



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 : GSI

Thème

Routage dans les réseaux de capteurs sans fil sous
contrainte de mobilité et l'énergie renouvelable

Encadré par

— DR. ABBAS Akli

Réalisé par

— AKLI Meziane

2020/2021

Remerciements

Je souhaite avant tout remercier **ALLAH** le tout puissant de m'avoir donné le courage, la patience et la volonté pour la réalisation de ce travail.

Au terme de la rédaction de ce mémoire, je tiens à exprimer ma profonde gratitude et mes sincères remerciements à mon encadreur Monsieur **Dr. Abbas Akli** pour ses orientations, ses encouragements, sa disponibilité et ses précieux conseils qui m'ont permis de mener à bien ce travail.

Mes vifs remerciements sont adressés aux membres du jury pour l'intérêt qu'ils ont porté à ma recherche en acceptant d'examiner mon travail et de l'enrichir par leurs propositions.

Je désire aussi remercier les professeurs de l'université de UAMO-Bouira département informatique, qui m'ont fourni les outils nécessaires à la réussite de mes études universitaires.

Je remercie également toute l'équipe pédagogique de l'université de UAMO-Bouira et les intervenants professionnels responsables de ma formation, pour avoir assuré la partie théorique de celle-ci.

Je tiens à exprimer ma gratitude à toute ma famille et mes amis qui m'ont apporté leur soutien moral et intellectuel tout au long de mon travail.

Dédicaces

Je dédie du profond de mon cœur ce mémoire :

*À ma chère mère **Saadia** pour ton amour pour moi, pour les sacrifices que tu consens pour rendre tes enfants heureux. tu as enduré beaucoup de peine pour mon bien-être et à ma réussite. Reçois ceci en guise de ma reconnaissance et que Dieu te garde longtemps afin que tu puisses goûter aux arbres que tu as plantés.*

*À mon cher frère aîné **Malik** pour m'avoir enseigné le chemin du labeur et de la vérité. Que Dieu t'accorde une longue vie et une santé de fer avant que tu sois récompensé pour tous les sacrifices que tu as toujours consentis pour moi.*

*À ma chère sœur **Hakima** pour ton affection et ton soutien.*

À toute ma famille mes chères frères et sœurs qui ont partagé avec moi tous les moments d'émotion lors de la réalisation de ce travail. Ils m'ont chaleureusement supporté et encouragé tout au long de mon parcours.

À tous mes amis et amies qui m'ont soutenu d'une manière ou d'une autre

Meziane.

Table des matières

Table des matières	i
Table des figures	iv
Liste des tableaux	vi
Liste des abréviations	vii
Introduction générale	1
1 Généralités sur les RCSF	3
1.1 Introduction	3
1.2 Capteurs	3
1.2.1 Définition	3
1.2.2 Architecture matérielle	4
1.2.3 Architecture logicielle	5
1.2.4 Capteur mobile avec l'énergie renouvelable	7
1.3 Réseau de capteurs sans fil (RCSF)	8
1.3.1 Définition	8
1.3.2 Architecture de réseaux de capteurs sans fil	9
1.3.3 Caractéristique de RCSF	10
1.3.4 Types de communication dans RCSF	10
1.4 Réseau de capteurs sans fil mobile	11
1.4.1 Caractéristiques générales de la mobilité dans les RCSF	11
1.4.2 Type d'élément mobile	12

1.4.3	Types de mouvements	13
1.5	Domaines d'application des RCSF	13
1.6	L'énergie solaire	15
1.6.1	Définition	15
1.6.2	ISEH system (Intelligent Solar Energy Harvesting)	15
1.6.3	Architecture d'un ISEH system	16
1.7	Consommation d'énergie	17
1.7.1	L'énergie ambiante	17
1.7.2	Les principales opérations dues à la consommation d'énergie	17
1.7.3	Consommation d'énergie dans les RCSF avec et sans énergie renouvelable	18
1.7.4	Modèle de consommation d'énergie	18
1.8	Conclusion	20
2	Routage dans RCSF	21
2.1	Introduction	21
2.2	Routage dans RCSF	21
2.2.1	Définition de routage	22
2.2.2	Classification des protocoles de routage dans les RCSF	22
2.2.3	Facteurs de conception de protocole de routage	25
2.3	Travaux connexes	26
2.3.1	Protocoles hiérarchique basé sur le clustring	27
2.3.2	Protocoles hiérarchique basé sur la récupération d'énergie	28
2.3.3	Protocoles hiérarchique basé sur la mobilité	30
2.4	Conclusion	33
3	Le Protocole EHMS-EEMCS (Energy Harvesting Mobile Sink-An Energy-Efficient Mobility-Based Cluster Head Selection for Lifetime Enhancement of Wireless Sensor Networks)	34
3.1	Introduction	34
3.2	Description de protocole	34
3.2.1	Motivation	34
3.2.2	Hypothèses	35

3.2.3	Principe de fonctionnement	36
3.2.4	Conclusion	49
4	Évaluation de performmence	50
4.1	Introduction	50
4.2	Les simulateurs des réseaux de capteurs	50
4.2.1	Matlab :	50
4.2.2	NS3 (Network Simulator version3) :	51
4.2.3	OMNeT++ :	51
4.3	Environnement de simulation	51
4.3.1	Python :	51
4.3.2	Paramétrés de simulation	52
4.3.3	Description du système	52
4.3.4	Variables suggestives du système	53
4.4	Évaluation des performances	53
4.4.1	Distribution des nœuds	54
4.4.2	Formation des clusters :	54
4.4.3	Modèle de source d'énergie	55
4.4.4	Première amélioration :station de base mobile	56
4.4.5	Deuxième amélioration :l'énergie renouvelable et station de base mobile	57
4.5	Conclusion	59
	Conclusion générale	60
	Bibliographie	62

Table des figures

1.1	Exemple d'un capteur	4
1.2	Architecture matérielle d'un capteur sans fil	4
1.3	Système d'exploitation conçu pour les capteurs	6
1.4	Capteur sans fil mobile basée sur la récupération d'énergie.	7
1.5	Architecture matérielle d'un capteur sans fil mobile basé sur la récupération d'énergie	7
1.6	Un réseau de capteur sans fil.	9
1.7	Exemple de changement de la topologie causé par la mobilité	11
1.8	Classification de la mobilité dans les RCSF	12
1.9	Architecture d'un RCSF avec station de base mobile	12
1.10	Architecture d'un RCSF avec des capteurs mobile	13
1.11	Domaine d'applications des RCSF	14
1.12	Architecture de ISEH système	16
1.13	Niveaux de consommation d'énergie au sein d'un nœud capteur	17
1.14	Consommation d'énergie dans les RCSF avec et sans énergie renouvelable .	19
1.15	Modèle de consommation d'énergie.	20
2.1	Classification des protocoles de routage.	22
2.2	Architecture à plate	23
2.3	Architecture hiérarchique.	24
2.4	Architecture basé sur la localisation	24
3.1	Organigramme de protocole EHMS-EEMCS	37

4.1	Déploiement des nœuds capteurs	54
4.2	Formation des clusters	55
4.3	La courbe de la puissance $P_s(t)$	55
4.4	Énergie total consommée par tour	56
4.5	Nombre de nœuds capteur survivant par tour	57
4.6	Énergie total consommée par tour	57
4.7	Énergie total consommée par tour	58
4.8	Résultats d'amélioration	59

Liste des tableaux

1.1	Densité de puissance pour certaines sources d'énergie ambiante	17
4.1	Paramétrés de simulation.	52
4.2	Variables suggestives du système	53

Liste des abréviations

CRPD	a novel clustering routing protocol for dynamic wireless sensor networks
CDMA	Code-Division Multip Accès
EEMCS	An Energy-Efficient Mobility-Based Cluster Head Selection for Lifetime
EHMS-EEMCS	Energy Harvesting Mobile Sink-An Energy Efficient Mobility Based Cluster Head Selection for Lifetime
GPS	Global Positioning System
HASS	Harvesting Aware Speed Selection
ISEH	Intelligent Solar Energy Harvesting
IMCH	I am Cluster Head
IoT	Internet of things
LEACH	Low Energy Adaptive Clustering Hierarchy
LEACH-M	Low Energy Adaptive Clustering Hierarchy Mobile
MPPT	Maximum Power Point Tracking
NS3	Network Simulateur version3
OS	Operating System
OPNET	Optimum NETwork Performance.
QoS	Quality of Servic
RCSF	Réseaux de capteur sans fil
RSSI	Received Signal Strength Indication
SB	Station de Base
TinyOS	Tiny Operating System
TDMA	Time Division Multiple Access.
WSN	Wireless Sensor Network

Introduction générale

Le monde d'aujourd'hui a besoin de recueillir des informations à partir d'environnements hautement sensibles et inaccessibles. À cette fin, la technologie de communication sans fil, a développé des capteurs avancés qui peuvent se déplacer et tourner autour de cette zone pour collecter des informations. Ces environnements nécessitent des réseaux de capteurs sans fil, qui se composent d'un grand nombre de petits capteurs, ces derniers ayant la capacité de collecter des grandeurs physiques telles que la température et la pression. . . et les distribuer sur son propre réseau. Actuellement, la majorité des recherches sur le domaine des réseaux de capteurs sans fil se concentrent sur le problème de la conservation de l'énergie car les capteurs sont de petite taille avec des ressources faibles et limitées et fonctionnent avec des batteries de faible puissance. Donc l'objectif primordial est le développement des techniques efficaces qui permettent l'économie d'énergie pour le réseau de capteurs sans fil. Les chercheurs ont étudié diverses méthodes de récupération d'énergie, ils ont développé des capteurs qui sont capables de récupérer l'énergie ambiante depuis l'environnement tel que l'énergie solaire.

L'objectif principal de ce mémoire est d'améliorer un protocole de routage basé sur la mobilité des capteurs et de prendre en compte l'énergie renouvelable avec une station de base mobile, le protocole tentera de prolonger la durée de vie d'un réseau cette démarche se concentre sur l'étude des protocoles de routage existant.

Dans ce mémoire nous avons proposé un protocole de routage hiérarchique dans le cas des réseaux de capteurs sans fil dynamiques et prenant en compte l'énergie renouvelable avec une station de base mobile. Ce protocole est appelé EHMS-EEMCS (Energy Harvesting Mobile Sink-An Energy-Efficient Mobility-Based Cluster Head Selection for Lifetime

Enhancement of Wireless Sensor Networks) qui est l'amélioration de protocole de routage EEMCS[29]. Les résultats obtenus montrent que EHMS-EEMCS permet de prolonger la durée de vie du réseau.

Le mémoire est structuré en quatre chapitres :

Le premier chapitre de notre travail est consacré à l'étude des généralités sur les réseaux de capteurs sans fil ainsi que les informations préliminaires dans cette technologie. D'abord nous avons expliqué la signification du réseau de capteurs sans fil (les différentes définitions, architecture et les caractéristiques). Ensuite nous avons expliqué les réseaux de capteurs sans fil et les domaines d'application de cette technologie, finirons sur les causes de la consommation énergétique.

Le deuxième chapitre nous étudions la technique de routage, débutons par la définition, la classification et les facteurs de conception de protocole de routage ensuite nous étudions quelques protocoles de routage sans fil qui sont connexes avec notre travail.

Le troisième chapitre nous proposons un protocole de routage hiérarchique que nous avons dénommé **EHMS-EEMCS (Energy Harvesting Mobile Sink-An Energy-Efficient Mobility-Based Cluster Head Selection for Lifetime Enhancement of Wireless Sensor Networks)** qui est l'amélioration de protocole (EEMCS), c'est un protocole de routage hiérarchique basé sur la mobilité.

Le dernier chapitre nous ferons l'évaluation des performances de notre protocole avec l'environnement de simulation python. Nous présenterons les différents paramètres utilisés dans la simulation, nous terminons par la démonstration des résultats obtenus.

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 offrent une solution économiquement intéressante pour la surveillance à distance et le traitement des données dans les environnements complexes et hostiles. Ils sont utilisés dans devers secteurs comprenant une structure technologique tell que l'agriculture, la santé, l'industrie, applications militaire et encore d'autres. Malgré ce riche potentiel d'application, les **RCSFs** sont soumis à plusieurs contraintes telles que la contrainte énergétique et la contrainte de mobilité des capteurs, qui rendent la durée de vie du réseau une notion critique. Dans ce chapitre introduit des connaissances de base nécessaires à la bonne compréhension de ce mémoire. Nous exposons une étude détaillée sur les **RCSFs** et leurs architectures, classifications, caractéristiques, domaines d'application, enfin nous introduisons le concept de consommation d'énergie.

1.2 Capteurs

1.2.1 Définition

Un capteur est un dispositif électronique informatique qui collecte et mesure une grandeur physique dans son environnement tel que la température, l'humidité, les vibrations et la pression et la convertit en une grandeur numérique capable d'être traitée informati-

quement(principalement un signal électrique).[1]

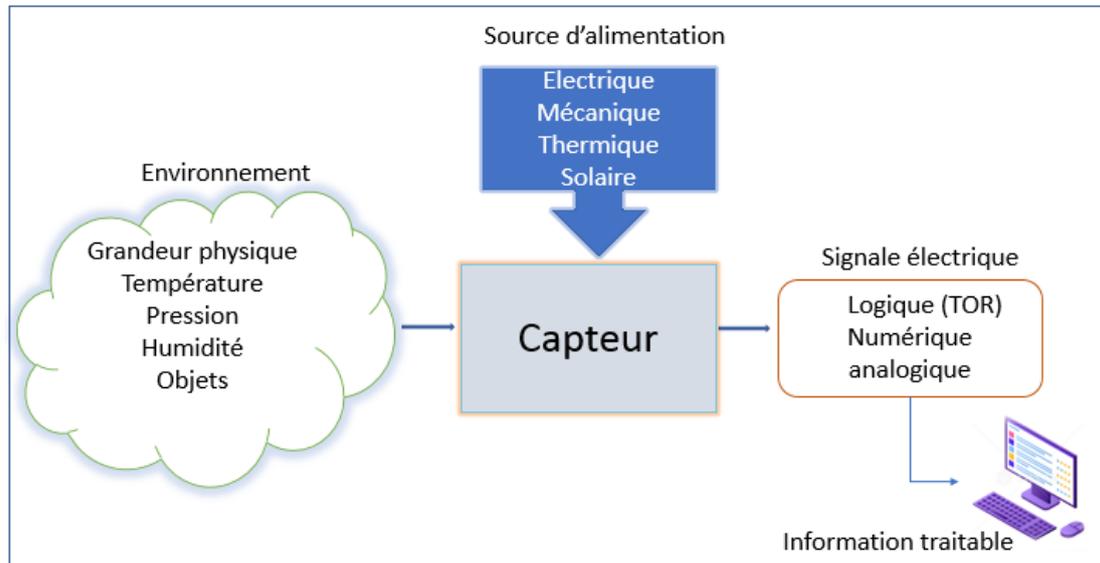


FIGURE 1.1: Exemple d'un capteur

1.2.2 Architecture matérielle

L'architecture matérielle d'un capteur est composée sur quatre unités. La figure1.2 ci-dessous présenté l'architecture matérielle.

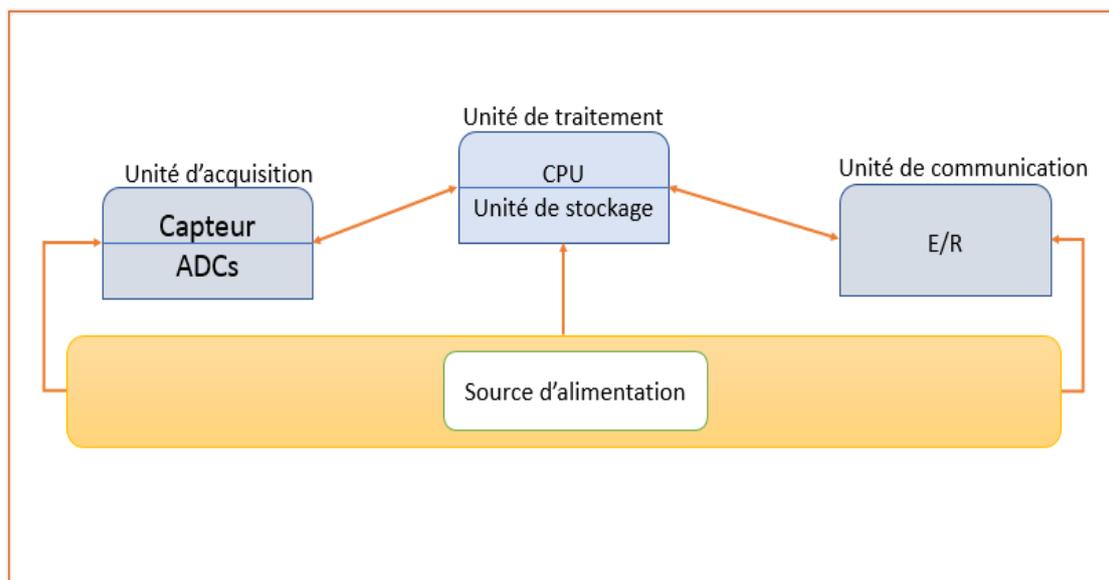


FIGURE 1.2: Architecture matérielle d'un capteur sans fil

- **Unité d’acquisition** : également appelée unité de collecte ou de captage, généralement composée de deux sous-unités : capteurs et ADC (convertisseur analogique-numérique).
 - Les capteurs permettant de mesurer les paramètres environnementaux pour fournir des signaux analogiques obtenus après dialogue avec ces données collectées.
 - ADC convertit ces signaux analogiques en signaux numériques.[2]
- **Unité de traitement** : elle contient le CPU et la mémoire de stockage, cette unité est connectée avec l’autres unités afin de contrôler leurs fonctionnalités. Le système d’exploitation nécessaire au fonctionnement du capteur peut être installé sur cette unité.[2]
- **Unité de communication** : cette unité est dotée d’un émetteur/récepteur permet d’effectuer toutes les communications entre les différents nœuds capteurs sur un canal de transmission sans fil.[2]
- **Unité de contrôle d’énergie(Batterie)** : le capteur dispose d’une source d’énergie (Batterie) pour alimenter tous ses composants. Cependant, en raison de sa taille réduite, l’énergie disponible est limitée et généralement irremplaçable.L’unité peut également gérer des systèmes de réapprovisionnement en énergie à partir de l’environnement observé, comme les cellules solaires, pour prolonger la durée de vie globale du réseau. Cette unité représente la contrainte principale lors de conception de protocole pour les réseaux de capteurs sans fil.[3]
 - Il existe des capteurs qui sont équipé d’autres composants additionnels tels que les systèmes de localisation GPS (Global Position System), système de récupération d’énergie solaire, système de mobilité.

