



République Algérienne Démocratique et Populaire



Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université Akli Mohand Oulhadj de Bouira

Faculté des Sciences et des Sciences Appliquées

Département d'Informatique

Mémoire de Master

en Informatique

Spécialité : ISIL

Thème

Routage dans les réseaux de capteurs sans fil sous
contrainte de l'énergie ambiante

Jury

— DR. ABBAS Akli **Encadreur**
— DR. BAL Kamal **Président**
— DR. BADIS Lyes **Examineur 1**
— DR. ALIOUAT Ouahiba **Examineur 2**

Réalisé par

— LAKEHAL Khedidja
— RABHI Rofaida

2019/2020

Remerciements

Au terme de ce travail, nous remerciant ALLAH qui nous aide et nous donne la patience et le courage durant ces années d'étude.

Nous exprimons nos profondes gratitude et remerciements à la personne qui nous a encadré Dr. Akli Abbas pour son soutien, ainsi que pour ses conseils instructifs durant toute la période de ce travail dans les meilleures conditions.

Nos plus vifs remerciements s'adressent aussi à tout le cadre professoral et administratif de la formation Master dans le spécialité Ingénierie des Systèmes d'Information et Logiciel, pour leur patience et savoir qui nous a illuminés durant ces années de formation.

Enfin, on remercie toutes personnes qui ont contribué soit de près ou de loin à la réalisation de ce travail, ainsi qu'au bon déroulement, et dont les noms ne figurent pas dans ce document.

Dédicaces

Je dédie mon travail à mes chers parents.

A mes Chers frères.

Et mes chers sœurs.

A la meilleure des binômes *Rofaida Rabhi*, pour tous les bons moments qu'on a passés ensemble.

A tous ma famille.

Merci à tous mes amis avec qui je partage des moments de ma vie au fil du temps.

Khadija.

Dédicaces

Je dédie mon travail à mes chers parents.

Mon cher père *Djillali*, pour son amour, son affection et son soutien qui ne me quittent
jamais.

Ma cher mère *Rbiha*, pour tout ce qu'elle est pour moi, ma mère, ma vie, ma amour.
Que ce travail soit l'expression de ma reconnaissance pour vos sacrifices Consentis, votre
soutient moral et matériel que vous n'avez cesse de prodiguer.

Vous avez tout fait pour mon bonheur et ma réussite.

Que dieu vous préserve en bonne santé et vous accorde une longue vie.

A mon fiancé *Salah Eddine* pour bon motivation, et tous les moments qu'il m'encourage
et me soutient

A mes Chers frères *Abd Essamed* et *Saif Eddine*, Et mes chers sœurs *Nafissa* et *soundous*
Pour leurs soutiens et leurs encouragements.

A la meilleure des binômes *khadija Lakehal*, pour tous les bons moments qu'on a passés
ensemble.

A tous ma famille et tous mes amie(é)s..

Surtout *Aida, rima, khadidja, hind, hadjer, sara, khawla, maroua, fatima, karima,*
fatma, zahra, chahinez, fairouz...

Merci à tous mes amis avec qui je partage des moments de ma vie au fil du temps.

Rofaida.

Table des matières

Table des matières	i
Table des figures	iv
Liste des tableaux	vi
Liste des abréviations	2
Introduction générale	4
1 Généralités sur les RCSF	7
1.1 Introduction :	7
1.2 Historique des réseaux de capteurs sans fil :	7
1.3 Les capteurs :	8
1.3.1 Capteur sans énergie renouvelable :	8
1.3.2 Capteur avec énergie renouvelable :	11
1.4 Systèmes embarqués pour les capteurs :	14
1.5 Les réseaux de capteurs sans fil (RCSF) :	16
1.5.1 Réseaux de capteurs sans fil sans énergie renouvelable(Energy Efficient Wireless Sensor Network EE-WSN)	16
1.5.2 Les réseaux de capteurs fil avec énergie renouvelable (Energy Harvesting Wireless Sensor Network EH-WSN) :	17
1.6 Applications d'un RCSF :	18
1.7 Communication dans les réseaux de capteurs sans fil :	20
1.8 Consommation d'énergie dans les réseaux de capteurs sans fil :	22

1.8.1	Les principales opérations dues à la consommation d'énergie	22
1.8.2	Une comparaison entre la consommation d'énergie dans les RCSF avec et sans énergie renouvelable :	23
1.8.3	Modèle de consommation d'énergie :	24
1.9	Conclusion	25
2	Le routage dans les RCSF	26
2.1	Introduction	26
2.2	Le routage dans le RCSF :	27
2.2.1	Définition de routage :	27
2.2.2	Classification des protocoles de routage :	28
2.3	Facteurs de conception de protocoles de routage :	30
2.3.1	Tolérance aux pannes :	30
2.3.2	Consommation d'énergie :	30
2.3.3	Limitations de capacités des nœuds :	30
2.3.4	Scalabilité :	31
2.3.5	Connectivité :	31
2.3.6	Hétérogénéité :	31
2.4	Le routage dans les RCSF sans énergie renouvelable :	32
2.4.1	Objectifs du routage dans les RCSF :	32
2.4.2	Quelques protocoles de routage utilisés dans les RCSF sans énergie renouvelable :	32
2.5	Le routage dans les RCSF avec énergie renouvelable :	36
2.5.1	Quelques algorithmes de routage dans ISEH :	36
2.6	Conclusion	38
3	Le protocole EH-LCHREP(Energy Harvesting Layer and Cluster based Hierarchical Routing and Energy Optimization Protocol) :	39
3.1	Introduction	39
3.2	Description de protocole :	39
3.2.1	Motivation	39
3.2.2	Hypothèse :	40
3.2.3	principe de fonctionnement :	41

3.3 Conclusion :	52
4 Evaluation des performance	53
4.1 Introduction	53
4.2 Les simulateurs des réseaux de capteurs :	53
4.2.1 OPNET (Optimum Network Performance) :	53
4.2.2 L'OPNET présente trois fonctions principales :	54
4.2.3 Qualnet Simulator :	55
4.2.4 NS3 (Network Simulator version3) :	55
4.3 Environnement de simulation :	56
4.3.1 Python :	56
4.3.2 Paramètres de simulation	56
4.3.3 Description du système :	57
4.3.4 Variables suggestives du système :	57
4.4 Évaluation des performances :	58
4.4.1 Distribution des nœuds :	58
4.4.2 Formation cluster :	59
4.4.3 La sélection des Clusters Head et les Faux Clusters Head :	59
4.4.4 Énergie restante après la sélection des Clusters Head :	60
4.4.5 Modèle de source énergie :	61
4.4.6 Énergie consommée après l'envoi de 100 paquets :	62
4.4.7 Nombre de nœuds mort en termes de paquets envoyés :	63
4.5 Conclusion	65
 Conclusion générale et perspective	 66
 Bibliographie	 68

Table des figures

1.1	Exemple d'un capteur [21].	8
1.2	Architecture matérielle d'un capteur.	9
1.3	Exemple d'un capteur sans fil basée sur la récupération d'énergie avec une cellule solaire.	11
1.4	Architecture générale d'un nœud capteur sans fil avec des capacités de récupération d'énergie.	12
1.5	Architecture d'un ISEH system.	13
1.6	Architecture d'un réseau de capteur sans fil	17
1.7	EHWSN architecture	18
1.8	Domaines d'applications des RCSF [19].	20
1.9	Types de communication dans réseaux de capteurs sans fil	21
1.10	Une comparaison de consommation d'énergie entre des RCSF avec et sans énergie renouvelable	23
1.11	Modèle de consommation d'énergie.	24
2.1	Protocole plat.	28
2.2	Protocole hiérarchique.	29
2.3	Protocole basé sur la localisation.. . . .	29
4.1	Exemple d'interface d'OPNET.	54
4.2	Exemple d'interface de QualNet.	55
4.3	Présentation de la topologie du réseau.	59
4.4	l'architecture du réseau après la formation des clusters.	59
4.5	Visualisation du réseau après la sélection des FCHs et CHs	60

4.6	Énergie restante après la sélection des CHs	61
4.7	La courbe de la puissance $P_s(t)$	62
4.8	Energie consommée après l'envoi de 100 paquet (EH-LCHREP).	62
4.9	Energie consommée après l'envoi de 100 paquets (EEPSC).	63
4.10	Nombre de nœuds mort en termes de paquets envoyés (avec distance).	64
4.11	Nombre de nœuds mort en termes de paquets envoyés (sans distance).	64
4.12	Nombre de nœuds mort en termes de paquets envoyés par protocole EEPSC.	65

Liste des tableaux

1.1	Les trois générations des nœuds de capteurs [1]	8
4.1	Paramètres de simulation.	57
4.2	Variables suggestives du système	58

Liste des abréviations

WSN	Wireless Sensor Network
RCSF	Réseau de capture sans fil
ISEH	Intelligent Solar Energy Harvesting
IoT	Internet of Thing
MPPT	Maximum Power Point Tracking
OS	Operating System
TinyOS	Tiny Operating System
TDMA	Time Division Multiple Access
CDMA	Code Division Multiple Access
EEWSN	Energy Efficient Wireless Sensor Network
EHWSN	Energy Harvisting Wireless Sensor Network
GPS	Global Positioning System
LEACH	Low Energy Adaptive Clustering Hierarchy
CM	Cluser Momber
CH	Cluster Head
FCH	Faux Cluster Head
HASS	Harvisting Aware Speed Selection
AFEPF	Adapptive Framework Energy Prediction
QoS	Qualitiy of Servic
NS3	Network Simulateur version3
OPNET	Optimum NETwork Performance
LZHCRP	Level and Zone based Hierarchical Routing Energie optimisation Protocol
EH-LCHREP	Energy Harvesting Layer and Cluster based Hierarchical Routing and Energy opti
EEPSC	Energy Efficient Protocol with Static Clustering

Introduction générale

Motivations :

Le monde actuel a besoin d'utiliser la technologie de communication sans fil, Pour cela, des capteurs de minuscules tailles et de faible puissance ont été développé afin de recueillir des informations à partir des milieux très sensible, inaccessible et indisponible. Ces environnements nécessitent des réseaux de capteurs sans fil, un réseau de capteurs sans fil consistent d'un grand nombre des petits capteurs, ces derniers sont dotés de la capacité de collecter des grandeurs physique telles que la température, la pression... et les diffuser au sien de réseau. Actuellement la plupart des travaux de recherche dans le domaine des réseaux de capteurs se concentrent sur le problème conservation d'énergie car les capteurs sont alimentés par des batteries de faible puissance. Donc l'objectif primordial est le développement des techniques efficaces qui permettant d'économie l'énergie pour le réseau de capteurs sans fil. Ces techniques concentrent sur la topologie du réseau et l'acheminement des données vers un centre de traitement.

Parce que les **RCSF** n'ont pas de moyen direct de recevoir de l'énergie pour fonctionner, les chercheurs ont étudié diverses méthodes de récupération d'énergie, ils ont développé des capteurs qui sont capable de récupérer l'énergie ambiante depuis l'environnement tel que l'énergie solaire, thermique...etc. Cette approche permet de prolonger la durée de vie des batteries ou les élimine complètement. Il existe plusieurs protocoles de routage qui ont été proposé avec cette approche ce qui permet de maximiser la longévité de réseau.

Objectif de notre étude :

L'objectif principal de ce mémoire est de proposer un nouveau protocole de routage qui prend en compte l'énergie ambiante, cet objectif est concentré sur l'étude des protocoles de routages sans énergie renouvelable et avec énergie renouvelable. Pour ensuite proposer un nouveau protocole de routage qui essayerai de prolonger la durée de vie de réseau et en même temps nous devons remarquer que la puissance environnementale est limitée. Donc ce protocole de routage doit être conscient a ces conditions environnementales.

Contribution :

Dans ce mémoire nous avons proposé un protocole de routage hiérarchique dans le cas des réseaux de capteurs sans fil prenant en compte l'énergie ambiante. Ce protocole est appelé **EH-LCHREP** (Energy Harvesting Layer and Cluster based Hierarchical Routing and Energy Optimization Protocol) qui est l'amélioration de protocole de routage LCHREP [16]. Les résultats obtenus montre que EH-LCHREP permet de prolonger la durée de vie du réseau.

Organisation du mémoire :

Ce mémoire est structuré en quatre chapitre :

Le premier chapitre de notre travail est consacré à l'étude des généralités sur les réseaux de capteurs sans fil. On a expliqué la signification du réseau de capteurs sans fil à base de récupération d'énergie (EH-WSN) et en quel point sont différentes d'un **RCSF** normal (EE-WSN), tel que l'architectures et les caractéristiques. Ensuite on a expliqué le fonctionnement des deux catégories des réseaux de capteurs, après on a défini les domaines d'applications de cette technologie et les causes de la consommation énergétique.

Dans **le deuxième chapitre** nous étudions des différentes techniques développées pour conserver l'énergie, débutons par la définition de routage ensuite nous expliquons les différentes méthodes de routages, après nous détaillons quelques protocoles de routage sans énergie renouvelable, et des protocoles avec énergie renouvelable.

Et dans **le troisième chapitre** nous proposons notre protocole de routage hiérarchique que nous avons dénommé **EH-LCHREP (Energy Harvesting Layer and Cluster based Hierarchical Routing and Energy Optimization Protocol)** et qui est l'amélioration de protocole de routage (LCHREP), **EH-LCHREP** c'est un protocole de routage hiérarchique basé sur la récupération d'énergie ambiante.

Dans **le dernier chapitre**, nous présenterons les résultats d'évaluation de notre protocole effectué avec Python.

Généralités sur les RCSF

1.1 Introduction :

Les avancées croissantes des technologies de miniaturisation et de développement de communication sans fil, ont poussé à la création et au développement des réseaux de capteurs sans fil (WSNs Wireless Sensor Networks en anglais).

Les réseaux de capteurs sans fil (**RCSF**) ont émergés pour des besoins militaires tels que sur le terrain de combat. Puis, ils ont trouvé leur chemin pour les applications civiles.

Aujourd'hui les **RCSF** jouent un rôle principale dans le monde physique réel, en fournissant des données collectés comme la luminosité, les radiations et les mouvements... vers un réseau informatique.

La technologie de capteurs combinée avec la puissance de traitement et la communication sans fil, la rend lucrative pour être exploitée en abondance dans le future.

Dans ce chapitre nous allons faire une présentation générale de la technologie **RCSF**.

1.2 Historique des réseaux de capteurs sans fil :

Les derniers développements des nouvelles technologies ont eu une grande importance dans le domaine des réseaux sans fil. La technologie de réseau de capteurs sans fil est devenue l'une des merveilleuses technologies du 21ème siècle, et les réseaux de capteurs ont montré leurs impact sur notre vie quotidienne [1].

Le tableau 1.1 suivant illustre l'évaluation des réseaux de capteurs.

Génération	Période	Taille	Poids	Batterie
1ère	Les années 80 et 90	Grande boîte à chaussures	Kilogrammes	Grosse
2ème	Entre 2000 et 2003	Boîte de cartes	Grammes	AA
3ème	2010	Particule de poussière	Négligeable	Solaire

TABLE 1.1 – Les trois générations des nœuds de capteurs [1]

1.3 Les capteurs :

On distingue deux types de capteurs sans fil, un capteur avec énergie renouvelable et un capteur sans énergie renouvelable.

1.3.1 Capteur sans énergie renouvelable :

Définition :

En générale les capteurs sont des objets électriques de petites tailles (Figure 1.1), déployés dans l’environnement pour capter des grandeurs physiques (la température, le taux d’humidité, des vibrations, pression) et les transformer à un signal électrique [2]. Chaque capteurs remplit trois fonctions principales : collecter, traiter et transférer des informations vers un ou plusieurs points de collecte appelés stations de base (**SB**) [3].



FIGURE 1.1 – Exemple d’un capteur [21].

Architecture :

Nous présentons ici un aperçu de l'architecture matérielle et logicielle d'un nœud capteur qui peut effectuer plusieurs fonctionnalités.

a - L'architecture matérielle :

Il existe plusieurs tailles et plusieurs formes des capteurs, dépend de choix de son application, tel que les nœuds capteurs de température, d'humidité, et de pression...

Mais l'architecture reste similaire pour tous les capteurs comme montre dans la figure 1.2 et elle se compose généralement des éléments suivants :

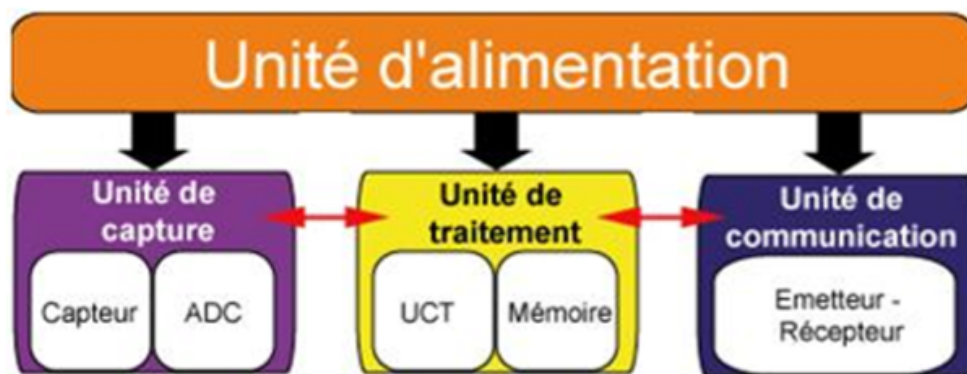


FIGURE 1.2 – Architecture matérielle d'un capteur.

-**Une unité de traitement** : C'est l'unité principale du capteur appelé aussi unité de calcul, elle se compose généralement d'un processeur et une mémoire vive. Elle a comme rôle le contrôle de bon fonctionnement des autres unités [8].

