REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université Mouloud Mammeri de Tizi-Ouzou

Faculté de Génie Electrique et d'Informatique

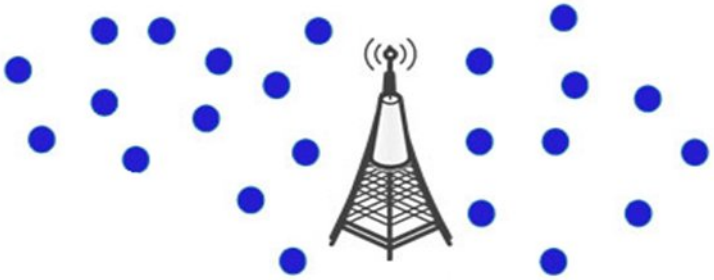
Département d'Informatique

Mémoire de fin d'études en vue de l'obtention

Du diplôme de Master 2

OPTION : Réseaux, mobilité et systèmes embarqués.

Thème : *Stratégie de tolérance aux pannes dans les RCSF*



Elaboré par *:*
M^me^ BEKKAR
Née DJABER Dehbia

Encadré par :
Mme AOUDJIT.R

Promotion 2014/2015

Dédicaces

Je remercie Dieu de m'avoir donné le courage pour accomplir ce modeste travail que je dédie :
A mon très cher mari Brahim qui m'a soutenue et encouragée durant toute cette année d'études.
A mon ange fille Maroua
A mes très chers parents
A mes sœurs et frères
A mes nièces et neveux
A ma belle famille, à tous mes amis et à tout le personnel de la daïra de Ouaguenoun
Et à tous ceux que j'aime et qui m'aiment

Remerciements

Je remercie Mme Aoudjit Rachida d'avoir accepter de m'encadrer et de m'orienter.
Je tiens aussi à remercier Mr Daoui.M, Mme Belkadi.M pour leur soutien et encouragement durant toute cette année d'étude
Sans oublier mon amie Malika qui m'a beaucoup aidée et soutenue

Table des matières

CHAPITRE 2 : PROTOCOLES DE ROUTAGE

CHAPITRE 3 : PROTOCOLES DE ROUTAGE TOLERANT AUX PANNES

CHAPITRE 4 : EVALUATION DE LEACH ET PROPOSITION DE T-LEACH

Introduction générale

Depuis leur création, les réseaux de communication sans fil ont connu un succès sans cesse croissant au sein des communautés scientifiques et industrielles. Grâce à ses divers avantages, cette technologie a pu s'instaurer comme acteur incontournable dans les architectures réseaux actuelles. Le média hertzien offre en effet des propriétés uniques, qui peuvent être résumées en trois points : la facilité du déploiement, l'ubiquité de l'information et le coût réduit d'installation. Au cours de son évolution, le paradigme sans fil a vu naître diverses architectures dérivées, telles que : les réseaux cellulaires, les réseaux locaux sans fils et autres. Durant cette dernière décennie, une architecture nouvelle a vu le jour : les réseaux de capteurs sans fil. Ce type de réseaux résulte d'une fusion de deux pôes de l'informatique moderne : les systèmes embarqués et les communications sans fil. Un réseau de capteurs sans fil (RCSF), ou "Wireless Sensor Network" (WSN), est composé d'un ensemble d'unités de traitements embarquées, appelées "motes", communiquant via des liens sans fil. Le but général d'un WSN est la collecte d'un ensemble de paramètres de l'environnement entourant les motes, telles que la température ou la pression de l'atmosphère, afin de les acheminer vers des points de traitement.

Les RCSF sont souvent considérés comme étant les successeurs des réseaux ad hoc. Ils ont su attirer un nombre croissant d'industriels, vu leur réalisme et leur apport concret. En effet, le besoin d'un suivi continu d'un environnement donné est assez courant dans diverses activités de la société. Les processus industriels, les applications militaires de tracking, le monitoring d'habitat, ainsi que l'agriculture de précision ne sont que quelques exemples d'une panoplie vaste et variée d'applications possibles du suivi continu offert par les RCSF.

Grâce à ce potentiel riche en applications, les RCSF ont su se démarquer et attirer de grandes firmes à travers le monde, telles que IBM, Sun, Intel et Philips. Malheureusement, les RCSF ne sont pas parfaits ! A cause de leur faible coût et leur déploiement dans des zones parfois hostiles, les motes sont assez

fragiles et vulnérables à diverses formes de défaillances : cassure, faible énergie, ... etc. Ces problèmes rendent les RCSF des systèmes à fragilité innée, qui doit être considérée comme une propriété normale du réseau.

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 du protocole 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.

Ce document s'articule autour de quatre chapitres. Le premier chapitre décrit les principes et les caractéristiques des réseaux de capteurs aussi que ses domaines d'application. Dans le deuxième chapitre, nous présentons les protocoles de routage des RCSF. Un troisième chapitre est consacré à la tolérance aux pannes, et on termine par le quatrième chapitre où on va étudier le protocole LEACH et sa version améliorée.

Introduction

Les progrès réalisés ces dernières années dans les domaines des techniques de communication sans fil ont permis de voir apparaître un nouveau type de réseau: les réseaux de capteurs sans fil (RCSF). Ces réseaux sont composés d'un ensemble de petits appareils, ou capteurs, possédant des ressources particulièrement limitées, mais qui leur permettent de collecter et transmettre des données environnementales (la température, l'humidité, la présence d'un gaz….etc.) vers un ou plusieurs points de collecte **[1]**.

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.1 Histoire des réseaux de capteurs

Dans les années 1990, dans le monde de la recherche, est apparue une idée qui paraissait plutôt un rêve pour cette époque : imaginer un système nerveux central pour la Terre, capable de surveiller en temps réel les événements, ayant comme principaux bénéfices de pouvoir empêcher les accidents et d'économiser l'énergie. (Cette poussière intelligente a mis longtemps à apparaître) dit le professeur Pister, de l'Université de Californie à Berkeley. (J'ai inventé l'expression il y a 14 ans. La poussière vraiment futée a mis le temps, mais elle est finalement arrivée).

Aujourd'hui les réseaux de capteurs sont devenus des systèmes pouvant atteindre un très grand nombre de nœuds, avec une zone de couverture déterminée et déployés d'une manière plus ou moins dense dans un environnement hétérogène dont on mesure ainsi son état global. Les derniers progrès en terme de miniaturisation, ainsi que le remplacement du câblage classique par des technologies de communication radio, ont généré de nouvelles

catégories d'applications qui visent de nombreux domaines : l'aéronautique, l'automobile, le médical, l'environnement, etc. De plus, les progrès des communications sans fil permettent aujourd'hui de répondre à des exigences peu envisageables auparavant.

I.2. Présentation des réseaux de capteurs

I.2.1 Définition d'un capteur

Les capteurs sont des dispositifs électroniques de taille extrêmement réduite avec des ressources très limitées, autonomes, capable de mesurer une valeur physique environnementale (température, lumière, pression, etc.) et de la communiquer à un centre de contrôle via une station de base **[2].**

I.2.2 Composants d'un capteur sans fil [3].

Un capteur sans fil est doté, principalement d'une unité de : captage, traitement, communication, stockage et énergie. D'autres modules peuvent être ajoutés selon le domaine d'application comme une unité de localisation, afin d'identifier la position géographique d'un capteur tel qu'un GPS (Global Position System), un mobilisateur pour que les capteurs puissent se déplacer et un générateur de puissance tel que des cellules solaires afin d'alimenter électriquement le capteur sans avoir à changer ses batteries .Ces éléments principaux et optionnels sont visibles sur la figure I.1.

- **Unité de captage** : elle est constituée de deux composants, un dispositif qui intercepte les données du monde physique et les transforme en signaux analogiques, et un convertisseur analogique/numérique qui transforme ces signaux analogiques en un signal numérique compréhensible par l'unité de traitement.

- **Unité de traitement** : composée d'un processeur et d'une mémoire intégrant un système d'exploitation spécifique (TinyOS , par exemple).

Cette unité possède deux interfaces, une interface pour l'unité d'acquisition et une interface pour l'unité de communication. Elle acquiert les informations en provenance de l'unité d'acquisition et les envoie à l'unité de communication. Cette unité est chargée aussi d'exécuter les protocoles de communications qui permettent de faire collaborer le capteur avec d'autres capteurs. Elle peut aussi analyser les données captées.

- **Unité de communicatio*n*** : elle est responsable des émissions et réceptions des données sur un medium sans fil. Elle se base sur les technologies sans fil à faible portée de communication, Zigbee, Bluetooth ou Wifi **.**

- **Unité d'alimentation énergétique** : un capteur est muni d'une batterie pour alimenter tous ses composants Cependant, à cause de sa taille réduite, la batterie dont il dispose est limitée et généralement irremplaçable. Pour cela, l'énergie est la ressource la plus précieuse puisqu'elle influe directement sur la durée de vie des capteurs, ce qui a rendu l'énergie Comme principale contrainte pour un capteur.

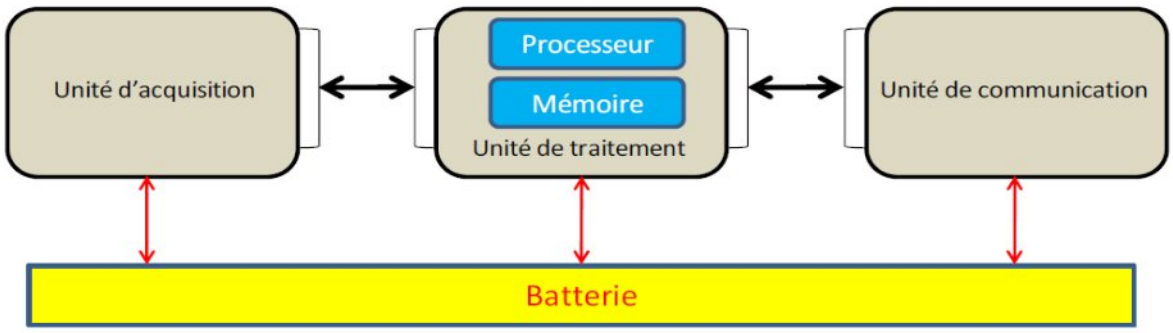


Figure I.1 Composants d'un capteur sans fil

I.2.3 Classification des capteurs [4]

Les capteurs ont plusieurs modes de classification :

I.2.3.1 Apport énergétique

a) Capteurs passifs :

Ils n'ont pas besoin d'apport d'énergie extérieure pour fonctionner (exemple : 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.2.3.2 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.

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

Quelques capteurs numériques typiques : les capteurs incrémentaux, les codeurs absolus.

I.3. Définition d'un RCSF ou WSN (Wireless Sensor Network)

Un réseau de capteurs sans fil (RCSF) est un type particulier des réseaux ad hoc **[4]**. Il est composé de centaines ou de milliers d'éléments nommés noeuds ou capteurs placés de manière plus au moins aléatoire. Dans ces réseaux, chaque noeud est capable de surveiller son environnement et de réagir en cas de besoin en envoyant l'information collectée à un ou plusieurs points de collecte, à l'aide d'une connexion sans fil **[5]**.

I.3.1 Architecture

Un réseau de capteurs est constitué essentiellement de : plusieurs noeuds capteurs, un noeud Sink et un centre de traitement des données **.**

- **Noeuds** : Sont des capteurs, leur type, leur architecture et leur disposition géographique dépendent de l'exigence de l'application en question. Leur énergie est souvent limitée puisqu'ils sont alimentés par des piles
- **Sink** : c'est un noeud particulier du réseau. Il est chargé de la collecte des données issues des différents noeuds du réseau. Il doit être toujours actif puisque l'arrivée des informations est aléatoire. C'est pourquoi son énergie doit être illimitée. Dans un réseau de capteur sans fils plus ou moins large et à charge un peu élevée, on peut trouver deux sinks ou plus pour alléger la charge.
- **Centre de traitement des données** : c'est le centre vers lequel les données collectées par le sink sont envoyées. Ce centre a le rôle de regrouper les données issues des noeuds et les traiter de façon à en extraire de l'information utile exploitable. Le centre de traitement peut être éloigné du sink, alors les données doivent être transférées à travers un autre réseau, c'est pourquoi on introduit une passerelle entre le sink et le réseau de transfert pour adapter le type de données au type du canal (comme c'est illustré dans la figure I.2

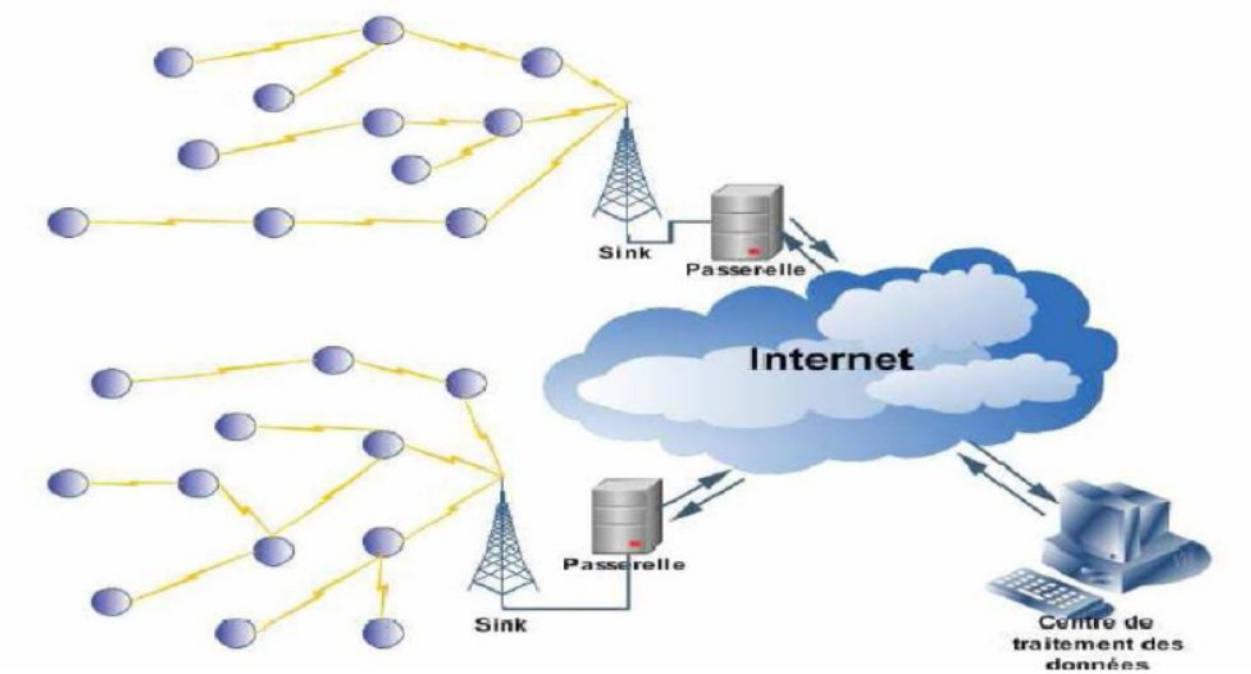


Figure I. 2 : Architecture générale d'un réseau de capteurs sans fil [3].

