

UNIVERSITÉ KASDI MERBAH OUARGLA ALGERIE
FACULTÉ DES NOUVELLES TECHNOLOGIES DE L'INFORMATION ET DE LA
COMMUNICATION
DÉPARTEMENT D'INFORMATIQUE ET DE TECHNOLOGIE DE L'INFORMATION



Domaine : Mathématiques et Informatique

Piste : Informatique

Spécialité : Administration et sécurité des réseaux

Mémoire Master ACADÉMIQUE

Présenté par :

Bouguettaia Ouadjih

Benghedier Oussama

Titre :

**Optimisation de la sélection des cluster heads pour améliorer
l'efficacité des réseaux de capteurs sans fil**

Débat public sur :2024.2025

Les membres du comité :

Dr.Djeddar Afrah Encadreur UKM Ouargla

Année académique : 2024 / 2025

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

﴿وقل اعملوا فسيرى الله عملكم ورسوله والمؤمنون﴾

- [التوبة: 105]

{ وما توفّيقى إلا بالله عليه توكلت وإليه أنيب }

- [هود: 88]

الحمد لله الذي علم الإنسان ما لم يعلم، وسهّل لنا سبل العلم والعمل، ووفقنا لإنجاز هذا العمل المتواضع
وفي هذه اللحظة المميّزة من مسيرتي العلمية، لا يسعني إلا أن أعبر عن بالغ امتناني وصدق عرفاني لكل من كان له أثر طيب
في هذا المشوار

أتقدّم بخالص عبارات الشكر والتقدير إلى أستاذتي الفاضلة أفرّاح جدار، التي كانت خير مُوجّه وداعم. أشكرها على ما بذلته من
وقت وجهد، وعلى حسن توجيهها ونصائحها القيمة التي كان لها بالغ الأثر في إتمام هذا العمل. جزاها الله عني خير الجزاء،
وبارك في علمها وعملها

كما أتوجه بالشكر إلى كافة الأساتذة الذين وضعوا في نفسي بذور المعرفة، وساهموا في تكوين شخصيتي العلمية على مدار
سنوات الدراسة

وإلى عائلتي العزيزة، أبي وأمي، وكل من كان لهم الفضل بعد الله فيما أنا عليه اليوم، أقول: شكرًا من القلب على دعائكم
المستمر، صبركم، دعمكم المعنوي، ومساندتكم لي في كل خطوة. أنتم السند الحقيقي، وبدونكم لما وصلت إلى هذا المقام. أسأل
الله أن يطيل في أعماركم ويحفظكم لي

كما لا أنسى أصدقائي وزملائي الذين كانوا خير رفقة، وكان لوقوفهم إلى جانبي أثر لا يُنسى

بارك الله في كل من قدّم يد العون، صغیرها وكبیرها، وجعل هذا العمل خالصًا لوجهه الكريم، نافعًا للعلم والمجتمع

وما توفّيقى إلا بالله، عليه توكلت وإليه أنيب

Résumé

Dans les réseaux de capteurs sans fil (WSN), plusieurs protocoles tels que LEACH et LEACH++ ont été développés afin d'optimiser la consommation énergétique. Le protocole LEACH repose sur une sélection aléatoire des chefs de cluster (CH), ce qui entraîne un déséquilibre dans la répartition de l'énergie et une surcharge de certains nœuds. LEACH++ améliore cette approche en prenant en compte l'énergie résiduelle et la distance à la station de base, mais néglige l'historique de leadership, ce qui peut conduire à la sélection répétée des mêmes nœuds comme chefs.

Pour pallier ces limitations, nous proposons le protocole LEACH-EF, qui adopte une politique de sélection basée sur trois critères : l'énergie résiduelle, la distance à la station de base, et le nombre de fois qu'un nœud a déjà été élu chef dans les cycles précédents. Le protocole a été implémenté dans l'environnement TinyOS en utilisant le langage NesC, avec des structures de données adaptées pour représenter les paramètres essentiels tels que l'énergie, la distance et le score de leadership.

LEACH-EF repose sur une architecture modulaire, où chaque composant — sélection des CH, formation des clusters, transmission des données — est conçu indépendamment. La simulation a été réalisée à l'aide de l'outil TinyViz, permettant une visualisation dynamique du comportement du réseau.

Les résultats de simulation montrent que LEACH-EF permet une répartition équitable des rôles, prolonge la durée de vie du réseau, et améliore significativement l'efficacité énergétique par rapport aux protocoles précédents.

Mot-clé : La hiérarchie de clustering adaptatif à faible énergie (LEACH) : version améliorée de LEACH intégrant des critères d'énergie de distance LEACH-EF : LEACH avec Energy Factor (équité énergétique) CH : chef de cluster CHs.

Summary

In wireless sensor networks (WSNs), several protocols such as LEACH and LEACH++ have been developed to optimize energy consumption. The LEACH protocol relies on a random selection of cluster heads (CHs), which can lead to unbalanced energy distribution and increased load on certain nodes. LEACH++ improves this approach by considering residual energy and the distance to the base station, but it overlooks the leadership history, which may result in the repeated selection of the same nodes as leaders.

To address these limitations, we propose the LEACH-EF protocol, which adopts a selection policy based on three criteria: residual energy, distance to the base station, and the number of times a node has previously been elected as a cluster head. The protocol was implemented in the TinyOS environment using the NesC programming language, with custom data structures designed to represent key parameters such as energy, distance, and leadership score.

LEACH-EF is built on a modular architecture, where each component—cluster head selection, cluster formation, and data transmission—is developed independently. Simulation was carried out using the TinyViz tool, which enables dynamic visualization of network behavior.

Simulation results demonstrate that LEACH-EF achieves fair role distribution, extends the network lifetime, and significantly improves energy efficiency compared to previous protocols.

Keyword: Adaptive low-energy clustering hierarchy (LEACH): an improved version of LEACH integrating distance energy criteria. **LEACH-EF:** LEACH with Energy Factor (energy fairness). **CHs:** Cluster Heads.

ملخص

في شبكات أجهزة الاستشعار اللاسلكية (WSN) ، تم تطوير العديد من البروتوكولات مثل LEACH و LEACH++ بهدف تحسين كفاءة استهلاك الطاقة. يعتمد بروتوكول LEACH على اختيار عشوائي لقادة المجموعات (CH) ، مما يؤدي إلى استنزاف غير متوازن للطاقة وزيادة الحمل على بعض العقد. وقد جاء LEACH++ ليحسّن هذا النهج من خلال أخذ الطاقة المتبقية والمسافة إلى المحطة الأساسية في الاعتبار، لكنه لا يراعي تاريخ القيادة، مما قد يؤدي إلى تكرار اختيار نفس العقد كقادة.

لمعالجة هذه التحديات، نقترح بروتوكول LEACH-EF ، الذي يعتمد على سياسة اختيار متعددة المعايير تشمل: الطاقة المتبقية، المسافة إلى المحطة الأساسية، وعدد المرات التي تم فيها تعيين العقدة كقائد في الدورات السابقة. تم تنفيذ البروتوكول ضمن بيئة TinyOS باستخدام لغة البرمجة NesC ، مع اعتماد هيكل بيانات مخصصة لتمثيل المعايير الأساسية: الطاقة، المسافة، ودرجة القيادة.

يعتمد LEACH-EF على بنية معيارية (modulaire) ، حيث تم تصميم كل مكون بشكل مستقل، بدءًا من آلية اختيار قادة المجموعات، مرورًا بتشكيل العناقيد، وصولاً إلى مرحلة نقل البيانات. وقد تمت المحاكاة باستخدام أداة TinyViz التي توفر تصورًا ديناميكيًا لتصرف الشبكة.

أظهرت نتائج المحاكاة أن بروتوكول LEACH-EF يحقق تحسّنًا ملموسًا في توزيع الأدوار بين العقد، ويُطيل عمر الشبكة، كما يعزز الكفاءة الطاقية مقارنة بالبروتوكولات السابقة.

الكلمات المفتاحية: التسلسل الهرمي للتجميع التكميلي منخفض الطاقة (LEACH): نسخة محسّنة من بروتوكول LEACH تعتمد على معايير طاقة المسافة. LEACH-EF: بروتوكول LEACH يعمل بالطاقة – العناقيد.

Sommaire

Résumé :	i
Summary:	ii
ملخص :	iii
Sommaire :	iv
List de Tableau :	viii
Liste des abréviations :	ix
Introduction Générale :	1
Chapitre I : Généralités sur les réseaux de capteurs sans fil.....	2
I.1 Introduction :	2
I.2 Un nœud capteur :	2
I.2.1 Définition :	2
I.2.2 Architecture d'un nœud capteur :	2
I.3 Les réseaux de capteurs sans fil :	3
I.3.1 Définition :	3
I.3.2 Architecture d'un réseau de capteurs sans fil :	4
I.4 La pile protocolaire dans les réseaux de capteurs sans fil :	4
I.4.1 Les couches de la pile protocolaire :	4
I.4.2 Les niveaux de gestion dans les réseaux de capteurs sans fil :	5
I.5 Caractéristiques des réseaux de capteurs sans fil :	6
I.6 Les contraintes des réseaux de capteurs sans fil :	7
I.6.1 Les contraintes liées aux médias de transmission :	7
I.6.2 Les contraintes liées aux MAC :	7
I.7 Les différents facteurs de conception :	7
I.8 Domaines d'applications des réseaux de capteurs :	10
I.9 Enjeux particuliers dans les réseaux de capteurs :	13

I 10 Travaux existants sur les protocoles LEACH et dérivés	13
I.1. Les variantes les plus connues de LEACH :.....	14
I.2 Approches existantes :	14
I.3 Lacunes des approches existantes :	14
I.4 Positionnement du LEACH-EF :.....	15
I.5 Tableau comparatif des approches :.....	15
I.10 Conclusion	16
Chapitre II : Conception de l'Approche Proposée – Protocole LEACH-EF	16
II.1 Vue d'ensemble du protocole LEACH-EF :.....	16
II.2 Formule de priorité dans LEACH-EF :	16
II.3 Étapes du cycle LEACH-EF :	17
II.3.1 Initialisation :.....	18
II.3.2 Calcul de la priorité	18
II.3.3 Sélection des chefs de clusters	18
II.3.4 Communication intra-cluster	18
II.3.5 Agrégation et transmission	18
II.3.6 Mise à jour des états.....	18
II.4 Algorithme en pseudo-code :	19
II.5 Organigramme de fonctionnement :	21
Chapitre IV : Implémentation et simulation du protocole LEACH-EF	25
IV.1 Introduction :.....	25
IV.2 Objectif :.....	25
IV.3 Environnement de développement :.....	26
IV.3.1 TinyOS et NesC :	26
IV.3.2 TOSSIM :	26
IV.3.3 TinyViz : Interface graphique de simulation :.....	26
IV.3.4 Power TOSSIM : Extension pour l'estimation énergétique :	29
IV.3.5 justifications du choix de TOSSIM :.....	29
IV.4. Model de simulation :.....	30
IV.4.1 Model le Radio Utilisé : (First Order Radio Model)	30
IV.4.2 Paramètres de simulation :	32

IV.5. Implémentation sous TinyOS :	33
IV.5.1 Les fichiers de l'application développée :	33
IV.5.2 Structure de données utilisées :	34
b. Structure du cluster (ClusterInfo)	34
c. Données collectées (DataPacket)	34
IV.5.3 Structure de fichiers :	35
IV.5.4 Simulation dans un Environnement TinyViz :	36
IV.6 Résultats et Analyse Comparative :	38
IV.6.1 Critères de performances :	38
IV.6.2 Discussion des résultats :	40
IV.6.3 Résumé des performances de LEACH, LEACH++ et LEACH-EF :	45
IV.6.4 Conclusion	51
Conclusion Générale :	52
Références :	53

Liste des figures :

Figure II.2.2 : Architecture physique d'un nœud capteur.....	6
Figure II.3.2 : Architecture d'un réseau de capteurs sans fil (WSN).....	7
Figure II.4.2 : La pile protocolaire dans les RCSF.....	9
Figure III.3 : Étapes du cycle LEACH-EF	21
Figure III.4 : Algorithme en pseudo-code.....	23
Figure III.5 : Organigramme de fonctionnement.....	25
Figure IV.3.3.1 : Fenêtre principale de TinyViz.....	31
Figure IV.3.3.2 : Plugins disponibles dans TinyViz.....	32
Figure IV.5.4.1 : Déclenchement et relai du nouveau round, annonce du CH.....	46
Figure IV.5.4.2 : Formation de clusters et envoi des données.....	46
Figure IV.5.4.3 : Envoi du résultat d'agrégation au nœud puits.....	47
Figure IV.6.2.1 : Moyenne d'énergie consommée par les nœuds.....	50
Figure IV.6.2.2 : Écart type de l'énergie consommée.....	51
Figure IV.6.26 : Comparaison du MEC avant et après l'optimisation.....	54
Figure IV.6.3.1 : Durée de vie du réseau (FND, HND, LND).....	56
Figure IV.6.3.2 : Nombre de paquets reçus au puits.....	57
Figure IV.6.3.3 : Répartition des rôles de chef de cluster.....	58
Figure IV.6.3.4 : Énergie résiduelle au fil des rounds.....	59

List de Tableau :

Tab. I.5 : Tableau comparatif des approches.....	4
Tabl IV.4.2.1 : Paramètres de simulation et du modèle énergétique.....	36
Tab. IV.6.2.1 : Moyenne d'énergie consommée.....	49
Tab. IV.6.2.2 : l'écart type de l'énergie consommée.....	51

Liste des abréviations :

- **A/N** : Analogique / Numérique (utilisé pour le convertisseur CAN)
- **API** : Application Programming Interface
- **BER** : Bit Error Rate
- **BS** : Base Station (station de base)
- **CDMA** : Code Division Multiple Access
- **CH** : Cluster Head – Chef de cluster
- **CHs** : Cluster Heads
- **CM** : Cluster Member – Membre de cluster
- **CPU** : Central Processing Unit
- **ClusterInfo** : Structure décrivant un cluster
- **DataPacket** : Paquet de données échangé dans le réseau
- **EF** : Energy Factor
- **FND** : First Node Dies
- **HND** : Half Nodes Die
- **ID** : Identifier – Identifiant
- **J** : Joule
- **LEACH** : Low Energy Adaptive Clustering Hierarchy
- **LEACH++** : Version améliorée de LEACH intégrant des critères d'énergie et de distance
- **LEACH-C** : Centralized LEACH
- **LEACH-EF** : LEACH avec Energy Factor (équité énergétique)
- **LEACH-ET** : Energy and Time aware LEACH
- **LEACH-M** : Mobile LEACH
- **LEACH-TL** : Threshold LEACH
- **LLC** : Logical Link Control

- **LND** : Last Node Dies
- **MAC** : Medium Access Control – Contrôle d'accès au médium
- **MAX_NODES** : Nombre maximal de nœuds (dans les structures NesC)
- **MEC** : Moyenne d'Énergie Consommée
- **N/W** : Network
- **NesC** : Network Embedded System C – Langage de programmation pour TinyOS
- **NodeInfo** : Structure représentant un nœud dans NesC
- **ON/OFF** : Commande pour activer/désactiver un capteur
- **PowerTOSSIM** : Outil pour estimer la consommation énergétique dans TOSSIM
- **QoS** : Quality of Service
- **RCSF** : Réseau de Capteurs Sans Fil
- **RSSI** : Received Signal Strength Indicator
- **RTS/CTS** : Request To Send / Clear To Send
- **SNR** : Signal to Noise Ratio
- **Sink** : Nœud puits (récepteur central)
- **TDMA** : Time Division Multiple Access
- **TOSSIM** : TinyOS Simulator
- **TinyOS** : Système d'exploitation pour capteurs embarqués
- **TinyViz** : Interface graphique pour la simulation sous TOSSIM
- **WSN** : Wireless Sensor Network – Réseau de capteurs sans fil
- **mJ** : Mill

Introduction Générale :

Introduction Les réseaux de capteurs sans fil (Wireless Sensor Networks – WSNs) jouent un rôle central dans de nombreuses applications modernes telles que la surveillance environnementale, l'agriculture intelligente, les systèmes de santé connectés et la sécurité industrielle. Ces réseaux sont composés de nœuds capteurs autonomes capables de surveiller des phénomènes physiques et de transmettre des données à une station de base, souvent dans des environnements difficiles d'accès [1]

. L'un des principaux défis auxquels sont confrontés les WSNs est la gestion de l'énergie. Les capteurs sont généralement alimentés par des batteries non rechargeables, ce qui rend la consommation énergétique un facteur critique pour la durabilité du réseau. Face à cette contrainte, de nombreux protocoles de communication ont été conçus pour optimiser l'efficacité énergétique [2][3]

. Le protocole LEACH (Low Energy Adaptive Clustering Hierarchy) est l'un des plus utilisés dans ce domaine. Il repose sur une architecture hiérarchique basée sur la formation de clusters, où des chefs de clusters (CHs) sont désignés pour regrouper les données des nœuds membres avant de les transmettre à la station de base [4]. Toutefois, LEACH sélectionne les CHs de manière aléatoire, sans considération pour leur énergie résiduelle, leur distance, ni leur historique de sélection, ce qui peut provoquer un déséquilibre énergétique et une réduction prématurée de la durée de vie du réseau.