-**L'unité de capture** : Il permet la capture de données, c'est-à-dire la mesure de grandeurs physiques ou analogiques et leur conversion en données numériques. Il se compose du capteur lui-même et ADC (Analog to Digital Converter). Le capteur est responsable de la récupération du signal analogique transmis au ADC, et le capteur a pour fonction de convertir et de transmettre les données analogiques en données numériques que l'unité de traitement peut comprendre [8].

-**L'unité de communication** : Sa fonction est d'envoyer et de recevoir des informations. Il est équipé d'une paire émetteur / récepteur. Si nous nous intéressons aux radio

fréquences (ondes radio), cela permet la communication au sein du réseau. Cependant, il existe d'autres possibilités de transmission (optique, infrarouge, etc.) [8].

-L'unité d'alimentation : Une partie très importante de l'architecture du capteur est d'alimenter toutes les autres unités. Il correspond généralement à des batteries ou batteries qui alimentent des capteurs, et ses ressources limitées en font un problème unique pour de tels réseaux. La dernière mise en œuvre des blocs d'alimentation basés sur des panneaux solaires tente de fournir une solution pour prolonger leur durée de vie [8].

b - Architecture logicielle :

L'architecture logicielle met en œuvre les fonctions suivantes : collecte de données, communication avec d'autres nœuds, sécurité pour protéger les informations échangées via le réseau, localisation dans les applications de suivi, etc. Selon la combinaison de l'architecture matérielle et logicielle de chaque nœud, trois types de nœuds peuvent être déterminés [4] :

-Nœud de capteur : il comprend des unités de collecte de données telles que la température et l'humidité. Envoyez ensuite ces données à d'autres nœuds.

-Nœud d'actionneur : il comprend des actionneurs qui agissent sur l'environnement.

-Nœud routeur : utilisé pour acheminer les données entre les nœuds de capteur. Certains nœuds peuvent agir en même temps comme des capteurs et des routeurs.

Caractéristiques :

Les nœuds capteurs peuvent coopérer entre eux et mesurer les conditions des deux environnements environnants, à savoir (lumière, température, son, vibration, etc.). Les nœuds dépendent de certaines caractéristiques pour la transmission, telles que [3] :

-L'énergie : L'énergie est considérée comme la contrainte principale et fondamentale des réseaux de capteurs sans fil. Chaque nœud peut fonctionner, grâce à sa petite batterie non rechargeable avec une capacité limitée. Dans la plupart des cas, l'utilisation de ces capteurs se fera dans un environnement hostile ou dans un environnement inaccessible, ce qui rendra impossible la récupération.

-La puissance de stockage et de traitement : Puisque les nœuds capteurs sont de petites tailles la puissance de stockage et la puissance de traitement reste faible.

-La portée de transmission : La portée de transmission est limitée par la capacité de rayonnement de l'antenne utilisée et la force du signal impliqué. Par exemple, la communication entre deux nœuds capteurs ne peut être effectuée que lorsque la distance entre eux est très grande (en fait soit des dizaines de mètres). De plus, plus la distance est grande, plus le coût énergétique est élevé. Cela est particulièrement vrai s'il existe une barrière physique entre deux nœuds qui empêche la communication. Le capteur a deux zones : la zone de détection (rayon de détection RS) et la zone de communication (rayon de communication RC).

1.3.2 Capteur avec énergie renouvelable :

Définition :

La récupération d'énergie implique l'extraction d'une petite quantité d'énergie à partir de sources de l'environnement ambiant, telle que l'énergie solaire, thermique, vibration... etc. Ces mécanismes sont efficaces pour améliorer la durée de vie d'un capteur (Figure 1.3) [5].

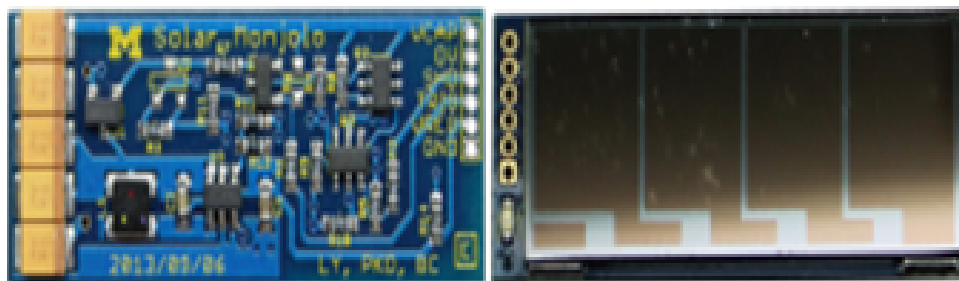


FIGURE 1.3 – Exemple d'un capteur sans fil basée sur la récupération d'énergie avec une cellule solaire.

Architecture :

L'architecture du système d'un nœud capteur sans fil comme montre dans la figure 1.4, Comprend les composants suivants [6] :

- a -le récupérateur d'énergie (*Energie Harvester*) : en charge de la conversion de l'énergie externe ambiante ou générée par l'homme en électricité.
- b -un module de gestion de l'énergie (*Power Managment*) : qui collecte l'énergie électrique

de la récolteuse et la stockée ou la livrée aux autres composants du système pour une utilisation immédiate.

c -stockage d'énergie (Energie Storage) : pour conserver l'énergie récoltée pour une utilisation future.

d -un microcontrôleur(low power microcontroller unit).

e -un émetteur-récepteur radio(radio transceiver) : pour transmettre et recevoir des informations.

f -équipement sensoriel(Sensor).

g -un convertisseur A / D(A/D converter) : pour numériser le signal analogique généré par les capteurs et le mettre à la disposition du microcontrôleur pour un traitement ultérieur.

h -une mémoire(Memory) : pour stocker les informations détectées, les données liées à l'application et le code.

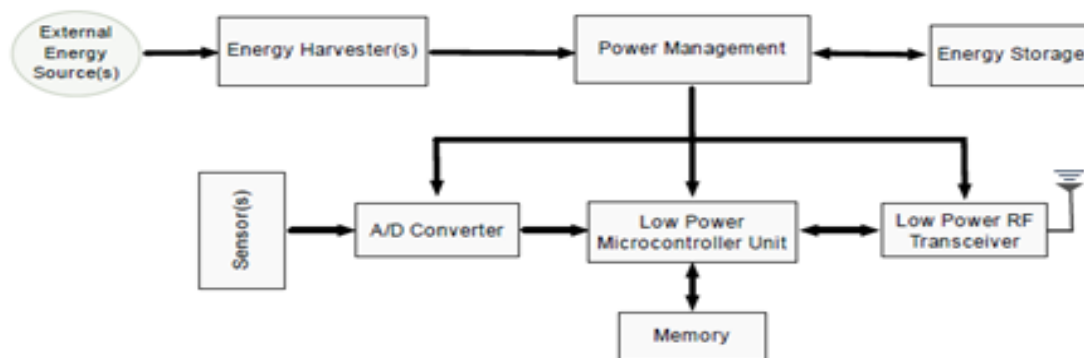


FIGURE 1.4 – Architecture générale d'un nœud capteur sans fil avec des capacités de récupération d'énergie.

Chaque méthode de récupération d'énergie a un système spécifique qu'on détaille dans ce qui suit :

- ISEH system (Intelligent Solar Energy Harvesting) :

Un système intelligent de récupération d'énergie solaire ISEH (Intelligent Solar Energy Harvesting) pour fournir une énergie stable et à long terme. Le système est composé d'un panneau solaire, d'une batterie au lithium et d'un circuit de commande. Le matériel, au lieu du logiciel, est utilisé pour la gestion de la charge de la batterie au lithium, ce qui améliore la fiabilité et la stabilité du système. Il préfère utiliser l'énergie solaire chaque fois que l'ensoleillement est suffisant, et la batterie au lithium est une alimentation

électrique complémentaire pour des conditions telles que le ciel couvert, la pluie et la nuit. Le système adapte un circuit de suivi du point de puissance maximale (Maximum Power Point tracking MPPT) pour tirer pleinement parti de l'énergie solaire, et il assure à la batterie au lithium une durée de vie extrêmement longue avec une méthode de charge appropriée, ce qui raccourcit la fréquence du cycle de charge-décharge de la batterie. Ce système peut être mis en œuvre avec un petit équipement électrique particulièrement adapté aux nœuds de capteurs sans fil basés à l'extérieur dans l'Internet des objets (IOT Internet of Thing).[7]

- Architecture d'un ISEH system :

Le système ISEH est physiquement composé d'un panneau solaire, d'une batterie au lithium et d'un circuit de commande. Le circuit de commande comprend un module solaire MPPT, un sous-circuit de charge, un sous-circuit de protection sur-déchargé et un module boost DC / DC pour la batterie au lithium. Le schéma de principe du système est représenté sur la figure 1.5. Le système a trois branches d'entrée, c'est-à-dire un panneau solaire, une batterie au lithium et une interface mini-USB. Il possède également une branche de sortie offerte par une interface USB ordinaire. Dans cet article, nous utilisons «bimodule» pour souligner que la sortie du système peut provenir du panneau solaire ou de la batterie au lithium, respectivement. L'interface mini-USB standard est une réservation pour charger la batterie au lithium par un adaptateur secteur externe si nécessaire. Les composants fonctionnels du système sont illustrés comme suit [7] :

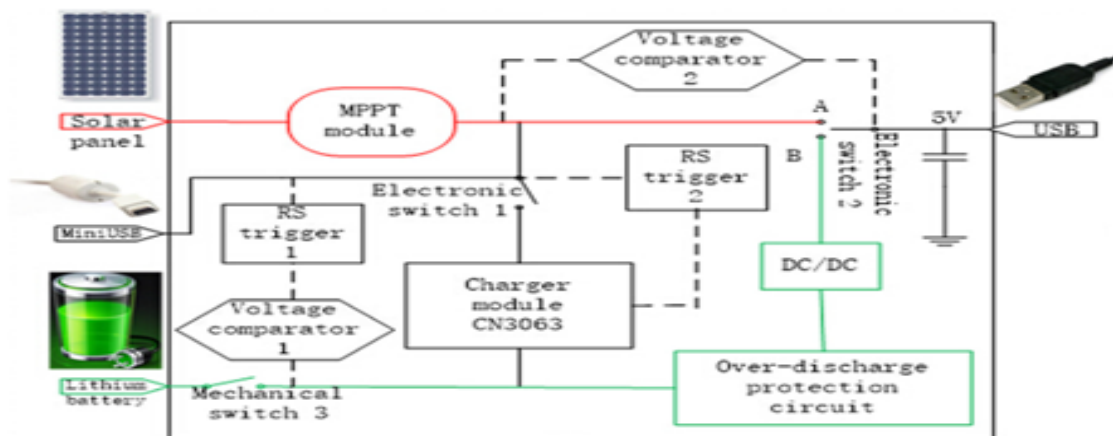


FIGURE 1.5 – Architecture d'un ISEH system.

- Récupération d'énergie à partir de nanogénérateurs piézoélectriques (piezoelectric nanogenerators) :

Les mécanismes conventionnels de récupération d'énergie solaire ne sont pas utilisables dans le cas ou les dispositifs très petits des nœuds capteurs, tels que ceux utilisés dans les réseaux de nanocapteurs sans fil (Wireless NanoSensor networks (WNSN) qui sont utilisés pour des applications telles que les systèmes d'administration de médicaments, ou les capteurs tactiles. Les raisons sont les suivantes :

- la récupération d'énergie solaire contient une nanocellule photovoltaïque à faible efficacité qui ne s'améliore pas même si de nouveaux nanocomposants tels que les nanotubes de carbone sont utilisés pour améliorer leur sensibilité.
- Habituellement, la lumière du soleil n'est pas disponible dans de nombreuses applications du WNSN.
- Les mécanismes pour récolter l'énergie cinétique du vent ne sont pas réalisables à l'échelle nanométrique en raison des limites de la technologie actuelle.

Pour surmonter le problème de la capacité énergétique limitée des nœuds de nanocapteurs, les recherches ont proposé l'utilisation de nanogénérateurs piézoélectriques pour recharger l'énergie des nanodispositifs.

1.4 Systèmes embarqués pour les capteurs :

Un système d'exploitation(en anglais Operating system ou OS) est un logiciel système qui fait une liaison entre le matériel informatique et les ressources logicielles. Parmi les OS qui ont existé pour les réseaux de capteurs nous pouvons citer comme exemple :

A- TinyOS (Tiny Operating System) :

Est un système d'exploitation intégré et une plate-forme pour les appareils sans fil à faible consommation d'énergie. C'est un système d'exploitation open source, conçu par l'université américaine de BERKELEY pour les appareils sans fil à faible puissance, il est utilisé pour les réseaux de capteurs, les réseaux personnels, les bâtiments intelligents et les compteurs intelligents [17]. Sa conception a été entièrement réalisée en NesC2, langage orienté composant syntaxique proche du C, il repose sur une architecture basée sur une

association composante, un modèle de programmation basé sur l'apparition de certains évènements et aussi repose sur un modèle de concurrence basé sur des évènements et des tâches [9].

B- Contiki :

Contiki est un système d'exploitation développé par Adam Dunkels pour un système qui est basé sur un réseau axé sur la faible consommation d'énergie utilisée dans les appareils sans fil dans l'IoT. Le système d'exploitation open source est hautement portable et prend en charge le multi tâche pour les systèmes intégrés dans les réseaux à mémoire efficace et dans les réseaux de capteurs sans fil.

Conkiti est conçu pour fonctionner sur des types de périphériques matériels fortement limités en mémoire, en puissance, en puissance de traitement et en bande passante de communication [17].

C- LiteOS :

Est un système d'exploitation open source et interactif de type Unix-like, développé par Huawei technologies co, conçu pour les réseaux de capteurs sans fil avec les outils fournis avec liteOS, vous pouvez faire fonctionner un ou plusieurs réseaux de capteurs sans fil de manière unix, transférer des données, installer des applications, récupérer des résultats ou configurer des capteurs, vous pouvez également développer des applications pour les nœuds et distribuer ces programmes sans fil aux nœuds capteurs [17].

1.5 Les réseaux de capteurs sans fil (RCSF) :

On distingue deux types de réseaux de capteur

1.5.1 Réseaux de capteurs sans fil sans énergie renouvelable (Energy Efficient Wireless Sensor Network EE-WSN)

Définition :

Un réseau de capteurs sans fil (**RCSF**) est un ensemble des nœuds capteurs de petites tailles, déployés d'une manière aléatoire dans une zone géographique afin de surveiller l'environnement, dans ces réseaux chaque nœud capteur capte et envoi ses données captés à un ou plusieurs points de collecte, à l'aide d'une connexion sans fil [10].

Architecture :

Le réseau de capteurs sans fil est construit autour des quatre entités principales suivantes [2].

- **Capteur** : Comme son nom l'indique, il est chargé de mesurer les valeurs liées à son environnement (température, pression, luminosité, présence, etc.).
- **Agrégateur** : Responsable de la collecte des messages reçus de plusieurs capteurs, puis de les envoi au récepteur (récepteur) sous la forme d'un seul message. L'objectif principal de cette opération est de limiter le trafic sur le réseau, prolongeant ainsi la durée de vie globale du réseau de capteurs.
- **Puit** : est le point final du réseau. Pour lui, toutes les valeurs mesurées par le réseau ont été envoyées. Il peut y avoir plusieurs puits sur le même réseau de capteurs.
- **Passerelle** : est un appareil spécial avec deux interfaces réseau. Il connecte le réseau de capteurs sans fil à un réseau plus traditionnel, généralement Internet. En fait, les réseaux de capteurs ne sont généralement utilisés que pour renvoyer les valeurs mesurées, et les applications qui traitent ces informations sont exécutées sur la machine de l'utilisateur final. Par conséquent, le fonctionnement global de l'architecture comprend les capteurs qui effectuent les mesures, qui sont renvoyés au puits via l'agrégateur. Par conséquent, l'application finale exécutée sur un ordinateur sur un autre réseau peut accéder à la valeur via la passerelle. Notez que

l'agrégateur est facultatif et que le récepteur et la passerelle sont généralement situés dans un seul appareil.

La figure 1.6 illustre l'architecture d'un RCSF.

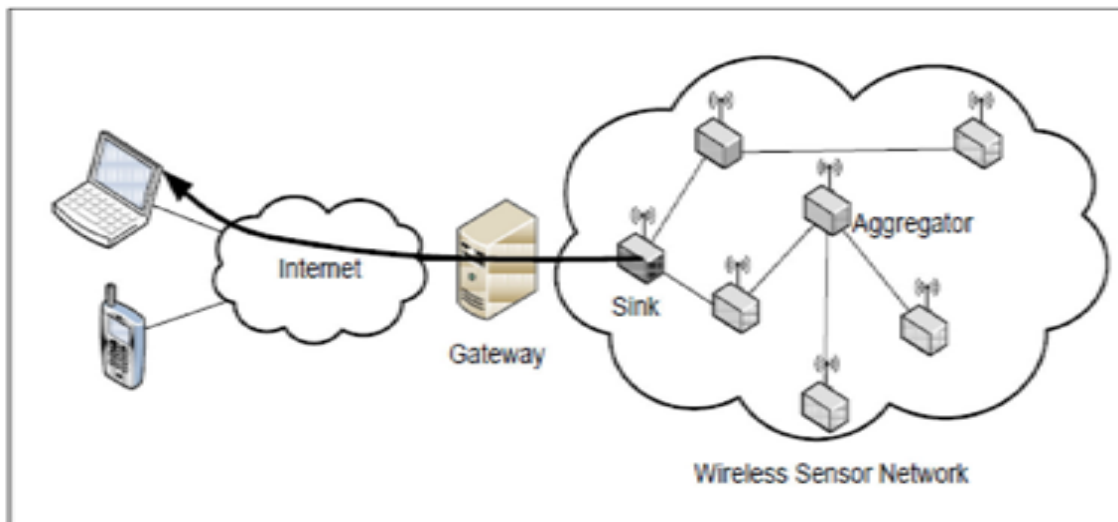


FIGURE 1.6 – Architecture d'un réseau de capteur sans fil

1.5.2 Les réseaux de capteurs fil avec énergie renouvelable (Energy Harvesting Wireless Sensor Network EH-WSN) :

Définition :

Les réseaux de capteurs sans fil basés sur la récupération d'énergie (EHWSN Energy harvesting wireless sensor network en anglais) sont le résultat de la dotation de nœuds WSN avec la capacité d'extraire de l'énergie du milieu environnant. La récupération d'énergie peut exploiter différentes sources d'énergie, telles que l'énergie solaire, le vent, les vibrations mécaniques, les variations de température, les champs magnétiques, etc. Fournissant en permanence de l'énergie et la stockant pour une utilisation future, les sous-systèmes de récupération d'énergie permettent aux nœuds **WSN** de durer potentiellement pour toujours.