- **Types des nœuds**

Dans un réseau de capteurs il existe deux types de nœuds : nœud source et nœud sink.

Un nœud source est n'importe quelle entité dans le réseau qui peut fournir de l'information, c'est à dire un simple nœud capteur.

Un nœud sink est l'entité où les données sont récupérées. Il y a essentiellement trois types de sink **[3]** : comme le montre la figI.2

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

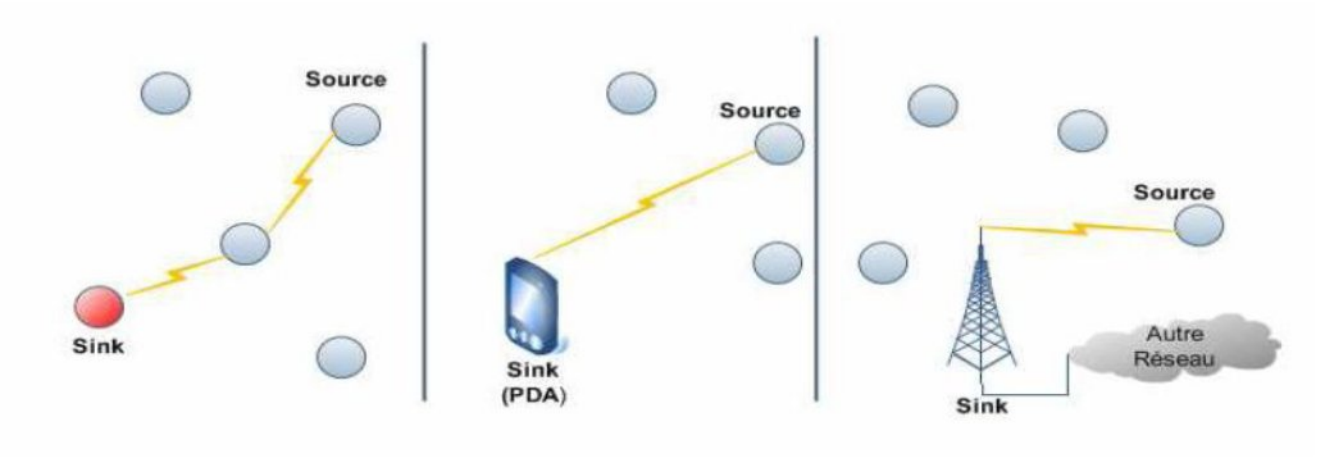


Figure I. 3 : Différents types de sink [3].

I.4. Architecture de communication dans les réseaux de capteurs

Le modèle de communication comprend cinq couches qui ont les mêmes fonctions que celles du modèle OSI ainsi que trois couches pour la gestion d'énergie, la gestion de la mobilité et la gestion des tâches.

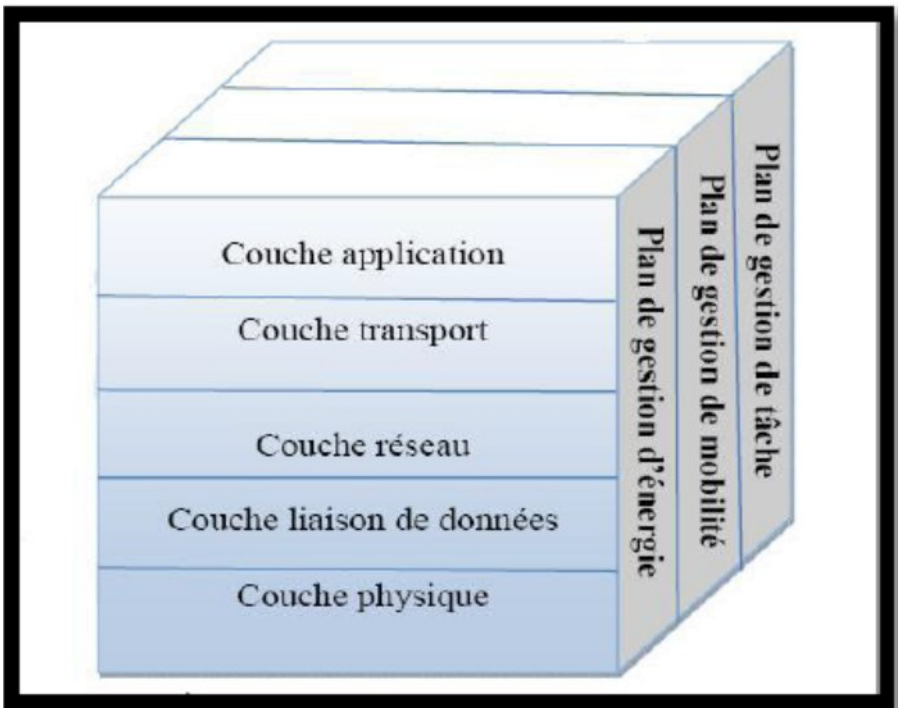


Figure I. 4: Modèle en couches du réseau de capteurs sans fil [3].

Rôles des couches :

- ✓ **Couche physique :** Matériels pour envoyer et recevoir les données.
- ✓ **Couche liaison de données :** Gestion des liaisons entre les noeuds et les stations de base, contrôle d'erreurs.
- ✓ **Couche réseau :** Routage et transmission des données.
- ✓ **Couche transport :** Transport des données, contrôle de flux.
- ✓ **Couche application :** Interface pour les applications au haut niveau.
- ✓ **Plan de gestion d'énergie :** Contrôle l'utilisation d'énergie.
- ✓ **Plan de gestion de mobilité :** Gestion des mouvements des nœuds.
- ✓ **Plan de gestion de tâche :** Balance les tâches entre les nœuds afin d'économiser de l'énergie.

I.5. Types de réseaux de capteurs sans fil [8].

I.5.1. Réseaux de poursuite

Ces réseaux sont généralement développés par l'armée, ils peuvent servir à surveiller toutes les activités d'une zone stratégique ou d'accès difficile, ainsi on pourra détecter des agents chimiques, biologiques ou des radiations avant des troupes. On peut aussi penser à des capteurs embarqués sur les soldats pour faciliter leur guidage et le contrôle de leur position depuis la base.

I.5.2. Réseaux de collection des données d'environnement

Les nœuds de ce type de réseau peuvent avoir plusieurs fonctionnalités et différents types de capteurs. Ce type de réseau nécessite généralement un flux de données faible, une durée de vie importante ; il sert à la collecte périodique des données environnementales puis leur transmission vers la station de base.

I.5.3. Réseaux de surveillance et sécurité

La différence entre ce réseau et le réseau de collection des données d'environnement est que les noeuds ne transmettent pas l'ensemble des

données collectées mais seulement les rapports concernant une violation de la sécurité. Ce sont en général des noeuds fixes qui contrôlent d'une façon continue la détection d'une anomalie dans le fonctionnement d'un système. Ainsi les altérations dans la structure d'un bâtiment, suite à un séisme, pourraient être détectées par des capteurs intégrés dans les murs ou dans le béton, sans alimentation électrique ou autres connexions filaires.

I.6. Topologies des réseaux de capteurs

Un réseau de capteurs sans fil est composé d'un ensemble de noeuds capteurs et des Gateway qui s'occupent de collecter les données des capteurs et de les transmettre à l'utilisateur via l'internet ou le satellite, il existe plusieurs topologies pour les réseaux de capteurs.

I.6.1. Topologie en étoile

La topologie en étoile est un système uni-saut. Tous les noeuds envoient et reçoivent seulement des données avec la station de base. Cette topologie est simple et elle demande une faible consommation d'énergie, mais la station de base est vulnérable et la distance entre les noeuds et la station est limitée.

Avantage : simplicité et faible consommation d'énergie des noeuds, moindre latence de communication entre les noeuds et la station de base.

Inconvénient : la station de base est vulnérable, car tout le réseau est géré par un seul noeud.

I.6.2 Topologie en toile (Mesh Network)

La topologie en toile est un système multi-saut. La communication entre les noeuds et la station de base est possible. Chaque noeud a plusieurs chemins pour envoyer les données. Cette topologie a plus de possibilités de passer à l'échelle du réseau, avec redondance et tolérance aux fautes, mais elle demande une consommation d'énergie plus importante.

Avantage : Possibilité de passer à l'échelle du réseau, avec redondance et tolérance aux fautes.

Inconvénient : Une consommation d'énergie plus importante est induite par la communication multi-sauts. Une latence est crée par le passage des messages des noeuds par plusieurs autres avant d'arriver à la station de base.

I.6.3 Topologie hybride

La topologie hybride est un mélange des deux topologies ci-dessus. Les stations de base forment une topologie en toile et les noeuds autour d'elles sont en topologie étoile. Elle assure la minimisation d'énergie dans les réseaux de capteurs.

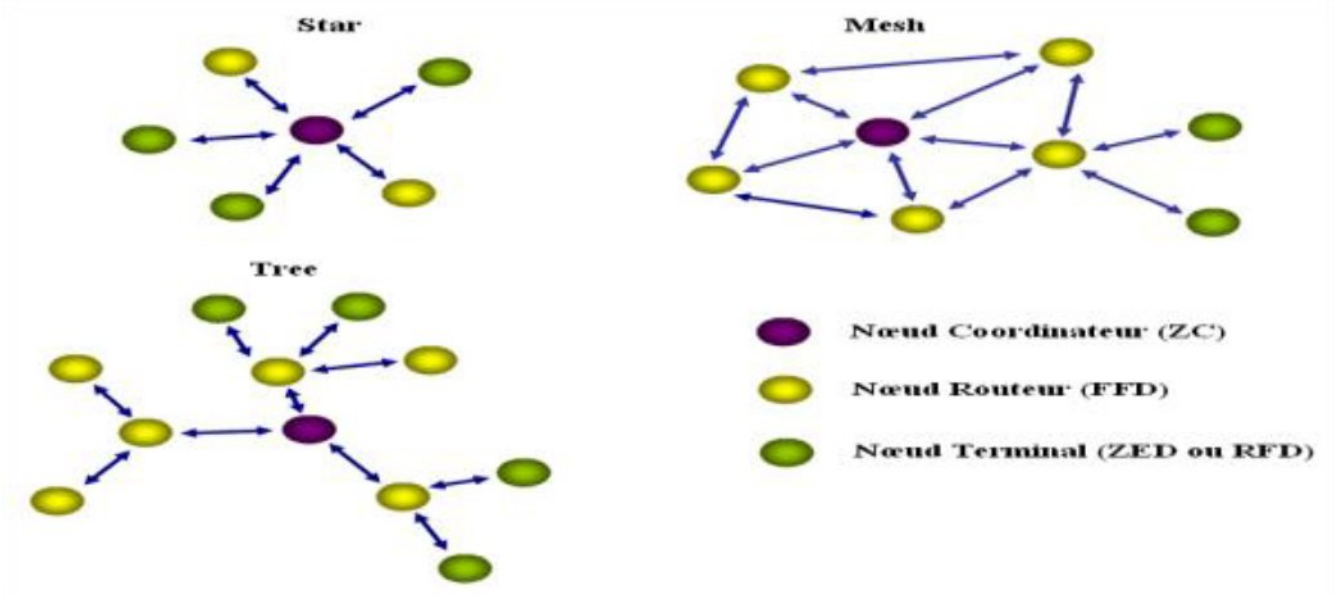


Figure I. 5 : Topologies des réseaux de capteurs

I.7 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.6.

