Université Abou Bekr Belkaid
Tlemcen Algérie
جامعة أبي بكر بلقايد
تلمسان الجزائر

République Algérienne Démocratique et Populaire
Université Abou Bakr Belkaid– Tlemcen

Faculté desSciences
Département d'Informatique

Mémoire de fin d'études

Pour l'obtention du diplôme de Master en Informatique

Option: Réseaux et Systèmes Distribués (R.S.D)

Thème

Evaluation de LEACH dans un environnement fortement bruité

Réalisé par :

REMMAS Fatima Zohra

Présenté le 25 Juin 2015 devant le jury composé de:

- *Mr BENMAMMAR Badr* — *Président*
- *Mr LEHSAINI Mohamed* — *Encadrant*
- *Mme TABET HELLEL Chifaa* — *Co-encadrante*
- *Mr MANA Mohammed* — *Examinateur*
- *Mr BENZIAN Yaghmoracene* — *Examinateur*

Année universitaire:2014-2015

Remerciements

Tout d'abord, je remercie « ALLAH» pour son aide et sa bénédiction.

Ce mémoire est le résultat d'un travail de recherche de près de cinq ans. En préambule, je veux adresser tous mes remerciements aux professeurs qui m'ont enseignés.

En commençant à remercier tout d'abord Monsieur « **LEHSAINI MOHAMED** », directeur de recherche de ce mémoire, pour son aide précieuse et pour le temps qu'il m'a consacré.

Merci à mme «**TABET HELLEL Chifaa**» pour son aide, pour son soutien moral et scientifique efficace et constant, durant toute cette durée de travail.

Merci à tous les membres du jury qui ont fait l'honneur de prendre notre modeste travail en considération et en suite de le juger : **Mr BENMAMMAR Badr** qui m'a fait l'honneur de présider ce jury, **Mr MANA Mohammed** et **Mr BENZIAN Yaghmoracene**.

Enfin, j'adresse mes plus sincères remerciements à ma famille, mes parents, plus particulièrement à ma grand-mère « AOUMRIA »qui nous a quitté depuis quelques temps, a mes amis, qui m'ont accompagnés, aidés, soutenus et encouragés tout au long de la réalisation de ce mémoire.

Merci à toutes et à tous.

Dédicaces

Je dédie ce mémoire à :

Mes chers parents qu'aucune dédicace ne saurait exprimer mon respect pour eux, mon amour éternel et ma considération pour les sacrifices qu'ils m'ont consenti. Puisse Dieu vous accorde santé, bonheur et longue vie.

Mes frères Djamel et Sid Ahmed, ma grand-mère paternel le .

Mon frère Amine, sa femme Anfel et son fils kaçimo

Mes oncles Sid Ahmed, Hadj bachir et sa femme Chahra, mes tantes Samira, Fadela, Aicha , Khadidja et sa fille Ilhem.

Mes cousins Kadirou, Belkacem, mes cousines Oumria et Djahida

Mes amies fatima, Manel et toute la promotion RSD

Sommaire

Introduction générale 1
Chapitre I Les réseaux de capteurs sans fil 3
I.1 Introduction 3
I.2 Les capteurs 3
I.2.1 Qu'est-ce qu'un capteur ? 3
I.2.2 Un capteur intelligent 3
I.2.3 Architecture d'un capteur 4
a) Architecture matérielle 4
b) Architecture Logicielle 5
I.2.4 Caractéristiques des capteurs 5
I.3 Réseaux de capteurs sans fil (RCSF) 6
I.3.1 Architecture d'un RCSF 6
I.3.2 Classification des RCSF 7
a) Selon le mode d'acquisition et de livraison des données au puits 7
b) Selon la distance entre les nœuds capteurs et le puits 8
c) Selon le modèle de mobilité dans le réseau 8
d) Selon les capacités des nœuds du réseau 8
I.3.3 Domaines d'application des RCSF 9
a) Applications militaires 9
b) Application médicales 9
c) Applications domestiques 9
f) Agriculture de précision 10
g) Application industrielles 10
I.3.4 Caractéristiques des réseaux de capteurs 11
I.4 Facteurs et contraintes de conception d'un RCSF 11
I.5 Conclusion 12
Chapitre II Les protocoles de routage dans les RCSF 14
II.1 Introduction 14
II.2 Classification des protocoles de routage pour les RCSF 14
II.2.1 Selon la topologie du réseau 15
a) Topologie plate 15

b) Topologie hiérarchique 16
II.2.2 Selon le paradigme de communication 16
a) Centré-nœuds (Node-centric) 16
b) Centré-données (Data-centric) 16
c) Basé-localisation (location-based) 17
II.2.3 Selon le mode de fonctionnement du protocole 17
a) Routage basé sur les multi-chemins 17
b) Routage basé sur les requêtes 17
c) Routage basé sur la négociation 18
d) Routage basé sur la qualité de service 18
II.2.4 Selon le mode l'établissement des chemins 18
a) Les protocoles proactifs 18
b) Les protocoles réactifs 18
c) Protocoles hybrides 19
II.3 LEACH et ses variantes 19
II.3.1 Le protocole LEACH 19
a) La phase de formation de clusters 20
b) La phase de transfert de données 22
II.3.2 Le protocole LEACH-C 22
II.3.3 Le protocole LEACH-F 22
II.4 Conclusion 23
Chapitre III Evaluation de LEACH dans un environnement non-idéal 24
III.1 Introduction 24
III.2 Préliminaires et hypothèses 25
III.2.1 Les causes de l'irrégularité de la radio 25
III.2.2 Les modèles radio 25
a) Le modèle radio isotrope 26
b) Le modèle de l'irrégularité de la radio 26
III.3 Outils logiciels 27
III.3.1 Le système d'exploitation TinyOs 27
III.3.2 Le langage de programmation NesC 27
a) Les Fichiers dans NesC 27
b) Eléments d'un programme NesC 28
c) Types de données utilisés par NesC 30

d) Types de fonctions en NesC 31
III.4 Implémentation et évaluation de LEACH 31
III.4.1 Les fichiers de l'application 31
III.4.2 Implémentation de LEACH 32
III.4.3 Environnement d'exécution du simulateur 32
III.4.4 Critères de performances 33
III.4.5 Paramètres de la simulation 33
III.4.6 Résultats et interprétations 34
a) Le nombre de paquets reçus (5 périodes) 34
b) Le nombre de paquets perdus (5 périodes) 35
c) Energie dissipée (5 périodes) 35
d) Nombre de paquets perdus vs. taille du réseau (avec des CH désactivés) 36
e) Le nombre de paquets reçus (100 nœuds) 37
f) Nombre de paquets perdus (100 nœuds) 37
g) Energie dissipée (100 nœuds) 38
h) Nombre de paquets perdus en fonction du temps (avec des CHs désactivés) dans LEACH 38
III.5 Conclusion 39
Conclusion générale 39
Références bibliographiques 40

Liste des Figures

Figure 1: Architecture d'un capteur sans fil [5] 4
Figure 2: Capteurs TelosB et MicaZ 5
Figure 3: Exemple de réseaux de capteurs 7
Figure 4: Quelques domaines d'applications pour les RCSF 10
Figure 5: Classification des protocoles de routage pour les RCSF [6] 15
Figure 6: Paradigme de routage dans LEACH 20
Figure 7: La phase de formation de clusters 21
Figure 8: Les temps de chacune des deux phases dans une période [2] 22
Figure 9: Le modèle de propagation DOI [32] 26
Figure 10: Interface Cygwin 33
Figure 11: Nombre de paquets reçus vs. nombre de nœuds pendant 5 périodes 34
Figure 12: Nombre de paquets perdus vs. nombre de nœuds (5 périodes) 35
Figure 13: Energie dissipée vs. nombre de nœuds (5 périodes) 36
Figure 14: Nombre de paquets reçus en fonction du temps (pour 100 nœuds) 37
Figure 15: Nombre de paquets perdus en fonction du temps (en périodes) 37
Figure 16: Energie dissipée en fonction du temps (100 nœuds) 38

Liste des Tableaux

Tableau 1: Caractéristiques de MicaZ [9] et TelosB [10] 6
Tableau 2: Paramètres de simulation 34
Tableau 3: Nombre de paquets perdus vs. taille du réseau (avec des CHs désactivés) dans LEACH 36
Tableau 4: Nombre de paquets perdus vs. taille du réseau (avec CHs désactivés) dans LEACH_modifié 36
Tableau 5: Nombre de paquets perdus en fonction du temps (avec des CHs désactivés) dans LEACH 38
Tableau 6: Nombre de paquets perdus en fonction du temps (avec des CHs désactivé) dans LEACH modifié 38

Introduction générale

Introduction générale

Ces dernières années, les réseaux sans fil sont devenus de plus en plus populaires du fait de leur facilité de déploiement et leur faible coût. Ils offrent des solutions prometteuses pour favoriser la mobilité ainsi que des services essentiels là où l'installation d'infrastructures n'est pas possible ou coûteuse.

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

Les RCSF sont souvent composés d'un grand nombre de nœuds capteurs de faibles capacités en termes d'énergie, de mémoire, de calcul et de communication. Le but des RCSF est de surveiller un environnement physique réel et d'en rendre compte au centre de contrôle les informations pertinentes sur l'environnement surveillé. Ils sont utiles dans de nombreux domaines (environnemental, militaire, médical, etc.).

Dans les RCSF, le routage des données vers la station de base est une opération fondamentale. Néanmoins, les schémas de routage conçus doivent prendre en considération les changements de la topologie du réseau, ainsi que d'autres caractéristiques comme la bande passante, la qualité des liens, la limitation d'énergie, etc. Par ailleurs, dans les RCSF les nœuds capteurs sont sujets à des pannes intentionnelles ou accidentelles et les liens radio sont volatiles. Ces deux facteurs ont un impact négatif sur les performances des protocoles de routage. Dans ce contexte, plusieurs recherches ont été menées notamment pour garantir la fiabilité de livraison dans des environnements perturbés.

L'objectif de ce mémoire est de traiter le problème de la tolérance aux pannes dans les RCSF en se focalisant sur le protocole LEACH (Low-Energy Adaptive Clustering Hierarchy) [1,2]. Au début, on a évalué LEACH dans un environnement idéal. Puis, on a proposé une version améliorée de LEACH qui est adaptable à un environnement non idéal.

Ce mémoire est organisé en trois chapitres encadrés par une introduction et une conclusion:

Dans le premier chapitre, on présente les concepts généraux relatifs au domaine des réseaux de capteurs sans fil. Ces concepts nous permettent de comprendre les spécificités de ces réseaux et les contraintes imposées pour concevoir d'autres protocoles.

Dans le deuxième chapitre, une brève classification des protocoles de routage dans les RCSF est donnée. Puis, une présentation plus ou moins détaillée du protocole LEACH et de ses variantes est exhibée.

Dans le troisième chapitre, on a présenté tout d'abord les modèles de propagation radio et on a mis l'accent sur le modèle d'irrégularité radio dans lequel on a évalué LEACH. Malheureusement, les résultats obtenus ont montré que LEACH perd grandement ses performances dans un environnement non-idéal.

Chapitre I

Les réseaux de capteurs sans fil

Chapitre I
Les réseaux de capteurs sans fil

I.1 Introduction

Les concepts des réseaux de capteurs sans fil (RCSF) sont bien particuliers comparativement aux réseaux sans fil puisque les RCSF sont composés d'équipements à ressources limitées en termes de calcul, de stockage et d'énergie.

Dans ce chapitre, on commence par présenter les capteurs, leurs architectures et leurs caractéristiques. Ensuite, on présente aussi les réseaux de capteurs, leurs architectures, leur classification, leurs caractéristiques, leurs domaines d'applications, et à la fin leurs contraintes de conception.

I.2 Les capteurs

I.2.1 Qu'est-ce qu'un capteur ?

Un capteur est le dispositif qui transforme une grandeur physique observée (température, pression, humidité, etc.) en une grandeur utilisable (intensité électrique, position d'un flotteur) [3]. Il possède au moins un transducteur dont le rôle est de convertir une grandeur physique en une autre.

I.2.2 Un capteur intelligent

Le terme capteur intelligent a été utilisé dans l'industrie des capteurs pour désigner des capteurs qui ne fournissent pas seulement des mesures, mais aussi une fonctionnalité aux mesures spécifiques [4]. Comparativement à un capteur classique, un capteur intelligent intègre de nombreux éléments électroniques additionnels, ainsi que des unités programmables et des aspects logiciels nécessaires au traitement des données, aux calculs, à la communication numérique [3]. Il est donc caractérisé par sa capacité d'effectuer une collecte de mesures, les traiter et à les communiquer au monde extérieur [4].

