        Ajouter_int (i) ;

                Sur_reception (checkpoint chk) :

                        Reinit_garde(g) ;

                        Save_chk(chk) ; } }

Restore_chk() ;

Diffuser (Liste liste_int) ;

Faire en parallèle {

        Sur_reception ( Intérêt  i ) :

                Ajouter_int (i) ;
                Diffuser (i) ;

        Sur_reception ( Donnée d) :

                Envoie_User (Traitement (d)) ;

        Tant que non_vide (liste_int)  faire

                Rafraichir (Liste liste_int) ;

Fin.
```

Figure 4.5 *Algorithme au niveau du SINK secondaire*

Nous avons vu les différents algorithmes suivis au niveau des deux SINKs, pour le reste des nœuds du RCSF, le contexte de communication reste basé sur Directed Diffusion et ses mécanismes. Nous allons évoquer dans ce qui suit, la latence du recouvrement de panne du SINK S_P, engendrée par différents facteurs liés à la communication.

4.6 Temps du recouvrement de panne

Dans Directed diffusion, il est nécessaire d'envoyer des données exploratoires avant de renforcer un ou plusieurs chemins selon des critères précis, en d'autres termes, le SINK envoie un intérêt exploratoire pour commencer à recevoir ensuite des données (exploratoires) avec lesquelles il renforcera un chemin donné. Un certain temps s'écoule donc avant que le SINK ne puisse recevoir les données avec le taux de transfert exigé par l'intérêt en question. C'est ce qui arrivera lors de la prise de relai du SINK secondaire, après détection d'une panne du SINK primaire.

Notons $\mathbf{T_E}$ le temps d'exploration, il s'agit du temps nécessaire à la réception des premières données exploratoires par le SINK après diffusion d'un intérêt.
Le temps de renforcement est noté $\mathbf{T_R}$, c'est le temps requis pour que le SINK reçoive les premières données renforcées après renforcement d'un ou de plusieurs chemins pour un intérêt.

Le temps nécessaire à la réception des données effectives (renforcées) après la première diffusion d'un intérêt n'est autre que ($\mathbf{T_R + T_E}$).

Lors d'une panne du SINK primaire à un moment noté T_P, S_S conclut à une panne qu'après l'écoulement de T_G, dans le pire des cas, la panne se produit juste au moment où S_P envoie son check-point qui signale qu'il est opérationnel (ou plus exactement, au moment où S_S réinitialise T_G), il faut alors un temps T_G pour une détection effective de la panne.

Une fois la panne détectée au niveau de S_S, et en supposant que les intérêts de la liste sont diffusés quasi simultanément, il faudra attendre un temps de $\mathbf{T_G} + [(\mathbf{T_R} + \mathbf{T_E})]_i$ avant de recevoir les premières données effectives de l'intérêt **i** de la liste dont le temps de réception effective des données est le plus grand (($\mathbf{T_R + T_E}$) étant différent d'un intérêt à l'autre).

Dans le meilleur des cas, le temps effectif de la panne est juste avant l'écoulement de T_G, le temps de recouvrement est alors égal à $[(\mathbf{T_R} + \mathbf{T_E})]_i$

D'une façon générale, le temps de recouvrement (rétablissement de la communication et du traitement au niveau du SINK secondaire) $\mathbf{T_{RC}}$ de la stratégie proposée est défini par :

$$[(\mathbf{T_R} + \mathbf{T_E})]_i \leq \mathbf{T_{RC}} \leq \mathbf{T_G} + [(\mathbf{T_R} + \mathbf{T_E})]_i$$

Comme évoqué précédemment, ce temps de recouvrement ne peut excéder le délai de garde ajouté au temps nécessaire à la réception des données effectives de l'intérêt le plus lent, ni être inferieur à $[(\mathbf{T_R} + \mathbf{T_E})]_i$, en supposant que la liste des intérêts à diffuser au niveau de S_S ne soit pas vide.

4.7 Consommation supplémentaire d'énergie

Les traitements et communications engendrés par un intérêt **i** lors du temps $[(\mathbf{T_R} + \mathbf{T_E})]_i$ diminuent la durée de vie du RCSF en consommant de l'énergie, lors de la prise de relai du SINK secondaire, la liste diffusée provoque une consommation supplémentaire d'énergie.
Chaque intérêt présent dans la liste implique son lot de messages échangés et de traitements effectués, la consommation engendrée par la stratégie proposée est donc :

$$C_S = C_0 + C_1 + \ldots\ldots + C_i$$

Où :

C_S : consommation supplémentaire

C_i : consommation engendrée par l'intérêt **i** pendant la durée $[(\mathbf{T_R} + \mathbf{T_E})]_i$

Il est clair qu'une fausse détection d'une panne du SINK primaire serait pénalisante en termes de consommation d'énergie entre autres, aussi, le délai de garde et l'intervalle d'envoi des check-points doivent être déterminés d'une façon efficace en tenant compte des différents aspects évoqués.

4.8 Fausse détection d'une panne de S_P

Si malgré tout, une fausse détection est constatée de la part du SINK secondaire et ce en recevant un check-point du SINK S_P bien après la prise de relai (ou en recevant **n** check-points pour s'assurer que la détection était réellement fausse), il est impératif pour le SINK secondaire de revenir vers un état de Watch-dog et de laisser S_P continuer sa mission, tout en

