

République Algérienne Démocratique et Populaire  
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique  
Université Abderrahmane Mira - Béjaïa -  
Faculté des Sciences Exactes  
Département d'Informatique



*Mémoire de fin de cycle en vue de l'obtention d'un Master Recherche  
Option : Réseaux et Systèmes Distribués*

## Thème

Réorganisation des clusters à base de localisation pour  
un routage multi-sauts avec efficacité énergétique dans  
les réseaux de capteurs sans fil

Réalisé par :

M<sup>lle</sup> KHANOUCHE Feriel

M<sup>lle</sup> MAOUCHE Lydia

Soutenu le 29 Juin 2016 devant le jury composé de :

Président	M. FARAH Zoubeyr	M.C.B	U. A/Mira Béjaïa, Algérie.
Examinatrice	M <sup>me</sup> KHOULALENE Nadjet	M.A.A	U. A/Mira Béjaïa, Algérie.
Examinateur	M. AISSANI Sofiane	M.A.A	U. A/Mira Béjaïa, Algérie.
Rapporteur	M. MIR Foudil	M.A.A	U. A/Mira Béjaïa, Algérie.

Promotion 2015-2016.

## *- Remerciements -*

Nos remerciements vont en premier lieu à notre encadreur **M. MIR Foudil** pour la préciosité de ses conseils, son infinie disponibilité et son orientation qui ont constitué un apport considérable grâce auquel ce travail a pu être mené à bon port.

Nous tenons à remercier également **M. KHANOUCHE Mohamed-Essaïd** pour son soutien et sa disponibilité qui ont été d'une grande aide pendant l'élaboration de ce travail.

Nous adressons nos sincères remerciements à chacun des membres du jury pour l'intérêt qu'ils ont porté à notre recherche en acceptant d'examiner notre travail et de l'enrichir par leurs propositions.

Nos remerciements s'étendent à tous nos enseignants et les membres du département d'Informatique de l'université **ABDERRAHMANE MIRA**.

Ainsi qu'à tous ceux et celles qui ont contribué de près ou de loin à l'accomplissement de ce travail.

- *Dédicaces* -

A nos parents, pour leurs sacrifices déployés à notre égard, pour leur patience, leur amour et leur confiance. Qu'ils trouvent dans ce modeste travail, le témoignage de notre profonde affection et de notre attachement indéfectible, nulle dédicace ne puisse exprimer ce qu'on leur doit.

A nos frères et sœurs et tous nos amis pour chaque mot reçu, chaque geste d'amitié, à chaque main tendue et pour toute attention témoignée.

# Table des matières

<b>Table des matières</b>	<b>i</b>
<b>Table des figures</b>	<b>ii</b>
<b>Liste des tableaux</b>	<b>iv</b>
<b>Liste des algorithmes</b>	<b>v</b>
<b>Liste des abréviations</b>	<b>vi</b>
<b>Introduction générale</b>	<b>1</b>
<b>1 Protocoles de routage dans les réseaux de capteurs sans fil : Revue de littérature</b>	<b>3</b>
1.1 Introduction . . . . .	3
1.2 Vue d'ensemble sur les réseaux de capteurs sans fil . . . . .	3
1.2.1 Architecture d'un nœud capteur . . . . .	4
1.2.2 Domaines d'application des réseaux de capteurs sans fil . . . . .	5
1.3 Routage dans les réseaux de capteurs sans fil . . . . .	6
1.3.1 Contraintes liées à la conception des protocoles de routage . . . . .	6
1.3.2 Modèles de trafic dans les réseaux de capteurs sans fil . . . . .	10
1.4 Classification des protocoles de routage dans les réseaux de capteurs sans fil . . . . .	11
1.4.1 Structure du réseau . . . . .	11
1.4.2 Modèle de communication . . . . .	22
1.4.3 Topologie . . . . .	27
1.4.4 Fiabilité du routage . . . . .	30
1.5 Conclusion . . . . .	32
<b>2 Approche de clustering dans les réseaux de capteurs sans fil</b>	<b>34</b>
2.1 Introduction . . . . .	34
2.2 Apport du clustering . . . . .	35
2.2.1 Équilibrage de charge . . . . .	35
2.2.2 Tolérance aux pannes . . . . .	35
2.2.3 Augmentation de connectivité et réduction des délais . . . . .	35
2.2.4 Nombre minimal de clusters . . . . .	36

2.2.5	Longévité maximale du réseau . . . . .	36
2.3	Paramètres du clustering . . . . .	36
2.3.1	Propriétés des clusters . . . . .	36
2.3.2	Capacités des cluster heads . . . . .	37
2.3.3	Processus de clustering . . . . .	38
2.4	Protocole de routage hiérarchique LEACH . . . . .	39
2.4.1	Présentation du protocole LEACH . . . . .	40
2.4.2	Modèle radio du premier ordre . . . . .	42
2.4.3	Avantages et limites de LEACH . . . . .	43
2.5	Améliorations du protocole LEACH . . . . .	43
2.5.1	LEACH-C : LEACH-Centralized . . . . .	43
2.5.2	FZ-LEACH : Far-Zone LEACH . . . . .	44
2.5.3	MG-LEACH : Multi Group Based LEACH . . . . .	46
2.5.4	EC-LEACH : Equalized Cluster LEACH . . . . .	46
2.5.5	U-LEACH : Universal-LEACH . . . . .	47
2.5.6	Q-LEACH : Quadrature-LEACH . . . . .	47
2.5.7	ML-LEACH : Multi-Level LEACH . . . . .	50
2.6	Conclusion . . . . .	52
<b>3</b>	<b>RCLME : Réorganisation des Clusters à base de Localisation pour un routage Multi-sauts avec efficacité Énergétique</b>	<b>53</b>
3.1	Introduction . . . . .	53
3.2	Problématique et motivations . . . . .	53
3.3	Propriétés du réseau et hypothèses . . . . .	54
3.4	RCLME : La solution proposée . . . . .	54
3.4.1	Phase de configuration des clusters . . . . .	55
3.4.2	Phase de réorganisation des clusters . . . . .	58
3.4.3	Phase d'acheminement des données . . . . .	64
3.5	Conclusion . . . . .	64
<b>4</b>	<b>Simulation et évaluation de performances</b>	<b>65</b>
4.1	Introduction . . . . .	65
4.2	Description de l'environnement de simulation . . . . .	65
4.3	Résultats de simulation . . . . .	65
4.3.1	Paramètres de simulation . . . . .	66
4.3.2	Discussion et comparaison des résultats . . . . .	66
4.4	Conclusion . . . . .	70
	<b>Conclusion générale et perspectives</b>	<b>72</b>
	<b>Bibliographie</b>	<b>74</b>

# Table des figures

1.1	Exemple d'un réseau de capteurs sans fil [52]. . . . .	4
1.2	Composants d'un nœud capteur [57]. . . . .	4
1.3	Taxonomie des applications des réseaux de capteurs sans fil [52]. . . . .	5
1.4	Divers modèles de trafic dans les réseaux de capteurs sans fil [47]. . . . .	10
1.5	Schéma du principe du protocole TORA [68]. . . . .	13
1.6	Exemple d'un réseau partagé en clusters [2]. . . . .	16
1.7	Exemple d'un réseau partagé en une grille virtuelle [38]. . . . .	18
1.8	Exemple d'un réseau utilisant une chaîne [39]. . . . .	19
1.9	Illustration du protocole de la diffusion dirigée [26]. . . . .	23
1.10	Illustration du protocole SPIN [47]. . . . .	26
2.1	Propriétés des clusters [2]. . . . .	37
2.2	Capacités des cluster heads [2]. . . . .	38
2.3	Processus du clustering [2]. . . . .	39
2.4	Formation des clusters [22]. . . . .	41
2.5	Modèle radio du premier ordre [22]. . . . .	43
2.6	Formation des FAR-ZONE [28]. . . . .	45
2.7	Principe du protocole Q-LEACH [41]. . . . .	48
2.8	Hierarchisation dans TL-LEACH [37]. . . . .	51
3.1	Principe de la méthode de trilatération. . . . .	55
3.2	Étape d'élection des Cluster Heads de l'algorithme <i>RCLME</i> . . . . .	56
3.3	Message <i>ADV</i> diffusé lors de l'association des membres. . . . .	57
3.4	Message <i>Join-REQ</i> envoyé par un nœud membre. . . . .	57
3.5	Partitionnement d'un cluster en quatre zones. . . . .	59
3.6	Message <i>NOTIFY</i> envoyé aux membres du cluster à la suite de la migration. . . . .	61
3.7	Formation des sous zones dans un cluster. . . . .	61
3.8	Message <i>NOTIFY_SCH</i> envoyé après le choix du sous cluster head. . . . .	62
3.9	Acheminement des données dans le protocole RCLME. . . . .	64
4.1	Nombre de nœuds vivants <i>versus</i> nombre de rounds. . . . .	67
4.2	Énergie résiduelle des nœuds <i>versus</i> nombre de rounds. . . . .	68
4.3	Consommation d'énergie <i>versus</i> nombre de nœuds déployés. . . . .	69

4.4 Durée de vie jusqu'à la défaillance du dernier nœud *versus* nombre de nœuds déployés. 70

# Liste des tableaux

1.1	Comparaison entre les protocoles proactifs et réactifs [47]. . . . .	14
1.2	Comparaison entre certains protocoles plats [47]. . . . .	15
1.3	Comparaison entre certains protocoles hiérarchiques [47]. . . . .	21
1.4	Comparaison entre certains protocoles à base de requêtes [47]. . . . .	23
1.5	Comparaison entre certains protocoles basés sur la cohérence et la non-cohérence [47].	25
1.6	Comparaison entre certains protocoles à base de négociations [47]. . . . .	26
1.7	Comparaison entre certains protocoles basés sur la localisation [47]. . . . .	28
1.8	Comparaison entre certains protocoles basés sur les agents mobiles [47]. . . . .	29
1.9	Comparaison entre certains protocoles basés sur le multi-chemins [47]. . . . .	31
1.10	Comparaison entre certains protocoles basés sur la QoS [47]. . . . .	32
4.1	Paramètres de simulation. . . . .	66
4.2	Énergie moyenne consommée <i>versus</i> nombre de nœuds déployés. . . . .	69
4.3	Défaillance du dernier nœud du réseau <i>versus</i> nombre de nœuds déployés. . . . .	70



# Liste des algorithmes

1	Sélection des cluster heads dans le protocole Q-LEACH [41]. . . . .	49
2	Association des membres aux cluster heads dans le protocole Q-LEACH [41]. . . . .	50
3	Élection du cluster head temporaire et association des membres dans <i>RCLME</i> . . . . .	58
4	Détermination de la zone la plus dense dans RCLME. . . . .	60
5	Partitionnement d'un cluster $C_i$ et formation des sous zones dans <i>RCLME</i> . . . . .	63