I.2.3 Architecture d'un capteur

Dans cette section, nous distinguons les deux parties qui composent un capteur :

a) Architecture matérielle

La figure I-1 est l'illustration la plus générale de l'architecture d'un capteur dit intelligent.

Figure 1: Architecture d'un capteur sans fil [5]

Cette architecture s'articule au tourde quatre unités :

- **l'unité de traitement**: c'est l'unité principale du capteur. Elle est généralement représentée par un processeur couplé à une mémoire vive. Son rôle est de contrôler le bon fonctionnement des autres unités. Sur certains capteurs elle peut embarquer un système d'exploitation pour faire fonctionner le capteur. Elle peut aussi être couplée à une unité de stockage, qui servira par exemple à y enregistrer les informations collectées par l'unité d'acquisition de données.
- **l'unité d'acquisition**: elle permet la mesure des grandeurs physiques ou analogiques et leur conversion en données numériques. Elle est composée du capteur lui-même et de l'ADC[1] qui permet la conversion des données. Le capteur est chargé de récupérer les signaux analogiques qu'il transmet à l'ADC qui a pour rôle de transformer et de communiquer les données analogiques en données numériques compréhensibles pour l'unité de traitement.
- **l'unité de communication** : elle a pour fonction de transmettre et recevoir les données. Elle est équipée d'un couple émetteur/récepteur pour communiquer au sein du

[1] ADC : Analog-to-Digital Converter

réseau. Il existe cependant d'autres possibilités de transmission (optique, infrarouge, etc. . .).

- **l'unité d'alimentation** : c'est un élément primordial de l'architecture du capteur, c'est elle qui fournit en énergie toutes les autres unités. Elle correspond le plus souvent à une batterie ou une pile alimentant le capteur, dont les ressources limitées en font une problématique propre à ce type de réseau puisque ces derniers sont généralement déployés dans des zones non accessibles. La réalisation récente d'unité d'alimentation à base de panneaux solaires tente d'apporter une solution pour prolonger sa durée de vie [6].

Par ailleurs, un capteur peut être doté d'autres unités. Citons, entre autres, la possibilité d'ajouter une unité de localisation, tel qu'un GPS, un mobilisateur (drone par exemple) pour assurer la mobilité du capteur, ou une unité spécifique de capture comme une caméra pour de l'acquisition vidéo.

b) Architecture Logicielle

La contrainte énergétique des capteurs exige l'utilisation de systèmes d'exploitation légers tels que TinyOS [7] ou Contiki [8]. Cependant, TinyOS reste toujours le plus utilisé et le plus populaire dans le domaine des RCSF. Il est libre et est utilisé par une large communauté de scientifiques dans des Simulations pour le développement et le test des algorithmes et protocoles réseau.

I.2.4 Caractéristiques des capteurs

Pour illustrer les caractéristiques des capteurs, on présente un type de chaque famille : MicaZ de la famille Mica et TelosB de la famille Telos comme montre dans la figure 2. Le tableau 1 résume les caractéristiques de ces deux types de capteurs.

Figure 2: Capteurs TelosB et MicaZ

Tableau 1: Caractéristiques de MicaZ [9] et TelosB [10]

Caractéristiques	TelosB	MicaZ
Flash	48Kb	512Kb
RAM	10Kb	4Kb
Antenne	Radio Chipcon Wireless Transceiver	Antenne Chipcon CC2420
MCU	1MHz T1 MPS430	8MHz ATmega128
Dimension	65x31x6 Mm	58x32x7 Mm
2 piles AA	Radio+cpu :75mW Sleep mode : 140µW	Radio+cpu mode :63mW Sleep mode : 30µW
External flash	512kB	512kB

I.3 Réseaux de capteurs sans fil (RCSF)

I.3.1 Architecture d'un RCSF

Un réseau de capteurs sans fil (RCSF) consiste en un grand nombre de nœuds capteurs variant de quelques dizaines d'éléments à plusieurs milliers, déployés de manière prédéfinie ou aléatoirement, dans une zone d'intérêt appelée zone de captage, afin de surveiller un événement ou un phénomène physique et de collecter leurs données d'une manière autonome. Les nœuds capteurs utilisent une communication sans fil pour acheminer les données captées vers un nœud collecteur appelé nœud puits ou station de base. La station de base transmet ensuite ces données par Internet ou par satellite à un centre de contrôle distant. Par ailleurs, l'utilisateur peut envoyer des requêtes aux autres nœuds du réseau pour connaitre l'état d'une telle région.

En outre, dans certaines architectures, le modèle peut impliquer des nœuds puissants en termes d'énergie appelés "Passerelles" pour relayer les données [11]. La figure 3 montre un exemple d'un réseau de capteurs.

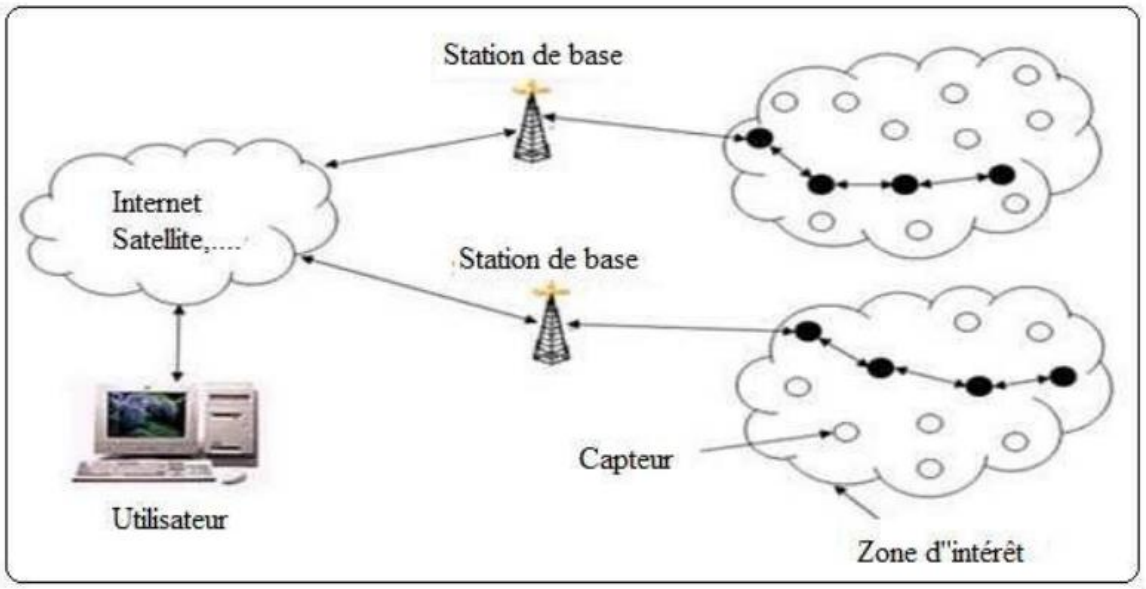

Figure 3: Exemple de réseaux de capteurs

I.3.2 Classification des RCSF

Il existe plusieurs critères pour classifier les RCSF [12, 13, 14]. En effet, pour chaque type d'application, ces réseaux ont des caractéristiques différentes. Ils se distinguent par le mode d'acquisition et de livraison des données au puits, la distance entre les nœuds capteurs et le puits, le modèle démobilisé dans le réseau, les capacités des nœuds du réseau, etc**.**

a) Selon le mode d'acquisition et de livraison des données au puits

Dans les RCSF, le modèle d'acquisition et de livraison des données au puits dépend de l'application et de ses exigences. Il peut être : continu (time-driven), événementiel (event-driven), à base de requête (query-driven), ou hybride. Dans le modèle continu, les nœuds doivent périodiquement réveiller leurs émetteurs pour envoyer les données captées au puits. Le type d'application visé concerne les applications de type " surveillance" où le but principal est d'avoir une information régulière de la zone surveillée.

Dans le modèle orienté événements, les capteurs envoient leurs mesures seulement lorsqu'il y a un évènement qui se produit. Ce type de modèle est recommandé pour les applications de surveillance d'évènements critiques où le but principal est l'obtention d'une information sur l'évènement le plus rapidement possible. Dans le modèle orienté requêtes, les capteurs mesurent des phénomènes et stockent ces mesures dans leur mémoire. Ils envoient ces mesures seulement lorsqu'ils reçoivent des requêtes de la station de base. Le modèle hybride est une combinaison des trois modèles précédents.

b) Selon la distance entre les nœuds capteurs et le puits

Dans cette classification, on distingue les réseaux multi-sauts des réseaux à un seul saut. Dans un RCSF à un seul saut, les nœuds capteurs sont dans le voisinage immédiat de la station de base. Ils envoient alors leurs données captées directement au puits sans passer par aucun autre nœud intermédiaire. Cependant, dans les RCSF multi-sauts, la distance entre les quelques nœuds capteurs et la station de base dépasse leur portée maximale. De ce fait, ces nœuds ne peuvent pas atteindre la station de base directement et ils doivent le faire par l'intermédiaire d'autres nœuds.

c) Selon le modèle de mobilité dans le réseau

Cette classification consiste en une combinaison entre la mobilité des nœuds capteurs et celle de la station de base. Par cette combinaison, nous pouvons distinguer deux grandes catégories de réseaux: réseaux statiques et réseaux dynamiques ou mobiles. On peut par exemple avoir un réseau constitué d'un ensemble de nœuds capteurs mobiles et d'une station de base fixe. Le but de tels réseaux est la plupart du temps l'exploration de zones inaccessibles ou dangereuses.

Un autre exemple est un réseau constitué de capteurs fixes servant à la surveillance d'occurrence d'événements sur une zone géographique et d'une station de base fixe.

d) Selon les capacités des nœuds du réseau

Dans cette catégorie, on distingue deux types de réseaux : les réseaux homogènes des hétérogènes [13].

Dans un réseau de capteurs homogène, tous les nœuds du réseau (nœuds capteurs, le(s) stations de base et les passerelles) ont les mêmes capacités du point de vue énergie, calcul et stockage. Alors que, dans un réseau de capteurs hétérogène il y a quelques nœuds particuliers qui ont plus de capacité de traitement et de communication que les nœuds normaux. Cela améliore l'efficacité énergétique et prolonge la vie du réseau. L'avantage d'un tel réseau est que ces nœuds particuliers peuvent être utilisés pour exécuter les tâches plus complexes comme les coordinateurs, les chefs du cluster (clusterhead), etc. Son inconvénient est qu'il est difficile de mettre en place un tel réseau du fait qu'au moins chaque type de nœuds du réseau sera programmé différemment. Cette opération supplémentaire pourrait augmenter le coût de développement.

I.3.3 Domaines d'application des RCSF

La taille de plus en plus réduite des micro-capteurs, leur coût de plus en plus faible, la large gamme des types de capteurs disponibles (thermique, optique, de vibrations, etc.) ainsi que le support de communication sans fil utilisé, permettent aux réseaux de capteurs d'envahir rapidement plusieurs domaines d'applications [15]. En effet, ces réseaux se révèlent très utiles dans de nombreuses applications lorsqu'il s'agit de collecter et de traiter des informations provenant de l'environnement. Parmi ces domaines, nous citons les domaines: militaire, environnemental, écologique, domestique, de santé, de sécurité, etc. Dans ce qui suit, nous présentons quelques exemples d'applications potentielles dans ces différents domaines.

a) Applications militaires

Le déploiement rapide, l'auto-configuration et la tolérance aux pannes des réseaux de capteurs sont des caractéristiques qui font de ce type de réseaux un outil appréciable dans un tel domaine. Dans les applications militaires, les capteurs peuvent être déployés dans des endroits stratégiques afin de surveiller toutes les activités des forces ennemies ou d'analyser le terrain avant d'y envoyer des troupes ou servent à la détection d'agents chimiques, biologiques ou de radiations.

b) Application médicales

Le champ de contrôle de santé représente un grand marché pour les RCSF qui a tendance à croître très rapidement. Ces réseaux de capteurs peuvent être utilisés pour assurer une surveillance permanente des organes vitaux de l'être humain (surveillance de la glycémie, détection de cancers, ..) grâce à des micro-capteurs qui pourront être avalés ou implantés sous la peau des patients. Ils peuvent aussi faciliter le diagnostic de quelques maladies en effectuant des mesures physiologiques telles que : la tension artérielle, le rythme cardiaque, ... à l'aide des capteurs ayant chacun une tâche bien particulière.

c) Applications domestiques

Avec le développement technologique, les capteurs peuvent être embarqués dans des appareils, tels que les aspirateurs, les fours à micro-ondes, les réfrigérateurs, les magnétoscopes, etc. Ces capteurs embarqués peuvent interagir entre eux et avec un réseau externe via Internet pour permettre à un utilisateur de contrôler les appareils domestiques localement ou à distance. En les plaçant, sur le plafond ou dans le mur, on peut économiser l'énergie en gérant l'éclairage ou le chauffage en fonction de la localisation des personnes et seulement si c'est nécessaire.

d) Applications environnementales

Les micro-capteurs dispersés dans des zones hostiles peuvent permettre de détecter des incendies, surveiller des catastrophes naturelles (inondations, séismes, éruptions volcaniques), surveiller des phénomènes météorologiques, de détecter de la pollution (taux de CO_2).

e) Applications écologiques

L'intégration de plusieurs micro-capteurs dans le système de climatisation et de chauffage des immeubles. Ainsi, la climatisation ou le chauffage ne sont déclenchés qu'aux endroits où il y a des personnes présentes et seulement si c'est nécessaire. Le système distribué peut aussi maintenir une température homogène dans les pièces. Ce type d'applications pourrait réduire la consommation de l'énergie dans le monde.

f) Agriculture de précision

Les RCSF sont capables d'apporter des bénéfices considérables au domaine d'agriculture, grâce à leur habilité de surveiller le taux d'humidité du sol et par la suite minimiser la quantité d'eau nécessaire à l'irrigation.

g) Application industrielles

L'intégration des micro-capteurs dans le domaine de l'industrie permet de suivre le procédé de production à partir des matières premières jusqu'au produit final livré et détecter les produits présentant des défauts.