Architecture :

Les EH-WSN sont composés des nœuds individuels qui, en plus de la détection et des communications sans fil, sont capables d'extraire l'énergie de plusieurs sources et de la

convertir en énergie électrique utilisable (voire la figure 1.7).

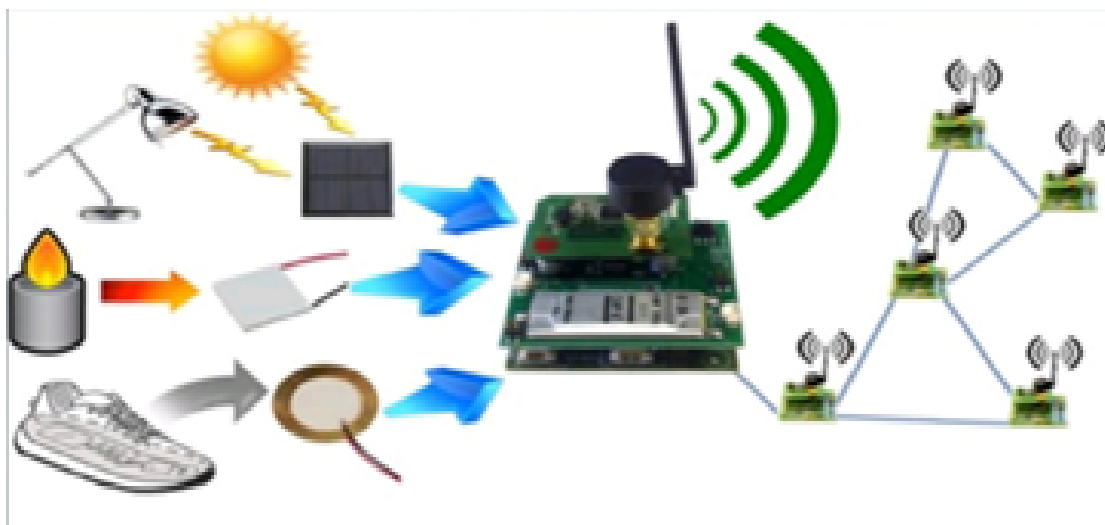


FIGURE 1.7 – EHWSN architecture

1.6 Applications d'un RCSF :

Les RCSF peuvent avoir beaucoup d'applications (voir figure suivante). Parmi elles, nous citons :

- Applications environnementales :

Les applications environnementales comprennent le suivi des mouvements et des modèles d'insectes, d'oiseaux ou de petits animaux [11].

- Applications militaires ou de surveillance des frontières :

Les WSN deviennent une partie intégrante des systèmes de commandement, de contrôle, de communication et de renseignement militaires. Des capteurs peuvent être déployés sur un champ de bataille pour surveiller la présence de forces et de véhicules et suivre leurs mouvements, ce qui permet une surveillance étroite des forces adverses [11].

- Demandes de soins de santé :

L'avancement du WSN offre également de nouvelles opportunités dans le système de santé. Dans la méthode traditionnelle, un patient doit consulter un médecin à intervalles

réguliers et signaler les symptômes ressentis par lui-même. Mais dans les soins à domicile intelligents, le **WSN** recueille des données dans la base des spécifications du médecin et fournit un enregistrement continu pour aider au diagnostic. Cette méthode est également utilisée pour les situations d'urgence et les rappels de médicaments [18].

- Surveillance des conditions environnementales :

Les applications **WSN** dans ce domaine incluent la surveillance des conditions environnementales affectant les cultures ou le bétail, la surveillance de la température, de l'humidité et de l'éclairage dans les immeubles de bureaux, etc. Ces modules de surveillance pourraient même être combinés avec des modules actionneurs qui peuvent contrôler, par exemple, la quantité d'engrais dans le sol, ou la quantité de refroidissement ou de chauffage dans un bâtiment, sur la base de mesures de capteurs distribués [11].

- Renseignements personnels :

Les réseaux de capteurs sans fil peuvent être utilisés pour fournir des environnements de vie plus pratiques et intelligents aux êtres humains. Par exemple, des capteurs sans fil peuvent être utilisés pour lire à distance des compteurs d'utilité dans une maison comme l'eau, le gaz, l'électricité, puis envoyer les relevés à un centre distant via une communication sans fil [11].

- Contrôle des processus industriels :

Dans l'industrie, les **WSN** peuvent être utilisés pour surveiller le processus de fabrication ou l'état de l'équipement de fabrication. Par exemple, les usines chimiques ou les raffineurs de pétrole peuvent utiliser des capteurs pour surveiller l'état de leurs kilomètres de pipelines. Ces capteurs sont utilisés pour alerter en cas de panne [11].

- Agriculture :

Un autre domaine d'utilisation du **WSN** peut être l'agriculture. De nombreuses tâches peuvent être effectuées avec **WSN**, comme la surveillance de l'eau d'alimentation par gravité et la pompe peut être contrôlée à l'aide d'un dispositif d'E / S sans fil [18].

- Surveillance structurelle :

Des capteurs sans fil peuvent être utilisés pour surveiller le mouvement à l'intérieur des bâtiments et des infrastructures tels que les ponts, les survols, les remblais, les tunnels, etc. permettant aux pratiques d'ingénierie de surveiller les actifs à distance sans avoir besoin de visites coûteuses sur site, ainsi que d'avoir avantage des données quotidiennes, alors que traditionnellement, ces données étaient collectées chaque semaine ou chaque mois, à l'aide de visites physiques des sites, impliquant la fermeture de routes ou de rails dans certains cas. Il est également beaucoup plus précis que toute inspection visuelle qui serait effectuée [11].



FIGURE 1.8 – Domaines d'applications des RCSF [19].

1.7 Communication dans les réseaux de capteurs sans fil :

Généralement, dans un réseau de capteurs sans fil, deux types de nœuds sont identifiés logiquement : les nœuds qui transmettent principalement les données capturées par eux-mêmes (nœuds de capteur) et les nœuds qui transmettent des messages à d'autres nœuds (nœuds de relais). Ils sont acheminés du nœud source au nœud cible via des nœuds intermédiaires, créant ainsi une topologie à sauts multiples. Comme le montre la figure 1.9, cette organisation logique implique quatre types de communication [12].

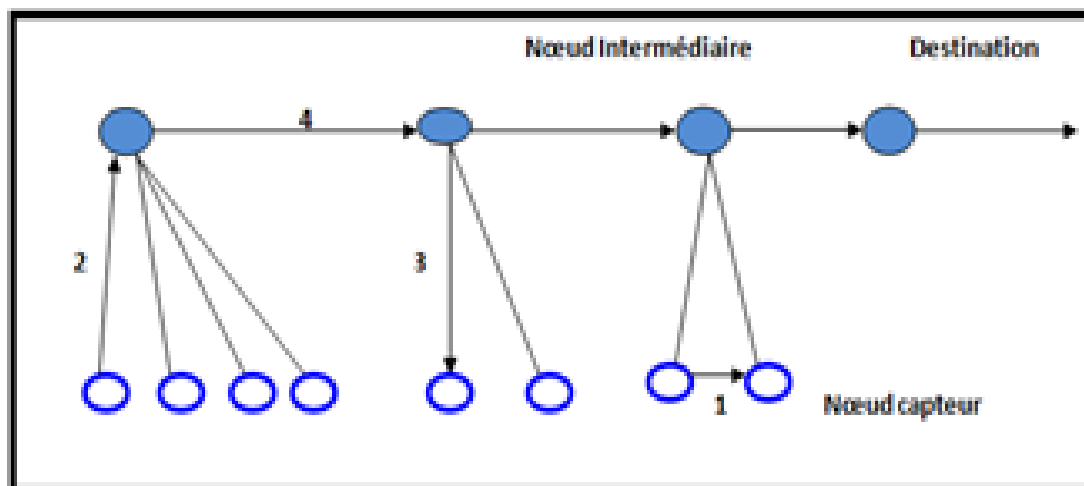


FIGURE 1.9 – Types de communication dans réseaux de capteurs sans fil .

- La communication d'un nœud capteur avec un autre nœud capteur :

Cette communication directe est utilisée pour les opérations locales, comme dans le processus de clustérisations «clustering» ou le processus de création de routes.

- La communication d'un nœud capteur avec un nœud intermédiaire :

Les données capturées sont transmises du capteur au nœud intermédiaire. Cette communication est généralement unicast.

- La communication d'un nœud intermédiaire avec nœud capteur :

La signalisation des demandes et des messages est généralement multicast. Ils sont diffusés par des nœuds intermédiaires (un sous-ensemble de nœuds de capteurs).

- La communication d'un nœud intermédiaire avec nœud intermédiaire :

La communication entre ces nœuds est principalement unicast.

1.8 Consommation d'énergie dans les réseaux de capteurs sans fil :

1.8.1 Les principales opérations dues à la consommation d'énergie

L'énergie consommée par les nœuds capteurs est principalement due aux opérations suivantes : capture, traitement et transmission des données [HCB00] [13].

- Énergie de capture :

L'énergie de capture est dissipée pour effectuer les tâches suivantes : échantillonnage, traitement du signal, conversion analogique-numérique et activation de la sonde de capture. En général, l'énergie capturée ne représente qu'une petite partie de l'énergie totale consommée par le nœud.

- Énergie de traitement :

L'énergie de traitement est divisée en deux parties : Énergie de commutation et énergie de fuite. L'énergie de commutation dépend de la tension d'alimentation et de la capacité totale commutée au niveau du logiciel (en exécutant le logiciel). En revanche, l'énergie de fuite correspond à l'énergie consommée lorsque l'unité de calcul n'effectue aucun traitement. En général, l'énergie de traitement est faible par rapport à l'énergie requise pour la communication.

- Énergie de communication :

L'énergie échangée est divisée en deux parties : la réception de l'énergie et la transmission de l'énergie. L'énergie est déterminée par la quantité de données à transmettre, la distance de transmission et les propriétés physiques du module radio. La transmission d'un signal est caractérisée par sa puissance. Lorsque la puissance d'émission est élevée, le signal aura une plus grande portée et l'énergie consommée sera plus élevée. Notez que l'énergie de communication représente la plus grande partie de l'énergie consommée par le nœud de capteur.

1.8.2 Une comparaison entre la consommation d'énergie dans les RCSF avec et sans énergie renouvelable :

La consommation d'énergie dans les réseaux de capteurs sans fils dépend au genre de capteur.

- En cas des capteurs normal qui utilisent des sources d'énergie limités (des batteries de grande capacité qui durent longtemps), c'est-à-dire n'ont qu'un taux d'auto-décharge négligeable, et qui peuvent fournir efficacement une petite quantité de courant. Donc la durée de vie des capteurs diminuer progressivement jusqu'au la mort du capteur.
- les nœuds de capteurs disposent également d'un dispositif de récupération d'énergie, la recharge des batteries avec l'énergie recueillie à partir de l'environnement, les cellules solaires ou la production d'énergie basée sur les vibrations sont des options envisageables.

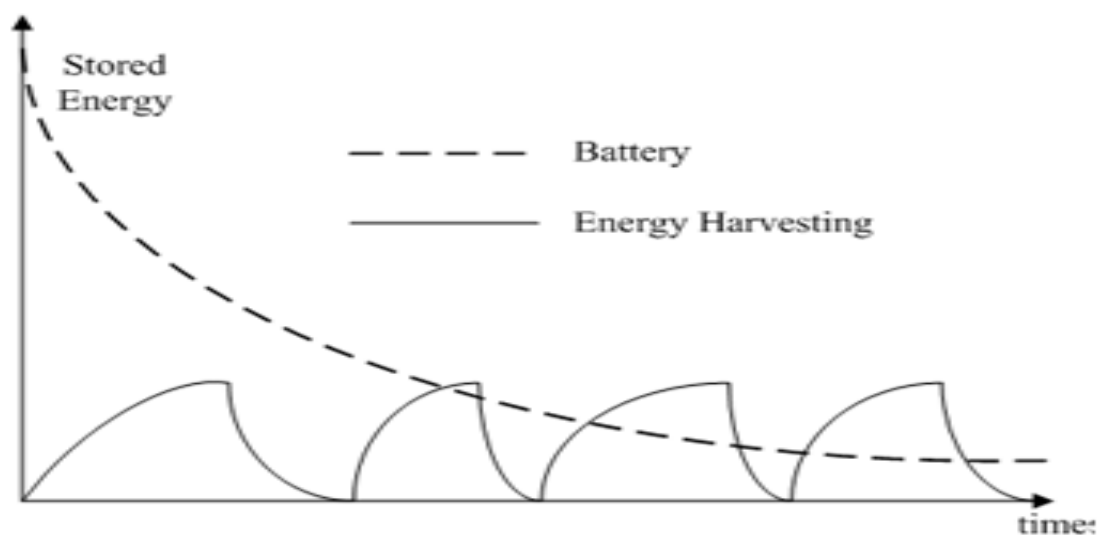


FIGURE 1.10 – Une comparaison de consommation d'énergie entre des **RCSF** avec et sans énergie renouvelable .

1.8.3 Modèle de consommation d'énergie :

Des travaux de recherche ont été effectués dans le domaine des radios de faible consommation d'énergie. Heinzelman et al [20], ont proposé un modèle dans lequel la radio absorbe $E_{elec} = 50 \text{ nJ/bit}$ (où $1 \text{ nJ} = 10^{-9}$) pour mettre en œuvre le circuit émetteur / récepteur, et $E_{amp} = 100 \text{ pJ/bit/m}^2$ (où $1 \text{ pJ} = 10^{-12} \text{ J}$) est utilisé pour l'amplificateur d'émetteur. Par conséquent, pour envoyer des messages s bit à la distance d , la radio consomme [20] :

$$E_{Tx}(s, d) = E_{elec}(s) + E_{amp}(s, d) \quad (1.1)$$

$$E_{Tx}(s, d) = s * E_{elec}(s) + s * E_{amp} * d^n \quad (1.2)$$

Alors le récepteur consomme pour recevoir un message de s bits :

$$E_{Rx}(s) = E_{elec}(s) \quad (1.3)$$

$$E_{Rx}(s) = s * E_{elec} \quad (1.4)$$

Où :

E_{elec} : représente l'énergie de transmission électronique.

E_{amp} : représente l'énergie d'amplification.

n : prendre la valeur 2 ou 4 selon la distance de transmission.

La figure ci-dessous illustre ce modèle de consommation d'énergie.



FIGURE 1.11 – Modèle de consommation d'énergie.

1.9 Conclusion

Dans ce chapitre, nous avons introduit tout d'abord les concepts de base derrière le domaine émergent du réseau de capteurs sans fil (**RCSF**), tels que les composants réseau des réseaux de capteurs sans fil, les techniques de communications dans le réseau. Au même temps, nous avons présenté également un aperçu de ses applications et la méthode de consommation d'énergie dans ce réseau.

Dans le chapitre suivant, nous parlerons de l'aspect routage dans RCSF.

Le routage dans les RCSF

2.1 Introduction

Les réseaux de capteurs sans fil (**RCSF**) sont un type spécial des réseaux informatiques qui ont trouvé de larges applications grâce à leur grande capacité.

Un **RCSF** est composé de nombreux petits nœuds capteurs avec une capacité limitée et une faible consommation d'énergie qui sont capables de détecter les paramètres tels que l'humidité, la température, la pression... de l'environnement.

Malgré les améliorations des **RCSF**, grâce à du grand nombre et de la petite taille des nœuds capteurs, ils nécessitent des batteries de faible puissance. A cause de l'utilisation de ces réseaux dans un environnement difficile et indisponible, il n'est pas possible de recharger ou de changer les nœuds capteurs. Par conséquent, l'un des problèmes importants dans les RCSF est le problème de limitation d'énergie.

Pour cette raison, plusieurs techniques et protocoles de routage ont été proposés pour les **RCSF** afin de maximiser la durée de vie du réseau et de minimiser la consommation d'énergie. Mais, tout réseau de capteurs finira par vider la batterie. Puis l'émergence d'une nouvelle technologie basée sur la récupération d'énergie ambiante, qui permet d'étendre la durée de vie des batteries ou les éliminer complètement.

Dans ce chapitre nous allons présenter les techniques de routage dans les **RCSF** avec et sans énergie renouvelable.

2.2 Le routage dans le RCSF :

Un réseau de capteurs sans fil **RCSF** est constitué d'un certain nombre de petits nœuds de réseau de capteurs et ils sont connectés entre eux via une liaison sans fil qui ne nécessite pas d'infrastructure de réseau fixe. En ce qui concerne la structure des nœuds, la structure des nœuds de **RCSF** elle a une courte portée de transmission. Faible puissance de traitement et capacité de stockage et dispose également de ressources énergétiques limitées. Les protocoles de **RCSF** ont une responsabilité cruciale pour garantir la fiabilité. Communication multi-sauts entre les nœuds. Les **RCSF** nécessitent des algorithmes de routage sophistiqués car l'énergie est le problème central de ces réseaux et des appareils sans fil basse consommation sont nécessaires pour garantir une faible consommation d'énergie.

