Table des matières :

Table des figures

Liste des tableaux

Introduction générale

Aujourd'hui, les réseaux sans fil sont de plus en plus populaires du fait de leur facilité de déploiement. Ces réseaux jouent un rôle primordial au sein des réseaux informatiques. Ils offrent des solutions ouvertes pour fournir la mobilité ainsi que des services essentiels là où l'installation d'infrastructures n'est pas possible.

Les réseaux sans fil sont généralement classés selon deux catégories : les réseaux sans fil avec infrastructure fixe qui utilisent généralement le modèle de la communication cellulaire et les réseaux sans fil et sans infrastructure fixe ou les réseaux ad-hoc. Un réseau ad-hoc consiste donc en un grand nombre d'unités mobiles se déplaçant dans un environnement quelconque en utilisant, comme moyen de communication, des interfaces sans fil.

Les principaux problèmes liés à ces réseaux sont la limitation de bande passante, la limitation de sources d'énergie et le caractère « pseudo aléatoire » de la mobilité des hôtes. Par conséquent, les protocoles de routage pour les réseaux classiques ne peuvent pas être directement utilisés dans les réseaux ad-hoc.

Les réseaux ad-hoc ont la particularité de se créer de manière spontanée, de s'auto-organiser et s'auto-administrer. Ils ne reposent sur aucune infrastructure fixe [1]. Les éléments les composant sont en général reliés par des liaisons radio, potentiellement mobiles, et peuvent être amenés à entrer ou sortir du réseau à tout moment. Ils sont caractérisés par la capacité de chaque participant d'agir à la fois comme client et comme routeur du réseau. Si un émetteur n'est pas à portée directe de la machine de destination, les informations devront être transmises de proche en proche, le long d'un chemin établi et maintenu par le réseau. Ils sont traditionnellement utilisés dans les applications militaires, les services d'urgence (tremblement de terre, feux, inondation, etc.).

Dans les réseaux sont fil, on est assisté à la naissance d'un nouveau type de réseaux appelés réseaux de capteurs [2]. Ces réseaux sont composés d'éléments pouvant non seulement communiquer entre eux et calculer, mais également numériser (ou agir sur) l'environnement physique dans lequel ils sont déployés.

Ces réseaux peuvent aussi prendre la forme de réseaux ad hoc et sont typiquement petits et autonomes. Ils ont généralement de faibles capacités en termes d'énergie, de mémoire, de calcul et de communication. Le but des réseaux de capteurs est d'obtenir

une meilleure compréhension d'un environnement physique réel et d'en rendre compte au système demandeur de cette numérisation. Ils sont utiles dans de nombreux domaines (climat, suivi des mouvements de la terre, etc.) [1, 2].

Dans les réseaux sans fil, on donne plus de l'importance à l'acheminement de l'information qui est assuré par des algorithmes de routage. Ces algorithmes de routage doivent prendre en considération les changements de la topologie du réseau, ainsi que d'autres caractéristiques comme la bande passante, le nombre de liens, la limitation d'énergie, etc. En particulier pour les réseaux de capteurs qui se caractérisent par des liens volatiles et des dispositifs fragiles, les protocoles de routage perdent leurs performances quand un lien est perdu ou un dispositif cesse de fonctionner. Dans ce contexte, plusieurs recherches ont été menées notamment pour garantir le routage de l'information de n'importe quel nœud vers la station de base.

L'objectif de ce mémoire est de traiter le problème de tolérance aux pannes dans les réseaux de capteurs pour garantir un routage efficace, surtout ceux à taille importante. Le souci principal est d'assurer la livraison de données à la station de base tout en prolongeant la vie du système. Pour cela, nous avons tout d'abord étudié les performances de LEACH dans un environnement qui n'est pas idéal. Les résultats ont montré que LEACH perd ses performances dans ce type d'environnement. Puis nous avons proposé une version améliorée de LEACH appelée T-LEACH. Dans cette version, nous avons partitionné le réseau en zones tel qu'au niveau de chaque zone il y a un cluster-head et son adjoint. En outre, nous avons conçu deux mécanismes d'acheminement d'informations vers la station de base. Le premier est un algorithme de routage intra-zone alors que le deuxième est inter-zone. Par ailleurs, pour faire face aux pannes au niveau de chaque zone un cluster-head et son adjoint sont élus. Si le cluser-head cesse de fonctionner, son adjoint le remplace dans sa mission.

Ce document s'articule autour de quatre chapitres. Le premier chapitre décrit l'architecture d'un capteur et présente les principes et les caractéristiques des réseaux de capteurs aussi que ses domaines d'application. Dans le deuxième chapitre, nous présentons la tolérance aux pannes dans les réseaux de capteurs et sa classification. Nous commençons tout d'abord par donner une définition à la tolérance aux pannes, et sa procédure générale en se basant sur un exemple. Dans le troisième chapitre, nous donnons un état de l'art sur les protocoles de routage dans les réseaux de capteurs et

nous focalisons notre présentation sur les protocoles de routage tolérants aux pannes. Le quatrième chapitre présente LEACH qui est un protocole de routage conçu aux réseaux de capteurs mais qui n'est pas tolérant aux pannes. Dans notre contribution, nous avons amélioré ce protocole de telle sorte qu'il soit tolérant aux pannes. Pour cela, nous avons proposé un algorithme distribué de partitionnement d'un réseau de capteurs en zones. Ceci est dans le but de minimiser les échanges entre les nœuds afin d'économiser leur énergie. Ensuite, nous avons présenté les phases de construction des tables de routage.

Chapitre I

Concepts des réseaux de capteurs

Chapitre I

Concepts des réseaux de capteurs

I .1 Introduction

Dans ce chapitre, nous allons présenter les réseaux de capteurs sans fil : leurs architectures de communication et leurs applications. Nous allons discuter également les principaux facteurs et contraintes qui influencent la conception des réseaux de capteurs sans fil.

I.2 Définition d'un capteur

Un capteur est un dispositif qui transforme l'état d'une grandeur physique observée en une grandeur utilisable, exemple : une tension électrique, une hauteur de mercure, une intensité, la déviation d'une aiguille….

Le capteur se distingue de l'instrument de mesure par le fait qu'il ne s'agit que d'une simple interface entre un processus physique et une information manipulable. Par opposition, l'instrument de mesure est un appareil autonome se suffisant à lui-même. Il dispose donc d'un affichage ou d'un système de stockage des données. Ce qui n'est pas forcément le cas du capteur.

Les capteurs sont les éléments de base des systèmes d'acquisition de données. Leur mise en œuvre est du domaine de l'instrumentation [4].

I.2.1 Classification des capteurs

Les *capteurs* ont plusieurs modes de classification :

I.2.1.1 Apport énergétique

a) Capteurs passifs

Ils n'ont pas besoin d'apport d'énergie extérieure pour fonctionner (exemple : thermistance, potentiomètre, thermomètre à mercure…). Ce sont des capteurs modélisables par une impédance. Une variation du phénomène physique étudié (mesuré) engendre une variation de l'impédance.

b) Capteurs actifs

Ils sont constitués d'un ou d'un ensemble de transducteurs alimentés (exemple :chronomètre mécanique, jauge d'extensomètrie appelée aussi jauge de

contrainte, gyromètre…). Ce sont des capteurs que l'on pourrait modéliser par des générateurs comme les systèmes photovoltaïques et électromagnétiques. Ainsi ils génèrent soit un courant, soit une tension en fonction de l'intensité du phénomène physique mesuré.

I.2.1.2 Type de sortie

Les capteurs peuvent aussi faire l'objet d'une classification par type de sortie:

a) Capteurs analogiques

Le signal des capteurs numériques peut être du type :

- sortie tension
- sortie courant
- règle graduée
- ...

Quelques capteurs analogiques typiques :

- capteur à jauge de contrainte
- LVDT

b) Capteurs numériques

Le signal des capteurs numériques peuvent être du type :

- train d'impulsions, avec un nombre précis d'impulsions ou avec une fréquence précise
- code numérique binaire
- bus de terrain
- ...

Quelques capteurs numériques typiques :

- les capteurs incrémentaux.
- les codeurs absolus.

I.3 Qu'est-ce un réseau de capteurs ?

Un réseau de capteur sans fil (Wireless Sensor Network: WSN) est un type particulier de réseau ad-hoc défini par un ensemble coopérant de nœuds capteurs dispersés dans une zone géographique appelée zone de captage afin de surveiller un phénomène et récolter ses données d'une manière autonome.

Un réseau de capteurs se compose de deux types de nœuds : des simples capteurs et des collecteurs d'informations appelés puits.

Le capteur est composé d'un microcontrôleur et d'un circuit radio:

- Le microcontrôleur est simple et peut être embarqué aisément. Cet appareil doit répondre à l'exigence d'une faible consommation d'énergie tout en ayant la possibilité d'exécuter de simples opérations et de posséder une mémoire permettant d'emmagasiner de l'information. L'appareil doit aussi présenter la possibilité d'avoir un état oisif durant lequel il consomme une quantité d'énergie infinitésimale. Ces états oisifs peuvent parfois durer très longtemps. Le capteur peut se réveiller seulement pour capter la grandeur physique à mesurer et aussi pour effectuer des opérations de réseaux comme dialoguer avec des capteurs voisins ou relayer l'information provenant d'autres capteurs.
- Le circuit radio assure la communication du capteur avec d'autres appareils via des liens radios. Ces derniers ont facilité l'implantation massive de capteurs et ont offert une indépendance précieuse car il a réduit les coûts du câblage et de l'ingénierie nécessaire pour les installations passées. Grâce à la communication sans fil, un installateur peut déposer facilement des capteurs sans se soucier de la complexité des opérations pour les atteindre afin de relever les mesures. Il suffit d'être dans le champ de couverture radio pour transmettre ou recevoir l'information requise.

Avec ses capacités de traitement et de mémorisation, le capteur peut devenir un nœud actif dans un réseau relativement large. Lorsque le nombre de capteurs devient conséquent, la communication en réseau devient indispensable. Il n'est en effet alors plus possible d'atteindre un capteur directement par un câble ou même par une connexion radio. C'est là alors qu'on peut parler de véritables réseaux de capteurs capables de s'auto-configurer et de s'auto-organiser de manière dynamique. Ces propriétés offrent un très large spectre d'applications, notamment dans les domaines militaires, de l'environnement, de l'écologie, etc.

Dans un réseau de capteurs, une autre entité, appelé puits, détient un rôle très important. Cette entité généralement possède des capacités supérieures en termes de puissance de traitement, de capacité de mémoire et d'autonomie d'énergie. Elle permet de collecter l'information en provenance des capteurs et apporte un soutien très fort au fonctionnement du réseau. Elle peut localement assurer des fonctions centrales dans le routage, l'agrégation des données, la configuration des nœuds ou encore l'organisation de l'ordre de transmission et de réception des différents capteurs avoisinants.

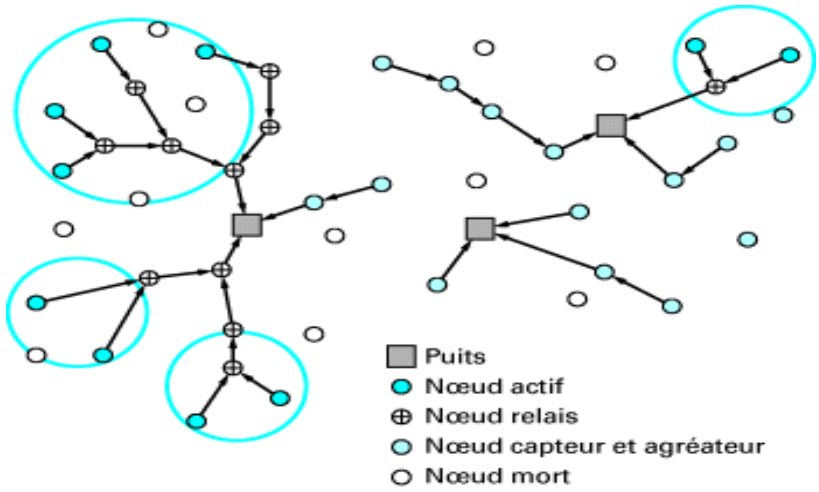

Figure I-1 : Exemple de réseau de capteurs.

La figure I.1 illustre un exemple de réseau de capteurs où nous pouvons distinguer différents scénarios possibles. Sur cette figure, nous pouvons repérer :

- des nœuds hors service car leurs batteries sont usées ;
- des nœuds actifs en étant soit source de l'information ou servant comme relais pour atteindre le puits de la collecte d'informations ;
- des nœuds endormis qui se trouvent dans leur état oisif.

Les nœuds actifs (en bleu sur la figure1) transmettent l'information qu'ils détiennent. Lors de la transmission, le nœud doit s'assurer qu'il est seul à occuper l'espace de transmission afin d'éviter des interférences sur les autres transmissions. Sa transmission se fait en diffusant l'information et seul le nœud dont l'adresse apparaît comme destination récupère l'information pour la relayer à son tour en la transmettant au nœud suivant. Ces nœuds sont appelés les nœuds relais. Le relais est un nœud indispensable dans un réseau large. En effet, il est impossible de couvrir tous les nœuds du réseau par une seule transmission. L'atténuation des signaux radio et d'autres phénomènes comme l'évanouissement ou le multi-chemin font que le signal, à partir d'une certaine distance, peut se dégrader et contenir un nombre important d'erreurs le rendant incompréhensible.

C'est pourquoi, relayer l'information permet de la récupérer dans un nœud intermédiaire, de la régénérer avec une énergie remise à neuf redonnant ainsi à l'information la possibilité d'atteindre une destination plus lointaine. Ainsi, la transmission de proche en proche permet de joindre le nœud final (le nœud puits mentionné en gris sur la figure I.1). Certains capteurs captent la grandeur physique et la gardent afin d'agréger l'information mesurée qui sera envoyée plus tard pour réduire la consommation d'énergie. Ces nœuds sont indiqués en bleu clair sur la figure I.1. L'agrégation, la synchronisation, la manière de relayer l'information, etc. sont toutes des procédures qui doivent être bien pensées afin d'optimiser la durée de vie du réseau qui est intimement liée au nombre de nœuds considérés comme finis car leur batterie est épuisée. Ces nœuds sont indiqués avec la couleur blanche sur la figure I.1.

Pour que l'acheminement de l'information se fasse de manière harmonieuse, on peut distinguer plusieurs fonctionnalités.

- **Endormir et réveiller les nœuds**

  Ici, l'algorithmique reste complexe car endormir un nœud voudrait dire que le nœud n'est plus là pour servir de relais et le processus d'endormissement doit prendre en considération le fait d'éviter qu'un groupe de capteurs se retrouve isolé du reste du réseau car tous ses nœuds relais sont en état oisif.

- **Accéder à la transmission sans collision ni interférence avec les voisins**

  La transmission physique se fait sur l'interface radio qui est partagée avec les capteurs voisins. Les ondes peuvent perturber à la fois les nœuds qui se retrouvent dans la couverture radio et ceux qui sont en dehors mais pas suffisamment loin de l'émetteur pour que leurs signaux soient largement atténués. Un protocole d'accès au support de transmission doit offrir au capteur la possibilité de transmettre à la demande, et parfois une contrainte d'accès déterministe s'impose lorsqu'il s'agit d'applications présentant des risques élevés. La conception du protocole d'accès doit tenir compte des contraintes énergétiques des nœuds capteurs car cela ne doit pas user les batteries inutilement. Il faut savoir que la transmission, la réception, l'écoute et l'interférence représentent les fonctions les plus gourmandes en énergie et ce loin devant d'autres fonctions comme l'accès à la mémoire, la mesure ou le traitement de l'information.

- **Acheminer l'information :**

Acheminer l'information utilise ce qu'on appelle le routage qui permet de faire véhiculer l'information du capteur vers le nœud puits destination. Le routage doit être dynamique et distribué. Dynamique car nous n'avons jamais les mêmes routes pour cause d'endormissement et de disparition de certains capteurs. Il est également distribué pour ne pas user toujours les mêmes relais. Ici, le capteur endosse des responsabilités importantes car il est considéré, à certains moments, comme étant un nœud routeur capable à la fois de réfléchir pour décider du prochain relais et d'intégrer dans cette décision des paramètres importants comme l'augmentation de la durée de vie du réseau de capteurs et la réduction de la consommation de sa propre énergie ainsi que celle de tous les nœuds de son réseau. L'aspect distribué du routage permet au réseau de capteurs d'accéder au principe du passage à l'échelle où localement l'évolution du nombre de capteurs dans une zone n'influence en aucun cas les autres nœuds du réseau.

- **Gérer l'énergie de façon à réduire le nombre de capteurs qui disparaissent :**

L'énergie est la contrainte la plus importante dans un capteur. Généralement, Les capteurs sont déployés dans des environnements hostiles. Par conséquent, aller recharger les capteurs en énergie ou remplacer leurs batteries devient une opération complexe et coûteuse. Par exemple, déployer des capteurs dans le pôle nord pour surveiller le réchauffement climatique est une tâche trop coûteuse si on met sur place un personnel pour changer les batteries de temps à autre. Il est donc important de comprendre que la durée de vie d'un capteur est relative à l'autonomie de son support d'énergie. Ainsi, toute opération imaginée ou envisagée doit être évaluée en termes de consommation d'énergie. Gérer l'énergie dans un réseau de capteurs peut devenir une opération très délicate selon le but à atteindre. Parfois, on favorise le but à atteindre au détriment de la consommation d'énergie surtout pour les applications de type event-driven et des fois on privilégie l'augmentation de la durée de vie d'un capteur auquel on a affecté une opération délicate ou urgente.

Toutes ces opérations sont organisées dans le but d'optimiser le fonctionnement du réseau de capteurs. Ce bon fonctionnement tient en répondant à un nombre de contraintes fixées par la fonction à laquelle est destiné le réseau de capteurs [5].

I.4 Architecture physique d'un capteur

La figure I. 2 illustre l'architecture physique d'un capteur qui est composée de trois unités:

- **l'unité d'acquisition** : est composée d'un capteur qui va obtenir des mesures numériques sur les paramètres environnementaux et d'un convertisseur Analogique/Numérique qui va convertir l'information relevée et la transmettre à l'unité de traitement.
- **l'unité de traitement** : est composée de deux interfaces, une interface pour l'unité d'acquisition et une interface pour l'unité de transmission. Cette unité est également composée d'un processeur et d'un système d'exploitation spécifique [6]. Elle acquiert les informations en provenance de l'unité d'acquisition et les envoie à l'unité de transmission.
- **l'unité de transmission** : est responsable de toutes les émissions et réceptions de données via un support de communication radio. Elle peut être de type optique (comme dans les capteurs Smart Dust [7]), ou de type radiofréquence (MICA2 [8], par exemple).

Ces trois unités sont alimentées par une batterie comme le montre la figure ci-dessous:

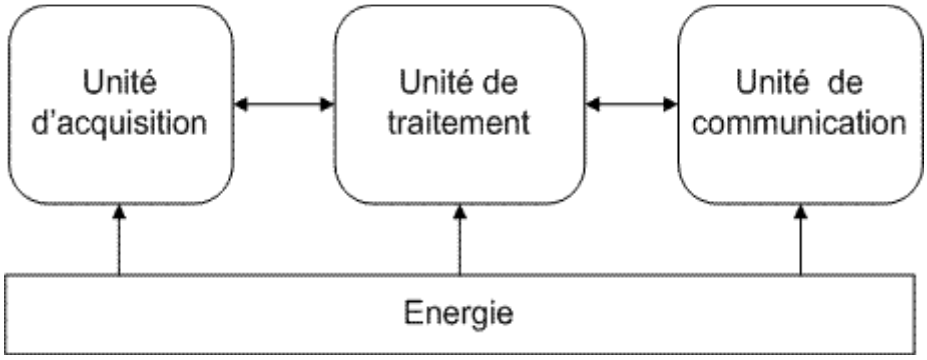

Figure I-2 : Architecture physique d'un capteur.

I.5 Architecture de communication dans les WSN

La pile des protocoles utilisée par un nœud collecteur ou capteur est donnée dans la figure I.3. Cette pile combine l'énergie et le routage, intègre les données avec les protocoles réseaux, communique efficacement à travers un medium sans fil et promeut des efforts coopératifs entre les nœuds capteurs.

La pile consiste en couches application, transport, réseau, liaison de données, physique et trois plans de gestion d'énergie, de mobilité et de tâche.