La figure 4 montre quelques domaines d'application cités précédemment.

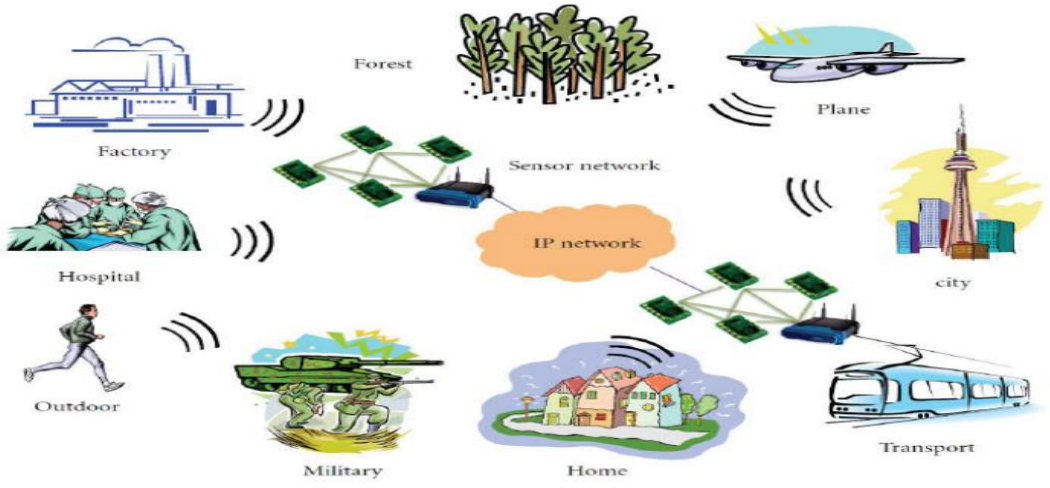

Figure 4: Quelques domaines d'applications pour les RCSF

I.3.4 Caractéristiques des réseaux de capteurs

Parmi les caractéristiques les plus importantes d'un réseau de capteurs:

- **Absence d'infrastructure** : les réseaux de capteurs se distinguent des autres réseaux par la propriété d'absence d'infrastructure préexistante et de tout genre d'administration centralisée.
- **Taille importante** : un réseau de capteurs peut contenir des milliers de nœuds.
- **Interférences** : les liens radio ne sont pas isolés, deux transmissions simultanées sur une même fréquence, ou utilisant des fréquences proches, peuvent interférer.
- **Topologie dynamique**: les capteurs peuvent être attachés à des objets mobiles qui se déplacent d'une façon libre et arbitraire rendant ainsi la topologie du réseau fréquemment changeante.
- **Sécurité physique limitée :** les RCSF font l'objet à des attaques comparativement aux réseaux filaires classiques.
- **Bande passante limitée** : une des caractéristiques primordiales des réseaux basés sur la communication sans fil est l'utilisation d'un médium de communication partagé. Ce partage fait que la bande passante réservée à un nœud est limitée.
- **Contrainte d'énergie** : l'énergie est considérée comme une ressource précieuse dans les RCSF puisque les capteurs sont dotés de faibles ressources (des piles ou des batteries). Afin de prolonger la durée de vie du réseau, une minimisation des dépenses énergétiques est exigée chez chaque

I.4 Facteurs et contraintes de conception d'un RCSF

Les principaux facteurs et contraintes influençant l'architecture des réseaux de capteurs peuvent être résumés comme suit [16]:

- **La tolérance aux pannes:** la tolérance aux pannes est la capacité de maintenir les fonctionnalités du réseau en présence de pannes. La fiabilité des RCSF est affectée par des défauts qui se produisent à cause de diverses raisons telles que le mauvais fonctionnement du matériel ou à cause d'un manque d'énergie.
- **Le passage à l'échelle :** la plupart des protocoles conçus pour les RCSF ont visé au début des réseaux de petite taille. Cependant, ces protocoles peuvent voir leurs performances se

dégradent quand le nombre de nœuds augmente. Ce facteur est connu sous le nom "passage à l'échelle".

- **La topologie de réseau:** le déploiement d'un grand nombre de nœuds nécessite une maintenance de la topologie. Cette maintenance consiste en trois phases : déploiement, post-déploiement (les capteurs peuvent bouger, ne plus fonctionner,...) et redéploiement de nœuds additionnels.
- **Les contraintes matérielles:** la principale contrainte matérielle est la taille du capteur. Les autres contraintes sont la consommation d'énergie qui doit être moindre pour que le réseau survive le plus longtemps possible, qu'il s'adapte aux différents environnements (fortes chaleurs, eau,..), qu'il soit autonome et très résistant vu qu'il est souvent déployé dans des environnements hostiles.
- **Les médias de transmission:** dans un réseau de capteurs, les nœuds sont reliés par une architecture sans fil. Pour permettre des opérations sur ces réseaux dans le monde entier, le média de transmission doit être standardisé. On utilise le plus souvent l'infrarouge, le Bluetooth [17] et les communications radio Zig Bee [18].
- **La consommation d'énergie:** la durée de vie d'un capteur dépend grandement de la durée de vie de sa batterie puisque le remplacement de la batterie est impossible et surtout si le capteur est déployé dans des zones hostiles. Dans un réseau de capteurs, chaque nœud collecte des données et les transmet à la station de base. Le dysfonctionnement de quelques nœuds nécessite un changement de la topologie du réseau et l'établissement de nouveaux mécanismes de routage. Ces opérations sont gourmandes en énergie, c'est pour cette raison que les recherches actuelles se concentrent principalement sur les moyens de réduire cette consommation [19].

I.5 Conclusion

Les réseaux de capteurs sans fil ne cessent de prendre une place très appréciée au sein de la communauté de la recherche vu leur déploiement assez simple et leurs applications qui se diversifient chaque jour pour élargir leurs horizons. Initialement, réservés pour les applications militaires aujourd'hui, les RCSF ont réussi à conquérir d'autres domaines civils plus larges et plus pratiques changeant le quotidien des êtres humains.

Dans ce chapitre, on a essayé de présenter brièvement la notion de capteurs, leurs caractéristiques. On a présenté également les réseaux de capteurs, leurs architectures, leur

classification, leurs caractéristiques, leurs domaines d'applications, et à la fin leurs contraintes de conception.

Dans le chapitre qui suit, on présente les protocoles de routage dans les réseaux de capteurs et on met l'accent sur le protocole de routage LEACH.

Chapitre II

Les protocoles de routage dans les réseaux de capteurs sans fil

Chapitre II
Les protocoles de routage dans les RCSF

II.1 Introduction

L'objectif principal d'un protocole de routage pour un réseau de capteurs sans fil est l'établissement correct et efficace de routes entre une paire de nœuds afin que les messages puissent être acheminés. Le protocole de routage permet aux nœuds de se connecter directement les uns aux autres pour relayer les messages par des sauts multiples et de transmettre les données vers la station de base.

Dans ce chapitre, on présente une taxonomie des principaux protocoles de routage dans les réseaux RCSF. Puis, on met l'accent sur le protocole LEACH qui fait l'objet d'une étude précise dans un environnement fortement perturbé (environnement non-idéal).

II.2 Classification des protocoles de routage pour les RCSF

Récemment, les protocoles de routage pour les RCSFs ont été largement étudiés. Les protocoles proposés présentent les points communs et donc peuvent être classifiés suivant un certain nombre de critères. La figure récapitule une classification qui se base sur quatre critères : la topologie du réseau, le mode d'établissement des chemins, le paradigme de communication et le mode de fonctionnement du protocole.

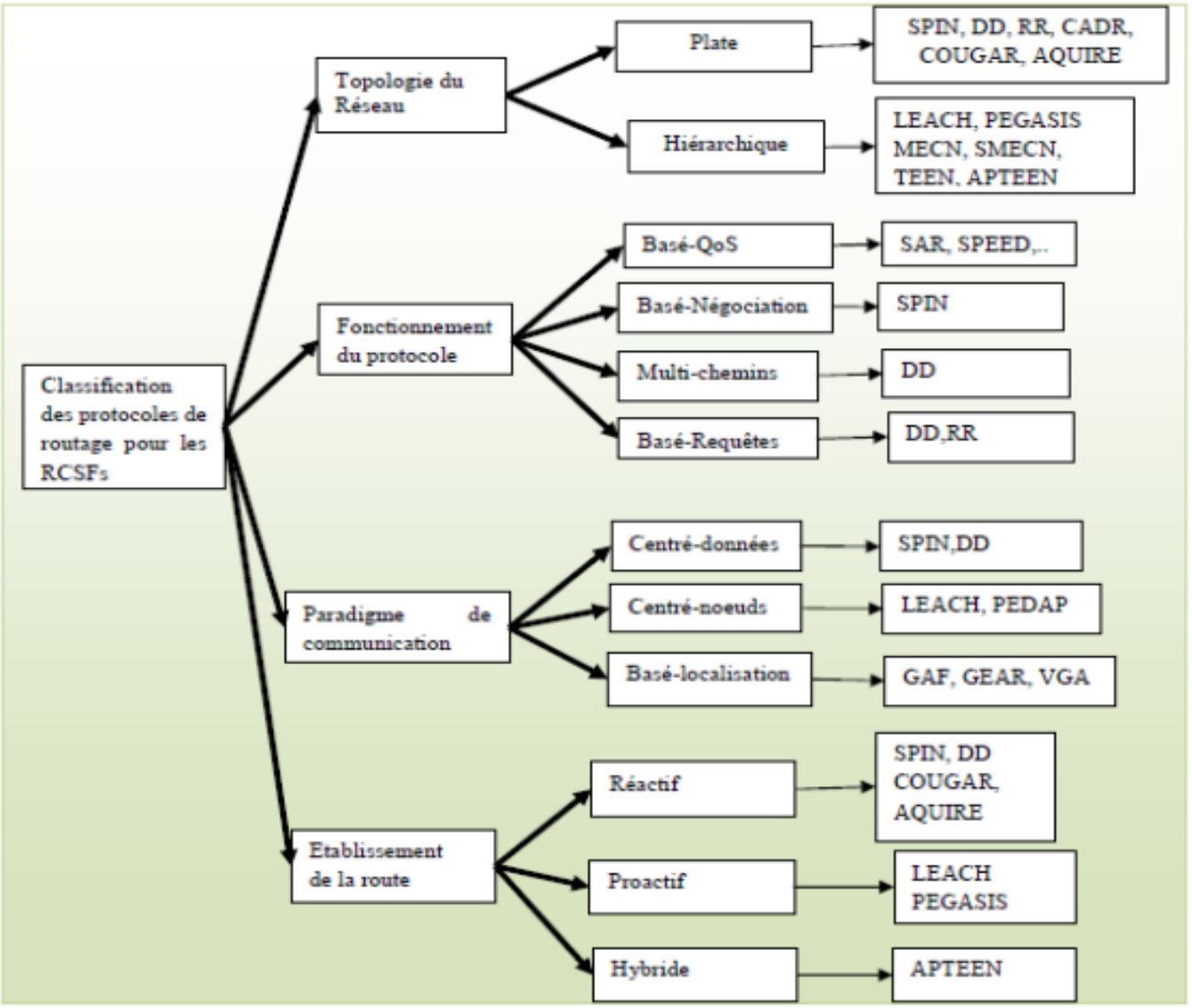

Figure 5: Classification des protocoles de routage pour les RCSF [6]

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 [20] : la topologie plate et la topologie hiérarchique.

a) Topologie plate

Dans une topologie plate, tous les nœuds capteurs ont le même rôle et collaborent entre eux pour accomplir la tâche de routage. Les réseaux plats sont caractérisés par : la simplicité des protocoles de routage, une grande tolérance aux pannes ainsi qu'une habilité à construire de nouveaux chemins suite aux changements de topologie. Cependant, Les nœuds proches de la station de base participent plus que les autres dans le processus de routage. De plus, les performances de ces réseaux se dégradent quand le nombre de nœuds augmente.

b) Topologie hiérarchique

Dans une topologie hiérarchique, les nœuds ont des rôles différents. En effet, certains nœuds sont sélectionnés pour exécuter des fonctions particulières telles que l'agrégation de données et l'acheminement des données. Une des méthodes les plus utilisées dans cette topologie est le clustering. Il consiste en un partitionnement du réseau en clusters. Un cluster est constitué d'un clusterhead et de ses membres. Selon l'architecture adoptée, les membres peuvent être des voisins directs du clusterhead ou pas. Cette topologie présente beaucoup d'avantages tels que l'agrégation des données collectées réalisée par les clusterheads ainsi qu'une grande scalabilité. Son inconvénient majeur est la surcharge des clusterheads qui induit un déséquilibre de la consommation d'énergie dans le réseau. Pour remédier à ce problème, les clusterheads peuvent être sélectionnés en fonction de leurs capacités énergétiques et leur élection devrait se faire périodiquement.