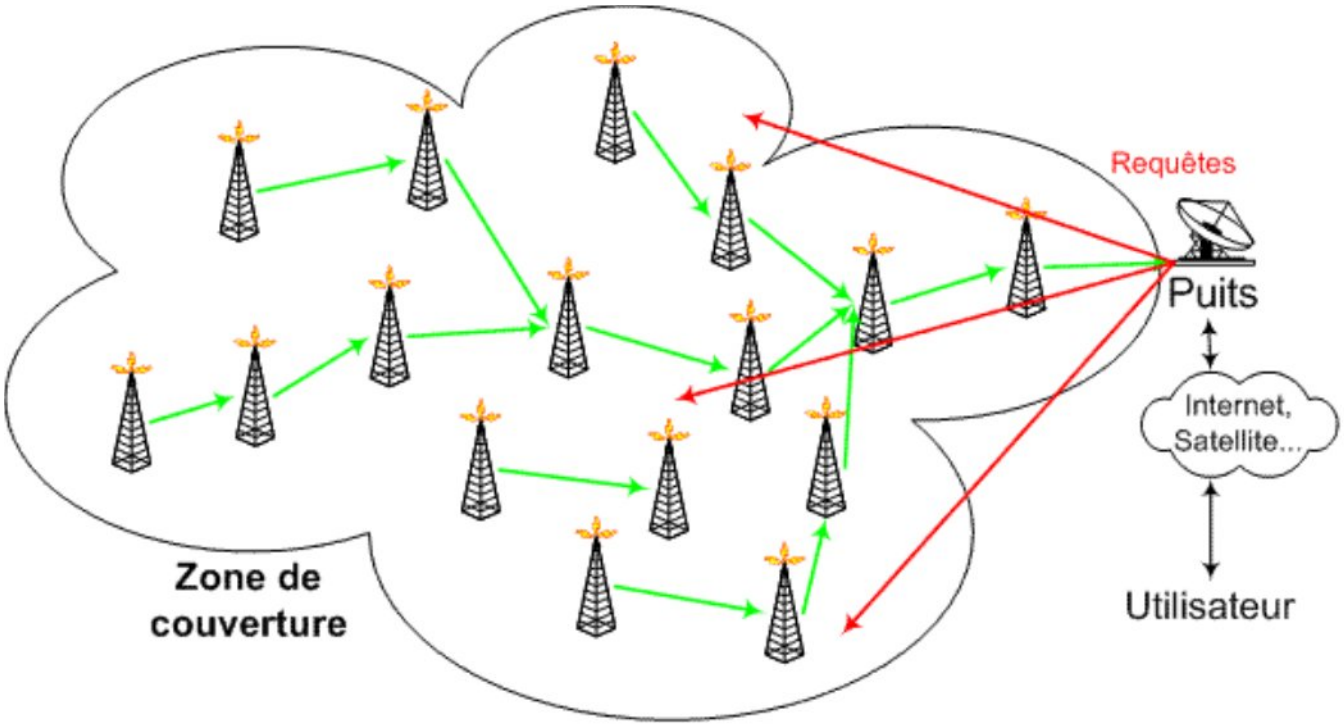


Figure I.6 : Collection des informations à la demande.

- **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.7.

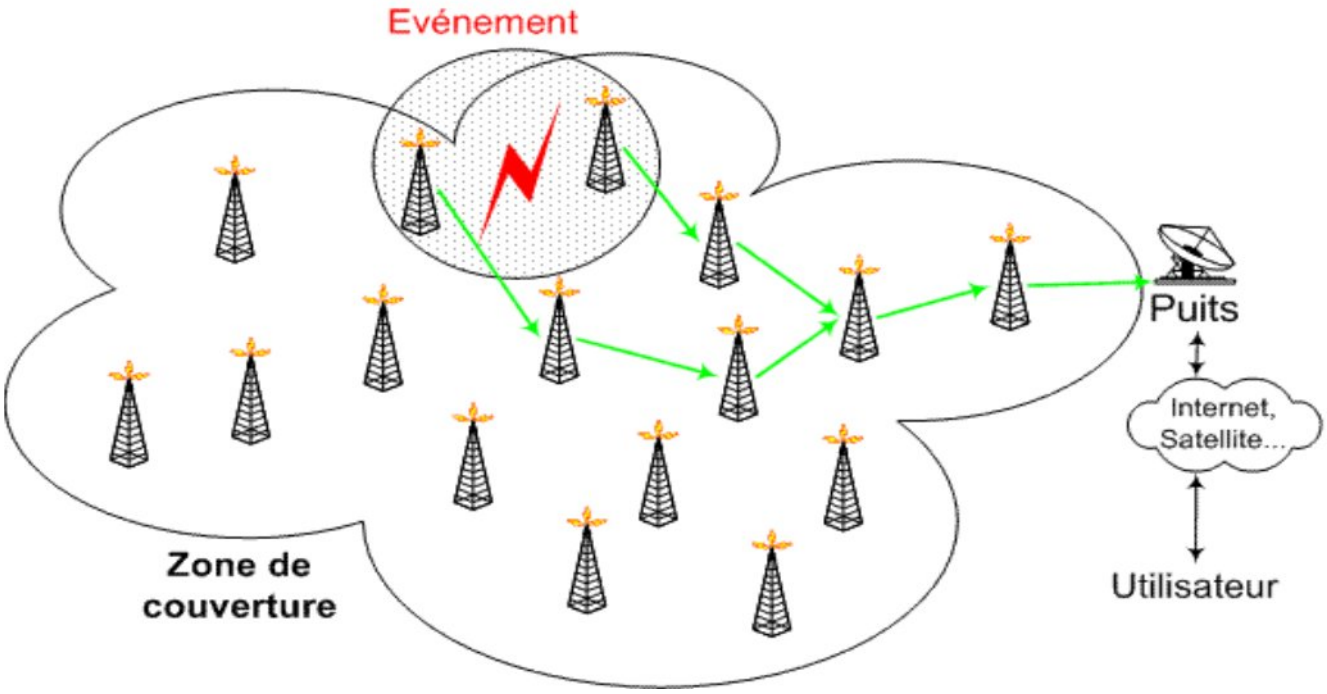


Figure I.7 : Collection des informations suite à un événement.

I.8 Caractéristiques des réseaux de capteurs [6].

Parmi les caractéristiques les plus importantes d'un réseau de capteurs, nous citons **:**

- ✓ **La durée de vie limitée** *:*

Les nœuds capteurs sont très limités par la contrainte d'énergie, ils fonctionnent habituellement sans surveillance dans des régions géographiques éloignées. Par conséquent recharger ou remplacer leurs batteries devient quasiment impossible.

- ✓ **Ressources limitées** :

Habituellement les nœuds capteurs ont une taille très petite, ce facteur de forme limite la quantité de ressources qui peuvent être mises dans ces

nœuds. En conséquence, la capacité de traitement et de mémoire est très limitée.

- ✓ **Topologie dynamique** :

La topologie des réseaux de capteurs change d'une manière fréquente et rapide car: les nœuds capteurs peuvent être déployés dans des environnements hostiles (par exemple un champ de bataille), la défaillance d'un nœud capteur peut donc être très probable. De plus, les nœuds capteurs et les nœuds finaux où ils doivent envoyer l'information capturée peuvent être mobiles.

- ✓ **Agrégation des données** :

Dans les réseaux de capteurs, les données produites par les nœuds capteurs sont très reliées, ce qui implique l'existence de redondances de données. Une approche répandue consiste à agréger les données au niveau des nœuds intermédiaires afin de réduire la consommation d'énergie lors de la transmission de ces données.

- ✓ **La scalabilité** :

Les réseaux de capteurs engendrent un très grand nombre de capteurs, ils peuvent atteindre des milliers voir des millions de capteurs. Le défi à relever par les RCFSs est d'être capable de maintenir leurs performances avec ce grand nombre de capteurs.

- ✓ **Bande passante limitée :**

En raison de la puissance limitée, les nœuds capteurs ne peuvent pas supporter des débits élevés.

- ✓ **Sécurité physique limitée**:

Cela se justifie par les contraintes et limitations physiques qui minimisent le contrôle des données transmises.

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

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. 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 celle ci, 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 Domaines d'application des réseaux de capteurs [1,7].

La diminution de taille et de coût des micro-capteurs, l'élargissement de la gamme des types de capteurs disponibles (thermique, optique, vibrations,...) et l'évolution des supports de communication sans fil, ont élargi le champ d'application des réseaux de capteurs. Ils s'insèrent notamment dans d'autres systèmes tels que le contrôle et l'automatisation des chaînes de montage.

Ils permettent de collecter et de traiter des informations complexes provenant de l'environnement (météorologie, étude des courants, de l'acidification des océans, de la dispersion de polluants, etc).

- ✓ **Applications militaires :** On peut penser à un réseau de capteurs déployé sur un endroit stratégique ou d'accès difficile, 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).
- ✓ **Applications domestiques :** En plaçant, sur le plafond ou dans le mur, des capteurs, on peut économiser l'énergie en gérant l'éclairage ou le chauffage en fonction de la localisation des personnes.
- ✓ **Applications environnementales :** Les réseaux de capteurs sont beaucoup appliqués dans ce domaine pour détecter des incendies, surveiller des catastrophes naturelles, détecter des pollutions et suivre des écosystèmes.
- ✓ **Applications agricoles :** 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.
- ✓ **Applications médicales :** Les réseaux de capteurs ont aussi des développements dans le domaine de diagnostic médical. Par exemple, des micro-caméras 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.

- ✓ **Applications transportés :** Il est possible d'intégrer des noeuds 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.

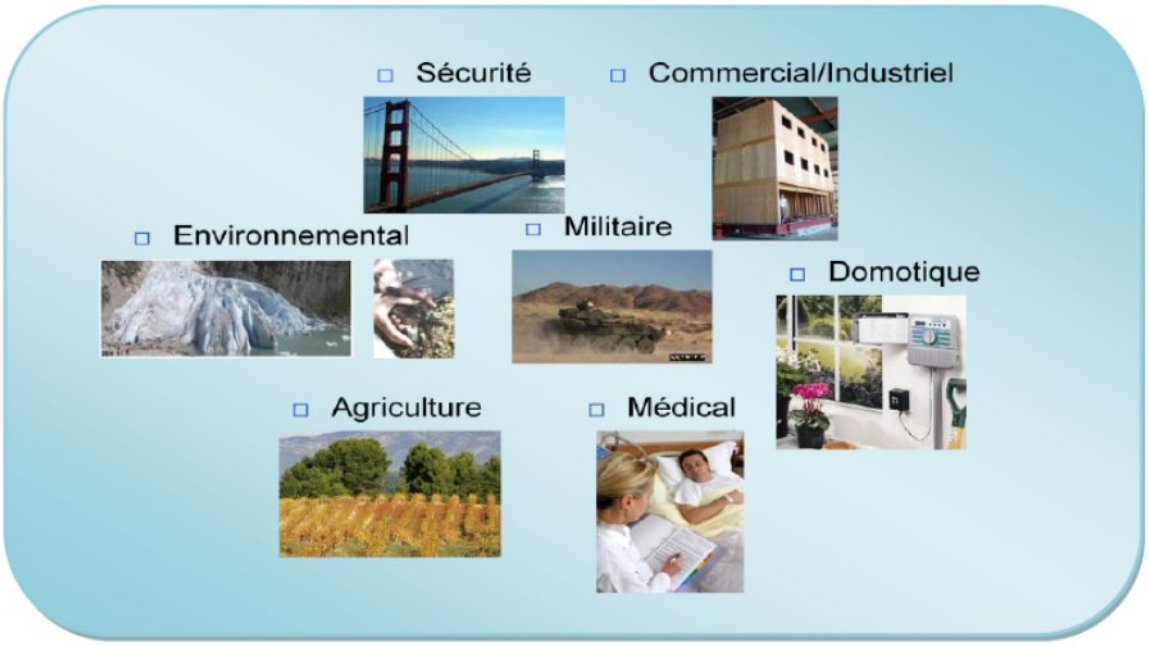


Figure I.8 : Applications des réseaux de capteurs

Conclusion

Dans ce chapitre, nous avons présenté les réseaux de capteurs, en parlant sur l'architecture, composants, fonctionnement, topologies utilisés, applications, ainsi que les caractéristiques. Nous avons montré l'importance des réseaux de capteurs sans fil, qui sont en plein développement et deviennent de plus en plus répandus.

Actuellement, ils constituent un thème de recherche très dynamique, tiré vers le haut, par leurs utilisations dans divers domaines. En effet, leurs applications sont de plus en plus nombreuses et diversifiées. Cependant la réalisation de ces

applications pose de grands défis auxquels il faut répondre ; le routage de ces réseaux est l'un des défis les plus importants à considérer.

Dans le chapitre suivant, nous introduirons en détail le routage dans les réseaux du capteur sans fil.

Introduction :

Dans les RCSF, les capteurs sont déployés en grand nombre pour surveiller un tel phénomène et faire remontrer l'information à un centre de contrôle distant. Pour atteindre cette finalité, les capteurs ont la capacité de communiquer et collaborer entre eux pour acheminer l'information collectée à la station de base en garantissant sa fiabilité et en empruntant le plus court chemin entre le nœud qui a détecté ce phénomène et la station de base. Cette opération permet le routage de l'information entre le nœud détecteur et le nœud puits et elle consiste à trouver les routes les plus courtes. Dans cette optique, plusieurs protocoles de routage ont été proposés dans la littérature [2].