Les plans de gestion d'énergie, de mobilité et de tâche contrôlent l'énergie, mouvement et la distribution de tâche au sein d'un nœud capteur. Ces plans aident les nœuds capteurs à coordonner la tâche de captage et minimiser la consommation d'énergie. Ils sont donc nécessaires pour que les nœuds capteurs puissent collaborer ensemble, acheminer les données dans un réseau mobile et partager les ressources entre eux en utilisant efficacement l'énergie disponible. Ainsi, le réseau peut prolonger sa durée de vie.

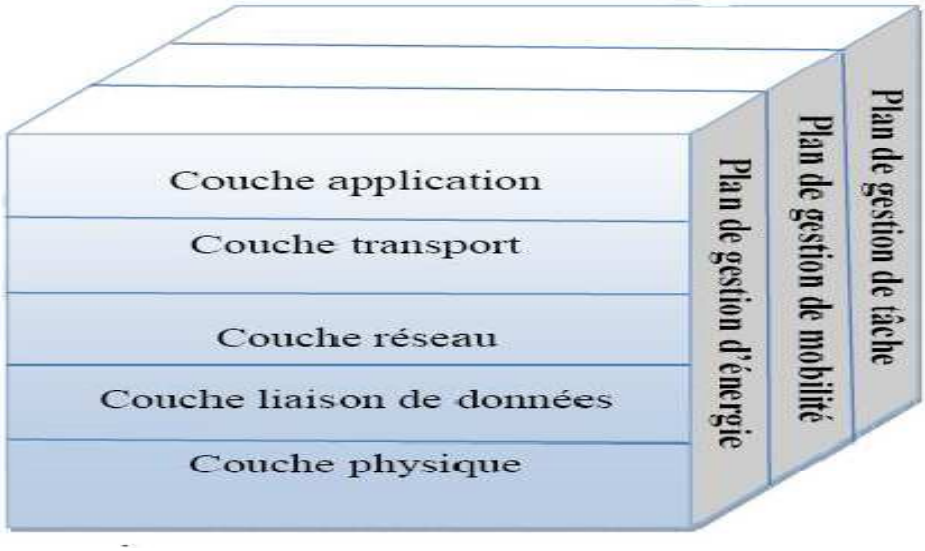

Figure I-3 : Pile protocolaire dans les réseaux de capteurs.

I.6 Evolution des réseaux de capteurs

L'avenir à court, moyen et long terme prévoit une très large place aux réseaux de capteurs. Jusqu'à récemment, ces réseaux étaient dans une phase expérimentale et de perfectionnement. Les agences de financement de la recherche ainsi que les centres de recherche et de développement industriels y consacrent beaucoup de budgets afin de développer des produits, standards, protocoles, etc. dans un but de faciliter leur mise en place. Actuellement, on peut utiliser les réseaux de capteurs dans plusieurs milieux, tels que la maison intelligente, le milieu hospitalier, le milieu industriel, le milieu militaire...

La recherche continue pour perfectionner le fonctionnement des futurs réseaux de capteurs. De nombreux travaux sont effectués pour résoudre les trois difficultés majeures auxquelles sont confrontés les capteurs : l'énergie, la puissance d'émission, la capacité de stockage et de calcul.

Ces données sont clairement en contradiction avec l'ambition des réseaux de capteurs de vouloir passer à l'échelle et celle de traiter le plus grand nombre de problèmes de mesures. C'est pourquoi, dans ce domaines, il est clairement besoin de déployer de l'algorithmique performante, des conceptions optimales et de la technologie hautement performante.

Les réseaux de capteurs se font de plus en plus indispensables dans un monde où les ressources deviennent davantage rares et où l'optimisation reste pour l'instant la seule issue pour faire face aux demandes incessantes de l'utilisation de ces ressources. Les réseaux de capteurs permettent d'offrir un support à cette optimisation en relevant les informations nécessaires, en prenant les décisions adéquates et en activant les actions appropriées.

La miniaturisation du matériel et la prolifération des moyens de connexions associées à l'augmentation des capacités de calcul et de mémoire en informatique ont permis aux réseaux de capteurs d'exister, et cela à une échelle très large au point d'accomplir les tâches les plus complexes pour l'humain. Ainsi, on peut les retrouver désormais dans l'armement, le sauvetage, la sauvegarde de l'environnement, la médecine, etc.

Les réseaux de capteurs ne peuvent pas être considérés comme un type de réseau simple car leur conception peut dépendre de l'application les utilisant. Cette conception reste complexe aujourd'hui et la recherche scientifique doit encore progresser pour améliorer la flexibilité et les performances des composants et des protocoles liés à ces réseaux.

L'article offre une vision descendante (*top-down*) des réseaux de capteurs, où l'on part d'une application et de l'étude de ses contraintes pour ensuite concevoir l'architecture du réseau de capteurs adéquat. Cette vision induit une difficulté supplémentaire pour ces réseaux comparés à d'autres supports de communication plus traditionnels. Plusieurs protocoles ont été conçus pour assurer les fonctionnalités de communication entre capteurs voisins et de relais pour atteindre des destinations plus lointaines que le voisinage. La propriété nouvelle offerte au capteur de pouvoir agir seul localement pour s'entendre avec son voisinage et organiser l'accès à la ressource et la collaboration a permis l'ouverture de l'utilisation du réseau de capteurs à grande échelle. Ainsi, tout problème lié à un grand nombre de mesures devient facile à résoudre grâce aux réseaux de capteurs.

I.7 Domaines d'application

La taille de plus en plus réduite des micro-capteurs, le coût de plus en plus faible, la large gamme des types de capteurs disponibles (thermique, optique, vibrations,...) ainsi que le support de communication sans fil utilisé, permettent aux réseaux de capteurs d'envahir plusieurs domaines d'application. Ils permettent aussi d'étendre les applications existantes et de faciliter la conception d'autres systèmes tels que le contrôle et l'automatisation des chaînes de montage. Les réseaux de capteurs ont le potentiel de révolutionner la manière même de comprendre et de construire les systèmes physiques complexes. Ils peuvent aussi se révéler très utiles dans de nombreuses applications lorsqu'il s'agit de collecter et de traiter des informations provenant de l'environnement. Parmi les domaines où ces réseaux peuvent offrir les meilleures contributions, nous citons les domaines : militaire, environnemental, domestique, santé, sécurité, etc [9]. Des exemples d'applications potentielles dans ces différents domaines sont exposés ci-dessous.

I.7.1 Applications militaires

Comme dans le cas de plusieurs technologies, le domaine militaire a été un moteur initial pour le développement des réseaux de capteurs. Le déploiement rapide, le coût réduit, l'auto-organisation et la tolérance aux pannes des réseaux de capteurs sont des caractéristiques qui rendent ce type de réseaux un outil appréciable dans un tel domaine. Comme exemple d'application dans ce domaine, on peut penser à un réseau de capteurs déployé sur un endroit stratégique ou difficile d'accès, afin de surveiller toutes les activités des forces ennemies, ou d'analyser le terrain avant d'y envoyer des troupes (détection d'agents chimiques, biologiques ou de radiations). Des tests concluants ont déjà été réalisés dans ce domaine par l'armée américaine dans le désert de Californie [10].

I.7.2 Applications liées à la sécurité

Les altérations dans la structure d'un bâtiment, suite à un séisme ou au vieillissement, pourraient être détectées par des capteurs intégrés dans les murs ou dans le béton, sans alimentation électrique ou autres connexions filaires. Les capteurs doivent s'activer périodiquement et peuvent ainsi fonctionner durant des années, voire des décennies.

Un réseau de capteurs de mouvements peut constituer un système d'alarme distribué qui servira à détecter les intrusions sur un large secteur. Déconnecter le système ne

serait plus aussi simple, puisqu'il n'existe pas de point critique. La surveillance de voies ferrées pour prévenir des accidents avec des animaux et des êtres humains peut être une application intéressante des réseaux de capteurs. La protection des barrages pourrait être accomplie en y introduisant des capteurs. La détection prompte de fuites d'eau permettrait d'éviter des dégâts. Les êtres humains sont conscients des risques et attaques qui les menacent. Pour cela, ils mettent à disposition toutes les ressources humaines et financières nécessaires pour leur sécurité. Cependant, des failles sont toujours présentes dans les mécanismes de sécurisation appliqués aujourd'hui, sans oublier leur coût très élevé. L'application des réseaux de capteurs dans le domaine de la sécurité pourrait diminuer considérablement les dépenses financières consacrées à la sécurisation des lieux et à la protection des êtres humains tout en garantissant de meilleurs résultats.

I.7.3 Applications environnementales

Des capteurs dispersés à partir d'un avion dans une forêt peuvent signaler un éventuel début d'incendie dans le champ de captage; ce qui permettra une meilleure efficacité pour la lutte contre les feux de forêt. Dans les champs agricoles, les capteurs peuvent être semés avec les graines. Ainsi, les zones sèches seront facilement identifiées et l'irrigation sera donc plus efficace. Sur les sites industriels, les centrales nucléaires ou dans les pétroliers, des capteurs peuvent être déployés pour détecter des fuites de produits toxiques (gaz, produits chimiques, éléments radioactifs, pétrole, etc.) et alerter les utilisateurs dans un délai suffisamment court pour permettre une intervention efficace. Une grande quantité de capteurs peut être déployée dans une forêt ou dans un environnement de conservation de la faune afin de recueillir des informations diverses sur l'état du milieu naturel et sur les comportements de déplacement. Par exemple, l'université de Pise en Italie a réalisé des réseaux de capteurs pour le contrôle des parcs naturels (feux, animaux,..). Il est ainsi possible "d'observer", sans déranger, des espèces animales difficiles à étudier dans leur environnement naturel et de proposer des solutions plus efficaces pour la conservation de la faune. Les éventuelles conséquences de la dispersion en masse des micro-capteurs dans l'environnement ont soulevé plusieurs inquiétudes. En effet, chaque micro-capteur est doté d'une batterie qui contient des métaux nocifs. Néanmoins, le déploiement d'un million de capteurs de 1 mm^3 chacun ne représente qu'un volume total d'un litre. Même si tout ce volume était constitué de batteries, cela n'aurait pas des répercussions désastreuses sur l'environnement.

I.7.4 Applications médicales

On pourrait imaginer que dans le futur, la surveillance des fonctions vitales de l'être humain serait possible grâce à des micro-capteurs qui pourront être avalés ou implantés sous la peau. Actuellement, des micro-caméras qui peuvent être avalées existent. Elles sont capables, sans avoir recours à la chirurgie, de transmettre des images de l'intérieur d'un corps humain avec une autonomie de 24 heures. Les auteurs d'une récente étude, présentent des capteurs qui fonctionnent à l'intérieur du corps humain pour traiter certains types de maladies. Leur projet actuel est de créer une rétine artificielle composée de 100 micro-capteurs pour corriger la vue. D'autres ambitieuses applications biomédicales sont aussi présentées, tel que : la surveillance du niveau de glucose, le monitoring des organes vitaux ou la détection de cancers à une étape précoce. L'utilisation des réseaux de capteurs dans le domaine de la médecine pourrait apporter une surveillance permanente des patients et une possibilité de collecter des informations physiologiques de meilleure qualité, facilitant ainsi le diagnostic de quelques maladies [11].

I.7.5 Applications commerciales

Il est possible d'intégrer des nœuds capteurs au processus de stockage et de livraison. Le réseau ainsi formé, pourra être utilisé pour connaître la position, l'état et la direction d'un paquet ou d'une cargaison. Il devient alors possible pour un client qui attend la réception d'un paquet, d'avoir un avis de livraison en temps réel et de connaître la position actuelle du paquet. Pour les entreprises manufacturières, les réseaux de capteurs permettront de suivre le procédé de production à partir des matières premières jusqu'au produit final livré. Grâce aux réseaux de capteurs, les entreprises pourraient offrir une meilleure qualité de service tout en réduisant leurs coûts. Dans les immeubles, le système de climatisation peut être conçu en intégrant plusieurs micro-capteurs dans les tuiles du plancher et les meubles. Ainsi, La climatisation pourra être déclenchée seulement aux endroits où il y a des personnes présentes et seulement si c'est nécessaire. Le système distribué pourra aussi maintenir une température homogène dans les pièces. Utilisée à grande échelle, une telle application permettrait de réduire la demande mondiale en énergie réduisant du même coup les gaz à effet de serre. Rien que pour les États-Unis, on estime cette économie à 55 milliards de dollars par an avec une diminution de 35 millions de tonnes des émissions de carbone dans l'air. Ainsi, dans un contexte mondial où le réchauffement de la planète devient une préoccupation

grandissante, une telle conséquence environnementale serait un pas dans la bonne direction [12].

I.8 Collection des informations

Il y a deux méthodes pour collecter les informations d'un réseau de capteurs.

- A la demande

Lorsque l'on souhaite avoir l'état de la zone de couverture à un moment t, le puits émet des brodcasts vers toute la zone pour que les capteurs remontent leur dernier relevé vers le puits. Les informations sont alors acheminées par le biais d'une communication multi-sauts comme ils sont indiqués sur la figure I.4.

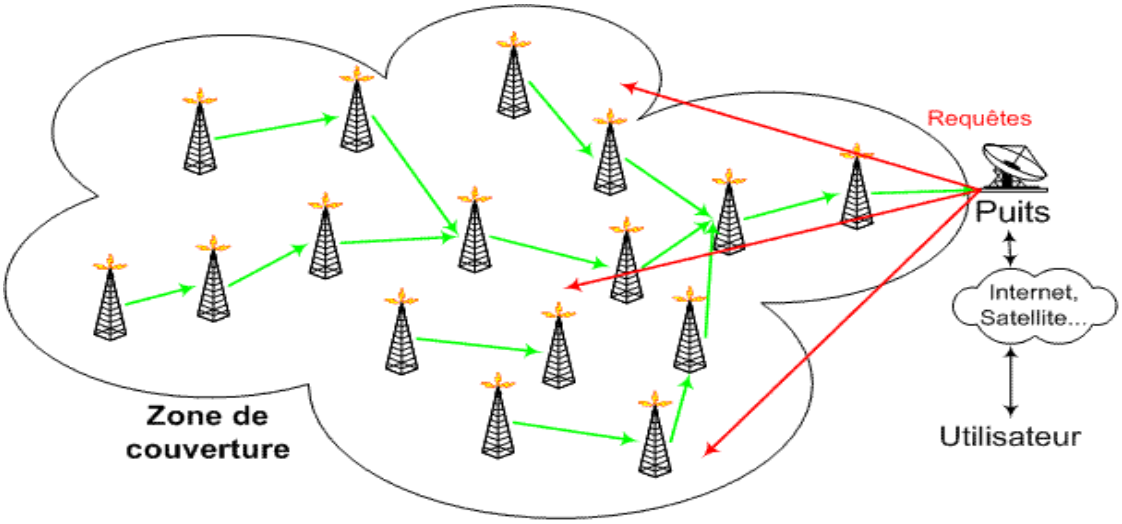

Figure I-4 : Collection des informations a la demande.

Suite à un événement

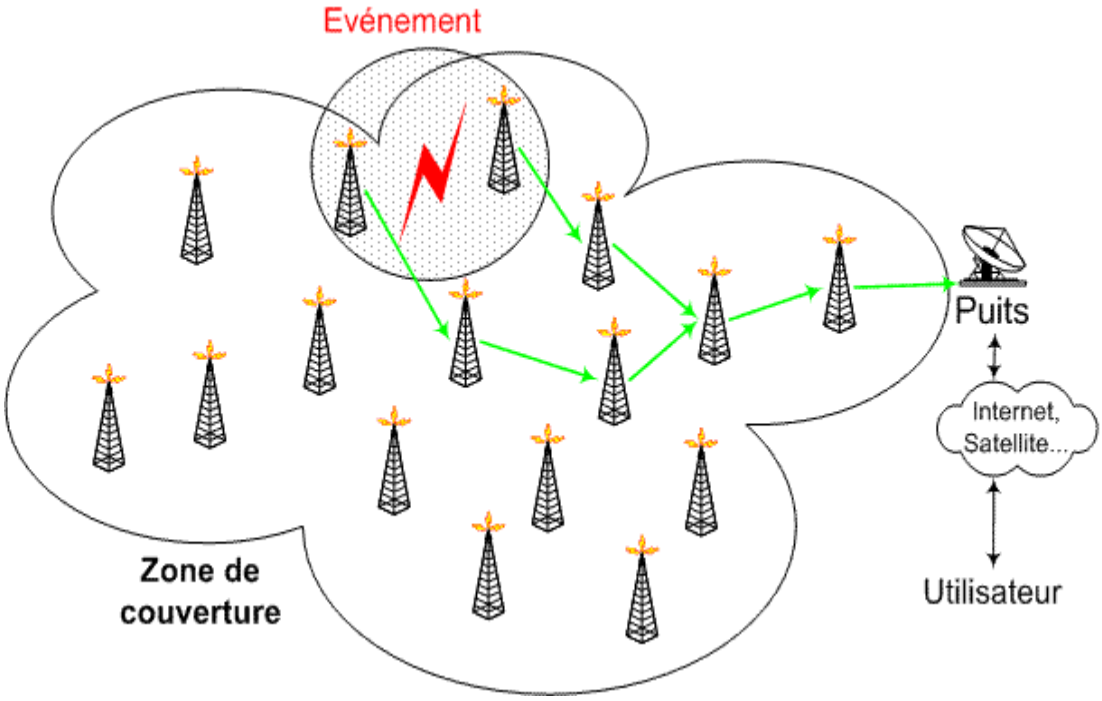

Figure I-5 : Collection des informations suite à un événement.

Un événement se produit en un point de la zone de couverture (changement brusque de température, mouvement...), les capteurs situés à proximité remontent alors les informations relevées et les acheminent jusqu'au puits comme ils sont indiqués sur la figure I.5.

I.9 Contraintes dans la conception d'un réseau de capteurs

Les réseaux de capteurs diffèrent des réseaux classiques où l'on peut être relativement générique et définir seulement un certain nombre de classes de service pour satisfaire le maximum de besoins. Ici, les contraintes sont plus nombreuses et empêchent la création d'un type spécifique du réseau de capteurs. Sans être exhaustif, voici une liste de contraintes possibles lors de la conception d'un réseau de capteurs.

I.9.1 Contraintes liées à l'application

Il est impossible aujourd'hui de créer un réseau de capteurs capable de répondre aux besoins de toutes les applications potentielles. On peut relever des mesures pour une infinité de situations et dans des environnements très variables tout en ayant une concentration faible ou forte des capteurs. Dans certains cas, il existe des applications qui nécessitent un grand nombre de capteurs pour être mises en place. La difficulté réside alors dans la recherche d'un dénominateur commun à toutes ces applications ce qui est pour l'instant très complexe et relève de l'impossible. C'est pourquoi,

l'application devient le principal paramètre lors de la conception de protocoles très spécifiques pour que le fonctionnement des capteurs produise le résultat attendu par l'application en question.

I.9.2 Contrainte énergétique

L'énergie est considérée comme la contrainte principale dans un réseau de capteurs. Déjà, comme pour tout réseau sans fil, il est important de tenir compte de cette contrainte car la plupart des machines fonctionnent sur batterie. Après la décharge de la batterie, l'utilisateur est obligé de trouver une source électrique pour la recharger.

Cependant, dans les réseaux de capteurs, il est pratiquement impossible de recharger de par le nombre élevé de capteurs déployés et de par la difficulté de l'environnement dans lesquels ils peuvent se trouver. On parle alors pour la pile ou la batterie d'âme du capteur. Une fois vide, le capteur est considéré comme mort ou hors service. L'objectif à atteindre devient l'augmentation de la durée de vie du réseau de capteurs. Ce paramètre peut être défini sous différentes formes telles que la consommation globale de tous les capteurs ou l'évitement qu'un capteur important perde son énergie ou la perte de la connectivité du réseau, etc.

I.9.3 Contraintes liées aux déterminismes

La plupart des réseaux de capteurs sont destinés à être déployés dans des environnements hostiles sur des sites industriels importants ou à opérer pendant des scénarios de crises. L'information que le capteur mesure doit parfois atteindre le collecteur d'informations en un temps borné bien défini. Au-delà de ce temps, l'information est considérée comme périmée ou non existante. Atteindre le déterminisme sur un réseau de capteurs sans fil n'est pas une tâche évidente. La raison vient du fait que pratiquement tous les standards de communication sans fil aujourd'hui utilisent des méthodes probabilistes pour accéder à cette interface radio.

I.9.4 Contraintes de passage à l'échelle

Le passage à l'échelle (scalability) indique que le réseau est suffisamment large et peut croître de manière illimitée. En d'autres termes, quand on passe à l'échelle, il est trop tard pour effectuer des mises à jour radicales au réseau. À chaque nouvel ajout, on doit prendre en considération les services existants et assurer leur pérennité. De plus, gérer un grand réseau par des humains devient une tâche difficile voire impossible à réaliser. Pour pouvoir opérer quand on passe à l'échelle, il faut que les capteurs soient

capables de s'auto-configurer seuls. L'auto-configuration peut aller de la simple attribution d'un identifiant jusqu'à l'application du protocole pour le bon fonctionnement du nœud dans son environnement. L'algorithmique distribué est la science la plus adaptée pour résoudre les problèmes du passage à l'échelle.