1.2.3 Architecture logicielle

La partie logicielle comprend principalement le système d’exploitation utilisé dans ces appareils. Parmi ces systèmes :



FIGURE 1.3: Système d'exploitation conçu pour les capteurs

- **TinyOS** : est un système d'exploitation open source conçue spécifiquement pour les réseaux de capteurs sans fil. Développé par l'Université de Berkeley aux États-Unis. Sa conception est complètement achevée en NesC, un langage orienté composants proches de la bibliothèque de composants C et TinyOS parce qu'il contient le protocole réseau, il est particulièrement complet capteurs et outils de collecte de données. Ce système est dans le domaine du RCSF, d'autres systèmes d'exploitation ont également été développés principalement pour les réseaux de capteurs sans fil Contiki, MantiOS et LiteOS.[4][5]
- **Contiki** : est un système d'exploitation open source développé par Adam Dunkels. Le système est basé sur un réseau axé sur la faible consommation d'énergie utilisée dans l'oT (Internet of Things). Conkiti est conçu pour fonctionner sur des types de périphériques matériels fortement limités en mémoire, en puissance d'énergie, en capacité de traitement et en bande passante de communication.[6]
- **LiteOS** : est un système d'exploitation du type Unix, open source et interactif. Développé par Huawei Technologies Co. Est conçu pour réseaux de capteurs sans fil.[6]

1.2.4 Capteur mobile avec l'énergie renouvelable

La récupération d'énergie consiste à collecter une infime quantité d'énergie à partir de sources de l'environnement ambiant, telle que l'énergie solaire, thermique, vibrations, etc. Ces mécanismes sont efficaces pour améliorer la durée de vie de capteur.[7]



FIGURE 1.4: Capteur sans fil mobile basée sur la récupération d'énergie.

-Architecture matérielle

L'architecture du capteur mobile sans fil avec d'énergie renouvelable comprend les composants suivants mentionnés ci-dessus. 1.5 est contient des composantes additionnelles qui lui permettent de déplacer et de récolter l'énergie ambiante.

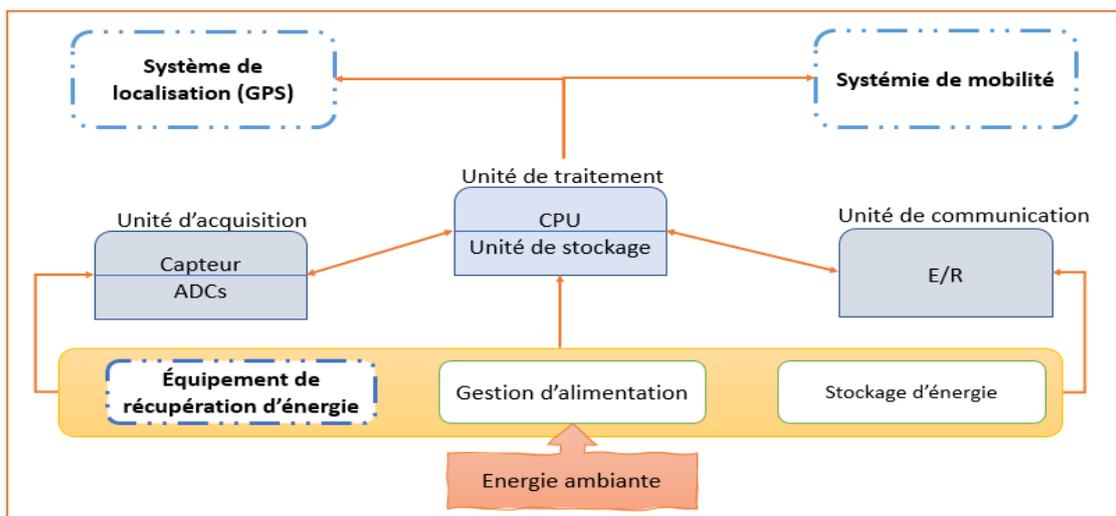


FIGURE 1.5: Architecture matérielle d'un capteur sans fil mobile basé sur la récupération d'énergie

- **Système de récupérateur d'énergie** : unité chargée pour convertir l'énergie ambiante externe en électricité.
- **Un module de gestion de l'énergie** : un système qui collecte l'énergie électrique de la récolteuse et la stockée ou alimente les autres composants du système pour une utilisation immédiate.
- **Stockage d'énergie** : pour sauvegarder l'énergie récoltée pour une utilisation future.
- **Système de localisation** : un nœud capteur soit capable de déterminer sa position géographique. Dans ce cas le capteur doit être équipé d'un système de localisation(GPS). Ou bien d'un module software implémentent des algorithmes de localisation.[8]
- **Système de mobilité** : un nœud capteur peut être doté d'un système de mobilité ou attaché dans une entité mobile chargé de le déplacer. Le support de mobilité exige des ressources énergétiques.[8]

1.3 Réseau de capteurs sans fil (RCSF)

1.3.1 Définition

Un réseau de capteurs sans fil (RCSF) est constitué d'un ensemble de nœuds de capteurs autonomes déployés dans une zone géographique ou un environnement d'intérêt pour une application précise (température, lecture de pression, contrôle, surveillance, détection d'intrusion, humidité d'air, agriculture, domestique, médecine, etc.). Les capteurs communiquent entre eux par des ondes radio, afin d'acheminer les informations au station de base(sink), qui communique avec l'interface utilisateur. L'emplacement (positionnement) de ces nœuds capteurs n'est pas nécessairement fixé mais peut être mobile. Ils peuvent être répartis au hasard (aléatoirement) sur la zone géographique appelée zone de surveillance ou zone de captage qui peut à des endroits inaccessibles.[9]

1.3.2 Architecture de réseaux de capteurs sans fil

Les réseaux de capteurs sans fil sont construits autour des quatre principales entités suivantes[10] :

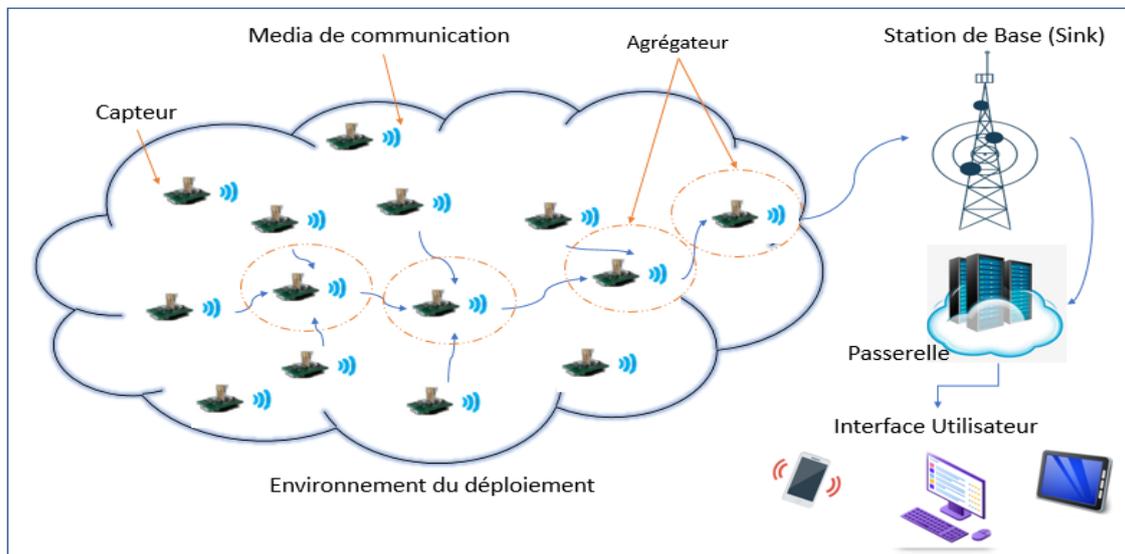


FIGURE 1.6: Un réseau de capteur sans fil.

- **Capteur(Sensor)** : il est en charge de mesurer une valeur liée à son environnement (température, luminosité, pression, etc.) comme son nom l'indique.
- **Agrégateur (cluster head)** : son rôle est de combiner les messages qu'il reçoit de plusieurs capteurs et les envoyer au station de base comme un seul message. Le principal objectif de cette opération est de réduire le trafic réseau et ainsi prolonger la durée de vie du réseau.
- **Station de base (Sink)** : c'est le nœud final du réseau. Il envoie la valeur mesurée via Internet (passerelle). dans certains cas peut être mobile.
- **Passerelle (Gateway)** : le dispositif a la particularité d'avoir deux interfaces réseaux ce qui permet de connecter le réseau de capteurs sans fil via Internet. Les applications traitent ces informations qui sont exécutées sur la machine de l'utilisateur final. Capteurs font des mesures qu'ils font remonter au station de base via les agrégateurs. L'application finale tournant sur une machine se situant sur un autre réseau à ainsi accès aux valeurs via une passerelle. Les agrégateurs sont facultatifs selon la topologie de réseau (hiérarchique), la station de base et la passerelle sont généralement localisées dans un seul dispositif.

1.3.3 Caractéristique de RCSF

Un réseau de capteurs se caractérise souvent par un déploiement dense et à grande échelle dans des environnements éloignés, dangereux et inaccessibles. Citons quelques caractéristiques de ces réseaux.

- **Durée de vie limitée** : le RCSF est conçu pour collecter les données dans les dangereux environnements où les humains ne pas toujours ont accès. Par conséquent, une fois le déploiement fait, les capteurs sont autonomes (en général, leurs batteries ne peuvent être ni rechargeables ni remplaçables) et donc, leur durée de vie dépend de la durée de vie de leurs batteries.[11]
- **Ressources limitées** : les capteurs sont des objets de communication limités en termes de (débit, transmission, puissance de calcul, stockage, énergie).
- **Agrégation des données** : les données produites par les nœuds de capteurs sont très reliées, une approche répandue consiste à agréger les données au niveau des nœuds intermédiaires pour éliminer la redondance des données et la consommation d'énergie lors de la transmission.[12]
- **Topologie dynamique** : la plupart des RCSF sont très denses. Cependant, le déploiement, la mobilité et les pannes modifient la topologie d'un réseau dont la connectivité n'est pas toujours garantie.[12]

1.3.4 Types de communication dans RCSF

Dans un réseau de capteurs sans fil, deux types de nœuds sont souvent distingués : ceux qui ne transmettent que leurs propres données (nœud de capteurs) et ceux qui transmettent des messages à d'autres nœuds (nœud relais). via le nœud intermédiaire, ce qui permet de créer une topologie multi-sauts.[13]

- **Unicast** : cette communication directe est utilisée pour les opérations locales, pour échanger des informations entre deux nœuds capteurs dans un réseau.
- **Broadcast** : la station de base diffuse des informations vers tous les nœuds du réseau. Ces informations peuvent être des requêtes de données, des mises à jour de programmes ou des paquets de contrôle.
- **Multicast** : c'est une communication entre un nœud et un groupe de nœuds dans le cas de la topologie hiérarchique lorsqu'un chef de groupe envoie les requêtes de contrôle à ces membres.

1.4 Réseau de capteurs sans fil mobile

Un réseau mobile, appelé généralement Mobile ad hoc NETWORK (MANET), est un ensemble de nœuds mobiles qui se déplacent sur n'importe quel territoire de manière autonome et coopérative sans utiliser d'infrastructure préexistante ou de gestion centralisée. Les ondes radio qui se propagent entre les nœuds mobiles sont le seul moyen de communication. Une fois qu'un groupe de nœuds mobile sont à portée radio les uns des autres, le réseau se forme spontanément mais temporairement. Il existe deux modes de communication entre deux nœuds mobiles, en fonction de la distance qui les sépare. Lorsque deux nœuds sont à portée radio et peuvent communiquer directement, ce mode est appelé transmission monôme-saut. En revanche, si deux nœuds ne sont pas à portée, ils doivent utiliser d'autres nœuds mobiles comme relais pour assurer la communication et acheminer les paquets de données vers la destination, ce mode est appelé transmission multi-sauts la figure 1.7 montre un exemple de changement de la topologie à cause de la mobilité des nœuds capteurs sans fil. Chaque fois qu'un nœud bouge, les tables de routage doivent être recalculées pour prendre en charge les mises à jour.[14]

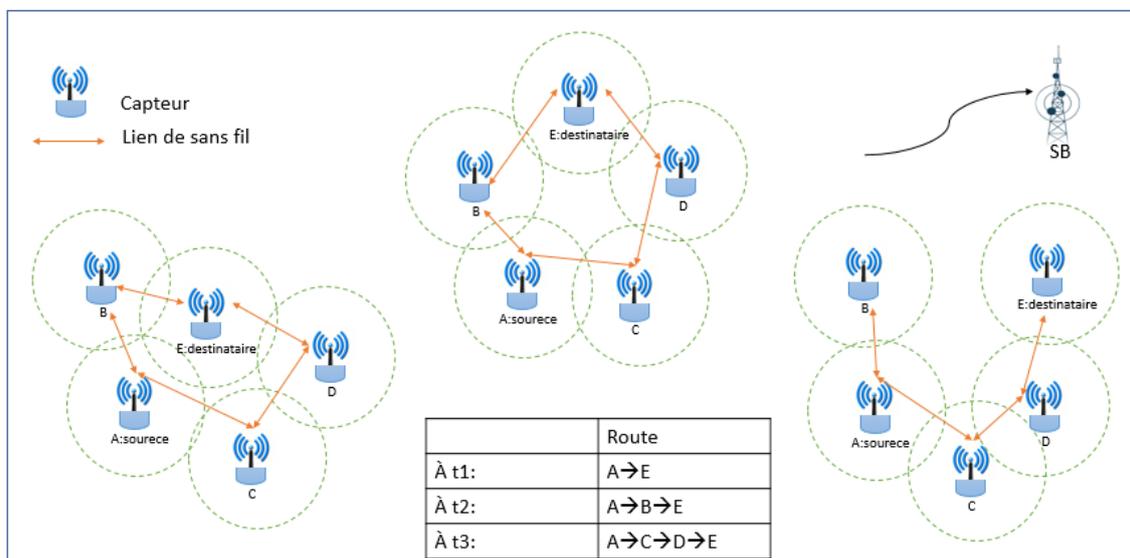


FIGURE 1.7: Exemple de changement de la topologie causé par la mobilité

1.4.1 Caractéristiques générales de la mobilité dans les RCSF

la mobilité dans les RCSFs peut être classée selon deux aspects, le type de l'élément mobile et le type de mouvement

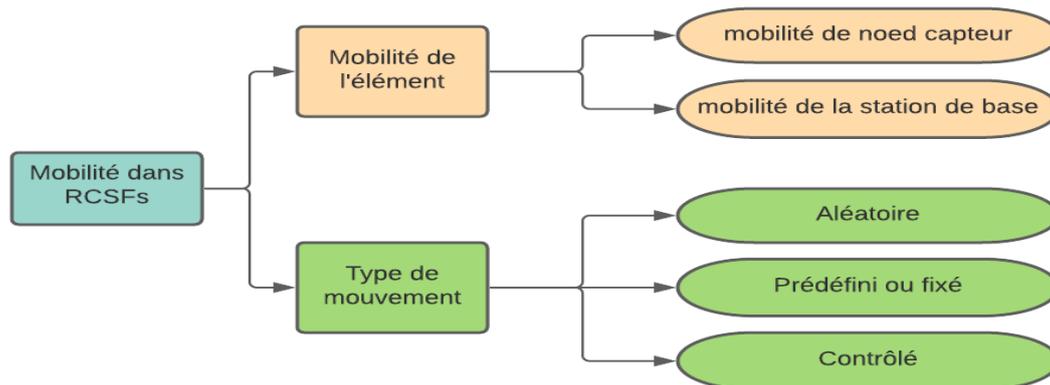


FIGURE 1.8: Classification de la mobilité dans les RCSF

1.4.2 Type d'élément mobile

La mobilité peut être intégrée au niveau des nœuds capteurs et/ou au niveau de la station de base.

- **Mobilité de la station de base** : la mobilité de la station de base est une approche introduite pour éviter le coût élevé du maintien de longs chemins multi-sauts afin d'économiser l'énergie.[8]

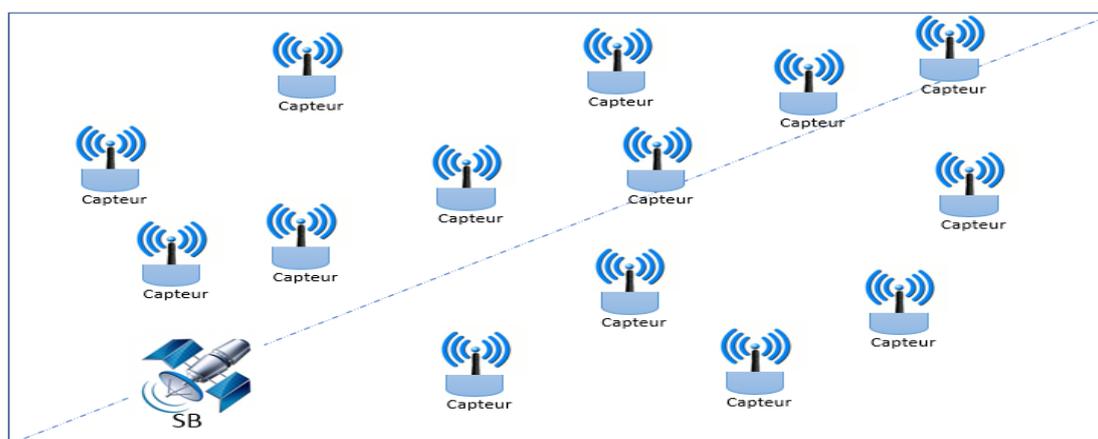


FIGURE 1.9: Architecture d'un RCSF avec station de base mobile

- **Mobilité des nœuds capteurs** : la mobilité d'un nœud capteur devient possible si le nœud a la capacité de se déplacer seul ou s'il est attaché à une entité mobile.[8]

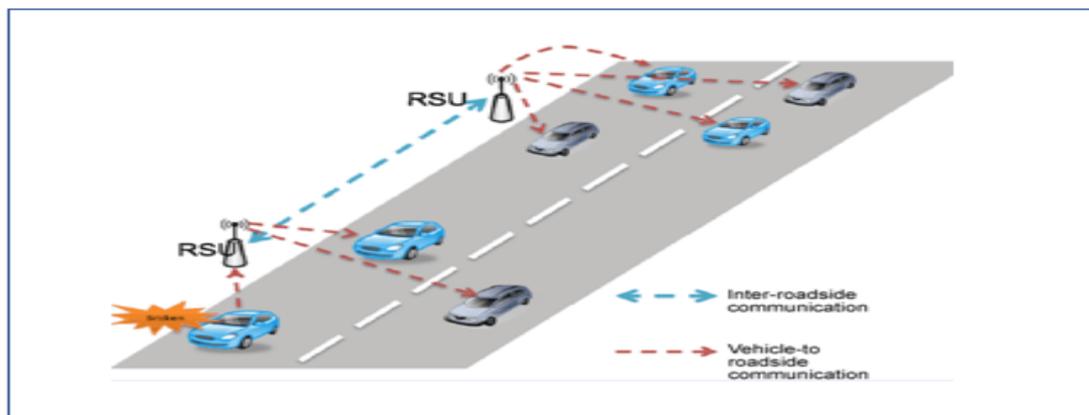


FIGURE 1.10: Architecture d'un RCSF avec des capteurs mobile

1.4.3 Types de mouvements

La mobilité dans les RCSFs peut également être classée en fonction du type de mouvement de l'élément mobile. Les types les plus cités dans la littérature sont : aléatoire, prédéfini et contrôlé.