II.2.2 Selon le paradigme de communication

Le paradigme de communication détermine la manière dont les nœuds sont interrogés. Dans les RCSFs, il existe trois paradigmes de communication [21]: centré-nœuds, centré-données et basé sur la localisation.

a) Centré-nœuds (Node-centric)

Ce paradigme est celui employé dans les réseaux conventionnels, où il est nécessaire de connaître et d'identifier les nœuds communicants comme l'adresse IP dans l'environnement de Contiki[8] et l'identifiant affecté par l'usager comme dans TinyOs [7]. Par ailleurs, dans les réseaux de capteurs, un routage basé sur une identification individuelle des nœuds ne reflète pas l'usage réel du réseau puisque les nœuds sont déployés en grand nombre et d'une manière aléatoire. D'où un autre paradigme a été introduit : data-centric. Néanmoins, le paradigme node-centric reste toujours performant dans le cas des réseaux de petite taille et nécessitent une interrogation individuelle des capteurs.

b) Centré-données (Data-centric)

Ce paradigme suppose qu'il est difficile d'avoir des identifiants comme les adresses MAC ou IP pour pouvoir communiquer entre les nœuds capteurs. Ainsi, le routage ne se fait pas en fonction d'une adresse de destination, mais suivant les données disponibles au niveau des capteurs. Ces données seront propagées de proche en proche pour y arriver à la station de base.

c) Basé-localisation (location-based)

Dans cette technique, les décisions de routage sont établies selon la position des nœuds. La distance entre les nœuds voisins peut être estimée sur la base de la puissance du signal arrivé (RSSI[2]). Un tel type de routage nécessite que les nœuds aient connaissance de leurs positions géographiques. Par conséquent, ce type de mécanismes nécessite un déploiement d'une solution de positionnement, dont le degré de précision requis dépend de l'application ciblée. L'utilisation du GPS reste trop coûteuse pour un RCSF puisque ce dernier consomme plus d'énergie. Néanmoins, d'autres méthodes de localisation et de positionnement des capteurs ont été développées comme par exemple la triangulation [22].

II.2.3 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 être classifiés en quatre catégories : routage basé sur la qualité de service "QoS[3]", routage basé sur les requêtes (query-based routing), routage multi-chemins (Multi-path routing), et routage basé sur la négociation (Negociation based routing) [20].

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, le chemin alternatif sera emprunté. Pour cette raison, certains protocoles construisent plusieurs chemins indépendants, c-à-d ils ne partagent qu'un nombre réduit de nœuds. Malgré leur grande tolérance aux pannes, ces protocoles requièrent plus de ressources énergétiques et plus de messages de contrôle.

b) Routage basé sur les requêtes

Dans ce type de routage, la station de base 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 tel que SQL[4]. Les nœuds qui détiennent les données requises doivent les envoyer au nœud demandeur à travers le chemin inverse de la requête. Les requêtes émises par la station de base peuvent aussi être ciblées sur des régions spécifiques du réseau.

[2] RSSI : Received Signal Strength Indication
[3] QoS : Quality of Service.
[4] SQL : Strutured Query Language

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 leurs données en échangeant des paquets de signalisation 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. Ceci permet de minimiser la consommation d'énergie dans le RCSF.

d) Routage basé sur la qualité de service

Dans les protocoles de routage basés sur 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, la latence, la consommation de l'énergie, largeur de bande passante, etc. Les protocoles de cette approche sont très recommandés pour les applications orientées surveillance.

II.2.4 Selon le mode l'établissement des chemins

Suivant la manière de création et de maintien des chemins pendant le routage, nous distinguons trois catégories de protocoles de routage : les protocoles proactifs, les protocoles réactifs et les protocoles hybrides [23].

a) Les protocoles proactifs

Ces protocoles utilisent l'échange régulier de messages de contrôle pour maintenir au niveau de chaque nœud des tables de routage vers toute destination atteignable depuis celui-ci. Ces tables sont maintenues mêmes quand les routes ne sont pas utilisées. Cette approche permet de disposer d'une route vers chaque destination immédiatement au moment où un paquet doit être envoyé. Les protocoles proactifs sont adaptés aux applications qui nécessitent un prélèvement périodique des données. Par conséquent, les capteurs peuvent se mettre en veille pendant les périodes d'inactivité, et n'enclencher leur dispositif de capture qu'aux instants lorsqu'ils basculent en mode actif.

b) Les protocoles réactifs

Ces protocoles créent les routes à la demande. Lorsqu'un nœud a besoin d'une route, une procédure de découverte globale est déclenchée. Cette procédure s'achève par la découverte de la route ou lorsque toutes les permutations de routes possibles ont été examinées. La route

trouvée est maintenue par une procédure de maintenance de routes jusqu'a ce que la destination soit inaccessible à partir du nœud source ou que le nœud source n'aura plus besoin de cette route.

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.3 LEACH et ses variantes

II.3.1 Le protocole LEACH

LEACH (Low Energy Adaptive Clustering Hierarch) [1,2] est considéré parmi les meilleurs protocoles qui sont performants en termes d'énergie. Il basé sur le clustering adaptatif, qui utilise la rotation randomisée des clusterheads pour distribuer équitablement la charge d'énergie entre les nœuds capteurs dans le réseau. LEACH est fondé sur deux hypothèses de base :

- La station de base est fixe et est placée loin des capteurs (en dehors de la zone de déploiement des capteurs),
- Tous les nœuds du réseau sont homogènes en termes d'énergie à l'exception de la station de base qui est considérée comme un nœud qui a plus d'énergie.

LEACH permet de former des clusters des nœuds capteurs selon la puissance du signal reçu (RSSI) et d'utiliser les clusterheads comme des routeurs pour acheminer les données directement à la station de base. Dans LEACH, on assiste aux tâches suivantes :

- Coordination et contrôle localisés pour l'initialisation et le traitement au niveau des clusters.
- Rotation randomisée des clusterheads pour que ces derniers n'épuisent pas leurs batteries rapidement.
- Agrégation des données par les clusterheads. Les CHs agrègent les données envoyés par les nœuds membres appartenant à leurs clusters correspondants, et envoient le paquet résultant de cette opération d'agrégation à la station de base.

La figure 6 montre le paradigme de routage adopté dans LEACH.

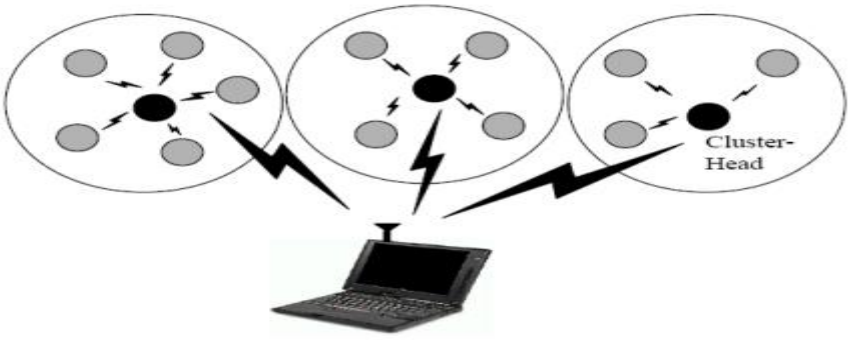

Figure 6: Paradigme de routage dans LEACH

a) La phase de formation de clusters

LEACH fonctionne selon des cycles dont la durée est constante. Chaque cycle commence par une phase d'initialisation (set-up phase) suivie d'une phase de transmission de données (steady phase). Dans la première phase, les clusters sont organisés et les CHs sont sélectionnés. Cette élection est basée sur le pourcentage désiré de CHs et le nombre de périodes au cours duquel un nœud a pris le rôle de CH. Ainsi, un nœud (ID_i) prend une valeur aléatoire entre 0 et 1. Si cette valeur est inférieure au seuil $T(ID_i)$ calculé selon l'équation (1)**,** le nœud se déclare CH pour le cycle courant.

$$T(ID_i) = \begin{cases} \dfrac{P}{1 - P \times \left(r \bmod \frac{1}{P}\right)} & Si\ (ID_i \in G) \\ 0 & Sinon \end{cases} \qquad (1)$$

tels que :

- **P** : représente le pourcentage désiré de CHs.
- **r** : le numéro de la période courante.
- **G** : ensemble des nœuds qui ont été sélectionnés comme CH durant les dernières (1/P) périodes.

Chaque CH élu émet un message de signalisation au reste des nœuds dans le réseau. Dans ce message il annonce qu'il est élu CH pour cette période. Tout nœud non affilié recevant ce message envoie une demande d'adhésion au clusterhead correspondant. Le choix du clusterhead est basé sur la puissance du signal reçu.

Après la formation des clusters, chaque CH crée un ordonnancement TDMA[5] et assigne à chaque nœud membre du cluster un slot de temps dans lequel ce dernier pourra envoyer les données collectées. Ce programme est émis à tous les nœuds du cluster.

L'algorithme LEACH utilise la technique de multiplexage temporel TDMA comme méthode d'accès au médium. Chaque nœud utilise la totalité de la bande passante allouée par le système de transmission durant son slot de temps. En fait, chaque CH agit comme un centre de commande local pour coordonner les transmissions des données dans son cluster. Etant donné que chaque nœud connaît d'avance le slot de temps qu'il va occuper, cela permet alors au nœud de passer à l'état "endormi" durant les slots inactifs. Ainsi, la perte d'énergie due aux états de sur écoute (overhearing) et d'écoute passive (idle) est évitée.

En outre, chaque CH choisit aléatoirement un code dans une liste de codes de propagation CDMA[6] pour éviter les interférences avec les CHs lors de l'envoi des données agrégées à la station de base. La figure 7 résume la phase de formation de clusters dans LEACH.