I.9.5 Contraintes liées à la qualité de service

La notion de qualité de service est légèrement différente ici de celle déployée dans les réseaux classiques. Souvent on parle de haut débit ou de faible débit, etc. Ici, avec des petits débits on peut parfois atteindre la qualité exigée. La qualité se définit par la capacité d'interpréter l'information collectée par le puits. Il n'existe donc pas de définition objective de la qualité. En fonction du réseau et du type de mesure, la qualité est alors précisée.

I.9.6 Contraintes liées à la protection de l'information

Comme pour tout réseau sans fil, l'information circule sur une interface partagée et non dédiée. N'importe quel intrus peut alors soit récupérer l'information, soit la modifier ou la rendre inexploitable. C'est pourquoi des mesures de sécurité doivent être mise en place pour protéger l'information. Cependant, tous les mécanismes de sécurité sont créés pour des réseaux où les nœuds disposent d'une forte capacité de traitement, ce qui n'est pas le cas des capteurs. À ce jour, très peu de solutions sont adaptées aux capteurs en termes de sécurité.

I.9.7 Contraintes liées à l'environnement

Les capteurs interagissent avec l'environnement où ils mesurent leurs grandeurs physiques. De façon générale, ces mesures sont relevées à des instants relativement espacés dans le temps puis soudainement, soit pour des raisons de catastrophe ou d'événement exceptionnel, ils se mettent en mode de forte fréquence de mesures et envoient de l'information en rafale. Il faut alors préparer le réseau à supporter ce type d'événement rare mais largement consommateur de ressources et sujet à des situations de congestions et de difficultés majeures.

I.9.8 Contraintes de simplicité

Enfin proposer des protocoles et des mécanismes simples et légers doit être la marque de fabrique du réseau de capteurs. Ces derniers sont de machines largement plus faibles qu'une machine de bureau ou même que des téléphones portables.

I.10 Conclusion

Les réseaux de capteurs sans fil se propagent dans plusieurs domaines d'application. Ils sont devenus indispensables pour les mesures de certaines grandeurs physiques telles que la température, l'humidité, la vibration, etc ou physiologiques. Cependant, les pannes sont inévitables dans ce type de réseaux. Ces pannes peuvent avoir des conséquences catastrophiques. De ce fait, il est nécessaire de concevoir la partie tolérance aux pannes dans la plupart des protocoles.

Le chapitre suivant est consacré pour détailler la notion de tolérance aux pannes dans les réseaux de capteurs sans fil et son utilité.

Chapitre II

La tolérance aux pannes dans les réseaux de capteurs

Chapitre 2

La tolérance aux pannes dans les réseaux de capteurs

II.1 Introduction

La limitation d'énergie dans les capteurs sans fil, et les environnements hostiles dans lesquels ils pourraient être déployés, sont des facteurs qui rendent ce type de réseaux très vulnérables. Ainsi la perte de connexions sans fil peut être due à une extinction d'un capteur suite à un épuisement de sa batterie, ou tout simplement à une destruction physique accidentelle ou intentionnelle par un ennemi.

Par ailleurs, l'absence de sécurité physique pour ce type de capteurs, et la nature vulnérable des communications radios sont des caractéristiques qui augmentent les risques de pannes sur ce type de réseau. Etant donné que les réseaux de capteurs reposent sur des protocoles de communication ad hoc, il est donc nécessaire de considérer la tolérance aux pannes comme critères indispensable dans la conception de ces protocoles.

Ce chapitre s'articulera sur la notion de tolérance aux pannes dans les réseaux de capteurs où nous commencerons par sa définition. Après une classification des pannes dans les systèmes distribués, un exemple illustrera davantage ce concept dans les réseaux de capteurs. Par la suite, une classification des protocoles tolérants aux pannes sera présentée selon différents critères.

II.2 Définition de la tolérance aux pannes

Afin d'assurer la communication entre le nœud collecteur et les autres nœuds d'un réseau de capteurs, les protocoles de routage sont basés sur la communication multi-sauts. Chaque nœud joue alors, en plus du rôle de source de données, le rôle d'un routeur. Toutefois, ces nœuds sont sujets à de nombreuses pannes, dues principalement à l'épuisement des batteries et aux destructions physiques (par exemple, suite à un écrasement par des animaux). Ainsi, la panne de nœuds entraîne la perte des liens de communication et donc un changement significatif dans la topologie globale du réseau. Ceci peut affecter d'une façon considérable la connectivité du réseau et diminuer, en conséquence, sa durée de vie.

La propriété de tolérance aux pannes est définie par l'aptitude du réseau à maintenir ses fonctionnalités, en cas de panne de certains de ses nœuds. Elle vise donc à minimiser l'influence de ces pannes sur la tâche globale du réseau [13].

II.3 Procédure générale de tolérance aux pannes

La conception d'une procédure pour la tolérance aux pannes dépend de l'architecture et des fonctionnalités du système. Cependant, certaines étapes générales sont exécutées dans la plupart des systèmes [14] comme s'est illustré dans la figure II-1

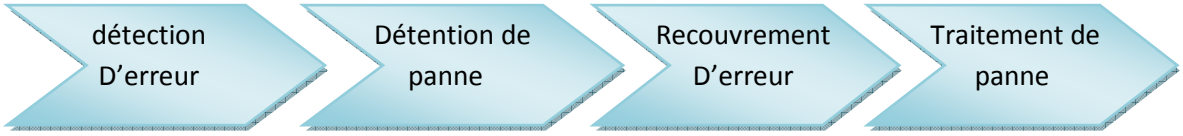

Figure II-1 : Procédure générale de tolérance aux pannes.

II.3.1 Détection d'erreurs

C'est la première phase dans chaque schéma de tolérance aux pannes, dans laquelle on reconnaît qu'un événement inattendu s'est produit. Les techniques de détection de pannes sont généralement classifiées en deux catégories : en ligne et autonome (offline). La détection offline est souvent réalisée à l'aide de programmes de diagnostic qui s'exécutent quand le système est inactif. La détection en ligne vise l'identification de pannes en temps réel et est effectuée simultanément avec l'activité du système.

II.3.2 Détention de la panne

Cette phase établit des limites des effets de la panne sur une zone particulière afin d'empêcher la contamination des autres régions. En cas de détection d'intrusion, par exemple, l'isolation des composants compromis minimise le risque d'attaque des composants encore fonctionnels.

II.3.3 Recouvrement d'erreur

C'est la phase dans laquelle on effectue des opérations d'élimination des effets de pannes. Les deux techniques les plus utilisées sont "masquage de panne" qui utilise l'information redondante correcte pour éliminer l'impact de l'information erronée, et "répétition" qui effectue, après la détection d'une panne, un nouvel essai pour exécuter une partie du programme, dans l'espoir que la panne soit transitoire.

II.3.4 Traitement de pannes

Dans cette phase, la réparation du composant en panne isolé est effectuée.

La procédure de réparation dépend du type de la panne. Les pannes permanentes exigent une substitution du composant avec un autre composant fonctionnel. Le système doit contenir un ensemble d'éléments redondants (ou en état standby) qui servent à remplacer les nœuds en panne.

II.4 Exemple de tolérance aux pannes dans un réseau de capteurs

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 II.2 [15]. On considère un réseau de capteurs pour la reconnaissance de personnes déployées dans une société pour identifier ses employés. Six personnes nommées A, B, C, D, E et F travaillent dans cette société. Le système de reconnaissance utilise deux types différents de capteurs : 1) capteur de taille (grandeur) ; 2) capteur pour la reconnaissance de voix 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. 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.

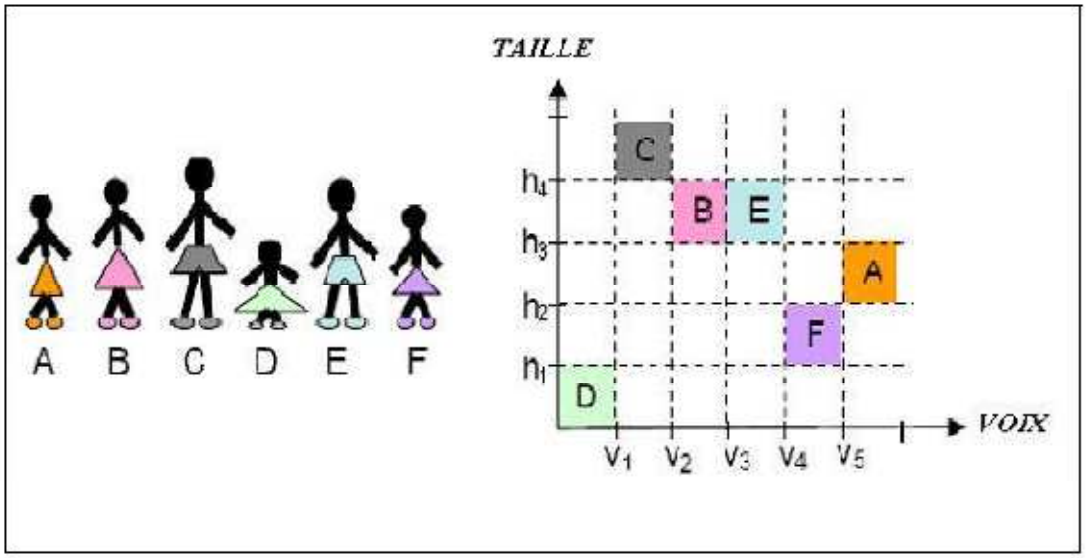

Figure II-2 : Exemple d'un réseau de capteurs multimodal.

Selon notre 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.

Il est nécessaire de faire une bonne modélisation d'un système complexe dans un réseau de capteurs permettant d'offrir une tolérance aux pannes hétérogène qui assure la fiabilité du système après la défaillance d'un nombre précis de capteurs d'une modalité donnée.

II.5 Classification des protocoles de tolérance aux pannes

Les protocoles tolérants aux pannes peuvent être vus de plusieurs angles différents. De ce fait, un ensemble de critères est défini pour les classifier. Nous citons, entre autre, deux principales catégories ; à savoir les classifications temporelles et architecturales.

II.5.1 Classification temporelle

Dans la classification temporelle, nous divisons l'ensemble des algorithmes en deux catégories, et cela selon la phase de traitement. Si le traitement est effectué avant la panne ; on parle donc d'algorithmes préventifs. Sinon, les algorithmes sont dits curatifs.

– algorithme préventif : implémente des techniques tolérantes aux pannes qui tentent de retarder ou éviter tout type d'erreur afin de garder le réseau fonctionnel le plus longtemps possible. La conservation d'énergie à titre d'exemple, permet de consommer moins d'énergie et évite donc une extinction prématurée de la batterie ce qui augmente la durée de vie des nœuds ;

– algorithme curatif : utilise une approche optimiste, où le mécanisme de tolérance aux pannes implémenté n'est exécuté qu'après la détection de pannes. Pour cela, plusieurs algorithmes de recouvrement après pannes sont proposés dans la littérature, par exemple : le recouvrement du chemin de routage, l'élection d'un nouvel agrégateur, etc.

II.5.2 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 :

II.5.2.1 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 ;

II.5.2.2 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 multipath : 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 chemin principal et choisissant 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 est implémentée au niveau de la couche MAC. Elle permet d'effectuer 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.

II.5.2.3 Gestion des données

Les protocoles classés dans cette catégorie offrent une meilleure gestion de données et de leur traitement. Deux principales sous-catégories sont déterminées :

– **Agrégation**: considérée comme approche préventive, l'opération d'agrégation effectue un traitement supplémentaire sur les données brutes captées depuis l'environnement. Un nœud agrégateur combine les données provenant de plusieurs nœuds en une information significative. Ce qui réduit considérablement la quantité de données transmises en consommant moins d'énergie pour leur dissémination. Ceci permet donc d'augmenter la durée de vie du réseau.

– Clustering : une des importantes approches pour traiter la structure d'un réseau de capteurs est le clustering. Il permet la formation d'un backbone virtuel qui améliore l'utilisation des ressources rares telles que la bande passante et l'énergie. Par ailleurs, le clustering aide à réaliser du multiplexage entre différents clusters. En outre, il améliore les performances des algorithmes de routage. Plusieurs protocoles utilisent cette approche préventive et parfois elle est considérée comme une approche curative.

II.6 La transmission de l'information dans un réseau de capteurs

Grâce à l'évolution de la micro-électronique et de l'informatique, les capteurs s'adaptent à tous les types d'applications tels que la supervision de sites énergétiques, la santé publique et la surveillance d'environnement. Ainsi, les réseaux de capteurs sans fil et mobiles deviennent de plus en plus répandus depuis une dizaine d'années. Face aux attentes industrielles et de services publics, les réseaux de capteurs doivent s'orienter vers une nouvelle approche incluant un routage prenant en compte la mobilité et assurant une connectivité plus efficace et plus fiable.

Le routage a été étudié au début dans les réseaux ad-hoc et standardisé par l'IETF (The Internet Engineering Task Force) depuis une dizaine d'années. En fonction du type de routage (proactifs ou réactifs), les routages AODV (Ad hoc On-Demand Distance Vector Routing), DSR (Dynamic Source Routing Protocol for Mobile Ad-Hoc Networks) et OLSR (Optimized Link State Routing Protocol) sont largement utilisés dans l'industrie et étudiés dans de nombreux projets de recherche. Cependant, ces protocoles de routage montrent leurs limites lorsque l'on désire les utiliser dans le contexte des réseaux de capteurs. Cela vient des fortes contraintes d'énergie, de débit et de portée inhérentes aux réseaux de capteurs.

Concrètement, un protocole de routage consiste à maintenir la connectivité entre deux entités (mobiles ou fixes) du réseau, un capteur et son nœud agrégateur par exemple, de façon à permettre la transmission d'informations. Autrement dit, il est nécessaire de maintenir automatiquement et dynamiquement un routage prenant en compte l'environnement et l'état du réseau. Dans les réseaux de capteurs, en plus de la puissance d'émission reçue à considérer comme un critère de choix, d'autres critères doivent également être considérés et intégrés dans le routage comme la consommation d'énergie ou encore la géo-localisation. Tout ceci conduit à résoudre un problème de routage dynamique et multicritères constituant un vrai défi.

De plus, les réseaux de capteurs sont sujets aux pannes (pannes de batterie ou physique, la perte de messages, perte de couverture suite aux conditions atmosphériques, à la mobilité et des données erronées), il est donc primordial de rendre le routage tolérant aux pannes. Pour cela, des approches auto-configurables seront privilégiées, notamment pour l'acheminement des messages en cas de rupture de liens entre des capteurs Des travaux existants ont déjà étudié les propriétés de tolérance aux pannes des protocoles de routage existants dans le contexte des réseaux sans fil en particulier les réseaux de capteurs. Ainsi, l'objectif sera d'identifier les propriétés nécessaires afin d'assurer un routage dynamique, robuste et tolérant aux pannes, et de proposer ensuite de nouveaux protocoles de routage adaptés au contexte des réseaux de capteurs.

La transmission de l'information dans un réseau peut se faire de deux façons :

II.6.1- L'envoi direct

Chaque nœud est en lien étroit avec l'unité de collecte, et aucun intermédiaire ne peut s'interposer dans cette relation directe privilégiée comme le montre la figure II-3.

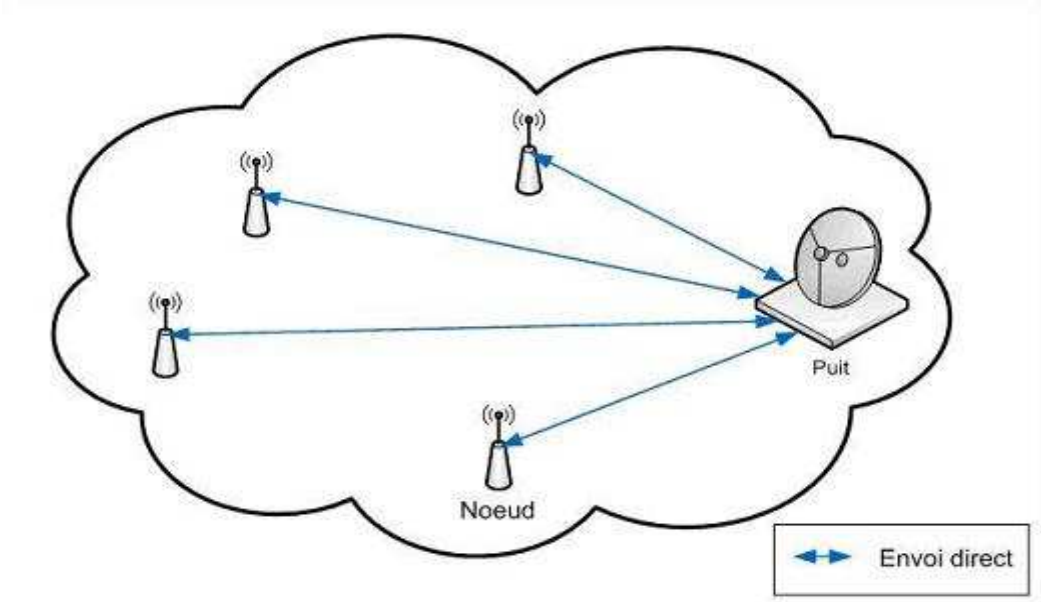

Figure II-3 : Envoi direct.

II.6.2 L'envoi par routage multi-sauts

Lorsque les nœuds sont distants de la station de base, l'envoi direct n'est pas possible car la portée des capteurs est limitée et toute communication distante directe peut épuiser rapidement la batterie d'un capteur. Pour se remédier à ce problème, il faut passer par des nœuds relais pour acheminer l'information à la station de base comme le montre la figure II-4 ci-dessous.

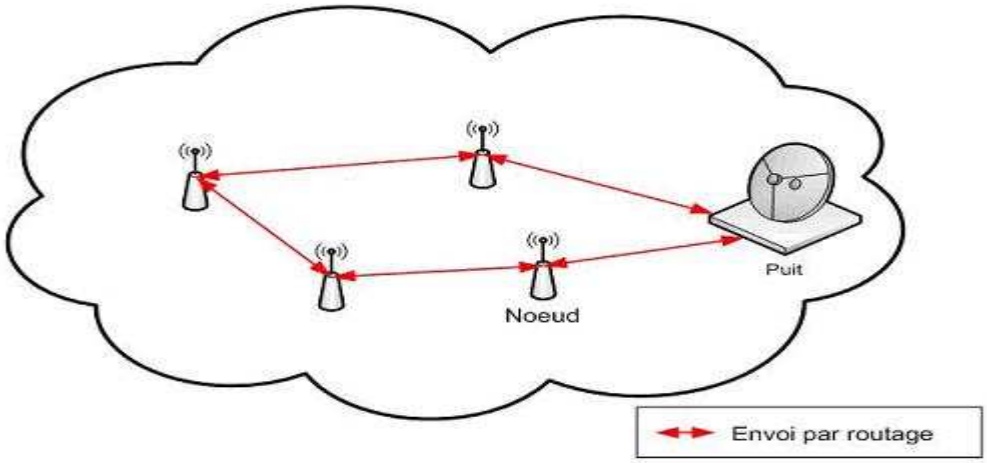

Figure II-4 : Envoie par routage.

II.7 La couverture de zone dans RCSF

Les capteurs fonctionnent avec un modèle à seuil, c'est à dire qu'un capteur possède deux zones: une zone de perception (SR) et une zone de communication (CR) comme montre la figure II-5.

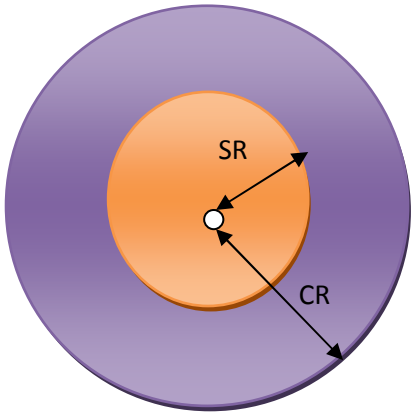

Figure II-5 : la couverture dans une zone.

En influant sur le rapport entre le rayon du SR et le rayon du CR, on va modifier les contraintes. Ainsi, on va pouvoir minimiser le nombre de nœuds actifs et maximiser la durée de vie du réseau.