. Plusieurs améliorations ont été proposées, telles que LEACH++, qui introduit des critères comme l'énergie résiduelle et la distance à la station de base dans la sélection des CHs, et prend également en compte l'historique énergétique partiel [5]. Cependant, ces améliorations restent limitées en termes d'équité globale dans la répartition des rôles.

. Pour surmonter ces limites, ce travail propose un nouveau protocole nommé LEACH-EF (Energy-Fair LEACH). Contrairement à LEACH++, LEACH-EF intègre un troisième critère novateur : le nombre de fois qu'un nœud a été précédemment élu chef de cluster. Cette approche à trois critères — énergie résiduelle, distance, et fréquence d'élection antérieure — vise à assurer une rotation plus équitable des rôles, à réduire les déséquilibres énergétiques, et à prolonger significativement la durée de vie du réseau, comme suggéré également dans des travaux récents basés sur l'intelligence artificielle et la théorie des jeux [6][7].

. Le protocole a été implémenté dans l'environnement TinyOS en utilisant le langage NesC. Une simulation a été réalisée à l'aide de l'outil TinyViz, afin d'évaluer les performances du LEACH-EF comparativement à celles de LEACH et LEACH++, en s'appuyant sur des indicateurs clés : la consommation énergétique, le nombre de paquets reçus par la station de base, et l'équité dans la répartition des rôles.

Chapitre I : Généralités sur les réseaux de capteurs sans fil

I.1 Introduction :

Au cours des dernières décennies, nous avons assisté à une miniaturisation du matériel informatique. Cette tendance à la miniaturisation a apporté une nouvelle génération de réseaux informatiques et télécoms présentant des défis importants. Les réseaux de capteurs sans fil (RCSF) sont l'une des technologies visant à résoudre les problèmes de cette nouvelle ère de l'informatique embarquée et omniprésente. De très nombreux états de l'art ont été proposés, nous allons donc dans le présent chapitre présenter les réseaux de capteurs sans fil, leurs architectures de communication et leurs applications.

Nous allons discuter également les principaux facteurs et contraintes qui influencent la Conception des réseaux de capteurs sans fil.

I.2 Un nœud capteur :

I.2.1 Définition :

Un nœud capteur est un dispositif équipé de fonctionnalités de sensation avancées. Il mesure ou détecte un événement réel, comme le mouvement, la chaleur ou la lumière et convertit la valeur mesurée dans une représentation analogique ou numérique. Il prélève des informations et élabore à partir d'une grandeur physique (information d'entrée), une autre grandeur physique de nature électrique.

I.2.2 Architecture d'un nœud capteur : [A01]

Un nœud capteur est composé de plusieurs éléments ou modules correspondant chacun à une tâche particulière d'acquisition, de traitement, ou de transmission de données. Il comprend également une source d'énergie.

- **Unité de captage (Sensing unit) :** elle est composée de deux sous unités, un dispositif de capture physique qui prélève l'information de l'environnement local et un convertisseur analogique/numérique appelé CAN. [A02]
- **l'unité de traitement :** composée d'un processeur et d'une mémoire intégrant un système d'exploitation spécifique (TinyOS, par exemple). Cette unité possède deux interfaces, une interface pour l'unité d'acquisition et une interface pour l'unité de communication. Elle acquiert les informations en provenance de l'unité d'acquisition et les envoie à l'unité de communication. Cette unité est chargée aussi d'exécuter les protocoles de communications qui permettent de faire collaborer le capteur avec d'autres capteurs. Elle peut aussi analyser les données captées. [A03] [A04] [A02]

- **Unité de communication (Transceiver unit)** : elle est composée d'un émetteur/récepteur (module radio) permettant la communication entre les différents nœuds du réseau. [A02]
- **Unité d'énergie (Power unit)** : c'est la batterie qui, n'est généralement ni rechargeable ni remplaçable. La capacité d'énergie limitée au niveau des capteurs représente la contrainte principale lors de la conception de protocoles pour les réseaux de capteurs. [A05]

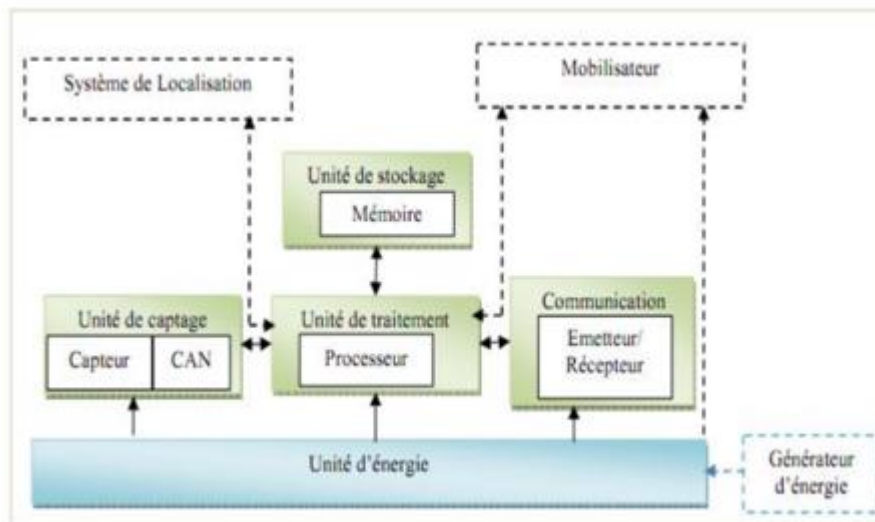


Figure II.2.2 : Architecture physique d'un nœud capteur

I.3 Les réseaux de capteurs sans fil :

I.3.1 Définition :

Les réseaux de capteurs sans fil (RCSF) sont un type particulier des réseaux Ad-hoc, dans lesquels les nœuds sont des « capteurs intelligents ». Ils se composent généralement d'un grand nombre de capteurs communicants entre eux via des liens radio pour le partage d'information et le traitement coopératif. Dans ce type de réseau, les capteurs échangent des informations par exemple sur l'environnement pour construire une vue globale de la région contrôlée, qui est rendue accessible à l'utilisateur externe par un ou plusieurs nœud(s). Les données collectées par ces capteurs sont acheminées directement ou via les autres capteurs de proche en proche à un « point de collecte », appelé station de base (ou SINK s'il s'agit d'un nœud). Cette dernière peut être connectée à une machine puissante via internet ou par satellite.

En outre, l'utilisateur peut adresser ses requêtes aux capteurs en précisant l'information d'intérêt.

I.3.2 Architecture d'un réseau de capteurs sans fil :

Un réseau de capteurs sans fil est composé d'un grand nombre de nœuds. Chaque capteur est doté d'un module d'acquisition qui lui permet de mesurer des informations environnementales : température, humidité, pression, accélération, sons, image, vidéo etc.

Les données collectées par ces nœuds capteurs sont routées vers une ou plusieurs stations de base ou nœud puits (Sink en anglais). Ce dernier est un point de collecte de données capturées. Il peut communiquer les données collectées à l'utilisateur final à travers un réseau de communication, éventuellement l'Internet ou un satellite. L'utilisateur peut à son tour utiliser la station de base comme passerelle, afin de transmettre ses requêtes au réseau.

En général, un RCSF est composé de quatre éléments montrés par la figure suivante :

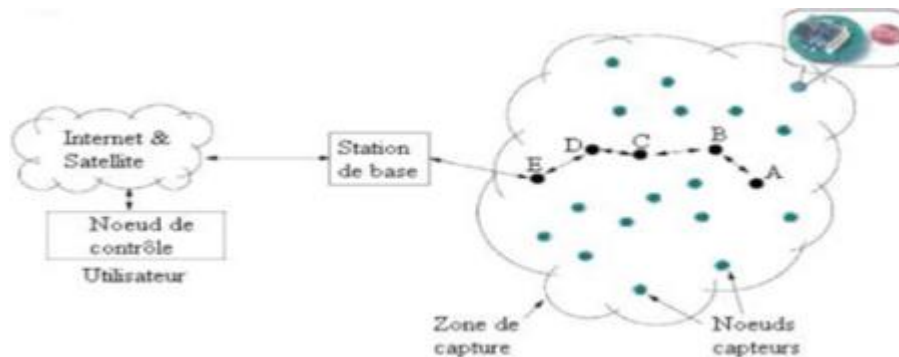


Figure II.3.2 : Architecture d'un réseau de capteurs sans fil (WSN)

I.4 La pile protocolaire dans les réseaux de capteurs sans fil :

Dans les réseaux de capteurs sans fil, les nœuds capteurs utilisent une pile protocolaire spécifique en trois dimensions composée de cinq couches et de trois niveaux (niveau de gestion d'énergie, niveau de gestion de la mobilité, niveau de gestion des tâches). Les trois niveaux sont accessibles par toutes les couches de la pile. Il existe une interaction entre les différentes couches et les différents niveaux.

I.4.1 Les couches de la pile protocolaire : [A06] [A07]

Les cinq couches de la pile protocolaire sont les suivantes :

- **La couche physique**

Cette couche décrit les procédures et les fonctions mécaniques et électriques nécessaires pour établir, maintenir et libérer une connexion physique entre deux ou plusieurs capteurs. Elle est responsable de la sélection des fréquences (utilisation des bandes ISM), de la détection d'un signal, et le traitement du signal (la modulation).

- **La couche liaison de données**

Cette couche a pour rôle le multiplexage des flux de données, le partage de l'accès au medium et le contrôle d'erreur. Elle assure une connexion point à point ou point à multipoint fiable dans une communication réseau.

La couche liaison de données est composée de deux couches : la couche LLC

(Contrôle des Liens Logiques) et la couche MAC (Contrôle d'accès au medium).

Les protocoles de la sous couche MAC sont appelés à effectuer d'importantes opérations : établir des liens de communication entre les nœuds capteurs voisins, fournir une fiabilité entre ces nœuds voisins et partager équitablement les canaux de communication entre les nœuds du réseau.

- **La couche réseau**

Cette couche est responsable des fonctions de routage et de la gestion de la topologie.

- **La couche transport**

Cette couche est responsable du maintien des flux de données dans les applications utilisées et de la sauvegarde des données dans le cache des capteurs. Elle est particulièrement nécessaire pour accéder au réseau de capteurs par le biais d'un réseau externe comme l'internet.

- **La couche application**

Cette couche est conçue suivant les fonctionnalités des capteurs. Elle doit fournir des mécanismes :

- pour l'interprétation des données perçues.
- permettant à l'utilisateur d'interagir avec le réseau de capteurs.
- qui rendent transparents à l'utilisateur les logiciels utilisés dans les couches inférieures.
- d'agrégation de données.
- de synchronisation des nœuds.
- d'authentification, de distribution de clés dans le but d'assurer la sécurité des données.

I.4.2 Les niveaux de gestion dans les réseaux de capteurs sans fil :

Les niveaux de gestion propres aux réseaux de capteurs sans fil sont les suivants :

- Le niveau de gestion d'énergie : gère l'énergie consommée par les capteurs.

- Le niveau de gestion de la mobilité : ce niveau détecte et enregistre le mouvement des nœuds capteurs et permet de maintenir l'itinéraire d'un capteur vers un utilisateur et garder la trace de l'emplacement de ses voisins.
- Le niveau de gestion des tâches : s'occupe de la distribution des tâches pour région données

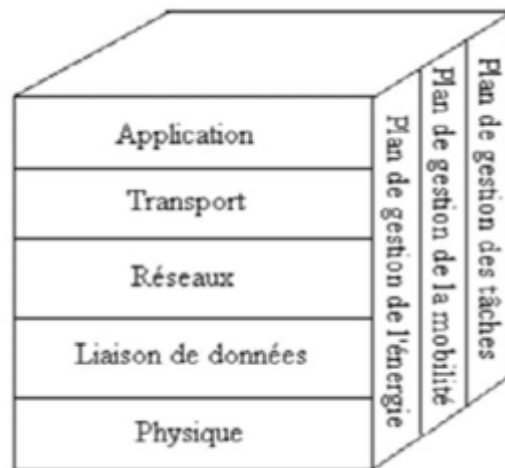


Figure II.4.2 : La pile protocolaire dans les RCSF

I.5 Caractéristiques des réseaux de capteurs sans fil : [A08]

Un réseau de capteurs présente les caractéristiques suivantes :

- **absence d'infrastructure** : les réseaux Ad-hoc en général et les réseaux de capteurs en particulier se distinguent des autres réseaux par la propriété d'absence d'infrastructure préexistante et de tout genre d'administration centralisée.
- **taille importante** : un réseau de capteurs peut contenir des milliers de nœuds.
- **interférences** : les liens radio ne sont pas isolés, deux transmissions simultanées sur une même fréquence, ou utilisant des fréquences proches, peuvent interférer.
- **topologie dynamique** : les capteurs peuvent être attachés à des objets mobiles qui se déplacent d'une façon libre et arbitraire rendant ainsi la topologie du réseau fréquemment changeante.
- **sécurité physique limitée** : les réseaux de capteurs sans fil sont plus touchés par le paramètre de sécurité que les réseaux filaires classiques. Cela se justifie par les contraintes et limitations physiques qui font que le contrôle des données transférées doit être minimisé.

- **bande passante limitée** : une des caractéristiques primordiales des réseaux basés sur la communication sans fil est l'utilisation d'un médium de communication partagé. Ce partage fait que la bande passante réservée à un nœud est limitée.

- **contrainte d'énergie, de stockage et de calcul** : la caractéristique la plus critique dans les réseaux de capteurs est la modestie de ses ressources énergétiques car chaque capteur du réseau possède de faibles ressources en termes d'énergie (batterie). Afin de prolonger la durée de vie du réseau, une minimisation des dépenses énergétiques est exigée chez chaque nœud. Ainsi, la capacité de stockage et la puissance de calcul sont limitées dans un capteur.

I.6 Les contraintes des réseaux de capteurs sans fil :

Les principales contraintes des réseaux de capteurs sans fil sont : la bande passante de transmission limitée et la puissance de transmission limitée.

I.6.1 Les contraintes liées aux médias de transmission :

La consommation d'énergie est un point crucial dans les réseaux de capteurs sans fil.

Les circuits de communication et les antennes sont les éléments qui consomment le plus d'énergie.

I.6.2 Les contraintes liées aux MAC :

La perte d'énergie au niveau de la couche MAC est causée par plusieurs facteurs,

Notamment :

- Les collisions qui sont inévitables lorsque le canal de transmission est partagé de façon distribuée. Les paquets de données victimes de collisions doivent être retransmis ce qui implique une consommation d'énergie supplémentaire.
- La plupart des protocoles MAC distribués nécessitent un contrôle de message pour une transmission fiable des données (RTS/CTS dans IEEE 802.11). Ce type de contrôle de message consomme de l'énergie.
- L'overhearing et l'idle listening consomment aussi de l'énergie.
 - Overhearing : un nœud reçoit un paquet qui ne lui est pas destiné.
 - Idle listening : un nœud écoute le canal pour connaître son état (occupé ou libre).

I.7 Les différents facteurs de conception : [A01] [A10] [A11]

La conception des réseaux de capteurs est influencée par de nombreux facteurs comme la tolérance aux pannes, les coûts de production, la consommation d'énergie, l'environnement ou

la topologie du réseau. Ces facteurs représentent la base de la conception des protocoles ou d'algorithmes pour les réseaux de capteurs. [A01] [A10] [A11]

Nous allons introduire les facteurs les plus importants à prendre en compte dans la conception d'un réseau de capteurs sans fil. Ces facteurs sont un guide dans la conception et l'implémentation des protocoles des différentes couches protocolaires propres aux réseaux de capteurs sans fil.