ayant la possibilité de paramétrer le délai de garde en conséquence, afin d'éviter une éventuelle autre future fausse alerte.

4.9 Conclusion

Nous avons vu dans ce chapitre et dans un contexte de communication basé sur Directed Diffusion, les principes de la stratégie proposée ainsi que les algorithmes devant être suivis au niveau de chaque SINK. Il a été question ensuite du temps nécessaire au SINK secondaire pour prendre le relai en définissant les bornes supérieures et inferieures de cette latence, pour évoquer ensuite l'impact de cette stratégie sur la durée de vie du RCSF ainsi que l'impact d'une éventuelle fausse détection. Dans le chapitre qui suit, nous validons la stratégie proposée par une modélisation du comportement du système dans UPPAAL, pour en vérifier des propriétés qui permette de conclure à la justesse de la proposition.

CHAPITRE 5 :

Modélisation et validation à l'aide du Model-checker UPPAAL

5.1 Le Model-Checker UPPAAL

UPPAAL [PET, BEH04, BEN96, PET99] est un outil pour la modélisation, la validation et la vérification des systèmes temps réel. Il est approprié pour les systèmes qui peuvent être modélisés par un réseau d'automates temporisés entre autres. Autrement dit, UPPAAL est approprié pour les systèmes qui peuvent être modélisés sous la forme de processus non-déterministes avec une structure de contrôle finie et des horloges avec des valeurs réelles, communiquant par des canaux ou des structures de données partagées.

Un automate temporisé est une machine à états finis étendu avec des variables de type horloge. Une variable horloge possède des valeurs dans l'intervalle des nombres réels et toutes les horloges progressent de manière synchrone. Dans UPPAAL, un système est modélisé comme un réseau de plusieurs de ces automates temporisés en parallèle. Le modèle est enrichi par des variables discrètes bornées qui font partie de l'état. Ces variables sont utilisées comme dans les langages de programmation classiques : elles peuvent être lues, écrites, et font l'objet des opérations arithmétiques communes. Un état du système est défini par les emplacements de tous les automates, les contraintes d'horloge, et les valeurs des variables discrètes. Chaque automate peut séparément franchir une transition ou bien synchroniser avec un autre automate, ce qui conduit à un nouvel état.

Nous présentons dans ce volet, UPPAAL, sa syntaxe et sa sémantique. Nous l'illustrons, par la suite, sur quelques exemples de comportements.

5.1.1 L'outil UPPAAL

UPPAAL est un ensemble d'outils pour la vérification automatique des propriétés de sûreté et de vivacité bornée, des systèmes temps réel. Il est construit à base d'architecture Client/serveur. La machine UPPAAL est le serveur, elle est développée en C++. L'interface graphique utilisateur (GUI) est le client, développé en JAVATM. La communication s'effectue via des protocoles internes.

Ainsi, la conception d'UPPAAL offre la possibilité d'exécuter le serveur et l'interface GUI sur deux machines différentes. La machine UPPAAL est constituée de plusieurs outils tels que checkta (Syntax Checker) et verifyta (Model Checker). L'interface graphique utilise des outils tels qu'atg2ta et hs2ta.

5.1.2 Syntaxe

La base du modèle d'UPPAAL est la notion des automates temporisés, développée par Alur et Dill [ALD94], comme extension des automates à états-finis classiques avec les variables d'horloges et de données. Pour avoir une modélisation plus expressive et pour la rendre plus facile, les automates temporisés sont étendus avec des types plus généraux de variables de données telles que les variables booléennes et entières. Le but est le développement d'un langage de modélisation plus proche que possible d'un langage de programmation des systèmes temps réel de haut niveau. Clairement, ceci va créer des problèmes de décidabilité. Cependant, on peut exiger que le domaine de valeurs des variables de données doive être fini afin de garantir la terminaison de la procédure de vérification.
Les transitions d'un automate temporisé, dans UPPAAL, sont étiquetées par trois types d'étiquettes (***Figure 5.1***) :

- Les gardes, exprimés sur les valeurs d'horloges et les variables entières qui doivent être satisfaites dans l'ordre pour les transitions qui vont être franchies ;
- Une action de synchronisation qui est exécutée quand les transitions sont franchies ;
- Les affectations des variables entières.

Tous ces trois types sont optionnels.

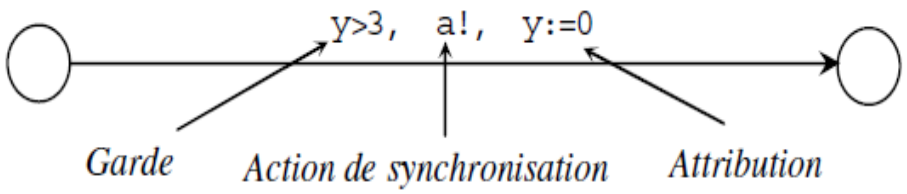

Figure 5.1 *Une Transition dans une UPPAAL*

En plus, les nœuds de contrôles peuvent être étiquetés avec des invariants, qui sont des conditions exprimant des contraintes sur les valeurs d'horloges pour demeurer dans un nœud de contrôle particulier.