Les contraintes présentées dans les RCSF ont données naissance à des protocoles de routage différents que ceux des autres réseaux sans fil puisque la contrainte énergétique se pose avec force dans les RCSF. De ce fait, les protocoles de routage conçus pour les RCSF doivent garantir l'acheminement de l'information entre tout nœud du réseau et la station de base à moindre coût en termes d'énergie.

Dans ce que suit, nous présentons quelques approches et techniques sur lesquelles se basent les protocoles de routage dans les réseaux de capteurs.

II.1 Définition d'un protocole de routage : [11]

Le routage est le mécanisme par lequel des chemins sont sélectionnés dans un réseau pour acheminer les données d'un expéditeur jusqu'à un ou plusieurs destinataires. Le routage est une tâche exécutée dans de nombreux réseaux, tels que les réseaux de capteurs. Il 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.

II. 2 Classification des protocoles de routage dans les RCSF [12]

Récemment, les protocoles de routage pour les RCSF ont été largement étudiés, Comme l'illustre la figure II.1, ils peuvent être classés selon plusieurs critères :

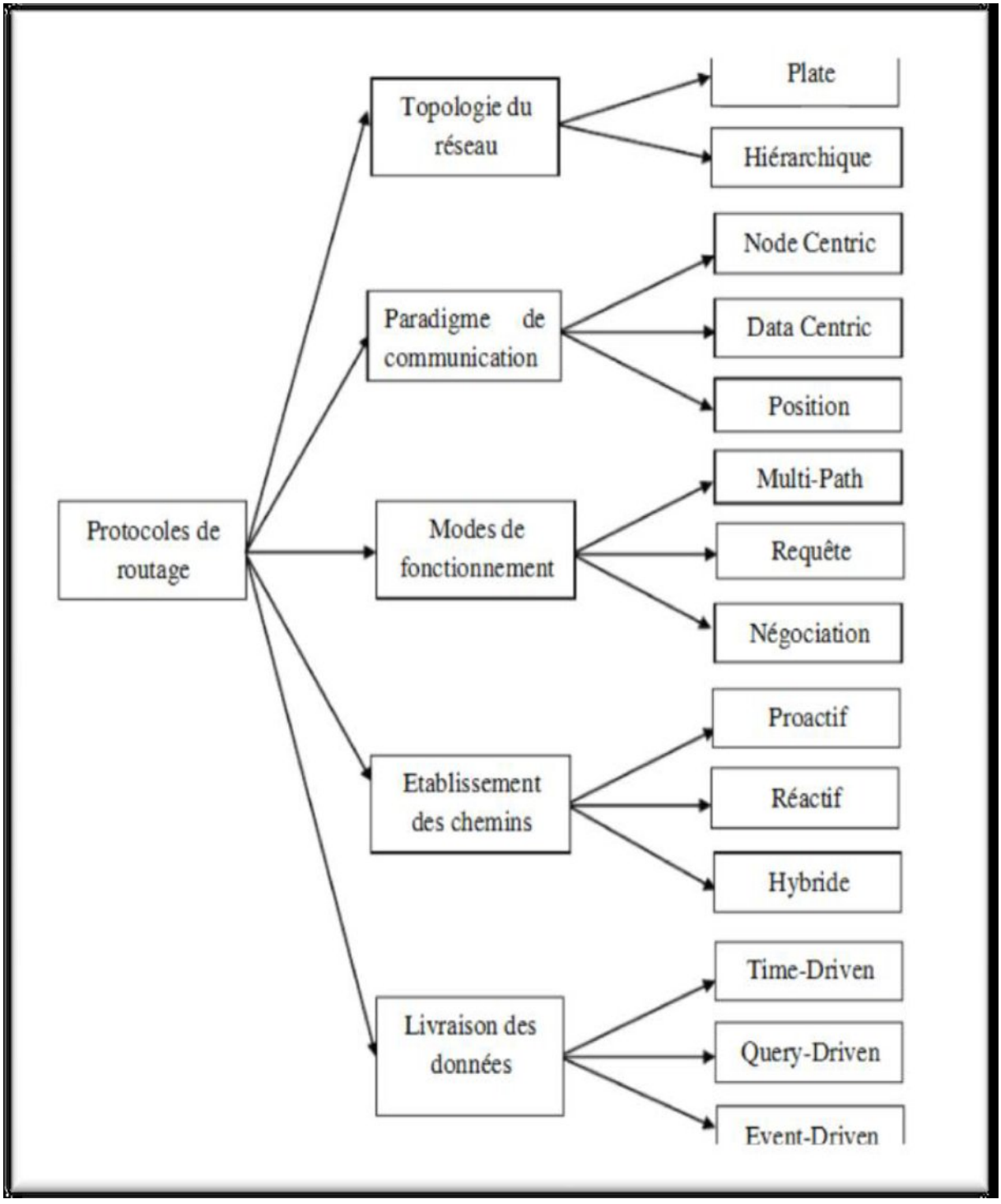


Figure II.1 : Classification des protocoles de routage dans les RCSF.

II.2.1 Selon la topologie du réseau

La topologie détermine l'organisation des capteurs dans le réseau. Globalement, il existe deux topologies dans les RCSF: La topologie plate et la topologie hiérarchique.

a. Topologie Plate :

Un réseau de capteurs sans fil plat est un réseau homogène, où tous les noeuds sont identiques en termes de batterie et des fonctions, excepté le « Sink ». Dans ce type de topologie, les capteurs communiquent entre eux afin d'acheminer l'information au nœud centralisé (station de base). Ce processus d'acheminement d'information peut prendre deux formes : communiquer directement avec la station de base Figure II.2 (a), ou via un mode multi-sauts Figure II.2 (b)). De plus dans ce type de topologie Figure (a) tous les nœuds peuvent envoyer leurs données à la station de base en utilisant une forte puissance, ceci peut conduire à la diminution de la durée de vie du réseau.

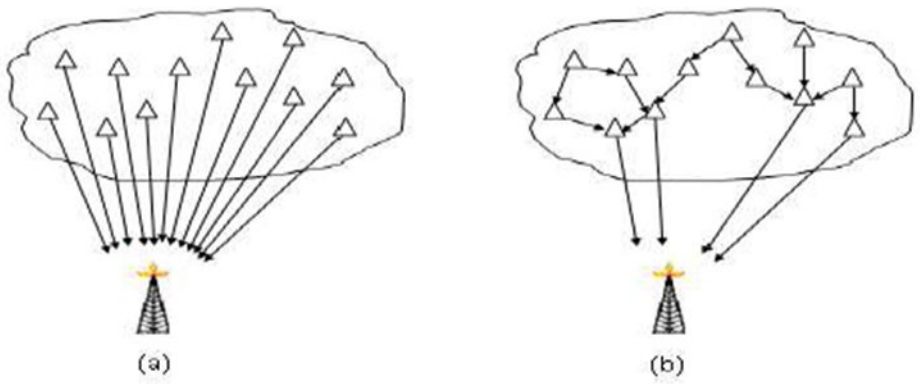


Figure II.2 : Architecture de communication dans une topologie plate [13]

b. Topologie hiérarchique :

Dans cette architecture, le réseau est constitué d'un ensemble de groupe de capteurs (clusters), tel qu'il est illustré dans la Figure II.3. Le nœud représentant le cluster, appelé **cluster-head**, a la responsabilité de transmettre les données à

la station de base. L'avantage majeur de ce type d'architecture est le prolongement de la durée de vie du réseau de capteurs. Ce résultat est achevé en désignant le cluster-head comme étant le nœud responsable de la transmission des informations (agrégées). Ce procédé est meilleur de celui où tous les nœuds envoient leurs données à un emplacement distant.

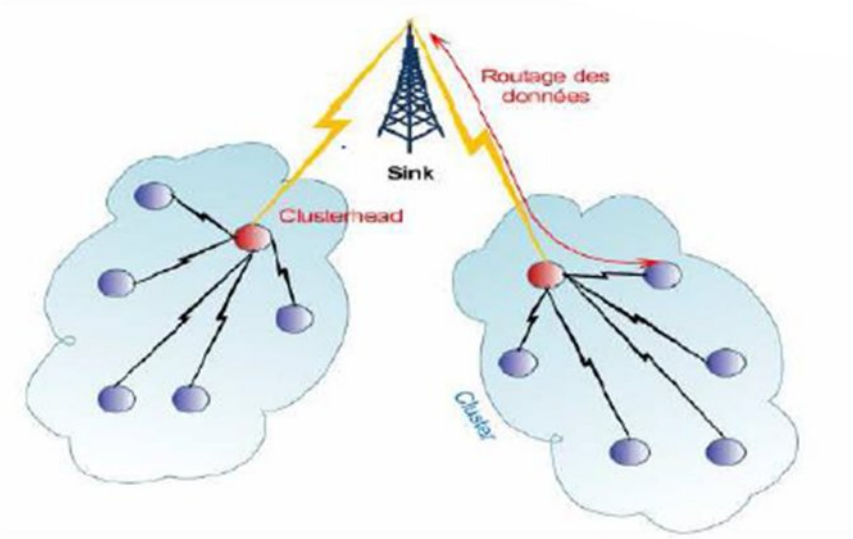


Figure II.3 : Topologie à base de cluster [13]

II.2.2 Selon la méthode d'établissement de routes

Suivant la manière de création et de maintien des chemins pendant le routage nous distinguons trois catégories de protocoles de routages : protocoles proactifs, réactifs ou hybrides.

a. Protocole proactif :

Ces protocoles de routage essaient de maintenir les meilleurs chemins existants vers toutes les destinations possibles au niveau de chaque nœud du réseau. Les routes sont sauvegardées mêmes si elles ne sont pas utilisées. Chaque nœud du réseau maintient une table de routage pour toutes les destinations indépendamment de l'utilité des routes. Les protocoles proactifs sont adaptés aux applications qui nécessitent un prélèvement périodique des

données. Et par conséquent, les capteurs peuvent se mettre en veille pendant les périodes d'inactivité, et n'enclencher leur dispositif de capture qu'à des instants particuliers.

b. Protocoles réactifs :

Ces protocoles (dits aussi, les protocoles de routage à la demande) 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.

c. Protocoles hybrides :

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

II.2.3 Selon les paradigmes de communication

Le paradigme de communication détermine la manière dont les nœuds sont interrogés. Dans le RCSF, il existe trois paradigmes de communication :

a. Centré-nœuds :

Ce paradigme est celui employé dans les réseaux conventionnels, où il est nécessaire de connaitre et d'identifier les nœuds communicants (comme l'adresse IP). Les réseaux ad hoc utilisent ce genre de paradigme, qui s'intègre bien avec l'utilisation de ce type d'environnement. Cependant pour les réseaux de capteurs, un routage basé sur une identification individuelle des nœuds ne reflète pas l'usage réel du réseau. Pour cela, un autre paradigme a été introduit :

Centré-données. Néanmoins, le paradigme Centré-nœuds n'est pas à écarter totalement, car certaines applications nécessitent une interrogation individuelle des capteurs.

b. Centré-données :

Dans les RCSF, la donnée est généralement plus importante que le nœud lui-même. De ce fait, le routage et l'identification, dans ce paradigme, se font en fonction des données disponibles au niveau des capteurs. Ainsi le système peut être vu comme une base de données distribuée, où les nœuds forment des tables virtuelles, alimentées par les données captées.

Le protocole Directed Diffusion (DD) est un exemple des protocoles de routage Centré-données.

c. Basé-localisation :

Dans cette approche, les positions des nœuds représentent le moyen principal d'adressage et de routage. Dans ce cas, le routage s'effectue grâce à des techniques géométriques afin d'acheminer l'information d'une zone géographique vers une autre.

Ce type de mécanismes nécessite le déploiement d'une solution de positionnement, dont le degré de précision requis dépend de l'application ciblée.

II.2.4 Selon le mode de fonctionnement du protocole

Le mode de fonctionnement définit la manière avec laquelle les données sont propagées dans le réseau. Selon ce critère, les protocoles de routage peuvent êtres classifiées quatre catégories : routage basé sur la qualité de service "QoS" (Quality of Service), routage base sur les requêtes (Query-Based Routing), routage multi-chemins (Multi-Path Routing), et routage basé sur la négociation (Negociation Based Routing).

a. Routage basé sur les multi-chemins :

Dans cette catégorie, les protocoles de routage utilisent des chemins multiples plutôt qu'un chemin simple afin d'augmenter la performance du réseau. La fiabilité d'un protocole peut être mesurée par sa capacité à trouver des chemins alternatifs entre la source et la destination en cas de défaillance du chemin primaire. Pour cette raison certains protocoles construisent plusieurs chemins indépendants, c.-à-d. : ils ne partagent qu'un nombre réduit (voir nul) de noeuds. Malgré leur grande tolérance aux pannes, ces protocoles requièrent plus de ressources énergétiques et plus de message de contrôle.

b. Routage basé sur les requêtes :

Dans ce type de routage, le puits génère des requêtes afin d'interroger les capteurs. Ces requêtes sont exprimées soit par un schéma valeur-attribut ou bien en utilisant un langage spécifique (par exemple SQL : Structured Query Language). Les noeuds qui détiennent les données requises doivent les envoyer au nœud demandeur à travers le chemin inverse de la requête. Les requêtes émises par le puits peuvent aussi être ciblées sur des régions spécifiques de réseau.

c. Routage basé sur la négociation :

En détectant le même phénomène, les nœuds capteurs inondent le réseau par les mêmes paquets de données. Ce problème de redondance peut être résolu en employant des protocoles de routage basés sur la négociation. En effet, avant de transmettre, les nœuds capteurs négocient entre eux leur données en échangeant des paquets de signalisation spéciales, appelés métadonnées. Ces paquets permettent de vérifier si les nœuds voisins disposent déjà de la donnée à transmettre. Cette procédure garantit que seules les informations utiles seront transmises et élimine la redondance des données.

d. Routage basé sur la qualité de service :

Dans les protocoles de routage basé sur la QoS, le réseau doit équilibrer entre la consommation d'énergie et la qualité de données. En particulier, le réseau doit satisfaire certaines métriques de QoS, par exemple, retard, énergie, largeur de bande passante, etc. Les protocoles de cette approche sont très recommandés pour les applications de surveillance (centrales nucléaires, applications militaires, etc).