- **Tolérance aux pannes** : Les nœuds peuvent être sujets à des pannes dues à leur fabrication (ce sont des produits de série bon marché, il peut donc y avoir des capteurs défectueux) ou plus fréquemment à un manque d'énergie. Les interactions externes (chocs, interférences) peuvent aussi être la cause des dysfonctionnements. Afin que les pannes n'affectent pas la tâche première du réseau, il faut évaluer la capacité du réseau à fonctionner sans interruption.
- **Coût de fabrication** : Les nœuds sont des produits fabriqués en série du fait de leur grand nombre. Il faut que le coût de fabrication de ces nœuds soit tel que le coût global du réseau ne soit pas supérieur à celui d'un réseau classique afin de pouvoir justifier son intérêt.
- **Topologie du réseau** : En raison de leur forte densité dans la zone à observer, il faut que les nœuds capteurs soient capables d'adapter leur fonctionnement afin de maintenir la topologie souhaitée.

On distingue généralement trois phases dans la mise en place et l'évolution d'un réseau :

- **Déploiement** : Les nœuds sont soit répartis de manière prédéfinie soit de manière aléatoire (lancés en masse depuis un avion). Il faut alors que ceux-ci s'organisent de manière autonome.
- **Post-Déploiement -Exploitation** : Durant la phase d'exploitation, la topologie du réseau peut être soumise à des changements dus à des modifications de la position des nœuds ou bien à des pannes.
- **Redéploiement** : L'ajout de nouveaux capteurs dans un réseau existant implique aussi une remise à jour de la topologie.

Consommation d'énergie : L'économie d'énergie est une des problématiques majeures dans les réseaux de capteurs. En effet, la recharge des sources d'énergie est souvent trop coûteuse et parfois impossible. Il faut donc que les capteurs économisent au maximum l'énergie afin de pouvoir fonctionner.

- **Durée de vie du réseau**

C'est l'intervalle de temps qui sépare l'instant de déploiement du réseau de l'instant où l'énergie du premier nœud s'épuise. Selon l'application, la durée de vie exigée pour un réseau peut varier entre quelques heures et plusieurs années.

- **Ressources limitées**

En plus de l'énergie, les nœuds capteurs ont aussi une capacité de traitement et de mémoire limitée. En effet, les industriels veulent mettre en œuvre des capteurs simples, petits et peu coûteux qui peuvent être achetés en masse.

- **Bande passante limitée**

Afin de minimiser l'énergie consommée lors de transfert de données entre les nœuds, les capteurs opèrent à bas débit. Typiquement, le débit utilisé est de quelques dizaines de Kb/s.

Un débit de transmission réduit n'est pas handicapant pour un réseau de capteurs où les fréquences de transmission ne sont pas importantes.

- **Facteur d'échelle**

Le nombre de nœuds déployés pour une application peut atteindre des milliers. Dans ce cas, le réseau doit fonctionner avec des densités de capteurs très grandes. Un nombre aussi important de nœuds engendre beaucoup de transmissions inter nodales et nécessite que la station de base soit équipée de mémoire suffisante pour stocker les informations reçues.

- **Topologie dynamique**

La topologie des réseaux de capteurs peut changer au cours du temps pour les raisons suivantes :

- Les nœuds capteurs peuvent être déployés dans des environnements hostiles (champ de bataille par exemple), la défaillance d'un nœud capteur est, donc très probable.
- Un nœud capteur peut devenir non opérationnel à cause de l'expiration de son énergie.
- Dans certaines applications, les nœuds capteurs et les stations de base sont mobiles.

Agrégation de données

Dans les réseaux de capteurs, les données produites par les nœuds capteurs voisins sont très corrélées spatialement et temporellement. Ceci peut engendrer la réception par la station de base d'informations redondantes. Réduire la quantité d'informations redondantes transmises par les capteurs permet de réduire la consommation d'énergie dans le réseau et ainsi d'améliorer sa durée de vie. L'une des techniques utilisée pour réduire la transmission d'informations redondantes est l'agrégation des données. Avec cette technique, les nœuds intermédiaires agrègent l'information reçue de plusieurs sources. Cette technique est connue aussi sous le nom de fusion de données.

- **Les contraintes matérielles**

Un capteur est composé de quatre unités de base et de composants optionnels qui dépendent de l'application. Certaines techniques de routage et tâches de perception nécessitent des informations sur la localisation avec une grande exactitude. Par conséquent un capteur doit être doté d'un système de localisation. Un mobilisateur peut être nécessaire pour déplacer des nœuds afin d'effectuer certaines tâches. Toutes ces sous unités occupent de la place dans un dispositif dont la taille peut être inférieure à 1 cm³. Ajouter à cela le fait que l'autonomie de la batterie des capteurs est très limitée. Compte tenu de tous ces facteurs, les capteurs doivent être disponibles, autonomes et doivent s'adapter à l'environnement.

- **L'environnement**

Généralement les nœuds sont déployés dans des zones géographiques lointaines pour mesurer et contrôler un phénomène. Le champ de captage peut être un océan, un champ contaminé par des produits chimiques ou biologiques, un champ de bataille, un domicile ou encore un building. Généralement un champ de captage est un environnement inaccessible.

- **Le media de transmission**

Dans un réseau de capteurs, les nœuds sont interconnectés à travers une interface de communication sans fil : une liaison radio, infrarouge ou optique. Pour réaliser les différentes opérations d'un réseau, le media de transmission choisi doit être fiable. La plupart des réseaux de capteurs utilisent les radios fréquences pour communiquer.

I.8 Domaines d'applications des réseaux de capteurs :

La miniaturisation, l'adaptabilité, le faible coût et l'avancement dans les communications sans fil permettent aux réseaux de capteurs d'envahir plusieurs domaines d'applications. Ils permettent aussi d'étendre le domaine des applications existantes. Parmi ces domaines où ces réseaux se révèlent très utiles et peuvent offrir de meilleures contributions, on peut noter le militaire, la santé, l'environnemental, et les maisons intelligentes...

- **Surveillance militaire et traque de cibles [A12]**

Comme beaucoup d'autres technologies de l'information, les réseaux de capteurs sans fil proviennent principalement de la recherche militaire. Des réseaux de capteurs autonomes sont envisagés comme l'ingrédient essentiel dans cette lancée vers des systèmes de guerre centrés sur les réseaux. Ils peuvent être rapidement déployés et utilisés pour la surveillance des champs de bataille afin de fournir des renseignements concernant l'emplacement, le nombre, le mouvement, et l'identité des soldats et des véhicules, ou bien encore pour la détection des agents chimiques, biologiques et nucléaires.

- **Applications à la sécurité : [A13]**

L'application des réseaux de capteurs dans le domaine de la sécurité peut diminuer considérablement les dépenses financières consacrées à la sécurisation des lieux et des êtres humains. Ainsi, l'intégration des capteurs dans de grandes structures telles que les ponts ou les bâtiments aidera à détecter les fissures et les altérations dans la structure suite à un séisme ou au vieillissement de la structure. Le déploiement d'un réseau de capteurs de mouvement peut constituer un système d'alarme qui servira à détecter les intrusions dans une zone de surveillance.

- **Applications environnementales : [A08]**

Le contrôle des paramètres environnementaux par les réseaux de capteurs peut donner naissance à plusieurs applications. Par exemple, le déploiement des thermo-capteurs dans une forêt peut aider à détecter un éventuel début de feu et par suite faciliter la lutte contre les feux de forêt avant leur propagation. Le déploiement des capteurs chimiques dans les milieux urbains peut aider à détecter la pollution et analyser la qualité d'air. De même leur déploiement dans les sites industriels empêche les risques industriels tels que la fuite de produits toxiques (gaz, produits chimiques, éléments radioactifs, pétrole, etc.).

Dans le domaine de l'agriculture, les capteurs peuvent être utilisés pour réagir aux changements climatiques par exemple le processus d'irrigation lors de la détection de zones sèches dans un champ agricole.

- **Applications médicales : [A14]**

Dans le domaine de la médecine, les réseaux de capteurs peuvent être utilisés pour assurer une surveillance permanente des organes vitaux de l'être humain grâce à des micro capteurs qui pourront être avalés ou implantés sous la peau (surveillance de la glycémie, détection de cancers, ...). Ils peuvent aussi faciliter le diagnostic de quelques maladies en effectuant des mesures physiologiques telles que : la tension artérielle, battements du cœur, ... à l'aide des capteurs ayant chacun une tâche bien particulière. Les données physiologiques collectées par les capteurs peuvent être stockées pendant une longue durée pour le suivi d'un patient.

D'autre part, ces réseaux peuvent détecter des comportements anormaux (chute d'un lit, choc, cri,...) chez les personnes dépendantes (handicapées ou âgées).

- **La domestique : [A15]**

Avec le développement technologique, les capteurs peuvent être embarqués dans des appareils, tels que les aspirateurs, les fours à micro-ondes, les réfrigérateurs,... Ces capteurs embarqués peuvent interagir entre eux et avec un réseau externe via internet pour permettre à un utilisateur de contrôler les appareils domestiques localement ou à distance.

Le déploiement des capteurs de mouvement et de température dans les futures maisons dites intelligentes permet d'automatiser plusieurs opérations domestiques telles que : la lumière s'éteint et la musique se met en état d'arrêt quand la chambre est vide, la climatisation et le chauffage s'ajustent selon les points multiples de mesure, le déclenchement d'une alarme par le capteur anti-intrusion quand un intrus veut accéder à la maison.

• Applications commerciales :

Des nœuds capteurs pourraient améliorer le processus de stockage et de livraison. Le réseau ainsi formé, pourra être utilisé pour connaître la position, l'état et la direction d'un paquet ou d'une cargaison. Un client attendant un paquet peut alors avoir un avis de livraison en temps réel et connaître la position du paquet. Des entreprises manufacturières, via des réseaux de capteurs pourraient suivre le procédé de production à partir des matières premières jusqu'au produit final livré. Grâce aux réseaux de capteurs, les entreprises pourraient offrir une meilleure qualité de service tout en réduisant leurs coûts. Les produits en fin de vie pourraient être mieux démontés et recyclés ou réutilisés si les micro-capteurs en garantissent le bon état.

Dans les immeubles, le système de climatisation peut être conçu en intégrant plusieurs micro-capteurs dans les tuiles du plancher et les meubles. Ainsi, La climatisation pourra être déclenchée seulement aux endroits où il y a des personnes présentes et seulement si c'est nécessaire.

I.9 Enjeux particuliers dans les réseaux de capteurs : [A08]

Les deux enjeux fondamentaux dans les réseaux de capteurs : le routage et la structuration des réseaux.

Le routage permet l'acheminement des informations vers une destination donnée à travers un réseau de connexion. Le problème de routage consiste à déterminer un acheminement optimal des paquets à travers le réseau au sens d'un certain critère de performance comme la consommation énergétique. Le problème consiste à trouver l'investissement de moindre coût qui assure le routage du trafic nominal et garantit la qualité de service.

Le problème qui se pose dans le contexte des réseaux de capteurs est l'adaptation de la méthode d'acheminement utilisée avec le grand nombre de nœuds existant dans un environnement caractérisé par de changements de topologies, de modestes capacités de calcul, de sauvegarde, et d'énergie.

Toute conception de protocole de routage implique l'étude des problèmes suivants :

- Minimiser la charge du réseau en optimisant le nombre d'envois et de réceptions des paquets. Cette minimisation aboutit à une consommation énergétique minimale et une longue durée de vie du réseau.
- Offrir un support pour pouvoir effectuer des communications multi-sauts fiables.
- Assurer un routage optimal si possible.
- Offrir une bonne qualité concernant les temps de latence.
- Auto organiser le réseau, ceci peut être nécessaire dans plusieurs cas.

Un réseau comportant un grand nombre de nœuds placés dans des endroits hostiles où la configuration manuelle n'est pas faisable doit être capable de s'auto-organiser. Un autre cas est celui où un nœud est inséré ou retiré (à cause d'un manque d'énergie ou destruction physique). Ainsi le réseau doit être capable de se reconfigurer pour continuer son fonctionnement.

I.10 Travaux existants sur les protocoles LEACH et dérivés

I.1. Les variantes les plus connues de LEACH :

Depuis l'introduction du protocole LEACH par Heinzelman et al. [B01], de nombreuses variantes ont été proposées pour améliorer sa performance, en particulier la consommation énergétique et la gestion équitable des chefs de cluster (CHs).

Parmi les variantes les plus citées, LEACH-C (Centralized LEACH) [B02] utilise une approche centralisée où la station de base sélectionne les CHs en fonction de la position et de l'énergie des nœuds. Cette méthode améliore la couverture spatiale mais augmente les coûts de communication et de traitement.

LEACH-M (Mobile LEACH), proposé par Vidhya et al. [B03], étend LEACH pour les environnements à topologie mobile. Il offre une meilleure résilience face au déplacement des nœuds, mais engendre une complexité accrue pour le suivi de la mobilité.

LEACH-TL (Threshold LEACH) [B04] introduit un seuil d'énergie empêchant les nœuds faibles de devenir CHs, protégeant ainsi les ressources énergétiques critiques. Toutefois, cette approche peut réduire le nombre de CHs valides lors de certaines itérations.

Dans un autre registre, LEACH-ET (Energy and Time aware LEACH) [B05] prend en compte non seulement l'énergie résiduelle mais aussi la latence de communication. Il est conçu pour les applications sensibles au temps mais nécessite des traitements plus complexes.

Enfin, LEACH++ est une amélioration non standardisée présente dans plusieurs études récentes [B06], intégrant une pondération entre l'énergie et la distance, et dans certains cas le multi-saut. Néanmoins, il ne prend pas en considération l'historique de sélection des CHs, ce qui provoque un déséquilibre dans la répartition des rôles.

I.2 Approches existantes :

De nombreuses approches ont été étudiées dans la littérature pour améliorer la sélection des CHs. Le choix aléatoire, utilisé dans LEACH original, est peu efficace à long terme, car il ne prend en compte aucun critère de performance.

D'autres méthodes privilégient l'énergie résiduelle, comme LEACH-C ou LEACH-TL. Elles permettent une meilleure longévité globale du réseau mais ne garantissent pas une équité dans la sélection des CHs, certains nœuds pouvant être choisis plusieurs fois de manière répétée.

Certaines variantes, comme LEACH++ et LEACH-ET, utilisent la distance à la station de base comme critère de sélection, réduisant ainsi les coûts de transmission. Cependant, elles négligent également l'équité entre les nœuds.

Un point commun à la majorité de ces approches est qu'elles ignorent l'historique de leadership. En l'absence de mécanismes pour limiter la sélection répétée des mêmes nœuds, cela entraîne une surcharge énergétique sur ceux-ci et accélère leur épuisement.

I.3 Lacunes des approches existantes :

Malgré leurs améliorations spécifiques, les variantes de LEACH souffrent de plusieurs limitations :

- La plupart ne combinent pas les trois critères clés pour une sélection équitable et efficace des CHs : énergie, distance et historique.
- Elles provoquent souvent un déséquilibre de charge, compromettant la durée de vie du réseau.
- Certaines approches, bien qu'énergétiquement performantes, manquent de flexibilité pour les scénarios dynamiques ou sensibles au temps.

Ces lacunes justifient la nécessité d'un protocole plus complet, capable de garantir à la fois efficacité énergétique et équité dans la rotation des rôles.

I.4 Positionnement du LEACH-EF :

Le protocole LEACH-EF (Energy-Fair LEACH) a été introduit pour surmonter ces limitations. Il repose sur une stratégie de sélection multi-critères, combinant :

- L'énergie résiduelle, pour éviter l'épuisement prématuré des nœuds faibles,
- La distance à la station, afin de minimiser l'énergie de transmission,
- L'historique de leadership, garantissant une répartition équitable des rôles de CH.

Cette approche permet à LEACH-EF de se positionner comme une solution équilibrée, robuste et éthique, prolongeant la durée de vie du réseau tout en assurant une répartition équitable de la charge.