2.2.1 Définition de routage :

- Le routage est le processus de sélection d'un chemin dans le réseau pour transférer les données de l'expéditeur vers un ou plusieurs destinataires. On parle de routage dans différents domaines : réseaux téléphoniques, réseaux électroniques (comme Internet), réseaux de transmission, etc [10].
- Le routage est un mécanisme par lequel un chemin peut être sélectionné dans le réseau pour acheminer les données de l'expéditeur vers un ou plusieurs destinataires. Le routage est une tâche exécutée dans de nombreux réseaux, tels que les réseaux téléphoniques, les réseaux de données électroniques (comme Internet) et les réseaux de transmission. Ses performances sont importantes dans un réseau décentralisé, par exemple, les informations ne sont pas distribuées par une seule source, mais échangées entre agents indépendants [10].
- Le terme routage se réfère à tous les mécanismes mis en œuvre dans le réseau pour déterminer l'itinéraire des paquets de données du terminal émetteur au terminal récepteur. On distingue généralement deux entités : les algorithmes de routage et les protocoles de routage [10].

2.2.2 Classification des protocoles de routage :

Il existe quatre types de protocoles de routage, les protocoles centrés sur les données (protocoles d'architecture plate et protocoles hiérarchiques), les protocoles de routage géographique, et les protocoles basés sur le QoS.

A- Les protocoles d'architecture plate :

Les données sont envoyées en fonction de leur contenu et non en fonction de l'adresse de destination. Si la requête comprend l'exécution d'une tâche spécifique, le nœud exécutant la requête (telle que la lecture de la température) sera adressé via cette requête. Lorsque la source transmet le paquet de données à la destination, le nœud intermédiaire peut regrouper plusieurs paquets de données d'une certaine manière pour minimiser la consommation d'énergie. Il existe plusieurs types de protocoles dans cette catégorie : inondation, rumeur, SPIN et diffusion directe [14].

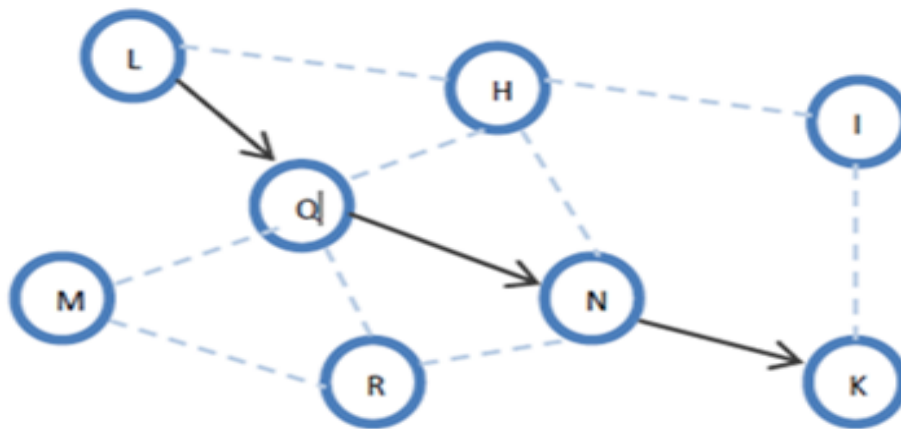


FIGURE 2.1 – Protocole plat.

B- Les protocoles hiérarchiques :

Les nœuds y sont groupés dans des cluster contrôlés par des Cluster Head. La formation des ensembles se fait sur la base de la réserve d'énergie et de la distance de chaque nœud par rapport à des Cluster Head.

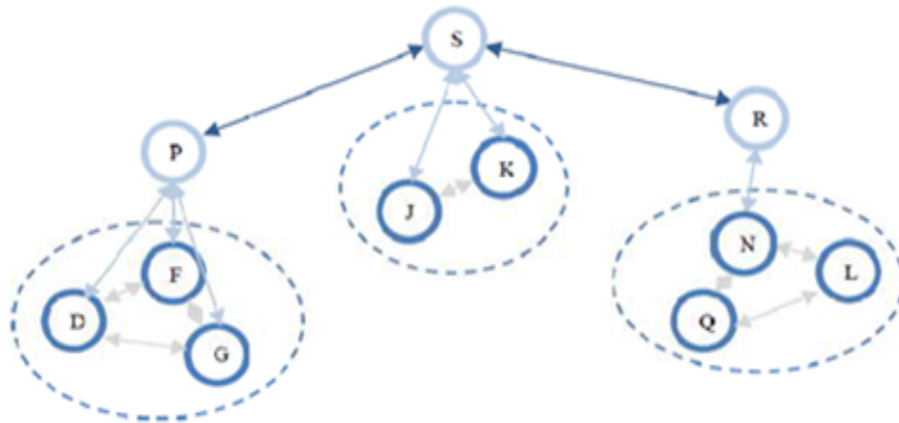


FIGURE 2.2 – Protocole hiérarchique.

C- Les protocoles de routage géographique(basé sur la localisation) :

Plusieurs solutions ont été proposées à partir de cette méthode. Généralement, la méthode utilisée suppose que chaque capteur a une connaissance exacte de sa localisation, grâce à la technologie GPS (où le capteur est équipé d'un récepteur GPS basse énergie, permettant la communication avec les satellites). Sinon, utilisez une méthode de positionnement basée sur les distances estimées par rapport aux autres nœuds. Dans un tel procédé, le positionnement est assuré en estimant la distance entre les deux capteurs, l'estimation dépend des caractéristiques du signal reçu (temps de propagation, atténuation du signal, etc.), ou de l'angle d'arrivée du signal reçu [12].

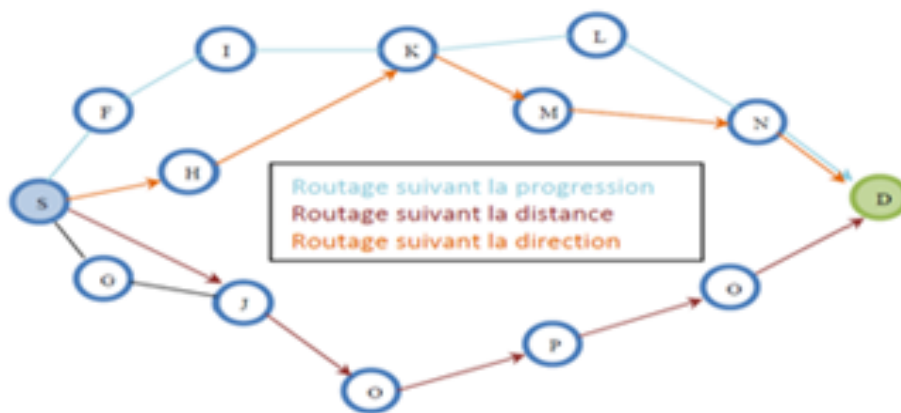


FIGURE 2.3 – Protocole basé sur la localisation..

D- Les protocoles basés sur le QoS (Quality of Service) :

Ces protocoles considèrent non seulement la consommation d'énergie comme un critère pour le choix de l'itinéraire, mais aussi le retard de transmission [14].

2.3 Facteurs de conception de protocoles de routage :

2.3.1 Tolérance aux pannes :

L'attribut de tolérance aux pannes est défini par la capacité du protocole de routage à maintenir sa fonctionnalité lorsqu'un petit nombre de nœuds tombe en panne. Le but de la tolérance aux pannes est d'empêcher toutes les pannes du système, même s'il y a des pannes dans un sous-ensemble de ses composants de base. En raison de la consommation d'énergie rapide et de la pression environnementale, le nœud subit des dommages physiques, de sorte que la fréquence d'occurrence des pannes est plus élevée, de sorte que les pannes peuvent être tolérées. Pour cette raison, le protocole de routage conçu doit atteindre le niveau de tolérance aux pannes requis par l'application. Pour cette raison, en cas de défaillance d'une liaison de communication, le protocole de routage doit continuer à former une nouvelle route entre les nœuds [12].

2.3.2 Consommation d'énergie :

Il convient de souligner que l'efficacité énergétique est un facteur de performance important. C'est pourquoi le concepteur peut ignorer d'autres facteurs lors du développement du protocole pour endommager cette contrainte. Dans la communication et le calcul, des protocoles basés sur des technologies d'économie d'énergie doivent être utilisés [12].

2.3.3 Limitations de capacités des nœuds :

Les nœuds ont une capacité de calcul, de stockage et de communication limitée. Le concepteur d'un protocole de routage doit inclure des opérations simples qui nécessitent peu de calcul et de capacité de stockage [12].

2.3.4 Scalabilité :

L'application de **RCSF** nécessite généralement un déploiement dense de nœuds. Par conséquent, le protocole de routage doit avoir une évolutivité élevée. En d'autres termes, le protocole de routage ne doit pas souffrir de dégradation des performances lorsque les nœuds sont endommagés et que le nombre de nœuds est plus grand [12].

2.3.5 Connectivité :

Il existe un grand nombre de nœuds dans le **RCSF**, ce qui signifie qu'ils sont généralement dispersés au hasard et inégalement répartis sur le site du puits. Cela signifie que certaines zones de l'espace de déploiement peuvent bénéficier d'une meilleure connectivité. Par conséquent, le protocole de routage conçu pour **RCSF** doit avoir la capacité de s'auto-organiser pour s'adapter à la distribution aléatoire des nœuds et à la topologie dynamique du réseau [12].

2.3.6 Hétérogénéité :

En général, les nœuds **RCSF** sont homogènes et ont la même capacité de calcul, de stockage et de ressources énergétiques. Cependant, selon les besoins de l'application, les nœuds peuvent avoir des rôles différents. Par conséquent, les besoins en ressources informatiques, de stockage, de communication et d'énergie peuvent varier d'un nœud à l'autre en fonction des tâches assignées au capteur. Pour résoudre ce problème, certaines applications envisagent la solution d'intégrer des nœuds spéciaux plus puissants que les autres nœuds, et ces nœuds seront chargés d'effectuer les tâches les plus coûteuses (en termes d'énergie). Cependant, l'intégration d'un groupe de nœuds hétérogènes dans un seul réseau imposera de nouvelles restrictions sur le routage des données. Par conséquent, la conception des protocoles de routage doit prendre en compte différents types de nœuds et les contraintes qui en résultent [12].

2.4 Le routage dans les RCSF sans énergie renouvelable :

2.4.1 Objectifs du routage dans les RCSF :

Le principal problème du routage est l'économie d'énergie, qui peut considérablement prolonger la durée de vie du **RCSF**. Un bon routage doit pouvoir acheminer les informations selon les conditions suivantes [9] :

- Une latence faible.
- En utilisant peu d'énergie, donc en utilisant les meilleures liaisons de données.
- En minimisant la charge de réseau, afin de maximiser la qualité total d'informations.
- pouvant être transmises sur le réseau dans son ensemble.
- En étant faible, et en particulier, en étant tolérant aux pannes de nœuds du réseau.

2.4.2 Quelques protocoles de routage utilisés dans les RCSF sans énergie renouvelable :

- LEACH (Low Energy Adaptive Clustering Hierarchy) :

Low Energy Adaptive Clustering Hierarchy (LEACH), est le premier protocole de routage hiérarchique dans les réseaux de capteurs sans fil et il est considéré parmi les meilleurs protocoles qui sont performants en termes d'énergie. L'idée est de diviser les nœuds capteurs de réseau en deux catégories : les Cluster Heads **CH** et les nœuds membres CM (cluster Member), ou chaque cluster possède un Cluster Head **CH**, qui joue le rôle de router, ceci réduire la consommation énergétique puis que ce sont les Cluster Head **CH** qui agrègent et compressent les données détectés et les envoient à la station de base BS. Le protocole se déroule en round (ou tour). Chaque round se compose de deux phases : construction et communication.

a- Phase de construction :

Divisée en trois sous phases : La sélection de **CH**, formation de cluster et assignation de TDMA (Time-division multiple Access) [15].

A- La sélection de CH :

Dans l'algorithme LEACH, certains des nœuds sont appelés Cluster Head. Le travail du CH est de collecter les données de leurs nœuds environnants et de les transmettre à la station de base. Les **CHs** peuvent être choisies de manière stochastique (basée au hasard). Sur cet algorithme suivant l'équation [15].

$$T(S_i) = \frac{p}{1 - (p * (r \bmod (1 \div p)))} \quad (2.1)$$

$T(S_i)=0$ Sinon

Où :

p : pourcentage souhaité de nœud à devenir un **CH**.

r : Numéro de tour courant.

G : l'ensemble des nœuds qui n'ont pas été élus **CH** pendant les $1/p$ tours précédents.

Si S_i inférieur à $T(S_i)$, alors ce nœud devient un Cluster Head, l'algorithme est conçu de sorte que chaque nœud devienne un **CH** de cluster au moins une fois.

B- Formation de cluster :

À la fin de processus de sélection, chaque **CH** émet un message de notification afin de permettre aux nœuds non **CH** de décider l'appartenance ou non à son cluster. La décision se fait sur la base de la force du signal reçu (RSSI), le **CH** avec le signal le plus fort sera choisi [23].

- Assignation de TDMA :

Chaque **CH** crée un planning de schedule TDMA et le transmet vers tous ses membres. Ce schedule définit le temps (slot) réservé pour chaque membre de cluster pour transmettre ses données au **CH** [23].

b- Phase de communication :

Qui utilise le schedule TDMA, les membres transmettent leurs données capturées dans leurs propres slots. Cela leur permet de fermer leurs interfaces de communication en

dehors des emplacements réservés pour économiser de l'énergie. Ensuite, ces informations sont agrégées pour être envoyées à la station de base. La communication entre le Cluster Head et la station de base se fait directement, c'est-à-dire que le Cluster Head adapte son émetteur radio pour atteindre directement la station de base [23].

Interférences entre les clusters :

Afin de réduire les interférences entre les signaux transmis par des membres de différents clusters, chaque **CH** sélectionne de manière aléatoire un code dans la liste de codes de propagation CDMA et notifie ses membres de son cluster afin qu'il puisse être utilisé dans sa transmission. Chaque **CH** utilise ces codes pour identifier les signaux de ses membres.

- Le protocole de routage LZHREP (Level and Zone based Hierarchical Routing and Energy optimization Protocol) :

Le protocole Hiérarchique de routage et d'optimisation d'énergie basé sur les niveaux et les zones, proposé par [2], qui est l'amélioration de ces deux protocoles (ECBDA et DECSA), il utilise la technique de clustérisation, qui découpe le réseau en différentes couches afin d'assurer le routage multi-sauts entre les nœuds du réseau et la **SB**. Puis partitionner chaque couche en plusieurs clusters de différentes tailles. Afin de prolonger la durée de vie de réseau.

A- La formation des niveaux :

Dans cette étape, la SB sépare le réseau en L niveaux, en calculant la distance (équation suivante) entre la **SB** et le nœud le plus éloigné comme suite :

$$\alpha = \max \forall_{i=1}^N d(sn_i, SB) \quad (2.2)$$

Où :

$d(sn_i, SB)$: est la distance entre la station de base et le nœud sn_i .

α : un angle dans la zone .

Le nombre de niveau L dépend de la distance maximale α et de la portée radio des nœuds

Tr :

$$L = \frac{\alpha}{T_r} \quad (2.3)$$

Où :

Tr : est la portée du signal d'un nœud capteur qui est supposée pareille et identique pour tous les nœuds de capteurs du réseau.

Pour partitionner le réseau en L niveau la **SB** envoi L messages " Hello " qui contient l'identifiant de la couche, et elle est basée sur la variation de la puissance du signal suivant le nombre de sauts (le saut représente la portée de transmission d'un nœud capteur, tel que les nœuds distants de n sauts appartiendront au n 'ième niveau), après avoir recules différentes distances de la **SB**, par les nœuds, ils calculent la distance de la **SB** et mettre à jour leurs identificateur du niveau ID_{niveau} à l . A la fin de cette étape, le réseau est partitionné en L niveaux, ou chaque nœud possède leur propre ID_{niveau} .

B- Formation des zones :

Chaque niveau sera divisé en un ensemble de K clusters selon une technique de découpage en zones utilisée dans cette étape, dont la couche la plus proche de la station de base SB (la couche inférieure) aura le maximum de K clusters. Le nombre de clusters K est proportionnel à la densité du niveau n (l), et proportionnel au nombre de niveaux L .

$$Kl = \frac{n(l)}{1 * L} \quad (2.4)$$

Après avoir calculé K , la **SB** balayera horizontalement avec un angle $\alpha=180/K$ chaque niveau du réseau. La **SB** envoie un paquet de données à chaque zone qui compris l'identifiant de chaque zone a tous les niveaux du réseau, donc chaque cluster aura les mêmes ID_{niveau} et mêmes $ID_{cluster}$.

C- Election de cluster head CH :

Après avoir formé les clusters (niveaux, zones), la **SB** sélectionne un Faux Cluster Head **FCH** aléatoirement dans chaque cluster, et qui a comme rôle l'élection de **CH** de cluster. Le **FCH** envoi des messages électoraux à tous les membres de son cluster, ce dernier répondez avec un message contenant son énergie restante. Ensuite le **FCH** calcule le seuil d'élection $K(i)$ pour chaque nœud de son cluster. Le nœud avec le plus grand

$\mathbf{K(i)}$ sera sélectionné comme **CH** du cluster associé. $\mathbf{K(i)}$ est représenté comme suit :

$$K(i) = \frac{E_n(i)}{d_0(i)} \quad (2.5)$$

Où :

$\mathbf{K(i)}$: est le seuil des élus **CH**.

$\mathbf{En(i)}$: est l'énergie résiduelle du nœud i .