- **Mouvement aléatoire** : signifie que l'entité en mouvement (qu'il s'agisse d'une station de base ou d'un capteur) se déplace de manière aléatoire dans la zone d'intérêt.
- **Mouvement prédéfini** : signifie que l'entité se déplace sur un chemin spécifique, avec une vitesse connue, atteignant chaque point d'intérêt à des instants connus et spécifiques.
- **Mouvement contrôlé** : signifie que le mouvement de l'entité est contrôlé par une entité externe en temps réel.

1.5 Domaines d'application des RCSF

Les RCSF peuvent être d'une large gamme de types de capteurs capables de surveiller une variété de paramètres. La température, l'humidité, la pression, le mouvement des véhicules, le niveau de bruit, la présence ou l'absence d'objets ce qui permet à ce genre de réseaux d'occuper rapidement plusieurs domaines d'application la figure 1.11 présente quelques exemples d'applications RCSF fixe et mobile :



FIGURE 1.11: Domaine d'applications des RCSF

1. **Domaine militaire** : les nœuds capteurs de ce type de réseau peuvent être des soldats, des chars, des avions de chasse ,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.[17]
2. **Domaine environnemental** : micro-capteurs distribué par avion dans une zone difficile d'accès peuvent surveiller des catastrophes naturelles.[15]
3. **Domaine médical** : la recherche applicative des capteurs intelligents dans le domaine médical comprend l'hospitalisation à domicile, l'intégration de micro-capteurs dans le corps (par exemple, la mise en place d'un BAN-Body Area Network) et la gestion des urgences [40]. Le domaine médical se développe rapidement. Ainsi, parmi les nouveaux arrivants, on peut citer le suivi à distance des signes vitaux et des niveaux d'activité à domicile par les personnes âgées ou handicapées, le déploiement de micro-capteurs pouvant être implantés sur le corps humain permet un suivi permanent des constantes importantes des patients.[16].
4. **Domaine aéronautique** : en termes de fiabilité technique et de robustesse, l'industrie aéronautique est une industrie très exigeante. Si la proportion d'accidents d'avion est bien inférieure à celle du transport routier, on estime des défaillances matérielles et 25% des erreurs humaines peuvent être évitées ou réduites par un suivi en temps réel des caractéristiques des avions en vol. Cela signifie qu'il est important de concevoir un capteur multiparamétrique miniaturisé qui consomme moins d'énergie et qui soit réparti dans différentes parties de l'avion. Ces capteurs

seront intégrés sur un substrat souple. Certains composants ne sont pas planaires, mais à terme, ils seront aussi directement intégrés dans la fibre composite de la plateforme.[41]

5. **Les interventions d'urgence** : opérations de recherche, de secourisme et de sauvetage en urgence durant les catastrophes naturelles comme lors du tremblement de terre, feux, inondation, etc.[17]
6. **VANET Vehicular Ad hoc NETWORKS (VANET)** : fournit un exemple de ce type. VANET est un groupe de véhicules qui forment un système de transport intelligent à portée de chacun. Le plus grand avantage du VANET est de contribuer à l'avenir à des routes sûres et efficaces en fournissant des informations en temps opportun aux conducteurs et aux autorités intéressées.[17]

1.6 L'énergie solaire

1.6.1 Définition

”L'énergie solaire photovoltaïque provient de la transformation directe d'une partie de la lumière en électricité. Cette conversion s'effectue par le biais d'une cellule dite photovoltaïque (PV) basé sur un phénomène physique appelé effet photovoltaïque qui consiste à produire une force électromotrice lorsque la cellule est exposée à la lumière. La tension générée peut varier en fonction du matériau utilisé pour la fabrication de la cellule”. [19] À partir de cette base les chercheurs ont développé plusieurs méthodes de récupération d'énergie chaque méthode de récupération d'énergie a un système spécifique qu'on détaille dans la suite.

1.6.2 ISEH system (Intelligent Solar Energy Harvesting)

ISEH(Intelligent Solar Energy Harvesting) est un système de récupération d'énergie solaire peut produire une énergie stable et à long terme. Le système se compose de panneaux solaires, de batteries au lithium et de circuits de commande. La gestion de la charge de la batterie au lithium utilise du matériel au lieu du logiciel, ce qui améliore la fiabilité et la stabilité du système. Il préfère utiliser l'énergie solaire lorsque le soleil est suffisant, la batterie au lithium est une alimentation électrique complémentaire pour des conditions

telles que le ciel couvert, la pluie et la nuit le système adopte le circuit de suivi du point de puissance maximale,(MPPT) utilise pleinement l'énergie solaire et assure la durée de vie extrêmement longue de la batterie au lithium grâce à une méthode de charge appropriée et raccourcit la fréquence de cycle. Le système peut être réalisé par de petits équipements électriques, particulièrement adaptés aux nœuds de capteurs externes dans l'internet des objets (Internet of Thing IoT).[18]

1.6.3 Architecture d'un ISEH system

Le système ISEH est en fait composé de panneaux solaires, de batteries au lithium et de circuits de commande. Le circuit de commande comprend des modules MPPT solaire, sous-circuit de charge, sous-circuit de protection contre les décharges excessives et module de suralimentation DC/DC pour batterie au lithium. Le schéma fonctionnel du système est montré dans la figure1.12, le système a trois branches d'entrée, à savoir le panneau solaire, la batterie au lithium et l'interface mini USB. Il dispose également d'une branche de sortie fournie par une interface USB normale. Dans cet article, nous utilisons le bimodule pour souligner que la sortie du système peut provenir respectivement de panneaux solaires ou de batteries au lithium. Si nécessaire, l'interface mini USB standard est réservée au chargement de la batterie au lithium via un adaptateur secteur externe. Les composants fonctionnels du système sont décrits comme suit :[18]

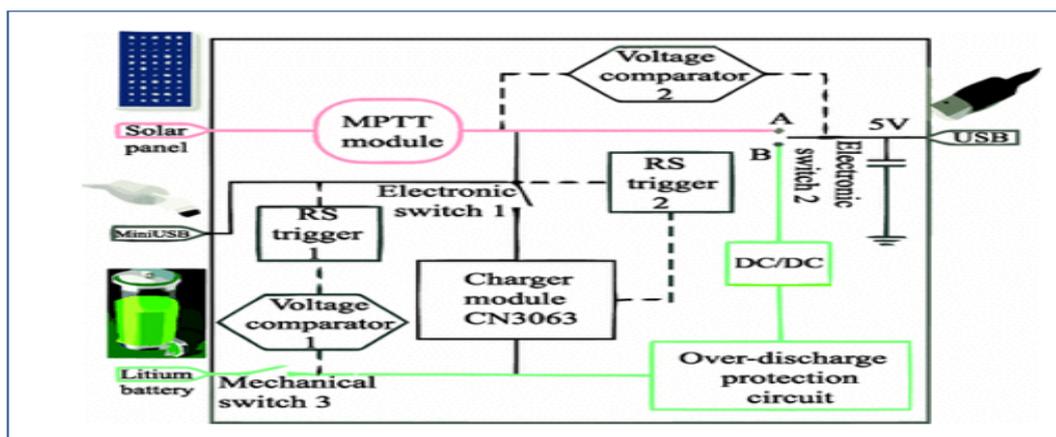


FIGURE 1.12: Architecture de ISEH système

1.7 Consommation d'énergie

1.7.1 L'énergie ambiante

Une autre alternative consiste à extraire de l'énergie (Energy harvesting ou energy scavenging) à partir de l'environnement de déploiement et alimenter ensuite les nœuds capteurs. La source d'énergie extraire est multiple [20]. Le tableau 1.1 montre quelques exemples de densités mesurées dans les meilleurs cas.

Technologies de l'extraction d'énergie	Densité de la puissance
Cellule solaire (à l'extérieur vers midi)	$15mW / m^3$
Piézoélectrique	$330\mu W / m^3$
Vibration (petite four a micro onde)	$160\mu W / m^3$
Thermoélectrique	$40\mu W / m^3$
Bruit acoustique	$960nW / m^3$

TABLE 1.1: Densité de puissance pour certaines sources d'énergie ambiante

1.7.2 Les principales opérations dues à la consommation d'énergie

L'énergie consommée par les nœud capteur est montre dans la figure 1.13 [21].

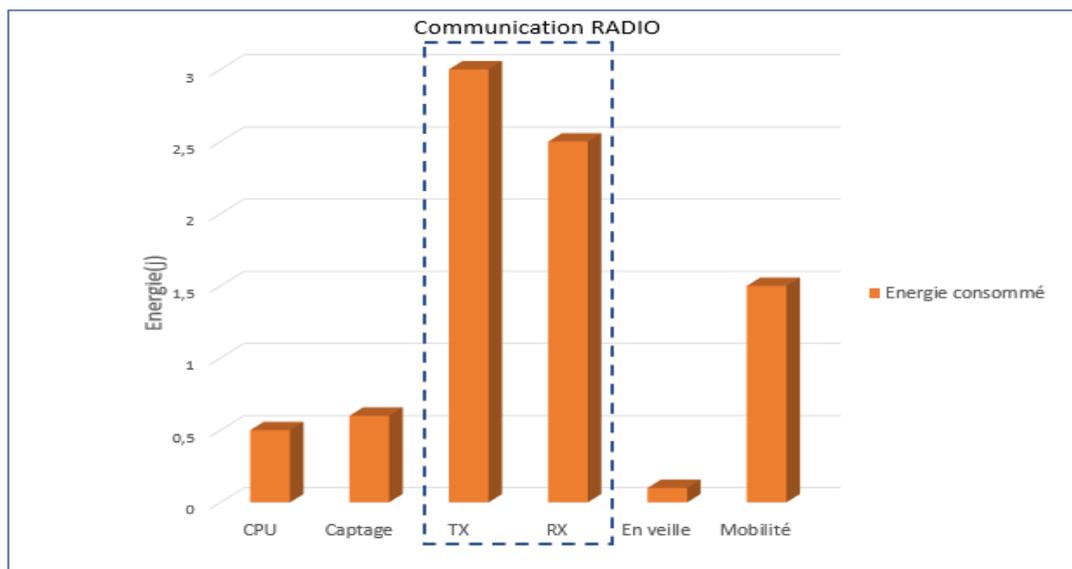


FIGURE 1.13: Niveaux de consommation d'énergie au sein d'un nœud capteur

- **Énergie de captage** : l'énergie consommée par les opérations échantillonnage, traitement du signal, conversion analogique-numérique et activation de la sonde de capture. généralement et négligeable.
- **Énergie de traitement** : énergie de traitement est divisée en deux parties : énergie de commutation et énergie de fuite. 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. noter que l'énergie de communication représente la plus grande partie de l'énergie consommée par le nœud de capteur.
- **Énergie de mobilité** : la mobilité est un facteur qui affecte également l'énergie consommée proportionnellement à la distance et à la vitesse de déplacement.

1.7.3 Consommation d'énergie dans les RCSF avec et sans énergie renouvelable

La consommation d'énergie dans les réseaux de capteurs sans fil dépend au genre de capteur :

- En cas des capteurs normaux, c'est-à-dire n'a qu'un taux d'auto-décharge négligeable. Donc la durée de vie des capteurs diminuer progressivement jusqu'à la mort du capteur.
- Les nœuds de capteurs contiennent également un dispositif de récupération d'énergie collectées dans l'environnement. Il est permis de recharger les batteries lui permettant de prolonger sa durée de vie.

1.7.4 Modèle de consommation d'énergie

Dans le domaine des radios de faible puissance, des recherches ont été menées. Heinzelman et coll. a proposé un modèle dans lequel la radio absorbe :

$E_{elec} = 50 \text{ nJ/bit}$ (où $1 \text{ nJ} = 10^{-9} \text{ J}$). pour mettre en œuvre le circuit (émetteur/récepteur)

$E_{mp} = 100 \text{ pJ/bit/m}^2$ (où $1 \text{ pJ} = 10^{-12} \text{ J}$) est utilisé pour l'amplificateur d'émetteur. [22] Si

la distance est courte entre les nœuds de capteur et la tête de cluster, nous utilisons un modèle d'espace libre, sinon le modèle multi-chemins est utilisé pour la communication.

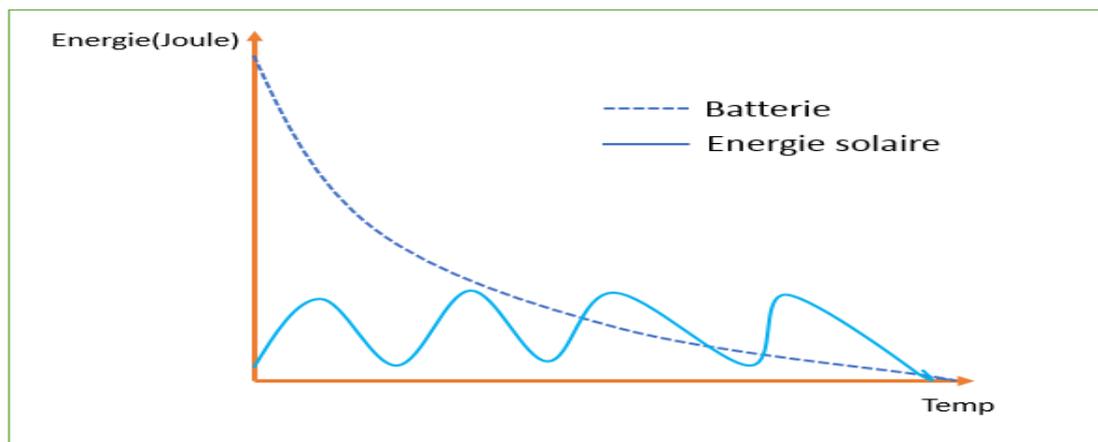


FIGURE 1.14: Consommation d'énergie dans les RCSF avec et sans énergie renouvelable

La consommation d'énergie lors de la transmission du nombre 's' de bits à un nœud placé à la distance 'd' mètres s'écrit :

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

$$E_{Tx}(s, d) = \begin{cases} E_{ele} * s + E_{fs} * s * d^2, & d \leq d_o \\ E_{ele} * s + E_{mp} * s * d^4, & d > d_o \end{cases} \quad (1.2)$$

$$E_{Rx}(s) = E_{Rx.ele}(s) + s * E_{ele} = E_{ele} * s \quad (1.3)$$

Ou :

- E_{ele} :est l'épuisement énergétique de l'émetteur ou du récepteur par bit.
- E_{mp} :est pour l'évanouissement par trajets multiples(Amplification)
- E_{fs} :est pour l'espace libre.
- E_{Tx} :est la dissipation d'énergie côté émetteur.
- E_{Rx} :est la dissipation d'énergie par bit du côté récepteur.
- s : est des données en bits.
- d :est la distance entre un nœud émetteur et récepteur.

On suppose également que la couche MAC idéale fonctionne pour une transmission sans collision. Lorsque la transmission se fait par saut unique, il n'y a pas de nœud relais entre les capteurs et la stations de base. Pour calculer si le nœud utilise un seul ou plusieurs sauts, nous utilisons 'do'. La valeur seuil pour 'do' est calculée comme selon la formule suivant :[29]

$$d_o = \sqrt{\frac{E_{fs}}{E_{mp}}} \quad (1.4)$$

La figure 1.15 ci-dessous illustre ce modèle de consommation d'énergie.

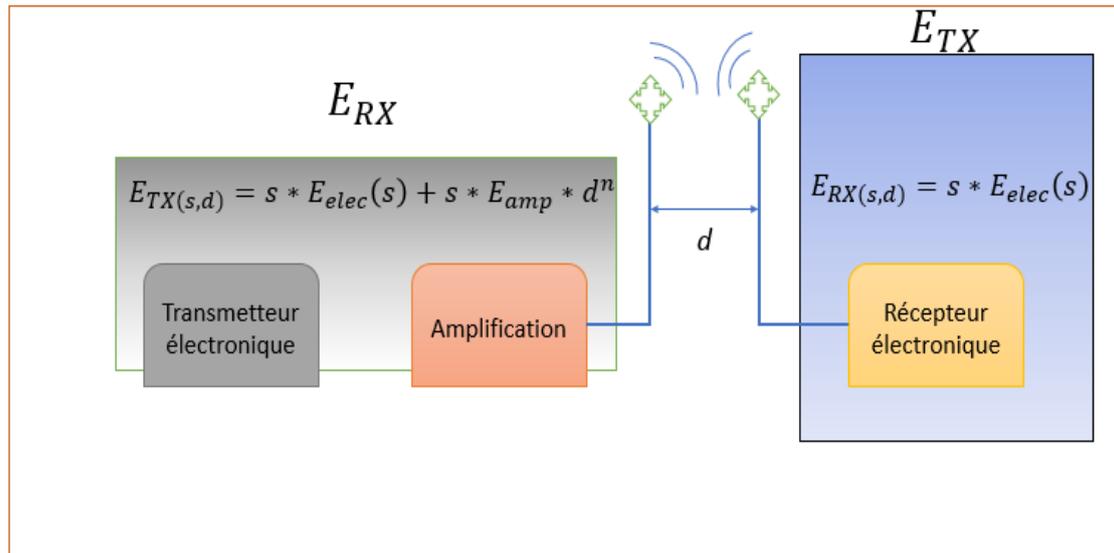


FIGURE 1.15: Modèle de consommation d'énergie.

1.8 Conclusion

Dans ce chapitre, nous avons parlé des concepts initiaux et des informations de base liées au domaine du réseau de capteurs sans fil (RCSF), tels que la définition du capteur, leurs composants, la définition d'un réseau de capteurs sans fil fixe et mobile. Nous avons également donné un aperçu sur le domaine d'application et le modèle de consommation d'énergie dans cette technologie dans le chapitre suivant, nous parlerons sur le routage.

Routage dans RCSF

2.1 Introduction

Le réseau de capteurs sans fil (RCSF) est un réseau informatique spécial, largement utilisé en raison de sa grande capacité. RCSF se compose de nombreux petits nœuds de capteurs avec des ressources limitées. Ces nœuds capteurs peuvent détecter des paramètres environnementaux, tels que l'humidité, la température et la pression. Bien que des améliorations aient été apportées aux RCSF, en raison du grand nombre et de la petite taille des nœuds de capteurs, ils nécessitent des batteries de faible puissance. Du fait de l'utilisation de ces réseaux dans un environnement difficile et indisponible, il n'est pas possible de recharger ou de changer les batteries. Par conséquent la limitation d'énergie est le problème principal de cette technologie. Pour cette raison, plusieurs technologies et protocoles de routage ont été proposés pour le RCSF afin de maximiser la durée de vie du réseau et de minimiser la consommation d'énergie. Dans ce chapitre, nous présentons la technologie de routage dans RCSF et les différents protocoles utilisés.

2.2 Routage dans RCSF

Le RCSF se compose de nombreux petits réseaux locaux. Les nœuds du réseau de capteurs sont connectés les uns aux autres via une liaison sans fil utilisant des ondes radio, ce qui ne nécessite pas d'infrastructure de réseau fixe dans laquelle on parle dans ce cas de réseau ad hoc. En ce qui concerne l'architecture des nœuds dans RCSF a une portée

de transmission courte et limitée. Il a également une faible puissance de traitement, une capacité de stockage et des ressources énergétiques limitées. L'accord RCSF est essentiel pour garantir la fiabilité et la communication multi-sauts entre les nœuds. Pour cela des algorithmes de routage avancé sont nécessaires.