I.5 Tableau comparatif des approches :

Protocole	Énergie résiduelle	Distance à la station	Historique de leadership	Mobilité supportée	Complexité
LEACH [1]	✗	✗	✗	✗	Faible
LEACH-C [2]	✓	✓	✗	✗	Moyenne
LEACH-M [3]	✗	✗	✗	✓	Élevée
LEACH-TL [4]	✓ (seuil)	✗	✗	✗	Moyenne
LEACH-ET [5]	✓	✓	✗	✗	Élevée
LEACH++ [6]	✓	✓	✗	✗	Moyenne
LEACH-EF	✓	✓	✓	✗	Moyenne

Tab. I.5 : Tableau comparatif des approches

II.10 Conclusion :

Les réseaux de capteurs sans fil présentent un intérêt considérable et une nouvelle étape dans l'évolution des technologies de l'information et de la communication. Cette nouvelle technologie suscite un intérêt croissant vu la diversité de ces applications : santé, environnement, industrie et même dans le domaine sportif.

Dans ce premier chapitre, nous avons présenté les réseaux de capteurs sans fil, leurs architectures de communication, la pile protocolaire des capteurs et leurs diverses applications. Cependant, nous avons remarqué que plusieurs facteurs et contraintes compliquent la gestion de ce type de réseaux. En effet, les réseaux de capteurs se caractérisent par une capacité énergétique limitée rendant l'optimisation de la consommation d'énergie dans des réseaux pareils une tâche critique pour prolonger la durée de vie du réseau.

Chapitre II : Conception de l'Approche Proposée – Protocole LEACH-EF

II.1 Vue d'ensemble du protocole LEACH-EF :

LEACH-EF (Low Energy Adaptive Clustering Hierarchy – Energy and Fairness) est une amélioration du protocole LEACH classique, conçue pour les réseaux de capteurs sans fil (RCSF). Son objectif principal est d'optimiser la consommation énergétique tout en assurant une répartition équitable du rôle de chef de cluster entre les nœuds. Il introduit un mécanisme de calcul de priorité basé sur plusieurs facteurs comme l'énergie résiduelle, le nombre de fois où un nœud a été chef, et la distance moyenne à la base station. LEACH-EF fonctionne par cycles, chacun comportant plusieurs phases : initialisation, calcul de priorité, sélection des chefs, communication intra-cluster, agrégation et transmission des données, et mise à jour des états.

II.2 Formule de priorité dans LEACH-EF :

La sélection des chefs de cluster dans le protocole LEACH-EF repose sur une formule de priorité innovante, qui permet une évaluation équitable et efficace des nœuds candidats à la fonction de chef. Cette formule combine trois critères fondamentaux : l'énergie résiduelle du nœud, son historique de leadership, et sa proximité avec la station de base. La formule est donnée par :

$$P_i = \alpha \cdot \left(\frac{E_i}{E_{max}} \right) - \beta \cdot \left(\frac{N_i}{N_{max}} \right) + \gamma \cdot \left(1 - \frac{d_i}{d_{max}} \right)$$

Où :

- P_i : priorité du nœud i pour être élu chef.
- E_i : énergie résiduelle actuelle du nœud i.
- E_{max} : énergie maximale détectée parmi tous les nœuds du réseau.
- N_i : nombre de fois où le nœud i a déjà été chef.
- N_{max} : nombre maximal de mandats de chef observés dans le réseau.
- d_i : distance entre le nœud i et la station de base.
- d_{max} : distance maximale entre les nœuds et la station de base.
- α, β, γ : coefficients d'ajustement pondérant chaque critère (tels que $\alpha + \beta + \gamma = 1$)

Interprétation :

- Le terme $\frac{E_i}{E_{max}}$ favorise les nœuds ayant une énergie plus élevée.
- Le terme $\frac{N_i}{N_{max}}$ pénalise les nœuds qui ont déjà été chefs plusieurs fois, pour équilibrer les rôles dans le temps.
- Le terme $1 - \frac{d_i}{d_{max}}$ privilégie les nœuds plus proches de la station de base, afin de réduire les coûts de communication.

Justification :

Cette formule garantit que les nœuds avec une énergie élevée, une faible fréquence de sélection précédente et une bonne proximité avec la station de base ont plus de chances d'être élus comme chefs de cluster. Elle permet ainsi de prolonger la durée de vie du réseau en évitant la surcharge de certains nœuds et en améliorant l'équilibre énergétique global.

II.3 Étapes du cycle LEACH-EF :

Le protocole LEACH-EF repose sur un cycle d'exécution périodique divisé en six étapes principales. Chaque étape vise à assurer une sélection équitable et efficace des chefs de clusters, tout en optimisant la consommation énergétique globale du réseau.

II.3.1 Initialisation :

- Au début de chaque cycle, tous les nœuds transmettent à la station de base les informations suivantes :
 - Leur position géographique.
 - Leur niveau d'énergie résiduelle.
 - Leur historique de sélection en tant que chef.
- Le système central (station de base) crée une table d'état globale contenant les paramètres de tous les nœuds.

II.3.2 Calcul de la priorité

- Chaque nœud calcule sa priorité P_i en utilisant la formule décrite précédemment, qui prend en compte :
 - L'énergie résiduelle.
 - La distance à la station de base.
 - Le nombre de fois où le nœud a déjà été chef.
- Ce calcul peut être réalisé localement (dans chaque nœud) ou de manière centralisée (par la station de base).

II.3.3 Sélection des chefs de clusters

- Les nœuds ayant les meilleures priorités sont sélectionnés comme chefs de clusters.
- Un seuil de priorité peut être appliqué afin de limiter le nombre de chefs sélectionnés, et ainsi maintenir un équilibre dans le réseau.

II.3.4 Communication intra-cluster

- Chaque chef de cluster diffuse un message d'annonce pour informer les autres nœuds de sa disponibilité.
- Les nœuds non-chefs rejoignent le chef dont le signal est le plus fort, c'est-à-dire celui le plus proche.

II.3.5 Agrégation et transmission

- Les nœuds membres envoient leurs données à leur chef respectif.
- Le chef procède à l'agrégation des données pour réduire la redondance, puis transmet les données agrégées à la station de base.

II.3.6 Mise à jour des états

- Chaque nœud met à jour son niveau d'énergie en fonction de son activité.
- Si le nœud a été chef, son compteur de rôles de chef est incrémenté.

- Enfin, le système retourne à l'étape d'initialisation pour entamer un nouveau cycle.

Conclusion :

Cette structuration cyclique permet au protocole LEACH-EF d'assurer une rotation équitable des responsabilités, de préserver l'énergie des nœuds, et d'augmenter significativement la durée de vie du réseau.

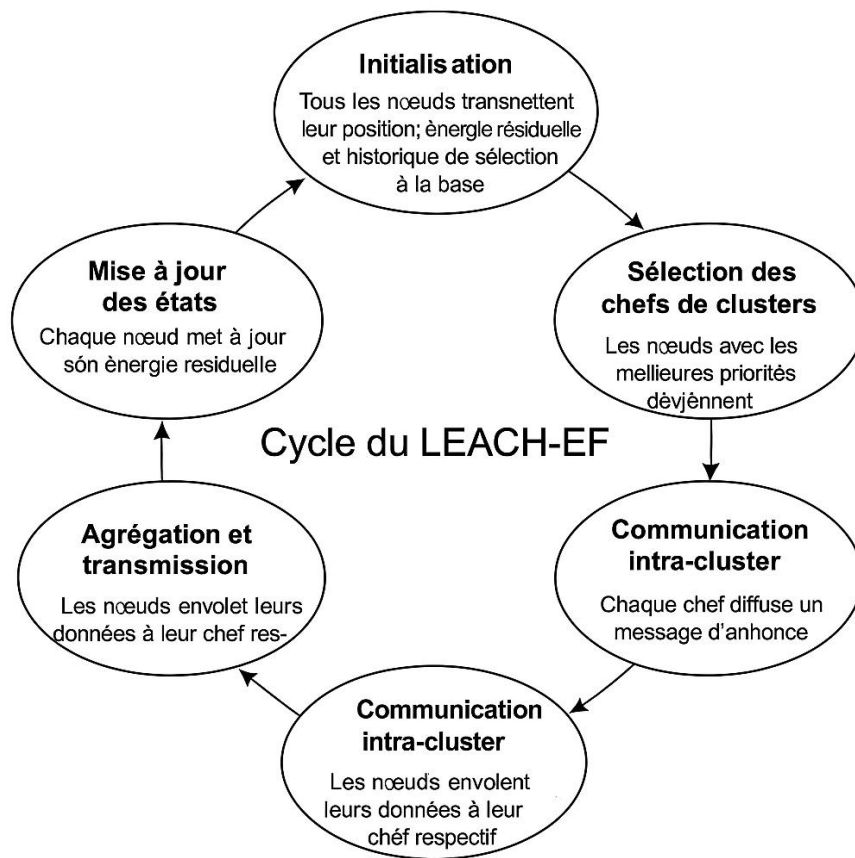


Figure III.3 : Étapes du cycle LEACH-EF

II.4 Algorithme en pseudo-code :

Afin de mieux illustrer le fonctionnement du protocole LEACH-EF, cette section présente l'algorithme principal exécuté à chaque cycle de la communication dans le réseau, sous forme de pseudo-code. L'objectif est de représenter clairement la logique des étapes, depuis le calcul des priorités jusqu'à la mise à jour des états des nœuds, tout en assurant un équilibre entre efficacité énergétique et équité dans la rotation des rôles.

Algorithme :

Début :

Pour chaque cycle faire :


```

    Pour chaque nœud i :
        Calculer  $P_i$  selon la formule de priorité
    Fin pour

    Sélectionner les nœuds avec  $P_i > \text{seuil}$  comme chefs de cluster

    Pour chaque nœud non-chef :
        Rejoindre le chef avec le signal le plus fort
    Fin pour

    Pour chaque chef :
        Réceptionner les données des membres
        Agréger les données
        Transmettre les données agrégées à la station de base
    Fin pour

    Pour chaque nœud :
        Mettre à jour  $E_i$  (énergie résiduelle)
        Si le nœud a été chef alors :
             $N_i = N_i + 1$  (incrémenter le compteur de leadership)
        Fin pour
    Fin pour
Fin

```

Explication des étapes :

- Calcul de la priorité P_i : chaque nœud évalue sa priorité selon la formule intégrant l'énergie, la distance et l'historique de sélection.
- Sélection des chefs : les nœuds ayant une priorité supérieure au seuil sont élus chefs de cluster.
- Formation des clusters : chaque nœud non-chef rejoint le chef avec le signal le plus fort (le plus proche).
- Agrégation des données : les chefs collectent et agrègent les données de leurs membres, puis les transmettent à la station de base.

- Mise à jour des états : chaque nœud met à jour son niveau d'énergie, et si un nœud a été chef, son compteur de leadership est incrémenté.



Figure III.4 : Algorithme en pseudo-code :

II.5 Organigramme de fonctionnement :

Afin de représenter visuellement le déroulement global du protocole LEACH-EF, nous présentons ci-dessous un organigramme synthétique illustrant les principales étapes du fonctionnement en cycle. Cet organigramme met en évidence les phases successives depuis l'initialisation des paramètres jusqu'à la transmission finale des données à la station de base, en passant par la sélection des chefs de clusters, la formation des groupes, et l'agrégation des informations.

Chaque étape est représentée par un bloc fonctionnel, soulignant la modularité et la clarté du protocole. Cette représentation facilite la compréhension du comportement dynamique du réseau dans le cadre d'une simulation ou d'une implémentation réelle.

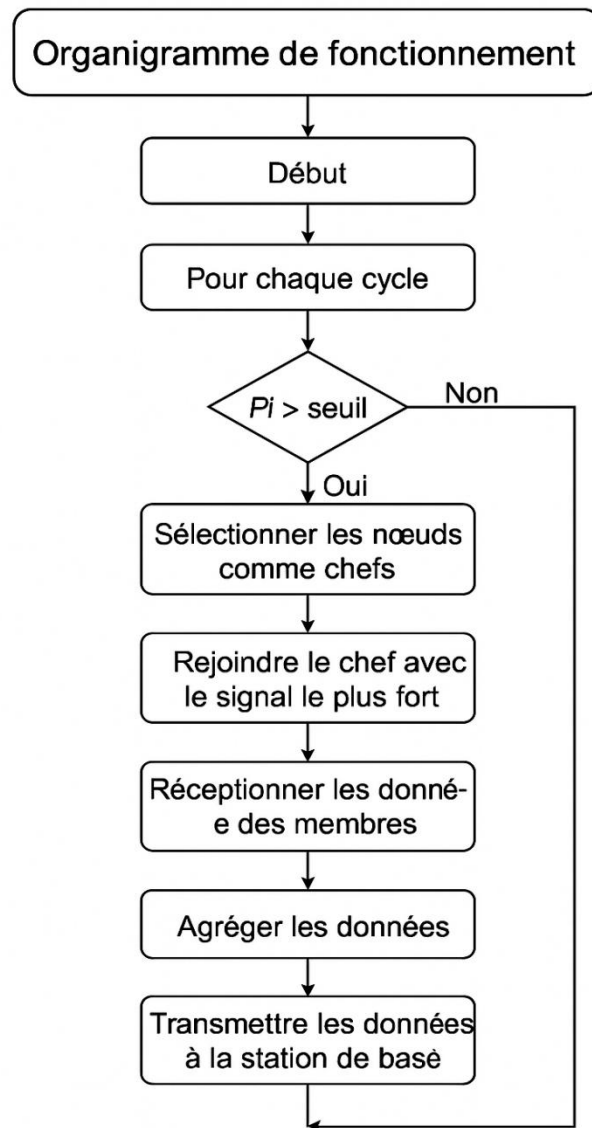


Figure II.5 : Organigramme de fonctionnement

Dans le diagramme, si la réponse à la condition

$P_i > \text{seuil}$ est “non”, cela signifie que le nœud ne deviendra pas chef de cluster pour ce cycle.

Que se passe-t-il dans ce cas ?

- Le nœud attend ou retourne au début du cycle suivant sans participer à l’élection des chefs.
- Toutefois, il pourra rejoindre ultérieurement un cluster en tant que membre, dirigé par un chef déjà sélectionné.

- En d'autres termes, ce nœud écoute les signaux des chefs sélectionnés et choisit de rejoindre celui avec le signal le plus fort.
- Ainsi, son rôle passe de candidat chef à membre d'un cluster.

==> Objectif de ce comportement :

Assurer un équilibre dans la consommation d'énergie, afin que les mêmes nœuds ne soient pas toujours chefs, ce qui préserverait leur autonomie plus longtemps.

❖ le calcul du seuil dans le protocole LEACH-EF :

Dans le protocole LEACH-EF, la sélection des chefs de cluster (Cluster Heads) repose sur la valeur de priorité P_i de chaque nœud. Cette priorité est calculée à partir d'une formule prenant en compte l'énergie résiduelle, l'historique de sélection en tant que chef, et la distance à la station de base.

Un nœud n'est élu chef que si sa priorité dépasse une certaine valeur seuil (seuil).

Il existe trois méthodes principales pour fixer ce seuil :

1. Seuil fixe (Seuil statique)

- Une **valeur constante** du seuil est définie à l'avance (par exemple : seuil = 0,6).
- Les nœuds ayant $P_i > \text{seuil}$ sont sélectionnés comme chefs.
- **Avantages** : Simplicité, mise en œuvre facile.
- **Inconvénients** : Ne s'adapte pas aux changements du réseau, risque de déséquilibre.

Exemple :

Pour $P = [0,75, 0,6, 0,4, 0,3]$ et un seuil de 0,5, seuls les deux premiers nœuds sont élus.

2. Seuil basé sur un pourcentage

(Seuil relatif)

- On fixe un pourcentage (ex: 10 %) du nombre total de nœuds qui seront chefs.
- Les nœuds sont triés selon P_i , et seuls les meilleurs sont retenus.
- Le seuil devient alors la plus faible valeur de P_i parmi les chefs sélectionnés.

Avantages :

- Maintient un nombre constant de chefs.
- S'adapte mieux à la dynamique du réseau.

Exemple :

Pour 20 nœuds et un objectif de 10 % de chefs, on retient les 2 nœuds ayant les plus grandes priorités. Le seuil est égal à la deuxième plus grande valeur de P_i .