II.2.5 Selon le modèle de livraison de données

Il est possible de distinguer trois modèles de livraison de données : time-driven, query-driven et event-driven **.**

a. Time-driven :

Cette approche consiste à la livraison des données de façon périodique. Cet aspect permet aux capteurs de se mettre en veille pendant les périodes d'inactivité, et n'enclencher leur dispositif de capture qu'à des instants particuliers. Ainsi, la durée de vie du réseau va être allongée.

Le modèle time-driven est approprié pour des applications qui nécessitent un prélèvement périodique des données. Par exemple, cela est utile dans des applications de monitoring (feu, météo).

b. Query-driven :

Dans les applications query driven, la collecte d'informations sur l'état de l'environnement et la livraison des données sont initiées par des requêtes envoyées généralement par le nœud puits.

c. Event-driven :

Ce modèle est généralement adopté dans les applications temps-réel où les capteurs doivent réagir immédiatement à des changements soudains des valeurs captées. Dans ce cas, le protocole de routage doit être réactif et doit donner des réponses rapides à l'occurrence d'un certain nombre d'évènements.

II. 3 Facteurs de conception de protocoles de routage [9]

Plusieurs facteurs sont décisifs pour toute conception d'un protocole de routage pour les RCSF, nous allons les mentionner comme suit :

a. Tolérance aux pannes

La propriété de tolérance aux pannes est définie par l'aptitude du protocole de routage à maintenir ses fonctionnalités, en cas de panne de quelques nœuds. Le but de la tolérance aux pannes est d'éviter la faille totale du système malgré la présence de fautes dans un sous ensemble de ses composants élémentaires.

b. Consommation d'énergie

Le facteur le plus important à prendre en considération est l'énergie consommée par un capteur lors de la détection et de la transmission des données captées sur le réseau. La transmission est la fonction qui consomme le plus d'énergie, elle est proportionnelle au carré de la distance de transmission, et à la taille du paquet à envoyer. Pour préserver de l'énergie et augmenter la durée de vie d'un réseau, les chercheurs ont opté pour des techniques qui favorisent le traitement local des données afin de réduire la taille du paquet, ces techniques évitent la redondance des informations à transmettre et qui mettent le capteur en mode sommeil le plus longtemps possible.

c. Limitations de capacités des nœuds

Un capteur est très limite, en ce qui concerne, les traitements locaux à cause de sa taille minimale. Cela signifie que le protocole de routage doit être simple et peu exigeant en capacité de calcul et de stockage.

d. Scalabilité

Les applications des RCSF nécessitent en général un déploiement dense des nœuds. Les protocoles de routage doivent donc être très scalables. Autrement dit, les protocoles de routage ne devraient pas souffrir d'une dégradation de

performances dans le cas d'endommagement de nœuds aussi bien qu'avec un nombre plus élevé de nœuds.

e. Hétérogénéité

Généralement, les nœuds d'un RCSF sont homogènes, ayant les mêmes capacités de calcul, de mémoire et de ressources énergétiques. Ces nœuds pourront être rapidement épuisés puisqu'ils réalisent plusieurs tâches à la fois comme le captage, le traitement et le routage de données. Pour y remédier, une solution envisagée par certaines applications consiste à intégrer des nœuds spéciaux plus puissants que les autres et qui seront chargés d'effectuer les tâches les plus coûteuses en termes de ressources énergétiques. Cependant, l'intégration d'un ensemble de nœuds hétérogènes dans un seul réseau impose de nouvelles contraintes liées au routage de données. En effet, les données récoltées par ces nœuds peuvent être soumises à des fortes qualités de service, et peuvent suivre des modèles de transmission de données différents. Par conséquent, la conception des protocoles de routage doit prendre en compte les différents types de nœuds, et les contraintes qui en résultent.

f. Modèles de transmission de données

Les RCSFs se caractérisent par une communication particulière par rapport aux autres réseaux; ou les données transitent, souvent, entre des capteurs qui scrutent l'environnement qui les entourent et envoient l'information vers un ou plusieurs nœuds dits puits. Selon l'application implémentée au niveau du puits, nous distinguons trois types de communications principales :

1 .Continue:

La collecte de données se fait périodiquement d'une façon continue ou bien selon une certaine distribution (i.e. gaussienne, géométrique,…) déterministe ou probabiliste. Le processus d'envoi planifie les périodes du

sommeil des capteurs qui ne participent pas (exemple: les applications de la météo).

2. Dirigée par un événement (event-driven):

Dans ce type, les capteurs ne transmettent leurs données que si un événement prédéfini est observé comme un changement brusque. Ainsi, le délai de la transmission est limité et la réception doit être assurée (des envois multiples sont à prévoir) (exemple: les applications de la surveillance).

3. Dirigée par l'application (application-driven):

La transmission de données est initiée par la réception de la requête envoyée par le puits qui définit le type et les fréquences des envois. Dans ce cas, des mécanismes de correspondance sont nécessaires pour que les capteurs réussissent à déchiffrer les requêtes reçues (exemple: les applications du contrôle des systèmes automatises).

Conclusion

Le routage de données est considéré comme le domaine le plus exploré parmi les domaines de recherche sur les réseaux de capteur. Il représente aussi un problème complexe car nous devons assurer la fiabilité de livraison de données, la performance du système et tout cela en consommant moins d'énergie.

Les protocoles de routage pour les RCSF sont nombreux avec un unique objectif : Assurer la délivrance des paquets collectés par les nœuds capteurs tout en parvenant à étendre la durée de vie du réseau.

Cependant, les pannes sont inévitables dans ce type de réseaux. Ces pannes peuvent avoir des conséquences catastrophiques. Dans le chapitre suivant nous détaillons, le routage permettant la tolérance aux pannes dans les réseaux de capteurs.

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.

III.1 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 [20].

III.2 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 [21] comme s'est illustré dans la figure III-1

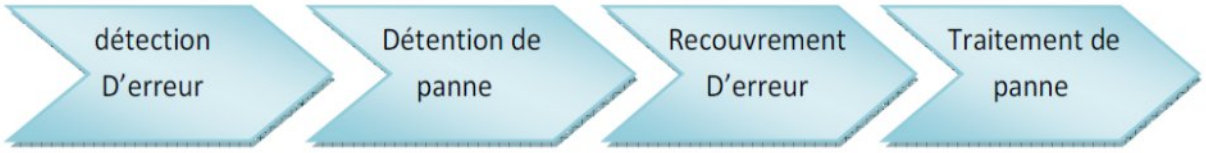


Figure III-1 : Procédure générale de tolérance aux pannes.

III.2.1 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.

III.2.2 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.

III.2.3 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.

III .2.4 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.

III.3 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.

III.3.1 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.

III.3.2 Classification architecturale

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

- ✓ **Gestion de la batterie**

  Cette catégorie est considérée comme une approche préventive, où les protocoles définissent une distribution uniforme pour la dissipation d'énergie entre les différents nœuds capteurs ; afin de mieux gérer la consommation d'énergie et augmenter ainsi la durée de vie de tout le réseau. En outre, le mécanisme de mise en veille est une technique de gestion de batterie. En effet, les protocoles déterminent des délais de mise en veille des nœuds capteurs inactifs pour une meilleure conservation d'énergie ;

- ✓ **Gestion de flux**

  Cette catégorie regroupe les techniques qui définissent des protocoles de gestion de transfert des données (routage, sélection de canal de transmission, etc.). Nous pouvons trouver des approches préventives ou curatives sur les différentes couches (réseau, liaison de données, etc.)

- ✓ **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.

III.4 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 III-2.

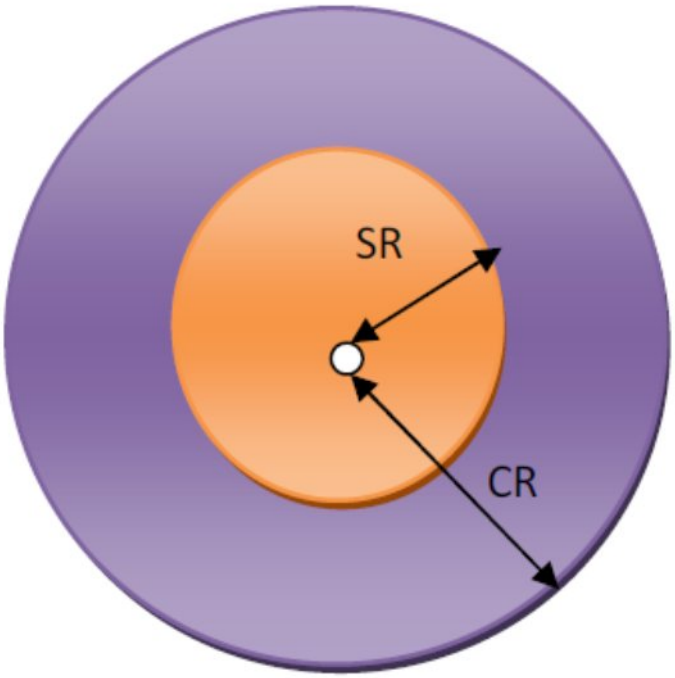


Figure III-2 : la couverture dans une zone.

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.

III.4.1 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 III.3 ci-dessous :

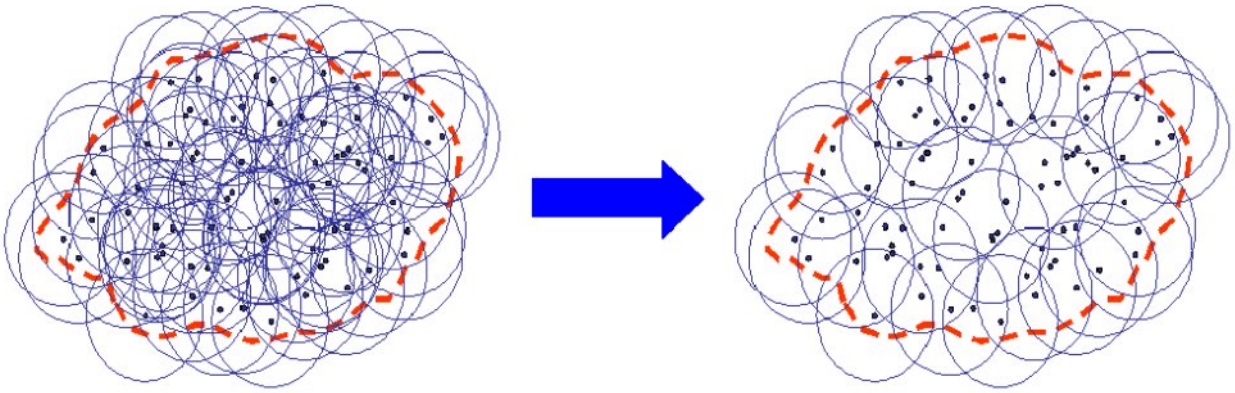


Figure III-3 : l'économisassions de l'énergie.

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.

III.4.2 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.

III.5 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.5.1 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 [15].

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 œuvre et établissement de chemin:

Chaque nœud est caractérisé par son identifiant nœud (Nj), le niveau (HCj), nœud parent (Pj), un tableau (Aj) 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 noeuds ordinaires avec d'autres HCj = ∞, Pj = 1.

Une fois les noeuds sont déployés, la station de base diffuse un message d'avertissement ADVT (Nj, HCj) pour découvrir les noeuds qui sont voisins à la station de base. Ces noeuds sont considérés comme des noeuds de niveau 1 puisqu'ils se trouvent à un saut de la station de base qui est considérée comme un noeud parent pour ces noeuds de niveau 1. Lorsqu'un noeud 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 noeud de niveau N +1 si le nombre de sauts reçu est N. Ainsi, le message ADVT est utilisé pour hiérarchiser le réseau.

- 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 Nj, HCj en fonction de la réponse au message Hello. Si la réponse provient d'un nœud de niveau inférieur, les nœuds fils gardent leur Nj , 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.5.2Protocole de routage tolérant aux pannes multi-niveaux avec ordonnancement d'activité de capteurs (FMS) :

Le protocole FMS [16] permet de maintenir la connectivité du réseau, même si un nœud est sur le point d'épuiser son énergie. Il permet aussi d'assurer la fiabilité et la rapidité de livraison des données à la station de base car il est conçu pour les applications orientées événement. Généralement, les capteurs sont déployés aléatoirement et en grand nombre. De ce fait, il y aura une redondance dans la livraison de données ce qui a une conséquence sur la durée de vie du réseau de capteurs. Pour se remédier à cet 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 de 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 transmettent les paquets de données. Avant qu'un nœud passe en mode veille, il devra informer ses nœuds fils afin qu'ils choisissent un autre nœud parent pour relayer les données. En outre, quand un nœud est en mode veille, il passera en mode actif que si son énergie est supérieure à une certaine valeur seuil. Le choix des noeuds actifs se fait aléatoirement et d'une manière périodique pour que le noeud 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 perdre son énergie. Ainsi, FMS est considéré comme un protocole fiable et tolérant aux pannes.