$\mathbf{d0(i)}$: est la distance moyenne entre le nœud i et tous les autres nœuds du cluster.

D- Phase de routage :

Communication inter-cluster et intra-cluster :

Chaque Cluster Head est responsable de la communication au sein de son cluster et gère les informations de routage qui lui permettant de joindre la station de base. Pour d'atteindre la station de base, les Clusters Head communiquent directement avec la station de base, sinon, d'autres Clusters Head sont utilisées comme communication multi-sauts de nœuds de relais.

E- Phase de maintenance :

Afin de s'adapter aux changements de la topologie du réseau, les mises à jour du cluster seront effectuées de manière dynamique pour identifier les nouveaux nœuds ajoutés ou les nœuds qui ont disparu (nœuds qui consomment de l'énergie). En revanche, si le Cluster Head garde son état le plus longtemps possible, même s'il n'a pas le poids maximum dans son propre cluster, il perdra sa fonction une fois la batterie épuisée.

2.5 Le routage dans les RCSF avec énergie renouvelable :

2.5.1 Quelques algorithmes de routage dans ISEH :

- Harvesting Aware Speed Selection (HASS) :

C'est un algorithme proposé pour maximiser les réserves d'énergie des nœuds dans le réseau tout en répondant aux exigences de performances des applications dans **RCSF**.

HASS est une méthode basée sur l'époque pour le contrôle de l'énergie dans les **RCSF** à performances limitées qui exploite la récupération d'énergie. Il utilise une technique de mise à l'échelle de tension dynamique pour économiser l'énergie de calcul en diminuant simultanément la tension et la fréquence d'alimentation du processeur, et utilise également la technique de mise à l'échelle de la modulation dynamique pour économiser l'énergie de communication en diminuant le niveau de la modulation de radio.

Le problème HASS est formulé comme un problème d'optimisation et résolu avec des algorithmes centralisés et distribués. Dans HASS, le niveau d'énergie T_i d'un nœud V_i à la fin d'une époque est donné dans l'équation suivante [5] :

$$T_i = T_i^{init} + P_i * S - \left[\frac{S}{\pi}\right] * e_i^c \quad (2.6)$$

Où :

T_i^{init} : est le niveau d'énergie initial de V_i .

S : est la longueur de l'époque.

S/π : sont des cadres d'une époque.

e_i^c : est la consommation d'énergie.

P_i : est la puissance récoltée de l'époque.

Les résultats de la simulation montrent que les performances HASS sont significativement plus élevées que les approches de base.

-Adaptive Framework for Energy Prediction (AFEPF) :

Ren et al ont conçu un algorithme centralisé et distribué pour un problème de couverture cible sensible à la qualité dans un réseau de capteurs avec la récupération d'énergie. Un plan de récupération d'énergie devrait être parfait tout au long des opérations du réseau. Cependant, une telle hypothèse est considérée comme problématique dans des applications réalistes. Pour cette raison, les auteurs ont suggéré un cadre adaptatif pour les fluctuations de prédiction d'énergie (AFEPF). Il vise à planifier les activités des capteurs par le concept d'intervalle dynamique. L'AFEPF comprend des tranches de temps séquentielles qui ne dépassent pas L . Il compte le nombre de tranches à atteindre avant qu'une autre recharge ne se produise. La charge de l'énergie solaire dépend des conditions météorologiques et la longueur de l'intervalle est planifiée de manière adaptative par la

justesse de la concentration énergétique. Ainsi, toute la période opérationnelle de L intervalles de temps se compose de quelques intervalles. Ensuite, une suggestion d'heuristique gourmande (Greedy–

Heuristic) est appliquée dans chaque intervalle. La précision de l'énergie prévue $O_i(\mathbf{v})$ d'un capteur \mathbf{v} appartient $\mathbf{V}(\mathbf{I}_i)$ dans un intervalle \mathbf{I}_i est formulée un [5] :

$$O_i(v) = \frac{Qv - Q'v}{Q'v} \quad (2.7)$$

Où :

$Q\mathbf{v}$ et $Q'\mathbf{v}$: Représentent les valeurs réelles et prévues de l'énergie récoltée. L'exactitude de la prévision moyenne de noeuds dans l'intervalle est désigné par l'énergie Figure 1.6 diagramme schématique du système ISEH de l'exactitude de prédiction de l'intervalle \mathbf{I}_i . Il s'agit de l'exactitude moyenne des prévisions d'énergie dans les noeuds actifs dans l'intervalle donné par :

$$O_i = \sum_{v \in V(I_i)} \left[\frac{O_i(v)}{V(I_i)} \right] \quad (2.8)$$

Les résultats de la simulation expérimentale montrent que les solutions proposées sont prometteuses.

2.6 Conclusion

Dans ce chapitre nous avons défini le routage et on a fait des classifications des protocoles de routage, puis on a étudié quelque protocoles de routage classique dans les réseaux de capteurs sans fil qui utilisent l'énergie efficace on présentant ces objectifs, aussi on a étudié les nouveaux protocoles de routage dans les **RCSF** basé sure la récupération d'énergie ambiante.

Le protocole EH-LCHREP (Energy Harvesting Layer and Cluster based Hierarchical Routing and Energy Optimization Protocol) :

3.1 Introduction

Après avoir étudié dans le premier et le deuxième chapitres les réseaux de capteurs sans fil avec et sans énergie renouvelable et les techniques de routage dans les deux, alors dans ce chapitre, nous allons présenter la conception de protocole de routage qu'on souhaite mettre en place. Nous commençons par une présentation générale de notre protocole puis nous présentons en détail les différentes étapes de sa conception.

3.2 Description de protocole :

3.2.1 Motivation

Plusieurs protocoles de routage efficace en énergie ont été proposés pour les réseaux de capteurs sans fil. L'objectif principal de la plupart de ces protocoles est se trouve des routes qui sont économes en énergie, de sorte que la durée de vie du réseau soit maximisée. Pour atteindre cet objectif de nombreux protocoles ont été développés en utilisant

plusieurs stratégies pour obtenir des chemins, certains de ces protocoles utilisent le clustering. . .

Dans une architecture basée sur le clustering, les nœuds capteurs sont divisés en plusieurs groupes appelés clusters. Chaque cluster a un chef connu sous le nom Cluster Head (**CH**). Tous les nœuds capteurs détectent les données locales et les envoient à leur **CH** correspondant. Les CH agrègent ensuite les données locales et les envoient finalement à la station de base (**BS**) directement ou via d'autres **CH**.

Dans ce chapitre, nous proposons un nouveau protocole de routage hiérarchique appelé **EH-LCHREP** (Energy Harvesting Layer and Cluster based Hierarchical Routing and Energy Optimization Protocol) qui est l'amélioration de protocole de routage LCHREP [16] l'amélioration de protocole LZHCRP [2]. Ce protocole vise à minimiser la consommation d'énergie pour les réseaux de capteurs sans fil. Le LCHREP utilise des clusters formés sur une couche et utilise également le multi-hop entre ces différentes couches. Il permet également de réduire le nombre de nœuds défaillants, et donc de garder plus de nœuds en vie après l'envoi des paquets et pour améliorer la longévité de réseau de capteurs sans fil, basées sur la récupération d'énergie.

3.2.2 Hypothèse :

Nous allons baser sur le modèle du réseau suivant, pour appliquer les propositions rédigé dans notre protocole :

- Les nœuds capteurs sont tous identiques (même énergie initiale, mêmes capacités de stockage et traitement de données).
- Tous les nœuds sont rechargeables.
- Les nœuds sont répartis aléatoirement dans la zone de capture et ils sont fixes.
- La station de base est considérée comme une ressource qui est non limité ni épuisable.
- La mort de chaque capteur n'est causée que par l'épuisement de son énergie.
- Chaque capteur a un identifiant unique et la station de base connaît tous les positions des capteurs.
- Les nœuds capteurs peuvent gérées leurs puissance d'émission.
- La station de base est capable d'envoyer des signaux unidirectionnels avec un angle α .

3.2.3 principe de fonctionnement :

Dans le but de maximiser la durée de vie de réseau plusieurs protocoles de routage ont été proposés, mais le problème principal de routage dans les **RCSF** a été dans la limitation d'énergie de capteurs, donc comment prolongé la durée de vie du réseau ? Et comment réduire la consommation d'énergie ? Sont des solutions de ce problème.

Idéalement, les nœuds capteurs disposent également d'un dispositif de récupération d'énergie, la recharge des batteries avec l'énergie recueillie à partir de l'environnement comme les cellules solaires.

Notre protocole qui est basé sur le clustering et prend en compte la récupération d'énergie ambiante qui permet d'étendre la durée de vie des batteries, et avec les techniques de minimisation de consommation d'énergie qu'on adopte, ces mécanismes permettent de diviser le réseau en couches pour assurer le routage multi-saut entre les différentes couches.

Aussi pour diviser les couches en zones, il penche une stratégie dépend du nombre des nœuds capteurs dans chaque niveau, dans lequel le niveau le plus proche à la station de base a le plus nombre de clusters.

Le fonctionnement de notre protocole est établi en trois phases qui sont :

- La phase d'initialisation.
- la phase de routage des données

Ces différentes phases seront détaillées dans ce qui suit.

Phase d'initialisation :

Dans cette phase on a utilisé l'architecture proposée dans [16] qui consiste à découper le réseau en couches pour assurer le routage multi-saut, puis partitionner ces niveaux en zones (Clusters), ou chaque niveau contient un nombre des zones dépend de numéro de niveau et le nombre de capteurs dans le niveau, notons que cette phase est exécutée une seule fois durant le processus de routage.

Élection de Cluster Head :

Après avoir devisé le réseau en couches et zones, la station de base doit de mettre en place un temporaire cluster head **TCH** aléatoirement dans chaque cluster, son rôle est

d'élire le **CH**. Après l'élection d'un **CH** le **TCH** devient un normal nœud capteur. Le **TCH** envoie un message d'élection à tous les membres de son cluster, ces derniers répondent avec un message contenant leurs énergies résiduelles et sa distance par rapport à la station de base, le **TCH** calcule la distance entre les différents capteurs de même cluster, On définit la formule d'élection $Electe_{node}$ comme suit [30] :

$$Electe_{node} = \frac{E_{node_i}}{DisAvecAutr_i} \quad (3.1)$$

Où :

E_{node_i} : est le l'énergie résiduelle du nœud i.

$DisAvecAutr_i$: est la distance entre le nœud i et les autres du même cluster.

Lors de la réception des messages par le **TCH**, ce dernier désigne comme **CH** le nœud ayant le score maximum $Electe_{node}$, en cas d'égalité, l'identité minimum des nœuds est prise en compte. Et le modèle de consommation d'énergie dans chaque **TCH** dans l'élection est suit :

$$EFCH_i = EFCH_i - EconstTCH \quad (3.2)$$

Où :

$EFCH_i$: est l'énergie résiduelle du **FCH**.

EconstTCH : l'énergie consommé par le temporaire Cluster Head.

L'algorithme de sélection de Cluster Heads est définir comme suit :

Déclaration :

Node : est le nœud capteur.

NbrNod : nombre total des nœuds.

L : le nombre de niveaux dans le réseau.

j , k : des variable d'entier.

IdNode , IdCH , IdtFCH , IdFCH : des variable entier qui défini le nombre d'identifiants des nœuds.

E_{max} , E_{consom} , E_{resevie} ,E_{constTCH} , E_{node} : des variables réel sont définies les différents énergie de chaque type de noueds.

List[j] : tableau d'entier qui stocke le nombre de nœuds dans chaque case pour chaque niveau.

Tab[] : tableau d'entier qui stocke le nombre de clusters dans chaque case pour chaque niveau.

SizeMsgReceve : variable réel qui définit la taille de message à recevoir.

dead : variable entier qui définit les nœuds qui sont morts.

Type={"TCH","CH","N"} : variable string pour les types des nœuds.

tfCH : tableau des entiers qui stockent les temporaires Cluster Head.

Algorithme 1 : Algorithme d'élection de Cluster Head

Debut

```

1 : IdFCH=tfCH(k)
2 : Emax=0 ,IdCH=0
3 : Pour j allant de 1 à NbrNod faire
4 :   IdNode = Tab[k,j]
5 :   Si ( IdNode >0)and (List[IdNode-1].Type!= "TCH")and(List[IdNode-1].dead=0)
6 :     EconsTCH = Eresevie * SizeMsgReceved
7 :     Si(List[IdtFCH-1].energie - EconsTCH > 0.01)
8 :       List[IdtFCH-1].energie=List[IdtFCH-1].energie - EconsTCH
9 :       Econsom=Econsom + EconsTCH
10 :      Enode =List[IdNode-1].energie / List[IdNode-1].DisAvecAutr
11 :      Si(Enode > Emax)
12 :        Emax=Enode
13 :        IdCH=List[IdNode-1].id
14 :      Sinon
15 :        List[IdtFCH-1].Type="N"
16 :        ChoixFCH()
17 :      retourne
18 :Pour j allant de 0 à NbrNod faire
19 :  IdNode=Tab[k,j]

```

```

20 :   Si(IdNode >0)and (IdNode!=IdCH) and (List[IdNode-1].dead =0)
21 :     List[Tab[k,j]-1].ItCH =IdCH
22 :     List[Tab[k,j]-1].ItFCH =-1
23 :     ch[List[IdCH-1].z , List[IdCH-1].cluster] = IdCH
24 :List[IdCH-1].Type="CH" ; tCH[K] = IdCH
25 :Fin

```

Phase de routage des données :

C'est la phase dont l'objectif est d'envoyer les données agrégées par chaque capteur à la station de base. Dans cette phase, On a proposé d'utiliser deux threads pour faire fonctionner le capteur, le fonctionnement de threads est de découper le processeur en deux threads.

- Le premier thread_A pour la phase de routage des données.

- Le deuxième thread_B pour la charge par l'énergie renouvelable des capteurs.

L'algorithme de création de threads_A lance un thread qui se charge du routage. L'algorithme de ce thread est représenté comme suit :

Algorithme 2 : Algorithme de Thread A

Debut

```

1 :   global Nodes //déclaré les nodes (capteurs) comme variable globale
2 :   Routing=Route(Nodes)
3 :   retourne

```

Dans l'algorithme de thread_B, On a utilisé la formule suivante [22] l'algorithme choisi une variable aléatoire depuis la formule $P(s)$, et il la considérer comme une valeur de chargement de nœud dans un temps t .

$$P(s) = |0.9 * R(t) * \cos\left(\frac{t}{7 * \pi}\right) * \cos\left(\frac{t}{10 * \pi}\right)| \quad (3.3)$$

Ou :

P(S) : proedcture source /temps, borne a une valeur maximale [0,0.9].

R(t) : les valeurs aléatoires entre 0 et 1(random).

Déclaration :

EH : liste des l'énergie renouvelable qui prend des variable aléatoire de a.

E : liste des l'énergie résiduelle de capture.

a : variable réel de prodecture source.

arret : variable booléen qui défini le mode de thread.

Xmax, Ymax : des valeurs réel constant.

Algorithme 3 :Algorithme de Thread B

```

arret = False
Eh , X = list()
Xmax = 500, Ymax = 0.00075
1 :Pour t=arange(1, Xmax, 0.1)
2 :   a = abs (Ymax * random() * cos (t / (7 *  $\pi$ )) * cos (t / (10 * $\pi$ )))
3 :   Eh.append(a)
4 :   X.append(t)
5 :cont=Eh.len()
6 :Pour Nodes=1 a NbrNod faire
7 :   j = 0
8 :   Tan que True :
9 :     Si (arret)
10 :       raise ValueError("arrêt demandé")
11 :     Sinon
12 :       Pour Nodes=1 a NbrNod faire
13 :         ValEH =random.randrange(0, cont)
14 :         E=Liste[i].energie + Eh[ValEH]
15 :         Si(E > Einit)
16 :           Liste[i].energie = Einit
17 :         Sinon
18 :           Liste[i].energie = E
19 :except

```

```

20 :ecrire("")
21 :ecrire("%s" % sys.exc_info()[1])

```

La phase de routage :

il fait le routage 100 fois, à chaque fois il stock l'énergie total des nœuds, et les nœuds morts :

Déclaration :

TotalDead : liste qui stocker les nœuds mort.

TotalE : liste qui stocker le total d'énergie des nœuds.

Algorithme 4 : Algorithme de routage

Debut

Tanque $i < 100$ faire

```

1 :   Routing(Nodes, clusterModel)
2 :   TotalE.append(Econsomme)
3 :   TotalDead(dead)
4 :   Econsomme = 0
5 :   dead = 0
6 :   i = i + 1
7 :   time.sleep(0.1)

```

L'algorithme de routage comporte plusieurs étapes qui sont ci-dessous :

1- Transmission des données de capteur :

Dans cette étape chaque nœud capteur calcule son énergie si elle est supérieure à 0, puis il va calculer l'énergie qui va consommer pour envoyer ces données agréger. Notons que l'énergie d'envoi dépend de la taille des données à agréger et la distance entre le nœud et **CH**.

Après il teste si l'énergie calculé est supérieur à zéro il envoie les données. Sinon, l'algo-

rithme va le considère comme un nœud mort. Et le modèle de consommation d'énergie dans chaque nœud capteur ESn_i dans cette transmission est comme suit :

$$E_{node_j} = E_{node_j} - E_{consNode} \quad (3.4)$$

Où :

E_{node_j} : est l'énergie résiduelle dans le capteur j.

$E_{consNode}$: L'énergie consommé par le capteur qui envoi les données.

Déclaration :

ch : variable entier qui définis Cluster Head.

SB : est la station de base, **dBs** : distance à la station de base .

tCH : table d'entier qui stocker les temporaire Cluster Head.

Dmin : la distance minimale.

i, m, n, c : des variable d'entier.

TC(k) : variable pour stocker le nombre de clusters pour chaque niveau.

fch, clusterCH :des variable entier sont définis les type de cluster head.