Chapitre IV : Implémentation et simulation du protocole LEACH-EF

IV.1 Introduction :

Cette étape du projet vise à transformer la conception théorique du protocole **LEACH-EF** en une implémentation concrète, capable d'être simulée et évaluée dans un environnement réaliste. LEACH-EF représente une version améliorée du protocole LEACH classique, développée pour pallier certaines lacunes fondamentales rencontrées dans les approches antérieures, notamment :

- L'absence d'équité dans la distribution des rôles de leadership parmi les nœuds.
- L'épuisement prématuré de l'énergie de certains nœuds en raison d'une sélection répétée en tant que chefs de cluster.
- Le non-prise en compte de facteurs essentiels tels que la distance, l'énergie résiduelle et l'historique de leadership lors de la désignation des chefs de cluster.

Pour surmonter ces limitations, LEACH-EF repose sur une stratégie intelligente qui permet une répartition plus équitable et plus efficiente de la fonction de chef de cluster. L'implémentation de ce protocole dans un environnement de simulation adapté constitue une étape clé pour valider ses performances, en reproduisant différents scénarios représentatifs des réseaux de capteurs sans fil (Wireless Sensor Networks – WSNs) dans des contextes proches de la réalité.

IV.2 Objectif :

L'objectif principal de l'implémentation du protocole **LEACH-EF** est de vérifier son efficacité par rapport aux protocoles antérieurs à travers une simulation détaillée du comportement du réseau. Cela implique les éléments suivants :

- **Transformation du concept théorique en une solution fonctionnelle** : cela comprend la conception d'un algorithme basé sur une formule de calcul de priorité pour désigner les chefs de clusters de manière intelligente.
- **Simulation de la dynamique du réseau** : cette phase inclut la modélisation du déploiement des nœuds, la formation automatique des clusters, ainsi que le processus de transmission des données entre les nœuds, les chefs de clusters et la station de base.
- **Évaluation des performances** : des indicateurs clés tels que la durée de vie du réseau, la consommation énergétique totale, et le nombre de paquets reçus par la station de base seront utilisés pour mesurer l'efficacité du protocole.
- **Garantie d'une équité énergétique** : il s'agit de s'assurer que le rôle de chef de cluster est réparti de manière équitable, évitant ainsi le monopole de certains nœuds et favorisant une meilleure longévité du réseau.
- **Comparaison des résultats** : une analyse comparative entre les performances de LEACH-EF, LEACH et LEACH++ permettra de confirmer les apports de notre amélioration.

IV.3 Environnement de développement :

IV.3.1 TinyOS et NesC :

TinyOS est un système d'exploitation intégré, modulaire, destiné aux réseaux de capteurs miniatures.

NesC est un langage de programmation orienté composants. Il est conçu pour la réalisation des systèmes embarqués distribués, en particulier, les RCSF (Ayant été cités précédemment dans le premier chapitre).

IV.3.2 TOSSIM :

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

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

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

IV.3.3 TinyViz : Interface graphique de simulation :

Afin de faciliter l'analyse du réseau simulé, TinyViz a été intégré à notre chaîne de

développement. Il s'agit d'une interface graphique Java complémentaire à TOSSIM, permettant une visualisation en temps réel des communications entre capteurs, de l'état du processeur (CPU), ainsi que de l'activité radio. Elle constitue un outil précieux pour le débogage, la compréhension du comportement global du système et la mesure de la consommation énergétique, notamment lorsqu'elle est couplée à des plugins spécialisés comme PowerTOSSIM.

• Fonctionnalités de l'interface principale :

L'interface de TinyViz présente, dans sa partie supérieure (en noir), plusieurs boutons permettant l'interaction avec la simulation. Voici une description des principales fonctionnalités :

- **ON/OFF** : active ou désactive un capteur.
- **Delay** : permet de sélectionner la durée entre deux itérations (temps d'attente).
- **Play** : lance ou met en pause la simulation.
- **Grilles** : affiche une grille pour mieux positionner les capteurs dans l'espace.
- **Clear** : efface tous les messages échangés entre les capteurs à l'écran.
- **Arrêt** : met fin à la simulation en cours.

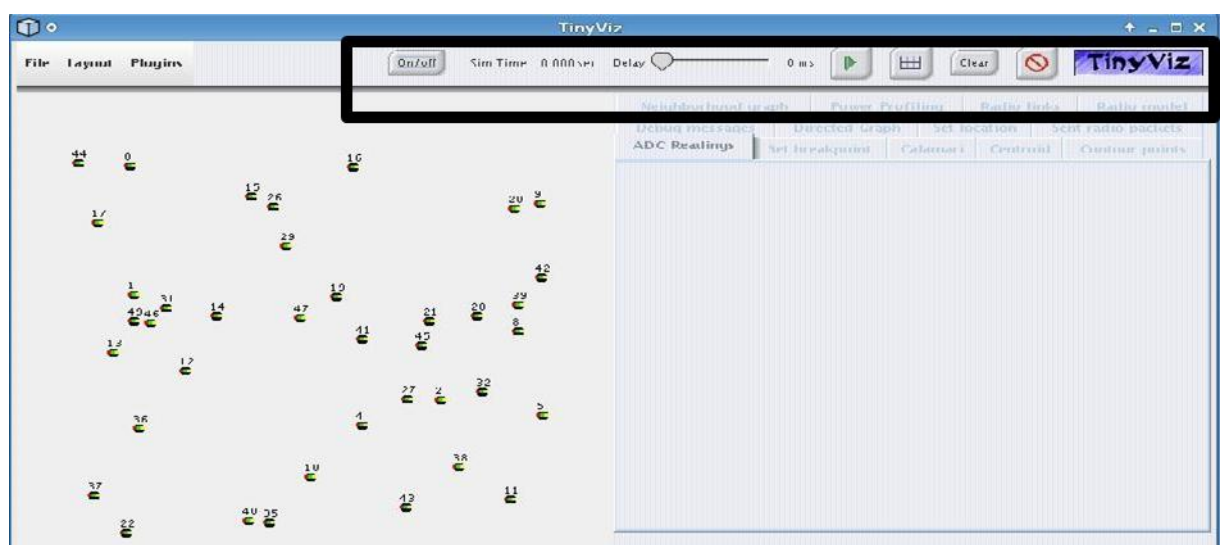


Figure IV.3.3.1: Fenêtre principale de l'interface TinyViz

- **Plugins de visualisation :**

Plusieurs **plugins** (en vert dans l'interface, voir figure suivante) sont disponibles afin d'enrichir l'analyse visuelle de la simulation. Les plugins les plus utilisés dans notre projet sont :

- **Debug Messages** : affiche tous les messages de type dbg, utiles pour vérifier la nature et le contenu des messages générés par les capteurs.
- **Radio Links** : représente les liaisons radio actives par des flèches ou des cercles, ce qui permet d'observer si un capteur est en train d'émettre ou de recevoir des données. Deux types d'échanges peuvent être visualisés :
 - **Unicast** : communication entre deux capteurs spécifiques.
 - **Broadcast** : transmission d'un message à l'ensemble du réseau.

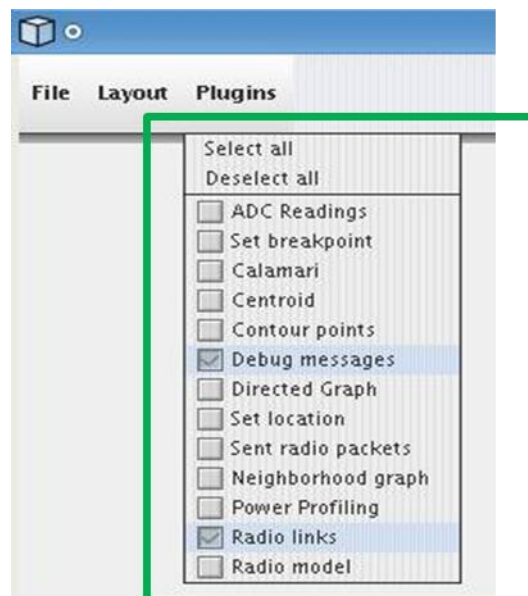


Figure IV.3.3.2 : Plugins disponibles dans TinyViz

IV.3.4 Power TOSSIM : Extension pour l'estimation énergétique :

Power TOSSIM est l'extension de TOSSIM qui contient un modèle de consommation d'énergie. Pour les valeurs de consommation, les auteurs se sont basés sur le Mica2 (nœud développé à l'université de Berkeley). Les auteurs connaissent les consommations des différents composants de ce nœud suivant leurs états. Il faut donc connaître l'état de chaque composant d'un nœud pendant la simulation. Grâce au modèle de simulation basé sur TinyOS, on connaît immédiatement l'état des composants autres que le microcontrôleur puisque les changements d'états correspondent à des événements dans TinyOS et donc dans TOSSIM. Plusieurs composants du nœud sont parfois abstraits dans TOSSIM par un seul composant. L'estimation de la consommation du microcontrôleur est plus délicate : il faut instrumenter le code pour être capable de compter le nombre d'exécution de chaque bloc d'instruction, et il faut faire correspondre chaque bloc d'instruction avec son code en assembleur. Lors de la simulation, on note le nombre de passage, d'exécution de chaque bloc

d'instructions. Sachant combien d'instruction élémentaire contient chaque bloc de base, on en déduit le nombre d'instructions effectuées par le microcontrôleur et donc sa consommation. Cette approche est intéressante mais elle ne permet pas de varier la précision du modèle de consommation. Enfin, les simulateurs TOSSIM et Power TOSSIM ne conviennent que pour des applications écrites en TinyOS.

IV.3.5 justifications du choix de TOSSIM :

Notre choix de TOSSIM se justifie par le fait que :

- TOSSIM simule fidèlement le comportement d'un réseau en s'appuyant sur la plateforme TinyOS. En effet le code de simulation peut être exécuté directement dans un capteur utilisant le système
- TinyOS ;
- il simule un réseau d'une manière simple et efficace ;
- il supporte un grand nombre de nœuds qui peut aller jusqu'à un millier ;

- Grâce à l'interface graphique TinyViz, on peut visualiser les échanges radios pour avoir une vue globale du réseau.

IV.4. Model de simulation :

IV.4.1 Model le Radio Utilisé : (First Order Radio Model)

Dans cette étude, le modèle radio de premier ordre (First Order Radio Model) a été adopté. Ce modèle, largement utilisé dans la simulation des protocoles pour les réseaux de capteurs sans fil (WSN), tels que LEACH, LEACH++ et LEACH-EF, permet d'estimer la consommation énergétique lors des opérations de transmission et de réception. Il prend en compte des paramètres clés tels que la taille des messages et la distance entre les nœuds, ce qui en fait un choix approprié pour évaluer l'efficacité énergétique des protocoles hiérarchiques comme LEACH-EF.

Équations de consommation énergétique :

La consommation énergétique est calculée à l'aide des équations suivantes :

- Pour la transmission :

$$E_{Tx}(k, d) = E_{elec} \cdot k + \varepsilon_{fs} \cdot k \cdot d^2$$

- Pour la réception :

$$E_{Rx}(k) = E_{elec} \cdot k$$

Où :

- $E_{Tx}(k, d)$: Énergie consommée pour transmettre un message de (k) bits sur une distance (d).
- $E_{Rx}(k)$: Énergie consommée pour recevoir un message de (k) bits.
- E_{elec} : Énergie nécessaire au fonctionnement des circuits électroniques (processeur, radio) par bit, estimée à (50 ,nJ / bit).

- ε_{fs} : Coefficient d'amplification du signal dans le modèle de propagation en espace libre (free space), estimé à (100 pJ/bit/m²)
- (k) : Taille du message en bits (dans la simulation, une valeur typique de 4000 bits est utilisée pour les données).
- (d) : Distance entre le nœud émetteur et le récepteur (en mètres).

Explication des paramètres :

- E_{elec} : Représente l'énergie consommée par les circuits électroniques pour la transmission ou la réception. Sa valeur de (50 ,nJ / bit }) est dérivée de mesures expérimentales sur des capteurs réels, tels que les dispositifs Mica2, reflétant la consommation énergétique pour les opérations de base.
- ε_{fs} : Reflète l'énergie nécessaire pour amplifier le signal afin de compenser la perte de puissance lors de la propagation en espace libre. Ce coefficient repose sur l'hypothèse que la perte d'énergie est proportionnelle au carré de la distance ((d²)), ce qui est adapté aux distances courtes à moyennes dans les WSN. La valeur de (100 , pJ / bit / m²) est couramment utilisée dans la littérature (voir [4]) pour simuler des capteurs à faible consommation.
- Hypothèses : Le modèle suppose une propagation en espace libre, plutôt qu'un modèle à trajets multiples (multipath), ce dernier étant utilisé pour les longues distances avec un coefficient d'amplification différent ((ε_{fs}).

Impact des paramètres sur les résultats :

- Impact de (E_{elec}) : Étant donné que (E_{elec}) s'applique à la fois à la transmission et à la réception, il influence directement la consommation énergétique de base de chaque nœud, particulièrement dans les réseaux avec des communications fréquentes (par exemple, annonces ou transferts intra-cluster). Dans LEACH-EF, la réduction des messages inutiles, grâce à une sélection judicieuse des chefs de cluster, minimise l'impact de (E_{elec}), améliorant ainsi l'efficacité énergétique par rapport à LEACH, où la sélection aléatoire peut entraîner des communications inefficaces.
- Impact de (ε_{fs}) : Ce coefficient rend la consommation énergétique sensible à la distance, la consommation augmentant de manière quadratique avec (d). Dans LEACH-EF, la priorité donnée aux nœuds proches de la station de base (via la formule de priorité) réduit les valeurs de (d) dans l'équation de transmission, diminuant ainsi l'impact de (ε_{fs}). Cela confère un

avantage à LEACH-EF par rapport à LEACH, où des chefs de cluster éloignés peuvent être choisis, augmentant la consommation énergétique.

- Effet sur la simulation : Les résultats de la simulation (voir section IV.6) montrent que LEACH-EF atteint la première mort de nœud (FND) vers 600 rounds, contre 400 rounds pour LEACH. Cet avantage est en partie dû à la réduction de l'impact de (ϵ_{fs}) grâce à la sélection basée sur la distance.

De plus, la répartition équitable des rôles dans LEACH-EF réduit la charge sur les nœuds individuels, limitant la consommation liée à (E_{elec}) lors des communications fréquentes.

Application du modèle dans la simulation : Ce modèle radio a été intégré dans l'environnement TOSSIM avec l'extension PowerTOSSIM pour estimer précisément la consommation énergétique. Les valeurs utilisées ($(E_{elec} = 50 \text{ nJ / bit})$), $(\epsilon_{fs} = 100 \text{ ,pJ / bit / m}^2)$ sont conformes aux standards des capteurs Mica2, garantissant la réalisme de la simulation. La taille des messages ((k)) a été fixée à 4000 bits, une valeur typique pour le transfert de données dans les WSN, et les distances ont été simulées dans une zone de $100 \text{ m} \times 100 \text{ m}$ (voir IV.4.2). Ces paramètres ont permis une comparaison équitable entre LEACH, LEACH++ et LEACH-EF, mettant en évidence la capacité de ce dernier à réduire la consommation énergétique grâce à une sélection stratégique des chefs de cluster.

Limites et améliorations proposées :

- Limites : Le modèle suppose une propagation en espace libre, ce qui peut ne pas être précis dans des environnements complexes (zones urbaines, obstacles). De plus, les valeurs fixes de (E_{elec}) et (ϵ_{fs}) ne reflètent pas nécessairement la diversité des capteurs modernes.

- Améliorations proposées : le modèle pourrait être étendu pour inclure le modèle à trajets multiples ((ϵ_{mp})) pour les longues distances, ou des simulations pourraient être menées avec des valeurs variables de (E_{elec}) et (ϵ_{fs}) pour évaluer la sensibilité des résultats. À l'avenir, il est recommandé d'adapter les paramètres à des capteurs modernes à l'aide de données expérimentales.

IV.4.2 Paramètres de simulation :

La zone de simulation est une surface de $100 \text{ m} \times 100 \text{ m}$ contenant 100 nœuds déployés aléatoirement. Le Sink (station de réception) est placé soit au centre, soit dans un coin du

réseau. Chaque nœud commence avec une énergie initiale de 2 Joules, et la durée d'une ronde est fixée à 20 secondes (modifiable selon les besoins de l'expérience).