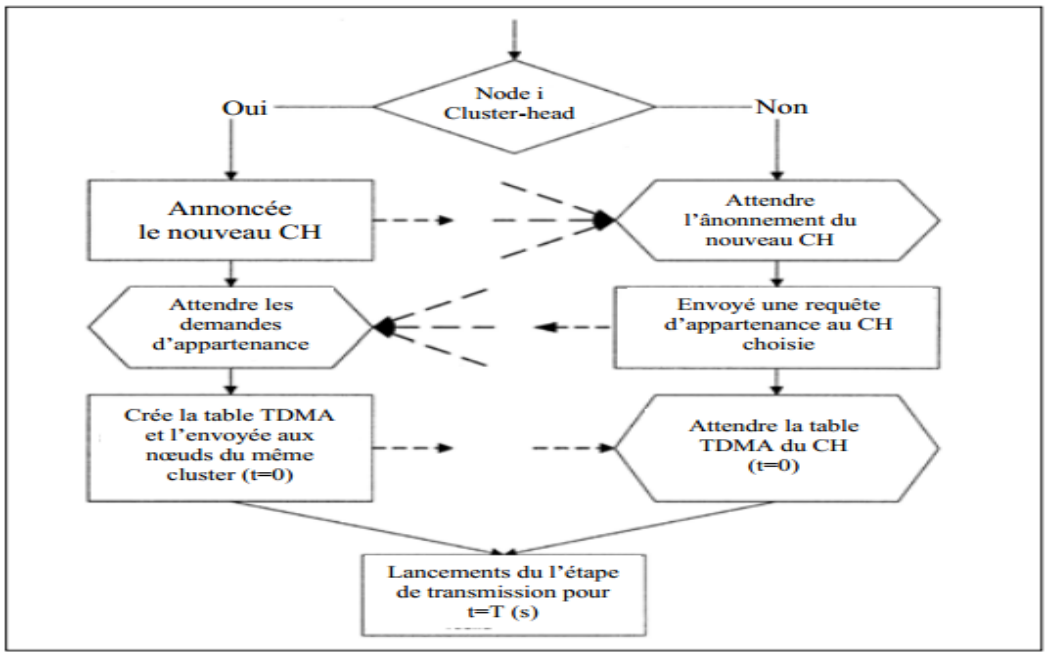

Figure 7: La phase de formation de clusters

[5] TDMA : Time Division Multiplexed Access
[6] CDMA : Code division multiple access

b) La phase de transfert de données

Dans la deuxième phase (steady phase), le transfert des données actuelles à la station de base aura lieu. La durée de la deuxième phase est plus longue que celle de la première phase comme montre la figure 8. Cependant, la collection de données est centralisée et est exécutée périodiquement. Par conséquent, ce protocole est plus performant en termes d'énergie.

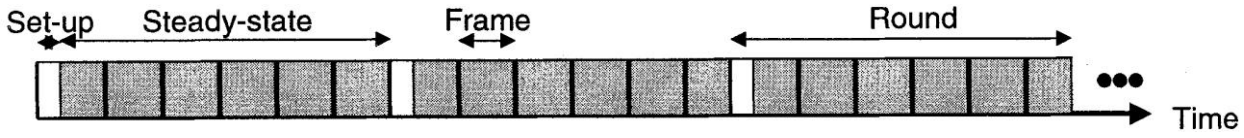

Figure 8: Les temps de chacune des deux phases dans une période [2]

Pour ne pas épuiser les batteries des CHs, une rotation randomisée du rôle de CH est introduite dans LEACH de sorte que la dissipation uniforme d'énergie dans le réseau de capteurs soit obtenue. Les auteurs ont trouvé, en se basant sur leur modèle de simulation, que seulement 5% des nœuds ont besoin d'agir comme clusterheads.

Vu que le protocole LEACH ne garantit pas la distribution équitable des CHs dans le réseau, une version centralisée de LEACH appelée LEACH-C [2] est donc proposée. Cette dernière permet de déterminer, à partir de la position exacte des nœuds, la configuration optimale pour minimiser l'énergie dépensée.

II.3.2 Le protocole LEACH-C

LEACH-C est une variante de LEACH où les clusters sont formés d'une manière centralisée par la station de base. LEACH-C utilise la même étape de transmission que la version originale de LEACH. Durant la phase de formation de clusters la station de base reçoit de chaque nœud des informations concernant leur localisation, et leur réserve d'énergie. Ensuite, elle exécute un algorithme de formation de clusters en désignant les CHs selon leurs positions dans le réseau. Néanmoins, LEACH-C n'est pas adaptée aux réseaux denses.

II.3.3 Le protocole LEACH-F

Le protocole LEACH-F a été proposé pour minimiser le coût de l'overhead lors de la formation des clusters. Ainsi, dans LEACH-F, les clusters sont formés une seule fois et le rôle de clusterhead tourne parmi les nœuds membres du même cluster.

L'avantage de LEACH-F est que, une fois que les clusters sont formés, aucune autre phase de formation de clusters n'aura lieu. LEACH-F utilise le même algorithme centralisé de

formation de clusters que LEACH-C. Les clusters fixes dans LEACH-F ne permettent pas à de nouveaux nœuds d'être ajoutés au système et n'ajustent pas leur comportement quand un ou plusieurs nœuds cessent de fonctionner.

II.4 Conclusion

Dans ce chapitre, on a présenté une classification des principaux protocoles de routage dans RCSF et on a mis l'accent sur le protocole LEACH qu'on va l'évaluer dans un environnement non-idéal.

Chapitre II

Evaluation de LEACH dans un environnement non-idéal

Chapitre III
Evaluation de LEACH dans un environnement non-idéal

III.1 Introduction

La conception de la plupart des protocoles de routage était basée sur une couche physique idéale modélisée par le graphe du disque unitaire [24]. Cependant, bien que ce modèle soit couramment utilisé, il ne peut pas être considéré comme un modèle réaliste car il suppose que les messages envoyés sont toujours reçus sans erreur si la distance entre l'émetteur et le récepteur est inférieure ou égale à la portée de transmission de l'émetteur. Cette hypothèse ne tient pas compte des fluctuations aléatoires du signal radio, qui a un impact significatif sur les transmissions à cause des erreurs générées par les fluctuations dans les messages échangés entre les nœuds. Par conséquent, il est intéressant d'étudier le comportement de ces protocoles de routage dans un environnement réaliste pour illustrer l'effet des fluctuations du signal radio sur les performances de ces protocoles. Dans ce contexte, nous avons choisi de mettre l'accent sur le protocole LEACH [1] car il fournit de bons résultats en utilisant une couche physique idéale.

Dans ce chapitre, nous avons utilisé un modèle de propagation présentant une irrégularité de la radio [25] pour un environnement de simulation réaliste et analyser les performances du protocole LEACH par ce modèle. Le modèle considéré prend en compte les fluctuations du signal radio et pourrait donc être plus réaliste que le modèle du disque unitaire. De plus, il calcule la probabilité de réception sans erreur entre les nœuds communicants d'une manière aléatoire indépendamment de la distance et fonction du degré d'irrégularité DOI.

Avant d'évaluer LEACH dans un environnement non-idéal, on présente tout d'abord le modèle radio utilisé et les outils logiciels nécessaires pour sa mise en œuvre tels que TinyOS [7], NesC et le simulateur TOSSIM [26]. Puis, on présente l'implémentation et évaluation de LEACH et les critères de performance pour illustrer l'impact de la fluctuation de la radio.

III.2 Préliminaires et hypothèses

Dans cette section, on présente les causes de l'irrégularité de la radio et le modèle sous-jacent à l'irrégularité de la radio.

III.2.1 Les causes de l'irrégularité de la radio

L'irrégularité de la radio est un phénomène non négligeable dans les communications sans fil. Il provoque des interférences et des liens asymétriques dans les couches supérieures. Par conséquent, il pourrait directement ou indirectement affecter les performances de ces couches en particulier la couche réseau.

L'irrégularité de la radio est causée par deux types de facteurs : les dispositifs et les médiums de propagation [27]. Les caractéristiques du dispositif impliquent le type d'antennes (directionnelle ou omnidirectionnelle), la puissance d'émission, les gains de l'antenne (au niveau de l'émetteur et du récepteur), la sensibilité du récepteur et le rapport signal-sur-bruit (SNR). Les propriétés du médium comprennent le type de support, le bruit et certains d'autres facteurs environnementaux, comme la température et les obstacles dans les médiums de propagation.

En général, l'irrégularité de la radio est causée par l'anisotropie des propriétés du milieu de propagation et l'hétérogénéité des propriétés des dispositifs. Parmi tous ces facteurs, on trouve les pertes de chemins non isotropes et les différences dans la puissance du signal d'émission.

III.2.2 Les modèles radio

La plupart des protocoles de routage dans les RCSF sont basés sur le modèle du disque unitaire (UDG) [24] où la communication entre deux nœuds dépend de la distance qui les sépare. Si la distance est inférieure à la portée de transmission, les messages échangés entre les nœuds sont reçus sans erreur. Toutefois, la propagation du signal radio pourrait être affectée par plusieurs facteurs qui contribuent à la dégradation de sa qualité. En plus, les effets de ces facteurs sont encore plus significatifs sur la propagation du signal radio qui a une faible puissance.

Par ailleurs, les liens radio dans les RCSF sont souvent imprévisibles [28]. La qualité de ces liens varie dans le temps [29] et dans l'espace [27] et la connectivité est généralement asymétrique [30]. Dans ce contexte, on distingue trois modèles de propagation illustrant l'irrégularité de la radio : Le modèle radio isotrope, le modèle de l'irrégularité de la radio et le modèle LogNormal Shadowing.

Dans ce qui suit on met l'accent seulement sur les deux premiers modèles qu'on les a utilisés pour évaluer les performances de LEACH dans un environnement non-idéal.

a) Le modèle radio isotrope

Dans le modèle radio isotrope, la puissance du signal reçu est généralement représentée par l'équation (2).

$$Received\ Signal\ Strength = Sending\ Power - PathLoss + Fading \qquad (2)$$

La puissance d'émission (Sending Power) d'un nœud est déterminée par l'état de la batterie et le type d'émetteur, d'amplificateur et d'antenne. La perte de chemin (PathLoss) décrit la perte d'énergie du signal lorsqu'il est transmis au récepteur. Plusieurs modèles sont utilisés pour estimer la perte de chemin, tels que le modèle de propagation dans l'espace libre (the freespace propagation model), le modèle de propagation à deux rayons (the two-ray model) et le modèle Hata [31]. Ces modèles sont dits isotropes puisque dans ces modèles l'atténuation du signal est la même dans toutes les directions.

b) Le modèle de l'irrégularité de la radio

Le modèle de l'irrégularité de la radio est une extension du modèle radio isotrope. Il améliore le modèle radio isotrope par approximation des trois principales propriétés du signal radio : la non-isotropie, la variation continue et l'hétérogénéité.

Le modèle de l'irrégularité de la radio est représenté par le modèle DOI (Degree Of Irregularity) [32] qui est utilisé pour désigner l'irrégularité du motif de la radio. La figure 9 illustre la variation de la propagation de la radio en fonction du coefficient DOI. Par exemple, quand DOI est nul le modèle généré correspond au modèle du disque unitaire. On pourra dire que ce modèle correspond bien à un modèle probabiliste.

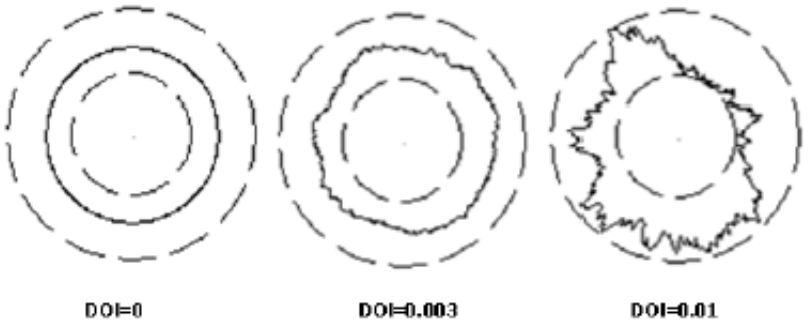

Figure 9: Le modèle de propagation DOI [32]

III.3 Outils logiciels

Dans cette section, on présente les outils logiciels nécessaires pour l'évaluation de LEACH dans un modèle de propagation probabiliste. Tout d'abord, on traduit le fonctionnement de LEACH en langage NesC. Puis, on le simule en utilisant TOSSIM [26].

III.3.1 Le système d'exploitation TinyOs

TinyOS [7] est un système d'exploitation open source pour les réseaux de capteurs, développé par l'université américaine de BERKELEY. Il est prévu pour fonctionner sur diverses plateformes. En effet, TinyOS peut aussi bien être installé sur Windows, Linux, Mac OS ou sur un capteur. Sa conception a été entièrement réalisée en NesC, langage orienté composant syntaxiquement proche du C. Il respecte une architecture basée sur une association de composants, réduisant ainsi la taille du code nécessaire à sa mise en place.

TinyOS s'appuie sur un fonctionnement évènementiel, c'est-à-dire qu'il ne devient actif qu'à l'apparition de certains évènements. La bibliothèque de composants de TinyOS est particulièrement complète qui peut être utilisée pour programmer. TinyOS offre des outils de simulation tels que TinyViz, TOSSIM et PowerTossim [7].

III.3.2 Le langage de programmation NesC

Le langage NesC (**N**etwork **e**mbedded **s**ystem **C**) est un dialecte de C basé sur les composants [33]. Il est orienté composants pour satisfaire les exigences des systèmes embarqués. De plus, il supporte un modèle de programmation qui agrège l'administration des communications, les concurrences provoquant les tâches et les évènements ainsi que la capacité de réagir par rapport à ces évènements.