# Liste des abréviations

<b>ACQUIRE</b>	<b>A</b> ctive <b>Q</b> Uery forwarding <b>I</b> n senso <b>R</b> n <b>E</b> tworks
<b>ADC</b>	<b>A</b> nalog to <b>D</b> igital <b>C</b> onverter
<b>AMRP</b>	<b>A</b> verage <b>M</b> inimum <b>R</b> eachability <b>P</b> ower
<b>APTEEN</b>	<b>A</b> daptive <b>P</b> eriodic <b>T</b> hreshold sensitive <b>E</b> nergy <b>E</b> fficient sensor <b>N</b> etwork protocol
<b>BAN</b>	<b>B</b> ody <b>A</b> rea <b>N</b> etwork
<b>BCDCP</b>	<b>B</b> ase- <b>S</b> tation <b>C</b> ontrolled <b>D</b> ynamic <b>C</b> lustering <b>P</b> rotocol
<b>CBMPR</b>	<b>C</b> luster- <b>B</b> ased <b>M</b> ulti- <b>P</b> ath <b>R</b> outing
<b>CDMA</b>	<b>C</b> ode <b>D</b> ivision <b>M</b> ultiple <b>A</b> ccess
<b>CSMA</b>	<b>C</b> arrier <b>S</b> ense <b>M</b> ultiple <b>A</b> ccess
<b>DCF</b>	<b>D</b> irectional <b>C</b> ontrolled <b>F</b> usion
<b>DD</b>	<b>D</b> irected <b>D</b> iffusion
<b>DEEC</b>	<b>D</b> istributed <b>E</b> nergy <b>E</b> fficient
<b>DGR</b>	<b>D</b> irectional <b>G</b> eographical <b>R</b> outing
<b>DHAC</b>	<b>D</b> istributed <b>H</b> ierarchical <b>A</b> gglomerative <b>C</b> lustering
<b>DREAM</b>	<b>D</b> istance <b>R</b> outing <b>E</b> ffect <b>A</b> lgorithm for <b>M</b> obility
<b>DSSS</b>	<b>D</b> irect <b>S</b> equence <b>S</b> pread <b>S</b> pectrum
<b>DT</b>	<b>D</b> istance <b>T</b> able
<b>DWEHC</b>	<b>D</b> istributed <b>W</b> eight- <b>B</b> ased <b>E</b> nergy- <b>E</b> fficient <b>H</b> ierarchical <b>C</b> lustering
<b>EC-LEACH</b>	<b>E</b> qualized <b>C</b> luster <b>LEACH</b>
<b>ELCH</b>	<b>E</b> xtending <b>L</b> ifetime of <b>C</b> luster <b>H</b> ead
<b>E-TORA</b>	<b>E</b> nergy-aware <b>T</b> emporarily <b>O</b> rdered <b>R</b> outing <b>A</b> lgorithm
<b>FZ-LEACH</b>	<b>F</b> ar- <b>Z</b> one <b>LEACH</b>
<b>GAF</b>	<b>G</b> eographic <b>A</b> daptive <b>F</b> idelity
<b>GBDD</b>	<b>G</b> rid <b>B</b> ased <b>D</b> ata <b>D</b> issemination
<b>GDSTR</b>	<b>G</b> reedy <b>D</b> istributed <b>S</b> panning <b>T</b> ree <b>R</b> outing
<b>GEAR</b>	<b>G</b> eographic and <b>E</b> nergy <b>A</b> ware <b>R</b> outing
<b>GEM</b>	<b>G</b> raph <b>E</b> mbedding for <b>R</b> outing
<b>GPS</b>	<b>G</b> lobal <b>P</b> ositioning for <b>S</b> ystem
<b>GRAB</b>	<b>G</b> RAdient <b>B</b> roadcast
<b>GSM</b>	<b>G</b> lobal <b>S</b> ystem for <b>M</b> obile <b>C</b> ommunications
<b>HEED</b>	<b>H</b> ybrid <b>E</b> nergy <b>E</b> fficient <b>D</b> istributed
<b>HGR</b>	<b>H</b> ybrid <b>G</b> eographic <b>R</b> outing

<b>HMRP</b>	<b>H</b> ierarchy- <b>B</b> ased <b>M</b> ultipath <b>R</b> outing <b>P</b> rotocol
<b>HPAR</b>	<b>H</b> ierarchical <b>P</b> ower <b>A</b> ware <b>R</b> outing
<b>IGF</b>	<b>I</b> mplicit <b>G</b> eographic <b>F</b> orwarding
<b>I-LEACH</b>	<b>I</b> mproved <b>LEACH</b>
<b>LCT</b>	<b>L</b> ink <b>C</b> ost <b>T</b> able
<b>LEACH</b>	<b>L</b> ow <b>E</b> nergy <b>A</b> daptive <b>C</b> lustering <b>H</b> ierarchy
<b>LEACH-C</b>	<b>L</b> ow <b>E</b> nergy <b>A</b> daptive <b>C</b> lustering <b>H</b> ierarchy <b>C</b> entralized
<b>LMR</b>	<b>L</b> abel-based <b>M</b> ultipath <b>R</b> outing
<b>MERR</b>	<b>M</b> inimum <b>E</b> nergy <b>R</b> elay <b>R</b> outing
<b>MG-LEACH</b>	<b>M</b> ulti <b>G</b> roup based <b>LEACH</b>
<b>MGR</b>	<b>M</b> ultimedia <b>G</b> eographic <b>R</b> outing
<b>MIMO</b>	<b>M</b> ultihop <b>V</b> irtual <b>M</b> ultiple <b>I</b> nput <b>M</b> ultiple <b>O</b> utput
<b>MIT</b>	<b>M</b> assachusetts <b>I</b> nstitute of <b>T</b> echnology
<b>ML-LEACH</b>	<b>M</b> ulti- <b>L</b> evel <b>LEACH</b>
<b>MMSPEED</b>	<b>M</b> ulti- <b>P</b> ath and <b>M</b> ulti- <b>S</b> PEED
<b>MRL</b>	<b>M</b> essage <b>R</b> etransmission <b>L</b> ist
<b>MTE</b>	<b>M</b> inimum <b>T</b> ransmission <b>E</b> nergy
<b>MWA</b>	<b>M</b> ultiple <b>W</b> inner <b>A</b> lgorithm
<b>NHRPA</b>	<b>N</b> ovel <b>H</b> ierarchical <b>R</b> outing <b>P</b> rotocol <b>A</b> lgorithm
<b>OGF</b>	<b>O</b> n-demand <b>G</b> eographic <b>F</b> orwarding
<b>PAGER-M</b>	<b>P</b> artial-partition <b>A</b> voiding <b>G</b> eographic <b>R</b> outing- <b>M</b> obile
<b>PEGASIS</b>	<b>P</b> ower <b>E</b> fficient <b>G</b> athering in <b>S</b> ensor <b>I</b> nformation <b>S</b> ystem
<b>Q-LEACH</b>	<b>Q</b> uadrature- <b>LEACH</b>
<b>QoS</b>	<b>Q</b> uality of <b>S</b> ervice
<b>RCLME</b>	<b>R</b> éorganisation des <b>C</b> lusters à base de <b>L</b> ocalisation pour un routage <b>M</b> ulti-sauts avec efficacité <b>E</b> nergétique
<b>RCSF</b>	<b>R</b> éseau de <b>C</b> apteurs <b>S</b> ans <b>F</b> il
<b>ROAM</b>	<b>R</b> outing <b>O</b> n-demand <b>A</b> cylic <b>M</b> ultipath
<b>RPL</b>	<b>R</b> outing <b>P</b> rotocol for <b>L</b> ow power and <b>L</b> ossy <b>N</b> etworks
<b>RR</b>	<b>R</b> umor <b>R</b> outing
<b>RREP</b>	<b>R</b> oute <b>RE</b> ply
<b>RREQ</b>	<b>R</b> oute <b>RE</b> quest
<b>RSSI</b>	<b>R</b> eceived <b>S</b> ignal <b>S</b> trength <b>I</b> ndication
<b>RT</b>	<b>R</b> outing <b>T</b> able
<b>SAR</b>	<b>S</b> equential <b>A</b> ssignment <b>R</b> outing
<b>SHPER</b>	<b>S</b> caling <b>H</b> ierarchical <b>P</b> ower <b>E</b> fficient <b>R</b> outing
<b>SIP</b>	<b>S</b> ingle agent-based <b>I</b> inerary <b>P</b> lanning
<b>SPIN</b>	<b>S</b> ensor <b>P</b> rotocols for <b>I</b> nformation via <b>N</b> egotiation
<b>SPIN-BN</b>	<b>SPIN</b> for <b>B</b> roadcast <b>N</b> etworks
<b>SPIN-EC</b>	<b>SPIN</b> with <b>E</b> nergy <b>C</b> onservation
<b>SPIN-PP</b>	<b>SPIN</b> for <b>P</b> oint to <b>P</b> oint <b>C</b> ommunication

<b>SPIN-RL</b>	<b>SPIN</b> with <b>R</b> eliability
<b>SWA</b>	Single <b>W</b> inner <b>A</b> lgorithm
<b>TBRPF</b>	<b>T</b> opology <b>D</b> issemination <b>B</b> ased on <b>R</b> everse- <b>P</b> ath <b>F</b> orwarding
<b>TDMA</b>	<b>T</b> ime <b>D</b> ivision <b>M</b> ultiple <b>A</b> ccess
<b>TEEN</b>	<b>T</b> hreshold sensitive <b>E</b> nergy <b>E</b> fficient sensor <b>N</b> etwork protocol
<b>TORA</b>	<b>T</b> emporarily <b>O</b> rdered <b>R</b> outing <b>A</b> lgorithm
<b>TTDD</b>	<b>T</b> wo <b>T</b> ier <b>D</b> ata <b>D</b> issemination
<b>U-LEACH</b>	<b>U</b> niversal- <b>L</b> EACH
<b>VGA</b>	<b>V</b> irtual <b>G</b> rid <b>A</b> rchitecture <b>R</b> outing
<b>WRP</b>	<b>W</b> ireless <b>R</b> outing <b>P</b> rotocol
<b>ZRP</b>	<b>Z</b> one <b>R</b> outing <b>P</b> rotocol

# Introduction générale

Durant ces dernières années, les domaines de l'électronique, la mécanique et les technologies de communication sans fil ont été marqués par un développement très rapide. Ce développement a permis la production à un coût raisonnable de petits dispositifs communicants équipés d'outils de capture. De nos jours, l'émergence des réseaux de capteurs sans fil (RCSF) représente une thématique de recherche prometteuse aussi bien sur le plan académique qu'industriel. Un tel type de réseaux est composé d'un grand nombre de nœuds ou capteurs déployés sur une zone d'intérêt et communiquent entre eux afin de réaliser une tâche particulière telle que mesurer une donnée relative à la température, l'humidité, la luminosité, etc. Les RCSF sont de plus en plus présents dans de nombreuses applications, telles que l'automatisation de processus industriels, les systèmes de contrôle aérien et le contrôle des patients dans des hôpitaux.

En raison de l'absence d'infrastructure de communication, ces réseaux posent de nombreux défis, en comparaison aux réseaux traditionnels, notamment la gestion de la consommation d'énergie au niveau des nœuds. L'objectif est de maximiser la durée de vie du réseau entier. L'énergie est la contrainte la plus cruciale dans le développement de ce type de réseau. En effet, le remplacement de la batterie des capteurs peut être très difficile, voire impossible, dans de nombreux scénarios d'application en particulier lorsque le réseau de capteurs est déployé dans des environnements hostiles ou distants. Par conséquent, la durée de vie de l'ensemble du réseau de capteurs dépend essentiellement de la durée de vie de la batterie de chaque nœud.

Par ailleurs, l'organisation hiérarchique des capteurs, appelée *clustering*, est l'un des mécanismes proposés pour faire face au problème de l'énergie limitée dans les réseaux de capteurs sans fil. Ce mécanisme consiste à regrouper et à attribuer des tâches spécifiques aux membres du groupe avant de transmettre l'information à des niveaux plus élevés. Le protocole LEACH (Low Energy Adaptive Clustering Hierarchy) est l'une des solutions hiérarchiques les plus populaires qui a introduit le principe du clustering lors du routage. La formation de clusters dans un RCSF est généralement réalisée en tenant compte de certains paramètres qui diffèrent d'un protocole à un autre, tels que l'énergie des nœuds, la distance entre les nœuds ou par rapport à la station de base, le nombre de voisins, etc.

Le routage hiérarchique dans les réseaux partagés en clusters est souvent basé sur le même principe. Ceci dit, chaque cluster comporte un chef de cluster appelé *cluster head* (CH) et des nœuds

membres (MC). Le CH agrège les données en provenance des MC puis les envoie à la station de base (SB).