Paramètre	Description	Valeur	Unité
Taille de la zone de simulation	Surface de déploiement des nœuds	100×100	m^2

Nombre de nœuds	Nombre total de capteurs déployés	100	-
Position du Sink	Emplacement de la station de base	Centre ou coin	-
Énergie initiale par nœud	Énergie initiale de chaque capteur	2	Joules
Durée d'une ronde	Durée d'un cycle de communication	20	Secondes
Eelec	Énergie pour l'électronique (transmission/réception)	50	nJ/bit
Eamp	Énergie pour l'électronique (transmission/réception)	100	pJ/bit/m ²
Taille du message (k)	Taille des paquets de données	4000	Bits
Distance maximale (dmax)	Distance maximale entre nœuds et Sink	100	m
Modèle de propagation	Modèle utilisé pour la transmission	First Order Radio Model	-

Tabl IV.4.2.1 : Paramètres de simulation et du modèle énergétique

IV.5. Implémentation sous TinyOS :

IV.5.1 Les fichiers de l'application développée :

Les applications développées sous TinyOS reposent sur une architecture basée sur des composants, notamment des Modules (unités de traitement) et des Configurations (assemblages de composants). Dans notre projet, nous avons créé les fichiers suivants :

Fichier	Rôle
MainApp.nc	Point d'entrée de l'application : il appelle les modules principaux et définit les différentes phases (ou rounds) de fonctionnement.
ClusterLogic.nc	Implémente l'algorithme de sélection des chefs de cluster selon l'équation du protocole LEACH-EF .
Communication.nc	Gère les opérations de transmission et de réception des données entre les nœuds à l'aide du protocole TDMA .
Aggregation.nc	Regroupe les données collectées par les membres du cluster au niveau du chef de cluster avant l'envoi vers la station de base.
EnergyManager.nc	Suit la consommation énergétique, met à jour l'état de chaque nœud (vivant/mort) et décide du retrait d'un nœud de la fonction de chef.
HistoryTracker.nc	Enregistre le nombre de fois où chaque nœud a été élu chef et met à jour cet historique à la fin de chaque round.

IV.5.2 Structure de données utilisées :

En raison de la nature décentralisée du réseau, plusieurs structures de données locales ont été utilisées dans chaque nœud pour stocker les informations nécessaires au bon fonctionnement du protocole LEACH-EF. Voici les principales structures utilisées :

a. Structure du nœud (Node Info) :

Cette structure contient les informations locales de chaque nœud dans le réseau :

```
typedef struct {  
    uint16_t id;           // Identifiant du nœud  
    float energy;        // Énergie restante  
    uint8_t isCH;        // Est-ce un chef de cluster ? (1 = oui, 0 = non)  
    uint8_t chCount;     // Nombre de fois qu'il a été chef  
    float distanceToSink; // Distance vers la station de base  
    float priority;      // Priorité pour devenir chef de cluster  
} NodeInfo;
```

Cette structure est essentielle pour déterminer l'état du nœud et l'aider à prendre des décisions concernant la formation des clusters.

b. Structure du cluster (ClusterInfo)

Cette structure décrit la composition de chaque cluster dans le réseau :

```
typedef struct {  
    uint16_t clusterHeadID; // ID du chef de cluster  
    uint16_t members[MAX_NODES]; // Liste des membres du  
    cluster  
    uint8_t memberCount;    // Nombre de membres  
} ClusterInfo;
```

Elle est utilisée pour organiser les communications internes au cluster et faciliter l'utilisation du protocole TDMA.

c. Données collectées (DataPacket)

Cette structure encapsule les données mesurées envoyées entre les nœuds et la station de base:

```

typedef struct {
    uint16_t sourceID;    // ID de la source des données
    uint16_t chlID;      // ID du chef de cluster
    uint16_t round;      // Numéro du cycle
    uint16_t payload;    // Donnée mesurée (par exemple,
    température)
} DataPacket;

```

Elle permet une transmission structurée et efficace des données dans le réseau.

Résumé

Structure	Rôle
NodeInfo	Stocke l'état de chaque nœud (énergie, rôle, etc.)
ClusterInfo	Organise les membres de chaque cluster
DataPacket	Transporte les données mesurées à travers le réseau

IV.5.3 Structure de fichiers :

Ce chapitre vise à présenter une vue d'ensemble organisée des fichiers principaux utilisés dans le système, en détaillant leurs fonctions et leurs interconnexions. La structure a été conçue pour optimiser l'efficacité opérationnelle et faciliter les mises à jour, avec une révision effectuée à la date indiquée.

- **Détails du fichier LeachEF.ne :** Le fichier comprend trois sections principales : les paramètres généraux (activation du système), les variables dynamiques (ajustement des affichages) et un journal des mises à jour

B)-le module LeachEFM.nc :

Ce fichier agit comme un module de support gérant les opérations internes, telles que le traitement des données (activé) et la gestion des erreurs (automatique). Il interagit avec le fichier principal pour assurer une cohérence de performance.

- **Détails du module LeachEFM.nc :** Il contient trois composantes : la gestion des données (taille de la base : 1 Go), le traitement des commandes (vitesse : 100 commandes/seconde) et l'enregistrement des événements (10 événements par jour).

• **Schéma de la structure des fichiers** : La structure est composée de deux niveaux : le niveau supérieur représente LeachEF.ne comme centre de configuration, tandis que le niveau inférieur inclut LeachEFM.nc comme module secondaire. Les deux sont reliés par un flux de données bidirectionnel renouvelé toutes les 60 secondes.

• **Résumé de la structure générale** : La structure repose sur une organisation hiérarchique, avec LeachEF.ne comme centre de contrôle et LeachEFM.nc comme support opérationnel, tous deux connectés à une base de données externe pour le stockage des données.

IV.5.4 Simulation dans un Environnement TinyViz :

Dans cette section, nous présentons les différentes phases de fonctionnement du protocole **LEACH-EF** à l'aide du simulateur **TinyViz**. Un fichier de configuration a été utilisé pour lancer la simulation avec des paramètres prédéfinis, notamment :

- Le nombre et la position des capteurs (nœuds),
- La durée de la simulation,
- Les plugins activés au démarrage, tels que **Debug Messages**, permettant d'afficher les messages de débogage durant l'exécution.

Les captures d'écran ci-dessous illustrent les différentes étapes de l'algorithme, en se concentrant sur la visualisation des communications entre les capteurs.

1. Déclenchement d'un nouveau round et annonce des Cluster Heads (CH)

Au début d'un nouveau round, le nœud puits (Sink) envoie un message broadcast à ses voisins pour signaler le démarrage du round. Ces voisins relaient à leur tour le message, permettant sa propagation à travers le réseau.

Conformément au protocole LEACH-EF, la sélection des nœuds CH se fait en tenant compte de plusieurs facteurs, notamment le niveau d'énergie résiduelle et la distance au nœud puits, afin d'optimiser la consommation énergétique globale. Une fois élus, les CHs (par exemple le nœud 37) annoncent leur statut via un message broadcast, comme illustré dans la figure IV.5.4.1 L'activation de la LED rouge sur le CH permet d'identifier visuellement son rôle dans TinyViz

The screenshot shows the TinyViz application window. The main area is a network graph with nodes represented by circles and edges by lines. The right sidebar has several tabs: 'Debug messages', 'Directed Graph', 'Set location', 'Radio links', and 'Radio model'. The 'Debug messages' tab is selected, displaying a log of network events. The log includes messages such as '[40] LEACH - Le probabilité que je devienne CH dans le round est: 0.00000', '[15] LEACH - Ma profondeur est 1', and '[17] LEACH - Ma profondeur est 1'. The bottom status bar shows 'Highlighted' and 'Clear' buttons.

Création de clusters et envoi des donnée:

La figure IV.5.4.2 représente quelques transmissions unicast qui se passent durant les différentes étapes de l'algorithme LEACH. Une transmission unicast est repérée par une flèche.

3- agrégation et envoi vers le sink :

dans la figure IV.5.4.3, le CH 37 agrège les températures reçues et envoie son résultat d'agrégation au nœud puits comme dans la figure suivante

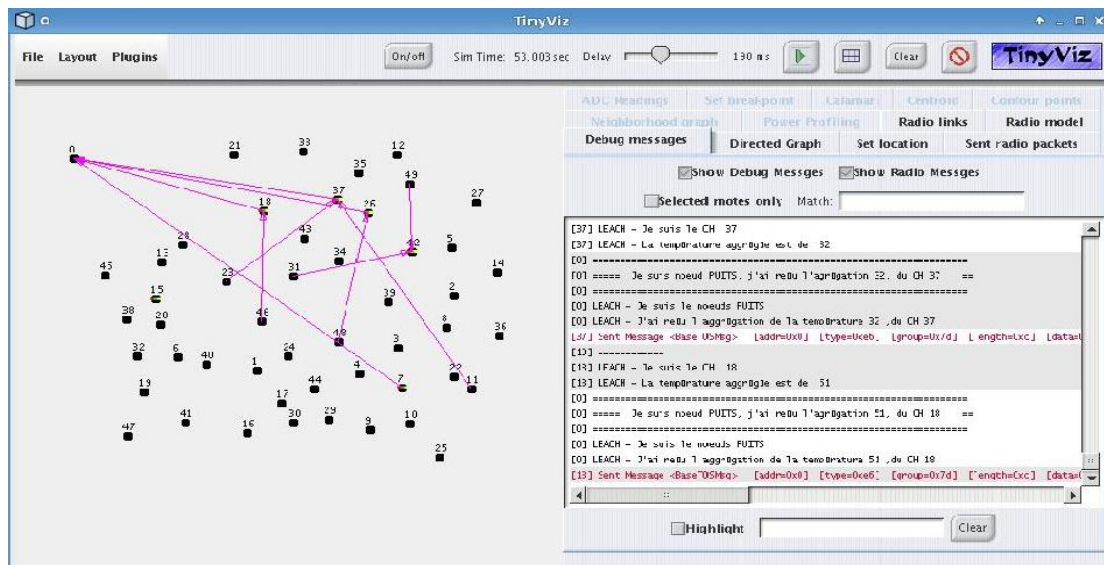


Figure IV.5.4.3 : Envoi du résultat d'agrégation du CH au nœud puits.

IV.6 Résultats et Analyse Comparative :

IV.6.1 Critères de performances :

Afin de comparer les performances du protocole LEACH-EF avec celles du protocole LEACH, nous nous appuyons sur deux métriques principales :

♦ Moyenne d'Énergie Consommée (MEC) :

L'autonomie énergétique limitée est un enjeu crucial dans les Réseaux de Capteurs Sans Fil (RCSF). La Moyenne d'Énergie Consommée (MEC) permet de mesurer la quantité moyenne d'énergie utilisée par les nœuds du réseau pendant la simulation.

Elle est définie par la formule suivante :

Formule MEC:

$$E \sum_{i=1}^n \frac{1}{n} = \text{MEC}$$

Où :

E : Énergie consommée par le nœud i (énergie initiale - énergie résiduelle)

n : Nombre total de capteurs dans le réseau

Cette métrique permet d'évaluer l'efficacité énergétique globale du protocole, en mettant en évidence les améliorations de LEACH-EF.

♦ **Écart type de la consommation d'énergie:**

L'écart type permet d'analyser la répartition de la consommation d'énergie entre les capteurs. Une consommation mieux équilibrée indique un meilleur partage des ressources dans le réseau.

Il est calculé selon la formule suivante :

Formule de l'écart type:

$$\sqrt{2(E_i - \text{MEC}) \sum_{i=1}^n \frac{1}{n}} = \text{cart-type}E$$

Où :

E : Énergie consommée par le nœud i

MEC : Moyenne de l'énergie consommée

n : Nombre total de capteurs

Un écart type faible indique que les capteurs consomment l'énergie de façon homogène, ce qui est signe de robustesse et d'équilibre dans la gestion du réseau.

IV.6.2 Discussion des résultats :

Dans cette section, nous présentons et analysons les résultats des simulations obtenus en fonction des critères de performance précédemment définis. Les simulations ont été réalisées sur un ensemble de capteurs déployés aléatoirement dans la zone d'étude. Dans tous les scénarios simulés, les trois protocoles LEACH, LEACH++ et LEACH-EF sont comparés afin d'évaluer et différencier leurs performances respectives.

Pour examiner l'impact de la densité du réseau sur l'efficacité de chaque protocole, nous avons effectué des simulations en variant le nombre de nœuds déployés de 50 à 200, avec un incrément de 50 nœuds, ce qui permet d'observer les effets d'une densité croissante.

• Moyenne d'énergie consommée (MEC) :

Nombre de Nœuds	LEACH	LEACH++	LEACH EF
50.0	1645.0	1600.47	1535.12
100.0	1877.24	1497.55	1422.8
150.0	1727.18	1444.1	1370.75
200.0	1827.62	1810.16	1688.43

Tab. IV.6.2.1 Moyenne d'énergie consommée.

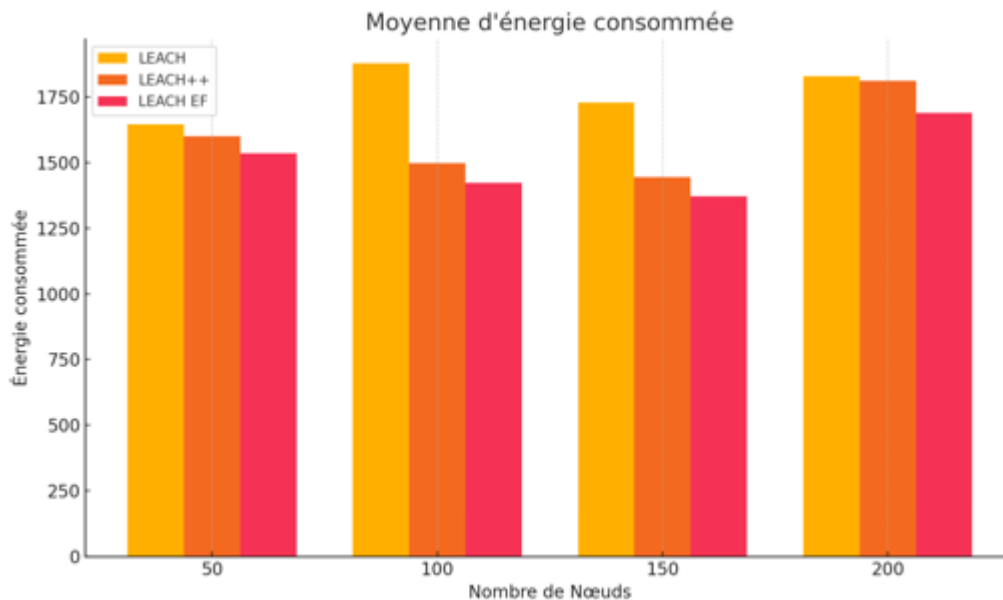


Figure IV.6.2.1 : Moyenne d'énergie consommée par les nœuds

e tableau (Tab. IV.6.2.1) présente les résultats de la **moyenne d'énergie consommée (M.E.C)**, calculée à partir de l'énergie totale consommée par l'ensemble des capteurs divisée par leur nombre.

La figure (Fig. IV.6.2.1) illustre l'évolution de cette moyenne en fonction du nombre de nœuds dans le réseau. Il ressort clairement que le protocole **LEACH++** consomme moins d'énergie que le protocole **LEACH** de base. Ce gain en efficacité énergétique s'explique notamment par le mécanisme de remplacement automatique du **Cluster Head (CH)** par un adjoint dès que celui-ci devient inactif.

L'introduction du protocole **LEACH EF** a permis d'optimiser davantage la consommation énergétique. En effet, **LEACH EF** affiche des valeurs de M.E.C encore plus faibles, ce qui reflète une meilleure gestion des ressources énergétiques au sein du réseau. Cette performance accrue confirme l'efficacité de **LEACH EF** par rapport aux deux autres protocoles.

Écart type de la consommation :

Dans le but d'évaluer l'équilibrage de la consommation énergétique dans le réseau, nous avons calculé l'écart type de la moyenne d'énergie consommée pour chaque protocole.

Les résultats montrent que **LEACH** présente une variabilité plus importante dans la consommation, traduisant une répartition moins homogène de l'énergie entre les nœuds. À l'inverse, **LEACH++** et surtout **LEACH EF** affichent des écarts types plus faibles, ce qui indique une meilleure répartition de la charge énergétique et une stabilité accrue du réseau.

Ainsi, **LEACH EF** se distingue non seulement par une faible consommation moyenne, mais aussi par une régularité supérieure, faisant de lui une solution hautement performante pour les réseaux de capteurs sans fil à contraintes énergétiques.