Les gardes :

Les gardes expriment les conditions sur les variables d'horloges et des variables entières qui peuvent être satisfaites dans l'ordre de franchissement des transitions. Formellement, les gardes sont la conjonction des contraintes temporisées et des contraintes de données. Une contrainte d'horloge est de la forme : $x \sim n$ ou $x - y \sim n$, où n est un nombre naturel et $\sim \in \{\leq, \geq, =, >, <\}$. Une contrainte de données est de la même forme, $i \sim j$ ou $i - j \sim k$, sauf que k est un entier arbitraire. La garde d'une transition est par défaut vraie.

Dans la (***Figure 5.3)***, la transition entre A0 et A1 n'est franchie que si la valeur de l'horloge est supérieure ou égale à 3.

Opération de remise à zéro :

La remise à zéro d'une horloge ou d'une variable de données sur une transition est une affectation de la forme w := e, sachant que w est une horloge ou variable de donnée et e une expression.

L'opération de remise à zéro d'une horloge est de la forme x := n, où n est un nombre entier. La remise à zéro des variables de données est de la forme i := c * i + c', où c et c' sont des constantes entières (sachant que c et c' peuvent être nuls ou négatifs).

Canaux, synchronisation et urgence

Un modèle UPPAAL consiste en un réseau d'automates temporisés (étendu). Les automates peuvent communiquer via des variables entières (dans UPPAAL elles sont globales), ou alors en utilisant des canaux de communication, permettant l'envoi et la réception d'un message. La communication sur un canal apparaît comme étant la synchronisation entre deux processus.

a! (Envoie) et a? (Réception) dénotent qu'un processus synchronise avec un autre. L'absence de l'action de synchronisation dénote une transition (de non synchronisation) interne. Pour empêcher un automate de s'attarder dans une situation, où deux composants sont capables de synchroniser, il faut déclarer le canal comme étant urgent. Pour des raisons d'efficacité, les transitions sont étiquetées avec des actions de synchronisation sur des canaux urgents, et elles doivent être privées des gardes sur les horloges pour que les transitions de deux processus puissent être franchies simultanément et ne pas bloquer un système synchronisé avec un autre.

Emplacement Committed (Committed location)

Soit un émetteur S qui émet un message m à deux destinataires R1 et R2 (***Figure 5.2***). La synchronisation entre ces trois processus ne peut pas être directement exprimée en UPPAAL. Elle ne peut être effectuée qu'entre deux processus uniquement. Cependant, l'émission sera modélisée sous la forme d'une séquence de synchronisation entre deux processus, où S synchronise avec R1 sur m1 puis avec R2 sur m2. Pour assurer l'atomicité de la synchronisation, on marque le nœud intermédiaire comme engagé (indiqué par C).

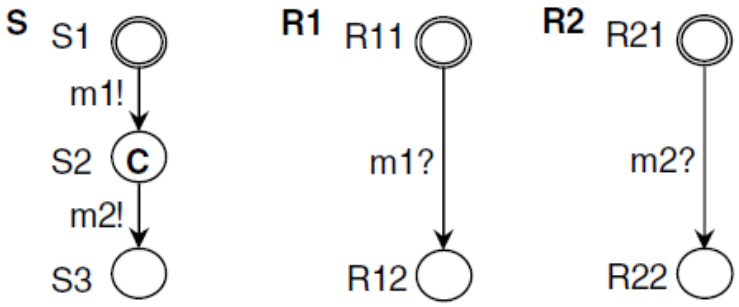

Figure 5.2 *Synchronisation entre 3 processus en utilisation un état Comitted*

Invariant

Pour renforcer la progression dans le temps, les nœuds de contrôles doivent être étiquetés par des invariants, qui expriment dans l'ordre les contraintes sur les valeurs d'horloges pour le nœud de contrôle, afin qu'il puisse y résider.

5.1.3 Sémantique

Formellement, les états du modèle UPPAAL sont de la forme (l, v), où pour chaque composant du réseau d'automate, l est un vecteur de contrôle indiquant le nœud de contrôle en cours, et pour chaque horloge et variable entière v est un assignement indiquant la valeur en cours.

Transition de délai :

Le temps doit évoluer sans l'affectation des vecteurs des nœuds de contrôles et en incrémentant les valeurs des horloges avec la durée du temps écoulé, aussi longtemps qu'aucun invariants des nœuds de contrôle n'est violé.

Transition d'action :

Si deux transitions complémentaires de deux composants différents sont rendues actives, dans un état, alors elles peuvent synchroniser.

Canaux urgents :

Aucun délai n'est permis dans un état, où deux composants doivent synchroniser sur un canal urgent. Ainsi, dans la (***Figure 5.3***), si un canal a est urgent, alors le temps ne doit pas s'écouler pendant 3,5 secondes à partir de l'état initial ((A0,B0), x = 0, y = 0, n = 0) puisque la synchronisation sur a est possible dans l'état ((A0,B0), x = 3, y = 3, n = 0).

5.1.4 Exemple général

Nous illustrons la modélisation graphique et textuelle dans UPPAAL avec un exemple général.

Modélisation Graphique

La **(*Figure 5.3)*** présente un système constitué de deux composants modélisés par les automates A et B.

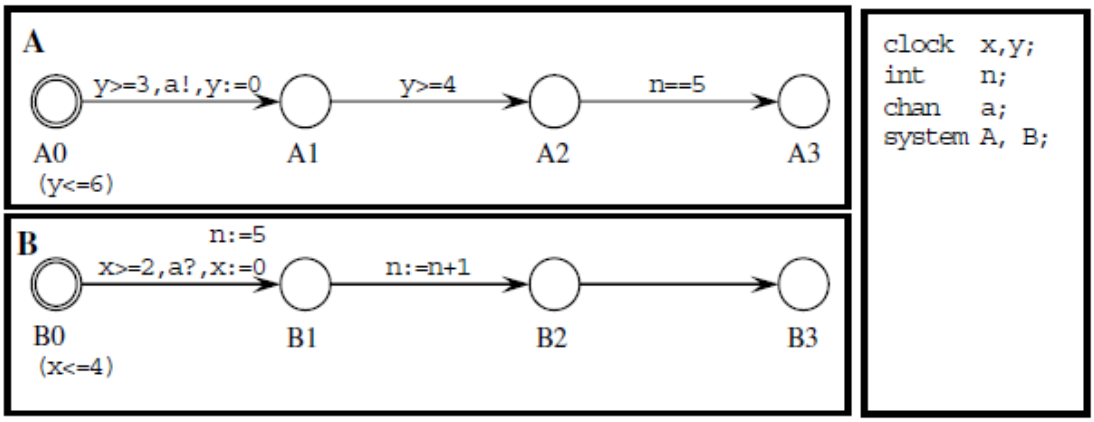

Figure 5.3 *Exemple d'un modèle UPPAAL*

Ces deux automates se synchronisent via le canal a.

Modélisation textuelle

L'exemple (***Figure 5.4***) suivant illustre la traduction du modèle précédent de la ***(Figure 5.3)*** en description textuelle dans UPPAAL.