Ce travail propose **RCLME** (pour **R**éorganisation des **C**lusters à base de **L**ocalisation pour un routage **M**ulti-sauts avec efficacité **E**nergétique), un protocole de routage basé sur le clustering pour les réseaux de capteurs sans fil. MRCL utilise la réorganisation des clusters ainsi qu'un routage multi-sauts intra et inter-clusters afin de minimiser la consommation énergétique et, par conséquent, étendre la durée de vie du réseau.

### **Ce mémoire est organisé en 4 chapitres :**

Le premier chapitre présente un aperçu sur les protocoles de routage dans les réseaux de capteurs sans fil, les contraintes de conception de ce type de réseaux ainsi qu'une classification tirée de la littérature.

Le second chapitre a été consacré à l'utilisation de la technique du clustering dans les protocoles de routage hiérarchiques. L'apport du clustering en comparaison à d'autres techniques ainsi que les paramètres du clustering ont été explicités. En outre, une étude critique du protocole LEACH et certaines de ces améliorations ont été discutées.

Dans le troisième chapitre, la description détaillée des étapes du protocole RCLME a été introduite. Ce protocole a été conçu pour pallier certains problèmes des protocoles LEACH et FZ-LEACH en particulier l'élection probabiliste des CHs et l'absence du routage multi-sauts.

Dans le quatrième chapitre, les performances du protocole proposé ont été évaluées en simulation. Une comparaison avec les protocoles LEACH et FZ-LEACH a été réalisée en termes de nœuds vivants, énergie consommée et durée de vie.

Enfin, ce travail s'achève par une conclusion et des perspectives à aborder dans un futur proche.

# Chapitre 1

## Protocoles de routage dans les réseaux de capteurs sans fil : Revue de littérature

### 1.1 Introduction

Au cours des deux dernières décennies, les besoins de surveillance et de contrôle de l'environnement ont engendrés l'apparition d'une nouvelle classe de réseaux sans fil à courte portée. Cette classe est ce en quoi consiste les *réseaux de capteurs sans fil* (RCSF). L'utilisation de ce type de réseaux touche les applications civiles et militaires en particulier dans les endroits hostiles ou inaccessibles. Un RCSF se compose d'un grand nombre de nœuds capteurs déployés dans une zone d'intérêt afin de collecter des données et de les acheminer à un nœud particulier appelé station de base (SB). Les RCSFs exigent la mise en place d'une architecture et une gestion prudente pour une meilleure couverture et une durée de vie étendue du réseau [57].

Le but des protocoles de routage est d'acheminer les données en tenant compte des ressources disponibles dans l'objectif de prolonger la durée de vie du réseau. Ce chapitre présente une synthèse des protocoles de routage à haute efficacité énergétique pour les RCSF et les classe selon quatre critères [47] : **structure du réseau**, **modèle de communication**, **topologie** et **fiabilité du routage**.

### 1.2 Vue d'ensemble sur les réseaux de capteurs sans fil

Les avancées technologiques récentes en matière d'équipements variés et à faible coût ont permis l'extension de la recherche dans le domaine des réseaux de capteurs sans fil (RCSF). Les RCSF sont un type particulier des réseaux mobiles *ad hoc*, ils sont composés d'une dizaine, voire d'une centaine, de nœuds capteurs communiquant par une liaison radio afin de collaborer dans le traitement des données. Les nœuds capteurs sont déployés dans des zones et s'occupent de la surveillance des changements d'un certain nombre de paramètres physiques. Les données collectées sont envoyées à un point de collecte appelé *Station de Base* (SB) pouvant être reliée à une machine

via un réseau utilisateur (Internet, satellite, etc.). La **figure 1.1** est un exemple d'un RCSF [5, 39, 48, 56].

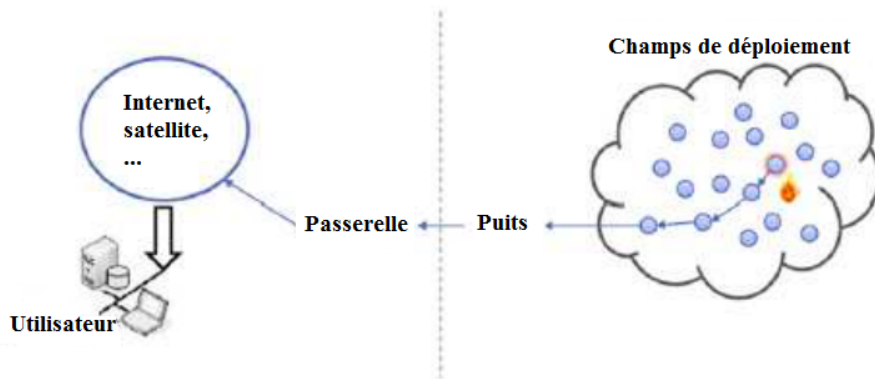


FIGURE 1.1 – Exemple d'un réseau de capteurs sans fil [52].

### 1.2.1 Architecture d'un nœud capteur

Un nœud capteur est un dispositif composé principalement d'une unité de communication, une unité de traitement, une unité d'acquisition, une unité d'énergie et une unité de stockage. Il transforme une grandeur physique observée (pression, humidité, température, etc.) en une grandeur électrique. Les interactions entre les unités d'un nœud capteur sont illustrées dans la **figure 1.2**. La consommation d'énergie est essentiellement due aux modules de communication sans fil et de traitement des données. L'unité optionnelle de localisation est également une source de consommation non négligeable d'énergie [57].

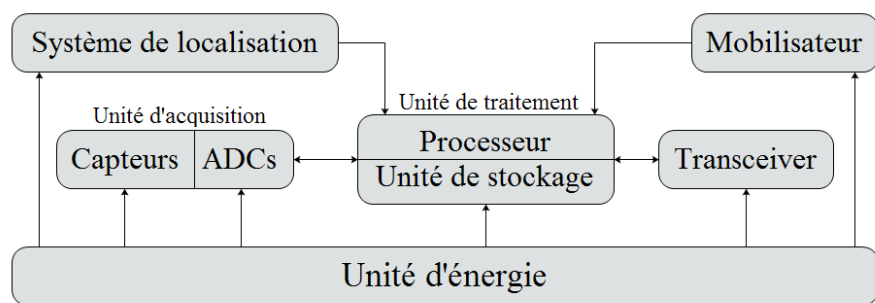


FIGURE 1.2 – Composants d'un nœud capteur [57].

- **Unité d'énergie** : elle est responsable de l'approvisionnement en puissance énergétique requise par les autres unités du système. Elle est généralement ni rechargeable ni remplaçable.



- **Unité de traitement** : c'est le noyau du capteur sans fil. Elle est habituellement composée d'un microprocesseur avec un espace limité de stockage. Les données captées sont communiquées au microprocesseur ou sont stockées dans la mémoire.
- **Unité d'acquisition** : Elle est composée de deux sous unités. L'une est responsable de la capture physique des données tandis que l'autre se charge de la conversion analogique/numérique (ADC).
- **Unité de communication** : Elle est composée d'un émetteur/récepteur (transceiver) permettant la communication entre les différents nœuds du réseau.
- **Unité de stockage** : Elle est d'une taille limitée. Cette unité inclut la mémoire des programmes et des données fournies par l'unité d'acquisition.

### 1.2.2 Domaines d'application des réseaux de capteurs sans fil

Les RCSF ont été utilisés dans un grand nombre de domaines d'application, tels que le domaine militaire, la prévention du crime, l'environnement, la santé (Body Area Network), l'industrie, l'agriculture, l'urbanisation et les infrastructures. Une taxonomie de ces applications est illustrée dans la **figure 1.3** [52].

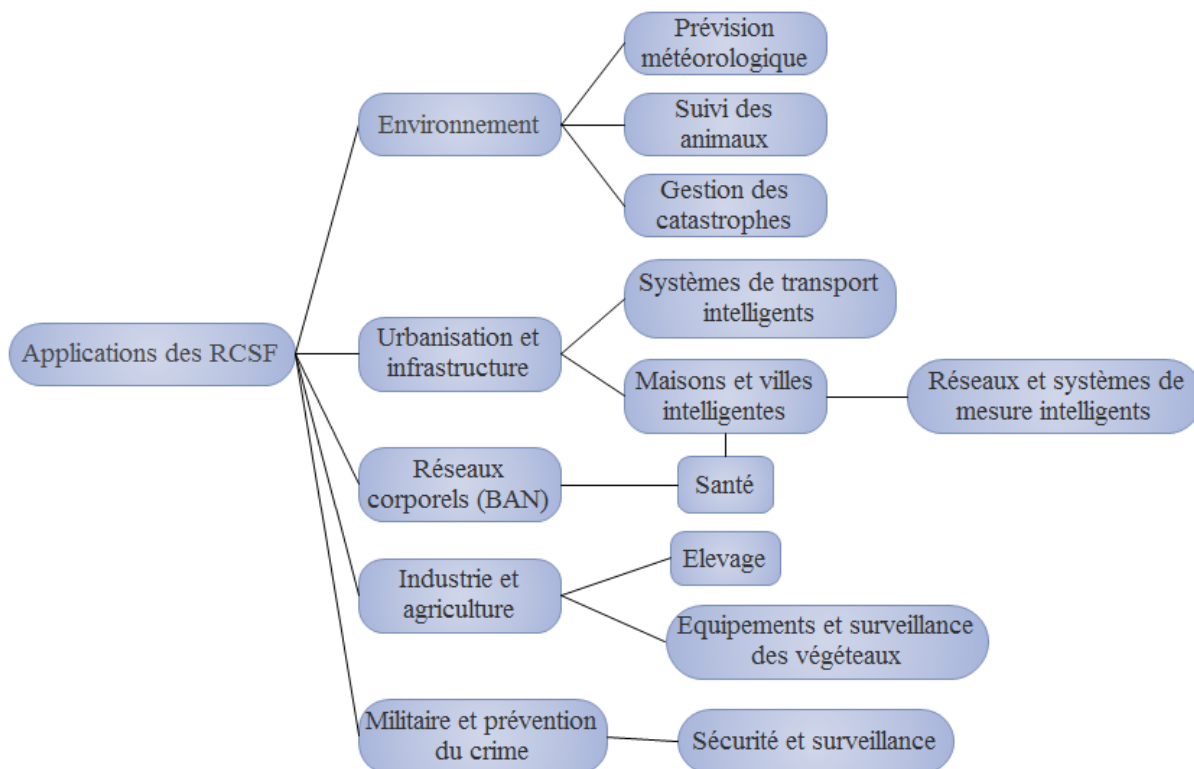


FIGURE 1.3 – Taxonomie des applications des réseaux de capteurs sans fil [52].

## 1.3 Routage dans les réseaux de capteurs sans fil

Afin d'assurer de meilleures performances, l'acheminement et la délivrance des données dans un RCSF doivent prendre en considération certaines caractéristiques des capteurs en l'occurrence : la durée de vie, la fiabilité, le temps de réponse, etc. En raison de la spécificité des RCSF, un nombre important de recherches a été axé sur le routage de données [5]. Le routage dans les RCSF requiert plus de spécificités que dans les réseaux mobiles ad hoc ou GSM (Global System for Mobile Communications) car [5, 47] :

- Les contraintes énergétiques des nœuds capteurs et les opérations effectuées demandent une gestion plus minutieuse ;
- Pour la plupart des applications des RCSF, les données capturées doivent être acheminées à une station de base ;
- La stationnarité des capteurs fait que les résultats soient prévisibles, étant donné la rareté des changements topologiques ;
- La connaissance de la position des capteurs étant importante pour la collecte des données dans certaines applications de surveillance ;
- La suppression des données redondantes pour conserver l'énergie et la bonne utilisation de la bande passante.

### 1.3.1 Contraintes liées à la conception des protocoles de routage

La conception des RCSF doit faire face à un certain nombre de contraintes. Les protocoles de routage doivent être conçus de façon à contourner ces contraintes et à faire en sorte qu'elles n'affectent pas l'efficacité du processus de l'acheminement des données.