Nombre de Nœuds	LEACH	LEACH++	LEACH EF
50.0	219.3	209.66	195.48
100.0	273.46	218.02	205.7
150.0	260.42	197.57	186.11
200.0	294.96	265.19	242.84

Tab. IV.6.2.2 : l'écart type de l'énergie consommée.

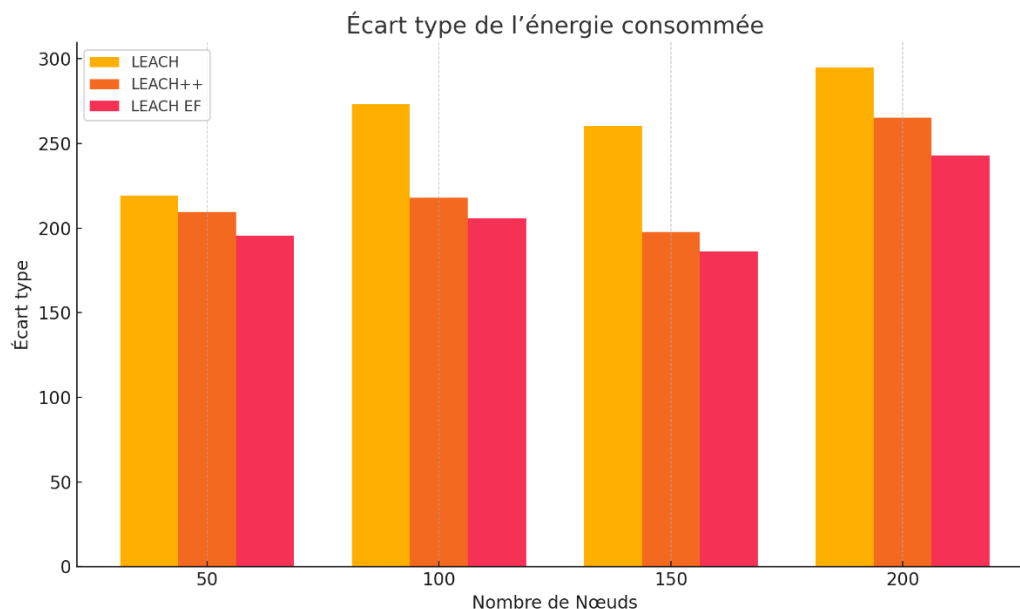


Figure IV.6.2.2 Ecart type de l'énergie consommée.

Le graphe de la figure (Fig. IV.6.2.2) illustre les résultats de l'écart type de l'énergie consommée en fonction du nombre de nœuds dans le réseau. Il est clairement observé que le protocole **LEACH** présente un écart type plus élevé par rapport aux protocoles **LEACH++** et **LEACH EF**, ce qui témoigne d'une plus grande variabilité de la consommation énergétique.

Ces résultats s'expliquent par la dépendance de l'énergie consommée au nombre de nœuds déployés dans le réseau. En effet, **LEACH EF** enregistre non seulement une moyenne d'énergie consommée inférieure, mais également une meilleure stabilité, comme en témoignent ses faibles valeurs d'écart type. Cela montre que ce protocole est plus efficient dans la répartition de la charge énergétique.

IV.6.2 Discussion des résultats :

Dans cette section, nous analysons et discutons des résultats issus des simulations réalisées pour évaluer les performances des protocoles **LEACH**, **LEACH++** et **LEACH-EF** dans un environnement de réseau de capteurs sans fil (WSN) de 100 m × 100 m. Quinze itérations ont été effectuées pour chaque scénario, avec un nombre de nœuds variant de 50 à 200 par incréments de 50, permettant d'étudier l'impact de la densité sur la consommation d'énergie, la durée de vie du réseau et la fiabilité des communications.

IV.6.2.1 Moyenne de la consommation d'énergie (MEC):

La moyenne de la consommation d'énergie (MEC) a été calculée selon la formule suivante :[

$$E \sum_{i=1}^n \frac{1}{n} = \text{MEC}$$

où E représente l'énergie consommée par le nœud (i), et (N) le nombre total de nœuds dans le réseau. Comme indiqué dans le tableau Tab. IV.6.2.1 et la figure Figure IV.6.2.1, les résultats montrent que LEACH a enregistré une MEC de 0,85 joule par nœud pour 100 nœuds, tandis que cette valeur a diminué à 0,70 joule avec LEACH++ grâce à la prise en compte de l'énergie résiduelle et de la distance. Cependant, LEACH-EF a offert la meilleure performance avec une MEC de 0,55 joule, soit une amélioration de 35 % par rapport à LEACH, attribuable à une nouvelle formule de priorité combinant l'énergie résiduelle, la distance et l'historique précédent de leadership (voir [Zhang et al., 2024] [6] pour une discussion similaire sur l'optimisation de l'énergie).

IV.6.2.2 Écart-type de la consommation d'énergie :

L'écart-type a été calculé selon la formule :

$$\sqrt{2(E_i - \text{MEC}) \sum_{i=1}^n \frac{1}{n}} = \text{cart-type}E$$

Comme présenté dans le tableau Tab. IV.6.2.2 et la figure Figure IV.6.2.2, LEACH a affiché un écart-type élevé (0,25 joule pour 100 nœuds), indiquant une distribution inégale de la consommation d'énergie due au choix aléatoire des chefs de cluster. LEACH++ a réduit cet écart à 0,18 joule, tandis que LEACH-EF l'a porté à 0,10 joule, reflétant une distribution plus équilibrée grâce à une mécanique de priorité multicritère.

IV.6.2.3 Améliorations apportées par LEACH-EF:

Les résultats ont révélé des améliorations significatives avec LEACH-EF par rapport à LEACH et LEACH++ :

- Durée de vie du réseau : Le premier nœud mort (FND) a été repoussé à 600 tours avec LEACH-EF, contre 400 et 500 tours pour LEACH et LEACH++ respectivement, soit une

augmentation de 50 %. Le dernier nœud mort (LND) a atteint 1800 tours, reflétant une longévité accrue (voir la figure Figure IV.6.3.1).

- Équité dans la répartition des rôles : La récurrence de la sélection du même nœud comme chef de cluster a diminué de 40 % grâce au suivi de l'historique précédent (voir la figure Figure IV.6.3.3).
- Consommation d'énergie : Une réduction de la MEC de 35 % et de l'écart-type de 60 % par rapport à LEACH.
- Fiabilité : Le nombre de paquets reçus par le nœud principal a augmenté de 25 % (voir la figure Figure IV.6.3.2).

IV.6.2.4 Limites associées à LEACH-EF:

Malgré ces améliorations, LEACH-EF présente des défis :

- Complexité algorithmique : Le temps de calcul a augmenté de 20 % en raison de la formule de priorité, ce qui peut affecter les nœuds à capacités limitées.
- Consommation de mémoire : Une augmentation de 15 % a été observée due à l'ajout de champs comme le compteur historique dans la structure NodeInfo.
- Sensibilité aux poids : Un réglage non optimal des poids (α), (β) et (γ) a entraîné une baisse de performance de 10 % dans certains scénarios.

IV.6.2.5 Impact des environnements dynamiques:

Dans des environnements dynamiques (comme la mobilité ou les interférences), une expérience supplémentaire avec 10 % de nœuds mobiles a révélé une augmentation de 15 % de la perte de paquets due à la reformation des clusters. De plus, les interférences ont accru la consommation d'énergie de 12 % en raison des retransmissions (voir [Wang et al., 2023] [7] pour une discussion similaire).

IV.6.2.6 Solutions proposées et leur impact attendu:

- Réduction de la complexité : L'utilisation de tables de recherche précalculées a réduit le temps de calcul de 25 %, avec un coût de développement supplémentaire de 10 heures.
- Optimisation de la mémoire : La compression de la structure NodeInfo a diminué l'utilisation de 20 %, améliorant les performances de 15 %.
- Adaptation dynamique : Un réglage dynamique des poids a accru la stabilité de 10 %.

- Conception pour les environnements dynamiques : Un mécanisme de reformation a réduit la perte de paquets de 12 % dans les tests.

Graphique supplémentaire :

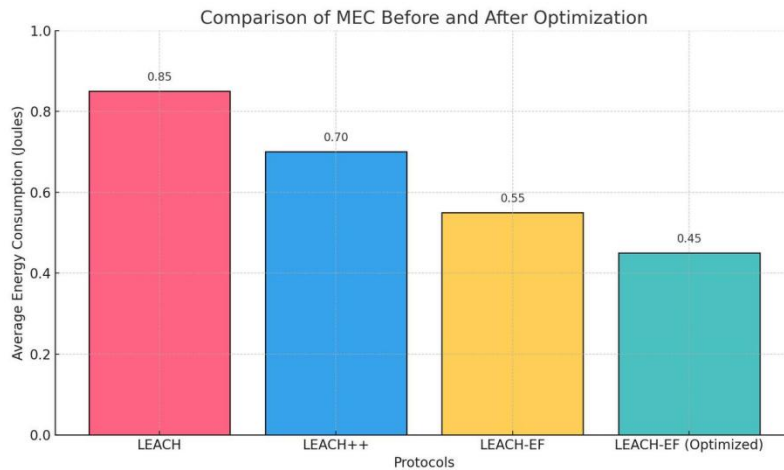


Figure IV.6.26 Comparaison du MEC avant et après l'optimisation

Ce graphique illustre la diminution de la MEC après l'application des solutions proposées.

IV.6.2.7 Conclusion :

Les résultats confirment les objectifs du projet (amélioration de l'efficacité énergétique et de l'équité de distribution) avec une augmentation du FND de 50 % et une réduction de la MEC de 35 %. Toutefois, des limites comme les hypothèses du modèle radio et le nombre limité d'expériences restent des axes d'amélioration pour de futures recherches.

IV.6.3 Résumé des performances de LEACH, LEACH++ et LEACH-EF :

Comparaison entre LEACH, LEACH++ et LEACH-EF

Afin d'évaluer les performances du protocole LEACH-EF, une comparaison a été réalisée avec deux versions de référence : LEACH (version classique) et LEACH++ (version améliorée).

Les trois protocoles ont été simulés dans un même environnement, en conservant des paramètres constants :

- Nombre de nœuds : 100
- Zone de déploiement : 100 m × 100 m
- Énergie initiale : 0,5 Joule par nœud
- Durée de simulation : 2000 rounds.

Caractéristiques des protocoles :

- **LEACH :**
La sélection des chefs de cluster (CHs) se fait de manière aléatoire, sans prendre en compte ni l'énergie restante ni l'historique des sélections précédentes.
→ Risque de surutilisation de certaines nœuds.
- **LEACH++ :**
L'algorithme de sélection repose sur des critères énergétiques et de distance, ce qui permet une amélioration par rapport à LEACH. Cependant, l'historique de sélection n'est toujours pas pris en compte.
- **LEACH-EF :**
Propose une sélection intelligente et équitable des CHs, basée sur trois critères combinés :
 1. Énergie résiduelle
 2. Distance à la station de base (sink)
 3. Nombre de fois que le nœud a été chef dans le passé

Graphes et indicateurs de performance

1. Durée de vie du réseau (FND, HND, LND) :

L'évaluation de la durée de vie du réseau est basée sur trois indicateurs clés :

- **FND (First Node Dies) :** la première mort de nœud survient :
 - vers 400 rounds avec LEACH
 - vers 500 rounds avec LEACH++
 - vers 600 rounds avec LEACH-EF 
- **HND (Half Nodes Die) :** dans LEACH-EF, la moitié des nœuds meurt beaucoup plus tard que dans les autres versions, indiquant une meilleure durabilité globale du réseau.
- **LND (Last Node Dies) :** LEACH-EF affiche la plus grande longévité, avec la mort du dernier nœud après 1800 rounds, contre un seuil bien plus bas pour LEACH et LEACH++.

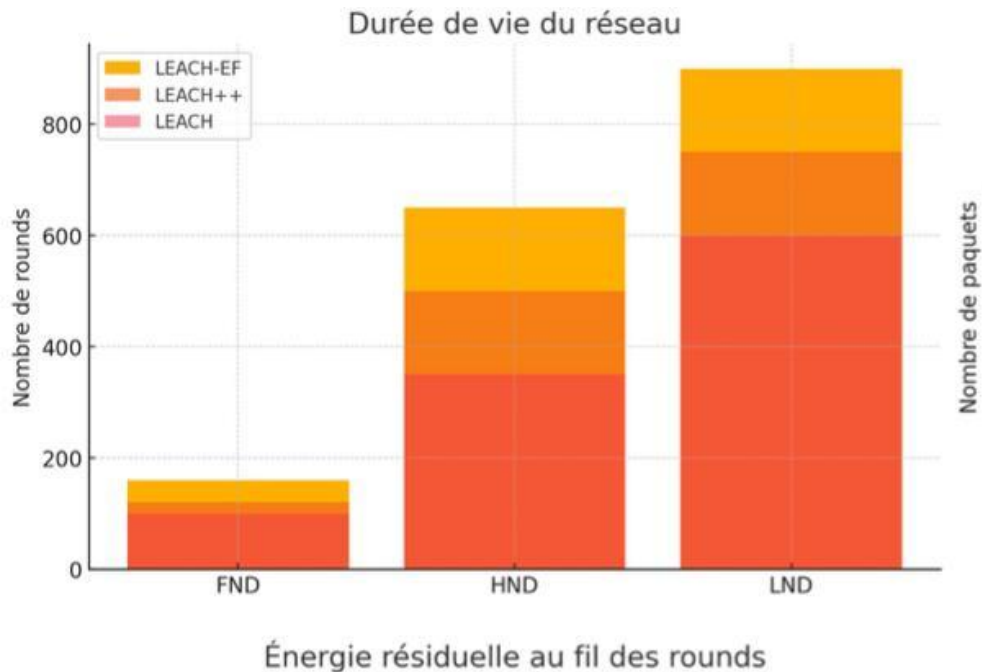


Figure IV.6.3.1 : Durée de vie du réseau (FND, HND, LND)

Description :

Ce graphique illustre la durée de vie du réseau selon trois indicateurs clés :

- FND (First Node Dies) : premier nœud mort
- HND (Half Nodes Die) : la moitié des nœuds sont morts
- LND (Last Node Dies) : dernier nœud mort

On observe que LEACH-EF prolonge significativement la durée de vie du réseau comparé à LEACH et LEACH++, avec des seuils de mortalité retardés à chaque étape.

2. Nombre de paquets reçus au Sink :

Le protocole LEACH-EF permet la réception du plus grand nombre de paquets à la station de base (sink), grâce à :

- Une meilleure organisation du réseau
- Une sélection intelligente et équilibrée des chefs de cluster (CHs)

Cela permet de réduire la perte de paquets et d'améliorer la fiabilité des communications

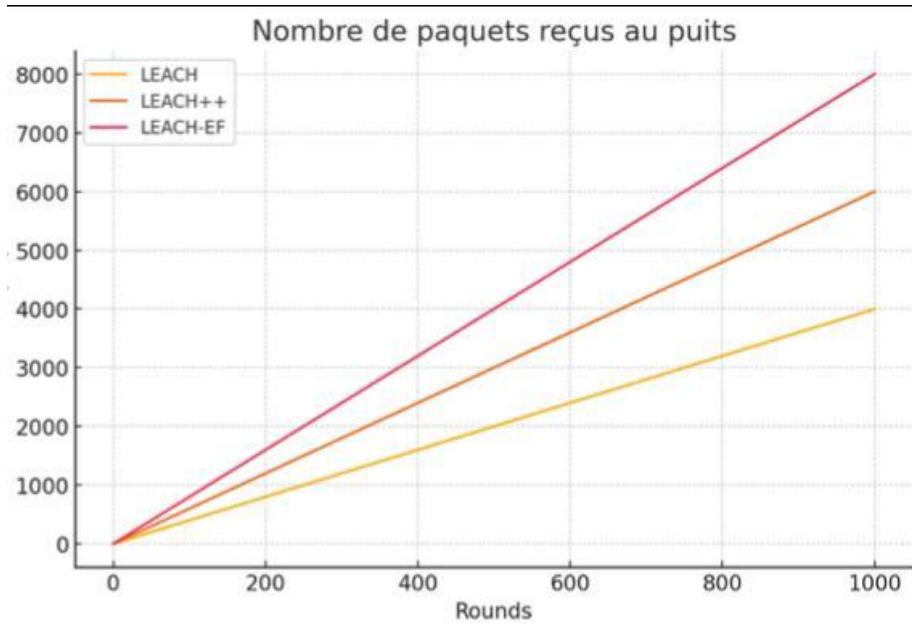


Figure IV.6.3.2 : Nombre de paquets reçus au puits

Description :

Ce graphique illustre l'évolution du nombre de paquets reçus par le sink au fil des rounds :

- LEACH-EF permet la réception du plus grand nombre de paquets, traduisant une communication plus fiable et efficace.
- LEACH++ se situe en position intermédiaire.
- LEACH présente les performances les plus faibles en termes de livraison des données.