2.2.1 Définition de routage

Le routage est le processus de définition d'un chemin dans le réseau pour transférer des données d'un expéditeur à un ou plusieurs destinataires. On parle de routage dans différents domaines : réseaux téléphoniques, réseaux électroniques (comme Internet), réseaux de transmission, etc.

-Le terme routage fait référence à tous les mécanismes appliqués dans le réseau pour déterminer le chemin des paquets de données de la station émettrice à la station réceptrice. Il existe généralement deux entités : les algorithmes de routage et les protocoles de routage.[23]

2.2.2 Classification des protocoles de routage dans les RCSF

Les protocoles de routage peuvent être classifiés selon la topologie dans laquelle ils opèrent ou selon leur fonctionnement.

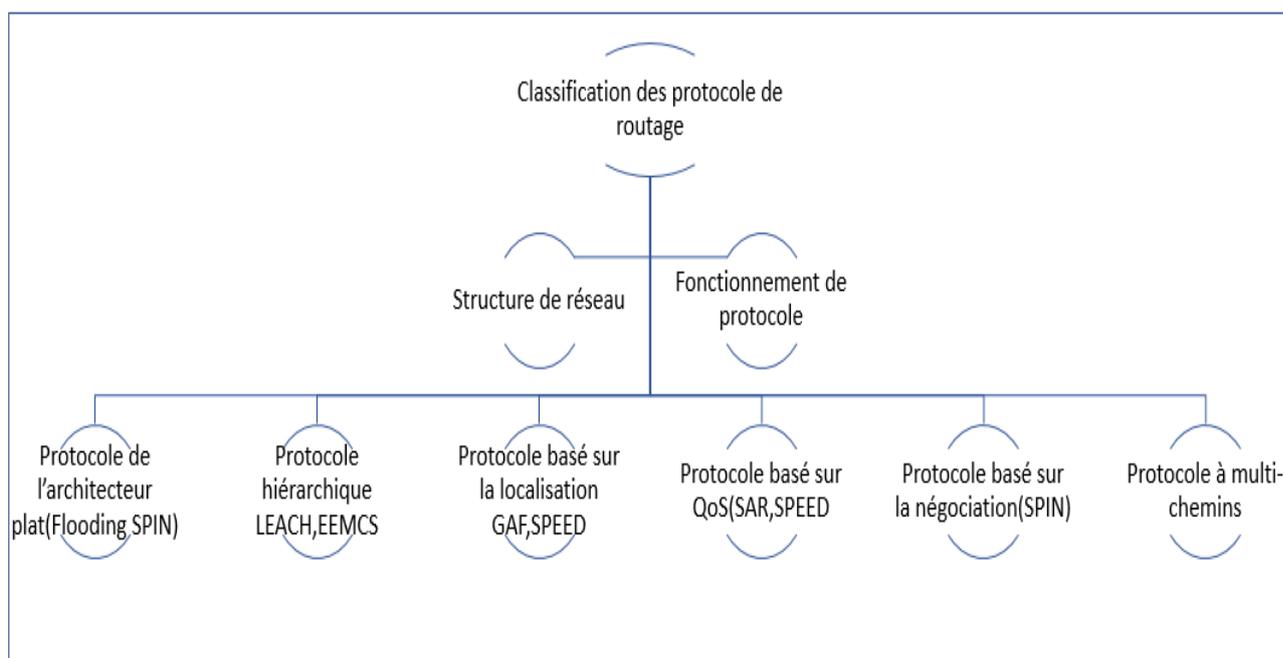


FIGURE 2.1: Classification des protocoles de routage.

A-Selon La structure de réseau

La topologie du réseau détermine la manière dont les capteurs sont organisés et joue un rôle indispensable dans le fonctionnement de protocole. Il y en a trois en tout dans RCSF, la structure est définie comme suit : topologie à plat, topologie hiérarchique et topologie basée sur la géographique.

1. **Protocoles de l'architecture plate** : les données sont envoyées en fonction de leur contenu plutôt que de l'adresse du destinataire. Si la demande inclut l'exécution d'une tâche spécifique, le nœud d'exécution de la commande (telle que la température) est gérée par cette demande. Lorsque la source envoie les paquets de données à la destination, le support peut collecter plusieurs paquets de données d'une manière spécifique pour réduire le coût énergétique. Il existe plusieurs types de protocoles dans cette catégorie, exemple : le protocole inondation (le protocole Flooding). [24]

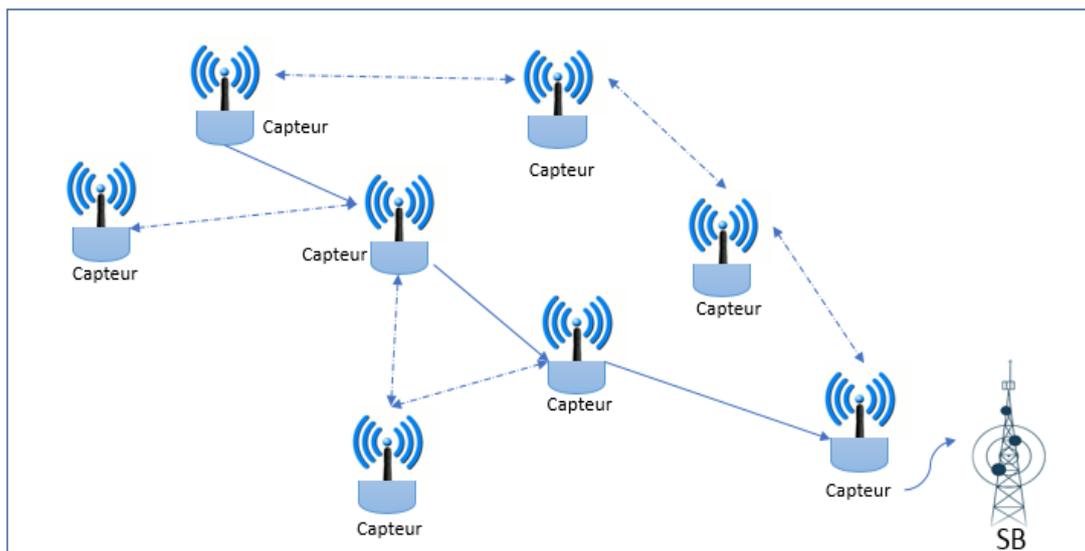


FIGURE 2.2: Architecture à plate

2. **Protocoles de l'architecture hiérarchique** : les nœuds sont regroupés en groupes supervisés par des chefs de groupe. Le développement du clustering dépend de la quantité d'énergie stockée et de la distance entre chaque nœud et les en-têtes de cluster. Il existe plusieurs types de protocoles dans cette catégorie : Exemple le protocole LEACH, EEMCS.

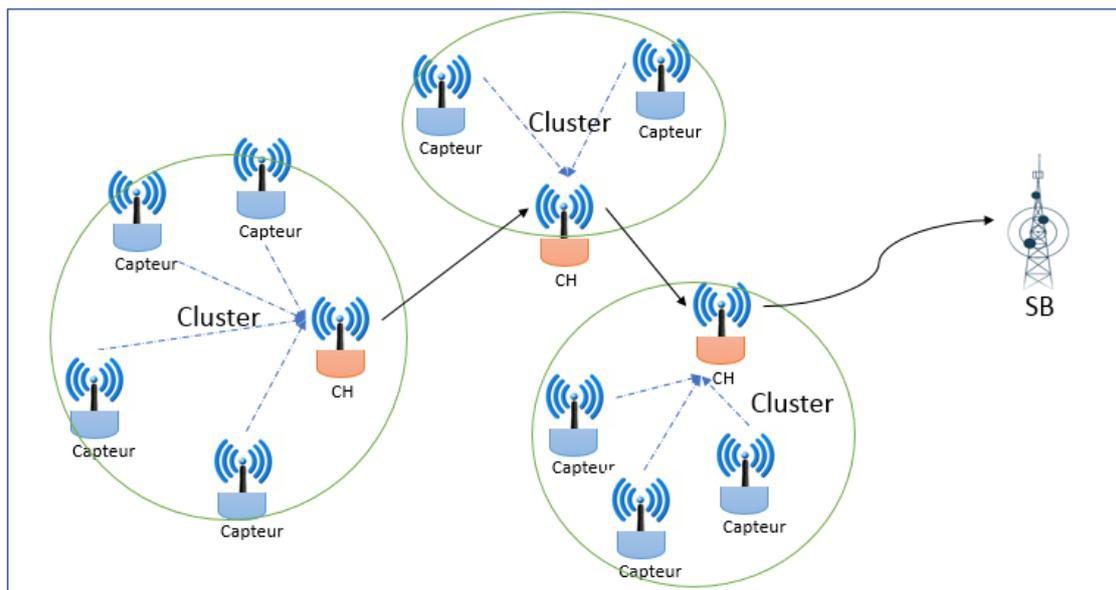


FIGURE 2.3: Architecture hiérarchique.

3. **Protocoles basé sur la géographique(localisation)** : a méthode suppose que chaque capteur a une connaissance précise de son emplacement, grâce à la technologie GPS. Sinon, une autre méthode est utilisée par laquelle le positionnement est garanti en estimant la distance entre les deux capteurs. L'estimation est basée sur les caractéristiques du signal reçu (temps de retard, atténuation du signal, etc.), ou sur l'angle d'arriver du signal. Exemple le protocole Le GAF.[25]

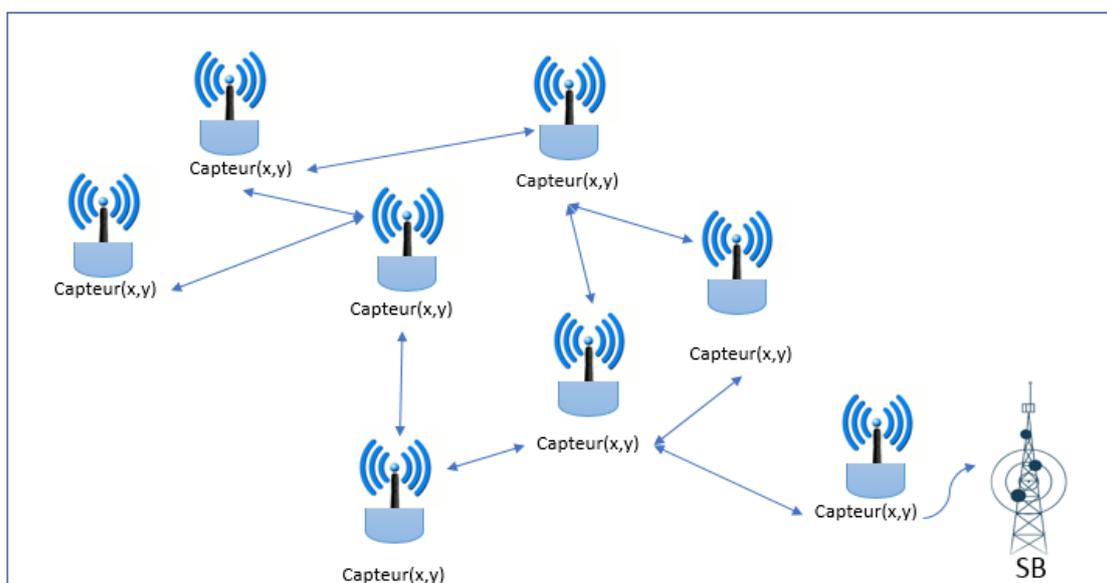


FIGURE 2.4: Architecture basé sur la localisation

B-Selon le fonctionnement

Le fonctionnement du protocole de routage dans un réseau de capteurs peut être utilisé comme un facteur de classement parce que le mode de fonctionnement détermine la manière dont les données sont réparties sur le réseau. Les protocoles de routage peuvent être classifiés en trois catégories en fonction de ce critère :

- Routage basé sur la qualité de service (Qualité of Service) (QoS).
- Routage basé sur le multi-chemins (Multi-path routing).
- Routage basé sur la négociation (Négociation based routing).[26]

1. **Protocoles de routage multi-chemins** : les données peuvent être acheminées de la source vers la destination, via des chemins alternatifs. Ces derniers sont maintenus par l'envoi périodique des messages . Malgré l'augmentation de la fiabilité et la robustesse du réseau, l'existence de chemins multiples provoquent la perte additionnelle d'énergie. [27]
2. **Protocoles basés sur la négociation** : l'objectif principal de ce type est basé sur la suppression des données redondantes. Les capteurs négocient entre eux leurs données en échangeant des paquets de signalisations spéciales ces paquets permettent de vérifier si les nœuds voisins disposent de mêmes données à transmettre. Cette procédure garantit la transmission des données utiles, mais elle risque une perte d'énergie.[28]
3. **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 prennent en compte d'autres critères exemple : la qualité de service, le retard de transmission, la fiabilité, la bande passante et vitesse de propagation.

2.2.3 Facteurs de conception de protocole de routage

La conception et la mise en place des protocoles sont influencées par plusieurs contraintes. Ces facteurs importants ils sont considérés également comme métriques de comparaison de performances entre les différents travaux dans le domaine.

- **Tolérance ou panne** : certains nœuds capteurs peuvent générer des erreurs ou ne plus fonctionner. Ces problèmes ne doivent pas affecter le reste du réseau.

- **Consommation d'énergie** : l'efficacité énergétique est un facteur clé de performance. Le concepteur peut choisir de ne pas tenir compte d'autres facteurs dans le but d'éviter cette contrainte.
- **Limitation de capacité** : les nœuds de capteurs nécessitent généralement peu de capacité de stockage et de calcul. Le concepteur du protocole de routage doit prendre en compte des opérations simples qui nécessitent peu de capacité de calcul et de stockage.
- **Scalabilité** : L'application de RCSF nécessite généralement le déploiement de nombreux nœuds, ce qui dans certains cas, conduit à un changement soudain de la structure du réseau en raison de dommages ou de l'ajout d'un nouveau nœud. Par conséquent, le protocole de routage doit avoir une évolutivité élevée. En d'autres termes, le protocole de routage ne devrait pas souffrir de dégradation des performances lorsque les nœuds sont endommagés et que le nombre de nœuds est plus grand.
- **Connectivité** : nœuds sont généralement dispersés au hasard et répartis sur la zone de déploiement. Certaines zones de l'espace de déploiement peuvent bénéficier d'une meilleure connectivité. Le protocole de routage conçu pour RCSF doit avoir la capacité de s'auto organiser pour s'adapter à la topologie dynamique.
- **Hétérogénéité** : les nœuds peuvent avoir différents rôles, de sorte que la conception des protocoles de routage doit prendre en compte les différents types de nœuds et les limitations qui en résultent.

2.3 Travaux connexes

Plusieurs protocoles de routage hiérarchique basé sur des clusters ont été développés pour rendre le réseau efficace énergiquement et prolonger la durée de vie du réseau en considérant différents paramètres tels que l'énergie, la distance, la couverture, la mobilité, la densité, etc. Dans la littérature, de nombreux algorithmes ont été proposés pour traiter avec des problèmes d'efficacité énergétique dans des domaines spécifiques quelques-uns de ces protocoles sont discutés dans cette section.[29]

2.3.1 Protocoles hiérarchique basé sur le clustering

-LEACH(Low-Energy Adaptive Clustering Hierarchy)

Le protocole LEACH est le protocole de routage hiérarchique le plus courant, proposé par Heinzelman et al (2000). le protocole LEACH divise les nœuds de réseau en deux catégories : les chefs de groupes et les nœuds membres

-Les chefs de groupe(cluster head) :joue le rôle de router. Il permet de collecter et de compresser les informations reçues de son groupe et puis les envoie à la station de base.

-Les nœuds membre : chargé de collecter des données de leur environnement et les envoyé à son chef.

Le protocole se déroule en rounds. Chaque round se compose de deux phases :la formation des clustres et la communication.[30]

1. **Formation des clusters** : divisé en trois sous étapes : la sélection de CH, formation de cluster , assignation de TDMA(Time-division multiple access).

-La sélection de CH : les CH peuvent être choisies de différentes manières selon le protocole utilisé. Dans ce cas le CH choisi par un calculer de probabilité selon la formule suivante :

$$T(S_i) = \begin{cases} \frac{p}{1-p(r \bmod \frac{1}{p})} & \text{if } s \in G \\ 0 & \text{if } s \notin G \end{cases} \quad (2.1)$$

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 CH.L'algorithme est conçu de sorte que chaque nœud devienne un CH de cluster au moi une seule fois

-Formation des clusters : à la fin du processus de sélection, chaque chef de groupe envoie un message de signalisation dans sa portée de transmission pour permettre aux nœuds d'indiquer à quel groupe il appartient, la décision d'appartenance se fait sur la base de la force de signale radio reçu(RSSI) le CH qui contient le signal plus fort(CH le plus proche) sera choisie..

-Assignation de TDMA : chaque chef de groupe crée un planning d'ordonnan-

ement et le partage avec ses membres ce planning contient le temps alloué pour chaque nœud.

2. **Phase de communication** : les membres utilisent le planning pour transférer leurs données à son chef et puis le chef de groupe les envoie directement à la station de base.

-Interférences entre les clusters : pour éviter les collisions entre différents clusters les CH utilise la technique de CDMA (code-division multiple access) qui consiste à utiliser les codes pour identifier les signaux de ces membres.

2.3.2 Protocoles hiérarchique basé sur la récupération d'énergie

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

le protocole [39]. Diviser le réseau en niveaux et en zone pour assurer un routage multi-sauts entre différentes couches. Le fonctionnement du protocole s'établit en deux phases, Phase d'initialisation, Phase de routage des données.

Les étapes sont décrites en détail ci-dessous.

- **Phase d'initialisation** : dans cette phase, le protocole consiste à découper le réseau en couches pour assurer un routage multi-saut, puis divise ces niveaux en zones (groupes) où chaque zone a un (ID_{niveau}, ID_{zone}) . Cette étape n'est effectuée qu'une seule fois pendant le processus de routage.
- **Election de Cluster Head** : la station de base choisie au hasard une tête de Cluster temporaire TCH dans chaque cluster, son rôle est d'élire le CH. 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, la formule d'élection $Electe_{node}$ comme suit :

$$Electe_{node} = \frac{E_{nodei}}{DisAvecAutri} \quad (2.2)$$

Où :

- E_{nodei} : est l'énergie résiduelle du nœud i .

- *DisAvecAutri* : est la distance entre le nœud i et les autres du même cluster.
- **Phase de routage des données** : c'est la phase d'envoyer les données agrégées par chaque capteur à la station de base. Dans cette phase, le protocole utilise deux threads pour faire fonctionner le capteur.
 - Thread A : charge pour la phase de routage des données.
 - Thread B : charge pour l'énergie renouvelable.

L'algorithme de création de threads A lance un thread qui se charge du routage.

Début

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

Dans l'algorithme de thread B, utilisé la formule suivante[37]. L'algorithme choisit une variable aléatoire depuis la formule $P(s)$, et considère comme une valeur de chargement de nœud dans un temps t .

$$P(s) = |0.9 * R(t) * \cos(\frac{t}{7 * \pi}) * \cos(\frac{t}{10 * \pi})| \quad (2.3)$$

Où :

- $P(S)$: producteur source / temps, borne a une valeur maximale [0,0.9].
- $R(t)$: les valeurs aléatoires entre 0 et 1 (random).