ItCH : variable entier qui défini le nombre d'identifiant de nœuds temporaire CH.

d : la distance.

Eaggr : variable réel qui définies l'énergie d'agrégation de donné.

NumCluster : numero de cluster.

nbrNodeVivant : varieble entier pour le nombre des nœuds vivant.

SizeMsgSend : variable réel qui défini la taille de message qui envoyé.

Algorithme 5 : Algorithme de transmission les données des capteur

Debut

1 :# transmesson data des noeud capteur

2 :Pour j allant de 1 à NbrNod faire

3 : **Si** (list[j].Type!="CH")and (List[j].ItCH != -1) and(List[IdNode-1].dead=0)

4 : $d = \sqrt{(\text{Liste}[\text{Liste}[j].\text{ItCH} - 1].x - \text{Liste}[j].x)^2 + (\text{Liste}[\text{Liste}[j].\text{ItCH} - 1].y - \text{Liste}[j].y)^2}$

5 : $E_{consNode} = E_{transfer} * d * \text{SizeMsgSend}$

```

6 :      Si(Liste[j].energie - EconsNode > 0)
7 :          Liste[j].energie = Liste[j].energie - EconsNode
8 :          Econsomme = Econsomme + EconsNode
9 :      Sinon
10 :          Liste[j].dead = 1
11 :          dead += 1
12 :          DisAvecAutre(Nodes,Liste[j].NumCluster)
13 :          Econsomme = Econsomme+Liste[j].energie
14 :          Liste[j].energie = 0
15 :FIN

```

2- Réception des données par le Cluster-Head :

Dans cette étape l'algorithme va parcourir tous les Cluster Head de réseau qui existent, puis il va tester l'énergie si elle est suffisante pour la réception des messages de tous les nœuds vivants. Si l'énergie du Cluster Head est supérieure à 10% il va recevoir les données. Sinon il va jouer le rôle de **TCH** et sélectionner un autre **CH**.

Et le modèle de consommation d'énergie dans chaque Cluster Head ECH_i est dans cette transmission est exprimé par cette formule :

$$ECH_i = ECH_i - EconsCH \quad (3.5)$$

Où :

ECH_i : est l'énergie résiduelle dans le Cluster Head i .

$EconsCH$: est la consommation d'énergie pour la réception de message.

Algorithme 6 : Algorithme de réception de message par le Cluster Head

Debut

```

16 :#reception Cluster Head
17 :Pour k allant de 1 à TC faire
18 :  IdCH = tch[k]
19 :  Pour j allant de 0 à NbrNod

```

```

20 :   IdNode=Tab[k, j]
21 :   Si(IdNode > 0 )and (IdNode!= IdCH) and (Liste[IdNode - 1].dead = 0)
22 :     EconsCH = Ereceive*SizeMsgReceved
23 :     Si(Liste[IdCH - 1].energie - EconsCH > 0.1)
24 :       Liste[IdCH - 1].energie = Liste[IdCH - 1].energie - EconsCH
25 :       Econsomme = self.Econsomme + EconsCH
26 :     Sinon
27 :       clusterCH=Liste[IdCH - 1].NumCluster
28 :       tfch[clusterCH]=IdCH
29 :       Liste[IdCH - 1].type ="TCH"
30 :     Pour m allant de 1 à NbrNod
31 :       IdNode=Tab[clusterCH, m]
32 :       Si(IdNode> 0 )and (IdNode!= IdCH) and (Liste[IdNode - 1].dead = 0)
33 :         Liste[IdNode - 1].ItCH = -1
34 :         Liste[IdNode - 1].ItFCH = IdCH
35 :         ch[Liste[IdCH - 1].z, Liste[IdCH - 1].cluster] = -1
36 :         fch[Liste[IdCH - 1].z, Liste[IdCH - 1].cluster] = IdCH
37 :         NodSendE(Nodes,clusterCH)
38 :         ElectCH(Nodes, clusterCH)
39 : Fin

```

3- Agrégation des données :

Après la réception des données, le Cluster Head doit agréger les données reçus de chaque nœud, puis il va transmit au Cluster Head de niveau supérieur.

Algorithme 7 : Algorithme de d'agrégation

Debut

```

40 :# agrégation
41 :Pour k allant de 1 à TC faire
42 :   nbrNodeVivant=0
43 :   IdCH = tch[k]

```

```

44 : Pour j allant de 1 à NbrNod faire
45 :   IdNode=Tab[k, j]
46 :   Si(IdNode> 0 ) and(IdNode!= IdCH) and (Liste[IdNode - 1].dead = 0)
47 :     nbrNodeVivant += 1
48 :   EconsCH =Eaggr * nbrNodeVivant * SizeMsgReceved
49 :   Si(Liste[IdCH - 1].energie - EconsCH)> 0.1)
50 :     Liste[IdCH - 1].energie = Liste[IdCH - 1].energie - EconsCH
51 :     Econsomme = self.Econsomme + EconsCH
52 :   Sinon
53 :     clusterCH =Liste[IdCH - 1].NumCluster
54 :     tfch[clusterCH] = IdCH
55 :     Liste[IdCH - 1].type = "TCH"
56 :   Pour m allant de 0 à NbrNod
57 :     IdNode = Tab[clusterCH, m]
58 :     Si(IdNode> 0 ) and(IdNode!= IdCH) and (Liste[IdNode - 1].dead = 0)
59 :       Liste[IdNode - 1].ItCH = -1
60 :       Liste[IdNode - 1].ItFCH = IdCH
61 :       ch[Liste[IdCH - 1].z, Liste[IdCH - 1].cluster] = -1
62 :       fch[Liste[IdCH - 1].z, Liste[IdCH - 1].cluster] = IdCH
63 :     NodSendE(Nodes,clusterCH)
64 :     ElectCH(Nodes,clusterCH)
65 : Fin

```

4- Transmission inter-Cluster :

Après avoir agrégé les données dans les **CHs**, chaque CH envoie ses données au CH de son niveau voisin inférieur et proche de la SB, tandis que les CHs du premier niveau communiquent les données reçues directement à la **SB**. La **SB** assigne à chaque **CH** les **CHs** auxquels il peut communiquer ses données.

Algorithme 8 : Algorithme de transmission les données di CH à CH ou SB**Debut**

```

66 :# transmesson de CH à CH ou BS
67 :Pour k allant de 1 à TC faire
68 :   Dmin = 1000000 ; IdCH =tch[k]
69 :   c =Liste[IdCH - 1].cluster ; L = Liste[IdCH - 1].z
70 :   Si(L!= 0)
71 :     n = k[L - 1]
72 :     Si(n!= 0)
73 :       Pour h allant de 1 à n faire
74 :         Si(ch[L - 1, h] != 0)
75 :           xCHs = Liste[ch[L - 1, h] - 1].x ; yCHs = Liste[ch[L - 1, h] - 1].y
76 :           x = Liste[IdCH - 1].x ; y = Liste[IdCH - 1].y
77 :           d = sqrt((xCHs - x) ** 2 + (yCHs - y) ** 2)
78 :           Si(d < Dmin)
79 :             Dmin = d
80 :           EconsCH = Ereceive * Dmin * SizeMsgReceived
81 :           Si((Liste[IdCH - 1].energie - EconsCH) > 0.1)
82 :             Liste[IdCH - 1].energie = Liste[IdCH - 1].energie - EconsCH
83 :             Econsomme = self.Econsomme + EconsCH
84 :           Sinon
85 :             clusterCH=Liste[IdCH - 1].NumCluster
86 :             tfch[clusterCH]=IdCH
87 :             NodeListe[IdCH - 1].type ="TCH"
88 :           Pour m allant de 1 à NbrNod faire
89 :             IdNode = Tab[clusterCH, m]
90 :             Si(IdNode> 0 )and(IdNode!= IdCH) and (Liste[IdNode - 1].dead = 0)
91 :               NodeListe[idnode - 1].ItCH = -1
92 :               Liste[idnode - 1].ItFCH = IdCH
93 :               ch[Liste[IdCH - 1].z, Liste[IdCH - 1].cluster] = -1

```

```

94 :          fch[Liste[IdCH - 1].z, Liste[IdCH - 1].cluster] = IdCH
95 :          NodSendE(Nodes, clusterCH) ; ElectCH(Nodes, clusterCH)
96 :  Sinon // Envois à la station de base
97 :          dBs =sqrt((x - Liste[IdCH - 1].x)**2 +(y - Liste[IdCH - 1].y)** 2)
98 :          EconsCH = Etransfer * dBs * .SizeMsgReceved
99 :          Si((Liste[IdCH - 1].energie - EconsCH) > 0.1)
100 :          Liste[IdCH - 1].energie = Liste[IdCH - 1].energie - EconsCH
101 :          Econsomme = self.Econsomme + EconsCH
102 :  Sinon
103 :          clusterCH =Liste[IdCH - 1].NumCluster
104 :          tfch[clusterCH] = IdCH ; Liste[IdCH - 1].type = "TCH"
105 :  Pour m allant de 1 à NbrNod faire
106 :          IdNode = Tab[clusterCH, m]
107 :          Si(IdNode > 0 )and(IdNode != IdCH) and (Liste[IdNode - 1].dead = 0)
108 :          Liste[IdNode - 1].ItCH = -1 ; Liste[IdNode - 1].ItFCH = IdCH
109 :          ch[Liste[IdCH - 1].z, Liste[IdCH - 1].cluster] = -1
110 :          fch[Liste[IdCH - 1].z, Liste[IdCH - 1].cluster] = IdCH
111 :          NodSendE(Nodes, clusterCH) ; ElectCH(Nodes, clusterCH)
112 : Fin

```

3.3 Conclusion :

L'énergie récoltée depuis l'environnement par les capteurs répond à des besoins de la longue durée et de hautes performances, et les systèmes de charge affectent les performances des nœuds et la topologie du réseau.

Pour cela nous avons adapté dans ce chapitre le protocole de routage LZHREP lequel nous avons amélioré pour former un nouveau protocole de routage dont le but de maximiser la durée de vie de réseau de capteur avec la prise en compte de l'énergie renouvelable.

Evaluation des performmence

4.1 Introduction

Nous avons abordé dans le troisième chapitre, la présentation de protocole EH-LCHREP (Energy Harvesting Layer and Cluster based Hierarchical Routing and Energy Optimization Protocol) qui est basé sur la récupération de l'énergie renouvelable, l'évaluation des performances de ce protocole sont proposés dans ce chapitre.

Dans ce chapitre, nous allons présenter les outils de simulations des réseaux de capteurs, ensuite nous allons évalué les performances de réseaux de capteurs sans fil (**EH-LCHREP**) que nous avons proposé.

4.2 Les simulateurs des réseaux de capteurs :

Pour simuler des réseaux de capteur sans fil nous avons besoins des outils de simulation peuvent être simples et d'autres sont relativement complexes, certains sont à accès libre dits "open source" comme ns2 et ns3 (network simulator), d'autres par contre sont commerciaux comme exemple OPNET (Optimum Network Performance). Parmi eux :

4.2.1 OPNET (Optimum Network Performance) :

C'est un outil très puissant et très complet, basé sur une interface graphique, son utilisation et sa manipulation sont relativement faciles. Il comporte trois niveaux hiérarchiques imbriqués [24] :

- Domaine réseau :

Utilisé pour définir la topologie du réseau en installant des routeurs, des hôtes et des périphériques (tels que des commutateurs) reliés entre eux via des commutateurs.

- Domaine de nœud :

Définit la composition des nœuds (routeurs, postes de travail, etc.). Définit le modèle à l'aide de blocs appelés modules. Certains modules ne sont pas programmables, ce sont des émetteurs et des récepteurs, d'autres sont entièrement programmables : ils sont processeur et queue. Le processeur est un module qui exécute des tâches spécifiques aux nœuds et peut communiquer entre eux via des flux de paquets de données.

- Domaine de processus :

Ce niveau définit le rôle de chaque module programmable.

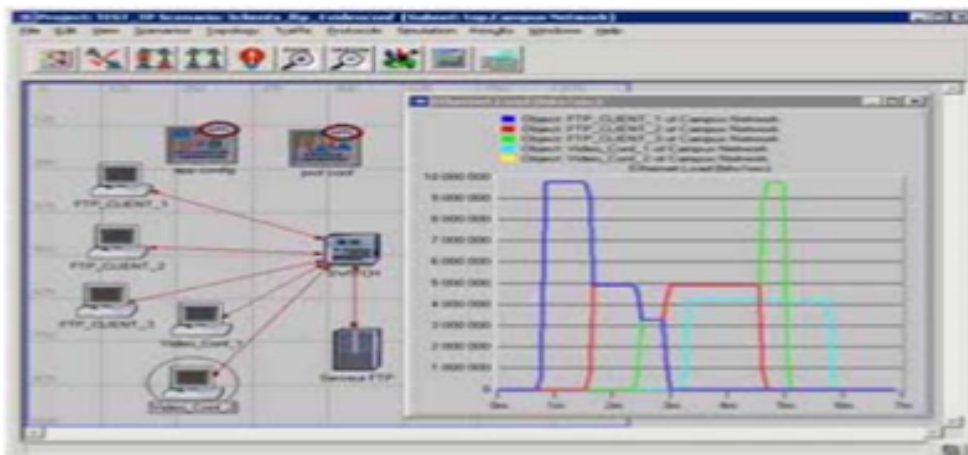


FIGURE 4.1 – Exemple d'interface d'OPNET.

4.2.2 L'OPNET présente trois fonctions principales :**-Modélisation :**

Il dispose d'un environnement graphique pour créer tout type de modèles de protocoles.

- Simulation :

Il utilise trois techniques de simulation avancées différentes.

- Analyse :

On peut utiliser des graphiques, des cartes, des données statistiques et d'autres moyens très simples pour analyser et afficher les données et les résultats de simulation [25].

4.2.3 Qualnet Simulator :

QualNet est un simulateur avancé pour les grands réseaux hétérogènes et les applications distribuées fonctionnant sur ces réseaux, utilisant C / C ++, QualNet est un programme GUI (interface utilisateur graphique) pour le développement de solutions réseau. Il peut être utilisé pour concevoir visuellement des schémas de réseau, puis simuler ces réseaux.

La fonction QualNet fournit une fonction unique qui peut simuler avec précision et efficacité des réseaux hétérogènes à grande échelle. QualNet est optimisé pour la vitesse et l'évolutivité sur un seul processeur. La vitesse d'exécution de la solution équivalente est de 5 à 10 fois supérieure à celle du QualNet commercial. Après avoir ajouté le processeur, la simulation s'exécute plus rapidement [26].

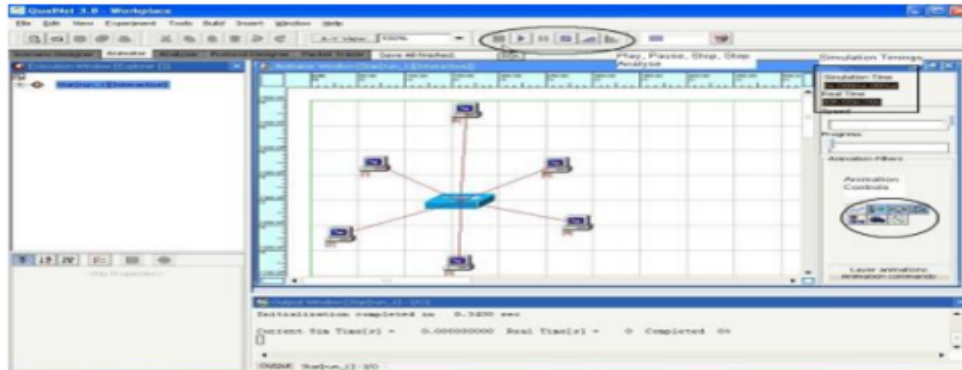


FIGURE 4.2 – Exemple d'interface de QualNet.

4.2.4 NS3 (Network Simulator version3) :

Le simulateur ns-3 est un simulateur de réseau d'événements discrets pour les systèmes Internet, principalement utilisé pour la recherche et l'éducation, c'est un projet open source pour le développement de ns-3. Ns-3 est un logiciel libre, utilisant C ++, Python, et il s'appuiera sur les contributions continues de la communauté pour développer de nouveaux modèles, déboguer ou maintenir les modèles existants et partager les résultats.

Les utilisateurs de NS-3 peuvent utiliser des générateurs de trafic, des protocoles tels que TCP / IP et des modèles d'appareils et de canaux tels que le WiFi pour créer des simulations de réseaux informatiques et analyser ou visualiser les résultats [27].

4.3 Environnement de simulation :

Après l'étude du simulateur Openet nous avons remarqué que son environnement est très compliqué et La consommation d'énergie des nœuds dans chaque transaction est élevée et au même temps il ne permet pas de modéliser le chargement des batteries des nœuds par le soleil. Pour ces raisons-là, nous avons choisi d'utiliser le logiciel de programmation "python" pour écrire notre propre simulateur adapté à nos besoins.

4.3.1 Python :



python¹ Est un langage utilisé la technique de programmation orientée objet interprété, a été conçu en 1991 par le Néerlandais Guido Van Rossum, et est langage interactive lui même destine a remplacer le langage basic.

L'avantage principale est leur mise on ouvre est très simple [29]

Python est conçu pour optimiser la productivité du développeur, la qualité du logiciel, la portabilité des programmes et intégration de composants. Les programmes python fonctionnant dans les plupart plate-forme couramment utilisé, des micro-ordinateurs aux super calculateur, sous les systèmes Linux, Windows, sous l'environnement Java et .Net ... [30].