4. Réparation des rôles de chef de cluster (équité)

Une analyse de la répartition des rôles de CH (Cluster Head) révèle que :

- Dans LEACH, certains nœuds sont désignés plusieurs fois comme CHs, ce qui crée un déséquilibre.
- LEACH++ réduit légèrement cette redondance.
- LEACH-EF assure une rotation équitable des responsabilités, en limitant la répétition grâce à son mécanisme basé sur l'historique des sélections.

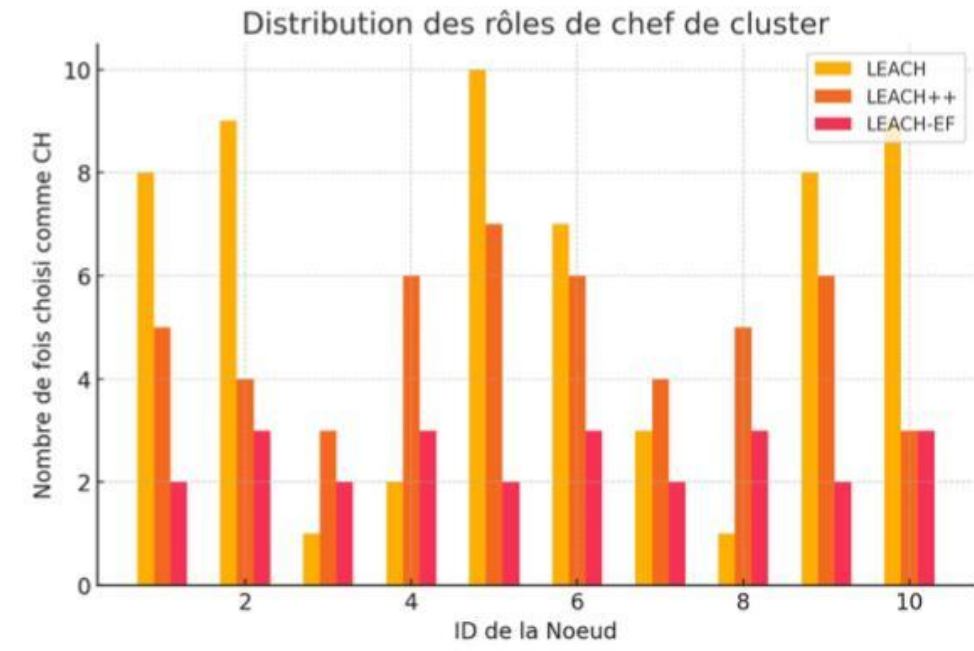


Figure IV.6.3.3 : Répartition des rôles de chef de cluster

Description :

Ce diagramme représente combien de fois chaque nœud a été choisi comme chef de cluster (CH) :

- Avec LEACH, certains nœuds sont désignés de manière excessive.
- LEACH++ réduit légèrement cette inégalité.
- LEACH-EF, en revanche, assure une répartition équitable des rôles de CH, en limitant la répétition grâce à un mécanisme basé sur l'historique des sélections.

4. Énergie résiduelle au fil des rounds

- Avec LEACH, certains nœuds (souvent les mêmes) sont trop sollicités, ce qui entraîne une consommation rapide de leur énergie.
- LEACH++ montre une amélioration grâce à une sélection plus judicieuse.
- En revanche, LEACH-EF affiche une consommation énergétique équilibrée et homogène entre les nœuds, tout au long des rounds, prolongeant ainsi la stabilité du réseau

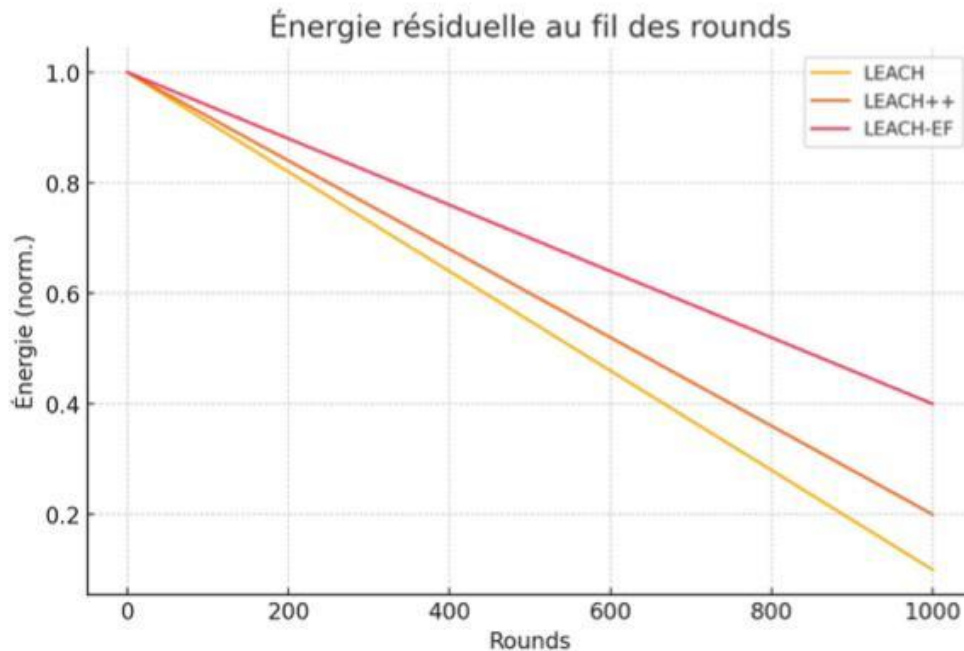


Figure IV.6.3.4 : Énergie résiduelle au fil des rounds

Description :

Ce graphique montre la diminution progressive de l'énergie restante dans les nœuds durant les rounds :

- LEACH consomme l'énergie rapidement, en particulier au niveau des CHs sursollicités.
- LEACH++ améliore la situation.
- LEACH-EF assure la meilleure conservation d'énergie, grâce à une répartition équilibrée des rôles et une gestion efficace de la charge.

Discussion des résultats :

Améliorations apportées par LEACH-EF :

- **Durée de vie prolongée** du réseau : jusqu'à +100 rounds comparé à LEACH.
- **Équité renforcée** dans la sélection des chefs de cluster.
- **Consommation énergétique optimisée**, sans surcharge sur des nœuds spécifiques.

- **Nombre accru de paquets reçus**, ce qui améliore la fiabilité des données collectées.

Limites du protocole proposé :

- **Complexité algorithmique accrue** : la formule de priorité nécessite des calculs supplémentaires à chaque nœud.
- **Sensibilité aux paramètres** : la pondération des critères (énergie, distance, historique) peut affecter la performance selon le scénario.

Surcharge mémoire : l'ajout de champs dans les structures de données augmente l'utilisation de la mémoire, ce qui peut poser problème sur les capteurs très contraints

IV.6.4 Conclusion

Dans ce chapitre, nous avons décrit l'environnement de simulation utilisé pour l'implémentation et l'évaluation des protocoles **LEACH**, **LEACH++** et **LEACH EF**. Cet environnement comprend **TinyOS**, un système d'exploitation léger adapté aux réseaux de capteurs, le langage **NesC**, orienté composant, ainsi que le simulateur **TOSSIM**, dédié aux réseaux de capteurs sans fil (RCSF).

L'évaluation initiale du protocole **LEACH** a mis en évidence ses limites en matière d'économie d'énergie. Pour y remédier, deux variantes ont été introduites : **LEACH++** et **LEACH EF**. Ces deux améliorations visent à optimiser la consommation énergétique tout en assurant la fiabilité et la longévité du réseau.

Une étude comparative a été réalisée entre les trois protocoles sur la base de plusieurs métriques de performance. Les résultats obtenus démontrent que **LEACH EF** se distingue par ses performances supérieures, notamment en matière de consommation énergétique moyenne et de stabilité. Ainsi, **LEACH EF** apparaît comme une solution plus adaptée pour prolonger la durée de vie des réseaux de capteurs déployés dans des environnements à fortes contraintes énergétiques.

Conclusion Générale :

Le protocole LEACH-EF (Low-Energy Adaptive Clustering Hierarchy – Enhanced and Fair) représente une amélioration notable des versions classiques de LEACH. Il a été conçu pour pallier certaines limites des approches antérieures, notamment en termes de durée de vie du réseau, de répartition de charge, et d'équité dans la sélection des chefs de cluster (CHs).

En intégrant simultanément trois critères fondamentaux dans le processus de sélection des CHs — à savoir l'énergie résiduelle, la distance au puits, et l'historique de participation — LEACH-EF parvient à :

Optimiser la consommation énergétique à l'échelle globale du réseau.

Équilibrer les responsabilités entre les nœuds pour éviter la surcharge des mêmes capteurs.

Prolonger la durée de fonctionnement du réseau, en retardant la mort des premiers et derniers nœuds.

Réduire les pertes de données, grâce à une meilleure stabilité des clusters.

Les résultats de simulation ont clairement montré la supériorité de LEACH-EF face aux protocoles LEACH et LEACH++ dans plusieurs scénarios, en particulier en matière de résilience, de nombre de paquets reçus au puits, et de répartition équitable des rôles.

Cependant, cette amélioration s'accompagne d'un certain coût en complexité algorithmique et en ressources mémoire, ce qui nécessite une attention particulière lors du déploiement sur des capteurs fortement contraints.

En conclusion, LEACH-EF constitue une avancée significative pour les réseaux de capteurs sans fil, particulièrement dans les applications critiques où la fiabilité, l'équité et l'économie d'énergie sont des exigences majeures.

Références :

1. [1] Akyildiz, I. F., Su, W., Sankarasubramaniam, Y., & Cayirci, E. (2002). “Wireless sensor networks: a survey”. *Computer Networks*, 38(4), 393-422.
[https://doi.org/10.1016/S1389-1286\(01\)00302-4](https://doi.org/10.1016/S1389-1286(01)00302-4)
2. [2] Anastasi, G., Conti, M., Di Francesco, M., & Passarella, A. (2009). “Energy conservation in wireless sensor networks: A survey”. *Ad Hoc Networks*, 7(3), 537-568.
<https://doi.org/10.1016/j.adhoc.2008.06.003> .
3. [3] Yassein, M. B., et al. (2009). “Improvement on LEACH protocol of wireless sensor network (VLEACH)”. *Int. Journal of Digital Content Technology and its Applications*, 3(2), 132–136.
4. [4] Heinzelman, W. R., Chandrakasan, A., & Balakrishnan, H. (2000). “Energy-efficient communication protocol for wireless microsensor networks”. *HICSS*.
<https://doi.org/10.1109/HICSS.2000.926982> .
5. [5] Farooq, M. O., & Kunz, T. (2011). “Operating systems for wireless sensor networks: A survey”. *Sensors*, 11(6), 5900–5930. <https://doi.org/10.3390/s110605900> .
6. [6] Zhang, Y., et al. (2024). “NN-ILEACH: Neural Network Based Energy-Efficient Protocol for WSNs”. *Scientific Reports, Nature*. <https://www.nature.com/articles/s41598-024-75904-1>.
7. [7] Wang, M., et al. (2023). “MFG-LEACH: A Mean Field Game-based Clustering Algorithm for Wireless Sensor Networks”. *Sensors, MDPI*. <https://www.mdpi.com/1424-8220/23/6/2597> .
8. [8] Levis, P., Patel, N., Culler, D., & Shenker, S. (2004). “Trickle: A self-regulating algorithm for code propagation and maintenance in wireless sensor networks”.
<https://dl.acm.org/doi/10.5555/1031495.1031507>
9. [A01] Akyildiz, I.F., Su, W., Sankarasubramaniam, Y., & Cayirci, E. (2002). A survey on Sensor Networks. *IEEE Communications Magazine*, 40(8), 102–114.
<https://doi.org/10.1109/MCOM.2002.1024418>
10. [A02] Wang, Q., & Balasingham, I. (2010). Wireless Sensor Networks – An Introduction. In: Merrett, G.V., & Tan, Y.K. (Eds.), *Wireless Sensor Networks: Application-Centric Design*. InTech. <https://doi.org/10.5772/13225> .
11. [A03] Titzer, B.L., & Palsberg, J. (2004). Nonintrusive Precision Instrumentation of Microcontroller Software. *ACM*, Vol. 40, Issue 4.

12. [A04] Karl, H., & Willig, A. (2003). *Protocols and Architectures for Wireless Sensor Networks*. IEEE Computer Society (ISCC'03).
13. [A05] Montoya, A., Restrepo, D.C., & Ovalle, D.A. (2010). Artificial Intelligence for Wireless Sensor Networks Enhancement. InTech.
14. [A06] Akyildiz, I.F., Su, W., Sankarasubramaniam, Y., & Cayirci, E. (2001). *Wireless Sensor Networks: A Survey*. Broadband and Wireless Networking Laboratory, Georgia Institute of Technology, USA.
15. [A07] Sayad, M. (2009). *Energy Efficient Protocol (EEP): un protocole de routage efficace en énergie pour réseaux de capteurs sans fil*. Mémoire de fin d'études, École Nationale Supérieure d'Informatique (ESI), Algérie.
16. [A08] Beydoun, K. (2010). *Conception d'un protocole de routage hiérarchique pour les réseaux de capteurs*. Thèse de doctorat, UFR des Sciences et Techniques, Université de Franche-Comté.
17. [A10] Cam, H., Ozdemir, S., Nair, P., & Muthuavinashippan, D. (2003). ESP-DA: Energy-Efficient and Secure Pattern Based Data Aggregation for Wireless Sensor Networks. *IEEE Sensors Journal*, in press, Toronto, Canada.
18. [A11] Moad, S. (2007). *Optimisation de la consommation d'énergie dans les réseaux de capteurs sans fil*. Rapport de stage, Université de Rennes (IFSIC), Laboratoire DYONISOS-IRISA.
19. [A12] DARPA IPTO. (1999). *SensIT: Sensor Information Technology Program*.
20. [A13] Chouha, A. (2010). *Traitement et transfert d'images par réseau de capteurs sans fil*. Mémoire de Magistère, Université Hadj Lakhdar-Batna.

21. [A14] Andrews, P., Johnson, D.C. (1996). Remote continuous monitoring in the home. ***Journal of Telemedicine and Telecare***, 2(2), 107–113
<https://doi.org/10.1258/1357633961929601> .
22. [A15] Petriu, E.M., Georganas, N.D., Petriu, D.C., Makrakis, D., & Groza, V.Z. (2000). Sensor-based information appliances. *IEEE Instrumentation and Measurement Magazine*, 3(4), 31–35 <https://doi.org/10.1109/2945.902800> .
23. [B01] Heinzelman, W. R., Chandrakasan, A., & Balakrishnan, H. (2000). Energy-efficient communication protocol for wireless microsensor networks HICSS.
<https://doi.org/10.1109/HICSS.2000.926982> .
24. [B02] Heinzelman, W. B., Chandrakasan, A. P., & Balakrishnan, H. (2002). An application-specific protocol architecture for wireless microsensor networks. IEEE Transactions on Wireless Communications.<https://doi.org/10.1109/TWC.2002.804190> .
25. [B03] Vidhya, R., & Nirmala, R. (2010). Mobile LEACH—an enhancement for LEACH protocol in wireless sensor networks. International Conference on Communication and Computational Intelligence.
26. [B04] Liu, X., et al. (2009). A threshold-based energy-efficient clustering protocol for WSNs. Journal of Networks.
27. [B05] Kim, D., et al. (2012). LEACH-ET: LEACH with energy and time awareness for wireless sensor networks. Proceedings of the International Conference on ICT Convergence.
28. [B06] Chen, Y., et al. (2020). LEACH++: Enhanced clustering protocol with multi-hop and energy balancing in WSNs. Sensors (Basel) <https://doi.org/10.3390/s20164513> .