-HASS(Harvesting Aware Speed Selection) :

Il s'agit d'un algorithme qui maximise la réserve d'énergie des nœuds du réseau tout en répondant aux exigences de performance des applications RCSF. HASS est une méthode de contrôle de l'énergie dans RCSF à performance limitée basée sur les temps, qui utilise la récupération d'énergie. Il utilise la technologie de mise à l'échelle de la tension dynamique pour économiser l'énergie de calcul en réduisant simultanément la tension d'alimentation et la fréquence du processeur, et il utilise également la mise à l'échelle de la modulation dynamique pour économiser la puissance de communication en réduisant le niveau de modulation radio. Le problème HASS est formulé comme un problème d'optimisation et résolu à l'aide d'algorithmes centralisés et distribués. En HASS, le niveau d'énergie T_i du

nœud V_i à la fin d'une période est donné par l'équation suivante[38] :

$$T_i = T_i^{init} + P_i * S - \left[\frac{S}{\pi}\right] * e_i^c \quad (2.4)$$

Où :

- T_i^{init} : est le niveau d'énergie initial de V_i .
- S : est la longueur de l'époque.
- $\frac{S}{\pi}$: sont des cadres d'une époque.
- e^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.

2.3.3 Protocoles hiérarchique basé sur la mobilité

-LEACH-M(Low-Energy Adaptive Clustering Hierarchy-Mobile) :

”Le protocole a amélioré le taux de transmission des données par rapport à LEACH mais consomme plus d'énergie. Il est formé spécifiquement pour les nœuds mobiles. La phase d'installation est la même que LEACH mais la phase stable est différente de l'ancien protocole. Dans une phase stable après chaque tour, CH vérifie la liste des données reçues et la fait correspondre avec son programme TDMA, si un nœud n'envoie pas de données dans son temps alloué, alors cet intervalle de temps est marqué. Si le même nœud reste absent au tour suivant et qu'aucune donnée n'est reçue pendant l'intervalle de temps marqué, alors ce nœud est considéré comme déplacé vers un autre cluster. CH supprime ce nœud de sa liste de cluster et planifier les emplacements pour les autres nœuds. Un nœud peut rejoindre un nouveau cluster en fonction de l'indicateur d'intensité du signal de réception (RSSI) du message diffusé. CH ajoute le nouveau nœud à venir et met à jour la liste des membres du cluster, le calendrier TDMA”, [31]

-CRPD(a novel clustering routing protocol for dynamic wireless sensor networks) :

Il s'agit d'un protocole de routage basé sur clustering pour les réseaux de capteurs sans fil dynamiques afin de réduire la consommation d'énergie et d'améliorer l'efficacité

énergétique en mettant à jour la topologie du réseau. Il se compose de quatre étapes [32] :

- **Découverte du voisin** : dans cet étape, chaque nœud diffuse un message de détection dans sa propre portée de transmission. Le message contient des informations sur ses coordonnées, son énergie. Les nœuds qui se trouvent dans cette plage qui détectent le message stockent les informations pour le nœud expéditeur.
- **Sélection de chef de cluster et formation des clusters** : CH est sélectionné sur la base de l'énergie résiduelle et du nombre de degrés voisins. Un nœud avec le plus grand degré de voisin et une énergie résiduelle supérieure au seuil d'énergie 40% est sélectionné comme tête de cluster. Ensuite, CH diffuse le message I'am-CH et les nœuds rejoignent leur chef de cluster en fonction de la force de leur signal.
- **Agrégation de données et construction de routes** : après la sélection des clusters head, la phase de collecte des données commencée. Les données peuvent être envoyées à la station de base en utilisant un seul saut si la distance est inférieure ou plusieurs sauts peuvent être utilisés si la station de base est éloignée.
- **Ré-clustering et Ré-routage** : cela peut être fait lorsqu'un nœud est mort, se déplace vers un nouvel emplacement ou rejoint un cluster. Tout d'abord, vérifiez s'il s'agit d'un nœud CH ou d'un nœud normal, puis le regroupement est effectué en fonction du type et de l'état du nœud.

-EEMCS(An Energy-Efficient Mobility-Based Cluster Head Selection for Lifetime Enhancement of Wireless Sensor Networks) :

L'objectif principal de ce protocole est la sélection efficace des têtes de cluster qui se traduit par l'amélioration de la durée de vie globale du RCSF. Le processus de sélection de tête de cluster comprend l'énergie résiduelle des nœuds capteurs, leur niveau de mobilité, le nombre de nœuds associés et la distance entre les nœuds et la station de base. Ce sont des paramètres qui ont un effet direct sur la consommation d'énergie. Ce protocole également développé à l'aide de protocole susmentionné (CRPD) ce qui traduit les mêmes étapes de fonctionnement. la sélection de la tête du cluster, la création de cluster, la collecte de données et la transmission de données.[29]

- **Calcul de l'énergie résiduel** : à chaque tour, chaque nœud recalcule son énergie restante. Lorsque les nœuds commencent à transmettre et à recevoir des données, leur énergie commence à diminuer et la consommation d'énergie de chaque nœud

est différente de celle des autres nœuds. L'énergie résiduelle est calculée à l'aide de la formule suivante :

$$Er = \frac{E_{current}}{E_{initial}} \quad (2.5)$$

— $E_{current}$: Énergie actuelle

— $E_{initial}$: Énergie initial

— E_r : Énergie résiduel

- **Calcul des distances** : la distance joue un rôle important dans le processus de consommation d'énergie des nœuds. Les nœuds les plus proches de la station de base ont besoin de moins de puissance pour transmettre des données par rapport aux nœuds éloignés. Pour calculer la distance entre le nœud récepteur et l'émetteur, la formule suivante est utilisée :

$$Distance = \sqrt{(X_j - X_i)^2 + (Y_j - Y_i)^2} \quad (2.6)$$

- **Découverte des voisins** : la phase de recherche de voisins commence lorsque la distance entre les nœuds et la station de base a été calculée. Pendant cette phase, tous les nœuds envoient un message pour détecter leurs voisins qui sont à portée de transmission radio. L'identifiant de chaque nœud est inclus dans ce message de détection. Les nœuds qui reçoivent le message de détection l'enregistrent dans leur liste de voisins avec l'identifiant du nœud expéditeur.
- **Calcul de mobilité** : le facteur de mobilité joue un rôle important dans tout réseau dynamique. Un niveau élevé de mobilité peut détruire la structure du réseau et provoquer un comportement inattendu. Le niveau de mobilité d'un nœud peut être calculé en mesurant la différence entre la position précédente et la nouvelle position du capteur du nœud. L'équation suivante est utilisée pour déterminer le niveau de mobilité d'un nœud :

$$ML = \sqrt{(x_{new} - x_{curr})^2 + (y_{new} - y_{curr})^2} \quad (2.7)$$

Où :

— (x_{new}, y_{new}) : Coordonnées de nouvelle position.

— (x_{curr}, y_{curr}) : Coordonnées de position actuelle.

- **Sélection des chefs des clusters** : l'objectif principal de la sélection des CH est de réduire la consommation d'énergie et de parvenir à une utilisation optimale des

ressources. Dans le protocole EEMCS, les têtes de clusters sont sélectionnées en tenant compte des paramètres importants tels de l'énergie restante, la mobilité, la distance à la station de base et du nombre de voisins. L'objectif est de choisir les têtes de clusters en fonction des paramètres qui affectent directement l'utilisation de l'énergie de capteurs.

- **Calcul de pondération** : la tête de cluster est formée initialement par le nœud ayant le poids le plus élevé. Le poids de chaque nœud est calculé à l'aide de l'équation ci-dessous :

$$Poid = \frac{E_r * w1 + Degree * w2}{ML * w3 + D_{toBS} * w4} \quad (2.8)$$

Où :

- E_r : Énergie résiduel.
- $Degré$: Nombre de voisins.
- ML : niveau de mobilité.
- D_{torSB} : distance vers la station de base
- $w1, w2, w3, w4$ égale respectivement 0.2, 0.2, 0.4, 0.2 des paramètres de réglage aident à obtenir la tête de cluster la plus stable, la plus équilibrée et la plus économe en énergie.
- **Ré-clustering** : comme le réseau est composé des nœuds dynamiques, nous devons faire face à diverses circonstances résultant de la mobilité des nœuds. Lorsqu'un nœud est déplacé d'un cluster à un autre ou qu'un chef de cluster est prêt à mourir, la topologie du réseau doit être modifiée et un nouveau clustering est nécessaire.

2.4 Conclusion

Dans ce chapitre nous avons défini tous les concepts fondamentaux sur le routage tel que la définition de routage classifications des déferents protocoles de routage, ensuite on a abordé quelques protocoles de routage hiérarchique existant basé sur la mobilité et d'autres basés sur la récupération d'énergie.

Le Protocole **EHMS-EEMCS** (**E**nergy **H**arvesting **M**obile **S**ink-An Energy-Efficient Mobility-Based Cluster Head Selection for Lifetime Enhancement of Wireless Sensor Networks

3.1 Introduction

Après avoir étudié dans les chapitres un et deux les généralités des réseaux de capteurs sans fil, les techniques de routage et quelques protocoles de routage existants liés à notre travail. Dans ce chapitre nous allons présenter la conception du protocole de routage que nous souhaitons mettre en place. On commence par un aperçu général et puis en détail les différentes étapes de sa conception.

3.2 Description de protocole

3.2.1 Motivation

En raison de déploiement croissant des RCSF dans presque tous les domaines, les RCSF sont devenus une composante intégrale de l'ère moderne se présentent sous diverses formes

notamment terrestres, souterraines, sous-marines, multimédias et mobiles.

La topologie de RCSF peut être statique ou dynamique. Dans un réseau statique tous les nœuds restent stationnaires pendant toute la durée de la vie du réseau. En revanche dans un réseau dynamique les nœuds capteurs sont libres de se déplacer d'une position à une autre. Par rapport aux réseaux sans fil classique le routage dans les RCSF est plus difficile.

Par conséquent, de nouvelles techniques de routage sont développées de temps en réponse aux difficultés de création de topologie. La construction incorrecte de la topologie entraîne une consommation d'énergie plus élevée et une durée de vie plus courte du réseau, la topologie hiérarchiques (clustering), l'un des mécanismes les plus populaires utilisés pour prolonger la durée de vie du RCSF. De nombreuses techniques de clustering notamment LEACH, LEACH-M, CRPD, EMCCS...etc, ont été conçues pour cette topologie.

L'objectif principal de la recherche proposée est de fournir un protocole de clustering à efficacité énergétique que nous dénommé **EHMS-EEMCS (Energy Harvesting Mobile Sink An Energy Efficient Mobility-Based Cluster Head Sélection for Lifetime Enhancement of Wireless Sensor Networks)** qui est l'amélioration de protocole de routage EEMCS. Notre protocole réduit la consommation d'énergie, prolonge la durée de vie d'un réseau de capteurs sans fil dans un environnement dynamique basé sur la technique de récupération d'énergie et la mobilité de station de base et la sélection efficace de tête de cluster en considérant différents paramètres qui ont effet direct sur l'énergie de capteur .la distance vers station de base, le niveau de mobilité de capteur, la densité ou nombre de voisins et l'énergie résiduelle.

Des simulations ont été effectuées dans l'environnement Python pour examiner les performances de notre protocole.

3.2.2 Hypothèses

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

- Les nœuds capteurs sont tous homogènes (même énergie initiale, même capacités de stockage et traitement de données.).
- les nœuds capteurs sont équipés de mécanisme de récupération d'énergie (panneaux solaires).
- Les nœuds capteurs sont répartis aléatoirement dans la zone de capture et ils sont mobiles.
- La station de base est considérée comme une ressource illimitée ni épuisable et mobile.
- 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 toutes les positions des capteurs.
- Les nœuds capteurs peuvent gérer leur puissance d'émission et de réception.

3.2.3 Principe de fonctionnement

Afin de maximiser la durée de vie du réseau, plusieurs protocoles de routage ont été proposés, mais le principal problème de routage dans le RCSF est l'énergie limitée du capteur, alors comment prolonger la durée de vie du réseau?.Et comment réduire la consommation d'énergie ?.Afin de résoudre ces problèmes. Idéalement, le nœud de capteur dispose également d'un dispositif de récupération d'énergie qui utilise l'énergie collectée dans l'environnement (telle que les cellules solaires) pour charger la batterie. La phase routage est divisé en plusieurs étapes qui inclut le déploiement de nœuds la sélection des chefs de cluster (CH), la formation des clusters et ré-clustering enfin l'agrégation des données et la communication. L'organigramme de fonctionnement générale de protocole proposée est montre dans la figure 3.1 et les étapes sont détaillées au-dessus :

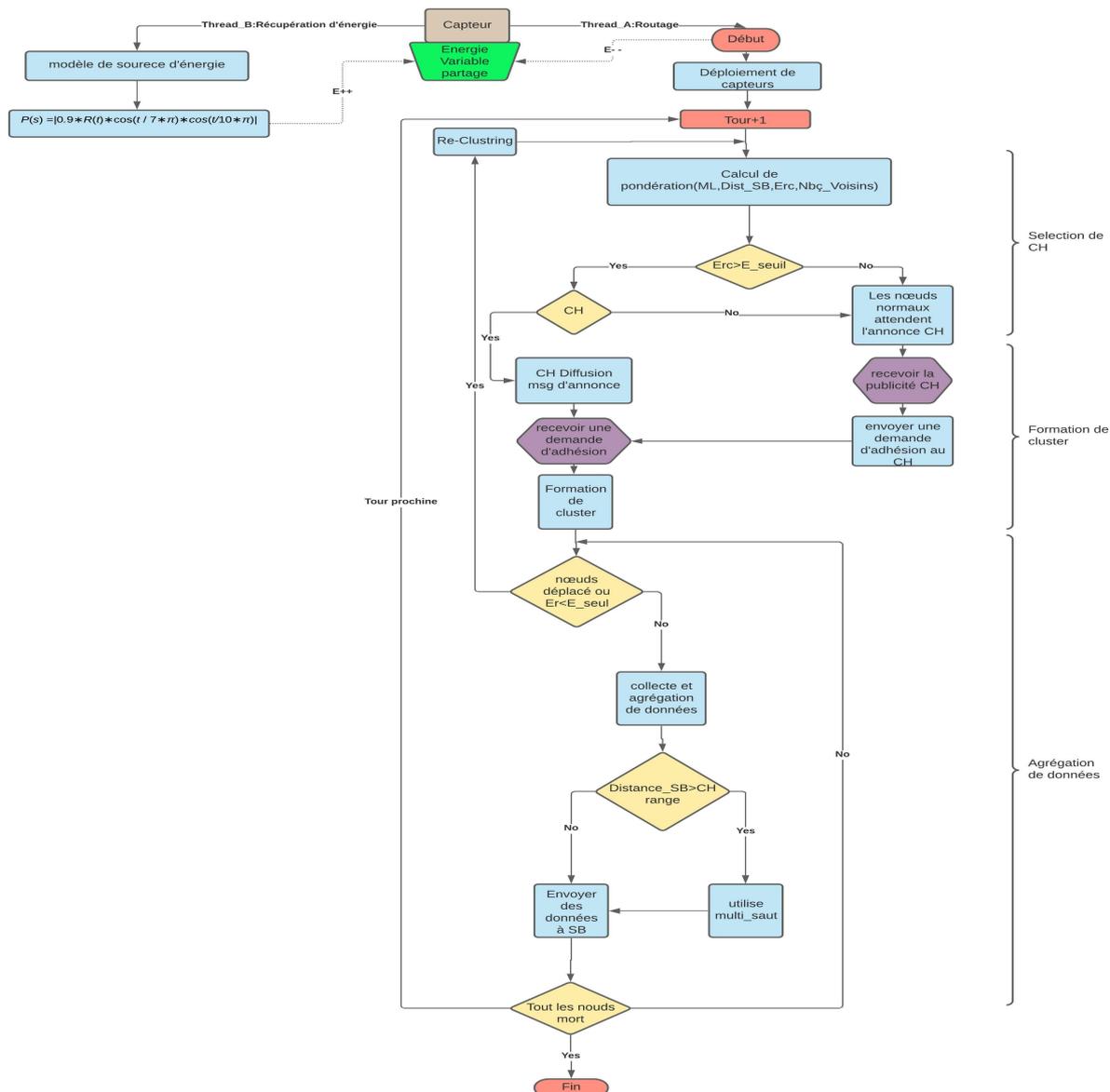


FIGURE 3.1: Organigramme de protocole EHMS-EEMCS

A.Phase de routage

1.Déploiement de nœuds

- **Création de nœuds et calcul de niveau de mobilité** : cette étape permet de crée des nœuds capteurs aléatoire et de calculer leurs niveaux de mobilité dans une zone de réseau $(X_m, Y_m)m$. Tous les nœuds sont libres de se déplacer da n'importe quelle direction selon les nouvelles coordonnées générées (x_{new}, y_{new}) . Pour calculer

le niveau de mobilité pour chaque nœud nous utilisons la formule suivante :

$$ML = \sqrt{(x_{new} - x_{curr})^2 + (y_{new} - y_{curr})^2} \quad (3.1)$$

Où :

- (x_{new}, y_{new}) :Coordonnées de nouvelle position.
- (x_{curr}, y_{curr}) :Coordonnées de position actuelle.
- **Mobilité de la station de base** : cette étape permet de crée une mouvement aléatoire linéaire pour la station de base selon la fonction suivant :

$$Y(x) = x \quad (3.2)$$

- **Calcul d'énergies résiduelles** : à chaque tour, les nœuds calculent leur énergie restante. Lorsque les nœuds commencent à transmettre et à recevoir des données, leur énergie commence à diminuer. L'énergie restante est calculée à l'aide de la formule suivante :

$$E_r = \frac{E_{current}}{E_{initial}} \quad (3.3)$$

Où :

- $E_{current}$: Énergie actuelle
- $E_{initial}$: Énergie initial
- E_r : Énergie résiduel
- **Calcul de nœuds survivant** : cette étape permet de calculer les nœuds restants survivant à chaque tour.
- **Calcul de distance et découvre de voisins** : la distance joue un rôle important dans le processus de dissipation d'énergie des nœuds. La distance euclidienne utilise pour calculer la distance entre les nœuds.

$$Distance = \sqrt{(X_{nœud_j} - X_{nœud_i})^2 + (Y_{nœud_j} - Y_{nœud_i})^2} \quad (3.4)$$

Après avoir calculé la distance entre les nœuds et la station de base, la phase de découverte de voisinage commence. Chaque nœud diffuse un message dans sa propre portée de transmission radio, les nœuds reçoivent le message considéré comme voisin. Chaque nœuds stockent les informations de leurs voisins tel que(id,

énergie... etc). à la fin de cette étape tous les nœud ont des information sur leurs voisins

- **Calcul de pondération** : après avoir calculé les différents paramètres la formule suivante permet de calculer la pondération de chaque nœud, le nœud avec la pondération la plus élevée devient candidat pour être tête de cluster :

$$Poid = \frac{E_r * w1 + Degré * w2}{ML * w3 + D_{toBS} * w4} \quad (3.5)$$

Où :

- E_r : énergie résiduel.
- $Degré$: nombre de voisins.
- ML : niveau de mobilité.
- D_{toBS} : distance vers la station de base
- w_1, w_2, w_3, w_4 : égale respectivement 0.2, 0.2, 0.4, 0.2 des paramètres de réglage aident à obtenir la tête de cluster la plus stable, la plus équilibrée et la plus économe en énergie.