4.3.2 Paramètres de simulation

Les nœuds capteurs sont fusionnés dans une zone carrée 500*500 m² composent 200 nœuds capteurs déployés aléatoirement ces capteurs possédons même énergie initiale 0.7 j, les mêmes capacités de calcul et mémoire, la même portée de transmission 100m, l'énergie de la station de base est illimitée et nombre paquet est 100, le tableau 4.1 sommé les paramètres utilisés :

Paramètres	valeurs
Nombre de stations de base	1
Nombre des nœuds	200
Surface de simulation	(500*500) m ²
Location de Station de Base	(250, 500)
Portée de transmission	100 m
Nombre de paquet	100

TABLE 4.1 – Paramètres de simulation.

4.3.3 Description du système :

Notre système traité sur une zone de surface 500*500 m², contient une station de base **SB** et 200 nœuds, les entités de ce système sont :

- La station de base **SB** : elle envoie des requêtes à chaque nœud et collecte les données du réseau.
- Les Cluster Heads **CHs** : sont des nœuds capteurs particuliers choisis selon des techniques spéciales, Ils reçoivent, agrègent les données qui transmettent par les nœuds et ils aident au routage des paquets dans le réseau.
- Les faux Clusters Heads **FCH** : ce sont des nœuds temporaires, responsables aux sélections de CHs dans chaque cluster, une fois le **CH** est sélectionné ils reviendront des **Sn**.
- Les nœuds capteurs **Sn** : Ils ont utilisé pour détecter les différents événements dans la zone et Afin de participer au routage des données vers la station de base.

4.3.4 Variables suggestives du système :

Le tableau suivant renomme les variables utilisées dans notre système :

Définition de la variable	Nom de la variable	Valeur
Energie initiale	initenergy	0.7 Joule
Energie consommée lors de l'émission d'un paquet	transenergy	$5 * 10^{-6}$ Joule
Énergie de nœuds restent après la transmission	ERest	Enode-Econs Joule
Énergie Consomme par les nœuds pour chaque transfère	Econsomme	transenergy * d *.SizeMsgSend
Energie consommée lors de l'agrégation des données	aggreenergy	$5 * 10^{-7}$ Joule
Energie consommée lors de la réception d'un paquet	recenergy	$5 * 10^{-6}$ Joule
Distance entre deux capteurs	DisAvecAutre	$\sqrt{(x_0 - x_1)^2 + (y_0 - y_1)^2}$
Position d'un capteur	(x,y)	(x,y) en mètre
La source énergie	P(S)	$ 0.9 * R(t) * \cos(\frac{t}{7*\pi}) * \cos(\frac{t}{10*\pi}) $
les valeurs aléatoires	R(t)	[0.1]

TABLE 4.2 – Variables suggestives du système

4.4 Évaluation des performances :

Nous découvrirons et analyserons les résultats de la simulation et les discussions. Le résultat de chaque scénario est de réaliser une simulation indépendante.

4.4.1 Distribution des nœuds :

Le réseau de captage est divisé en couches (des niveaux de L1 à L5) et en clusters (voir figure 4.3). Et le déploiement aléatoire des nœuds de capteurs, des **FCH** et des **CH** sélectionnés avec notre protocole sont montre dans la figure 4.4, le demi-cercle orange représente la station de base **SB**.

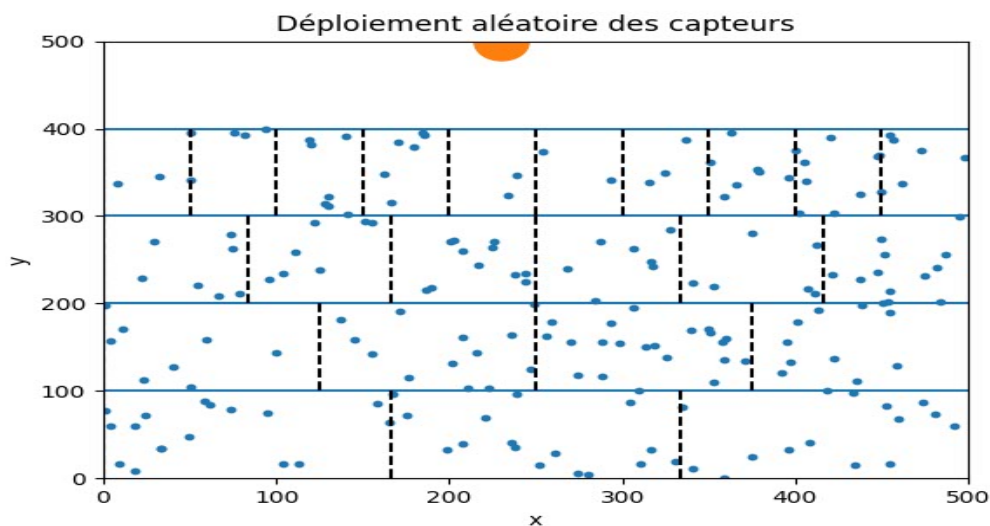


FIGURE 4.3 – Présentation de la topologie du réseau.

4.4.2 Formation cluster :

La figure suivante montre la zone de captage après la formation de chaque niveau et chaque de cluster, et chaque couleur représente le nœud du même cluster.

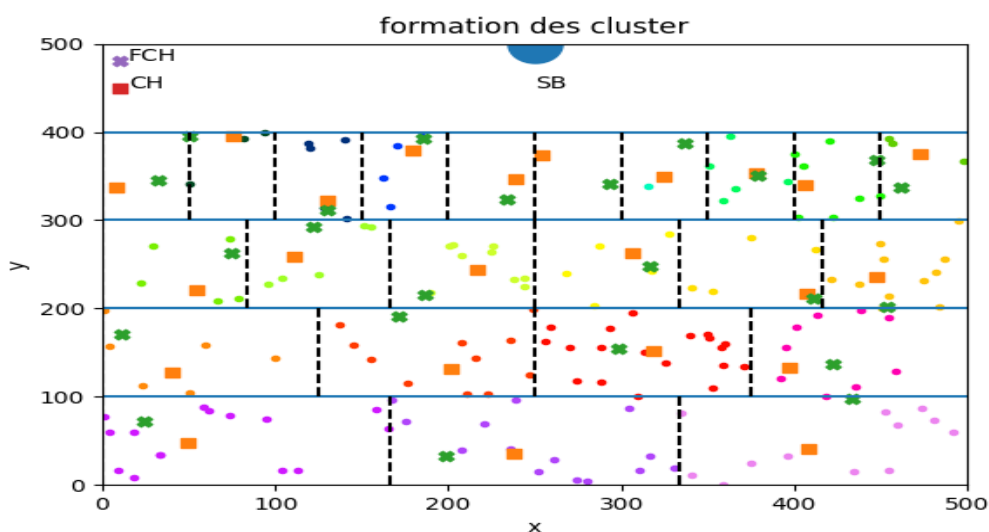


FIGURE 4.4 – l'architecture du réseau après la formation des clusters.

4.4.3 La sélection des Clusters Head et les Faux Clusters Head :

Après avoir sélectionné Les **FCHs** et les **CHs** pour chaque cluster, la zone de captage sera représentée sur la figure 4.5, de sorte que **CH** puisse être représenté par un petit carré et **FCH** peut être représenté par un signe plus.

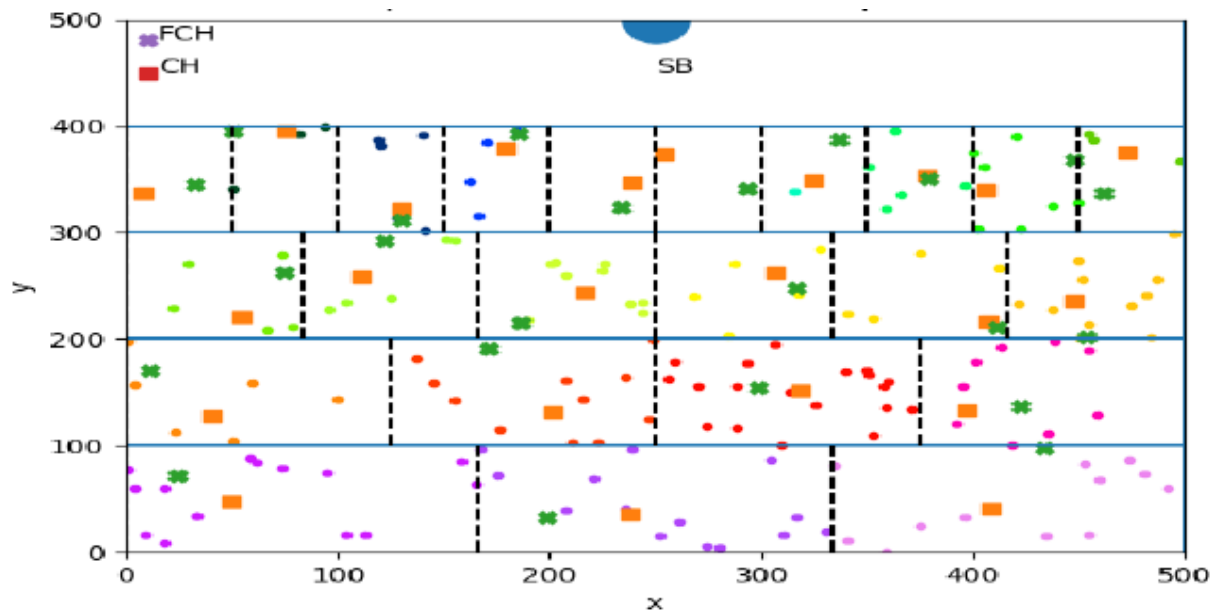


FIGURE 4.5 – Visualisation du réseau après la sélection des **FCHs** et **CHs**.

4.4.4 Énergie restante après la sélection des Clusters Head :

La figure 4. 6 suivante montre l'énergie restante de chaque capteur après avoir sélectionné **CH** par notre protocole EH-LCHREP. De cette manière, l'énergie restante de la plupart des nœuds est d'environ 0,7 J et l'énergie restante des autres nœuds est réduite à 0,65J. Les nœuds fournissant de l'énergie sont proches de 0,7 j, et ces nœuds sont sélectionnés comme **FCH**. Ils consomment de l'énergie pour sélectionner CH en calculant le seuil d'élection pour chaque capteur et en envoyant et recevant des messages de tous les nœuds de leur cluster. D'autre part, uniquement lors de l'envoi et de la réception de paquets de données depuis **FCH**, les autres nœuds consomment un minimum d'énergie.

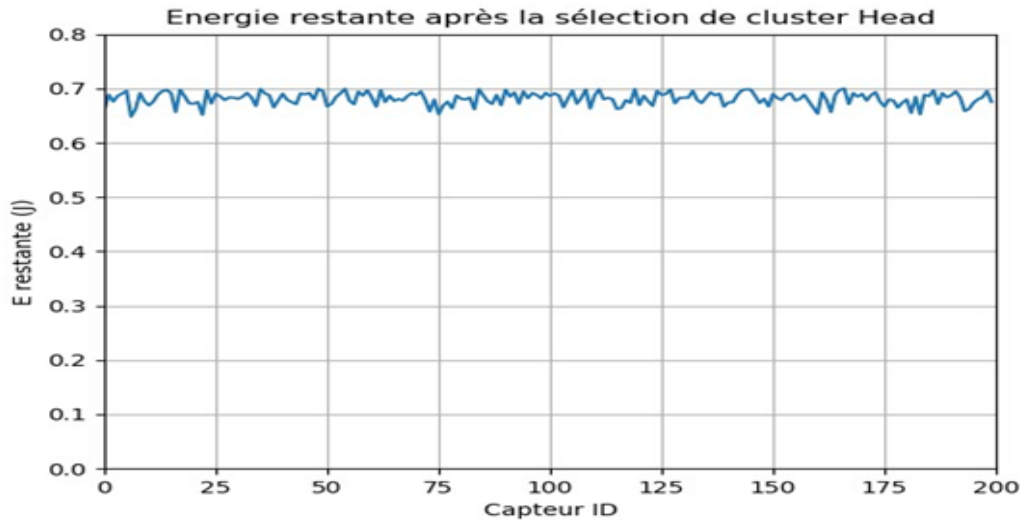


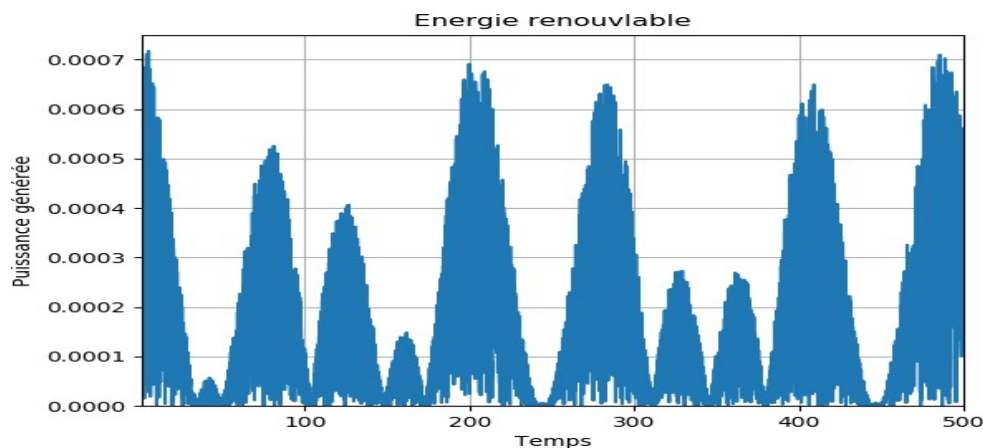
FIGURE 4.6 – Énergie restante après la sélection des CHs.

4.4.5 Modèle de source énergie :

Nous supposons que l'énergie environnementale (comme l'énergie solaire) est collectée et convertie en énergie électrique pour compléter les batteries du système du véhicule. Afin de simuler le comportement de la source solaire, nous utilisons le modèle suivant [22] :

$$P(s) = |0.9 * R(t) * \cos(\frac{t}{7 * \pi}) * \cos(\frac{t}{10 * \pi})| \quad (4.1)$$

Où $R(t)$ représente une variable aléatoire uniformément répartie entre 0 Et 1. Coupez la valeur P_s à la valeur P_s , la valeur maximale = 0,9. Comme le montre la figure, la courbe de puissance obtenue $P_s(t)$ simulant le jour et la nuit est similaire à celle des cellules solaires L'environnement extérieur.

FIGURE 4.7 – La courbe de la puissance $P_s(t)$.

4.4.6 Energie consommée après l'envoi de 100 paquets :

La figure 4.8 suivante montre la puissance consommée par l'envoi de paquets de données par notre protocole. On peut voir qu'à mesure que la transmission de paquets augmente, la consommation d'énergie du protocole augmente proportionnellement. En fait, chaque nœud qui dissipe de l'énergie à travers un nœud de capteur ou un nœud **CH** la recharge grâce à l'énergie renouvelable.

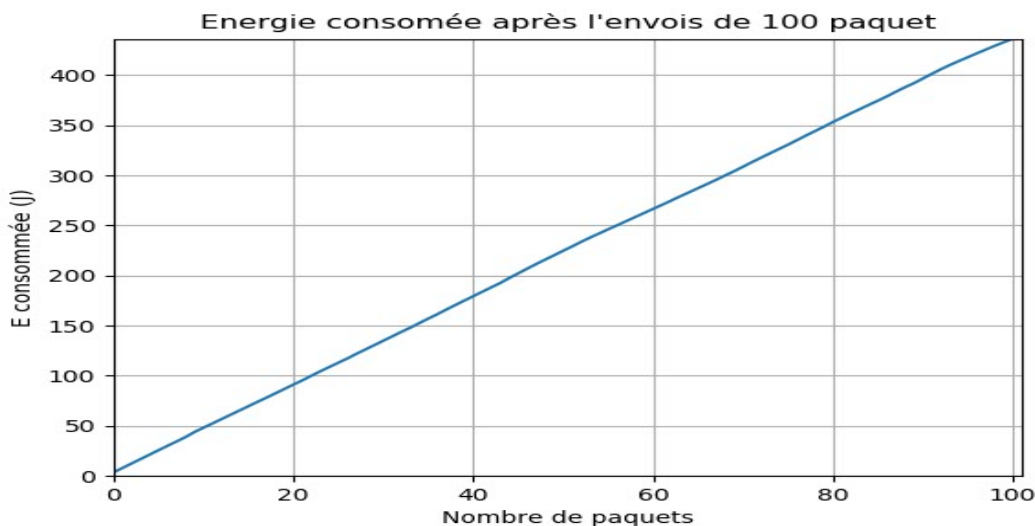


FIGURE 4.8 – Energie consommée après l'envoi de 100 paquet (EH-LCHREP).

Nous comparons notre résultat avec le protocole EEPSC sans énergie renouvelable, nous remarquons que le nombre de paquets que notre protocole peut envoyer est supérieur au nombre de paquets que le protocole EEPSC peut envoyer. Où ce dernier n'envoie que

60 paquets, puis ses nœuds échouent car son énergie s'épuise.

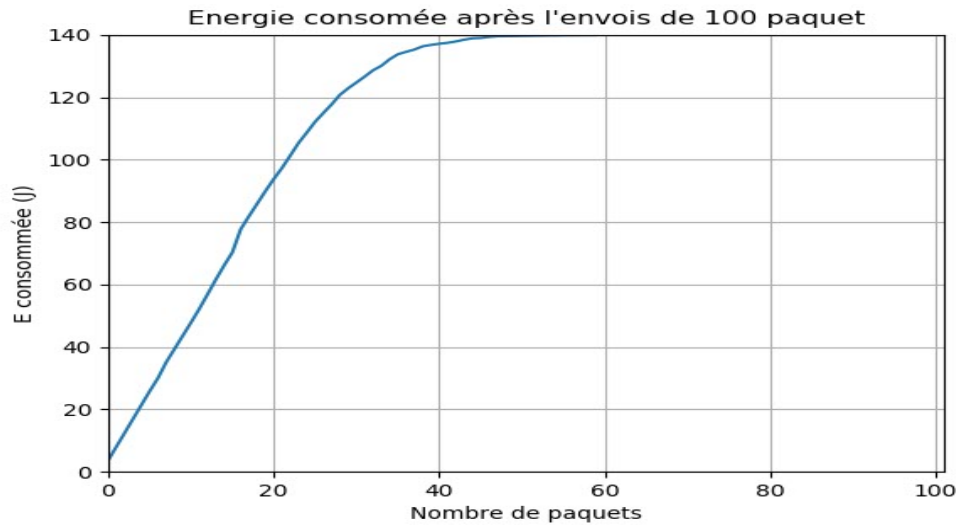


FIGURE 4.9 – Energie consommée après l'envoi de 100 paquets (EEPSC).