NesC réalise aussi une optimisation dans la compilation du programme, en détectant les carrières possibles de données qui peuvent produire d es modifications concurrentes à une même section de mémoire (concurrence d'accès mémoire entre threads), et quand au moins l'un des accès est en écriture. NesC simplifie aussi le développement d'applications et réduit la taille du code.

a) Les Fichiers dans NesC

Les fichiers dans NesC sont classés en trois types: Interfaces, Modules et Configurations.

- **Interface**: Ce type de fichiers déclare les services fournis et les services qui seront utilisés. Il définit les interactions entre deux composants qui se trouvent dans le répertoire **/tos/interface**.

  Exemple: **StdControl.nc**

- **Module**: Le type Module contient le code de l'application, en mettant en œuvre une ou plusieurs interfaces.

  Exemple **: BlinkM.nc**

- **Configuration**: Dans ces fichiers est déclarée la manière d'unir les différents composants et comment effectuer le contrôle des flux.

  Exemple**: Blink.nc**

b) Eléments d'un programme NesC

- **Composants :** TinyOS définit un nombre important de concepts qui sont exprimés dans NesC. D'abord, les applications NesC sont construites par des composants avec des interfaces bidirectionnelles [34]. Aussi, NesC définit un modèle basé sur les tâches et les captures d'évènements matériels, et détecte des éclatements d'information pendant la compilation. Un Composant implémente des interfaces utilisées par d'autres composants pour communiquer avec ce composant.

  Un composant, du point de vue programmation, est constitué de plusieurs sections et l'ensemble de toutes ces sections donne lieu à la création de ce composant.

- **Implémentations :** Cette partie définit les connexions entre les différents composants qu'utilise l'application. Dans une implémentation sont donc principalement définis les composants qui fournissent les interfaces à notre application qui sont souvent des composants primitifs. Habituellement, nous devons utiliser les interfaces que nous fournissent d'autres composants, primitifs ou non primitifs, et en définitive pour chacune de ces interfaces, que nous utiliserons dans la création de notre composant. Nous devons obligatoirement définir des relations avec les composants qui fournissent ces interfaces. Le processus définissant ces relations s'appelle (wiring).

Exemple:

```
Implémentation {
components Main, MonAppliM;
```

```
Main.StdControl - >MonAppliM.StdControl ;
}
```

- **Configuration :** C'est à cet endroit que l'on déclare les autres composants dont se servira l'application. Cette possibilité offerte par le langage permet de faire de la programmation modulaire et de réutiliser des composants préalablement définis.

  La structure de la partie Configurations est la même que celle de la partie Implémentation.

Exemple:

```
Configuration MonAppli {
implementation {
components GenericComm ,TimerC ;
TimerC . StdControl ->GenericComm . StdControl ;
    }
}
```

- **Module :** Cette partie du code est généralement plus étendue et c'est dans celle -ci que l'on programme réellement le comportement qu'on souhaite voir réalisé par l'application.

  Elle est divisée en trois sous-sections: Uses, Providers et Implementation.

Exemple:

```
Module MonAppliM{
Provides{…}
uses{…}}
implementation{…}
```

La première sous-section (provides), indique au compilateur les interfaces fournies par le composant. Par exemple, si le composant est une application, on doit fournir au moins l'interface StdControl.

```
provides {
interface interface_que_nous _fournissons;
}
```

La sous-section (uses) informe le compilateur que nous allons utiliser une interface. D'où, on pourra donc effectuer des appels aux méthodes de cette interface. Pour ce faire, nous avons besoin de respecter quelques règles. Ainsi, si nous utilisons une interface, il nous faut avoir dans la section (implementation) un lien (''wiring'') reliant cette interface avec un composant qui la fournit. Finalement, utiliser une interface oblige implicitement à gérer les évènements pouvant se produire du fait d'avoir utilisé cette interface précise.

```
uses {
interface interface que nous utilisons ;
}
```

La sous-section (implementation), est celle qui contiendra toutes les méthodes nécessaires pour fournir le comportement souhaité à ce composant ou à une application. Cette sous-section doit contenir au moins:

- Les variables globales qui sont utilisées par l'application
- Les fonctions que doivent mettre en œuvre pour les interfaces que nous fournissons.
- Les évènements que doivent mettre en œuvre venant des interfaces que nous utilisons.

c) Types de données utilisés par NesC

Les types de données qui peuvent être utilisés en NesC sont tous ceux que fournit le langage C standard plus quelques autres qui n'apportent pas de puissance de calcul mais qui sont très utiles pour la construction de paquets puisqu'ils fournissent à l'utilisateur le nombre de bits qu'ils occupent. Ceci est important au moment de la transmission des informations par l'intermédiaire des ondes radio.

Quelques exemples de ces types additionnels :

- **uint16_t** : entier non signé sur 16 bits.
- **uint8_t :** entier non signé sur 8 bits.
- **Result_t** : utilisé pour savoir si une fonction a été exécutée e avec succès ou non, c'est comme un booléen mais avec les valeurs de retour SUCCESS et FAIL.
- **bool :** valeur booléenne qui peut être TRUE ou FALSE.

En NesC, il est possible de faire une utilisation dynamique de la mémoire mais ce n'est pas très recommandé à moins que cela ne soit absolument nécessaire. Pour pouvoir l'utiliser il

existe un composant particulier appelé MemAlloc qui permet une gestion dynamique de la mémoire.

d) Types de fonctions en NesC

En NesC les fonctions peuvent être de types très variés. Il y a d'abord les fonctions classiques avec la même sémantique qu'en C et la façon de les invoquer est aussi la même qu'en C. Il y a aussi des types supplémentaires de fonctions: task, event, command. Les fonctions (command) sont principalement des fonctions qui sont exécutées de manière synchrone, c'est-à-dire que lorsqu'elles sont appelées elles sont exécutées immédiatement. La manière d'appeler ce genre de fonction est:

```
Call interface.nomFonction
```

Les fonctions (task) sont des fonctions qui sont exécutées dans l'application, utilisant la même philosophie que les threads. C'est principalement une fonction normale qui est invoquée de la manière suivante:

```
post interface.nomTache
```

Immédiatement après son invocation, l'exécution du programme qui l'a invoqué se poursuit. Les fonctions (event) sont des fonctions qui sont appelées quand on relèvera un signal dans le système. Ces fonctions possèdent principalement la même philosophie que la programmation orientée événements, de sorte que, lorsque le composant reçoit un événement, on effectue l'invocation de cette fonction. Il existe une méthode pour pouvoir invoquer manuellement ce type de fonctions:

```
signal interface.nomEvenement
```

On utilise donc le mot clef **call** pour déclarer les évènements et le mot clef **post** pour déclarer les tâches.

III.4 Implémentation et évaluation de LEACH

III.4.1 Les fichiers de l'application

Notre application s'articule autour des composants suivants :

- un module, appelé «MHLeachPSM.nc ».
- une configuration, appelée «MHLeachRouter.nc ».

- le fichier d'entête, appelé «MH.h ».

III.4.2 Implémentation de LEACH

Dans notre travail, on applique le modèle probabiliste pour envoyer les données de CH à la station de base.

1/ On génère un nombre aléatoire qui représente la probabilité d'une réception sans erreur entre un CH et la station de base dans l'interface "**AggregerTimer"**:

Si la probabilité>=0.3 alors envoi direct à la station de base.

Sinon le CH implique un autre CH dans cette opération.

On a créé un "**pubTimer**" qui teste si le CH choisi (P) est fiable ou non

Si il est fiable il fait appel à une interface "**probabilite_ReceiveMsg**" qui envoie le paquet à la station de base.

Sinon le paquet est perdu.

2/ On a ajouté un test dans l'interface "**OrdonnecementTimer**" pour le CH:

- Si la probabilité>=0.3 alors :

- Si le CH est en panne il devra être remplacé par un autre membre qui fait partie de son cluster

- Tout d'abord on a créé une liste, appelée "liste1" qui contient tous les CH.

- On a ajouté dans cette liste l'attribut "panne". Cet attribut prend la valeur "TRUE" si le CH est en panne et "FALSE" sinon.

- Si le CH est fiable : les membres envoient les données à leur CH correspondant.

- Si le CH est en panne alors les membres envoient les données à son adjoint qui est désigné.

- Si probabilité<0.3 alors : le paquet est perdu.

III.4.3 Environnement d'exécution du simulateur

On a utilisé le simulateur TOSSIM pour évaluer les performances de LEACH dans un environnement probabiliste. Pour cela, nous avons choisi Tinyos-1.x sous l'émulateur Cygwin.

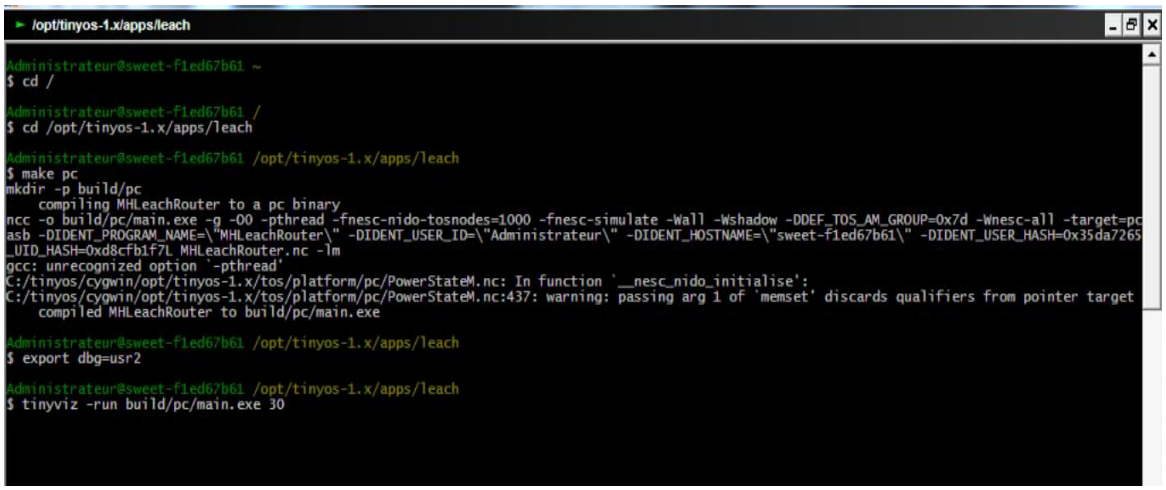

Figure 10: Interface Cygwin

- On accède au fichier home par la commande suivante : **cd /**
- Après, on saisit le chemin suivant: **cd opt/tinyos-1.x/apps/leach** pour accéder à l'application "LEACH".
  - Ensuite, on la compile par la commande : **make pc**. Cette commande est nécessaire pour une simulation sous TOSSIM. Si on utilise des capteurs réels tels que TelosB ou MicaZ, la compilation se fera selon le type du capteur, par exemple make telosb pour un capteur telosb.
  - Enfin on l'exécute par : la commande **export DBG=usr3**, et la commande **build/pc/main.exe –t= "temps de simulation" -p "nbre_de_nœuds" >fichier.txt**

III.4.4 Critères de performances

Pour illustrer l'impact d'un modèle fortement perturbé sur les performances de LEACH, on propose évaluer les métriques suivantes : le nombre de paquets reçus, le nombre de paquets perdus, le taux de paquets perdus et l'énergie dissipée.

Pour cela, nous implémentons deux versions de LEACH. La première correspond à la version originale de LEACH et la deuxième est une amélioration de la première pour adapter LEACH à un environnement non-idéal.

III.4.5 Paramètres de la simulation

Pour l'évaluation de LEACH, nous avons varié le temps de simulation et la taille du réseau pour voir l'impact de ces deux critères sur les performances de LEACH. On a assigné à une période 100 secondes.

Tableau 2: Paramètres de simulation

Paramètre	Valeur
Temps de simulation	1, 2, 3, 4,5 (périodes)
Zone de déploiement	100 x 100
Le nombre des nœuds	50, 100, 150, 200

III.4.6 Résultats et interprétations

Dans cette partie, on a évalué LEACH dans le modèle probabiliste en faisant varier le nombre des nœuds et les périodes. Dans le premier scénario, on a fixé le nombre de périodes à 5 et on a fait varier le nombre de nœuds alors que dans le deuxième scénario on a fixé le nombre de nœuds à 100 et on a fait varier le nombre de périodes de 1 jusqu'à 5.

a) Le nombre de paquets reçus (5 périodes) :

La quantité d'informations reçues au niveau de la station de base est un critère de performance. La figure 11 illustre le nombre de paquets reçus au niveau de la station de base dans les deux versions.