III.5.3Protocole de routage temps réel tolérant aux pannes (DMRF)

DMRF [18] 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 noeuds, la liste de l'état du réseau (information sur la congestion d'un noeud, 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 noeud, 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 noeud relais en se basant sur le taux de transmission de données et des informations locales. Une fois les noeuds

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 noeud 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 noeud 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.

Conclusion

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.

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 modifié le LEACH, 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 performance via l'implémentation et l'évaluation des deux protocoles. Pour cela, nous commençons 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.1 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 les codes des deux protocoles sont programmés. 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.1.1 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 au 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 [20]**.** 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 [21]**.** 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.1.1.1 Pourquoi TinyOS [22]

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, Contiki, MANTIS. TinyOS reste néanmoins le plus répandu pour les RCSF. 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.

- 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.1.1.2 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.1.2 NesC [23]

NesC est un langage de programmation orienté composants syntaxiquement proche du langage C. Il est conçu pour la réalisation des systèmes embarqués et 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.

- 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.

IV.1.3 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 .

IV.1.3.1 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.2 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.

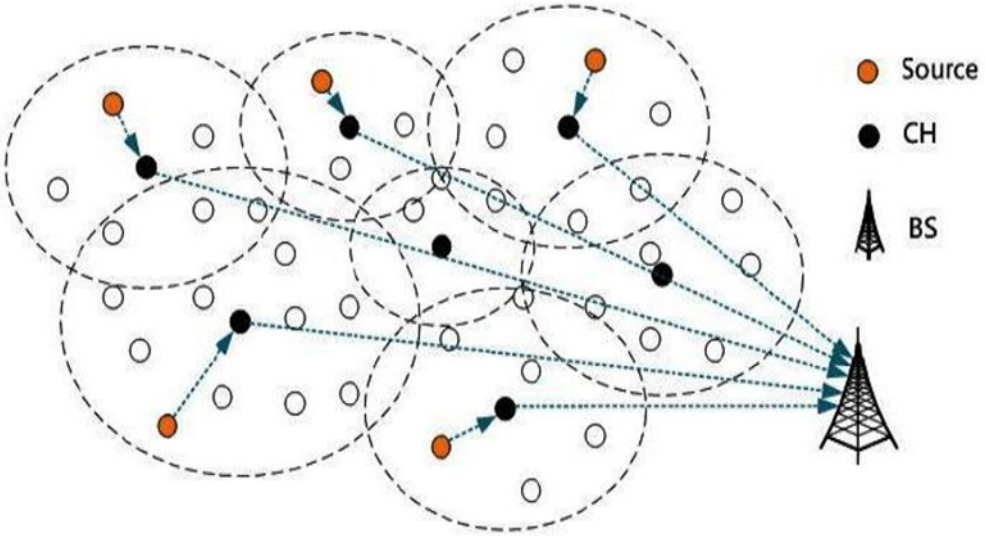


Figure IV. 2 : Architecture du routage hiérarchique LEACH

IV.2.1 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 station de base 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 noeuds. Pour éviter ce grave problème, LEACH utilise la notion de cycles (Rounds). Au début de chaque cycle, chaque noeud 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 noeud.

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 noeuds 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 noeuds 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-head effectueront une transmission vers le puits. Le nombre optimal de cluster-head 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-head ce qui peut diminuer le gain en énergie.

- Schéma montrant l'élection de Cluster-Head dans le protocole LEACH

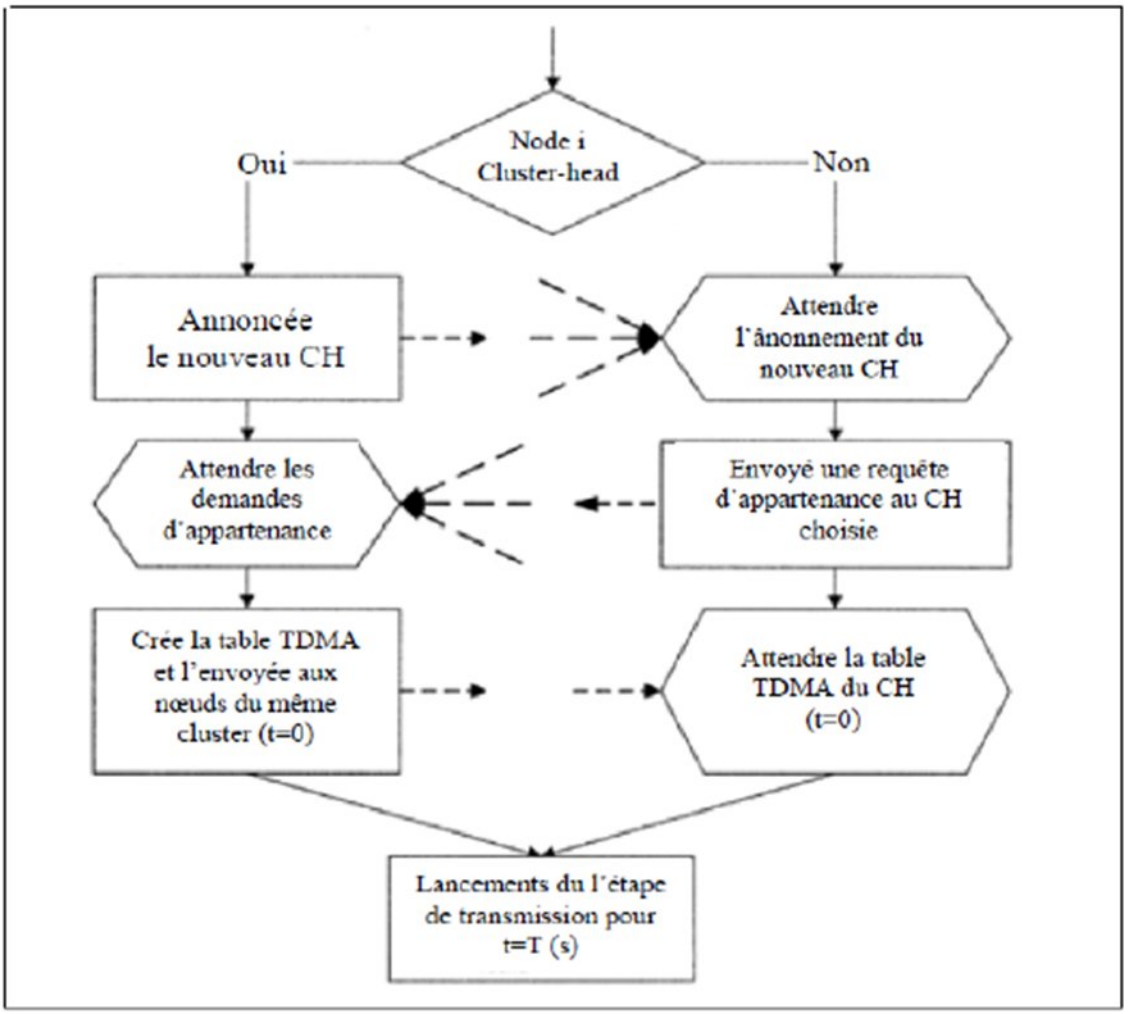


Figure IV. 3 : Algorithme de l'élection de cluster-head.

Cette phase commence par l'annonce du nouveau round par le nœud puits, et, par la prise de décision locale d'un nœud pour devenir CH avec une certaine probabilité Pi(t) au début du round r+1 qui commence à l'instant t. Chaque nœud i génère un nombre aléatoire entre 0 et 1. Si ce nombre est inférieur à Pi(t), le nœud deviendra CH durant le round r+1. Pi(t) est calculé en fonction de K et de round r **,k** est le nombre de cluster- head .

Après qu'un nœud soit élu CH, il doit informer les autres nœuds non-CH de son nouveau rang dans le round courant. Pour cela, un message

d'avertissement ADV contenant l'identificateur du CH est diffusé à tous les nœuds non-CH. La diffusion permet de s'assurer que tous les nœuds non-CH ont reçu le message. La décision est basée donc sur l'amplitude du signal reçu; le CH ayant le signal le plus fort (i.e. le plus proche) sera choisi. En cas d'égalité des signaux, les nœuds non-CH choisissent aléatoirement leur CH.

- **Description de T-Leach :**

Lorsque le cluster-head tombe en panne, les nœuds capteurs qui étaient ses fils dans le cluster doivent trouver un autre chemin afin de transmettre leurs informations à la station de base, sinon il y'aura un grand risque de la défaillance de tout le réseau.

Les nœuds (clusters-head) envoient périodiquement des paquets «hello » à leurs nœuds fils, après une certaine période qu'un nœud n'as pas reçu ce paquet de son père, il le considère comme défaillant, et à ce moment là que le capteur doit trouver un autre cluster-head qui est le cluster-head adjoint dans le même cluster (groupe).

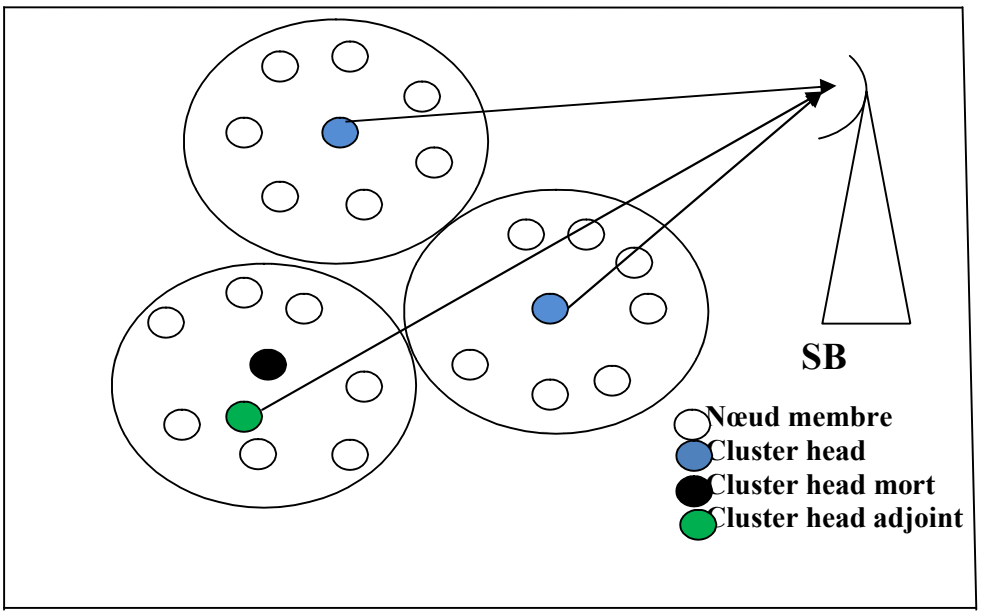


Figure IV. 4 : Description de T- LEACH

IV.2.2 La durée de vie du réseau de LEACH [24]

Au niveau du protocole LEACH, la durée de vie du réseau est faible, parce que dans LEACH, les noeuds 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-5. 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.

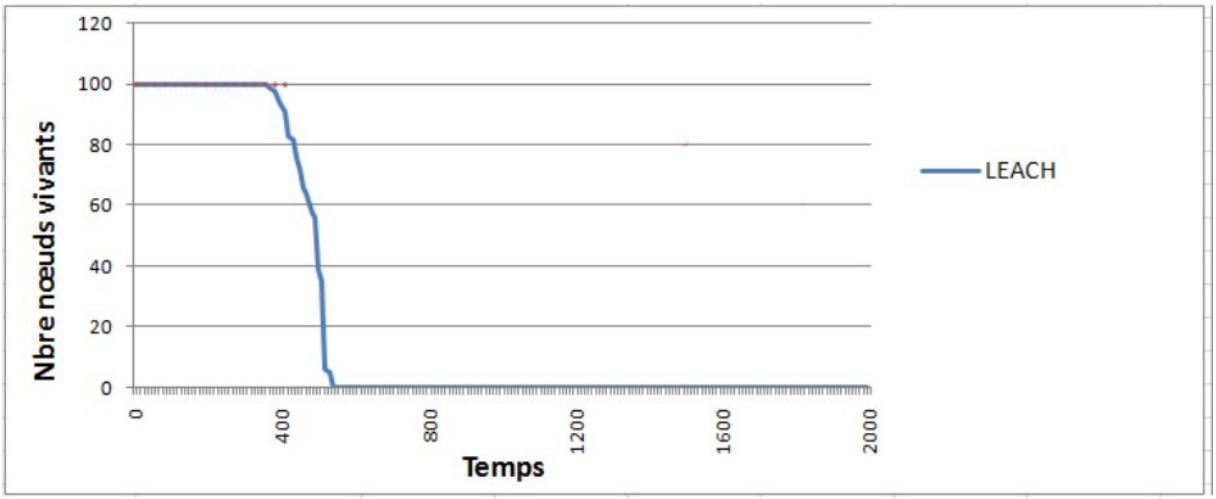


Figure IV-5 : La durée de vie du réseau au niveau de LEACH.