2.Élection de Cluster Head

Après avoir calculé le critère défini (pondération) pour la sélection de la tête de cluster, l'énergie de chaque nœud est vérifiée par rapport à la valeur seuil 40% car un certain niveau d'énergie est la première exigence pour devenir un candidat pour la sélection de la tête de cluster. Tous les nœuds qui satisfont le seuil d'énergie sont éligibles pour la sélection de la tête de cluster.

3.Formation des clusters

Après sélection des têtes de cluster, chaque CH diffuse un message (IMCH) dans sa propre portée de transmission(demande d'adhésion), les nœuds qui reçoivent le message calculent la distance entre chaque CH et envoient la demande de jonction au CH le plus proche de lui. Le CH le plus proche est sélectionné en fonction de la force du signal reçue.

4.Re-culstring

Le réseau étant basé sur des nœuds dynamiques, nous devons faire face à différents scénarios en raison de la mobilité des nœuds. Si un nœud se déplace d'un cluster vers un

autre cluster ou si un chef de cluster est sur le point de mourir, nous devons modifier la topologie du réseau et la ré-clustering est nécessaire.

2.Phase de Récupération d'énergie

Dans la phase de la récupération d'énergie, nous avons utilisé la formule suivante.[37] L protocole choisir une variable aléatoire depuis la formule P(s), cette valeur considérée comme une valeur de chargement des nœuds dans le temps t.

$$P(s) = |0.9 * R(t) * \cos(\frac{t}{7 * \pi}) * \cos(\frac{t}{10 * \pi})| \quad (3.5)$$

Où :

- P(S) :producteur source /temps, borne a une valeur maximale[0,0.9].
- R(t) :les valeurs aléatoires entre 0 et 1(random).

l'algorithme complet de notre protocole est composé de 4 classe deux classes sont secondaire(Cluster,Nœud) et les deux autre sont principale(Routage,main).

- **Classe Node** : cette classe permet de configurer les nœuds capteur avec plusieurs paramétrés l'énergie initial,les coordonnées,la porté de transmission,le type..etc) .
- **Classe Cluster** :cette classe permet de stocker et de configurer les nœud choisis comme cluster head.
- **Classe main** c'est la classe principale qui permet de faire fonctionner le capteur avec deux tâches le routage et récupération d'énergie.
- **Classe Routage** cette classe contient plusieurs algorithme pour mettre en place notre protocole.

Dans ce code algorithmique, nous déclarons seulement les deux classes principales.

```
=====
.
                                La classe main()
=====
```

1 :Déclaration des paramétrés

2 :xm=150;

3 :ym=150;

4 :sink_x=0;

5 :sink_y=0;

```

6 :Elec=5*10**(-8);
7 :Efs=10**(-11);
8 :Emp=1.3*10**(-15);
9 :ED=5*10**(-9);
10 :packetLength=4000;
11 :ctrPacketLength=200;
12 :d0=0;
13 :nbr_node=100;
14 :rmax=160;
15 :noud_vive=list();
16 :Econsome_tour=0;
17 :num_cluster=0;
18 :Econsome_Total=list();
19 :length=list();
20 :List_desMax=list();
21 :noud_dead=list();
22 :R=50;
23 :position=0;
24 :distanceBroad=sqrt(xm*xm+ym*ym);
25 :d0=sqrt(Efs/Emp);
26 :E0=0.1;
27 :x_histo=list();
28 :y_histo=list();
29 :Déclaration d'objets
30 :c=Cluster();
31 :node=Node();
32 :routing=Routage(node,c,xm,ym,sink_x,sink_y,Elec,Efs,Emp,ED,packetLength,
33 :                                     ctrPacketLength,nbr_node);

```

Thread_A : pour la phase le routage

```

34 :class Thread_A(threading.Thread) :

```

```

35 : Fonction Constructeur() :
36 :   threading.Thread._init_();
37 : Fonction run() :
38 :   Écrire("Routage Démarrer");
39 :   global node;
40 :   Routing=Routage(node,c);
41 :   return;

```

Thread_B pour la phase de l'énergie renouvelable

```

42 : class Thread_B(threading.Thread) :
43 :   Fonction Constructeur() :
44 :     threading.Thread._init_();
45 :     arret=False;
46 :     Eh=list();
47 :     X=list();
48 :     Xmax=500;
49 :     Ymax=0.00075;
50 :   Pour t de (1, Xmax, 0.1) :
51 :     a = abs(Ymax * random.random() * cos(t/(7 * pi)) * cos(t/(10 * pi)))
52 :     Eh.ajouter(a);
53 :     X.ajouter(t);
54 :     cont=length(Eh);
55 :   Fonction run() :
56 :     global node;
57 :     j = 0;
58 :     try :
59 :       Tan-que(Vrai) :
60 :         Si(arret) :
61 :           raise ValueError("Stop");
62 :         Si_non :
63 :           Pour(i de 0, à nbr_node)

```

```

64 :           ValEH =random.randrange(0,cont);
65 :           E=node.NodeList[i].Ecr + Eh[ValEH];
67 :           Si(E ≥ E0) :
68 :               node.NodeList[i].Er = node.NodeList[i].E0;
69 :           Si_non :
70 :               node.NodeList[i].Er = E;
71 :           except :
72 :               Ecrire(" ",sys.exc_info()[1]);
73 :   Fonction stopthread() :
74 :       arret = True;
75 :       return;
76 : :x=Thread_A();
77 : :y=Thread_B();
78 : :x.start();
79 : :y.start();
80 : :x.join();
81 : :y.stopthread();

```

```

=====
.                               La classe Routage()
=====

```

82 : Déploiements des capteurs

```

83 : fonction route() :
84 :   Pour(i de 1 à rmax) :
85 :     Pour(i de 0 à nbr_node) :
86 :       node.NodeList[i].x_node=random.random()*xm;
87 :       node.NodeList[i].y_node=random.random()*ym;
89 :       node.NodeList[i].x_new.ajouter(node.NodeList[i].x_node);
90 :       node.NodeList[i].y_new.ajouter(node.NodeList[i].y_node);
91 : Calcul de mobilité
92 :   node.NodeList[i].ML=sqrt((node.NodeList[i].x_histo[r-1]-node.NodeList[i].x_histo[r])**2)+

```

```

93 :                               ((node.NodeList[i].y_histo[r-1]-node.NodeList[i].y_histo[r])**2));
94 : Mobilité de la SB
95 :   d_max_SB=Réel(random.random()*xm);
96 :   sink_x=random.randint(0,d_max_SB);
97 :   sink_y=sink_x;
98 : Calcul de distances Nœuds to SB
99 :   pour(i de 0 à nbr_nœuds) :
100 :     node.NodeList[i].d_to_SB=sqrt((node.NodeList[i].x_node-sink_x)**2)+
101 :                                   (node.NodeList[i].y_node-sink_y)**2));
102 : Calcul d'énergies résiduelle
103 :   Pour(i de 0 à nbr_node) :
104 :     node.NodeList[i].Er=node.NodeList[i].Ecr/node.NodeList[i].E0;
105 : Calcule les nœuds survivants
106 :   Pour(i de 0 à nbr_node) :
107 :     Si(node.NodeList[i].Er ≥ 0.4) :
108 :       node.NodeList[i].type = 'N';
109 :       node.NodeList[i].CH = 0;
110 :       noud_vive[r-1] = noud_vive[r-1]+1;
111 :       Si(noud_vive[r-1]==0) :
112 :         break;
113 : Calcul de distances et découvre de voisins
114 :   Pour(i de 0 à nbr_node) :
115 :     Pour(j de 0 à nbr_node) :
116 :       d_i_j=sqrt((node.NodeList[i].x_node-node.NodeList[j].x_node)**2)+
117 :                 ((node.NodeList[i].y_node-node.NodeList[j].y_node)**2));
118 :       Si(d_i_j ≤ R) :
119 :         node.NodeList[i].List_Voisins.ajouter(node.NodeList[j].id);
120 :         node.NodeList[i].nbr_Voisins=length(node.NodeList[i].List_Voisins);
121 : Calcule des pondérations
122 :   Pour(i de 0 à nbr_node) :
123 :     node.NodeList[i].poid=((node.NodeList[i].Er*0.2)+(node.NodeList[i].nbr_Voisins*0.2))/
124 :                           ((node.NodeList[i].ML*0.4)+(node.NodeList[i].d_to_SB*0.2));

```

125 : Trouver le poid maximal parmi ces voisins

126 : **Pour**(*i* de 0 à *nbr_node*) :

127 : **Pour**(*j* de 0 à *nbr_node*) :

128 : **Si**(*j* dans *node.NodeList[i].List_Voisins*) :

129 : *node.NodeList[i].List_Poid_Voisins.ajouter(node.NodeList[j].poid)* ;

130 : **Si**(*node.NodeList[i].poid* non dans *List_desMax*) :

131 : *List_desMax.ajouter(max(node.NodeList[i].List_Poid_Voisins))* ;

.

Election Cluster Head

132 : *num_cluster=-1* ;

133 : **Pour**(*i* de 0 à *nbr_node*) :

134 : **Si**(*node.NodeList[i].poid* dans (*List_desMax*) et (*node.NodeList[i].Er* ≥ 0.4) :

135 : *node.NodeList[i].IMCH=1* ;

136 : *node.NodeList[i].type='CH'* ;

137 : *num_cluster=num_cluster+1* ;

138 : *c.CHList[num_cluster].x_ch = node.NodeList[i].x_node* ;

139 : *c.CHList[num_cluster].y_ch = node.NodeList[i].y_node* ;

140 : *c.CHList[num_cluster].d_to_SB = node.NodeList[i].d_to_SB* ;

141 : *c.CHList[num_cluster].id = node.NodeList[i].id* ;

142 : *c.CHList[num_cluster].Er = node.NodeList[i].Er* ;

143 : **Pour**(*k* de 0 à *nbr_node*) :

144 : *d_n_ch=sqrt((node.NodeList[i].x_node-node.NodeList[k].x_node)**2)+*

145 : *(node.NodeList[i].y_node-node.NodeList[k].y_node)**2))* ;

146 : **Si**(*d_n_ch* $\leq R$) :

147 : **Si**(*node.NodeList[k].IMCH == 0*) :

148 : *node.NodeList[k].List_CHs.ajouter(node.NodeList[i].id)* ;

:

Phase de communication et d'agrégation des données CH

149 : **Si**(*distanceBroad* $\geq d0$) :

150 : *node.NodeList[i].Ecr=node.NodeList[i].Ecr-(Elec*ctrPacketLength+Emp*ctrPacketLength**

```

151 :                                                                 distanceBroad**2);
152 :   Econsome_tour=Econsome_tour+Elec*ctrPacketLength+Emp*ctrPacketLength*
153 :                                                                 distanceBroad**4;
156 : Si_non :
157 :   node.NodeList[i].Ecr=node.NodeList[i].Ecr-(Elec*ctrPacketLength+Efs*
158 :                                                                 ctrPacketLength*distanceBroad**2);
159 :   Econsome_tour=Econsome_tour+Elec*ctrPacketLength+Efs*ctrPacketLength*
160 :                                                                 distanceBroad**2);
161 : Si(node.NodeList[i].d_to_SB ≥ d0) :
162 :   node.NodeList[i].Ecr=node.NodeList[i].Ecr-((Elec+ED)*packetLength+Emp*
163 :                                                                 packetLength*node.NodeList[i].d_to_SB**4);
164 :   Econsome_tour=Econsome_tour+(Elec+ED)*packetLength+Emp*packetLength*
165 :                                                                 node.NodeList[i].d_to_SB**4;
166 : Si_non :
167 :   node.NodeList[i].Ecr=node.NodeList[i].Ecr-((Elec+ED)*packetLength+Efs*
168 :                                                                 packetLength*node.NodeList[i].d_to_SB**2);
169 :   Econsome_tour=Econsome_tour+(Elec+ED)*packetLength+Efs*packetLength*
170 :                                                                 node.NodeList[i].d_to_SB**2;

```

Formation de clusters

```

171 : Pour(i de 0 à nbr_node) :
172 :   Si((node.NodeList[i].type == 'N') et (node.NodeList[i].Er ≥ 0.4)) :
173 :     Si(num_cluster ≥ 0) :
174 :       length = list();
175 :       Pour(k de 0 à num_cluster) :
176 :         length[k]=sqrt((node.NodeList[i].x_node-c.CHList[k].x_ch**2)+
177 :                       (node.NodeList[i].y_node-c.CHList[k].y_ch**2));
178 :       min_dis_CH_N=min(length);
179 :       min_dis_num_cluster=length.index(min(length));
180 :       node.NodeList[i].Ecr=node.NodeList[i].Ecr-Elec*ctrPacketLength
181 :       Econsome_tour= Econsome_tour+Elec*ctrPacketLength;

```

```

182 :   Si( $min\_dis\_CH\_N < d0$ ) :
183 :      $node.NodeList[i].Ecr = node.NodeList[i].Ecr - (Elec * (ctrPacketLength + packetLength) + Efs * (ctrPacketLength + packetLength) * min\_dis\_CH\_N^{**2})$ ;
184 :
185 :      $Econsome\_tour = Econsome\_tour + Elec * (ctrPacketLength + packetLength) + Efs * (ctrPacketLength + packetLength) * min\_dis\_CH\_N^{**2}$ ;
186 :
187 :   Si\_non :
189 :      $node.NodeList[i].Ecr = node.NodeList[i].Ecr - (Elec * (ctrPacketLength + packetLength) + Emp * (ctrPacketLength + packetLength) * min\_dis\_CH\_N^{**4})$ ;
190 :
191 :      $Econsome\_tour = Econsome\_tour + Elec * (ctrPacketLength + packetLength) + Emp * (ctrPacketLength + packetLength) * min\_dis\_CH\_N^{**4}$ ;
192 :
193 :      $node.NodeList[i].CH = c.CHList[min\_dis\_num\_cluster].id$ ;
194 :      $node.NodeList[i].ListCHs = c.CHList[min\_dis\_num\_cluster].id$ 
195 :   Si( $min\_dis\_CH\_N > 0$ ) :
196 :      $node.NodeList[c.CHList[min\_dis\_num\_cluster].id].Ecr =$ 
197 :      $node.NodeList[c.CHList[min\_dis\_num\_cluster].id].Ecr - (Elec + ED) * packetLength$ ;
198 :      $node.NodeList[c.CHList[min\_dis\_num\_cluster].id].Ecr =$ 
199 :      $node.NodeList[c.CHList[min\_dis\_num\_cluster].id].Ecr - (Elec * ctrPacketLength)$ ;
200 :      $Econsome\_tour = Econsome\_tour + (Elec + ED) * packetLength + Elec * ctrPacketLength$ ;
201 :   Si( $min\_dis\_CH\_N > d0$ ) :
203 :      $node.NodeList[c.CHList[min\_dis\_num\_cluster].id].Ecr =$ 
204 :      $node.NodeList[c.CHList[min\_dis\_num\_cluster].id].Ecr - (Elec * ctrPacketLength + Emp * ctrPacketLength * min\_dis\_CH\_N^{**4})$ ;
205 :
206 :      $Econsome\_tour = Econsome\_tour + Elec * ctrPacketLength + Emp * ctrPacketLength * min\_dis\_CH\_N^{**4}$ ;
207 :
208 :   Si\_non :
209 :      $node.NodeList[c.CHList[min\_dis\_num\_cluster].id].Ecr =$ 
210 :      $node.NodeList[c.CHList[min\_dis\_num\_cluster].id].Ecr - (Elec * ctrPacketLength + Efs * ctrPacketLength * min\_dis\_CH\_N^{**2})$ ;
211 :
212 :      $Econsome\_tour = Econsome\_tour + Elec * ctrPacketLength + Efs * ctrPacketLength * min\_dis\_CH\_N^{**2}$ ;
213 :
214 :   Si\_non :
115 :     Si( $node.NodeList[i].d\_to\_SB \leq d0$ ) :

```

```

216 :      node.NodeList[i].Ecr=node.NodeList[i].Ecr-(Elec*packetLength+Efs*packetLength*
217 :                                          node.NodeList[i].d_to_SB**2);
218 :      Econsome_tour=Econsome_tour+Elec*packetLength+Efs*packetLength*
219 :                                          (node.NodeList[i].d_to_SB**2);
220 :      Si_non :
221 :      node.NodeList[i].Ecr=node.NodeList[i].Ecr-(Elec*packetLength+Emp*packetLength*
222 :                                          node.NodeList[i].d_to_SB**4);
223 :      Econsome_tour=Econsome_tour+Elec*packetLength+Emp*packetLength*
224 :                                          node.NodeList[i].d_to_SB**4;

```

Ré-Clustering

225 : **cas 1 :Mort de capteur**

```

226 :      Pour i de 0 à nbr_node) :
227 :      Si node.NodeList[i].Er ≤ 0.39 :
228 :      Si(node.NodeList[i].type=='N') :
229 :      Pour(j de 0 à nbr_node) :
230 :      Si(node.NodeList[i].id dans node.NodeList[j].List_Voisins :
231 :      node.NodeList[j].List_Voisins.supprimer(self.node.NodeList[i].id);
232 :      node.NodeList[j].List_Poid_Voisins.supprimer(node.NodeList[i].poid);
233 :      Si_non_Si(node.NodeList[i].type=='CH') :
234 :      maxVoisins=max(node.NodeList[i].List_Poid_Voisins);
235 :      position=node.NodeList[i].List_Poid_Voisins.index(maxVoisins);
236 :      Si(node.NodeList[i].List_Voisins[self.position]ge0.4) :
237 :      Pour(l de 0 à nbr_node) :
238 :      Si(node.NodeList[l].CH==node.NodeList[i].id) :
239 :      node.NodeList[l].CH=node.NodeList[i].List_Voisins[position];

```

240 : **cas 2 :déplacement de capteur**

```

241 :      Pour(i de 0 à nbr_node) :
242 :      Si((node.NodeList[i].x_node !=node.NodeList[i].x_new) Ou
243 :      (node.NodeList[i].y_node !=node.NodeList[i].y_new)) :
244 :      Si(node.NodeList[i].type=='N') :

```

```

245 :   Pour(j de 0 à nbr_node) :
246 :       Si(node.NodeList[i].id dans node.NodeList[j].List_Voisins) :
1247 :           node.NodeList[j].List_Voisins.remove(node.NodeList[i].id);
248 :           node.NodeList[j].List_Poid_Voisins.supprimer(node.NodeList[i].poid);
249 :       Si_non_Si(node.NodeList[i].type=='CH') :
250 :           maxVoisins=max(node.NodeList[i].List_Poid_Voisins);
251 :           position=node.NodeList[i].List_Poid_Voisins.index(maxVoisins);
252 :       Si(node.NodeList[i].List_Voisins[position] ≥ 0.4) :
253 :           Pour(l de 0 à nbr_node) :
254 :               Si(node.NodeList[l].CH==node.NodeList[i].id) :
255 :                   node.NodeList[l].CH=node.NodeList[i].List_Voisins[position];
256 : Ré-initialisation des variables
257 :   Pour(i de 0 à nbr_node) :
258 :       node.NodeList[i].nbr_Voisins=0;
259 :       node.NodeList[i].List_Voisins=list();
260 :       node.NodeList[i].List_Poid_Voisins=list();
261 :       List_desMax.clear();

```

Affichage

```

262 :   Fonction Affichage()
263 : Fin

```

3.2.4 Conclusion

L'énergie renouvelable et la mobilité de la station de base. Réponds aux exigences de haute performance à long terme des réseaux de capteurs sans fil. Pour cette raison, nous avons adopté dans ce chapitre le protocole de routage amélioré pour former un nouveau protocole de routage (EHMS-EEMCS,) dans le but de maximiser la durée de vie du réseau de capteurs tout en tenant la mobilité des capteurs.