Les zones CR et SR représentent la zone de couverture d'un capteur. Pour qu'une zone soit complètement couverte, il faut que la densité de capteurs soit suffisante. Comme les capteurs sont généralement déployés aléatoirement sur une zone d'intérêt, il est nécessaire de disposer d'une densité importante de capteurs. Si la densité de capteurs est trop importante et que la zone que l'on veut surveiller est "trop" couverte, alors des capteurs vont fonctionner inutilement. De ce fait, il faut ordonnancer le mode d'activité des capteurs en passant quelques capteurs en mode veille tout en assurant la couverture totale de la zone.

II.7.1 Solution pour économiser de l'énergie

Afin de ne pas gaspiller d'énergie, les capteurs qui fonctionnent inutilement vont se mettre en veille. Ce mécanisme va devenir une stratégie à part entière pour augmenter la durée de vie du réseau. En effet, en choisissant une densité volontairement élevée de capteurs, on va multiplier le nombre de capteurs redondants. Ainsi de nombreux capteurs seront en mode "veille" et pourront se substituer aux capteurs défaillants si nécessaire. C'est ce que montre la figure II.6 ci-dessous :

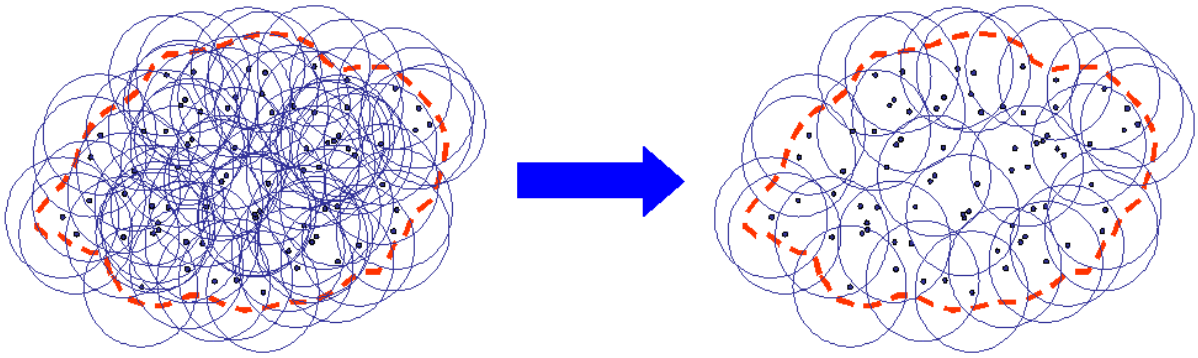

Figure II-6 : l'économisassions des l'énergie.

Au moment de la mise en place du réseau, tous les capteurs sont actifs. Ceux dont la zone est déjà couverte se mettent en veille. Puis, les capteurs en veille effectueront régulièrement des requêtes pour savoir s'ils ont besoin de s'activer.

II.7.2 La k-couverture de surface dans les réseaux de capteurs

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

II.8 Conclusion

Dans ce chapitre, nous avons présenté d'une manière générale le principe de la tolérance aux pannes dans les réseaux de capteurs. Puis, nous avons fait le point sur le concept de la tolérance aux pannes dans les réseaux de capteurs pour assurer la fiabilité de délivrance de paquets à la station de base et pour garantir une couverture totale de la zone d'intérêt.

Dans le chapitre suivant nous détaillons, le routage permettant la tolérance aux pannes dans les réseaux de capteurs.

Chapitre III :

Etat de l’art sur les protocoles de routage tolérants aux pannes

Chapitre III

Etat de l’art sur les protocoles de routage tolérants aux pannes

III.1 Introduction

La défaillance des nœuds dans un réseau de capteurs peut être engendrée par plusieurs causes, notamment l’épuisement d’énergie, l’endommagement physique, ou les interférences liées à l’environnement.

La propriété de tolérance aux pannes est définie par l’habilité du réseau à maintenir ses fonctionnalités sans interruptions provoquées par la panne des capteurs. Elle vise donc à minimiser l’influence de ces pannes sur la tâche globale du réseau [16].

L’approche la plus célèbre de tolérance aux pannes est le routage multi-chemins, où plusieurs chemins multiples entre les nœuds source et la station de base sont déterminés au détriment de la consommation d’énergie accrue et la génération du trafic.

Dans ce chapitre, nous tenons à présenter une synthèse sur quelques protocoles de routage tolérants aux pannes proposés dans la littérature.

III.2 Protocoles de routage dans les RCSF

Suivant la manière de création et de maintenance des routes, les protocoles de routage peuvent être répartis en trois catégories: proactifs, réactifs et hybrides [17].

III.2.1 Protocole proactif : Les protocoles de routage proactifs maintiennent à jour les tables de routage avant de commencer la transmission [18]. Chaque nœud maintient donc une table pour stocker les informations de routage. Du fait de l'aspect dynamique de la topologie des réseaux, la maintenance des tables de routage nécessite l'envoi périodique par chaque nœud un message de signalisation indiquant sa présence à tous ses voisins. L'idée majeure est de conserver dans chaque nœud des informations de routage vers tous les autres nœuds du réseau pour accélérer le routage des paquets par la suite [19]. Les changements topologiques du réseau sont gérés par la propagation à tous les voisins la mise à jour des routes afin que chacun d’eux puisse maintenir une vue globale sur réseau. Malheureusement ces protocoles atteignent rapidement leurs limites avec l'accroissement du nombre de nœuds et de leur mobilité. Or les changements topologiques sont fréquents et le réseau sera ainsi constamment inondé par les paquets

de contrôle qui réduisent considérablement la bande passante. Mais ils permettent, en cas d'envoi successif d'informations d'une même source vers une même destination, d'utiliser toujours la même route connue à l'avance.

III.2.2 Protocoles réactifs : Ces protocoles créent et maintiennent des routes selon les besoins. Lorsque le réseau a besoin d'une route, une procédure de découverte de route est lancée. Ce type de protocoles est pratique pour des applications temps réel où les capteurs doivent réagir immédiatement à des changements soudains des valeurs captées. En effet, un prélèvement périodique des données aurait été inadapté pour ce type de scénarios.

III.2.3 Protocole hybride : Ces protocoles combinent les deux idées citées dans les protocoles proactifs et réactifs. Ils utilisent un protocole proactif pour connaitre le proche voisin (par exemple le voisinage à deux ou à trois sauts), ainsi, ils disposent de routes immédiatement. Au-delà de la zone du voisinage, le protocole hybride fait appel à un protocole réactif pour chercher des routes.

Récemment, les protocoles de routage pour les RCSF ont été largement étudiés, et différentes études ont été publiées. Les méthodes employées peuvent être classifiées suivant plusieurs critères comme illustré sur la figure III-1.

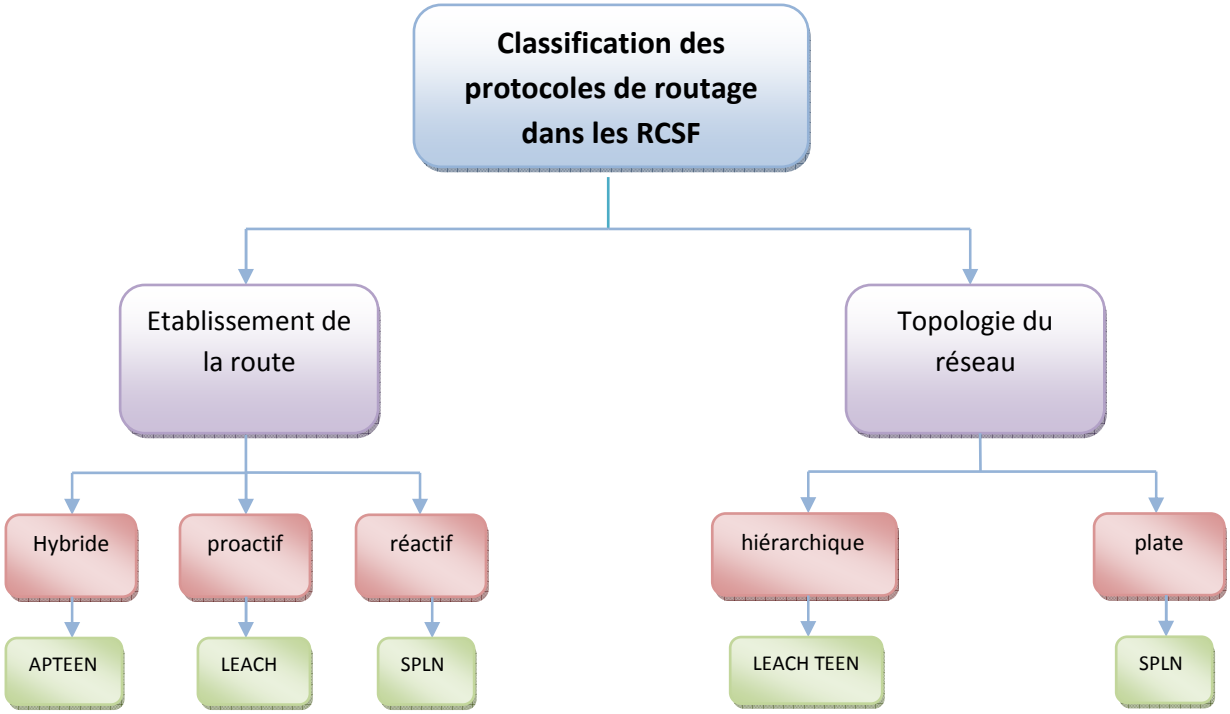

Figure III-1: classification des protocoles de routage [20].

III.3 Les protocoles de routage tolérants aux pannes dans les RCSF

La propagation et la délivrance des données dans un RCSF représentent la fonctionnalité la plus importante du réseau. Elle doit prendre en considération toutes les caractéristiques des capteurs afin d'assurer les meilleures performances du système : durée de vie, fiabilité, temps de réponse, ... etc.

Les protocoles de routage proposés pour les réseaux de capteurs [21,22] peuvent être classés en trois groupes en fonction des méthodes utilisées pour trouver le chemin: routage proactif dans lequel tous les chemins sont calculés et maintenu à l'avance est stocké dans une table de routage, routage réactif où tous les chemins sont créées à la demande, et le routage hybride qui est qui combinent les deux types de routage précédents.

La tolérance aux pannes pour assurer une fiabilité de délivrance de paquets à la station de base est traitée au niveau de la couche réseau. Dans ce qui suit, nous présentons les fonctionnalités de certains protocoles de routage tolérants aux pannes et nous discutons leurs limites:

III.2.1 Protocole de routage dynamique tolérant aux pannes pour prolonger la durée de vie dans RCSF

L'objectif de ce protocole est de maintenir la connectivité du réseau, même si un nœud est sur le point d'épuiser son énergie pour assurer la livraison de données à la station de base tout en prolongeant la durée de vie du réseau [23].

Dans ce protocole, quand un nœud capteur est sur le point d'épuiser son énergie, il essaie de trouver un chemin alternatif pour établir une nouvelle connexion avec ses nœuds voisins. Ce chemin alternatif augmente la fiabilité de transmission de données entre les nœuds source et leurs voisins dans la direction de la station de base qui relaient les paquets envoyés par ces derniers. En outre, les applications de type Event-Driven exigent que les informations recueillies par les capteurs doivent être transmises immédiatement à la station de base. Ce protocole s'exécute en trois phases:

- Mise en ouvre et établissement de chemin:

Chaque nœud est caractérisé par son identifiant nœud (N_j), le niveau (HC_j), nœud parent (P_j), un tableau (A_j) pour stocker les paquets de données jusqu'à ce qu'un accusé

de réception soit reçu. La station de base est initialisé avec HC = 0, P = BS, tandis que les nœuds ordinaires avec d'autres $HC_j = \infty$, $P_j = 1$.

Une fois les nœuds sont déployés, la station de base diffuse un message d'avertissement ADVT (N_j, HC_j) pour découvrir les nœuds qui sont voisins à la station de base. Ces nœuds sont considérés comme des nœuds de niveau 1 puisqu'ils se trouvent à un saut de la station de base qui est considérée comme un nœud parent pour ces nœuds de niveau 1. Lorsqu'un nœud reçoit un message ADVT, son HC sera augmenté de un que de celui qui lui a envoyé le message ADVT et il est considéré comme un nœud de niveau N +1 si le nombre de sauts reçu est N. Ainsi, le message ADVT est utilisé pour hiérarchiser le réseau en des niveaux relativement à la station de base.

- Transmission de données:

Une fois que la hiérarchisation de niveaux est établie, la phase de transmission de données commence. De ce fait, lorsqu'un événement survient au niveau du nœud source. Ce dernier transmet le paquet de données relatif à l'événement au nœud parent et stocke une copie de ce paquet de données. Quand un parent reçoit le paquet de données émis, il envoie un accusé de réception (ACK) au nœud qui a transmis le paquet. De son côté le nœud source, une fois qu'il reçoit le paquet ACK, il supprime la copie du paquet de données correspondant. Cela continue jusqu'à ce que la station de base reçoive le paquet de données. Un numéro de séquence est attribué à chaque paquet de données transmis pour assurer la fiabilité et garantir sa livraison à la station de base. Si un paquet de données est perdu, il pourra être récupéré à partir de dernier expéditeur.

- Rétablissement de chemin

Si un nœud est sur le point d'épuiser son énergie, il envoie un message de notification à ses voisins fils en leur demandant de changer leurs nœuds parents pour maintenir la connectivité. Les nœuds fils qui reçoivent ce message, utilisent des paquets « Hello » pour découvrir les nouveaux parents ou leurs voisins. Les nœuds fils modifient leurs paramètres N_j, HC_j en fonction de la réponse au message Hello. Si la réponse provient d'un nœud de niveau inférieur, les nœuds fils gardent leur N_j sinon c'est-à-dire le message provient d'un nœud voisin, les nœuds fils doivent incrémenter leurs niveaux de 1.

La limitation de ce protocole est que le temps pris pour trouver un nouveau nœud voisin affecte la durée de livraison de données. En outre, dans ce protocole, les auteurs

ont supposé que l'environnement est idéal car ils n'ont pas pris en compte la présence d'interférences et de bruit qui se trouvent dans le monde réel.

III.2.2 Protocole de routage tolérant aux pannes multi-niveaux avec ordonnancement d'activité de capteurs (FMS).

Le protocole FMS [24] permet est de maintenir la connectivité du réseau, même si un nœud est sur le point d'épuiser son énergie. Il permet aussi d'assurer la fiabilité et la rapidité de livraison des données à la station de base car il est conçu pour les applications orientées événement. Généralement, les capteurs sont déployés aléatoirement et en grand nombre. De ce fait, il y aura une redondance dans la livraison de données ce qui a une conséquence sur la durée de vie du réseau de capteurs. Pour se remédier à ce handicap, FMS permet un ordonnancement d'activité des capteurs en passant un certain nombre de capteurs en mode « veille » sans affecter la fiabilité de livraison de données. Ceci est dans le but d'économiser l'énergie. Dans FMS, on suppose que chaque nœud possède un identifiant unique, dénoté (Nr), et la communication entre les nœuds voisins est bidirectionnelle. En outre, on suppose que les nœuds sont contraints en termes de puissance de traitement, de stockage et de l'énergie, tandis que les nœuds la station de base est considérée comme un nœud qui a plus de ressources pour effectuer des tâches ou de communiquer avec les autres nœuds.

Le protocole FMS effectue deux opérations de base:

- Mise en œuvre de niveau et établissement de chemin: cette phase est analogue à celle du protocole cité ci-dessus.
- Ordonnancement d'activité et transmission de données: l'ordonnancement d'activité des capteurs consiste à faire passer un certain nombre de nœuds périodiquement en mode veille. Au cours de cette période, les nœuds actifs transmissent les paquets de données. Avant qu'un nœud passe en mode veille, il devra informer ses nœuds fils afin qu'ils choisissent un autre nœud parent pour relayer les données. En outre, quand un nœud est en mode veille, il passera en mode actif que si son énergie est supérieure à une certaine valeur seuil. Le choix des nœuds actifs se fait aléatoirement et d'une manière périodique pour que le nœud n'épuise pas son énergie rapidement. Quand un nœud est en mode actif, il est sensé de participer à l'opération de transmission de données à la station de base. De ce fait, la connectivité est toujours maintenue même si un nœud est mis en sommeil ou il est sur le point de perde son énergie. Ainsi, FMS est considéré comme un protocole fiable et tolérant aux pannes.

La hiérarchisation en niveaux dans ce protocole est analogue que celle du protocole cité précédemment. Seulement dans FMS, les auteurs ont introduit la procédure d'ordonnancement d'activité des capteurs pour économiser l'énergie et par suite assurer une longue longévité aux RCSF. En outre, le passage du mode veille au mode actif est contraint en terme d'énergie.

FMS présente les mêmes limitations que le protocole cité précédemment. Il est performant dans un environnement optimal mais ses performances se dégradent dans un environnement réel.

III.2.3 RERP est un protocole de routage adaptatif tolérant aux pannes pour les RCSF

Un RCSF est formé d'un grand nombre de capteurs qui sont déployés aléatoirement dans une zone d'intérêt. Dans RERP [25], il est supposé que chaque nœud a au moins deux voisins dans la direction vers la station de base. De ce fait, il aura au moins deux chemins alternatifs pour acheminer les données vers la station de base. Les nœuds qui sont loin de la station de base n'empêchent pas la formation de la boucle. La capacité d'un nœud tolérant aux pannes dépend du nombre des nœuds voisins actifs, c'est-à-dire si un nœud a N voisin, il peut tolérer N-1 nœuds en panne.

Le protocole RERP est protocole de routage proactif et sa conception comporte deux tâches:

- Mise en place de RERP : cette tâche s'exécute en cinq phases:
  - ✓ phase de publicité:

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

  - ✓ phase d'initialisation:

    Dans cette phase, les nœuds qui n'ont pas de chemin direct vers la station de base diffusent une requête de découverte de routes (RREQ) vers la station de base. Quand un concentrateur reçoit le paquet (RREQ), il diffuse une réponse (RREP). De même si c'est ce nœud a déjà reçu la requête (RREQ) il diffuse un paquet (RREP) s'il existe un chemin entre lui et le concentrateur, sinon le paquet (RREQ) sera ignoré.

✓ Route de sélection:

Une table de routage est utilisée pour construire et entretenir les routes. Le choix de l'itinéraire des nœuds relais est basé sur l'énergie restante des nœuds.

✓ Phase de transfert de données :

Les nœuds capteurs génèrent des paquets de données à chaque fois qu'ils détectent toute nouvelle information. Cette information est transmise à la station de base dans un mode multi-sauts.

✓ Table de sauvegarde:

En plus du chemin principal, un chemin alternatif est prévu pour tous les nœuds du réseau. Chaque fois qu'un nœud reçoit un paquet RREP, si il ne dispose pas de chemin direct vers la station de base, il stocke le chemin dans la table de routage, et il stocke les paquets (RREP) dans une table de sauvegarde. La table de routage de secours dispose de deux champs, l'identifiant du nœud ID et son énergie.

- Rapport d'erreurs: Les fonctions pour reporter les erreurs sont incorporées dans ce protocole de routage dont lesquelles figurent les types de messages suivants:

✓ Echec des liens: le message d'échec de liens est généré dans les deux cas. Le premier se produit quand un RTS est envoyé mais aucune CTS correspondant n'est reçu et le nombre maximal de tentatives est dépassé. Le second se passe quand un paquet de données a été transmis, mais il n'a jamais reçu un ACK de réception et le nombre maximal de tentatives est dépassé.

✓ Message de batterie critique: Ce message est généré lorsque le niveau de la batterie d'un nœud est inférieur à une valeur seuil dite critique. Ce message est envoyé au nœud source qui a envoyé les données et également aux voisins de ce nœud. Quand les autres nœuds reçoivent ce message ils suppriment l'identifiant du nœud défaillant de leurs tables de routage ou de leur table de voisins.

✓ Message de destination inaccessible: Ce message est généré lorsque le paquet de données est mis au rebut sans être transmis au nœud de destination en raison de l'indisponibilité du chemin vers la station de base.

✓ Sélection du chemin de secours: Chaque nœud possède une table de routage de secours dans laquelle il stocke un chemin de secours vers la destination. Quand un nœud échoue dans la transmission de paquet de donnée, alors son voisin consulte la

table de secours pour trouver le chemin alternatif afin qu'il puisse transmettre le paquet de données.

Dans RERP la communication entre les nœuds est réalisée par des messages Requête/Réponse où le nœud expéditeur envoie une requête « Hello » au nœud de destination. Ce type de messages est utilisé pour vérifier si le voisin est accessible et pour calculer le temps de parcours.