#### 1.3.1.1 Déploiement des nœuds capteurs

Cette opération dépend de l'application, elle peut être déterministe ou aléatoire. Dans le cas d'un déploiement déterministe, les nœuds capteurs sont placés manuellement dans des endroits précis (dans une usine, par exemple) et les données sont routées à travers des chemins prédéterminés. En revanche, le déploiement aléatoire consiste à disperser les nœuds capteurs (largage par avion) créant une infrastructure *ad hoc* [2, 62, 47, 48, 57]. Le schéma de déploiement initial vise à réduire le coût de l'installation, accroître la flexibilité de l'arrangement et assurer l'auto-organisation ainsi que la tolérance aux pannes.

#### 1.3.1.2 Hétérogénéité des nœuds capteurs

Les nœuds capteurs d'un réseau sont supposés être homogènes, c.à.d. ils ont des capacités égales en termes de calcul, de communication et d'énergie. Cependant, selon l'application, des nœuds hétérogènes peuvent être nécessaires à la réalisation d'une fonction particulière (relais, détection

et l'agrégation). Néanmoins, l'existence d'un ensemble de nœuds capteurs hétérogènes donne lieu à de nombreux problèmes techniques liés au clustering [47].

### 1.3.1.3 Modèles de communication de données

La façon dont les données sont collectées influence les protocoles de routage avec la prise en compte de la consommation d'énergie et la stabilité des chemins. En effet, la façon de collecter les données peut être continue, déclenché par un événement (*Event-Driven*), déclenché à la suite d'une requête (*Query-Driven*) ou bien hybride [47].

### 1.3.1.4 Évolutivité du réseau

Les protocoles de routage devraient être assez évolutifs pour répondre aux changements tel que l'augmentation du nombre de nœuds capteurs dans un réseau. La densité peut être comprise entre quelques nœuds capteurs à quelques centaines de nœuds capteurs dans une région, elle peut être calculée comme suit [5] :

$$\mu(R) = \frac{(N\pi^2)}{A}, \tag{1.1}$$

où  $N$  et  $R$  représentent respectivement le nombre de nœuds déployés dans la région  $A$  et la portée de transmission.  $\mu(R)$  donne le nombre de nœuds à l'intérieur du rayon de transmission de chaque nœud dans la région  $A$ .

### 1.3.1.5 Dynamique du réseau

En plus des différentes configurations des nœuds capteurs mobiles, les trois composants principaux dans un réseau de capteurs sont les nœuds capteurs, la station de base (SB) et les événements à surveiller. La mobilité de la SB ou de certains nœud particulier (*Cluster-Heads*) est parfois nécessaire. Le routage des messages en provenance des nœuds capteurs mobiles est plus difficile, la stabilité devient un facteur important d'optimisation, en plus de l'énergie et la bande passante [2].

### 1.3.1.6 Tolérance aux pannes

La tolérance aux pannes est la capacité de maintenir les fonctionnalités du réseau lors de la défaillance d'un ou plusieurs nœuds capteurs. Le niveau de tolérance aux pannes dépend de l'application. La tolérance aux pannes  $R_k(t)$  est modélisée selon une distribution de poisson, elle signifie la probabilité de ne pas avoir une défaillance à l'intérieur de l'intervalle de temps  $(0, t)$  [5] :

$$R_k(t) = \exp(-\lambda_k t), \tag{1.2}$$

où  $\lambda_k$  est le taux d'échec du capteur  $k$  pendant la période  $t$ .

### 1.3.1.7 Connectivité du réseau

La connectivité dépend de la façon dont les nœuds capteurs sont distribués. Une densité importante des nœuds capteurs assure la connectivité du réseau mais ne le met pas à l'abri d'un changement de topologie/taille car les nœuds capteurs meurent ou sont endommagés au fil du temps [50].

### 1.3.1.8 Support de transmission

Dans un RCSF, les nœuds sont reliés par un moyen de communication sans fil. Ces liens peuvent être formés par radio, infrarouge ou supports optiques. Les exigences croissantes des applications des RCSF rendent le choix du support de transmission plus difficile. De plus, les supports de transmission doivent être pris en charge par les différents systèmes de codage et de modulation [5].

### 1.3.1.9 Couverture du réseau

Dans les RCSF un nœud capteur n'a qu'une vision partielle du réseau, du fait que sa portée de transmission est limitée. Donc une large couverture est essentielle au bon fonctionnement du réseau [57].

### 1.3.1.10 Agrégation de données

Les nœuds capteurs peuvent générer des données redondantes qui seront agrégées par les nœuds capteurs agrégateurs de sorte que le nombre de transmissions soit réduit. L'agrégation permet de combiner des données provenant de différentes sources en utilisant des fonctions telles que la *suppression* (pour éliminer les doublons), *min*, *max* et *moyenne*. Sachant que ces calculs consomment moins d'énergie que la communication, des économies d'énergie substantielles peuvent être obtenues par l'agrégation de données. Cette technique a été utilisée pour atteindre l'efficacité énergétique et optimiser le trafic dans un certain nombre de protocoles de routage [4]. Les techniques d'agrégation dans les RCSF peuvent être classées en quatre catégories [58] :

- **Agrégation de données centralisée** : chaque nœud capteur du réseau envoie les données collectées à un nœud capteur *central* (un nœud puissant) en multi-sauts, celui-ci se charge de les agréger.
- **Agrégation de données basée cluster** : l'ensemble du réseau est divisé logiquement en *clusters* et chacun d'entre eux possède un *Cluster-Head* (CH) qui agrégera les données envoyées par les membres de son *cluster* et les transmettra par la suite à la station de base (SB).

- **Agrégation de données multi-sauts** : chaque nœud capteur transmet ses données à ses voisins. L'agrégation se fera au niveau des nœuds capteurs intermédiaires se trouvant entre la source et la station de base. Cette approche permet la découverte de chemins alternatifs lors de la défaillance d'un nœud capteur ou d'un lien.
- **Agrégation de données basée arbre** : l'agrégation se fait à l'aide de la construction d'arbre pour chaque transmission de données (arbre de recouvrement minimal) dont les nœuds capteurs sources et la station de base sont considérés comme *feuilles*. Chaque nœud capteur transmet ses données à un parent qui les agrègera.

### 1.3.1.11 Consommation d'énergie

Le nœud capteur sans fil, étant un dispositif micro-électronique, ne peut être équipé que d'une source limitée d'énergie. Dans certains cas, la reconstitution des ressources pourrait être impossible. La mort de quelques nœuds capteurs peut provoquer des changements significatifs dans la topologie. Par conséquent, la conservation et la gestion de l'énergie prennent une importance supplémentaire [5]. La consommation d'énergie se fait lors de la détection, la communication et le traitement de données. Elle varie avec la nature des applications.

La complexité de la détection d'événements joue un rôle crucial dans la détermination de l'énergie dépensée [5]. Dans [55], la consommation d'énergie a été présentée par la formule suivante :

$$P_c = N_T[P_T(T_{on} + T_{st}) + P_{out}(T_{on})] + N_R[P_R(R_{on} + R_{st})]. \quad (1.3)$$

$P_{T/R}$  est l'énergie consommée par l'émetteur/récepteur,  $P_{out}$  est la puissance de sortie (output) de l'émetteur,  $T/R_{on}$  est l'instant d'activation de l'émetteur/récepteur,  $T/R_{st}$  est le temps de démarrage de l'émetteur/récepteur,  $R_{T/R}$  est le nombre de fois que l'émetteur/récepteur a été mis sous tension par unité de temps.  $T_{on}$  peut être réécrit comme  $L/R$  ( $L$  étant la taille du paquet et  $R$  le débit de données).

### 1.3.1.12 Coûts de production

Étant donné que les RCSF sont constitués d'un grand nombre de nœuds capteurs, le coût d'un nœud capteur unique doit être aussi minime que possible [5].

### 1.3.1.13 Qualité de service (QoS)

Les données devraient être livrées dans leurs délais. Cependant, dans un bon nombre d'applications, la conservation de l'énergie est considérée comme relativement plus importante que la qualité des données transmises [47].

### 1.3.2 Modèles de trafic dans les réseaux de capteurs sans fil

Les différentes applications donnent lieu à divers modèles de trafic, présentés dans la **figure 1.4** [47] :

- **Communication locale** : Utilisée pour diffuser le statut d'un nœud capteur à ses voisins. En outre, elle sert à transmettre les données entre deux nœuds directement.
- **Communication point à point** : Utilisée pour envoyer un paquet de données à partir d'un nœud capteur arbitraire vers un autre nœud capteur arbitraire. Communément utilisée dans un environnement réseau sans fil.
- **Convergence** : Les paquets de données de multiples nœuds sont acheminés vers un nœud capteur de base unique. Couramment utilisée pour la collecte de données dans les réseaux de capteurs.
- **Divergence** : Elle est utilisée pour envoyer une commande du nœud capteur de base aux autres nœuds capteurs.
- **Agrégation** : Les paquets de données peuvent être traités dans le nœud capteur relais et le résultat global est acheminé vers le nœud capteur de base plutôt que des données brutes.

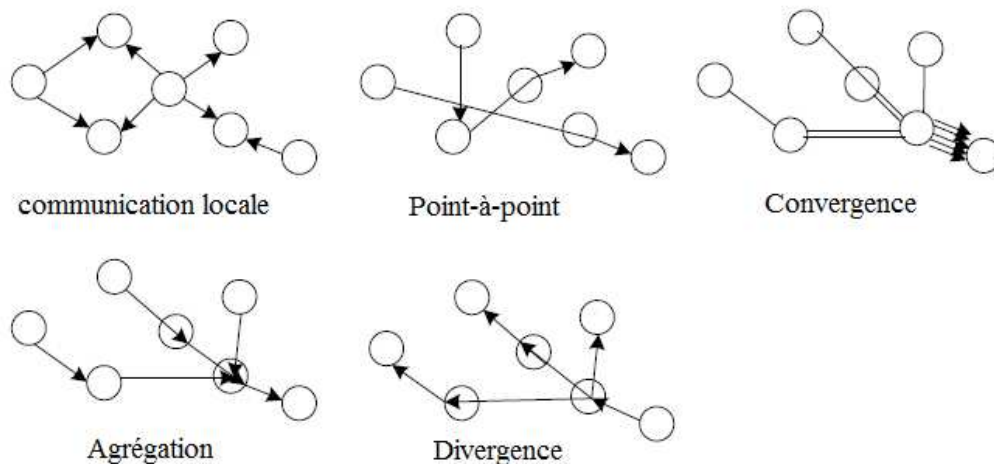


FIGURE 1.4 – Divers modèles de trafic dans les réseaux de capteurs sans fil [47].

## 1.4 Classification des protocoles de routage dans les réseaux de capteurs sans fil

Les protocoles de routages sont élaborés en fonction des besoins des applications et l'architecture du réseau désirée. Cependant, il y a des facteurs qui doivent être pris en considération lors de l'élaboration des protocoles de routage dans les RCSF. Leur objectif commun étant de réaliser l'efficacité énergétique lors du routage qui prolongerait considérablement la durée de vie du réseau [47]. Il existe différentes définitions pour la durée de vie d'un réseau de capteurs (fondées sur la fonctionnalité désirée). Elle peut être définie par la durée jusqu'au moment où le premier nœud meurt. Elle peut également être définie par le temps jusqu'au moment où une proportion de nœuds meurt [61] [63]. La classification de *Al-Karaki et al.* donne lieu à quatre (04) catégories, les protocoles de routage basés sur la structure de réseau, modèle de communication, la topologie et la fiabilité du routage [6].

### 1.4.1 Structure du réseau

Les protocoles sont répartis en fonction de la disposition des nœuds au sein du réseau ce qui est important pour le routage. Les protocoles inclus dans cette catégorie sont divisés selon l'uniformité des nœuds (existence ou non d'une distinction entre eux) en protocoles plats et protocoles hiérarchiques.