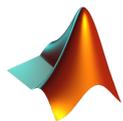
Évaluation de performance

4.1 Introduction

Avant d'avoir une expérience du monde réel, nous devons souvent l'expérimenter de manière virtuelle utilisant des simulateurs. La simulation est une technique de modélisation du monde réel. Elle consiste à modéliser un système en représentant toutes ses entités, leurs comportements et leurs interactions. Ces simulations peuvent être simples, certaines sont relativement complexes et certaines sont tout à fait accessibles. Certains dits "open source" comme NS2 et NS3 (émulateur de réseau), d'autres sont commerciaux comme OPNET (optimisation de réseau). Dans ce chapitre nous présentons l'évaluation des performances de protocole proposé. Nous allons commencer par un aperçu sur les simulateurs des réseaux de capteurs. Ensuite, nous détaillerons l'environnement de simulation de notre protocole. Enfin, nous présenterons les différents résultats fournis par le protocole.

4.2 Les simulateurs des réseaux de capteurs

4.2.1 Matlab :



Matlab est un langage matriciel qui combine les mathématiques, la modélisation graphique et la programmation. MATLAB dispose d'une grande bibliothèque de fonctions intégrées et pré-écrites avec des notations simples et puissantes pour de nombreuses tâches de calcul courantes. Il est optimisé pour résoudre des problèmes scientifiques et techniques.[33]

4.2.2 NS3 (Network Simulator version3) :



NS-3 est un simulateur de réseau gratuit qui comprend un simulateur de réseau à événements discrets. Son noyau et ses modèles sont écrits en C++, mais il dispose d'une interface de script Python. L'objectif de NS-3 est de créer une plateforme de simulation ouverte utile pour la recherche sur les réseaux. Les outils nam et ns3-viz du simulateur NS permettent de visualiser les animations générées par le simulateur. telle que le transfert et la taille des paquets échangés, l'état des files d'attente, etc..[34]

4.2.3 OMNeT++ :



OMnet ++ IDE (Integrated Development Environment) est un environnement open source basé sur la plateforme Eclipse qui comprend des outils pour la création de configuration des modèle de réseau (fichiers NED et INI), ainsi que des outils pour exécuter un lot de programmes et analyser les résultats de la simulation.[35]

4.3 Environnement de simulation

Après avoir étudié les simulateurs existants, nous avons remarqué que leur environnement est très complexe. Dans le même temps, il n'est pas permis de modéliser la charge des batteries de nœuds par le soleil. Pour ces raisons, nous avons choisi d'utiliser le logiciel de programmation "python" pour écrire notre propre simulateur qui correspond à nos besoins.

4.3.1 Python :



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 tr'es simple. 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 . . . [36]

4.3.2 Paramétrés de simulation

Les nœuds capteurs sont déployés dans une zone de $(150*150)$ mètres carrés composée de 100 nœuds capteurs dispersés aléatoirement avec l'option de mobilité. Ces capteurs sont homogène, ont la même énergie initiale $1j$, la même capacité de calcul et de stockage, même port de transmission $50m$ et La puissance de la station de base est illimitée et mobile et le nombre de tour est de 160.les paramètres utilisés sont résume dans tableau 4.1 :

Paramétrés	Valeurs
Nombre de stations de base	1
Nombre des nœuds	100
Surface de simulation	$(200*200)m^2$
Localisation de la Station de Base	mobile
Portée de transmission des capteurs	$50m$
Nombre de tour	160

TABLE 4.1: Paramétrés de simulation.

4.3.3 Description du système

Notre système traité sur une zone de surface $(150*150)m^2$, contient une station de base SB et 100 nœuds capteurs, les entités de ce système sont :

- **La station de base(SB)** : elle recevoir les données du réseau.
- **Les Cluster Heads (CH)** : sont des nœuds capteurs particuliers choisis selon des techniques spéciales, Ils reçoivent, agrègent les donnes qui transmettent par les nœuds et ils aident au routage des paquets dans le réseau.
- **Nœuds capteurs** : utilisés pour détecter divers événements dans la zone de déploiement et envoie ses informations à son CH.

4.3.4 Variables suggestives du système

Définition de la variable	la variable	Valeurs
Énergie initiale	E_0	1 Joule
La seuil d'énergie	E_{seuil}	0.4 Joule
Distance entre deux capteurs	d_{ij}	$\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$
Énergie consommée lors de l'émission d'un paquet	E_{elec}	$5 * 10^{-8}$ Joule
Énergie résiduel après la transmission	E_r	$\frac{E_{cr}}{E_0}$
Énergie consommé par les nœuds pour chaque transféré	$E_{consome_{tour}}$	$E_{elec} * d * packetLength$
Énergie consommée lors de l'agrégation des données	ED	$5 * 10^{-9}$ Joule
Énergie consommée lors de la réception d'un paquet	E_{lec}	$5 * 10^{-8}$ Joule
Position d'un capteur	(x_{noud}, y_{noud})	en mètre
L'évanouissement par trajets multiples	E_{mp}	$1.3 * 10^{-15}$
Pour l'espace libres	E_{fs}	$1 * 10^{-11}$ en mètre
La valeur seuil de distance	d_0	$\sqrt{\frac{E_{fs}}{E_{mp}}}$
Le nombre de tour maximal	R_{max}	entier
La porté de transmission de capteur	R	50 mètre
La source énergie	$P(s)$	$ 0.9 * R(t) * \cos(\frac{t}{7*\pi}) * \cos(\frac{t}{10*\pi}) $
le valeur aléatoire	R(t)	[0,1]

TABLE 4.2: Variables suggestives du système

4.4 Évaluation des performances

Pour évaluer notre protocole nous prendrons deux paramétrés d'évaluation :

- L'énergie totale consommée dans chaque tour.
- Le nombre de nœud capteur survivant dans le réseau.

Le protocole a vu deux améliorations de la mobilité de la station de base et l'énergie renouvelable. Les résultats du simulation représente ci-dessous :

4.4.1 Distribution des nœuds

La figure ci-dessous représente le déploiement des nœuds aléatoirement dans la zone de captage avec la possibilité de déplacer dans n'importe quelle position. La station de base représentée avec un rectangle elle peut déplacer linéairement selon le mouvement créé par l'équation($Y=x$).

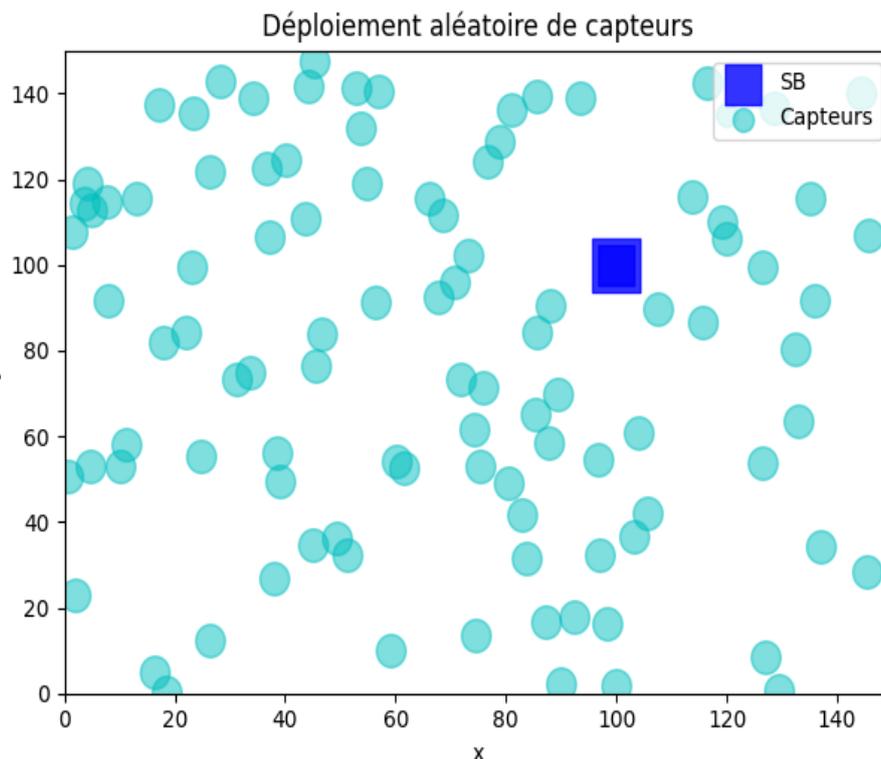


FIGURE 4.1: Déploiement des nœuds capteurs

4.4.2 Formation des clusters :

Les figures suivantes représentent les clusters qui sont formés après l'exécution de phase de formation de cluster de notre protocole nous avons pris deux scénarios. Nous pouvons observer le déplacement de nœuds numéro 2 d'un cluster un autre cluster et en plus le mouvement de la station de base.

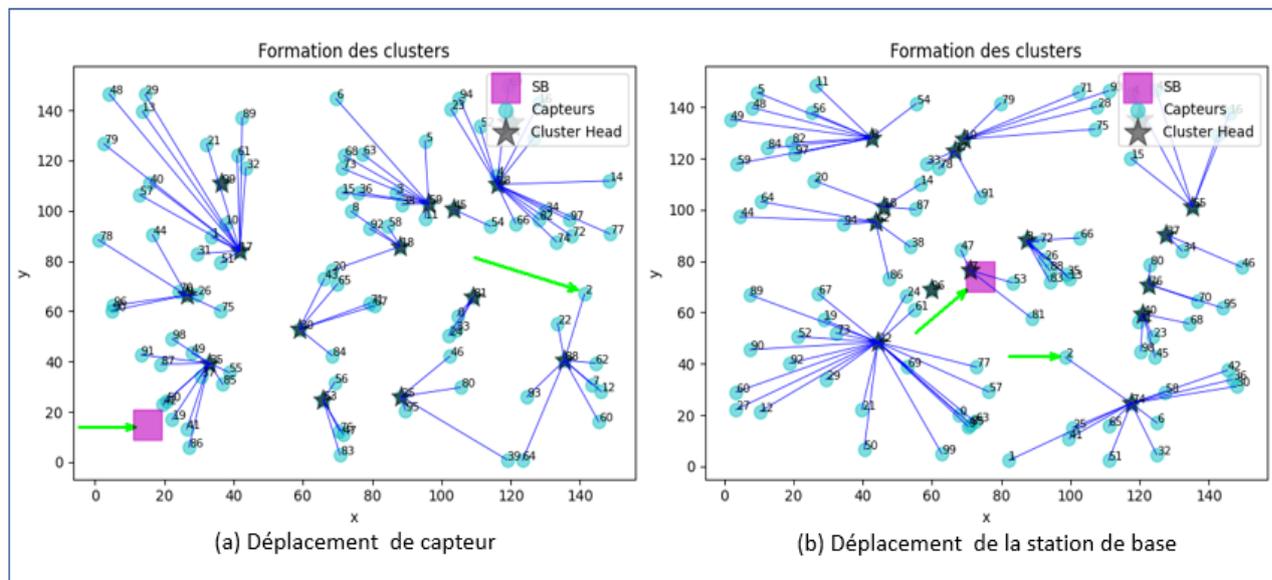


FIGURE 4.2: Formation des clusters

4.4.3 Modèle de source d'énergie

L'énergie environnementale (comme l'énergie solaire) peut être collectée et convertie en énergie électrique. Le modèle suivant est utilisé pour simuler le comportement de la source solaire. Comme le montre la figure 4.3, la courbe de puissance obtenue $P_s(t)$ simulant le jour et la nuit et similaire à celle des cellules solaires.

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

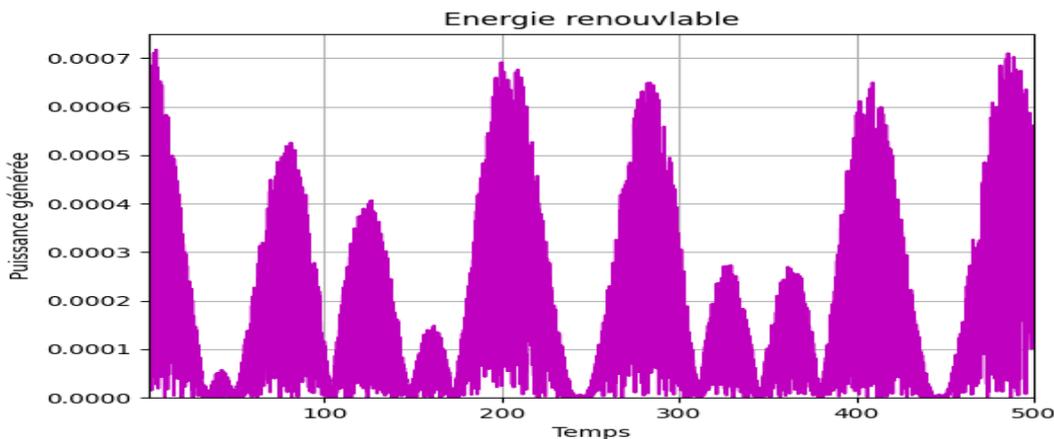


FIGURE 4.3: La courbe de la puissance $P_s(t)$

Où :

- $R(t)$:représente une variable aléatoire uniformément répartie entre 0 et 1,la valeur.

4.4.4 Première amélioration :station de base mobile

-L'énergie total consommée

Nous comparons le protocole EEMCS(SB fixe)avec notre protocole MS-EEMCS(SB mobile). Nous pouvons observer dans la figure 4.4 que l'énergie total consommée dans le protocole EEMCS supérieure à ce qui a été consommé par notre protocole. Ainsi nous pouvons dire que notre protocole est performant et efficace comparativement au protocole EEMCS de point de vue de la consommation d'énergie.

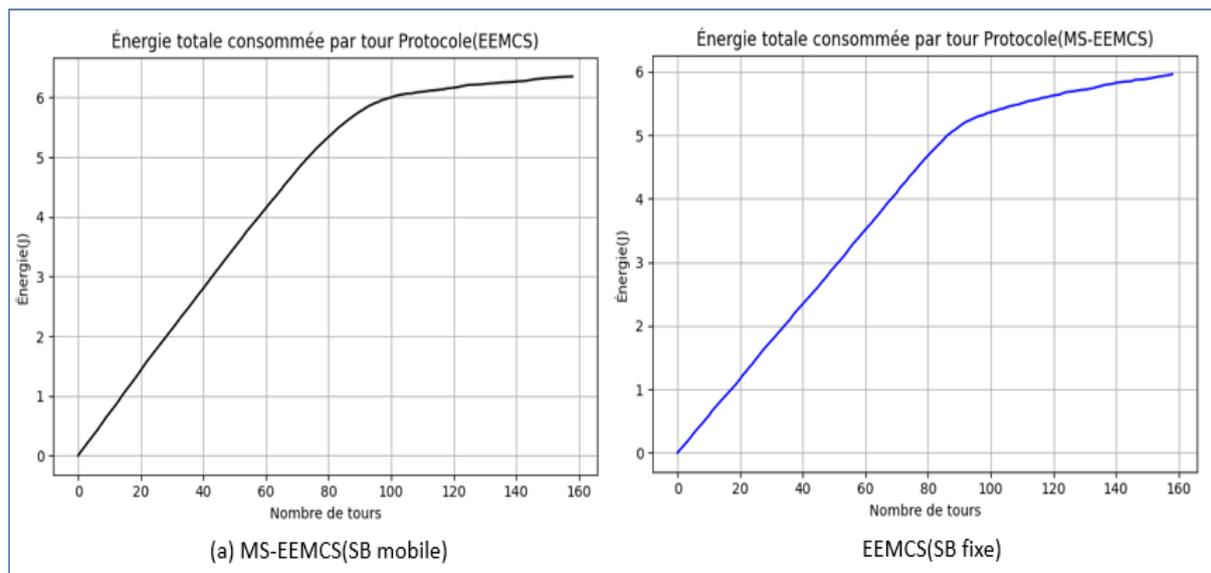


FIGURE 4.4: Énergie total consommée par tour

-Nombre de nœuds survivant

Dans la figure 4.5 nous pouvons observer que le nombre de tours dans le protocole(EEMCS)inférieur à ce qui a été obtenu dans notre protocole MS-EEMCS.Nous pouvons dire que notre protocole est performant et efficace comparative-ment au protocole EEMCS de point de vue de la durée de vie des nœuds capteurs.

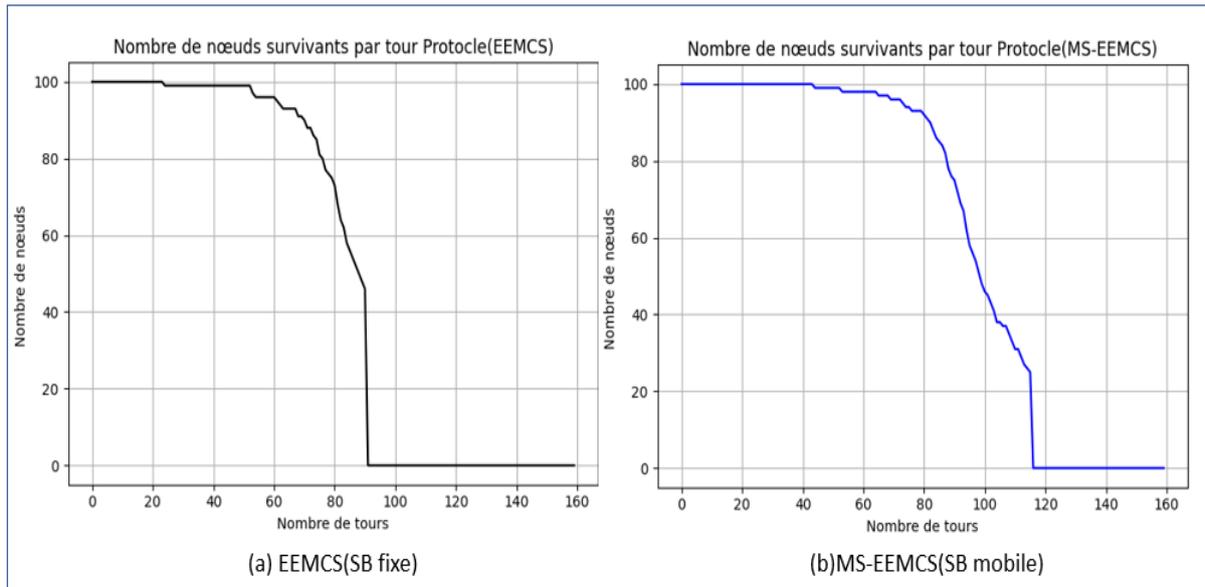


FIGURE 4.5: Nombre de nœuds capteur survivant par tour

4.4.5 Deuxième amélioration : l'énergie renouvelable et station de base mobile

-Énergie total consommée

Si l'on compare notre protocole EHMS-EEMCS avec les résultats obtenus précédemment nous pouvons observer dans la figure 4.6 que l'énergie totale dépensée diminuée considérablement et cela indique l'efficacité de notre amélioration.