RERP présente certaines limitations telles que la consommation d'énergie qui est assez grande lors de la diffusion des rapports d'erreurs.et aussi le paquet de données sera perdu lorsque le chemin vers la station de base n'est pas disponible à partir d'un nœud.

III.2.4 Protocole de routage temps réel tolérant aux pannes (DMRF)

DMRF [26] fonctionne en deux modes de transmission de données: saut-à -saut et le mode de transmission «Jumping». Chaque nœud utilise le temps restant pour transmettre un paquet à la station de base et l'ensemble des nœuds de transfert FCS (Set candidat Forwarding) pour choisir dynamiquement le prochain saut. Quand un nœud présente une défaillance, alors la congestion du réseau ou d'une région vide se produit. Le mode de transmission sera passé en mode « Jumping », ce qui peut réduire le délai de transmission, et assure la fiabilité de la livraison des paquets de données envoyés à la station de base dans un délai spécifié. Il est théoriquement prouvé que DMRF peut répondre en temps réel aux exigences de tolérance aux pannes.

Dans DMPF, le processus de transmission est divisé en cinq étapes:

- Phase d'initialisation: dans cette phase, DMRF initialise la liste de voisinage des nœuds, la liste de l'état du réseau (information sur la congestion d'un nœud, les zones vides, …), liste des candidats FCS, table des probabilités de transition, et la voie de transmission initiale.
- Phase de transmission des données: Dans cette phase, DMRF détecte la défaillance d'un nœud, la congestion du réseau et de les régions vides. Le temps restant pour acheminer un paquet de données jusqu'à la station de base sera contrôlé. A partir de ce temps, le paquet sera transmis en mode « Jumping » ou non. Si aucune des conditions ci-dessus ne se produit, DMRF sélectionne dynamiquement un membre du FCS comme nœud relais en se basant sur le taux de transmission de données

et des informations locales. Une fois les nœuds défaillants sont détectés, ou le temps restant est inférieur à un certain seuil, le mode de transmission « Jumping » sera utilisé.

- Phase de transmission « Jumping »: au cours de cette phase, chaque nœud ajuste dynamiquement le contenu de FCS et calcule la probabilité pour transiter à chacun de ces nœuds. Dans ce mode, le paquet de données peut un saut d'une grande portée pour éviter les nœuds défaillants. Cependant, il ne peut pas garantir le succès de la transmission. Donc la phase d'ajustement de probabilités de transitions est effectué après chaque transmission « Jumping ».
- La phase de probabilité d'ajustement : Dans cette phase, DMRF ajuste la probabilité de saut en fonction du résultat de la transmission « Jumping » (succès ou l'échec) et renvoie l'information à son nœud en amont. Lorsque le paquet de données arrive au nœud récepteur, on considère que la transmission est terminée. Dans DMRF, le nœud peut transmettre directement des données à la station de base ou en passant par des nœuds relais en fonction de la probabilité de transition vers ces nœuds. Si la transmission de données échoue, la probabilité de transition sera mise à jour via un mécanisme de rétroaction, qui peut non seulement éviter l'effet causé par la défaillance des nœuds, mais aussi d'améliorer le taux de transmission. Ceci permet de réduire la consommation d'énergie.

DMRF présente certaines limitations en particulier dans le mode « Jumping » qui ne garantit la fiabilité de livraison de données et qui consomme plus d'énergie quand il utilise une grande portée.

III.2.5 Amélioration protocole de routage tolérant aux pannes AODV (ENFAT-AODV).

AODV (ENFAT-AODV) [27] est un protocole de routage permet la tolérance de pannes, l'auto-démarrage de le routage multi-sauts entre les nœuds participants qui souhaitent créer et maintenir un réseau de capteurs sans fil. ENFAT-AODV offre une mise en place d'itinéraire rapide et efficace entre les nœuds de communication désirant.

En outre, ENFAT-AODV permet aux nœuds sur une voie principale de transmettre des données pour obtenir un chemin de secours, qui est utilisé lorsque leur chemins principal seront perdu, pour répondre à établir un lien entre les ruptures des nœuds dans les meilleurs délais. Le nombre de sauts long d'un trajet est utilisé en tant que métrique d'une sélection de chemin. Si des multiples RREP avec le numéro de destination on la même séquence sont reçus par la source, le chemin le plus court avec le comptage de

houblon est choisi. Ce qui fait que les demandes RREQs et les réponses RREP du même type sont acheminées. Tels qu'elles sont définis par AODV.

Toutefois, pour ENFAT-AODV, certains champs sont ajoutés dans les paquets de contrôle tels que "BACKUP" drapeau (en RREQ et RREP), "UPDATE" drapeau (en RREQ) et "Distance pour Dest" de terrain (dans RREQ). En outre, ENFAT-AODV nécessite que certains messages (par exemple, RREQ) doivent être largement diffusées, peut-être à travers le réseau. La zone de diffusion de ces RREQs est indiquée par le TTL dans l'en-tête IP.

En outre, ENFAT-AODV réduit également une certaine complexité de mise en œuvre par l'élimination d'un ensemble d'éléments du cahier des charges, les messages sont supprimés afin de réduire les paquets de contrôle inutiles dans le réseau. Ensuite, le fonctionnement local de réparation n'est pas inclus dans ENFAT-AODV.

✓ Découverte de la route principale:

Quand un chemin principal de livraison des données vers la station de base est nécessaire, le nœud source coulera un message (découverte d'une route principal), un processus diffuse un paquet de demande principale (principale RREQ) a la station de base. À chaque nœud intermédiaire, lorsqu'un RREQ principale est reçu, un chemin inverse à la source est créé. Si le nœud de réception n'a jamais reçu ce RREQ principal avant, et si le nœud ne connaît pas la route principale menant à la destination, il transmet le RREQ principale à ses voisins. Si le nœud de réception est la destination ou bien s'il connaît la route principale menant à la destination, il va générer un itinéraire principal Réponse (principale RREP). Ensuite, le RREP principal est uni coulé par saut à la source. Comme le RREP principale est renvoyé à la source, chaque nœud intermédiaire qui traite le RREP principale crée un chemin principal vers la station de base. Lorsque la source reçoit le RREP principale, elle enregistre la route principale menant à la destination dans sa table de routage principale.

✓ La construction la route de sauvegarde:

Au cours de la phase de croisière réponse principale, les nœuds d'un chemin principal qui reçoivent un RREP principale créent un chemin de sauvegarde vers la station de base en diffusant un paquet de sauvegarde RREQ. Après la diffusion du RREQ sauvegarde, le nœud attend un paquet RREP sauvegarde de la destination elle-même ou un nœud intermédiaire qui peut satisfaire les conditions spécifiées comme suit:

- Il dispose d'une entrée de sauvegarde active du chemin principal vers la station de base,
- Il n'est pas un nœud sur le chemin principal,
- Et le nombre de sauts du chemin de sauvegarde active à partir du nœud intermédiaire à la destination est inférieur au domaine de la sauvegarde RREQ, pour garantir qu'il fournira un chemin de sauvegarde court.

✓ Entretien de la route:

Pendant la période de livraison de paquets de données, lorsque le chemin principal n'est pas valide ou reçoit un paquet de données destiné au nœud de destination pour laquelle il ne dispose pas d'un chemin actif principal, le nœud utilise immédiatement sa route de sauvegarde pour livrer les prochains paquets de donnée qui viennent, sans interruption. Par la suite, le nœud sur le nouveau chemin principal, qui utilise une route de secours, dirige un processus "Découverte de route de secours" visant à trouver une voie nouvelle. Par conséquence, il augmente de plus la fiabilité et la disponibilité par rapport à l'original du protocole AODV routage.

La limitation de ce protocole repose sur la consommation de l'énergie à cause de l'inondation des messages de contrôle.

III.2.6 FaT2D: Diffusion par tolérance aux pannes Réalisé pour les RCSF

FaT2D [28] est un protocole de tolérance aux pannes basée sur la diffusion. Ce dernier a l'avantage de fournir une forte tolérance contre les défaillances de nœuds grâce à sa construction des trajets multiples et l'exploration périodique des routes. FaT2D définit une nouvelle technique qui permet de détecter rapidement une panne et la recouvrir quand il y a une collision entre les nœuds et des changements de topologie.

Il s'exécute selon la démarche qui suit:

- **Détection de panne**: Un nouveau délai d'attente de détection de pannes, noté TFD, est défini afin de réduire le temps de recouvrement de la panne et par conséquent la réparation des nœuds et le chemin local se font rapidement, tout en tolérant les pannes intermittentes dues aux pertes de paquets.

Si TFD s'épuise, FaT2D transmet immédiatement un nouveau message appelé Explore-Request pour notifier l'événement de détection de la panne et demande une nouvelle

exploration pour trouver un autre chemin fiable qui remplace le chemin défaillant. Par conséquent, tout nœud appartenant au chemin défaillant supprime le gradient correspondant pour éliminer la panne.

- **Recouvrement du chemin**: Quand Tfd s'épuise, il déclare la défaillance d'un nœud. De ce fait, FaT2D lance un processus pour réparer le chemin défectueux en envoyant un message de demande d'exploration appelé « ExploreRequest ». Ce message contient les informations sur la route défectueuse et il est acheminé pour atteindre le nœud cible sans utiliser les transmissions bouclées ou rechercher les nœuds non adéquats.

Quand le nœud cible reçoit le message « ExploreRequest », il arrête de le transmettre, puis il lance une exploration par inondation comme dans « Direct Diffusion ». Cela génère une phase d'exploration afin de trouver un nouveau chemin fiable. L'élection de ce chemin se fait selon les règles utilisées dans « Direct Diffusion ».

- **Elimination des pannes:** Pour chaque nœud intermédiaire recevant le message ExploreRequest, FaT2D vérifie si ce nœud appartient au chemin défectueux. Si c'est le cas, il aura un effet négatif pour renforcer son gradient. Ce dernier sera réélu par une 'exploration envoyé par le nœud source du chemin correspondant. Ainsi, chaque nœud exécute un renfort local négatif à son voisin en amont, afin de supprimer le chemin brisé et arrêter l'envoi de données perdues.

Un des principaux défis dans la conception des protocoles de routage pour les réseaux de capteurs est l'efficacité énergétique en raison des ressources limitées que présentent les capteurs.

Dans les RCSF, l'objectif de la conception de protocole de routage est d'assurer la fiabilité de livraison de données à la station de base tout en prolongeant la durée de vie du réseau. Or, la consommation d'énergie dans les RCSF est dominée par la transmission de données et la réception. Par conséquent, les protocoles de routage dans RCSF doivent minimiser au moins les messages qui sont retransmis lors de l'occurrence d'une panne.

III.4 Conclusion

Le routage dans les réseaux de capteurs est un problème complexe car nous devons assurer la fiabilité de livraison de données tout en consommant moins d'énergie. En outre, les protocoles de routage conçus pour les réseaux ad-hoc ne sont pas recommandés pour les RCSF car ces derniers sont composés de nœuds à ressources limitées.

Dans ce chapitre, nous avons présenté les protocoles de routage conçus aux RCSF et nous avons fait une étude sur les protocoles de routage tolérants aux pannes. Notre constat nous a permis d'illustrer les limites de ces protocoles. D'où, nous avons pensé à améliorer LEACH qui est un protocole bien réputé mais qui n'est pas tolérant aux pannes.

Chapitre IV :

Evaluation de LEACH et proposition de T-LEACH

CHAPITRE IV

Evaluation de LEACH et proposition de T-LEACH

IV.1 Introduction

L'objectif principal de notre travail est la mise en œuvre d'une solution qui se charge d'améliorer le protocole de routage LEACH de telle sorte qu'il soit tolérant aux pannes. Notre premier but est d'atteindre un niveau de tolérance aux pannes acceptable sans dégrader les performances du réseau. De ce fait, nous avons établi un nouveau protocole appelé T-LEACH qui est en mesure de pallier les limites de LEACH dans un environnement non idéal.

Dans ce chapitre, nous montrons l'apport du protocole T-LEACH par rapport au protocole LEACH en termes de tolérance aux pannes en comparant des métriques de performances via l'implémentation et l'évaluation des deux protocoles. Pour cela, nous commencerons par définir les outils nécessaires pour l'implémentation et la simulation des deux protocoles. Ensuite, nous décrirons la mise en œuvre de toutes les structures de données et processus décrits lors de la conception. Nous terminerons ce chapitre par une présentation des résultats relevés lors des tests de performances des deux protocoles LEACH et T-LEACH.

IV.2 Environnement de simulation

Dans cette section, nous présentons les outils utilisés pour la mise en œuvre des protocoles LEACH et T-LEACH. Nous commençons tout d'abord par TinyOS, le système d'exploitation conçu pour les dispositifs à ressources limitées en particulier les RCSF. Nous décrivons ensuite le langage de programmation NesC avec lequel nous avons programmé les codes des deux protocoles. Nous terminons cette partie par la présentation d'un simulateur des RCSF TOSSIM qui offre deux mécanismes permettant d'émuler le réseau : l'interface graphique TinyViz pour visualiser le déroulement de la simulation, et, le simulateur PowerTOSSIM pour simuler et évaluer la consommation d'énergie. Nous fournissons des informations plus détaillées dans l'annexe.

IV.2.1 TinyOS

Suite aux différents défis des RCSF qu'on a vus dans les chapitres précédents, l'université de Berkeley, en plus de nombreux contributeurs ont développé un système d'exploitation destiné aux RCSF afin de faciliter l'implémentation et l'exécution de protocole dédiés à ce type de réseaux.

L'objectif consiste à minimiser la taille du code afin de respecter les contraintes de ressources énergétiques et physiques des nœuds capteurs. Ce système est intitulé TinyOS [29]**.** Il a l'avantage de permettre une programmation simple et puissante tout en gardant la portabilité du code pour les nombreuses plateformes supportées. Il est utilisé par plus de 500 universités et centres de recherche dans le monde vu la caractéristique open-source qu'il détient [30]. Il respecte une architecture basée sur une association de composants et utilise une programmation entièrement réalisée en langage NesC.

Figure IV-1 : Sigle de TinyOS.

IV.2.1.1 Pourquoi TinyOS ?

Les systèmes d'exploitation pour les nœuds capteurs sont généralement moins complexes que les autres systèmes. Plusieurs systèmes d'exploitation ont été proposés pour les RCSF parmi lesquels on trouve SOS [31], Contiki [32], MANTIS [33].

TinyOS reste néanmoins le plus répandu pour les RCSFs car il répond aux exigences particulières des applications des RCSF. Il convient alors de mentionner les propriétés qui rendent TinyOS aussi populaire et réputé pour ce genre de réseaux [34]**.**

– Une taille de mémoire réduite.
– Une basse consommation d'énergie.
– Des opérations robustes.
– Applications orientées composants: TinyOS fournit une réserve de composants systèmes utilisables au besoin.

– Programmation orientée évènement : Généralement sur TinyOS, un programme s'exécute suivant le déclenchement des événements. Sinon, les capteurs restent en veille ce qui maximise la durée de vie du réseau.

IV.2.1.2 Notions principales

TinyOS est construit autour des différents concepts décrits ci-dessous.

- **Les composants :** constitués de :
  - **Frame :** est un espace mémoire de taille fixe permettant au composant de stocker les variables globales et les données qu'il utilise. Il n'en existe qu'un seul par composant.
  - **Tâches :** contiennent l'implémentation des fonctions. Elles sont décomposées en deux catégories : les commandes et les évènements.
- **Les interfaces :** représentent le descriptif des fonctions définies dans les tâches.

IV.2.2 NesC

NesC est un langage de programmation orienté composants syntaxiquement proche du langage C. Il est conçu pour la réalisation des systèmes embarqués distribués, en particulier, les RCSF.

Il existe trois types de fichiers sources des applications NesC: les fichiers interfaces et les fichiers configurations et modules qui constituent les composants [35].

- Une configuration définit les composants et/ou les interfaces utilisés par l'application déployée sur le capteur. Elle définit aussi la description des liaisons entre eux.
- Un module constitue la brique élémentaire du code et implémente une ou plusieurs interfaces.
- Une interface définit d'une manière abstraite les interactions entre deux composants. Elle définit un fichier décrivant les commandes et les évènements proposés par le composant qui les implémente. Une commande doit être implémentée par le fournisseur de l'interface et un évènement doit être implémenté par l'utilisateur de l'interface.

On distingue les modules et les configurations dans le but de permettre aux concepteurs d'un système de construire des applications rapidement et efficacement. Par exemple, un concepteur peut fournir uniquement une configuration qui relie un ensemble de modules qu'il ne développe pas lui même. De plus, un autre développeur

peut fournir une librairie de modules qui peuvent être utilisés dans la construction d'autres applications [36].

IV.2.3 TOSSIM

Avant sa mise en place, le déploiement d'un RCSF nécessite une phase de simulation afin de s'assurer du bon fonctionnement de tous les protocoles de communication qu'il utilise.

En effet, pour de grands réseaux, le nombre de capteurs peut atteindre plusieurs milliers et entraine donc un coût financier relativement important. Ainsi, il faut réduire au maximum les erreurs de la conception. Malgré cela, il reste des facteurs réels qui ne peuvent être pris en compte par la simulation, tels que les contraintes physiques (perturbations électromagnétiques, inondations, etc.) ou les aléas (détériorations dues à un animal, etc.). Pour arriver à simuler le comportement des capteurs au sein d'un RCSF, un outil très puissant a été développé et proposé pour TinyOS sous le nom de TOSSIM. Le principal but de TOSSIM est de créer une simulation très proche de ce qui se passe dans les RCSF dans le monde réel. Une économie d'effort et une préservation du matériel sont possibles grâce à cet outil**.**

Pour une compréhension moins complexe de l'activité du réseau, TOSSIM peut être utilisé avec une interface graphique TinyViz. Cette dernière est équipée par plusieurs API plugins qui permettent d'ajouter plusieurs fonctions à notre simulateur comme par exemple suivre la dépense d'énergie en utilisant un autre simulateur qui s'appelle PowerTOSSIM [37].

IV.2.3.1 TinyViz

TinyViz est une interface graphique Java. Elle permet de donner un aperçu des capteurs à tout instant ainsi que des divers messages qu'ils émettent. Elle détermine un délai entre chaque itération des capteurs afin de permettre une analyse pas à pas du bon déroulement des actions en activant différents modes comme Radio, CPU, etc.

Nous allons détailler un peu ce que fait chaque bouton présent dans l'interface :

- **ON/OFF** : il met en marche ou éteint un capteur.
- **Delay** : il permet de sélectionner la durée au bout de la quelle se déclenche le timer.
- **Play** : il permet de lancer la simulation ou de la mettre en pause.
- **Grilles** : il permet d'avoir une grille pour situer les capteurs en espace.

- **Clear** : il efface tous les messages qui transitent entre les capteurs.
- **Arrêt** : il met fin à la simulation.

IV.3 LEACH (Low-Energy Adaptive Clustering Hierarchy)

LEACH est un protocole de routage destiné aux réseaux de capteurs. Son principal avantage est de minimiser la consommation énergétique des éléments du réseau. C'est un protocole hiérarchique, car le réseau est divisé en clusters, et chaque cluster possède un nœud 'maître' appelé cluster-head. Ce dernier prend en charge la gestion de son cluster. Il est élu périodiquement parmi les nœuds formant le cluster, en fonction de l'état de sa batterie.

Ce protocole permet ainsi la structuration du réseau de manière hiérarchique dans le but est d'économiser l'énergie des capteurs comme le montre la figure IV-2.

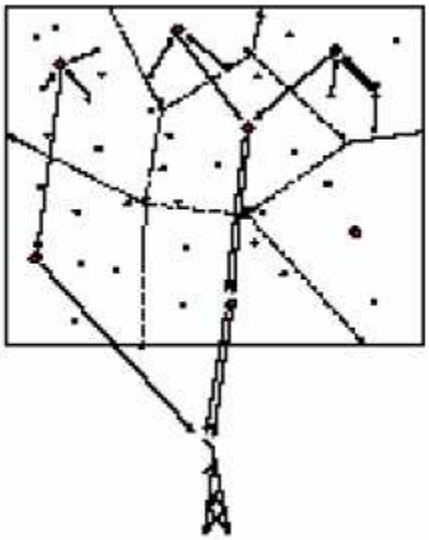

Figure IV-2: Schémas de communication dans LEACH.

IV.3.1 Description de l'algorithme LEACH

L'avantage du protocole LEACH c'est qu'il permet de réduire le nombre de nœuds qui communiquent directement avec la base station et ceci en formant des chefs de groupes (cluster-heads). Ensuite, les autres nœuds voisins se connectent et deviennent membre de ce cluster, ainsi ils dépensent le minimum d'énergie. Seuls les cluster-heads sont autorisés à communiquer avec la station de base.