#### 1.4.1.1 Protocoles plats

Dans cette catégorie, tous les nœuds jouent le même rôle et collaborent ensemble dans la capture de données et la transmission à la station de base qui se chargera de les rassembler puis les transmettre vers l'utilisateur final. Ces protocoles sont classés selon la stratégie de routage en trois catégories : proactifs, réactifs et hybrides. Ils ne présentent pas les mêmes caractéristiques bien qu'ils aient été conçus pour le même type de protocoles [47].

#### A) Protocoles proactifs (Table-driven)

C'est des protocoles basés sur l'échange périodique des informations de routage entre les différents nœuds, chaque nœud construit sa propre table de routage qui peut être utilisée pour trouver un chemin de destination. Pour maintenir une vue cohérente du réseau chaque nœud envoie les mises à jour à travers le réseau sans fil. Par conséquent, quand un chemin dans une destination est nécessaire à un nœud où un paquet doit être transmis, le trajet est déjà connu donc il n'y a pas de retard dû à la découverte de route. Cependant, en gardant les informations mises à jour, il peut exiger beaucoup de bande passante [47, 56]. Plusieurs protocoles ont été développés dans cette catégorie, nous pouvons citer WRP (Wireless Routing Protocol)[44], TBRPF (Topology Dissemination Based on Reverse-Path Forwarding) [46],

etc.

- **WRP : WIRELESS ROUTE PROTOCOL** [44]

C'est un protocole qui hérite des propriétés de l'algorithme de *Bellman-Ford*. Il Maintient une vue mise à jour du réseau en utilisant des tables (Table de distances (DT), table de routage (RT), table des coûts des liens (LCT) et une liste de retransmission de messages (MRL)). Chaque entrée dans la MRL contient le numéro de séquence du message de mise à jour, un compteur de retransmission, un vecteur des acquittements requis avec une entrée pour chaque voisin et la liste des mises à jour envoyées. MRL enregistre quelles mises à jour doivent être retransmises ainsi que les voisins qui doivent acquitter la retransmission.

Un message de mise à jour est transmis seulement entre les nœuds voisins, il contient l'identificateur du nœud destination, la distance vers cette destination, l'identificateur du nœud prédécesseur de cette destination ainsi que la liste des nœuds qui doivent acquitter la mise à jour. L'utilisation de l'information sur le nœud prédécesseur du nœud destinataire permet d'éviter la formation de boucles.

Dans le cas de la perte d'un lien entre deux nœuds quelconques, les nœuds transmettent des messages de mise à jour à leurs voisins. Les voisins changent leurs entrées de table de distance et enregistrent de nouvelles routes entre d'autres nœuds. Un nœud qui n'envoie pas de paquets doit envoyer périodiquement des messages *Hello* pour notifier son existence. Les nœuds reconnaissent leurs voisins grâce à la réception des acquittements ou des paquets de données. Ce protocole n'est pas évolutif et ne convient pas aux grands réseaux.

## B) Protocoles réactifs (On-demand)

C'est des protocoles à la demande c'est-à-dire que la procédure de découverte de route est déclenchée seulement en cas de besoin. Il n'est plus question du constant envoi de mises à jour dans le réseau. Cependant, ce processus provoque des retards si les routes demandées ne sont pas disponibles. L'ensemble du processus se termine dès qu'un trajet est trouvé ou toutes les combinaisons de routes possibles ont été examinées [62, 47]. Plusieurs protocoles ont été développés dans cette catégorie, nous pouvons citer TORA (Temporarily Ordered Routing Algorithm) [68], gossiping [20], flooding [35], etc.



- **TORA : TEMPORARILY ORDERED ROUTING ALGORITHM** [68]

Dans ce protocole, chaque nœud  $i$  connaît sa propre hauteur (height) et la hauteur de chaque voisin  $j$  relié directement. La hauteur de chaque nœud est attribuée en fonction de son emplacement par rapport à la destination (**figure 1.5**). Lorsqu'un nœud reçoit un paquet de données, il le transmet toujours dans la direction en aval (de la hauteur la plus grande à la petite). TORA choisit les routes avec le moins de sauts et cela entraîne une surcharge des nœuds se trouvant sur le chemin principal. Si certains trajets comprennent plusieurs reprises du même nœud, celui-ci sera vidé de son énergie beaucoup plus tôt que les autres nœuds. Ainsi, l'utilisation des nœuds dans le plus court chemin, sans tenir compte leurs énergies conduit à la diminution de la durée de vie du réseau.

TORA a été conçu afin de minimiser la surcharge des liens de communication et donc de réduire la consommation d'énergie. Cependant, TORA ne tient pas compte de la multi-diffusion dans son fonctionnement de base. E-TORA [68] vient pour pallier à ce problème, dans ce protocole le chemin est sélectionné selon le nombre de sauts et l'énergie résiduelle des nœuds. Les nœuds qui ont plus d'énergie ont plus de probabilité d'être choisis. En outre, la consommation d'énergie des nœuds est plus équilibrée.

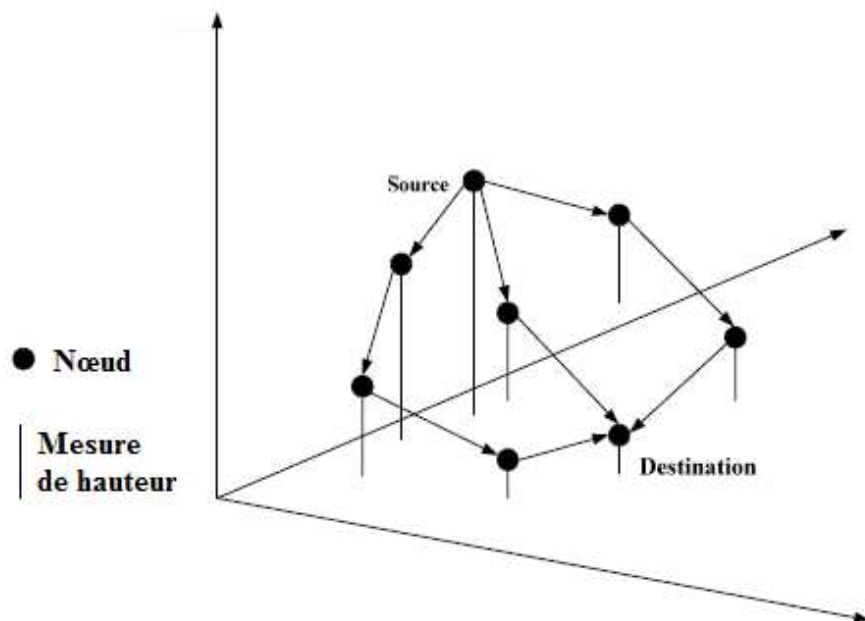


FIGURE 1.5 – Schéma du principe du protocole TORA [68].

Les catégories de protocoles précitées (proactifs et réactifs) présentent différentes caractéristiques, ci-dessous un tableau comparatif de celles-ci.

	Protocoles proactifs	Protocoles réactifs
Protocole à la demande		X
Mise à jour continue des chemins	X	
Retard d'acquisition de chemin	X	
Mises à jour périodiques	X	
Maintenir l'information de routage pour tous les nœuds du réseau	X	
Envoyer des messages de mise à jour lorsque la topologie change		X
Approprié pour les grandes charges	X	
Sporadique		X
Entraîne une grande surcharge	X	

TABLE 1.1 – Comparaison entre les protocoles proactifs et réactifs [47].

### C) Protocoles hybrides

C'est des protocoles qui combinent à la fois les avantages des protocoles proactifs et réactifs. Ils utilisent localement le routage proactif et inter-localement le routage réactif [47, 56]. Plusieurs protocoles ont été développés dans cette catégorie, nous pouvons citer à titre d'exemple ZRP (Zone Routing Protocol) [16].

- **ZRP :ZONE ROUTING PROTOCOL** [16]

Dans ce protocole, le réseau est divisé en zones. Ces dernières maintiennent de manière proactive la topologie de la zone. Cependant, il n'y a pas de changement de topologie dans tout le réseau, les nœuds voisins sont informés qu'à des intervalles réguliers.

Chaque nœud connaît toujours comment atteindre un autre nœud dans la même zone. Cela limite également le nombre de mises à jour déclenchées par un changement d'état des liens. Quand un nœud a besoin d'un chemin pour atteindre un nœud en dehors de sa zone, il envoie une *RREQ* (Route REQuest) à chaque *nœud frontière* de cette zone. À la réception d'un tel paquet, le nœud vérifie l'existence d'un chemin vers le nœud de destination dans sa table de routage intra-zone. S'il existe, un *RREP* (Route REPLY) est envoyé. Sinon, il renvoie *RREQ* à un autre nœud. Cette opération est répétée jusqu'à ce qu'un chemin soit trouvé.

Le principal avantage de ZRP est qu'il nécessite une petite quantité d'informations de routage pour chaque nœud, donc il produit beaucoup moins de trafic de routage. Cependant, il donne lieu à des retards et des frais généraux en raison des nombreux paquets de contrôle inutiles qui sont envoyés à travers le réseau. Par conséquent, la charge du réseau est augmentée et donc une diminution de la performance du réseau.

Le tableau ci-dessous fait la comparaison entre quelques protocoles plats selon quelques métriques comme l'évolutivité, la mobilité, le type de messages périodiques, etc .

Protocole	Évolutivité	Mobilité	Métrique de routage	Type de message périodique	Robustesse
WRP	Limitée	Limitée	Plus court chemin	Échange de table	Faible
TBRPF	Limitée	Bonne	Plus court chemin	Messages <i>Hello</i>	Bonne
TORA	Bonne	Bonne	Plus court chemin	Contrôle <i>IMEP</i>	Faible
Gossiping	Bonne	Bonne	Aléatoire	Aucun	Bonne
Flooding	Limitée	Faible	Plus court chemin	Aucun	Bonne
RR	Bonne	Faible	Plus court chemin	Messages <i>Hello</i>	Bonne
E-Tora	Bonne	Bonne	Meilleur chemin	Contrôle <i>IMEP</i>	Faible
ZRP	Limitée	Bonne	Meilleur chemin	Messages <i>Hello</i>	Bonne

TABLE 1.2 – Comparaison entre certains protocoles plats [47].

#### 1.4.1.2 Protocoles hiérarchiques

C'est des protocoles où les nœuds capteurs sont organisés logiquement en *groupes*, l'élection d'un *chef de groupe* se fait selon un algorithme bien précis. Les chefs de groupes sont utilisés pour une meilleure communication et afin de réduire la surcharge du trafic.

L'usage d'un routage hiérarchique tend à réduire le nombre des tables de routage et offre ainsi une meilleure mise à l'échelle. Les protocoles de cette catégorie sont classés par la façon dont les groupes sont organisés en protocoles à base de clusters, protocoles à base de chaînes et protocoles à base de grilles virtuelles [57].

## A) Protocoles à base de clusters

Dans cette organisation, le réseau en entier est divisé logiquement en plusieurs *clusters* selon une métrique spécifique ou une combinaison de métriques. Chaque cluster est géré par un *cluster-head* (**figure 1.6**). Le *cluster-head* joue le rôle d'agrégateur des données provenant des membres du cluster et transmet le résultat de l'agrégat à la SB. Plusieurs protocoles ont été développés dans cette catégorie, nous pouvons citer LEACH (Low Energy Adaptive Clustering Hierarchy) [23], HEED (Hybrid Energy Efficient Distributed) [67], TEEN (Threshold sensitive Energy Efficient sensor Network protocol) [40], etc. [2].