4.4.7 Nombre de nœuds mort en termes de paquets envoyés :

Lors de l'étape d'élections de Cluster Head de notre protocole EH-LCHREP, nous avons adopté deux scénarios suivant la formule d'élection donnée dans le chapitre 3 (eq 3.1) :

1er scénario : Considéré l'énergie résiduel et la distance :

Dans ce cas un nœud est élu CH si son facteur d'élection $Electe_{node}$ (donné ci-dessous) est le maximum parmi ceux des autres nœuds du même cluster

$$Electe_{node} = \frac{E_{node_i}}{DisAvecAutr_i} \quad (4.2)$$

Les résultats obtenus, de point de vue du nombre de nœuds mort dans ce cas sont montré dans la figure 4.10 ci-dessous. Nous remarquons que le nombre de nœud mort augmente au fure à mesure d'envois de paquets, où les 100 paquets sont envoyés avec la morte (défaillance) de 50 nœuds.

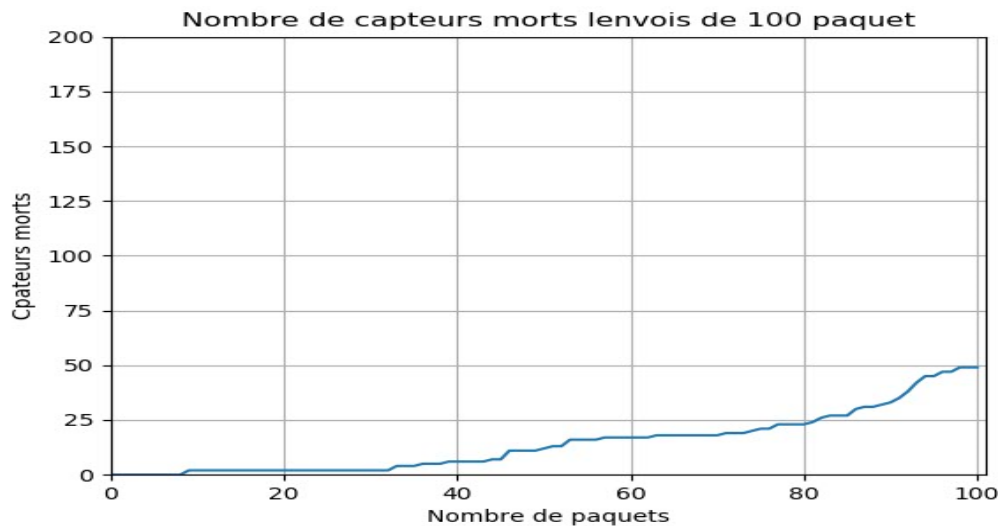


FIGURE 4.10 – Nombre de nœuds mort en termes de paquets envoyés (avec distance).

2ème scénario : Considéré uniquement l'énergie résiduel :

Dans ce cas un nœud est élu CH si son énergie résiduel (cas du protocole EEPSC) est maximal parmi ceux des autres nœuds du même cluster

$$Electe_{node} = E_{node_i} \quad (4.3)$$

La figure 4.11 ci-dessous montre l'augmentation du nombre de nœuds mort (75 nœuds mort) par rapport au cas d'utilisation de la distance entre les nœud expliqué dans le 1er scénario.

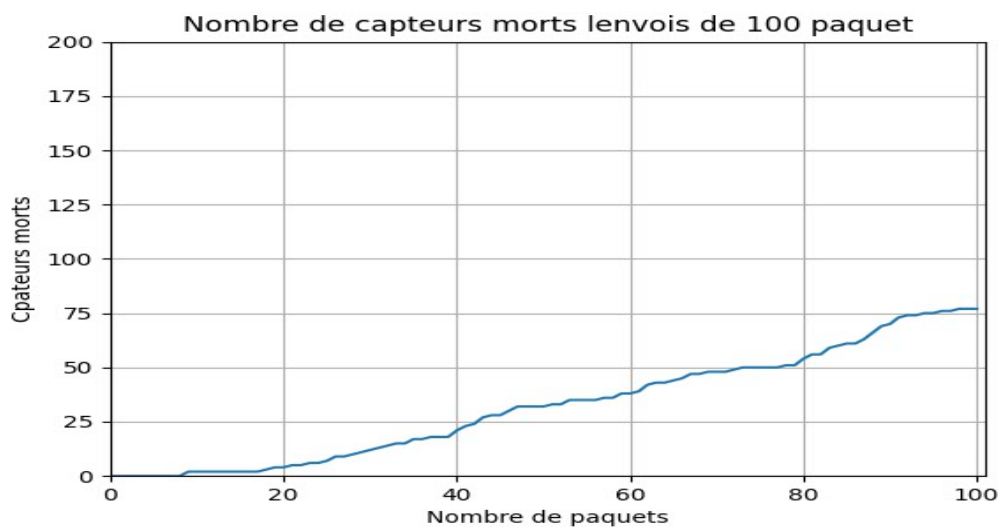


FIGURE 4.11 – Nombre de nœuds mort en termes de paquets envoyés (sans distance).

La comparaison de notre protocole EH-LCHREP avec le protocole EEPSC (voir figure 4.11), montre que le nombre de nœuds mort en fonction d'envoi de paquets avec EEPSC est supérieur celui induit par notre protocole. Ce qui l'utilité d'utiliser la distance entre nœuds lors de l'élection des CH et aussi l'apport de l'énergie renouvelable

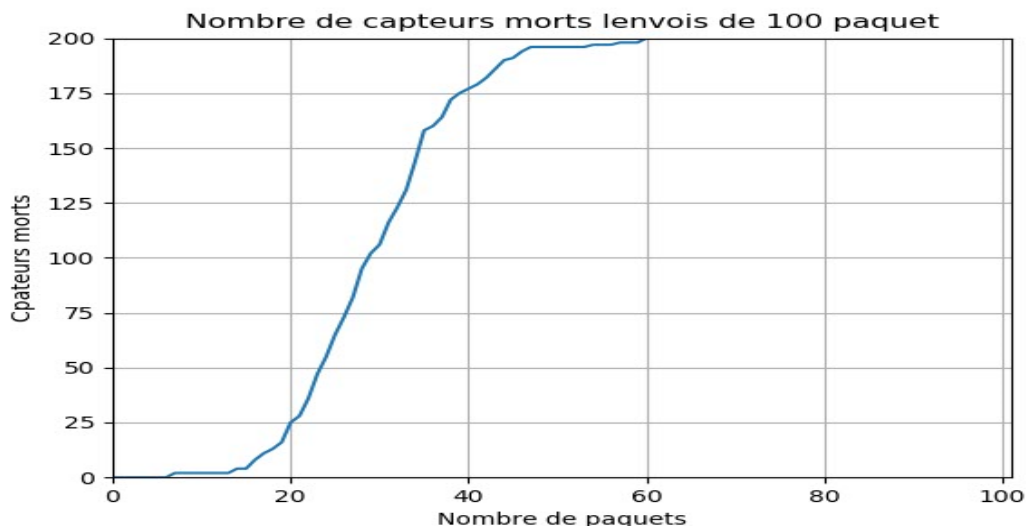


FIGURE 4.12 – Nombre de nœuds mort en termes de paquets envoyés par protocole EEPSC.

Ainsi nous pouvons dire que notre protocole est performant et efficace comparative-ment au protocole EEPSC de point de vue durée de vie et la consommation d'énergie.

4.5 Conclusion

Dans ce chapitre, nous avons présenté les simulateurs des réseaux de capteurs. En suite, nous avons présenté l'environnement de simulation de notre protocole EH-LCHREP. Par la suite, nous avons évalué les performances du protocole EH-LCHREP. Les résultats de simulation obtenus montrent que notre protocole permet d'étendre la durée de vie du réseau comparativement au protocole EEPSC.

Conclusion générale et perspective

Les réseaux de capteurs sans fil constituent un champ de recherche très riche, il peut être appliqué dans plusieurs domaines différent. Mais l'usage des batteries par des capteurs a été une contrainte critique pour les réseaux de capteurs sans fil, pour cela plusieurs techniques et protocoles de routage efficaces en énergie ont été développés afin de maximiser la durée de vie de réseau.

Mais pour garantir la longévité du réseau, l'architecture logicielle est insuffisante car les nœuds capteurs sont déployés dans des environnements inaccessibles ou le changement des batteries est impossible, il faut aussi améliorer l'architecture matérielle et la rendre compatible avec les changements environnementaux, par l'ajout des techniques avancées qui sont capables de récupérer l'énergie ambiante qui existe dans notre environnement. Ce travail se focalise donc sur l'adaptation de certains des meilleurs protocoles de routage hiérarchiques efficaces en énergie pour les réseaux de capteurs sans fil (EE-WSN), avec la technologie de récupération d'énergie pour les RCSF (EH-WSN). Afin d'éliminer le problème d'épuisement des batteries des capteurs.

Notre principal objectif dans ce mémoire est de proposer un nouveau protocole de routage hiérarchique EH-LCHREP (Energy Harvesting Layer and Cluster based Hierarchical Routing and Energy Optimization Protocol), qui prend en considération l'énergie du réseau pour augmenter sa durée de vie en réduisant la perte d'énergie par l'usage de routage multi-saut dans la communication inter-cluster, et aussi utilise des clusters formés sur un réseau structuré en couches. Et par une élection de Cluster Head, qui utilise la distance moyenne ce qui permet de minimiser la perte d'énergie dans la communication intra-cluster. Notons que ce protocole doit être conscient des changements physiques de l'environnement, par exemple la lumière du soleil change toujours régulièrement pendant la journée avec le

nuage qui minimise la luminosité du soleil donc la récupération d'énergie sera faible. L'évaluation des performances de notre protocole EH-LCHREP est simulée dans Python, et nous avons trouvé que notre proposition est meilleure que le protocole EEPSC. A la fin, comme une perspective, nous prévoyons d'améliorer notre protocole et le comparer avec d'autres protocoles qui prenant compte les contraintes d'énergie renouvelable. Il sera préférable aussi d'implémenter notre solution dans des simulateurs dédiés aux réseaux de capteurs sans fil (comme OPNET, NS2, MATLAB...).

Bibliographie

[1] HAFIR_Latifa , SLIMANI_Radhia . "Etude et évaluation des performances des protocoles de routage pour les réseaux de capteurs sans fil ",Master en Informatique, Administration et Sécurité des Réseaux, Université A/Mira de Béjaia ,2016 .

[2] AMOKRANE Fatma,"Routage avec optimisation de la consommation d'énergie dans les réseaux de capteurs sans fil",Mémoire Master, Université A/Mira de Béjaia,2016.

[3] BANANA, Abdallah, and Mohammed Kaddi. Vers une plate-forme efficace en énergie pour les réseaux de capteurs sans fil. Diss. Université Ahmed Draia-Adrar, 2018.

[4] Kifouche, Abdenour. Ingénierie dirigée par les modèles pour la conception et la mise en œuvre des réseaux de capteurs. Diss. Université Paris-Est, 2019.

[5] Anisi, Mohammad Hossein, et al. "Energy harvesting and battery power based routing in wireless sensor networks." *Wireless Networks* 23.1 (2017) : 249-266.

[6] Stefano Basagni, M.Yousof Naderi, Chiara Petrioli,and DoraSpenza."WIRELESS SENSOR NETWORKS WITHENERGY HARVESTING",Electrical and Computer Engineering Department, Northeastern University, Boston,MA, U.S.A,Dipartimento di Informatica, Universit'a di Roma "La Sapienza" ,Roma, Italy.

[7] Li, Yin, and Ronghua Shi. "An intelligent solar energy-harvesting system for wireless sensor networks." *EURASIP Journal on Wireless Communications and Networking* 2015.1 (2015) : 1-12..

[8] Bourai, Amar, and Abdelhamid Bentabet. la localisation dans les réseaux de capteurs Cas étudié : feu de foret. Diss.

[9] HAFIR Latifa et SLIMANI Radhia , Etude et évaluation des performances des protocoles de routage pour les réseaux de capteurs sans fil, Master en Informatique, Administration et Sécurité des Réseaux , université de bijaia.

- [10] Mekki nabil, Mohammedi kada, "Techniques de conservation d'énergie pour les réseaux de capteur sans fil", MEMOIRE de Master, univ-saida, 2018.
- [11] Prasanna, Srinivasa, and Srinivasa Rao. "An overview of wireless sensor networks applications and security." *International Journal of Soft Computing and Engineering (IJSCE)*, ISSN 2231 (2012) : 2307.
- [12] HELAILI Nabila et MEKHNACHE Salima , Simulation du Routage dans les Réseaux de Capteurs Sans Fils , Master en Informatique, Administration et Sécurité des Réseaux, université de bijaia.
- [13] Sofiane MOAD, La consommation d'énergie dans les réseaux de capteurs sans fil ,Etude bibliographique, Master Recherche 2 en Informatique ,IFSIC-Rennes1.
- [14] Ziani, Youcef. Étude comparative de méthodes de routage dans les réseaux de capteurs sans fil pour le domaine résidentiel. Diss. Université du Québec à Trois-Rivières, 2013.
- [15] Farahani, Gholamreza. "Energy Consumption Reduction in Wireless Sensor Network Based on Clustering." *International Journal of Computer Networks and Communications (IJCNC)* Vol 11 (2019).
- [16] ABBAS, A., AMAD, M., AMOKRANE, F. (2020). LCHREP : Layer and Cluster based Hierarchical Routing and Energy Optimization Protocol in Wireless Sensor Networks. *Journal of Digital Information Management*, 18(4), 139.
- [17] Krogh, Einar. "An Introduction to the Internet of Things." (2020).
- [18] Ostadabbasi, Negin. Analysis of routing algorithms for Energy harvesting wireless sensor network. Diss. MS thesis, DTU Comput. Sci. Eng., Tech. Univ. Denmark, Lyngby, Denmark, 2013.
- [19] http://www4.utc.fr/sit60/co/Module_RCSF_11.html
- [20] Bousnindja, Hesna, and Said Encadreur Rouibah. Les protocoles de routage hiérarchiques dans les réseaux de capteurs sans fil étude de cas. Diss. Université de Jijel, 2019.
- [21] SL, Libelium Comunicaciones Distribudas. "Wasp mote datasheet." (2010)
- [22] Abbas, Akli, et al. "A real-time feedback scheduler for environmental energy with discrete voltage/frequency modes." *Computer Standards & Interfaces* 44 (2016) : 264-273.
- [23] Bousnindja, Hesna, and Said Encadreur Rouibah. Les protocoles de routage hiérarchiques dans les réseaux de capteurs sans fil étude de cas. Diss. Université de Jijel, 2019.
- [24] http://malm.tuxfamily.org/doc/qr_chap4_opnet.htm.

[25] SAADANE HOUDA, La qualité de service d'un streaming vidéo Dans un réseau ad hoc (égale à égal), MAGISTER ,2012.

[26] Dinesh, Sorout, and Goyal Sonal. "Qualnet simulator." International Journal of Information & Computation Technology, ISSN (2014) : 0974-2239.

[27] Siraj, Saba, A. Gupta, and RinkuBadgular. "Network simulation toolssurvey." International Journal of Advanced Research in Computer and Communication Engineering 1.4 (2012) : 199-206.

[28] Yves Bailly, Initiation à la programmation avec Python et C++, 2009.

[29] Mark Lutz, Python : précis & [et] concis : [memento des programmeurs ,page1 ,2005.

[30] Akli ABBAS, Mourad AMAD, Juba BELLAL, Adel MOUSSAOUI, "EEPSCZ : An Energy-Efficient Protocol with Static Clustering Zones", 2020

Résumé

Les réseaux de capteurs composent d'un ensemble des nœuds capteurs développées spécialement pour suivi l'environnement, ces nœuds sont coopérés entre eux et communiquer par des liens sans fil. La source d'énergie pour alimenter les nœuds capteurs est fournie par des batteries, ceux-ci peuvent être rechargeable ou non rechargeable. De nos jours, l'énergie renouvelable offre des avantages pour les **RCSF**, qui sont capables de prolongées la durée de vie, mais ce n'est pas suffisant car les conditions environnementales sont changeables, pour cela il nécessite une architecture efficace et des routes qui permet d'économie d'énergie. Dans ce travail, nous avons proposé un protocole de **rou-tage hiérarchique** avec la capacité de récupération d'énergie, nous avons conduit une simulation à l'aide de Python, dont le but de confirmer les améliorations apportées par notre proposition.

Mots clés : **RCSF**, **rou-tage hiérarchique**.

Abstract

Sensor networks are made up of a set of sensor nodes developed specifically for monitoring the environment, these nodes are cooperating with each other and communicate via wireless links. The energy source to power the sensor nodes is provided by batteries, these can be rechargeable or non-rechargeable. Nowadays, renewable energy offers advantages for **RCSFs**, which are able to extend the life span, but this is not enough because the environmental conditions are changeable, for this it requires an efficient architecture and energy-saving roads. In this work, we proposed a **hierarchical routing** protocol with energy recovery capacity, we conducted a simulation using Python, with the aim of confirming the improvements made by our proposal.

Key words : **RCSF**, **hierarchical routing**.