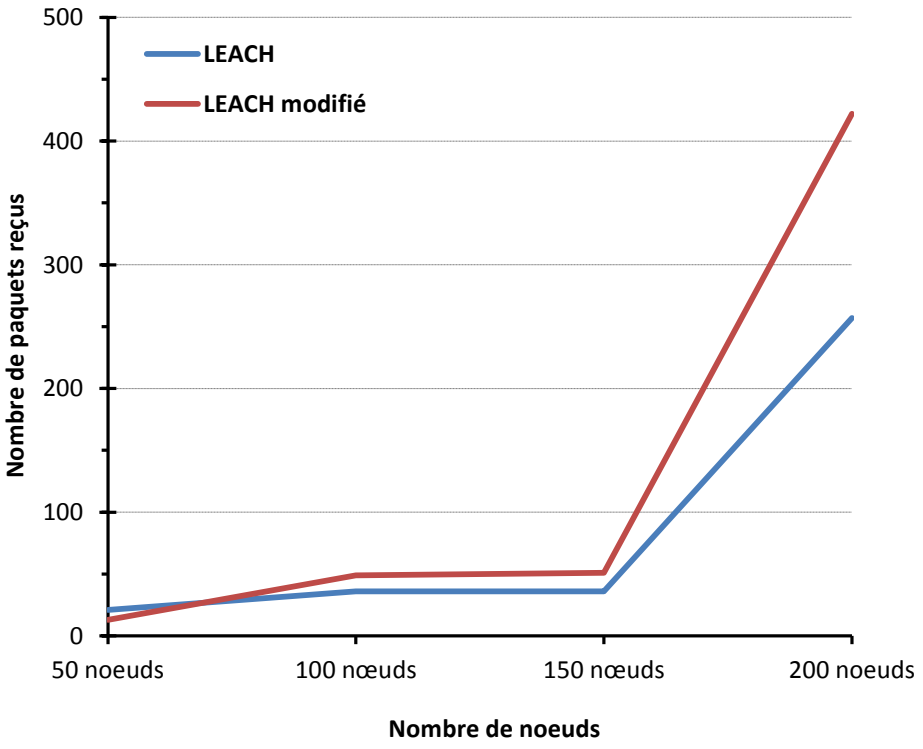

Figure 11: Nombre de paquets reçus vs. nombre de nœuds pendant 5 périodes.

La figure 11 montre que le nombre de paquets reçus dans LEACH est inférieur à celui du LEACH modifié.

Le nombre de paquets perdus (5 périodes) :

La quantité d'informations perdues au niveau de la station de base est un autre critère de performance. La figure 12 montre le nombre de paquets perdus dans les deux versions.

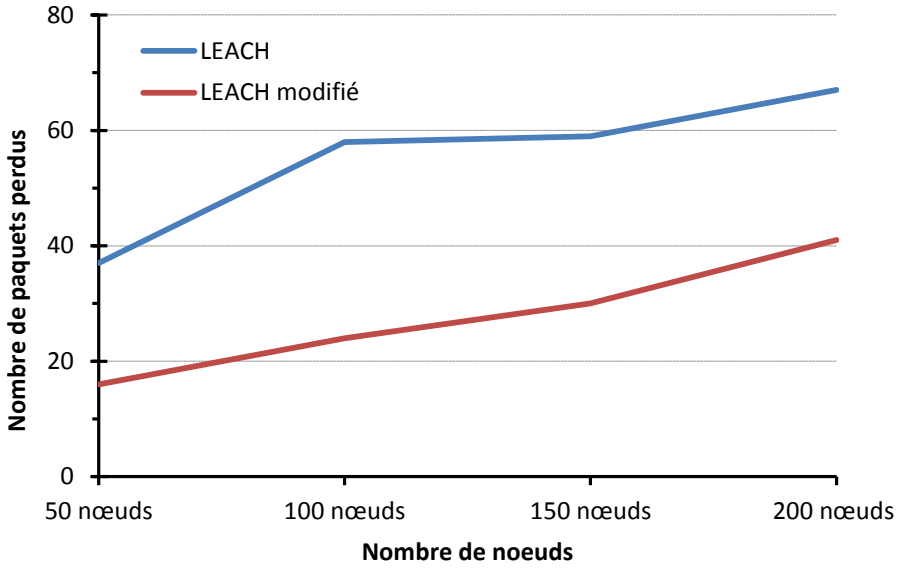

Figure 12: Nombre de paquets perdus vs. nombre de nœuds (5 périodes)

Le nombre de paquet perdus est un peu plus élevé dans LEACH que dans LEACH modifié car dans LEACH modifié le CH transmet les donnés à un autre CH intermédiaire si la probabilité<0.3.

b) Energie dissipée (5 périodes) :

La figure 13 montre l'énergie dissipée dans les deux versions.

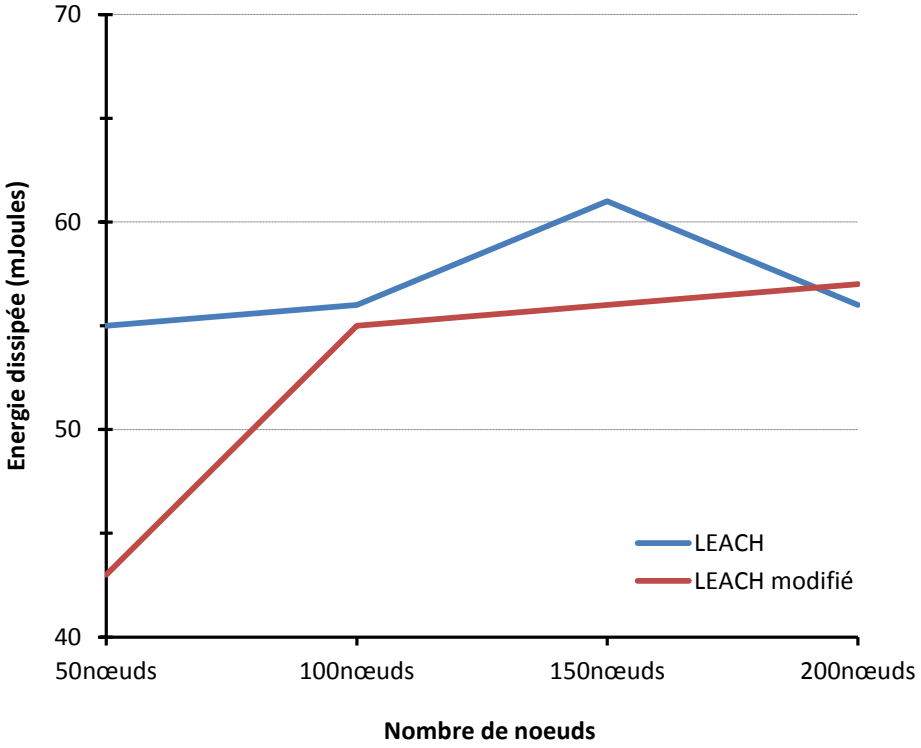

Figure 13: Energie dissipée vs. nombre de nœuds (5 périodes)

L'énergie dissipée dans LEACH est supérieur à celui du LEACH modifié car l'envoi d'un paquet ne garantit pas sa réception donc le nœud retransmit l'envoi de ce paquet.

c) Nombre de paquets perdus vs. taille du réseau (avec des CH désactivés)

Tableau 3: Nombre de paquets perdus vs. taille du réseau (avec des CHs désactivés) dans LEACH

LEACH	50 nœuds	100 nœuds	150 nœuds	200 nœuds
Nombre de CH désactivés	2	2	3	2
Nombre de paquets perdus	37	58	59	67

Tableau 4: Nombre de paquets perdus vs. taille du réseau (avec CHs désactivés) dans LEACH_modifié

LEACH modifié	50 nœuds	100 nœuds	150 nœuds	200 nœuds
Nombre de CH désactivés	1	2	1	1
Nombre de paquets perdus	16	24	30	41

d) Le nombre de paquets reçus (100 nœuds) :

La figure 14 illustre le nombre de paquets reçus dans les deux versions.

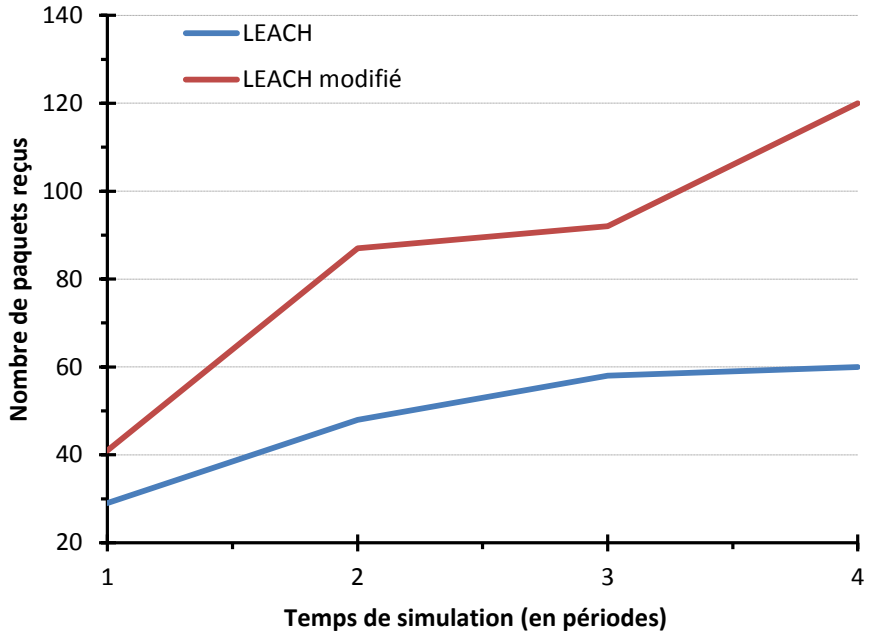

Figure 14: Nombre de paquets reçus en fonction du temps (pour 100 nœuds)

La figure 14 montre que le nombre de paquets reçus dans LEACH modifié est supérieur à celui du LEACH originale.

e) Nombre de paquets perdus (100 nœuds) :

La figure 15 montre le nombre de paquets perdus dans les deux versions.

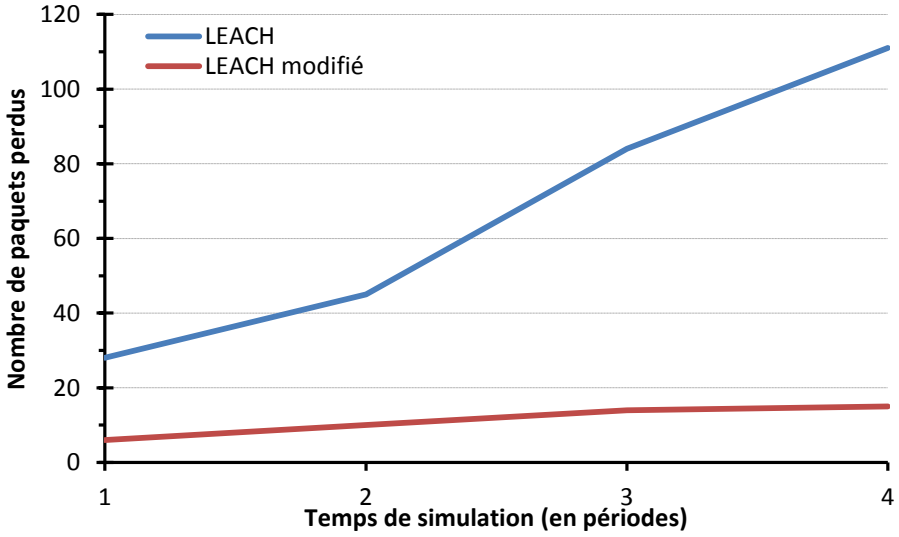

Figure 15: Nombre de paquets perdus en fonction du temps (en périodes)

Le nombre de paquet perdus dans LEACH modifié est inférieur à celui du LEACH, car dans LEACH modifié si la probabilité<0.3 le CH transmet les donnés à un autre CH intermédiaire.

f) Energie dissipée (100 nœuds) :

La figure 16 montre l'énergie dissipée dans les deux versions.