Chaque cluster-head alloue une durée bien déterminé à un voisinage pour établir un lien de communication, d'où ces nœuds peuvent alors passer en mode endormi pendant le reste du temps. Une fois que les clusters sont fixés, ces derniers sont appelés à

consommer beaucoup d'énergie, ce qui va engendrer la mort de ces nœuds. Pour éviter ce grave problème, LEACH utilise la notion de cycles (Rounds). Au début de chaque cycle, chaque nœud doit décider s'il doit être sélectionné comme un cluster en se basant sur un facteur probabiliste et sur le fait qu'il n'était pas cluster-head dans les cycles antérieurs, ou bien il doit joindre un cluster. Ainsi ce protocole dynamique permet de réduire énormément la perte d'énergie causée par un statique clustering et permet alors d'étendre la durée de vie de chaque nœud.

L'objectif de protocole LEACH est d'optimiser la consommation d'énergie afin d'assurer une durée de vie plus longue au réseau d'une part et d'autre part il répartit la charge entre les nœuds de telle sorte que la différence entre la mort du premier et du dernier soit réduite.

LEACH est considéré comme étant le premier protocole de routage hiérarchique basé sur les clusters (Figure.IV-3). Il est aussi l'un des algorithmes de routage hiérarchiques les plus populaires pour les RCSF. L'idée est de former des clusters de nœuds capteurs en se basant sur la puissance du signal reçu et d'employer le cluster-head local comme routeur vers la station de base. Cela économiserait de l'énergie puisque seul les cluster-heads effectueront une transmission vers le puits. Le nombre optimal de cluster-heads dans un réseau de capteurs est de 5% par rapport au nombre total de nœuds. Tous les processus de données tel que la fusion et l'agrégation sont locaux aux clusters. Le cluster-head est élu périodiquement en fonction de son niveau d'énergie pour équilibrer la consommation d'énergie des nœuds.

LEACH est totalement distribué et n'a besoin d'aucune connaissance globale du réseau. Cependant il utilise un routage à saut unique où chaque nœud peut transmettre directement au cluster-head. Mais il n'est pas applicable aux réseaux qui sont déployés sur une grande surface. De plus, le clustering dynamique ajoute une grande surcharge comme le changement des cluster-heads ce qui peut diminuer le gain en énergie.

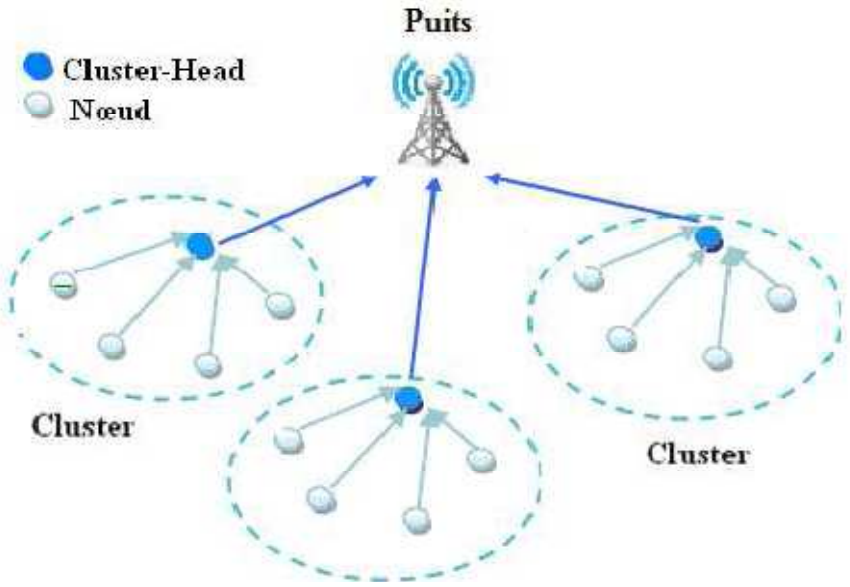

Figure IV-3 : Routage hiérarchique basé sur le clustering.

IV.3.2 La durée de vie du réseau

Au niveau du protocole LEACH, la durée de vie du réseau est faible, parce que dans LEACH, les nœuds s'épuisent plus rapidement vue la distance entre les CHs et leurs membres d'un coté et la distance entre les CHs et la station de base comme dans la figure IV-4 En effet, les phases d'initialisation c'est-à-dire les phases de formation de clusters qui induisent un nombre important de messages de contrôle vont se faire à chaque nouveau round impliquant une consommation d'énergie supplémentaire.

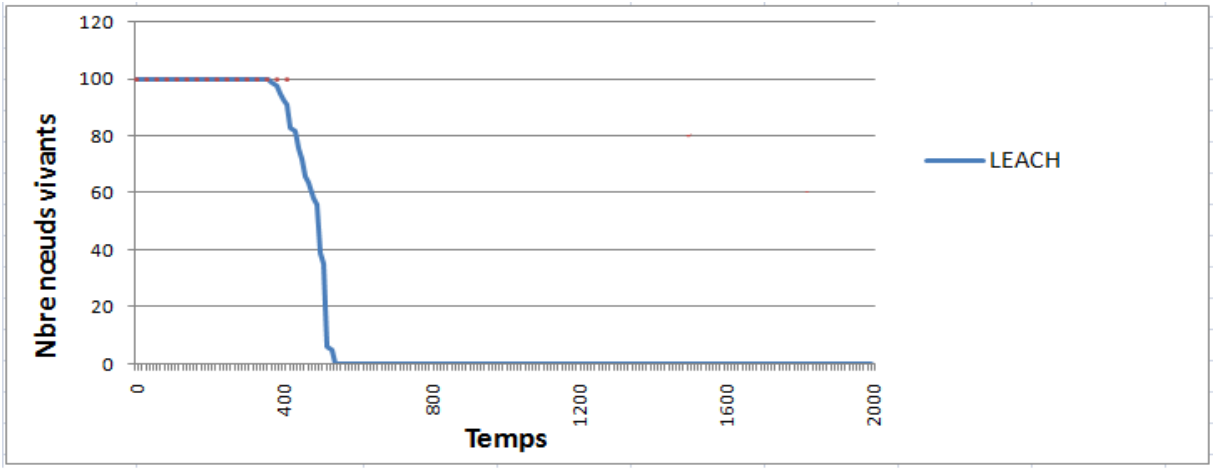

Figure IV-4 : La durée de vie du réseau au niveau de LEACH.

VI.4 Implémentations et déroulements

Notre travail s'est déroulé en deux phases :

1. Implémentation et évaluation du protocole LEACH.
2. Implémentation et évaluation du nouveau protocole T-LEACH.

Nous donnons donc un aperçu de notre implémentation afin de voir l'évolution de LEACH jusqu'à T-LEACH.

VI.4.1 Les fichiers de l'application

Notre application est formée des composants suivants :

- un module, appelé «MHLeachPSM.nc ».
- une configuration, appelée «MHLeachRouter.nc ».
- le fichier d'entête, appelé «MH.h ».

VI.4.2 Implémentation du protocole LEACH

Dans cette section, nous décrivons les structures de données ainsi que les principaux commandes et événements nécessaires pour l'implémentation du protocole LEACH.

VI.4.2.1 Structures de données

Le paquet dans TinyOS est envoyé dans une structure appelée TOS_Msg, qui est contenue dans un champ « int8_t data[TOSH_DATA_LENGTH] ». Les structures de données du paquet diffèrent selon le rang du nœud (station de base, CH ou membre).

A) Le nœud puits

```
typedef struct PUITS
{uint16_t ID; //l'identificateur du puits qui correspond à tos_local_address=0
uint8_t round; //le round courant
float probability; //la probabilité que chaque nœud devienne CH
uint8_t Depth; //la puissance du signal d'un CH dans le réseau
}PUITS;
```

B) Le nœud CH

```
typedef struct CLUSTER_HEAD
{
uint16_t ID_CH; //l'identificateur de chaque CH qui correspond à tos_local_address
uint16_t ID_MEMBRE; //l'identificateur du membre qui appartiendra à ce CH
uint8_t data_agre; //la donnée agrégée à envoyer au nœud puits
uint16_t SLOT_ATT; //le slot attribué à chaque membre
uint16_t FREQ; //la fréquence avec laquelle un membre envoie sa donnée
}CLUSTER_HEAD;
```

C) Le nœud membre

```
typedef struct MEMBRE
{
uint16_t ID_MEMBRE;//l'identificateur de chaque membre qui correspond à
tos_local_adress
uint16_t ID_CH; //l'identificateur du CH auquel appartiendra le nœud membre
uint8_t temp; //la température captée
}MEMBRE;
```

IV.4.2.2 Environnement d'exécution du simulateur

Cygwin est une couche d'émulation de l'API Linux qui permet d'avoir une interface Unix sous Windows comme le montre la figure IV-5.

Figure IV-5 : l'interface Cygwin.

- Tout d'abord, on accède au fichier home par la commande suivante : cd /
- Après, on met le chemin de notre application : **cd opt/tinyos-1.x/apps/leach** pour accéder à l'application « LEACH».
- Ensuite, on la compile par la commande : **make pc**.
- Et enfin on l'exécute par : la commande **export DBG=usr2**, et la commande **tinyviz –run build/pc/main.exe nbre_capteurs.**

VI.4.3 Déroulement

Dans cette partie, nous expliquons et nous déroulons les phases de l'algorithme LEACH en faisant appel à TinyViz. Un fichier de configuration est créé et permet à TinyViz de se lancer avec des paramètres spécifiés. Ces derniers représentent : le nombre et l'emplacement des capteurs, la durée de la simulation et les plugins que nous souhaitons activer dès le début de la simulation comme Debug Messages. A propos des captures d'écran de TinyViz, nous nous limitons, dans cette étape, à la partie où l'on peut visualiser les capteurs.

1. Déclenchement du nouveau round, et annonce des CHs: La figure VI-1 représente les transmissions broadcast qui se passent durant les différentes étapes de l'algorithme LEACH. Une transmission broadcast est repérée par un cercle bleu. Le nœud puits envoie un broadcast aux nœuds voisins pour l'annonce du round. Ses voisins prennent le relai en envoyant à leur tour selon une transmission broadcast. De plus, nous pouvons voir que le nœud 15 est élu CH. Cet évènement est marqué par l'activation du LED rouge des CH. Ensuite, le CH 15 diffuse une annonce pour signaler son statut comme dans la figure IV-6.

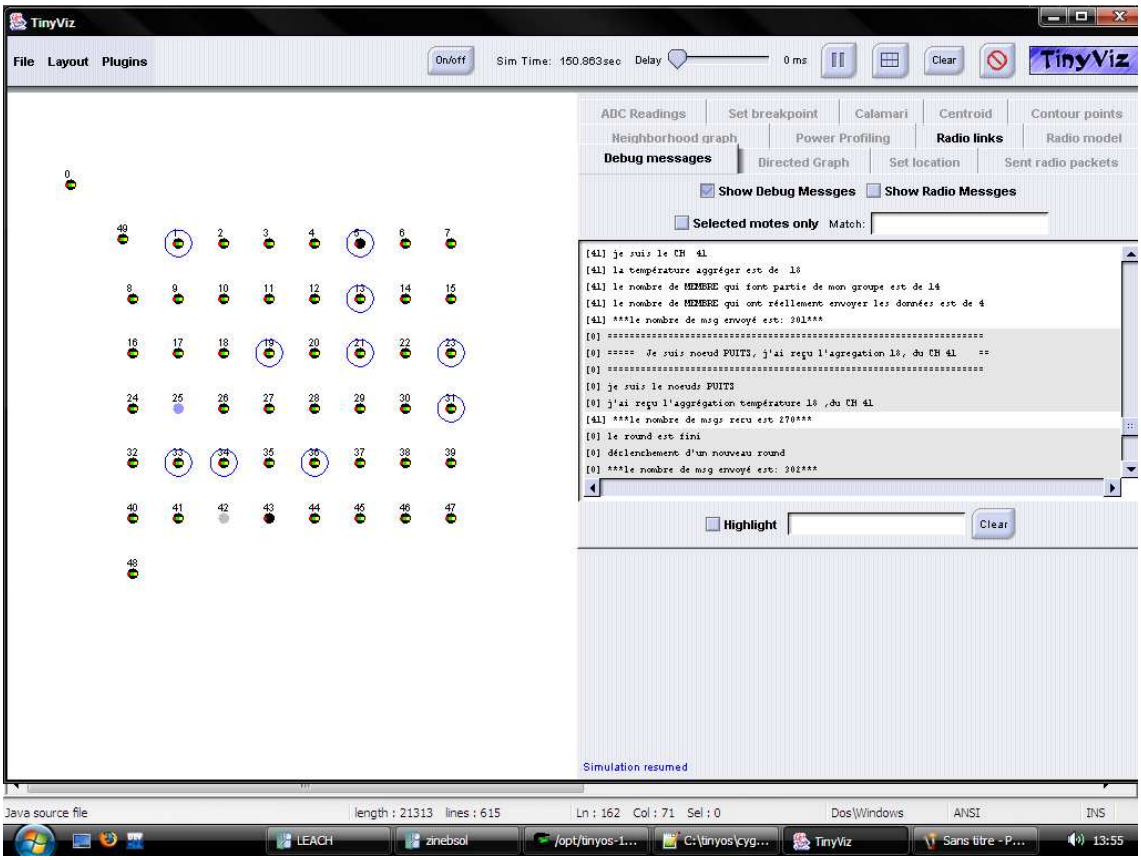

Figure IV-6 : Déclenchement et relai du nouveau round, annonce du CH 15.

2. Formation de clusters et envoi des données:

La figure VI-7 représente quelques transmissions unicast qui se passent durant les différentes étapes de l'algorithme LEACH. Une transmission unicast est repérée par une flèche.

Durant la première étape, les nœuds non-CH répondent à l'annonce du CH le plus proche. La figure VI-7 illustre la formation du cluster du CH 4**.** Quant à la seconde étape, chaque membre capte une donnée dans notre cas il s'agit de la température et attend le début de son slot pour qu'il puisse l'envoyer à son CH. Nous avons utilisé une application qui retourne la température sur une zone donnée afin de pouvoir valider l'implémentation du protocole LEACH.

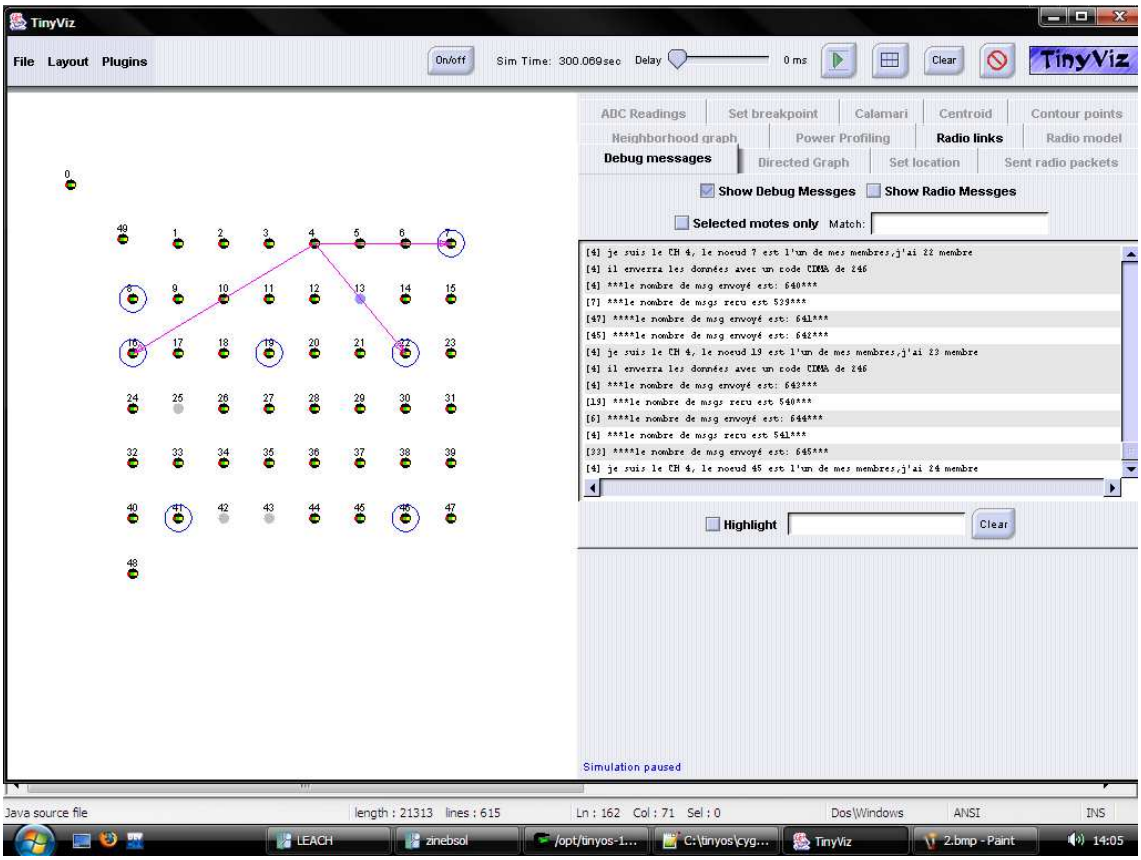

Figure IV-7: Formation de groupes et envoi des données.

3. Envoi des résultats d'agrégation des températures au nœud puits : Dans la figure IV-7, le CH 4 agrège les températures reçues et envoie son résultat d'agrégation au nœud puits comme dans la figure IV-9.

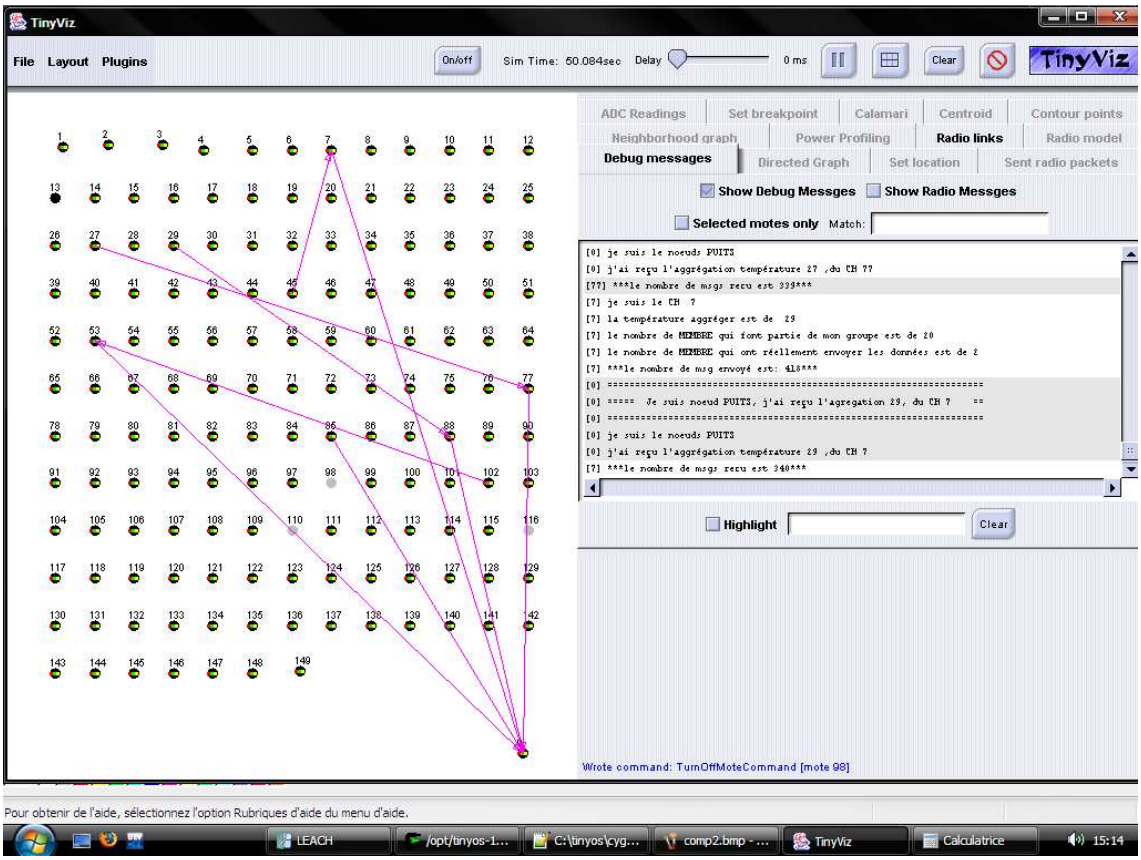

Figure IV-8: Envoi du résultat d'agrégation du CH au nœud puits.