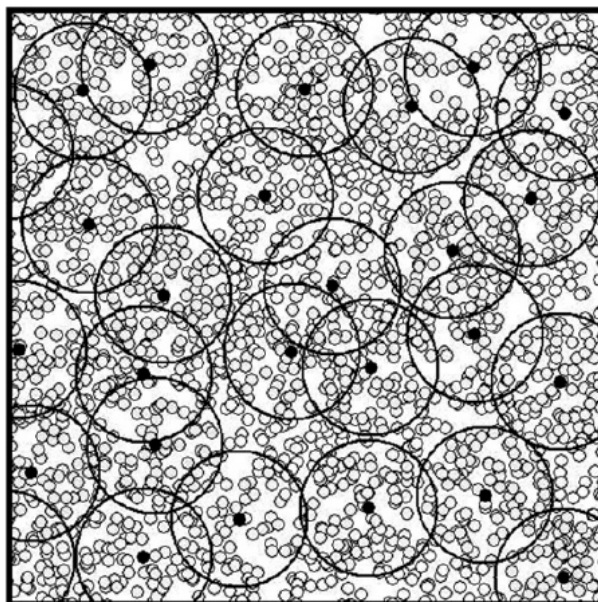


FIGURE 1.6 – Exemple d'un réseau partagé en clusters [2].

- **LEACH : LOW ENERGY ADAPTIVE CLUSTERING HIERARCHY** [23]

LEACH est l'un des algorithmes les plus populaires de clustering pour les réseaux de capteurs sans fil. Il forme des clusters en se basant sur la force du signal reçu et utilise les Cluster Heads comme routeurs à la station de base. Tout le traitement de données telles que la fusion et l'agrégation de données sont locales aux clusters. LEACH forme les clusters en utilisant un algorithme distribué où les nœuds prennent des décisions autonomes sans aucun contrôle centralisé. LEACH peut être décomposé en 2 phases : la phase de configuration et la phase de stabilité qui seront détaillées dans le chapitre suivant.

L'avantage de LEACH réside dans le fait qu'il surpasse les protocoles conventionnels en termes de consommation énergétique et facilité de configuration. Par contre, il

utilise le routage à saut unique où chaque nœud transmet les données directement au *Cluster-Head* puis à la station de base et donc non recommandable pour les réseaux déployés dans de vastes régions. Le clustering dynamique peut entraîner une surcharge au sein du réseau.

- **HEED : HYBRID ENERGY EFFICIENT DISTRIBUTED** [67]

Ce protocole ne dépend pas de la topologie du réseau ni de sa taille mais suppose que les nœuds peuvent modifier leur puissance de transmission (augmentation ou réduction). L'élection des Cluster-Heads (CHs) se fait à l'aide d'un critère hybride alliant l'énergie disponible au coût de la transmission. Il vise à distribuer de façon uniforme les CHs et à équilibrer la charge des clusters. Dans HEED, chaque nœud appartient à exactement un seul cluster et peut communiquer directement avec son CH. L'algorithme est divisé en trois phases :

- **La phase d'initialisation :**

L'algorithme définit d'abord un premier pourcentage de CHs auprès de tous les capteurs. La valeur de ce pourcentage  $C_{prob}$  est utilisée pour limiter les annonces CH initiales pour les autres capteurs. Chaque capteur définit sa probabilité de devenir un cluster-head  $CH_{prob}$  comme suit :

$$CH_{prob} = C_{prob} * \frac{E_{residual}}{E_{max}}, \quad (1.4)$$

où  $E_{residual}$ ,  $E_{max}$  sont respectivement l'énergie résiduelle et l'énergie maximale (ce qui correspond à une batterie complètement chargée) du capteur.  $CH_{prob}$  n'est pas autorisée à être au-dessous d'un certain seuil  $p_{min}$  qui est choisi pour être inversement proportionnel à  $E_{max}$ .

- **La phase de répétition :**

Pendant cette phase, chaque capteur passe plusieurs itérations jusqu'à ce qu'il trouve le CH auquel il peut transmettre avec le minimum d'énergie. S'il n'est à la portée d'aucun CH, le capteur lui-même choisit d'être un CH et envoie un message d'annonce à ses voisins les informant du changement de statut. Enfin, chaque capteur double sa valeur  $CH_{prob}$  et va à la prochaine itération de cette phase. Il arrête l'exécution de cette phase lorsque son  $CH_{prob}$  atteint 1. Par conséquent, il existe deux types de statuts que pourrait annoncer un capteur à ses voisins :

- *Statut provisoire* : Le capteur devient un CH provisoire si son  $CH_{prob}$  est inférieur à 1. Il peut devenir un nœud régulier à une itération plus tard s'il trouve un CH atteignable avec un coût minimal.
- *Statut final* : Le capteur devient définitivement un CH si son  $CH_{prob}$  à 1.

— **La phase de finalisation :**

Pendant cette phase, chaque capteur prend une décision finale sur son statut. Il choisit soit un CH qui atteindra avec un coût minimal ou se prononce lui-même en tant que CH.

Ding et al. [13] ont proposé DWEHC (Distributed Weight-Based Energy-Efficient Hierarchical Clustering) qui est une amélioration du protocole HEED. Les clusters générés par DWEHC sont plus équilibrés. Ce protocole réalise également moins de consommation d'énergie que HEED grâce à la communication intra et inter-cluster.

B) **Protocoles à base de grilles virtuelles**

La **figure 1.7** montre le principe de base des protocoles de cette catégorie qui est de diviser la zone à surveiller en grille virtuelle et de sélectionner un nœud leader pour chacune d'entre elles [57]. Plusieurs protocoles ont été développés dans cette catégorie, nous pouvons citer TTDD (Two-Tier Data Dissemination) [38], GBDD (Grid Based Data Dissemination) [54], etc.

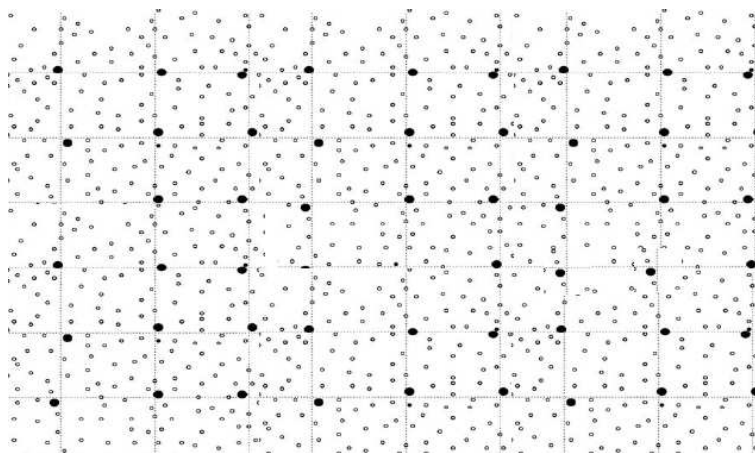


FIGURE 1.7 – Exemple d'un réseau partagé en une grille virtuelle [38].

- **TTDD : TWO-TIER DATA DISSEMINATION** [38]

Lorsqu'un événement est détecté, le nœud *source* s'occupe de structurer la grille virtuelle et se choisit comme *point de passage*. TTDD se base sur le fait que les nœuds sont stationnaires et que chacun est informé de la position des autres nœuds. À l'aide d'une diffusion géographique gloutonne, la source diffuse un message aux quatre (04) points de passages adjacents. Le message arrêtera sa course lorsqu'il atteindra un point proche du point de passage ou arrivera aux limites du réseau.

Le principal inconvénient de ce protocole est que chaque nœud de source construit une grille virtuelle pour transmettre les données à la SB.

### C) Protocoles à base de chaînes

Dans cette approche [57], les nœuds du réseau sont organisés de façon à former une grande chaîne de voisins proches où un seul nœud est sélectionné pour transmettre à la SB. La **figure 1.8** montre un exemple de construction d'une chaîne où  $S_7$  est le nœud *leader* chargé de transmettre les données à la SB [39]. Plusieurs protocoles ont été développés dans cette catégorie, nous pouvons citer GAF (Geographic Adaptive Fidelity) [64], PEGASIS (Power Efficient Gathering in Sensor Information System) [36], etc.

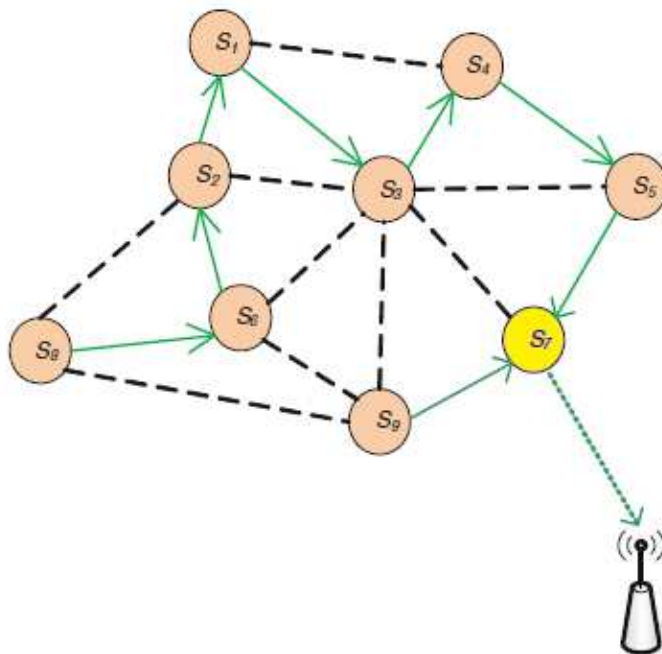


FIGURE 1.8 – Exemple d'un réseau utilisant une chaîne [39].

- **PEGASIS : POWER EFFICIENT GATHERING IN SENSOR INFORMATION SYSTEM** [36]

Considéré comme amélioration de LEACH, les nœuds dans PEGASIS forment une chaîne partant du principe que chaque nœud ne peut communiquer qu'avec son voisin le plus proche et ce jusqu'à l'arrivée des données à la station de base ce qui permet de réduire la consommation de l'énergie.

La chaîne peut être formée par les nœuds à l'aide d'un algorithme glouton ou bien par la SB. Pour la transmission de données, une approche à jeton est utilisée. Le leader envoie le jeton à la queue de la chaîne pour débiter la transmission. Les nœuds capteurs fusionnent les données reçues de leurs voisins proches avec les leurs et les transmettent, en plus du jeton au nœud suivant dans la chaîne. Le leader attend la réception des données de tous ses voisins avant de les agréger avec ses données et les transmettre à la SB.

PEGASIS réduit la consommation d'énergie en minimisant le trafic de données grâce à l'agrégation. Cependant, il introduit des retards excessifs pour les nœuds lointains dans la chaîne.

Le tableau ci-dessous est une comparaison de quelques protocoles hiérarchiques selon un certain nombre de paramètres d'évolutivité, mobilité, métrique de routage, type de messages périodiques ainsi que la robustesse.



Protocole	Évolutivité	Mobilité	Métrieque de rou- tage	Type de message périodique	Robustesse
LEACH	Bonne	SB fixée	Plus court chemin	Aucun	Bonne
LEACH-C	Bonne	SB fixée	Meilleur chemin	Aucun	Bonne
PEGASIS	Bonne	SB fixée	Sélection gloutonne de chemin	Aucun	Bonne
TEEN	Bonne	SB fixée	Meilleur chemin	Aucun	Limitée
APTEEN	Bonne	SB fixée	Meilleur chemin	Contrôle <i>IMEP</i>	Bonne
VGA	Bonne	Non	Sélection gloutonne de chemin	Aucun	Bonne
TTDD	Faible	Non	Sélection gloutonne de chemin	Aucun	Bonne
BCDCP	Limitée	Non	Meilleur chemin	Aucun	Limitée
MIMO	Bonne	Non	les données de multiples source seront transmises en multi-sauts au puits	Aucun	Limitée
HPAR	Faible	Non	Sélectionne le plus court chemin en se basant sur la consommation d'énergie	Aucun	Bonne
Sleep/Wake	Bonne	Non	Meilleur chemin	Aucun	Limitée
GBDD	Bonne	Limitée	Dans la présence d'une grille valide, le puits découvre le nœud du coin le plus proche	Aucun	Bonne
ELCH	Limitée	SB fixée	Sélectionne le nœud qui a le maximum d'éner- gie restante	Aucun	Bonne
NHRPA	Bonne	SB fixée	Meilleur chemin	Aucun	Bonne
SHPER	Bonne	SB fixée	Meilleur chemin	Aucun	Bonne
DHAC	Bonne	Non	Meilleur chemin	Messages <i>Hello</i>	Limitée

TABLE 1.3 – Comparaison entre certains protocoles hiérarchiques [47].