VI.3 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.3.1 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.3.2 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.3.2.1 Structures de données

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;
```

B) 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.

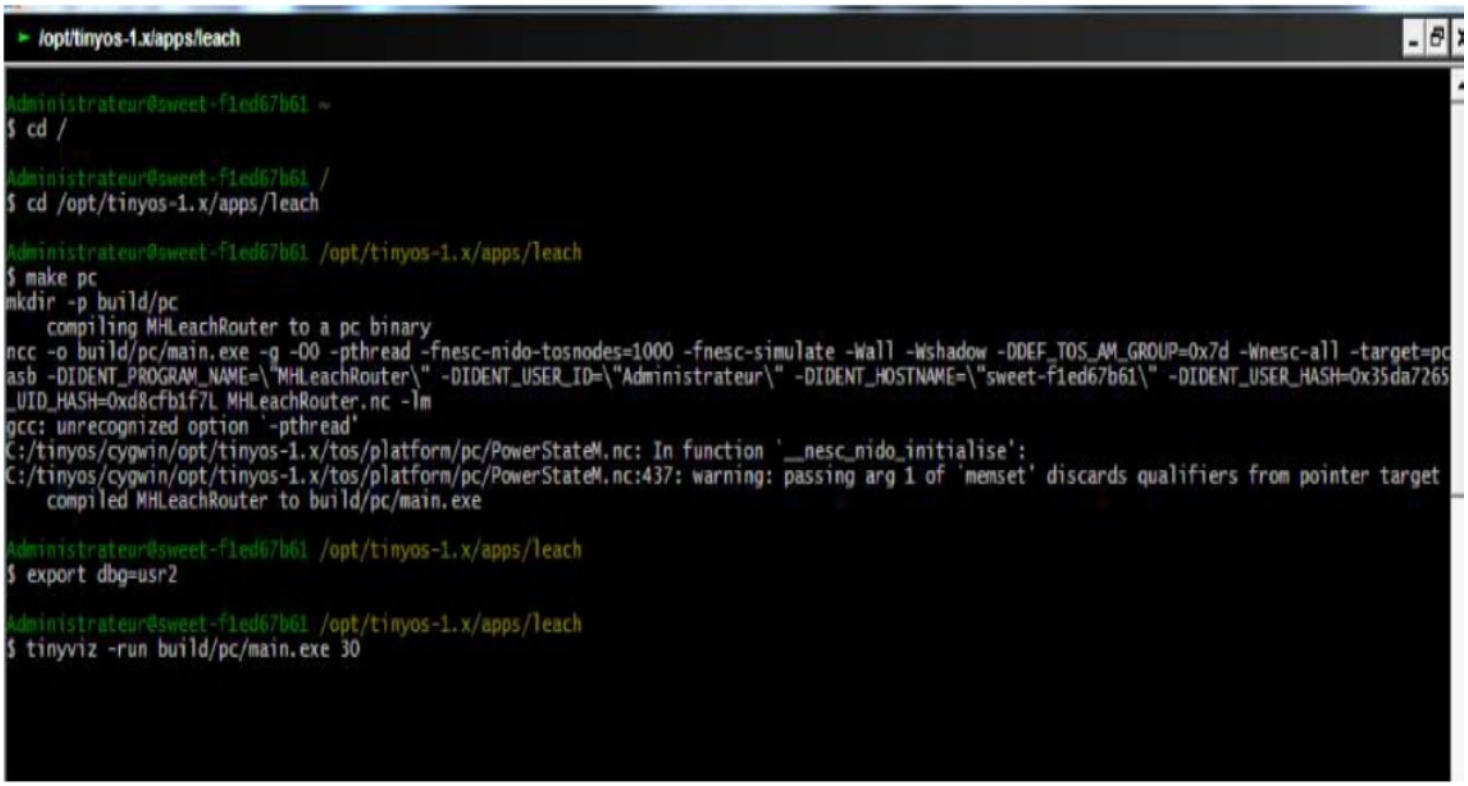


Figure IV-6 : l'interface Cygwin.

- 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 , les transmissions broadcast qui se passent durant les différentes étapes de l'algorithme LEACH. 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-5.

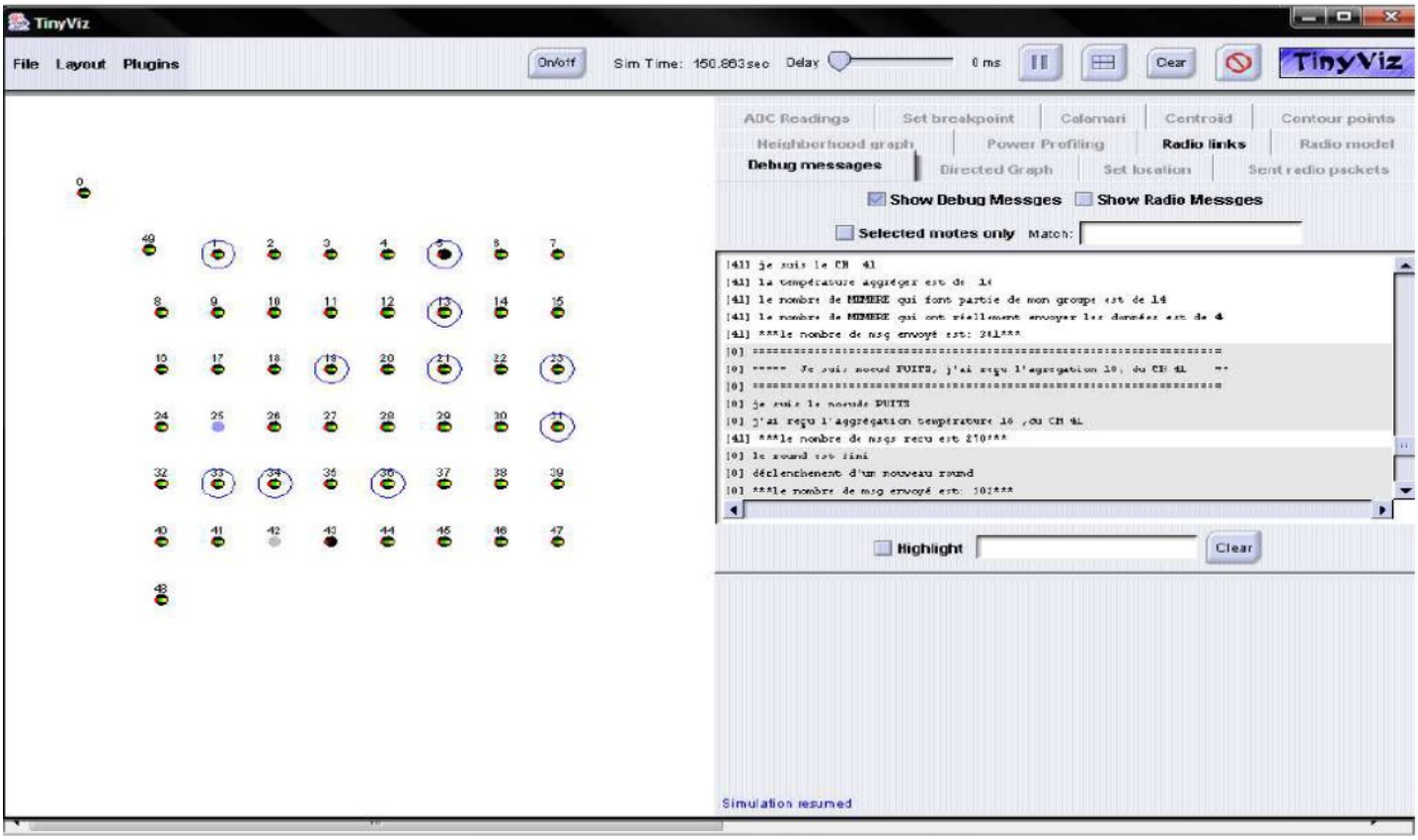


Figure IV-7 : Déclenchement et relai du nouveau round, annonce du CH 15.

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-8 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.

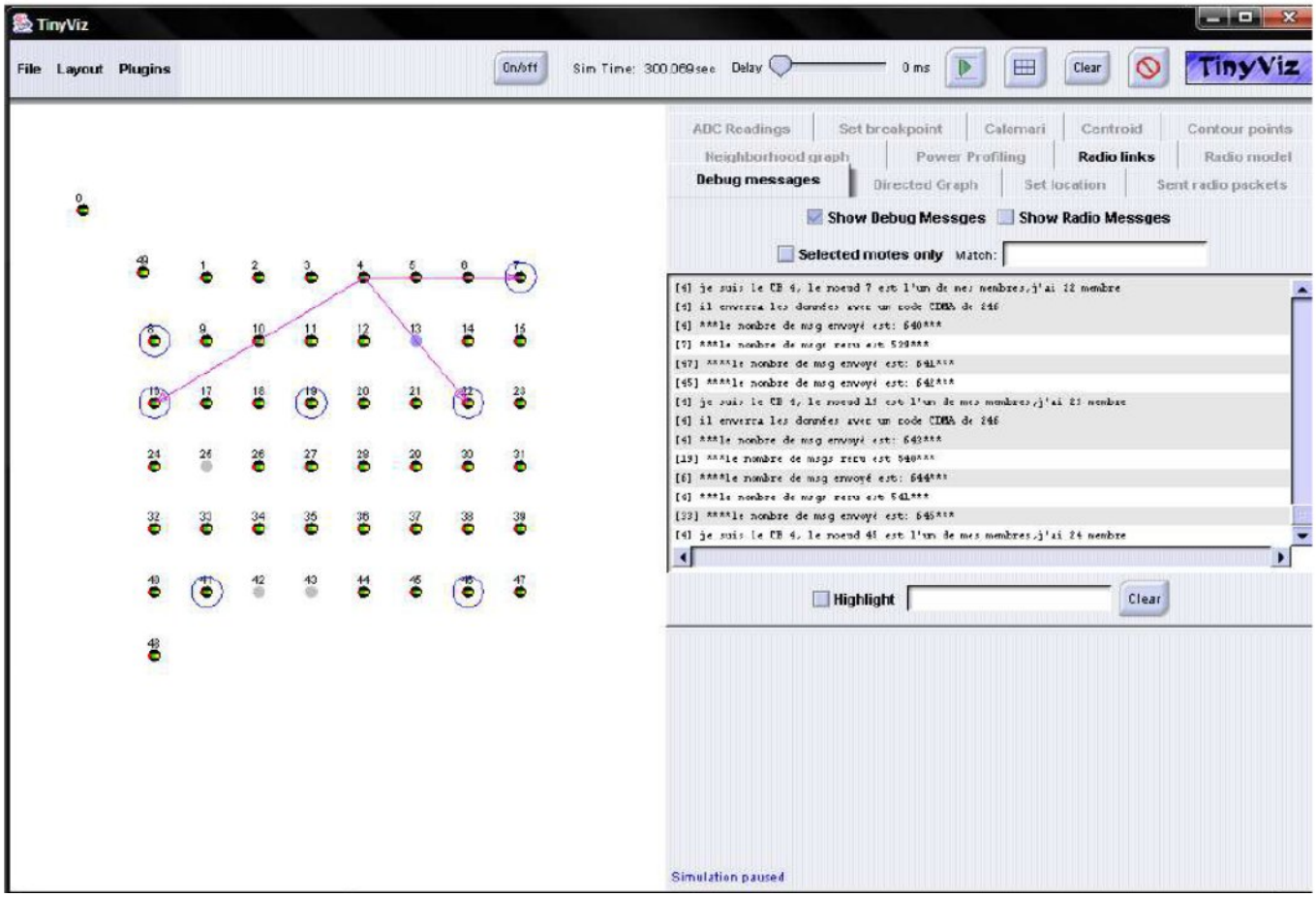


Figure IV-8: Formation de groupes et envoi des données.

3. Envoi des résultats d'agrégation au nœud puits. Dans la figure IV-7, le CH 4 agrège les données reçues et envoie son résultat d'agrégation au nœud puits comme dans la figure IV-8

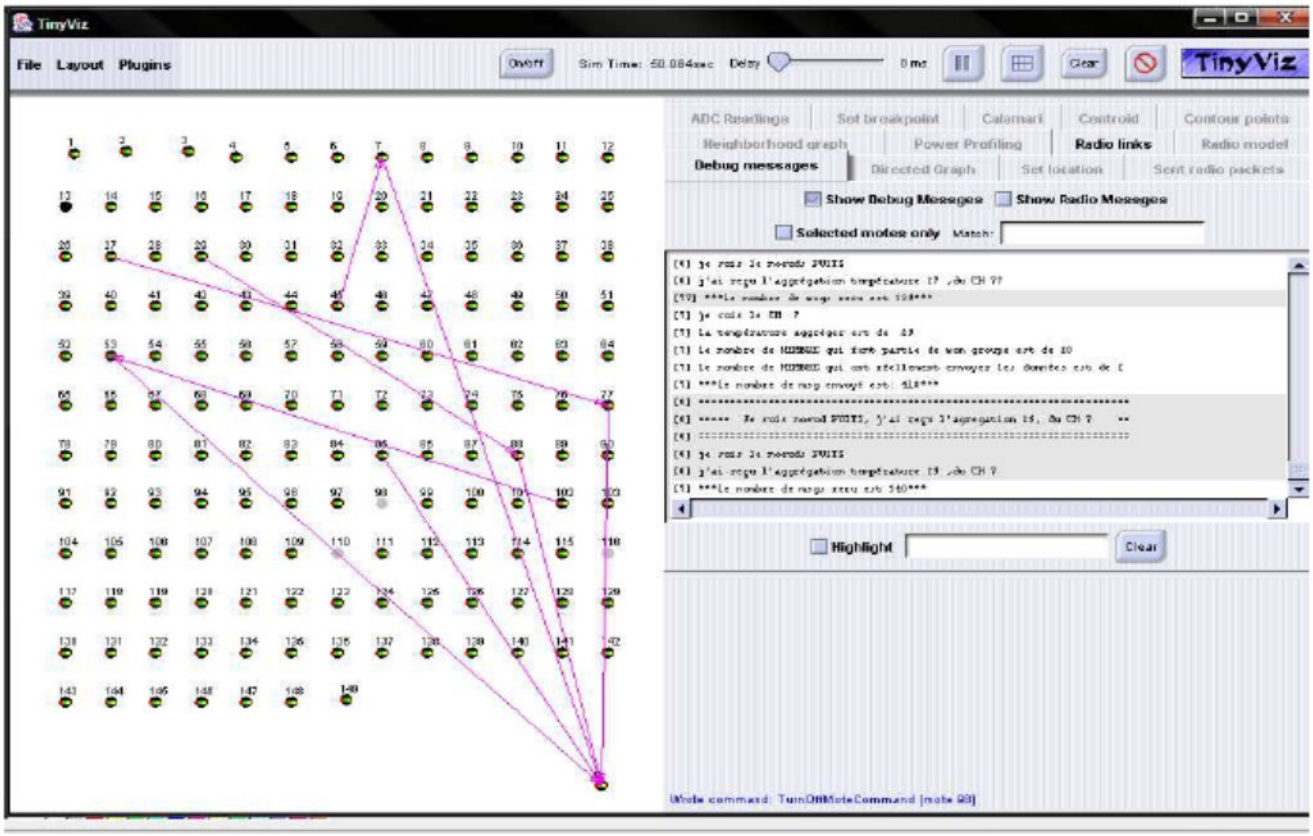


Figure IV-8: Envoi du résultat d'agrégation du CH au nœud puits.

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 noeud 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 noeud 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 dès que le cluster- head principal tombe en panne, ce qui permet d'assurer la livraison des données à la station de base.

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. Pour cela nous avons utilisé MATLAB qui va nous permettre de générer les différents graphes de comparaison ainsi déduire une conclusion.

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 données 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 le 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 decluster- head désactivé de 1 jusqu'à 5 clusters.

_ Pour 50 nœuds : Dans le tableau IV-1 l'exécution des 50 nœuds avec les cluster-heads 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 noeuds)

Comme illustre la figure IV-9, 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.

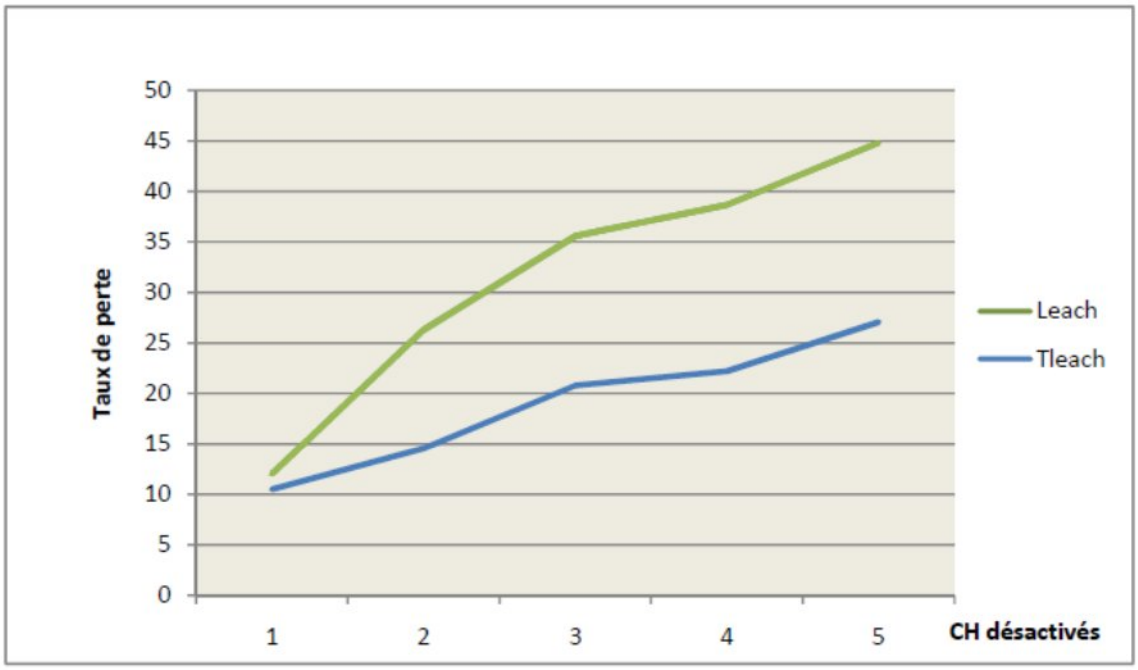


Figure IV-9: Taux de pertes de paquets pour 50 nœuds.

Le tableau IV-2 nous montre l'un des cas de l'exécution dans un réseau de taille de 100 nœuds.

	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%

Comme le montre la figure IV-10 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.

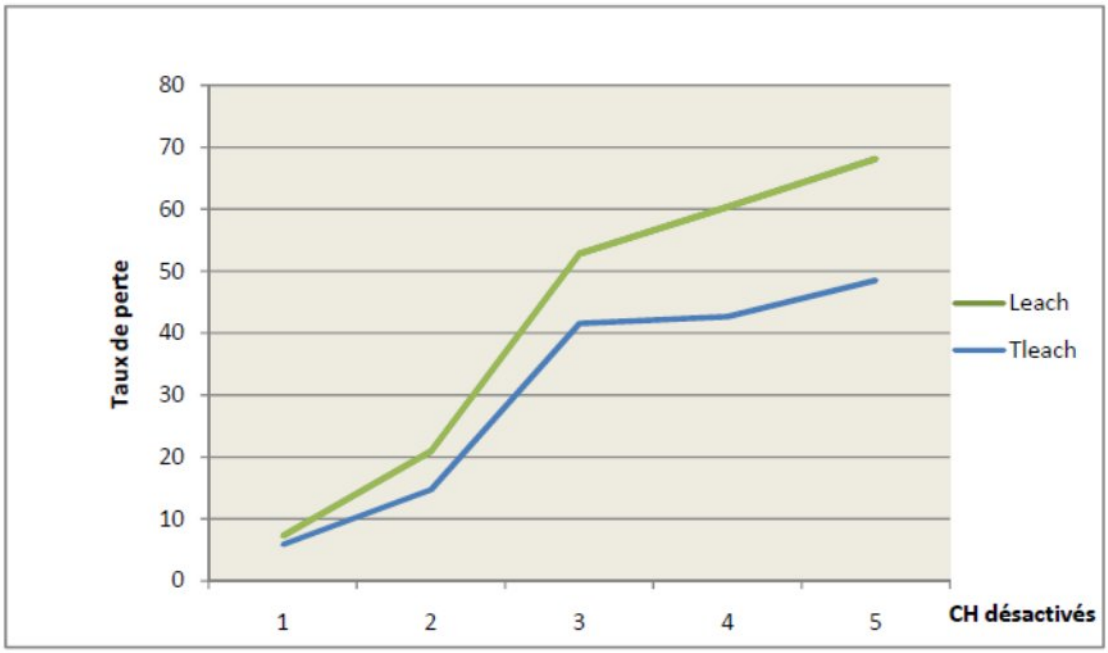


Figure IV-10: Taux de pertes de paquets pour 100 nœuds.

Dans le deuxième cas c'est le nombre des clusters head adjoint désactivé qui est fixé et les nombres des nœuds du réseau varient. Le tableau IV.3 suivante montre l'un des cas des exécutions.

	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 passer les données à la station de base.

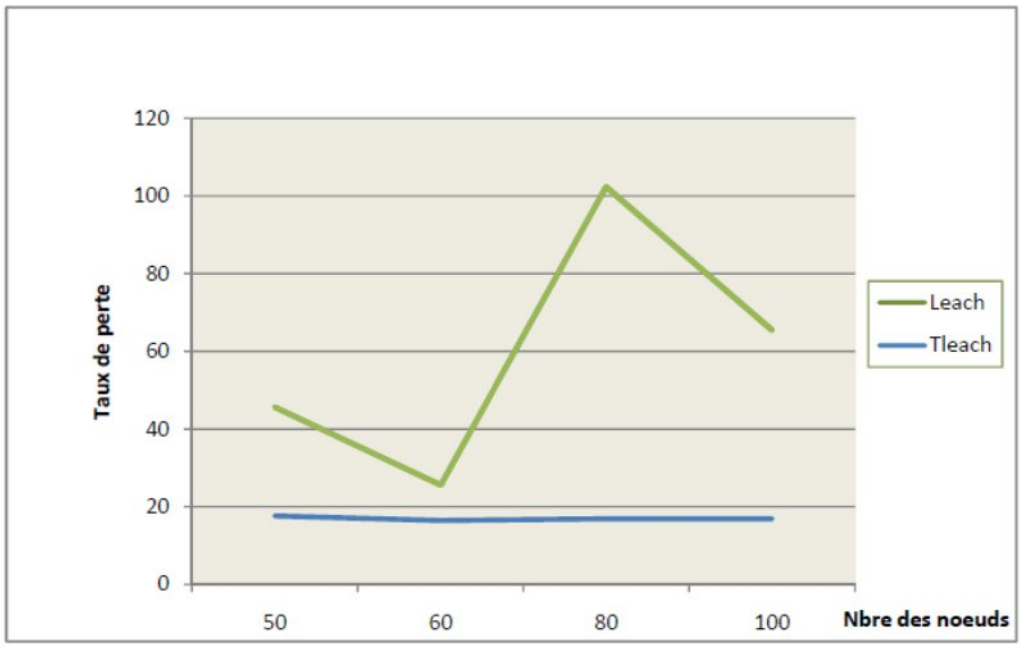


Figure IV-11 : Taux de perte pour 5 CH désactivés.

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.

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.

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.