IV.4.4 Implémentation du protocole T-LEACH

Dans cette section, nous donnons les commandes et les évènements nécessaires pour l'implémentation du protocole T-LEACH, ainsi que les modules et les outils de tolérance aux pannes utilisés pour assurer les services de tolérance intégrés dans ce protocole.

IV.4.4.1 Structures de données

En plus des champs utilisés dans les structures de données mis en place pour LEACH, T-LEACH a besoin des informations suivantes :

A) Le nœud puits

```
typedef struct PUITS
{uint16_t ID; //l'identificateur du puits qui correspond à tos_local_address=0
uint8_t round; //le round courant
float probability; //la probabilité que chaque nœud devienne CH
uint8_t Depth; //la puissance du signal d'un CH dans le réseau
}PUITS;
```

B) Le nœud CH

```
typedef struct CLUSTER_HEAD
{
uint16_t ID_CH; //l'identificateur de chaque CH qui correspond à tos_local_address
uint16_t ID_CH2; //l'identificateur de chaque CH adjoint
uint16_t ID_MEMBRE; //l'identificateur du membre qui appartiendra à ce CH
uint8_t data_agre; //la donnée agrégée à envoyer au nœud puits
uint16_t SLOT_ATT; //le slot attribué à chaque membre
uint16_t FREQ; //la fréquence avec laquelle un membre envoie sa donnée
}CLUSTER_HEAD;
```

C) Le nœud membre

```
typedef struct MEMBRE
{
uint16_t ID_MEMBRE;//l'identificateur de chaque membre qui correspond à
tos_local_adress
uint16_t ID_CH; //l'identificateur du CH auquel appartiendra le noeud membre
uint16_t ID_CH2; //l'identificateur du CH adjoint
uint8_t temp; //la température captée
}MEMBRE;
```

Dans LEACH l'occurrence d'une panne permet la perte des données car c'est le cluster-head qui envoie les données qu'il a reçu de la part de ces membres à la station de base. Mais dans T-LEACH, un autre cluster-head adjoint sera élu des que le cluster head principal tombe en panne ce qui permet d'assurer la livraison des données à la station de base. Dans cet exemple, c'est le cluster-head 21 qui est en panne. Donc avant que les paquets envoyés par les nœuds membres seront perdus, le cluster-head adjoint 22 va le remplacer et ceci est réalisé après notre amélioration du protocole LEACH qui sera tolérant aux pannes. La figure IV-9 résume le fonctionnement de T-LEACH dans le cas où un cluster-head est en panne.

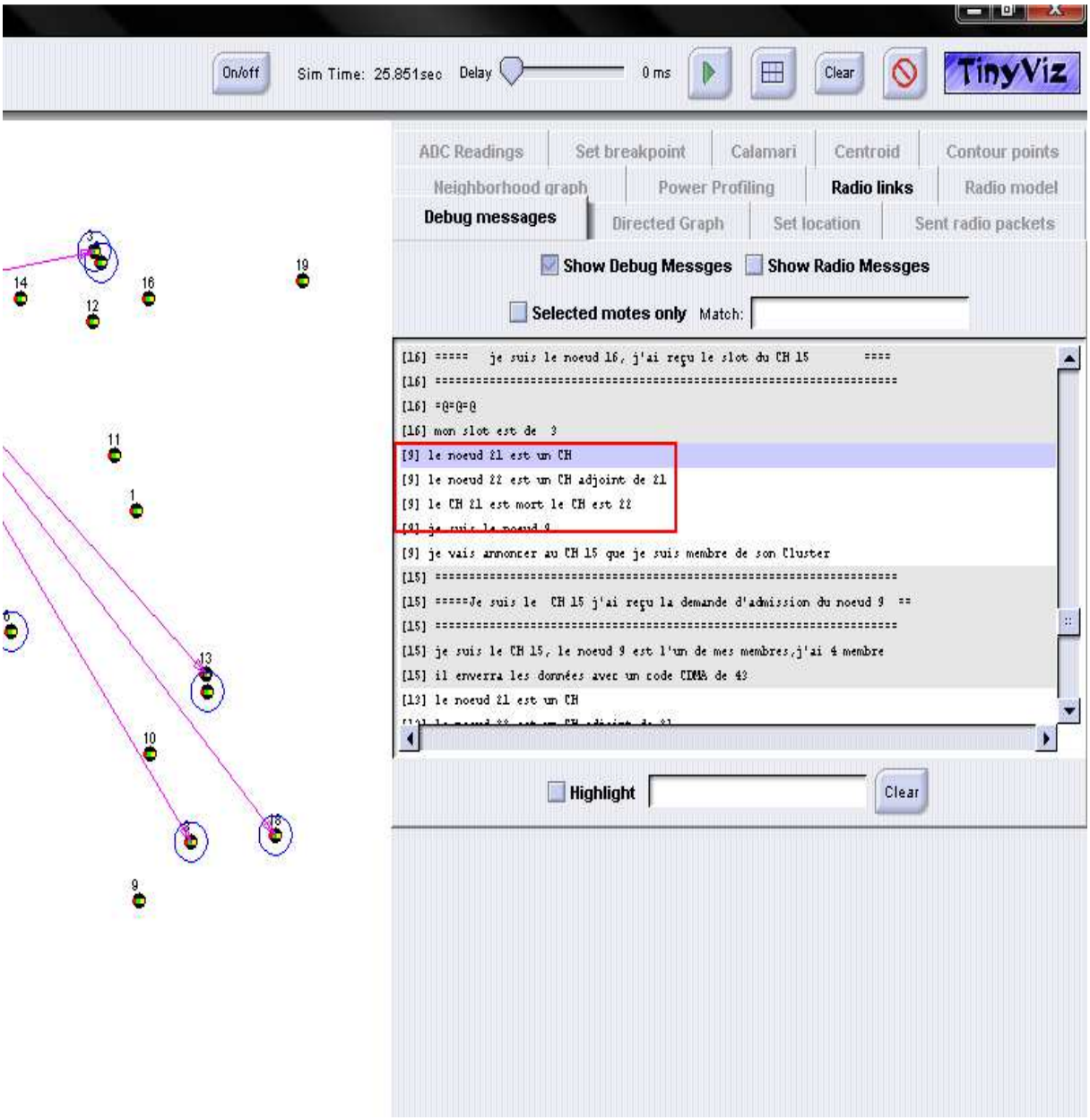

Figure IV-9 : le cluster head 22 remplace le cluster head 21.

IV.5 Simulation et évaluation de performances

Pour évaluer les performances T-LEACH, nous avons procédé à le comparer au protocole de routage LEACH. Pour cela, nous avons effectué des simulations avec les mêmes paramètres et métriques pour les deux protocoles.

IV.5.1 Métriques à évaluer

Pour pouvoir comparer les performances T-LEACH avec celles de LEACH, il est commode de mesurer une certaine métrique :

IV.5.1.1 Perte de paquets

Le choix de cette métrique, comme étant un critère de performance, revient à sa nécessité dans certaines applications où les données échangées sont très critiques. Pour la mesurer, nous calculons la moyenne des taux de perte de paquets de températures entre les membres et leurs CH, et de paquets d'agrégation de ces températures entre les CH et la station de base. Ainsi, le protocole T-LEACH ne doit pas mener à une forte perte de paquets de données par rapport à LEACH. De plus, nous vérifions, pour les deux protocoles, l'effet de la panne du cluster-head sur l'augmentation de nombre de paquets de données perdus.

IV.5.2 Résultats et interprétations

Dans cette partie, nous évaluons d'abord les métriques déjà citées et nous les comparons pour les deux protocoles LEACH et T-LEACH. Par la suite, nous simulons la panne d'un cluster-head dans le but de vérifier son effet pour les deux protocoles.

IV.5.2.1 Perte de paquets

Pour tester le taux de pertes de paquets, il est nécessaire de calculer le ratio des paquets perdus et des paquets envoyés. Voici le tableau de résultats de ce test:

On a fait la test avec plusieurs façons pour bien présenter le taux de perte des paquets.

Dans le premier cas on a fixé le nombre des nœuds mais on a varié le nombre de cluster head désactivé de 1 jusqu'à 5 clusters.

- Pour 50 nœuds : Dans la figure IV-10 l'exécution des 50 nœuds avec les cluster-heads désactivés

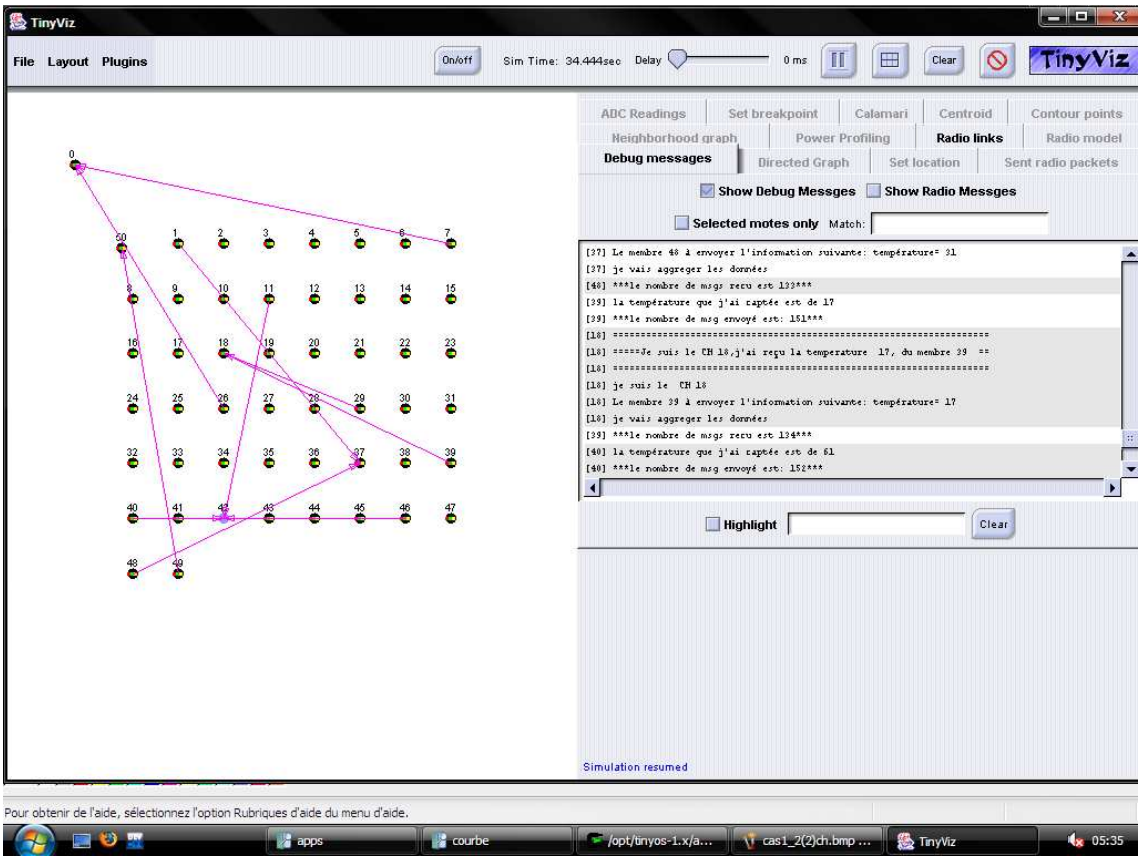

Figure IV-10 exécution des 50 nœuds avec 1 CH désactivés.

Le taux de perte est présenté dans le tableau suivant.

	Nombre de cluster-heads désactivés pour 50 nœuds				
	1	2	3	4	5
LEACH	10,493%	14,556%	20,8%	22,223%	27,079 %
T-LEACH	1,51%	11,76%	14,772%	16,463%	17,772 %

Tableau IV-1 : Taux de perte de paquets (50 nœuds)

Comme illustre la figure IV-11, nous remarquons que le taux de paquets perdus dans LEACH augmente plus que dans T-LEACH à chaque fois qu'on augmente le nombre des cluster-heads désactivé.

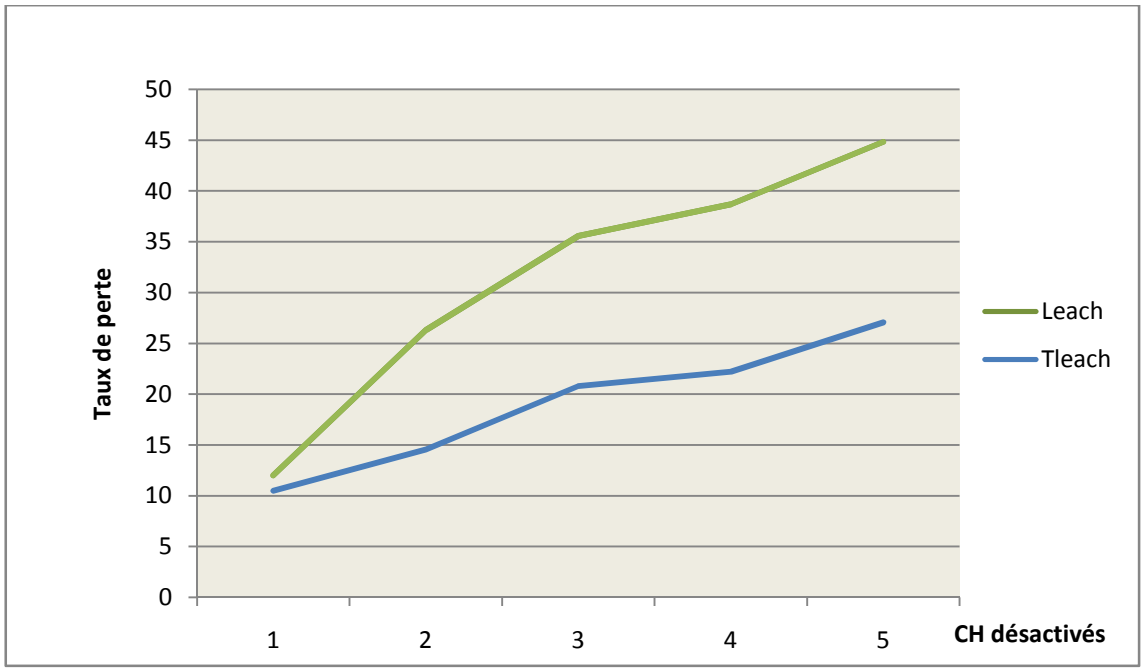

Figure IV-11: Taux de pertes de paquets pour 50 nœuds.

❖ Pour 100 nœuds : la figure IV-12 nous montre l'un des cas de l'exécution dans un réseau de taille de 100 nœuds.

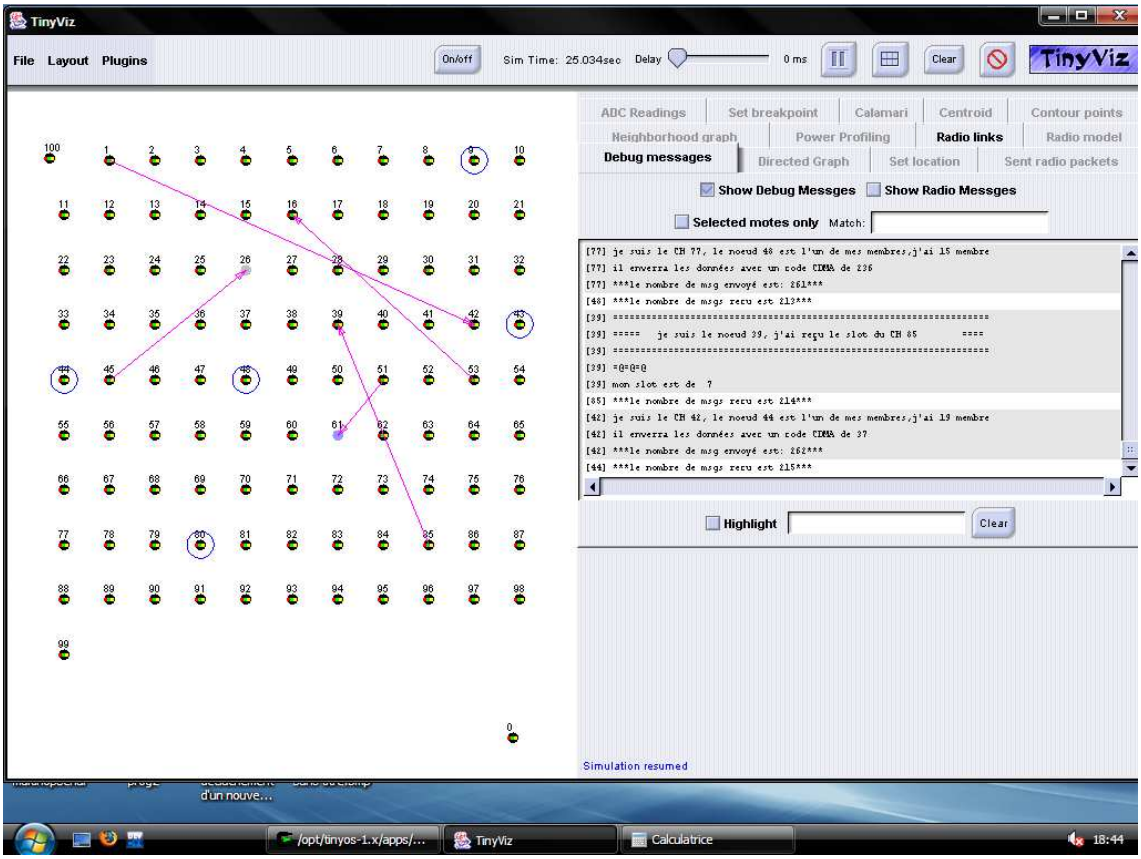

Figure IV-12: exécution des l'un des cas de désactivation des CH pour 100 nœuds.

Le tableau suivant montre bien le l'augmentation de perte des paquets.

	Nombre de cluster-heads désactivés pour 100 nœuds				
	1	2	3	4	5
LEACH	5,882%	14,748%	41,545%	42,666%	48,497 %
T-LEACH	1,403%	6,206%	11,267%	17,721%	19,642%

Tableau IV-2: Taux de perte de paquets (100 nœuds)

Comme le montre la figure IV-13 ci-dessous, nous remarquons que les taux de pertes de paquets échangés sont tolérables pour les deux protocoles. À chaque fois que le nombre des cluster-heads désactivé augmente, le taux de perte augmente pour les deux protocoles. De plus nous pouvons bien distinguer que le taux de perte est plus élevé dans LEACH que dans T-LEACH.

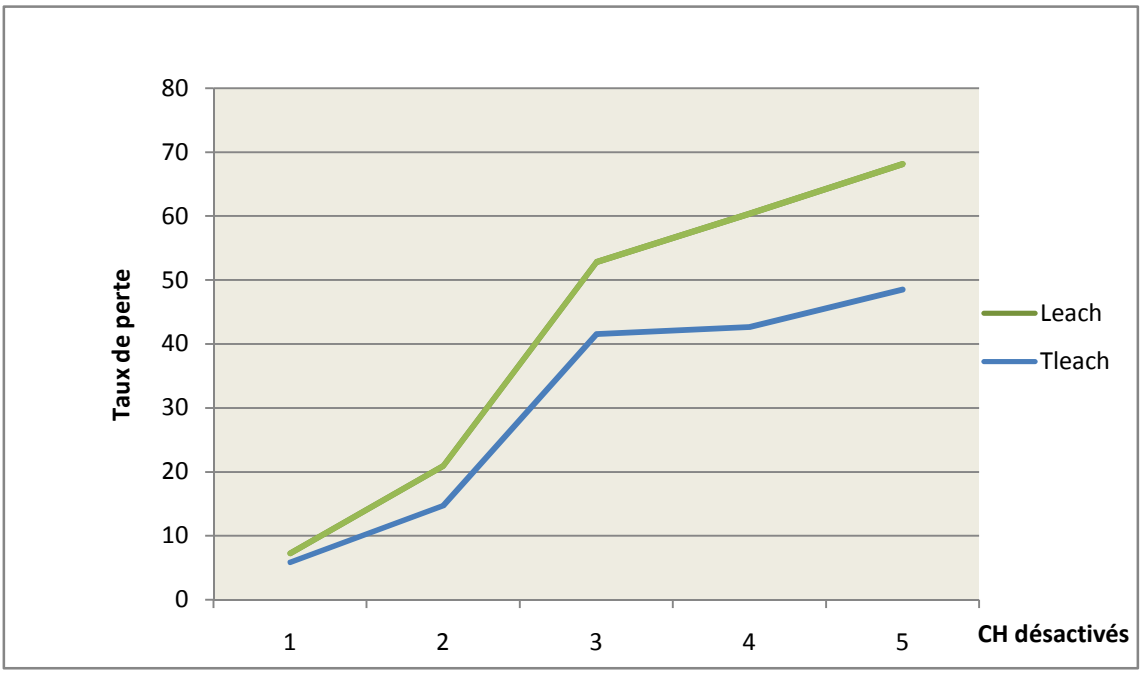

Figure IV-13: Taux de pertes de paquets pour 100 nœuds.