```
// Déclaration Globale

clock   x, y ;
int       n ;
chan      a ;

// Description des composants

process  A {
        state A0 { y<=6 }, A1, A2, A3 ;
        init A0 ;
        trans A0 -> A1 {guard y>=3 ;
                sync a !;
                assign y :=0 ; },
        A1 -> A2 {Guard n==5 ; } ; }

process B {
        state B0 { x<=4 }, B1, B2, B3 ;
        commit B1 ;
        init B0 ;
        trans B0 -> B1 {guard x>=2 ;
                sync a ?;
                assign n :=5,x :=0 ; },
        B1 -> B2 {Assign n :=n+1 ;},
        B2 -> B3 {} ; }

// Description du Système

system A, B ;
```

Figure 5.4 *Description textuelle en UPPAAL*

Vérification dans UPPAAL

UPPAAL permet de vérifier différentes propriétés d'un système modélisé, ceci en utilisant un langage de vérification formel, permettant de définir les propriétés à vérifier. Une expression de vérification se compose d'une formule d'état et d'une formule de chemin ; une formule d'état décrit des états individuels, tandis que la formule de chemin permet l'évaluation sur des chemins donnés.

Une formule d'état est une expression qui peut être évalué à un état donné, un exemple est celui de la formule d'état ***deadlock*** qui peut être évalué pour tous les chemins possible du modèle pour garantir que le système modélisé n'est pas bloquant.

UPPAAL est capable de vérifier les propriétés : d'atteignabilité, de sûreté, de vivacité simple, de vivacité bornée, et de non-blocage par des techniques de vérification ou de model–checking variées.

Un exemple de vérification de la propriété d'atteignabilité peut être exprimé comme suit : existe-il un chemin commençant à l'état initial et au long duquel une expression ***exp*** est satisfaite. Formellement dans le langage de définition des propriétés à verifier : **E < > exp**

5.2 Modélisation et vérification du RCSF dans UPPAAL

La modélisation du RCSF dans UPPAAL se base sur deux automates principaux (***Figure 5.5*** et ***5.6***), qui représentent les deux SINK du RCSF, ce réseau d'automates communique par échange de différents messages à l'aide de canaux de synchronisation.

L'automate du SINK primaire dans UPPAAL se compose des états *traitement* et *crash*, le SINK secondaire a pour états *surveillance* et *remplacement*. Le changement d'état s'effectuera en fonction de variables (horloges) symbolisant les intervalles d'envoi de check-point ainsi que le délai de garde permettant la détection de pannes.

5.1.3 Description détaillée du modèle

Le modèle du RCSF réalisé dans UPPAAL se compose de quatre éléments (automates) distincts :

- L'automate du SINK primaire
- L'automate du SINK secondaire
- L'automate du RCSF (une abstraction de l'ensemble des nœuds du RCSF)
- L'automate de l'environnement (responsable du crash de S_P)

Nous allons passer en revue dans ce qui suit, chacun des automates précédemment cités.

SINK Primaire

L'automate du SINK primaire (figure 5.5) possède deux principaux états, un état de fonctionnement normal qui est ***traitement,*** et un état puits nommé ***crash,*** symbolisant l'arrêt du SINK primaire. En plus des états cités, le SINK primaire possède des états relatifs aux événements d'envoi de check-point, de réception de données ainsi qu'au rafraichissement de l'intérêt, la différence est que ces états sont du type ***comitted,*** ceci pour souligner le fait que ces tâches sont exécutées dans un laps de temps très court. Nous utilisons plusieurs canaux de synchronisation pour l'envoi des différents types de messages :