[1] J. Champ et C Saad, «Un Nouvel Algorithme de Routage Géographique dans les Réseaux de Capteurs», Schedae, 2007, prépublication n° 19, (fascicule n° 2, p. 95-103).

[2] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam and E. Cayirci, «A Survey on Sensor Networks», IEEE Communications Magazine, August 2002.

[3] C. D. Faundez, «Transmission d'images sur les réseaux de capteurs sans fil sous la contrainte de l'´energie », Thèse de Doctorat en automatique, Université Henri Poincaré, Nancy 1,2009.

[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] P. Mohapatra and S. V. Krishnamurthy: «Ad Hoc Networks Technologies and Protocols», Springer Verlag Telos, 2004, ISBN: 0-387 22689-3.

[6] M. Badet et W. Bonneau. «Mise en place d'une plateforme de test et d'expérimentation», Projet de Master Technologie de l'Internet 1ere année, Université Pau et des pays de l'Adour. 2006.

[7] C. F. GARCIA-HERNANDEZ, P. H. IBARGUENGOYTIA-GONZALEZ, J. GARCIAHERNANDEZ and J. A. PEREZ-DIAZ: «Wireless Sensor Networks and Applications: a Survey», IJCSNS International Journal of Computer Science and Network Security, 2007, Vo.7, No.3, pp. 264-273.

[8] K.Beydoun . « Conception d'un protocole de routage hiérarchique pour les réseaux de capteurs »Thèse de doctorat, Spécialité :Informatique, l'u.f.r des sciences et techniques de l'université de Franche-Comté,2009.

[9] Yacine Challal, « Réseaux de Capteurs Sans Fils », Cours, Systèmes Intelligents pour le Transport, Université de Technologie de Compiègne, France, 17 Novembre 2008.

[10] Wassim Znaidi, « Modélisation formelle de réseaux de capteurs à partir de TinyOS », Projet de fin d'études, Ecole Polytechnique de Tunisie, 2006.

[11] https://fr.wikipedia.org/wiki/Routage

[12] I. Mahgoub and J. Ibriq, «Cluster-Based Routing in Wireless Sensor Networks: Issues and Challenges», International Symposium on Performance Evaluation of Computer and Telecommunication Systems' 2004 (SPECTS' 04), Page(s):759-769, California University, 2004.

[13] H. Hadjammar et N. Doufene, « Routage dans les réseaux de capteurs optimisation du protocole Directed Diffusion», Projet de fin d'étude, Institut National de formation en Informatique INI, Algérie, 2006.

[14] W. Bechkit, « Un nouveau protocole de routage avec conservation d'énergie dans les réseaux de capteurs sans fil », Mémoire d'ingénieur, Ecole nationale Supérieure d'Informatique ESI, Juin 2009.

[15] 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.

[16] 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.

[17] 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.

[18] 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).

[19] 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.

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

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

[22] 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.

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

[24] LEA2C : Une nouvelle approche de routage dans les réseaux de capteurs pour l'optimisation de la consommation d'énergie Lahcene DEHNI , Younès BENNANI , Francine KRIEF

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

6- Aller à **shell** et taper: **cd /opt/tinyos-1.x/tools/java/net/tinyos/sim**

make clean

make