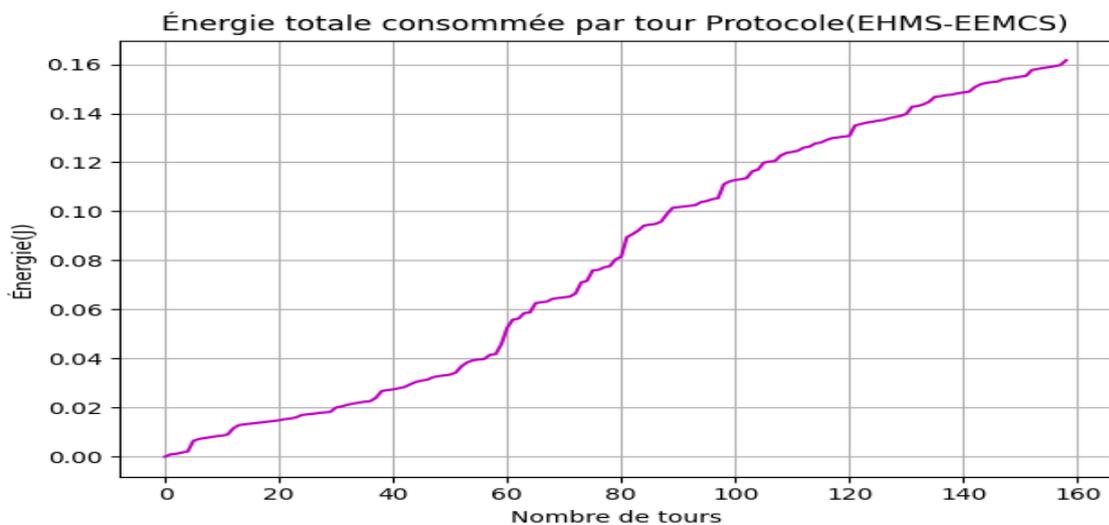


FIGURE 4.6: Énergie total consommée par tour

-Nombre de nœuds survivant

La figure 4.7 montre les résultats obtenus du point de vue du nombre de nœuds survivant dans le réseau. Au fur et à mesure que le nombre de tours augmente, le nombre de nœuds survivant diminue. La comparaison de notre protocole EHMS-EEMCS avec le protocole EEMCS montre que le nombre de nœuds survivant dans notre protocole à chaque tour est plus important par rapport au protocole (EEMCS). Ainsi nous pouvons dire que notre protocole est performant et efficace comparativement au protocole EEMCS de point de vue de la durée de vie des nœuds capteurs.

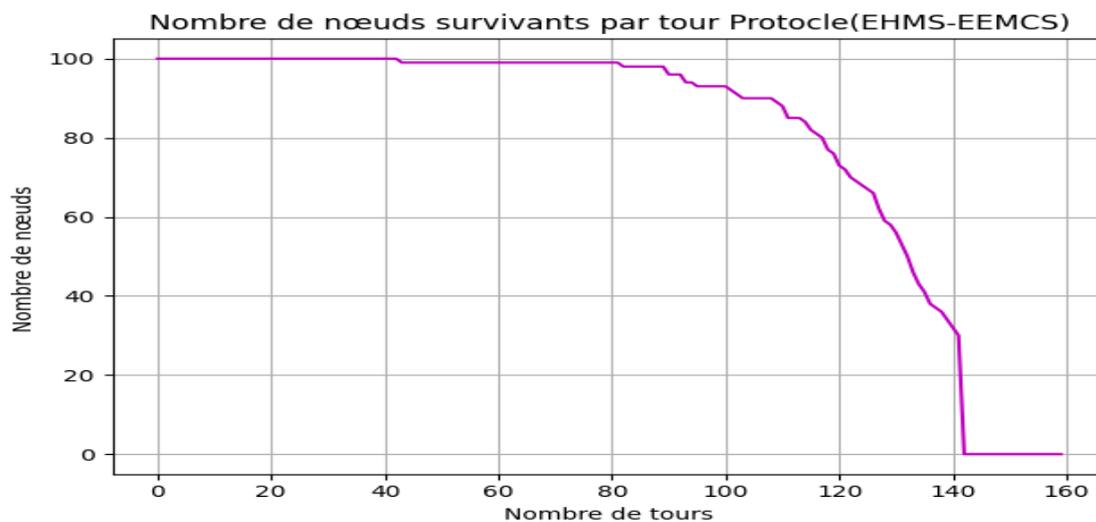


FIGURE 4.7: Énergie total consommée par tour

Bilan sur le processus d'amélioration de protocole

Le bilan final des résultats de processus d'amélioration est résumé dans le tableau 4.8.
 -Le protocole EEMCS consommée environ 6,50 joule d'énergie et le nombre de tours atteint 80.

-Noter que l'énergie consommée après la première amélioration atteint environ 5,98 joule et le nombre de tours atteint 80.

-Après la deuxième amélioration sur le protocole, l'énergie consommée réduit considérablement environ 0,16 joule et le nombre de tours augmenté jusqu'à 142, Ce qui prouve l'efficacité de notre amélioration.

Tours	Energie total consommé (joule)			Nombre de nœud capteur survivants		
	EEMCS	MS-EEMCS	EHMS-EEMCS	EEMCS	MS-EEMCS	EHMS-EEMCS
20	1,35	1,09	0,015	100	100	100
40	2,73	2,28	0,025	99	100	100
60	4,08	3,45	0,05	96	99	99
80	5,28	4,62	0,08	75	95	98
100	5,98	5,34	0,11	0	0	96
120	6,15	5,61	0,13	0	0	76
140	6,26	3,80	0,15	0	0	30
160	6,5	5,98	0,16	0	0	0
E max consommé	6,50	5,98	0,16	-	-	-
E min consommé	1,35	1,9	0,015	-	-	-
E avg consommé	4,55	4,03	0,09	-	-	-
Nombre de tours	-	-	-	80	80	140

FIGURE 4.8: Résultats d'amélioration

4.5 Conclusion

Dans ce chapitre, nous avons présenté des simulateurs de réseaux de capteurs. Ensuite, nous avons fourni l'environnement de simulation pour notre protocole EHMS-EEMCS. Ensuite, nous avons évalué les performances du protocole proposé (EHMS-EEMCS). Les résultats de simulation obtenus montrent que notre protocole permet de prolonger la durée de vie de réseau par rapport au protocole EEMCS qui ne prend pas en compte l'énergie renouvelable et la mobilité de la station de base.

Conclusion générale

Les réseaux de capteurs sans fil constituent un domaine de recherche très riche, et ils peuvent être appliqués dans nombreux domaines différents. Mais l'utilisation de batteries par les capteurs a été une limitation critique des réseaux de capteurs sans fil. À cette fin, de nombreuses technologies et protocoles de routage économes en énergie ont été développés pour répondre à ces exigences. Mais pour assurer la longévité du réseau, l'architecture logicielle est insuffisante car les nœuds capteurs sont dispersés dans des environnements inaccessibles ou changer les batteries est impossible, il faut aussi améliorer l'architecture matérielle et la rendre compatible avec les changements environnementaux, par exemple en ajoutant des technologies de pointe capable de récupérer l'énergie ambiante.

Ce travail se concentre sur l'adaptation de certains des meilleurs protocoles de routage hiérarchique économe en énergie pour les réseaux de capteurs sans fil mobile, avec une technologie de récupération d'énergie pour éliminer le problème d'épuisement de la batterie des capteurs.

Notre objectif principal dans cette mémoire est de proposer un nouveau protocole de routage hiérarchique **EHMS-EEMCS**, qui prend en compte la mobilité des capteurs et l'énergie renouvelable avec une station de base mobile et une sélection efficace de têtes de cluster pour l'amélioration à vie des réseaux de capteurs sans fil.

L'évaluation des performances de notre protocole **EHMS-EEMCS** a été simulée en Python, et nous avons trouvé que notre proposition était meilleure que le protocole EEMCS qui ne prend pas en compte l'énergie renouvelable et la mobilité de la station de base. En fin de compte, comme possibilité prévisible, nous prévoyons d'améliorer notre protocole

et de le comparer avec d'autres protocoles qui prennent les contraintes d'énergie renouvelable et la mobilité de capteurs. Il serait également préférable d'implémenter notre solution dans des simulateurs de réseaux de capteurs sans fil dédié (par exemple OPNET, NS2, MATLAB...).

Bibliographie

- [1] C. Bernard. "Réseaux de capteurs présentation". Université de Renne, Novembre 2011.
- [2] SARI Mounya Amal, "Etude de RSSI pour l'estimation de la distance dans les réseaux de capteurs sans fil", Mémoire Master, Université Abou Bakr Belkaid-Tlemcen, 12/6/2017
- [3] ROUAM Rania, MADJDOB Ismail, "Optimisation de déploiement et de localisation de cible dans un réseau de capteur sans fil", Mémoire Master, Université AMOU-Bouira, 2016
- [4] W. Su Y. Sankarasubramaniam E. Cayirci Akyildiz, I.F. "A Survey on Sensor Networks". IEEE Communications Magazine, 2004.
- [5] A. de Castro I. Teixeira, J. F. de Rezende and A. C. P. "Pedroza. Wireless Sensor Network : Improving the Network Energy Consumption". in XXI Symposium Brazilian Telecommunications. SBT'04, Belem, Brazil, September 2004.
- [6] Krogh, Einar. "An Introduction to the Internet of Things", (2020).
- [7] Anisi, Mohammad Hossein, et al. "Energy harvesting and battery power based routing in wireless sensor networks." Wireless Networks 23.1 (2017) : 249-266.
- [8] MESSAI Sarra, "Gestion de la Mobilité dans les Réseaux de Capteurs Sans Fil", Thèse

de Doctorat, Université Ferhat Abbas Sétif 1 et l'Université Claude Bernard Lyon 1, 24/11/2019

[9] Amina HENTATI. "Sélection des nœuds dans les réseaux de capteurs sans fil avec récolte d'énergie". Mémoire de maîtrise science appliquées, département de génie électrique, école polytechnique de Montréal, Juin 2006.

[10] D. Rey. Collecte des données d'un réseau de capteurs sans fil en utilisant en surcouche réseau pair à pair. Mémoire de fin de cycle, département de génie informatique et génie logiciel école polytechnique de Montréal. david rey, Avril 2010.

[11] S. Moad. Optimisation de la consommation d'énergie dans les réseaux de capteurs sans fil. Mémoire master, Université IFSIC-Rennes 1, 2008.

[12] 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..

[13] 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 Bjaia.

[14] Beigh B.M. and Prof.M.A.Peer. "Classification of current routing protocols for ad hoc networks - a review. International Journal of Computer Applications", 7(8) :26-32, October 2010.

[15] P. Corke, T.Wark, R.Jurdak, Wen Hu, P.Valencia, D.Moore. "Environmental wireless sensor networks", Proceedings of the IEEE", Nov. 2010.

[16] S. Sentilles. "Architecture logicielle pour capteurs sans fils en réseau", Rapport de recherche, Université de Pau et des pays de l'Adour, juin 2006.

[17] NAIMI Sabrina, "Gestion de la mobilité dans les réseaux Ad Hocpar anticipation

des métriques de routage ”,Thèse de Doctorat,Université Paris-SUD,22/7/2015

[18] 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.

[19] BOUZEGANE Mhenna,GAOUAOUI Ferhat,” Alimentation d’un moteur à courant continu par une source photovoltaïque”,Mémoire Master,Université mouloud mameri de Tizi-Ouzou,26/9/2018

[20] KECHAR Bouabdellah,”Problématique de la consommation d’énergie dans les réseaux de capteur sans fil” ,Diplôme de doctorat d’état,Université d’Oran,13/7/2010

[21] Sofiane MOAD,”La consommation d’énergie dans les réseaux de capteurs sans fil Étude bibliographique”, Master Recherche 2 en Informatique ,IFSIC-Rennes1.

[22] 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.

[23] Mekki nabil,Mohammedi kada,”Techniques de conservation d’énergie pour les réseaux de capteur sans fil” , MEMOIRE de Master,univ-saida, 2018.

[24] Ziani, Youcef.”Etude comparative de méthodes de routage dans les réseaux de capteurs sans fil pour le domaine résidentiel” .Diss. Université du Québec à Trois-Rivières,2013.

[25] 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.

[26] aribi Bilal Hadjadj Abdelkrim.”Réseaux de capteurs sans fil pour le contrôle des paramètres ambiants dans un bâtiment d’élevage de volaille”. mémoire fin de cycle, Université de Bouira, 2018.

- [27] C.Intanagonwiwat, R.Govindan and D.Etrein."Directed diffusion : a scalable and robust communication paradigm for sensor networks", In Proc. ACM Mobicom, Boston, USA, 2000..
- [28] H.Zahra, S.Nadia. "Clustérisation avec équilibrage de charge et routage à basse consommation d'énergie dans les RCSF". Mémoire de fin de cycle en Informatique, Université A.MIRA de Bejaia, Algerie, 2010.
- [29] S. Umbreen, D. Shehzad, N. Shafi, B. Khan and U. Habib, "An Energy-Efficient Mobility-Based Cluster Head Selection for Lifetime Enhancement of Wireless Sensor Networks," in IEEE Access, vol. 8, pp. 207779-207793, 2020, doi : 10.1109/ACCESS.2020.3038031.
- [30] W. R. Heinzelman, A. Chandrakasan and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," Proceedings of the 33rd Annual Hawaii International Conference on System Sciences, 2000, pp. 10 pp. vol.2-, doi : 10.1109/HICSS.2000.926982.
- [31] D. Kim and Y. Chung, "Self-Organization Routing Protocol Supporting Mobile Nodes for Wireless Sensor Network," First International Multi-Symposiums on Computer and Computational Sciences (IMSCCS'06), 2006, pp. 622-626, doi : 10.1109/IMSCCS.2006.265.
- [32] Wang, S., Yu, J., Atiquzzaman, M. et al. "CRPD : a novel clustering routing protocol for dynamic wireless sensor networks". Pers Ubiquit Comput 22, 545–559 (2018). <https://doi.org/10.1007/s00779-018-1117-602>
- [33] Matlab. <http://fr.mathworks.com/products/matlab/>. Consulté le : 15-08-2021.
- [34] Ns-3. <https://www.nsnam.org/>. Consulté le : 15-08-2021.
- [35] Leila Imane NIAR," Analyse Graphique pour la surveillance dans un réseau de

capteur sans fil(RCSF)”,mémoire Magister,université ORNA,juillet 2012

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

[37] Akli, ABBAS, et al. ”A real-time feedback scheduler for environmental energy with discrete voltage/frequency modes.” *Computer Standards & Interfaces* 44 (2016) : 264-273.

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

[39] LAEHAL Khadidja ,RABHI Rofiada,”Routage dans les réseaux de capteurs sans fil sous contrainte de l’énergie ambiante ”,Mémoire Master,Université AMOU-Bouira,2019-2020

[40] L. Konrad et al,”Sensor networks for emergency response : Challenges and opportunities ”, *IEEE Pervasive Computing*, 2004, Vol. 3(4), pp.16 -23.

[41] DANIELA DRAGOMIRESCU Dragomirescu,” Nouvelle génération de réseaux de capteurs sans fil pour l’aéronautique et l’espace” ,juin 2011

Résumé

Le monde d'aujourd'hui a besoin collecter des informations à partir d'environnements hautement sensibles, hostiles et inaccessibles. Pour cela la technologie de communication sans fil a conduit au développement du réseau de capteurs sans fil.

Un réseau de capteurs sans fil est un réseau composé d'un grand nombre de nœuds de capteurs. qui peuvent être mobile ou statique collaborent entre eux pour fournir un service bien déterminé cependant, ces réseaux soulèvent des problèmes fondamentaux pour la communauté scientifique. Le problème majeur de ce type de réseau est la ressource énergétique limitée donc la conservation d'énergie constitue l'un des axes de recherche les plus importants afin de maximiser la durée de vie du réseau.

Dans ce mémoire, nous avons proposé un nouveau protocole de routage hiérarchique qui vise la minimisation de la consommation d'énergie, dénommé **EHMS-EEMCS** (récupération d'énergie et station de base mobile - Sélection de têtes de groupe basée sur la mobilité et efficace sur le plan énergétique pour améliorer la durée de vie des réseaux de capteurs sans fil.) qui est l'amélioration de protocole EEMCS, ce protocole prend en compte la récupération d'énergie à travers d'une source solaire et aussi la mobilité de station de base et la mobilité des capteurs. Ce protocole utilise un mécanisme de sélection de tête de cluster basé sur la mobilité et économe en énergie pour surmonter ces limitations. La sélection CH est basée sur des paramètres dédiés qui ont un impact énorme sur la consommation d'énergie du capteur. La pondération de chaque nœud est calculée sur la base du niveau de mobilité du nœud, l'énergie résiduelle, distance par rapport au station de base et de la densité de voisins. Les résultats de simulation obtenus sous le simulateur python montrent que les performances sont globalement satisfaisantes.

Mots clés : Réseaux de capteurs sans fil, Protocole de Routage hiérarchique, Consommation d'énergie, EEMCS, EHMS-EEMCS...

Abstract

Today's world needs to collect information from highly sensitive, hostile and inaccessible environments. For this purpose, wireless communication technology has led to the development of the wireless sensor network.

A wireless sensor network is a network composed of a large number of sensor nodes, which can be mobile or static, that collaborate with each other to provide a specific service. However, these networks raise fundamental problems for the scientific community. The major problem of this type of network is the limited energy resource, so energy conservation is one of the most important research axes in order to maximize the network lifetime.

In this thesis, we have proposed a new hierarchical routing protocol that aims at minimizing energy consumption, named EHMS-EEMCS (Energy harvesting Mobile Sink-An Energy-Efficient Mobility-Based Cluster Head Selection for Lifetime Enhancement of Wireless Sensor Networks) which is the improvement of EEMCS (An Energy-Efficient Mobility-Based Cluster Head Selection for Lifetime Enhancement of Wireless Sensor Networks) protocol this protocol take into account the energy recovery through a solar source and also the mobility of base station and sensors. This protocol uses a mobility-based and energy-efficient cluster head selection mechanism to overcome these limitations. The CH selection is based on dedicated parameters that have a huge impact on the sensor energy consumption. The weighting of each node is calculated based on the node's mobility level, residual energy, distance to sink, and neighbor density. The simulation results obtained under the python simulator show that the performance is overall satisfactory.

Key words : Wireless Sensor Networks, Hierarchical Routing Protocol, Power Consumption, EEMCS, EHMS-EEMCS...