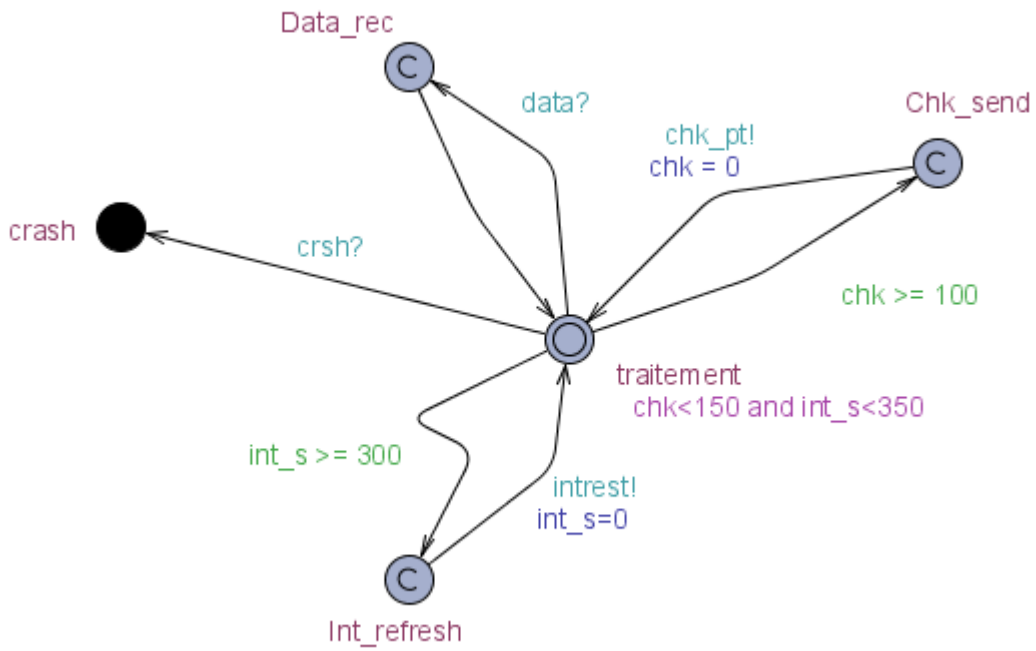

Figure 5.5 *Automate du SINK Primaire dans UPPAAL*

- le canal ***intrest*** pour l'envoi d'intérêts,
- le canal ***data*** pour la réception de données provenant du RCSF,
- le canal ***chk_pt*** pour l'envoi de check-points,
- et enfin, le canal ***crsh*** pour l'envoi du signal de crash au SINK primaire.

Pour les intervalles d'envoi de messages, nous utilisons des variables de type horloge ; une horloge ***int_s*** pour l'intervalle d'envoi des intérêts, une autre nommée ***chk*** pour l'intervalle d'envoi des check-points ; ces horloges sont réinitialisées à chaque envoi. L'état traitement bénéficie en plus d'un invariant assurant l'envoi des messages évoqués dans les temps.

SINK Secondaire

L'automate du SINK secondaire contient également deux principaux états, ***Surveillance*** et ***Remplacement_P_S***. les autres états sont du type ***comitted*** et représentent différents événements se produisant au niveau du SINK secondaire, on trouve l'état ***chk_reception*** relatif à la réception et à la sauvegarde du check-point envoyé par le SINK primaire, ainsi que l'état ***restore_chk_diffuse_int*** qui représente un état de transition après la détection de la panne du SINK primaire ; cet état regroupe la tâche de restauration du processus à partir du check-point sauvegardé, et celle de la diffusion des intérêts encore actif de S_P .

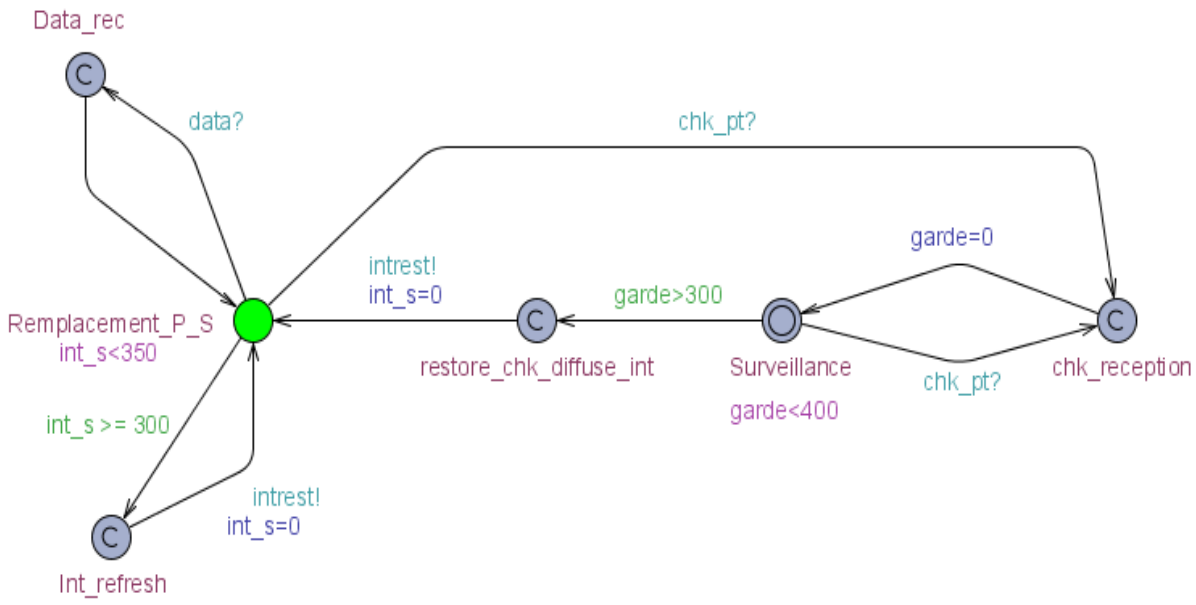

Figure 5.6 *Automate du SINK secondaire dans UPPAAL*

Nous employons une variable horloge nommée ***garde*** pour la détection d'absence de check-points de la part du SINK primaire, ceci est obtenu en liant à la fois une garde à l'arc menant de l'état ***surveillance*** à l'état ***restore_chk_diffuse_int***, et un invariant à l'état ***surveillance*** de S_S. Une fausse détection remet le SINK secondaire dans l'état surveillance (détectée par la réception d'un check-point tardif).

RCSF et Environnement

L'automate du RCSF (***figure 5.7***) génère les messages de données dans son état ***fonctionnel*** suivant l'horloge ***dat***, quand il reçoit un intérêt, le RCSF met à jours ses gradients, ceci est symbolisé par l'état ***gradients_update.***

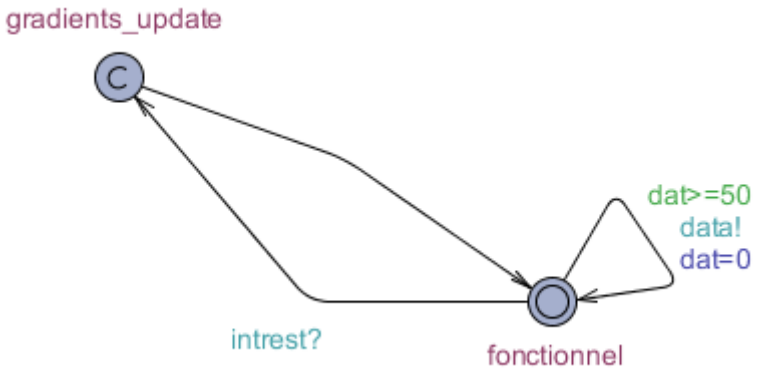

Figure 5.7 *Automate du RCSF dans UPPAAL*

Quant à l'automate modélisant l'environnement (***figure 5.8***), celui-ci envoi un signal de crash au SINK primaire après l'écoulement d'un certain temps, se basant sur la variable horloge ***Env_x***.

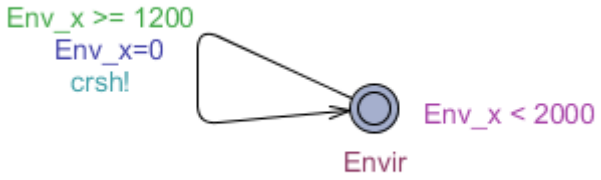

Figure 5.8 *Automate l'environnement dans UPPAAL*

5.1.4 Vérification du modèle proposé

Nous définissons dans ce qui suit, les propriétés importantes du système devant être garanties afin de valider la stratégie proposée :

Absence de blocage

Cette propriété est vérifiée par l'expression suivante :

- ✓ **A[] not deadlock**

Propriété de Sureté du système