## 1.4.2 Modèle de communication

C'est la manière dont l'opération principale du protocole est suivie afin d'acheminer les données. L'avantage de cette méthode est qu'il y a un plus grand acheminement pour une quantité donnée d'énergie et que le débit de diffusion et le rendement énergétique sont proches de l'optimum théorique dans une communication point à point ou multi-points. Par contre, il n'y a pas de garantie de l'acheminement des paquets. Les protocoles de cette catégorie peuvent être classés comme basés sur les requêtes, basés cohérent non cohérent et basés sur les négociations [39].

### 1.4.2.1 Protocoles à base de requêtes

Dans ces protocoles, les nœuds de destination propagent une requête pour les données à partir d'un nœud du réseau et le nœud ayant cette donnée envoie les données qui lui correspondent vers le nœud qui a initié la requête. Ces requêtes sont généralement décrites en langage naturel ou en langages de requêtes de haut niveau. Tous les nœuds ont des tables constituées de requêtes des tâches de détection qu'ils reçoivent et envoient des données qui correspondent à ces tâches [39]. Plusieurs protocoles ont été développés dans cette catégorie, nous pouvons citer DD (Directed Diffusion) [26], COUGAR [65], ACQUIRE (ACTIVE QUery forwarding In sensoR nEtworks) [53], etc.

- **DD : DIRECTED DIFFUSION** [26]

L'idée de base de ce protocole est de diffuser des données à travers les nœuds de capteurs en utilisant un schéma de nomination pour les données. La principale raison de l'utilisation d'un tel principe est de se débarrasser des opérations inutiles de la couche réseau afin d'économiser l'énergie. C'est un protocole centré sur les données, qui se fait en 3 phases [47] :

- **Phase 1** : La station de base émet une requête sous forme d'*intérêt* (incluant le type de la tâche, la zone ciblée, etc.) et la diffuse à son voisinage (voir **figure 1.9 (A)**).
- **Phase 2** : Si un nœud n'est pas intéressé par l'*intérêt* reçu, il l'enregistre ainsi que le voisinage du nœud expéditeur, lorsqu'il reçoit les données recherchées il les envoie au dernier nœud avant le nœud expéditeur. C'est l'établissement du *gradient*. Celui-ci étant un vecteur représentant l'*intérêt* et est caractérisé par une direction et une amplitude (voir **figure 1.9 (B)**).
- **Phase 3** : Lorsqu'un nœud reçoit l'*intérêt*, il fortifie les chemins menant à la destination, il transmet ensuite via ceux qui ont un gradient maximal, si l'un d'eux échoue il transmettra via ceux dont le *gradient* est plus faible. L'accomplissement de ces phases contribue à la stabilité de la diffusion dirigée (voir **figure 1.9 (C)**).

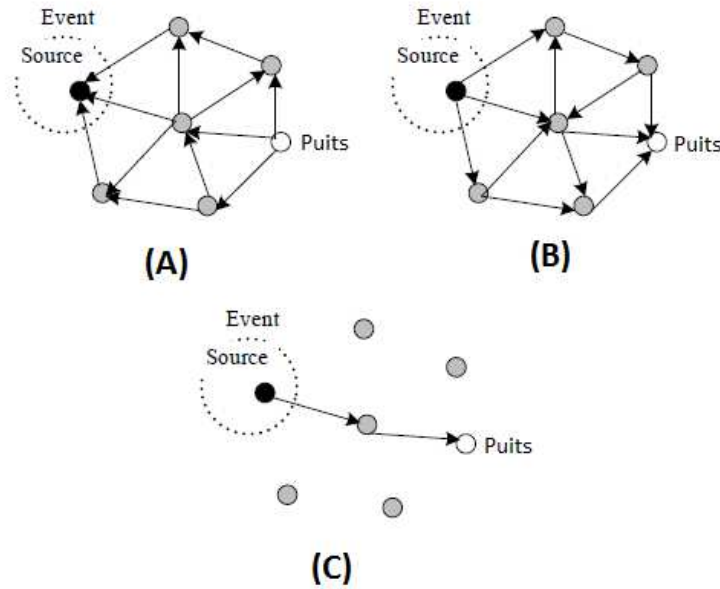


FIGURE 1.9 – Illustration du protocole de la diffusion dirigée [26].

Le tableau ci-dessous compare quelques protocoles de cette catégorie en se basant sur les paramètres d'évolutivité, mobilité, métrique de routage, type de message périodiques et robustesse.

Protocole	Évolutivité	Mobilité	Métrique de routage	Type de message périodique	Robustesse
DD	Bonne	Limitée	Meilleur chemin	Messages de requêtes	Faible
COUGAR	Limitée	Non	Meilleur chemin	Messages de requêtes	Faible
ACQUIRE	Limitée	Limitée	Plus court chemin	Messages de requêtes	Faible

TABLE 1.4 – Comparaison entre certains protocoles à base de requêtes [47].

#### 1.4.2.2 Protocoles basés sur la cohérence et la non-cohérence

Dans les RCSF le traitement des données est nécessaire au niveau des nœuds. Les nœuds de capteurs collaborent pour le traitement de données. Le mécanisme de routage qui déclenche le module de traitement de données est proposé dans [59]. Beaucoup de protocoles ont été développés dans cette catégorie, nous citerons par exemple SWA (Single Winner Algorithm) [9], MWE (Multiple Winner Algorithm) [9], etc. Le mécanisme adopté dans les protocoles décrits dans cette

partie est divisé en deux (02) catégories :

- **Routage cohérent basé traitement de données** : C'est un mécanisme efficace pour l'énergie où seulement le minimum de traitement est effectué par les nœuds de capteurs. Marquage temporel, suppression des messages dupliqués sont les tâches accomplies dans le traitement minimum. Après le traitement minimum, les données sont transmises aux agrégateurs.
- **Routage non-cohérent basé traitement de données** : Dans cette catégorie, les nœuds de capteurs traitent localement les données réelles puis les envoient vers les autres nœuds pour la poursuite du traitement. Les nœuds qui effectuent un traitement ultérieur sont appelés les agrégateurs. Il y a trois phases du traitement de données dans le routage non-cohérent. (a) La détection de cible, la collecte de données et le pré-traitement, (b) la déclaration d'adhésion et (c) l'élection du nœud central.

- **SWA : SINGLE WINNER ALGORITHM [9]**

Dans cet algorithme un seul nœud agrégateur est élu pour un traitement complexe. Ce nœud est appelé *CN*(Central Node). Il est choisi en fonction de sa réserve d'énergie et sa capacité de calcul. Afin de sélectionner le *CN*, chaque nœud diffuse un message où il annonce qu'il est lui-même un candidat *CN*. En réponse au premier lot de messages, les nœuds qui ont déjà reçu, vont commencer à comparer les candidats proposés avec le *CN* actuel et répondent avec un deuxième lot de messages qui porte le résultat de cette comparaison initiale. Le second lot de messages va générer plus d'échanges. Le message qui présente le meilleur candidat est enregistré dans le registre et peut ensuite être transmis à tous les voisins, sinon le message est rejeté.

- **MWA : MULTIPLE WINNER ALGORITHM [9]**

C'est une extension de SWE, lorsque tous les nœuds envoient leurs données au nœud central(chargé de l'agrégation), une grande quantité d'énergie est consommée, elle pourrait être plus faible s'il y avait une limite du nombre de nœuds qui envoient au nœud central. Au lieu de tenir un registre pour le meilleur nœud candidat uniquement, chaque nœud tiendra un registre allant jusqu'à  $n$  nœuds de ces candidats. Chaque nœud de capteur dans le processus de MWE a un ensemble de chemins avec un minimum d'énergie pour chaque nœud Source (SN). SWE est utilisé pour trouver le nœud qui donne le minimum de consommation d'énergie. Ce nœud peut alors servir comme nœud central de traitement cohérent.

Le tableau ci-dessous compare les protocoles de cette catégorie en se basant sur les paramètres d'évolutivité, mobilité, métrique de routage, type de message périodique et robustesse.

Protocole	Évolutivité	Mobilité	Métrique de rou- tage	Type de message périodique	Robustesse
SWE	Bonne	Non	Plus court chemin	Messages <i>Hello</i>	Faible
MWE	Limitée	Non	Plus court chemin	Messages <i>Hello</i>	Faible

TABLE 1.5 – Comparaison entre certains protocoles basés sur la cohérence et la non-cohérence [47].

### 1.4.2.3 Protocoles à base de négociations

Les protocoles de cette catégorie font appel à des *métadonnées* (descriptions sur les données) afin d'éliminer la redondance des données lors des négociations et les décisions liées à la communication dépendent des ressources disponibles. Plusieurs protocoles ont été développés dans cette catégorie, nous pouvons citer SPIN (Sensor Protocols for Information via Negotiation) [24], SPIN-PP (SPIN for Point to Point Communication) [24], etc.

- **SPIN : SENSOR PROTOCOLS FOR INFORMATION VIA NEGOTIATION** [24]

SPIN a été proposé pour pallier aux problèmes du flooding et s'adapte aux besoins en termes de ressources. Chaque nœud prend connaissance de sa consommation de ressources et cela grâce à son manager des ressources (choisi par l'application avant toute transmission/traitement). SPIN possède trois (03) types de messages (**figure 1.10** [47]) :

- **ADV** : pour l'échange de métadonnées.
- **REQ** : pour la demande de données.
- **DATA** : pour l'envoi des données.

Si un nœud a des données à envoyer, tout d'abord, il envoie un paquet *ADV* les nœuds ayant reçus et s'ils sont intéressés envoient un paquet de requête *REQ* et lors de la réception, le nœud envoie les données réelles.

L'avantage de SPIN, c'est que les nœuds ne connaissent que leur prochain saut et donc l'adaptation aux changements de topologie est assurée. Par contre, les *métadonnées* introduisent un coût supplémentaire pour leur stockage, maintien et suppression, de plus il n'y a pas de garantie de livraison pour les données, les nœuds peuvent ne pas être intéressés par les données reçus et donc ne transfèrent pas à la destination.

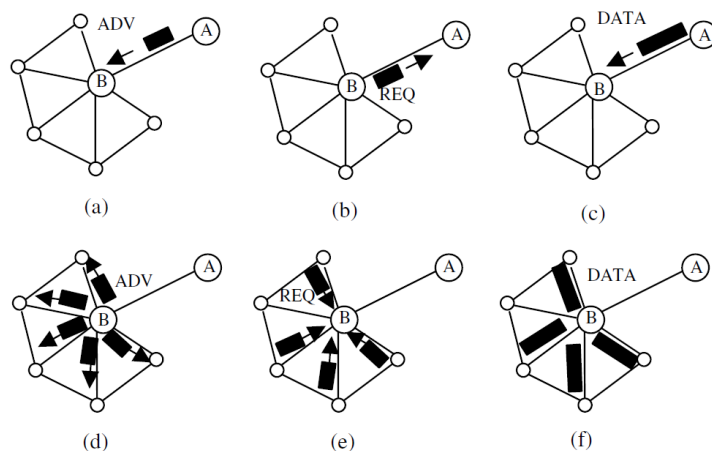


FIGURE 1.10 – Illustration du protocole SPIN [47].

Le tableau ci-dessous compare quelques protocoles de cette catégorie en se basant sur l'évolutivité, mobilité, métrique de routage, type de message périodique et robustesse.