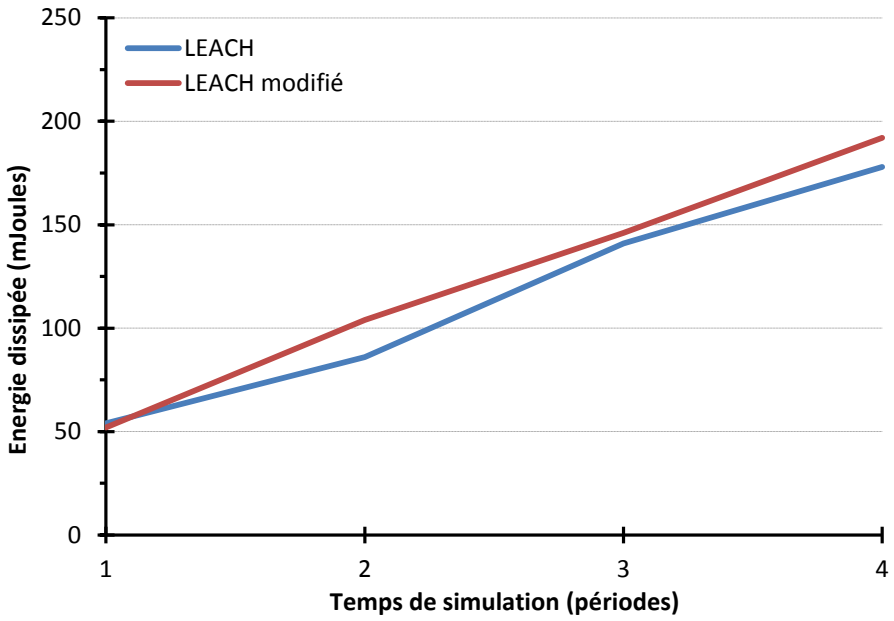

Figure 16: Energie dissipée en fonction du temps (100 nœuds)

L'énergie dissipée dans LEACH modifié est supérieur à celui du LEACH, car il consomme d'énergie dans la réception de paquets.

g) Nombre de paquets perdus en fonction du temps (avec des CHs désactivés) dans LEACH

Tableau 5: Nombre de paquets perdus en fonction du temps (avec des CHs désactivés) dans LEACH

LEACH	1 période	2 périodes	3 périodes	4 périodes
Nombre de CHs désactivés	1	3	1	5
Nombre de paquets perdus	28	45	85	111

Tableau 6: Nombre de paquets perdus en fonction du temps (avec des CHs désactivé) dans LEACH modifié

LEACH modifié	1 période	2 périodes	3 périodes	4 périodes
Nombre de CHs désactivés	3	1	2	3
Nombre de paquets perdus	6	10	14	15

Les résultats de simulation montre que le nombre de paquets reçus dans LEACH modifié est supérieur à celle de la version originale de LEACH par contre le nombre de paquets perdus est inférieur.

Le nombre de paquets perdus augmente quand le nombre de nœuds et le nombre des CHs désactivés augmentent.

On remarque que LEACH modifié est plus tolérant aux pannes par rapport à la version originale de LEACH car dans LEACH modifié si la probabilité de réception sans erreur est faible un CH intermédiaire sera impliqué dans l'acheminement des données.

III.5 Conclusion

Dans ce chapitre, on a évalué LEACH dans un environnement de propagation probabiliste présentant une irrégularité de la radio. Les résultats obtenus ont montré que les performances de LEACH se dégradent dans ce type d'environnements. Dans ce contexte, on a proposé une version de LEACH qui s'adapte à un environnement de propagation non-idéal. Les résultats obtenus ont montré une certaine amélioration.

Conclusion générale

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 RCSF rencontrent plusieurs problèmes qui affectent leur bon fonctionnement dû à leurs caractéristiques telles que la limitation de la batterie, le type de communication, les environnements hostiles où sont déployés les capteurs. Par ailleurs, ces réseaux sont caractérisés par les pannes des nœuds qui peuvent causer un dysfonctionnement du réseau en entier. Dans cette optique, il est commode de proposer des protocoles de routage tolérants aux pannes.

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

Dans ce mémoire, on a réalisé une étude pour atteindre un routage efficace dans un environnement de propagation probabiliste avec tolérance aux pannes dans les RCSF. Cet aspect est fondamental pour ce genre de réseaux où le routage se réalise en collaboration avec les différents nœuds du réseau. De ce fait, un protocole de routage doit prendre en compte l'environnement de propagation de la radio. Dans ce cadre, on évalué LEACH dans un environnement de propagation de la radio qui est non-idéal. Puis on a proposé une version qui s'adapte à ce type de modèle.

En perspectives, on propose de faire adapter les autres protocoles de routage conçus aux RCSF à un environnement de propagation non-idéal.

Références bibliographiques

Références bibliographiques

[1] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," in Proceedings of the 33rd Annual Hawaii International Conference on System Sciences, January 2000, pp. 1–10.

[2] W. Heinzelman, A. Chandrakasan and H. Balakrishnan, "An application specific protocol architecture for wireless microsensor networks," IEEE Transactions on Wireless Communications, vol. 1, no. 4, pp. 660–667, 2002.

[3] F. Brissaud, D. Charpentier, A. Barros and C. Berenguer, "Capteurs intelligents : nouvelles technologies et nouvelles problématiques pour la sûreté de fonctionnement", Actes du 16ième Congrès de maîtrise des risques et sûreté de fonctionnement, 6-10 octobre 2008, Université d'Avignon.

[4] Vernon S. Somerset, Intelligent and Biosensors, Edited by Vernon S. Somerset. INTECH, Croatia, January 2010.

[5] David Martins, "Sécurité dans les réseaux de capteurs sans fil : Stéganographie et réseaux de confiance", Thèse de Doctorat, L'U.F.R. des Sciences et Techniques de l'université de Franche-Comté, 2010.

[6] Yacine Younes, "Minimisation d'énergie dans un réseau de capteurs", Mémoire de Master, Département d'Informatique, Université Mouloud Mammeri de Tizi-Ouzou, Septembre 2012.

[7] Tinyos : http:/www.tinyos.net/, Modified, May 14, 2013.

[8] Dunkels, A., B. Grönvall and T. Voigt. "Contiki: a Lightweight and Flexible Operating System for Tiny Networked Sensors", In Proceedings of the First IEEE Workshop on Embedded Networked Sensors, pages 455-462, Florida, USA, 2004.

[9] B. Malli. (2011, May) Micaz datasheet @ONLINE :https://fr.scribd.com/doc/56641260/Micaz-Datasheet

[10] C. Mettu. (2011) Telosb datasheet @ONLINE. : https://fr.scribd.com/doc/68138250/Telosb-Datasheet-t

[11] Jolly, G.; Kuscu, M.C.; Kokate, P.; Younis, M., "A low-energy key management protocol for wireless sensor networks," In Proceedings of the 8th IEEE International Symposium on Computers and Communication (ISCC'2003), vol., 1, pp.335-340, 30June-3 July 2003.

[12] S. Tilak, N. B. Abu-Ghazaleh et W. B. Heinzelman, "A taxonomy of wireless microsensor network models", Mobile Computing and Communications Review, vol. 6, no. 2, pp. 28-36, April 2002.

[13] H. Karl et A. Willig, "Protocols and Architectures for Wireless Sensor Networks", John Wiley & Sons, 2005.

[14] S. K. Singh et al., "Applications, Classifications, and Selections of Energy-Efficient Routing Protocols for Wireless Sensor Networks", International Journal of Advanced Engineering Sciences and Technologies, vol.1, no.2, pp.85–95, 2010.

[15] K. Romer and F. Mattern, "The design space of wireless sensor networks", in IEEE Wireless Communications, 2006.

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

[17] http://www.bluetooth.com/Pages/Bluetooth-Home.aspx, modified June 2015.

[18] Kothari, N.B.; Sudarshan, T.S.B.; Gurunarayanan, S.; Chandrasekhar, "SOC design of a Low Power Wireless Sensor network node for Zigbee Systems," Advanced Computing and Communications, 2006. ADCOM 2006. International Conference on , vol., no., pp.462,466, 20-23 Dec. 2006.

[19] Souto, E.; Gomes, R.; Sadok, D.; Kelner, J., "Sampling energy consumption in wireless sensor networks," In Proceedings of the IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing, vol.1, pp.4, 5-7 June 2006.

[20] J.N Al-Karaki and A. E. Kamal, "Routing Techniques in Wireless Sensor Networks: A Survey", Magazine: IEEE Communications, vol. 11, no. 6, pp. 6-28, December 2004.

[21] D. Niculescu, "Topics In Ad-Hoc Networks: Communication Paradigms for Sensor Networks", NEC Laboratories America, IEEE Communications Magazine, Mars 2005.

[22] C. Townsend and S. Arms, "Wireless Sensor Networks: Principles and Applications",MicroStrain, Inc.

[23] R. Jurdak, "Wireless Ad Hoc and Sensor Networks:A Cross-Layer Design perspective", University College Dublin, 2007.

[24] N. C. Brent, J. C. Charles, and S. J. David, "Unit disk graphs," Discrete Mathematics, vol. 86, no. 1-3, pp. 165–177, 1990.

[25] T. He, C. Huang, B. M. Blum, J. A. Stankovic, and T. Abdelzaher, "Range-free localization schemes for large scale sensor networks," in Proceedings of the 9th ACM Annual International Conference on Mobile Computing and Networking (MobiCom'03), pp.81–95, 2003.

[26] P. Levis, N. Lee, M. Welsh, and D. Culler, "TOSSIM: accurate and scalable simulation of entire tinyos applications," in Proceedings of the 1st ACM International Conference on Embedded Networked Sensor Systems (SenSys'03), pp.126–137, 2003.

[27] G. Zhou, T. He, S. Krishnamurthy, and J. A. Stankovic, "Models and solutions for radio irregularity in wireless sensor networks," ACM Transactions on Sensor Networks, 2010.

[28] N. Baccour, A. Koubâa, L. Mottola, M. A. Zúñiga, H. Youssef, C. A. Boano, and M. Alves, "Radio link quality estimation in wireless sensor networks : A survey," ACM Transactions on Sensor Networks, vol. 8, no. 4, pp. 1–34, September 2012.

[29] A. Cerpa, J. L. Wong, M. Potkonjak, and D. Estrin, "Temporal properties of low power wireless links: Modeling and implications on multihop routing," in Proceedings of the 6th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc'05), pp. 414–425, 2005.

[30] A. Abdi, C. Tepedelenlioglu, M. Kaveh, and G. Giannakis, "On the estimation of the k parameter for the rice fading distribution," IEEE Communications Letters, vol. 5, no. 3, pp. 92–94, March 2001.

[31] P. M. Shankar, "Introduction to Wireless Systems", New York, NY, USA : John Wiley & Sons, Inc., 2001.

[32] T. He, C. Huang, B. M. Blum, J. A. Stankovic, and T. Abdelzaher, "Range-free localization schemes for large scale sensor networks," in Proceedings of the 9th ACM Annual International Conference on Mobile Computing and Networking (MobiCom'03), pp. 81–95, 2003.

[33] B. W. Kernighan and D. M. Ritchie, "The C Programming Language, Second Edition", Prentice Hall, 1988.

[34] David GAY, Philip LEVIS, David CULLER, Eric BEWER ; “nesC 1.1 Language Reference Manual“, 2003.

Résumé: Evaluation du protocole de routage LEACH dans un environnement bruité

La plupart des protocoles de routage conçus pour les RCSF ont été développé dans un environnement idéal. Cependant, la fluctuation du signal causée par les obstacles, la distance, le bruit, etc.… pourrait avoir un impact sur leurs performances.

Dans ce travail, on a évalué LEACH dans un environnement fortement bruité. Les résultats de la simulation ont montré que ce protocole n'est pas tolérant aux pannes. Puis, on a proposé une version améliorée de LEACH qui s'adapte à un environnement de propagation non-idéal.

Mots clés : Le modèle du disque unitaire, LEACH, le modèle d'irrégularité de la radio, TOSSIM, TinyOS.

Abstract: Evaluation of LEACH routing protocol in a noisy environment

Most routing protocols designed for WSN have been developed in an ideal environment represented by the unit disk model. However, the signal fluctuation caused by obstacles, distance, noise, etc ... could have an impact on their performance.

In this work was evaluated LEACH in a noisy environment. The simulation results showed that this protocol is not fault tolerant. Then, we proposed an improved version of LEACH that adapts to a non-ideal propagation environment.

Keywords: Unit disk model, LEACH, Radio irregularity model, TOSSIM, TinyOS.

ملخص: تقييم LEACH في بيئة غير مثالية

اقترحت معظم بروتوكولات التوجيه المصممة لWSN في بيئة مثالية. لكن اضطراب الموجات الهيرتزية الناجمة عن وجود العوائق، وبعد المسافة، والضوضاء، الخ ... يكون لها تأثير على أدائهم.

في هذا العمل تم تقييم LEACH في بيئة غير مثالية . وأظهرت نتائج المحاكاة أن هذا البروتوكول لا يتوافق مع وجود أعطاب. ثم اقترحنا نسخة محسنة ل LEACH تناسب بيئة انتشار غير المثالية.

الكلمات المفتاحية: نموذج القرص الوحدوي، LEACH, نموذج الموجات الغير منتظمة، TinyOS, TOSSIM