Cette propriété garantie que le SINK secondaire atteindra surement l'état de surveillance après l'écoulement du délai de ***garde***, lorsque le SINK primaire se trouve dans l'état ***crash,*** ceci se traduit par l'absence d'un état global du système où le SINK secondaire se trouve dans l'état ***surveillance***, le SINK primaire dans l'état ***crash***, et la valeur de l'horloge ***garde*** supérieur au délai de garde fixé :

- ✓ **A[] not(P_Sink.crash and garde>400 and S_Sink.Surveillance)**

Propriété de vivacité/réactivité du système

Cette propriété garantie que l'état ***crash*** du SINK primaire conduira éventuellement le SINK secondaire vers l'état ***remplacement*** (réactivité du SINK secondaire face au crash du SINK primaire).

- ✓ **P_Sink.crash --> S_Sink.Remplacement_P_S**

Ainsi, la vérification du modèle garantie des propriétés de sureté dont l'absence d'inter blocage ainsi que l'impossibilité après un certain temps, d'avoir un état global où l'automate du SINK secondaire soit dans l'état surveillance alors que l'automate du SINK primaire est dans l'état crash, d'où la garantie que la panne du SINK primaire sera toujours détecté après l'écoulement du délai de garde.

Conclusion générale

La technologie des RCSFs est relativement jeune et nécessite encore des recherches approfondies sur des aspects divers tels que la consommation d'énergie, le routage, la sécurité, la localisation, la tolérance aux fautes etc. Cette dernière revêt une importance toute particulière du fait de la prédisposition aux pannes des nœuds d'un RCSF, prédisposition due à la nature hostile et inaccessible de beaucoup d'environnements susceptibles d'accueillir leurs déploiements.

Les causes d'une panne d'un nœud sont multiples, mais la gravité de la panne varie selon le type du nœud affecté, il est donc clair qu'un crash du SINK serait très handicapant, voir fatal au réseau, en cas d'absence de politique de gestion de ce type de pannes. Il est alors impératif d'assurer la survie et la continuité de la mission d'un RCSF en cas d'arrêt de la station de base.

Le travail contenu dans ce mémoire a consisté en une étude des RCSFs d'une façon générale, et de leur tolérance aux fautes, plus particulièrement celle du SINK. Quelques travaux traitant de cette problématique (panne du SINK) ont été exposé tout en soulignant la faiblesse des solutions proposées.

Il a été question par la suite, de situer la stratégie de tolérance aux pannes du SINK proposée, dans un contexte de communication bien défini ; celui de Directed Diffusion : un paradigme de communication très répandu dans la littérature ; nous en avons exposé et expliqué le principe de fonctionnement ainsi que les différents mécanismes le composant.

Nous avons proposé enfin, une stratégie de tolérance aux pannes d'une station de base (SINK), reposant sur l'idée intuitive d'avoir un second SINK dans le RCSF, jouant le rôle de Watch-dog, prêt à prendre le relai en cas de panne du SINK primaire (Sp). Le SINK secondaire (Ss) capte et garde la trace des différents intérêts diffusés par le SINK primaire, comme il envoie également à ce dernier et d'une façon périodique, un check-point lui permettant une reprise ultérieure en cas de panne. Après détection de celle-ci à l'aide d'un délai de garde relatif à l'envoi périodique des check-points et une rediffusion des intérêts encore actifs, la reprise à partir du check-point le plus récent est alors initiée par le sink