Dans le deuxième cas c'est le nombre des clusters head adjoint désactivé qui sont fixée et les nombres des nœuds du réseau varient. La figure suivante montre l'un des cas des exécutions.

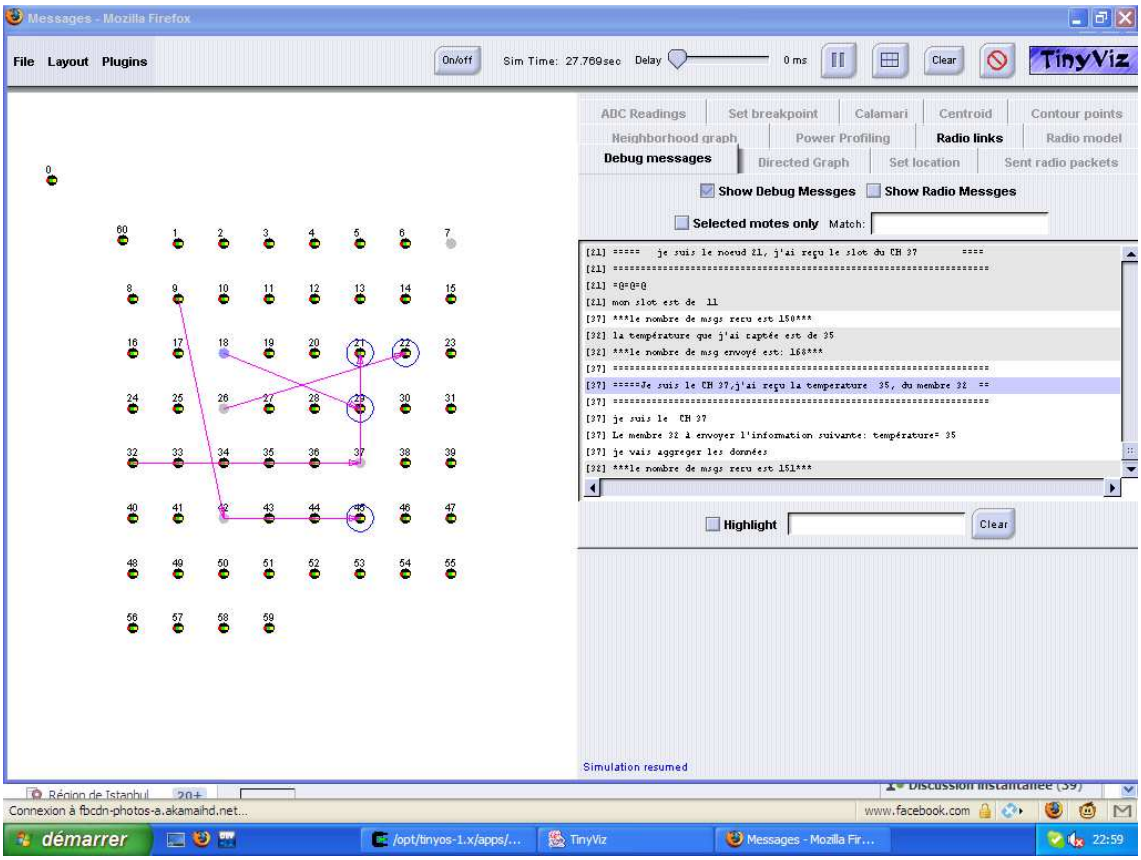

Figure IV-14: exécution des nœuds pour 5CH désactivé.

Le tableau suivant montre le taux de perte des différents nœuds du réseau.

	Nombre de nœuds dans le réseau pour 5 CH désactivés			
	50	60	80	100
LEACH	27,972%	17,090%	85, 446%	48, 497%
T-LEACH	17,647%	16,470%	16,888%	19, 642%

Tableau IV-3: Taux de perte de paquets (5 CH désactivés)

Comme le montre la figure IV-15 le taux de perte varie selon le nombre des nœuds, des fois pour un certain nombre le taux est élevé et pour certain le taux diminue pour LEACH mais pour T-LEACH le taux de perte est bien plus petit et presque stable car les cluster-head adjoint prennent la responsabilité de faire passé les données à la station de base.

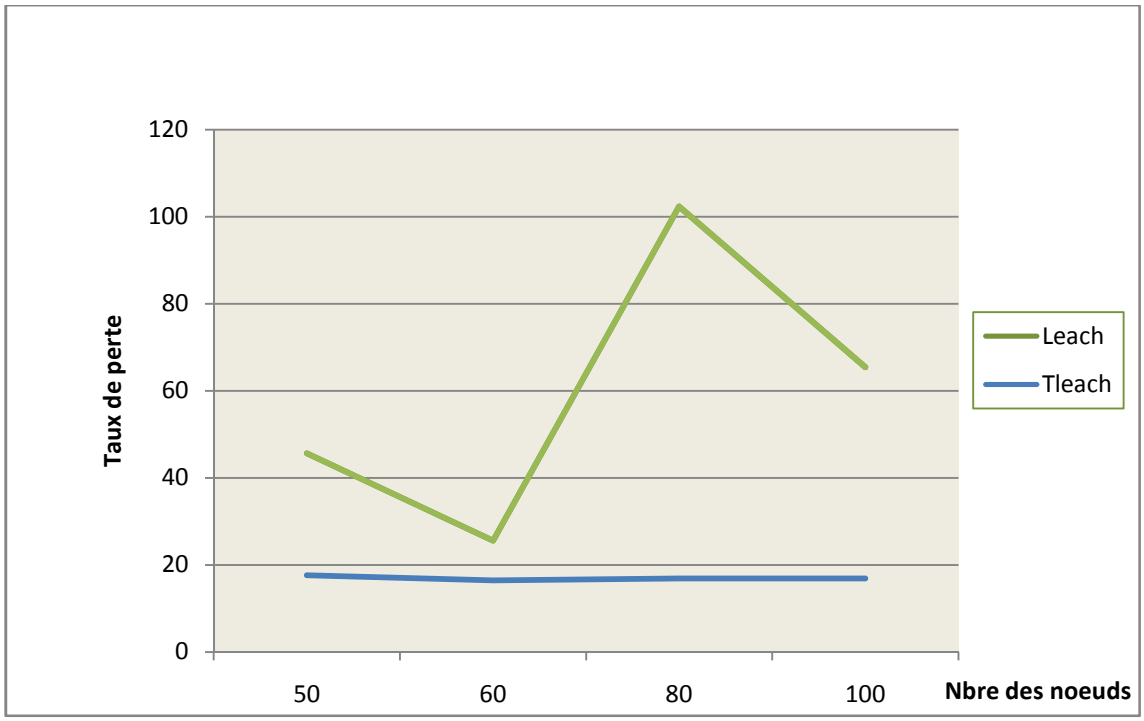

Figure IV-15 : Taux de perte pour 5 CH désactivés.

IV.6 Conclusion

Dans ce chapitre, nous avons présenté l'environnement pour implémenter et évaluer le protocole LEACH. Dans cet environnement, on trouve TinyOS qui est un système d'exploitation léger dédié pour les réseaux de capteurs, le langage NesC qui est un langage orienté composant et TOSSIM qui est un simulateur pour les RCSF.

L'évaluation du protocole LEACH , nous a permis de déduire que ce protocole n'est pas tolérant aux pannes. Dans cette optique, pour pallier cette limite, nous avons proposé une version améliorée de ce protocole appelée T-LEACH de telle sorte qu'il soit tolérant aux pannes.

Les résultats de simulation ont montré que le taux de paquets perdus dans T-LEACH est inférieur à celui dans LEACH.

Conclusion générale

Les réseaux de capteurs sont composés d'un très grand nombre de dispositifs de communication ultra petits, autonomes avec des ressources de calcul et d'énergie limitées. Ils sont actuellement considérés comme l'une des technologies qui bouleverse notre façon de vivre, grâce à leur utilisation dans différents domaines d'application.

Cependant, les réseaux de capteurs sans fil rencontrent plusieurs problèmes qui affectent leur bon fonctionnement dû à leurs caractéristiques ; tels que les limitations de batterie, le type de communication, les environnements hostiles où sont déployés les capteurs ou encore leur faible coût. Par ailleurs, ces réseaux sont caractérisés par les pannes des nœuds qui peuvent causer un dysfonctionnement du réseau en entier. Dans cette optique, il est commode de proposer des protocoles de routage tolérants aux pannes.

Une panne au niveau d'un capteur peut se produire à cause d'une perte de connexions sans fil due à l'extinction du capteur suite à l'épuisement de sa batterie, ou tout simplement à une destruction physique accidentelle ou intentionnelle par un ennemi. Par conséquence, il faut faire face à ces pannes en proposant des protocoles tolérants aux pannes.

Dans ce mémoire, nous avons réalisé une étude pour atteindre un routage efficace avec tolérance aux pannes dans les réseaux de capteurs sans fil. Cet aspect est fondamental pour ce genre de réseau où le routage se réalise en collaboration avec les différents nœuds du réseau. De ce fait, un protocole de routage doit prendre en compte les contraintes matérielles d'un capteur : une batterie faible, une capacité de stockage modeste, une bande passante faible, etc.

L'approche clustérisée qui permet de partitionner le réseau en zones, est une approche prometteuse. Pour atteindre cet objectif, nous avons proposé une amélioration de protocole hiérarchique de routage nommé LEACH basé sur une topologie structurée en zones.

Afin de valider les améliorations apportées par notre protocole en termes de prolongement de vie du réseau, nous avons simulé le fonctionnement de notre algorithme dans le système d'exploitation TinyOS en utilisant TOSSIM et l'émulateur Tinyviz puis on a comparé les résultats fournis avec le protocole LEACH.

Les résultats fournis par notre implémentation du protocole LEACH offre une tolérance aux pannes ce qui permet d'assurer une livraison fiable des données dans les RCFS.

Enfin, comme perspectives nous envisageons d'améliorer les performances de notre protocole de routage et l'implémenter sur des capteurs réels.

Références bibliographique

[1] M. Sedrati, L. Aouragh, L. Guettala & A. Bilami – « Etude des performances des protocoles de routage dans les réseaux mobiles ad-hoc », dans *Proc.4th Int'l Conf. on Computer Integrated Manufacturing (CIP'2007)*, 2007.

[2] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam & E. Cayirci – « Wireless sensor networks: a survey », *IEEE Communications Magazine* **40** (2002), no. 8, p. 102–114.

[3] C. Basile, M.-O. Kill¼ian & D. Powell – « A survey of dependability issues in mobile wireless networks », Tech. Report 02637, LAAS, Toulouse, 2002.

[4] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. Wireless sensor networks :a survey. Computer Networks (Elsevier), vol.38, no.4, pp.393-422, March 2002.

[5] Paolo Santi, "Topology Control in Wireless Ad Hoc and Sensor Networks". Hardcover,july 2005.

[6] S.J., Pister Kristofer. http://webs.cs.berkeley.edu/tos/. TinyOS. [En ligne] 2009.

[7] Kristofer, Pister. http://robotics.eecs.berkeley.edu/~pister/SmartDust/. SmartDust. [En ligne] 2001.

[8] Crossbow. MICA2 Data sheet. [Online] 2009. http://www.xbow.com/products/Product_pdf_files/Wireless_pdf/MICA2_Datasheet

[9] I. F. Akyildiz, W. Su , Y. Sankarasubramaniam , E. Cayirci. Wireless sensor networks: a survey. 2002, Vol. 38, pp. 393-422.

[10] Brown, M. J. Users Guide Developed for the JBREWS Project. Los Alamos National Laboratory of California University. 1999. Technical repport LA-UR-99-4676.

[11] Andrews, P. Johnson and D.C. Remote continuous monitoring in the home. Telemedicine and Telecare. June 1996, Vol. 2, 2, pp. 107-113.

[12] Michael Fitzgerald. Technnology Review : Tracking a Shopper's Habits. Technology Review. [En ligne] 04 August 2008. http://www.technologyreview.com/computing/21161/.

[13] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. "wireless sensor networks : a survey". Elsevier Science, 38(4), 2002.

[14] B. Selic. "fault tolerance techniques for distributed systems". http ://www.ibm.com/developerworks/rational/library/114.html, 2004.

[15] F. Koushanfar, M.Potkonjak, and A. Sangiovanni-Vincentelli. "Handbook of Sensor Networks : Compact Wireless and Wired Sensing Systems", volume 36, chapter Fault Tolerance in Wireless Sensor Networks. CRC, 2005.

[16] I.F. Akyildiz, W.Su, Y.Sankarasubramaniam, E.Cayirci, les réseaux de capteurs sans fil:l'un des Réseaux de l'enquête informatique,: la revue internationale du travail en réseau informatique et des télécommunications, v.38 n.4, pp.393-422, 2002.

[17] Jamal N. Al-Karaki Ahmed E. Kamal, “Routing Techniques in Wireless Sensor Networks: A Survey”, Dept. of Electrical and Computer Engineering Iowa State University.

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

[19] D. Elorrieta, Protocoles de routage pour l'interconnexion des réseaux Ad-Hoc et UMTS. Mémoire DEA 2007.

[20] K. Yedavalli, B. Krishnamachari, “Enhancement of the IEEE 802.15.4 MAC Protocol for Scalable Data Collection in Dense Sensor Networks”. Proc. WiOpt08, Berlin, Germany, April 2008.

[21] JN Al Karaki et AE Kamal, «Techniques de routage dans les réseaux de capteurs sans fil: Une enquête", IEEE Wireless Communications, vol. 11, Issue 6, Egypte, 2004,PP. 6-28.

[22] K. Akkaya et M. Younis, "Une enquête sur les protocoles de routage pour les réseaux de capteurs sans fil", dans le Journal Elsevier ad hoc Réseau, vol. 3/3, 2005, PP. 325-349.

[23] Ajay, N.Tarasia, S. Dash, S.Ray, ARSwain "Une erreur dynamique tolérant protocole de routage pour prolonger la durée de vie des réseaux de capteurs sans fil» (IJCSIT)International Journal of Computer Science et Technologies de l'Information, Vol. 2 (2),2011, 727-734.

[24] Ajay, N.Tarasia, S. Dash, S.Ray, ARSwain protocole de routage tolérant aux fautes multi-niveaux avec des horaires du sommeil (FMS) pour les réseaux de capteurs sans fil" European Journal of Scientific Research ISSN 1450-216X Vol.55 n ° 1 (2011),pp.97-108.

[25] K. Kulothungan, J. Angel Arul Jothi, A. Kannan «Une erreur de protocole de routage adaptatif tolérant pour les réseaux de capteurs sans fil»European Journal of Scientific Research ISSN 1450-216X N ° 1 Vol.60 (2011), pp 19-32.

[26] Guowei Wu, ChiLin, Feng Xia, Lin Yao, il Zhang et Liu Bing «Saut dynamique en temps réel Fault-Tolerant protocole de routage pour les réseaux de capteurs sans fil» de la Fondation nationale des sciences naturelles de Chine par la concession numéro60703101 et n ° 60903153 (2010).

[27] Zamree Che-Aron, Wajdi Al-Khateeb, et Farhat Anwar «Le Renforcement de tolérance de panne Mécanisme de protocole de routage AODV pour le réseau de capteurs sans fil» IJCSNS International Journal of Computer Science et de sécurité réseau, Vol.10 No.6, Juin 2010.

[28] F. Z Benhamida, Y. Challal «FaT2D: Fault Tolerant Diffusion Réalisé pour les réseaux de capteurs sans fil »2010 de la Conférence internationale sur la disponibilité, la fiabilité et la Security 2010 IEEE DOI 10.1109/ARES.2010.35 112.

[29] Mathieu Badnet, Nicolas Belloir «Réseaux de capteurs : Mise en place d'une plateforme de test et d'expérimentation », Master Technologie de l'Internet 1ère année, France, 2005/2006.

[30] Sylvie Tixier, **« TinyOS »**, Mini rapport, LIF12, Université Lyon 1, 6 Décembre 2007.

[31] C. Han, E. Kohler, M. Srivastava, R. Kumar, R. Shea, « A Dynamic Operating System for Sensor Nodes», Proceedings of the 3rd International Conference on Mobile Systems, Applications and Services (Mobisys), Page(s): 163-176, University of California, Los Angeles, June 2005.

[32] Adam Dunkels, Björn Grönvall, Thiemo Voigt, « Contiki - a Lightweight and Flexible Operating System for Tiny Networked Sensors », 29th Annual IEEE International Conference on Local Computer Networks, Pages: 455–462, Swedish Institute of Computer Science, 2004.

[33] Cormac Duffy, Cormac J. Sreenan, John Herbert, Utz Roedig, « A Performance Analysis of MANTIS and TinyOS », Technical Report CS-2006-27-11, University College Cork, Ireland, November 2006.

[34] H. Alatrista, J. Mathieu, K. Gouaïch S. Aliaga, « Implémentation de protocoles sur une plateforme de réseaux de capteurs sans fil », TER master 1 informatique, Université de Montpellier II, 29 Avril 2008.

[35] David Gay, Philip Levis, « TinyOS Programming », Livre, ISBN: 0521896061, Nombre de Pages: 264, Presse de l'université de Cambridge, 28 Juin 2006.

[36] Yacine Challal, « Réseaux de Capteurs Sans Fils », Cours, Systèmes Intelligents pour le Transport, Université de Technologie de Compiègne, France, 17 Novembre 2008.
[37] Wassim Znaidi, « Modélisation formelle de réseaux de capteurs à partir de TinyOS », Projet de fin d'études, Ecole Polytechnique de Tunisie, 2006.

Annexe

A.1 Procédure d'installation sous Windows XP

Ce guide propose l'installation du principal outil nécessaire au bon fonctionnement du système, notamment Cygwin (couche d'émulation de l'API Linux) qui permet d'avoir une interface Unix sous Windows. Cygwin est un environnement d'émulation Linux qui permet d'avoir un shell et de compiler et exécuter les programmes Linux (On dispose ainsi de gcc, apache, bash, etc.).

Figure A-1 : Cygwin.

1- Télécharger le fichier tinyos-1.1.0-1is.exe de la source http://www.tinyos.net/dist-1.1.0/tinyos/windows/ .

2- Exécuter ce fichier pour installer la version 1.1.0 sous windows XP. L'installation se fait automatiquement. Un raccourci de Cygwin est sauvegardé sur le bureau.

3- Accéder à **C:\tinyos\cygwin\opt\tinyos-1.x\doc\tutorial\verifyhw.html** et suivre les étapes que contient cette page afin de vérifier si l'installation est bien réussie.

A.2 Installation de TinyViz

Les concepteurs développent au fur et à mesure l'outil TinyViz sans mettre à jour les fichiers sources déjà existants dans les anciennes versions. Cela ne permet pas de lancer TinyViz dans des conditions normales. Pour pouvoir le lancer, il est nécessaire de passe par les étapes suivantes:

1- Installer TinyOS-1.0

2- Accéder à: **cd /opt/tinyos-1.x/tools/java**

ET taper : **make**

3- Installer les mises à jour de NesC1.1.1 and TinyOS1.1.15.

Pour se faire, rechercher sur le net http://www.tinyos.net/dist-1.1.0/tinyos/windows/ ces mises à jour en téléchargeant le **rpm** et le mettant dans **C:\tinyos\cygwin\home\PLANETE PC**

Et taper dans le **shell**:

rpm -ivh --ignoreos nesc-1.1.2b-1.cygwin.i386.rpm

rpm -ivh --ignoreos --force tinyos-1.1.15Dec2005cvs-1.cygwin.noarch.rpm

4- Aller à **opt/tinyos-1.x/tools/java/net/tinyos/sim** et vérifier si ces fichiers sont présents:

SimObjectGenerator.java et **MoteSimObjectGenerator.java**

S'ils existent, alors les supprimer de ce répertoire.

5- Editer le **makefile** qui est dans **C:\tinyos\cygwin\opt\tinyos-1.x\tools\java\net\tinyos\sim** et écrire cette instruction :

net/tinyos/message/avrmote/*.class

(Voir ** makefile pour vérifier là où il faut insérer cette instruction)

6- Aller à **shell** et taper:

cd /opt/tinyos-1.x/tools/java/net/tinyos/sim

make clean

make