secondaire Ss. Nous avons examiné ensuite, le temps de recouvrement d'une panne de S_P en fonction du temps d'occurrence de la panne, ainsi que la consommation supplémentaire d'énergie engendrée par le mécanisme proposé. Pour finir, La stratégie proposée a été validée par une modélisation et une vérification dans le model-checker UPPAAL.

La solution proposée permet à un RCSF de survivre à un crash avéré de son SINK primaire, assurant ainsi la continuité de sa mission, et présente l'avantage d'être réalisable en couche applicative en tant que programme autonome. Néanmoins, plusieurs voies restent à explorer comme travail en perspective, en effet, cette solution peut être étendue afin de gérer un ensemble de SINK incluant un sous ensemble de SINK primaires, et un autre de SINK secondaires, en se basant sur un mécanisme d'élection prenant en compte des critères d'optimisation spécifiques tels que : l'énergie, la durée de vie d'un RCSF, la sécurité etc. bien que cette perspective augmenterait le coût de déploiement, elle permettrait d'offrir une très grande fiabilité, la nature et l'aspect critique de l'application du RCSF devront évidemment en justifier l'usage. D'un autre coté, la modification dynamique de la valeur du délai de garde en fonction du comportement du réseau tout au long de la vie de celui-ci, représente un point essentiel permettant une reprise efficace après une panne du SINK et minimiserait la perte de messages engendrés par le temps de recouvrement d'une panne.

BIBLIOGRAPHIE

[AFBI04] Marcelo Dias de Amorim, Serge Fdida, Farid Benbadis, Luigi Iannone. « Les réseaux de capteurs : état de l'art, exemples, recherche » *notes, Euronetlab*, 2004.

[ALD94] R. Alur and D. Dill. « A theory of timed automata ». *Theoretical Computer Science, Volume 126, Issue 2, 25 April 1994, Pages 183-235*.

[ASSC01] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. "wireless sensor networks : a survey". *Elsevier Science*, 2002.

[BEH04] Gerd Behrmann, Alexandre David, and Kim G. Larsen. « A Tutorial on Uppaal », *Department of Computer Science, Aalborg University*, Denmark.

[BEN96] Johan Bengtsson, Fredrik Larsson. « UPPAAL : A Tool for Automatic Verification of Real-time Systems », *UPPSALA university*, 1996.

[DHM03] Jing Deng, Richard Han and Shivakant Mishra « Enhancing Base Station Security in Wireless Sensor Networks » *University of Colorado, Technical Report CU-CS-951-03*, 2003.

[DN] Dragos Niculescu. «Communication Paradigms for Sensor Networks » *IEEE Communications Magazine*, March 2005.

[EE] Jeremy Elson and Deborah Estrin. « Sensor Networks: A bridge to the physical world » *Wireless Sensor Networks (e-book), 2004 - Springer*.

[FS04] E.Fleury, David Simplot. « Réseaux de capteurs » *CITI/INSA, ARES/INRIA*, 2004.

[HRR05] S Harte, A Rahman, K M Razeeb. « Fault Tolerance In Sensor Networks using Self Diagnosing Sensor Nodes ». *Intelligent Environments, 2005. The IEE International Workshop*.

[HW05] Holger Karl and Andreas Willig. « Protocols and Architectures for Wireless Sensor Networks » *Wiley e-book, Introduction, 2005*.

[IGE] Chalermek Intanagonwiwat, Ramesh Govindan, Deborah Estrin « Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks ». *MobiCom '00 Proceedings of the 6th annual international conference on Mobile computing and networking, New York, USA, 2000*.

[IGEHS] Chalermek Intanagonwiwat, Ramesh Govindan, Deborah Estrin, John Heidemann, and Fabio Silva. « Directed Diffusion for Wireless Sensor Networking » *Networking, IEEE/ACM Transactions*, février 2003.

[INT02] Chalermek Intanagonwiwat. « Directed Diffusion: an application-specific and data-centric communication paradigm for wireless sensor networks » *PhD thesis, université de Californie du sud*, décembre 2002.

[KPS] Farinaz Koushanfar, Miodrag Potkonjak , Alberto Sangiovanni-Vincentelli. « fault tolerance in wireless sensor networks » *Handbook of Sensor Networks e-book, Compact Wireless and Wired Sensing Systems, chapter 36*, 2005.

[MCP] Daniel F. Macedo, Luiz H. A. Correia, Aldri L. dos Santos, Antonio A. F. Loureiro, José Marcos S. Nogueira and Guy Pujolle. « evaluating fault tolerance aspects in routing protocols for wireless sensor networks », *Challenges in Ad Hoc Networking, IFIP International Federation for Information Processing, Springer*, 2006.

[PJ] Jean-Marc Percher, Bernard Jouga. « Détection d'intrusions dans les réseaux ad hoc », *SSTIC'03, 1er Symposium sur la Sécurité des Technologies de l'Information et de la Communication. Rennes, France,* juin 2003.

[SALEH] Iman Saleh, Mohamed Eltoweissy « In-Network Fault Tolerance in Networked Sensor Systems ». *DIWANS '06 Proceedings of the 2006 workshop on Dependability issues in wireless ad hoc networks and sensor networks, ACM New York, NY, USA*, 2006.

[SMZ07] Kazem Sohraby, Daniel Minoli, Taieb Znati « Wireless Sensor Networks : Technology, Protocols, and Applications » *Wiley e-book*, 2007.

[SVB] Luciana Moreira Sa de Souza, Harald Vogt, Michael Beigl. « A Survey on Fault Tolerance in Wireless Sensor Networks » *Rapport technique interne, Faculté des sciences informatiques, Université de Karlsruhe, Allemagne*; 2007.

[Tan02] A. S. Tanenbaum and M. van Steen. « Distributed Systems: Principles and Paradigms. » *Prentice Hall*, 2002

[YC08] Yacine Challal. « Réseaux de capteurs sans fil », université de technologie de Compiègne 2008.

[YMM] M.Yu, H.Mokhtar, and M.Merabti. « Fault Management in Wireless Sensor Networks », School of Computing & Mathematical Science Liverpool John Moores University, *Wireless Communications, IEEE*, dec 2007.

[XIA] Xiaobing Wu and Guihai Chen. « Dual-Sink: Using Mobile and Static Sinks for Lifetime Improvement in Wireless Sensor Networks », *Computer Communications and Networks, 2007. ICCCN 2007. Proceedings of 16th International Conference, Aug 2007.*

[ILK] E. Ilker Oyman and Cem Ersoy. « Multiple Sink Network Design Problem in Large Scale Wireless Sensor Networks », *Communications, 2004 IEEE International Conference*, June 2004.

[CIC] Pietro Ciciriello, Luca Mottola, and Gian Pietro Picco. « Efficient Routing from Multiple Sources to Multiple Sinks in Wireless Sensor Networks » *Wireless Sensor Networks, Lecture Notes in Computer Science, 2007, Volume 4373/2007, 34-50, Springer*.

[PET] Kim G. Larsen Paul Pettersson and Wang Yi. « UPPAAL in a Nutshell », Department of Computer Systems, Uppsala University, Sweden.

[PET99] Paul Pettersson. « Modelling and Verication of Real-Time Systems Using Timed Automata: Theory and Practice », PHD Thesis, Department of Computer Systems Uppsala University Feb 1999.