Protocole	Évolutivité	Mobilité	Métrique de routage	Type de message périodique	Robustesse
SPIN-PP	Bonne	Oui	Chaque nœud envoie les données aux voisins du prochain saut	nœuds ayant les données l'annonce à tous se voisins	Bonne
SPIN-EC	Bonne	Oui	Chaque nœud envoie les données aux voisine du prochain saut	nœuds ayant les données l'annonce à tous se voisins	Bonne
SPIN-BC	Bonne	Oui	Chaque nœud envoie les données aux voisine du prochain saut	nœuds ayant les données l'annonce à tous se voisins	Bonne
SPIN-RL	Bonne	Oui	Chaque nœud envoie les données aux voisine du prochain saut	nœuds ayant les données l'annonce à tous se voisins	Bonne

TABLE 1.6 – Comparaison entre certains protocoles à base de négociations [47].

### 1.4.3 Topologie

Les protocoles de cette catégorie utilisent le principe que chaque nœud dans le réseau conserve les informations de topologie et que le procédé de l'opération de ces protocoles est basé sur celles-ci. Ils peuvent être classés comme protocoles basés sur la localisation et protocoles basés sur les agents mobiles [47].

#### 1.4.3.1 Protocoles basés sur la localisation

Ce type de protocoles reconnaît l'influence de la distance et la distribution des nœuds sur les performances du réseau. Il est basé sur deux hypothèses principales :

- Chaque nœud connaît les positions de ses réseaux voisins.
- La source d'un message est supposée être informée de la position de la destination.

Cette méthode fonctionne sans tables de routage car une fois que la position de la destination est connue, toutes les opérations sont strictement locales, chaque nœud doit garder une trace de son voisin direct uniquement. Cependant, l'efficacité dépend de la répartition géographique équilibrée par rapport à l'apparition du trafic. Plusieurs protocoles ont été développés dans cette catégorie, nous pouvons citer DREAM (Distance Routing Effect Algorithm for Mobility) [8], GEAR (Geographic and Energy Aware Routing) [69], GEM (Graph Embedding for Routing) [45], etc.

#### • **DREAM :DISTANCE ROUTING EFFECT ALGORITHM FOR MOBILITY** [8]

C'est un protocole proactif dont chaque nœud mobile  $MN$  maintient une table de localisation de tous les autres nœuds dans le réseau. Pour maintenir la table, chaque  $MN$  transmet l'emplacement des paquets aux  $MNs$  voisins dans le réseau via une fréquence donnée et aux  $MNs$  lointains via une autre fréquence plus faible. Les  $MNs$  lointaines semblent se déplacer plus lentement que les  $MNs$  voisins, il n'est pas nécessaire pour un  $MN$  de maintenir à jour le lieu de l'information pour les  $MNs$  lointains. Ainsi, en différenciant  $MNs$  entre voisins et lointains, DREAM tente de limiter la surcharge des paquets de localisation.

Lorsqu'un nœud  $S$  veut envoyer un message  $m$  à un nœud destinataire  $R$ , il se réfère à sa table de localisation afin de récupérer sa localisation. Après cela, il sélectionne à partir de  $S$  parmi ses voisins les nœuds qui sont dans la direction de  $R$  et leur transmet  $m$ . Le processus se répète jusqu'à l'arrivée à  $R$ . Il est important de bien choisir les nœuds pour la diffusion, afin de s'assurer que le message va être délivré avec une probabilité  $P$ . L'avantage de ce protocole est qu'il assure la transmission efficace des données, car une route bout à bout est toujours disponible. Cependant, la bande passante est gaspillée et il y a des dépenses

énergétiques inutiles à cause de la réception des mêmes messages plusieurs fois.

le tableau ci-dessous compare les protocoles de cette catégorie en se basant sur l'évolutivité, la mobilité, métrique de routage, type de message périodique et robustesse.

Protocole	Évolutivité	Mobilité	Métrique de routage	Type de message périodique	Robustesse
DREAM	Limitée	Bonne	Chemin minimisant la consommation totale d'énergie	Messages de contrôles	Limitée
GEAR	Limitée	Limitée	Meilleur chemin	Messages <i>Hello</i>	Bonne
GEM	Bonne	Limitée	Plus court chemin	Aucun	Bonne
IGF	Limitée	Bonne	Meilleur chemin	Aucun	Bonne
SELAR	Limitée	Limitée	Chemin incluant les nœuds ayant le maximum d'énergie	Messages de contrôles	Bonne
GDSTR	Limitée	Non	Plus court chemin	Messages <i>Hello</i>	Bonne
MERR	Limitée	Faible	Chemin minimisant la consommation totale d'énergie	Aucun	Bonne
OGF	Bonne	Limitée	Meilleur chemin	Aucun	Bonne
PAGER-M	Bonne	Bonne	Plus court chemin en utilisant un algorithme glouton	Messages <i>Hello</i>	Bonne
HGR	Bonne	Bonne	Chemin minimisant la consommation totale d'énergie	Aucun	Bonne

TABLE 1.7 – Comparaison entre certains protocoles basés sur la localisation [47].



### 1.4.3.2 Protocoles basés sur les agents mobiles

Les systèmes d'agents mobiles emploient des codes de migration afin de faciliter la transformation locale et la collaboration pour le traitement du signal et de l'information. Cela peut fournir au réseau une flexibilité supplémentaire, ainsi que de nouvelles capacités par rapport aux RCSF classiques qui sont basés sur le modèle client-serveur [47]. Plusieurs protocoles ont été développés dans cette catégorie, nous pouvons citer à titre d'exemple MIP (Multi-agent based Itinerary Planning) [11].

- **MIP : MULTI-AGENT BASED ITINERARY PLANNING** [11]

Cet algorithme utilise SIP (Single Agent Based Itinerary Planning) dans un grand réseau, cela peut conduire à des retards et à une charge déséquilibrée. Ainsi, l'utilisation de MIP est plus efficace. L'idée de base de l'algorithme est de distribuer le facteur d'impact de chaque source à d'autres nœuds de source. Par exemple, en considérant que  $N$  est le nombre de sources, chaque nœud source recevra  $n - 1$  facteurs d'impact provenant d'autres nœuds et un de lui-même. Après que le facteur d'impact est accumulé et calculé et l'emplacement de la source avec le plus grand facteur d'impact accumulé est sélectionné.

Les résultats de simulation montrent que la consommation d'énergie de MIP est plus élevée que celle de SIP dans le cas où le numéro de source est petit. Cependant, ce protocole est conçu pour une utilisation lorsque le nombre de sources est grand. Ainsi, sur la base des résultats lorsque le numéro de la source est de 40, la consommation d'énergie de MIP est bien meilleure que celle des algorithmes SIP.

le tableau ci-dessous compare tous les protocoles de cette catégorie en se basant sur l'évolutivité, mobilité, métrique de routage, type de message périodique et robustesse.

Protocole	Evolutivité	Mobilité	Métrique de routage	Type de message périodique	Robustesse
MIP	Limitée	Bonne	Chemin minimisant la consommation totale d'énergie	Aucun	Bonne
IEMF/IEMA	Limitée	Bonne	Chemin minimisant l'énergie totale	Aucun	Bonne

TABLE 1.8 – Comparaison entre certains protocoles basés sur les agents mobiles [47].

## 1.4.4 Fiabilité du routage

Les protocoles de cette dernière catégorie sont plus résistants aux échecs de routes, puisqu'ils visent à équilibrer la charge en la dispersant sur les différents chemins ou bien en satisfaisant certains paramètres de qualité de service comme le retard, l'énergie et la bande passante. Les nœuds capteurs du réseau peuvent souffrir de la surcharge dû au maintien des tables de routage ainsi que les paramètres de qualité de service [47].

### 1.4.4.1 Protocoles basés sur le multi-chemins

Le routage multi-chemins a l'avantage d'atteindre l'équilibrage de charge et d'être plus résistant aux échecs de route. Il existe plusieurs protocoles de routage multi-chemins et les évaluations de leurs rendements montrent qu'ils peuvent atténuer la surcharge du trafic au sein du réseau en comparaison avec les protocoles de routage à chemin unique. Nous décrivons ci-dessous les protocoles de routages de cette catégorie. Plusieurs protocoles ont été développés dans cette catégorie, nous pouvons citer ROAM (Routing On-demand Acyclic Multipath) [51], LMR (Label-based Multipath Routing) [25], GRAB (GRAdient Broadcast) [66], etc.

- **GRAB : GRADIENT BROADCAST** [66]

Il a été conçu pour la livraison de données robuste pour faire face aux nœuds et liaison non fiables. Il construit et maintient un champ de coût en propageant les paquets publicitaire *ADV* dans le réseau. Dès qu'un nœud reçoit un paquet *ADV* contenant le coût de l'expéditeur, il calcule son coût en ajoutant le coût du lien entre lui-même et l'expéditeur. Il compare ce coût à celui précédemment enregistré et définit le nouveau coût comme le plus petit des deux. Comme il obtient un coût inférieur à l'ancien, il diffuse un paquet *ADV* contenant le nouveau coût.

L'effort collectif de plusieurs nœuds pour fournir des données sans dépendance sur tous les nœuds du réseau est l'un des points forts de GRAB. Cependant, il peut avoir des frais généraux par l'envoi des données redondantes.

le tableau ci-dessous compare tous les protocoles de cette catégorie en se basant sur l'évolutivité, mobilité, métrique de routage, type de message et robustesse.

Protocole	Evolutivité	Mobilité	Métrique de rou- tage	Type de message périodique	Robustesse
ROAM	Limitée	Limitée	Peu importe le che- min	Messages <i>Hello</i>	Limitée
LMR	Bonne	Bonne	Peu importe le che- min	Aucun	Bonne
GRAB	Bonne	Bonne	Ensemble disjoint de chemins satisfai- sant les exigences de QoS	Messages <i>Hello</i>	Bonne
HMRP	Bonne	Faible	Peu importe le che- min	Aucun	Limitée
CBMPR	Limitée	Faible	Meilleur chemin	Messages <i>Hello</i>	Limitée
DGR	élevée	Non	Chemin avec un voisinage initial différent	Aucun	élevée
DCF	élevée	élevée	Meilleur chemin	Aucun	Bonne
RPL	Bonne	Bonne	Plus court chemin	Messages <i>DIO</i>	Bonne

TABLE 1.9 – Comparaison entre certains protocoles basés sur le multi-chemins [47].

#### 1.4.4.2 Protocoles basés sur la qualité de service (QoS)

Dans cette catégorie le réseau doit équilibrer entre le consommation d'énergie et la qualité des données. En particulier, le réseau doit satisfaire certains métriques de qualité de service (délai, énergie, bande passante, ect.) lors de la transmission des données à la station de base. Plusieurs protocoles ont été développés dans cette catégorie, nous pouvons citer SAR (Sequential Assignment Routing) [60], SPEED [19], MGR (Multimedia Geographic Routing) [12], etc.

- **SAR : SEQUENTIAL ASSIGNMENT ROUTING** [60]

SAR est l'un des premiers protocoles à introduire la qualité de service dans le routage pour contourner le problème de l'échec d'un chemin (mort ou panne de capteurs). SAR utilise une approche multi-chemins et un schéma de restauration. Les décisions de routage sont liées à 3 facteurs : la ressource énergétique de chaque chemin, la qualité de service offerte par chaque chemin et le niveau de priorité de chaque paquet.

Pour pouvoir construire un chemin menant de la source à la destination SAR utilise un arbre enraciné à la source, couvrant l'ensemble des nœuds, il calcule le produit des métriques

de qualité de service supplémentaires et le poids des coefficients associés aux priorités des paquets, SAR vise à minimiser la moyenne du produit tout au long de la durée de vie du réseau. À mesure préventive, la station de base recalcule les chemins en cas de changements de topologie. Le recouvrement lors d'une défaillance se fait en harmonisant les tables de routage des nœuds se trouvant aux extrémités de chaque chemin.

Les résultats de simulation ont montré que SAR offre une consommation minimale de l'énergie car à l'instar des autres algorithmes il prend en compte la priorité des paquets, réalise la tolérance aux fautes et gère les défaillances. Cependant, le maintien des tables et des états de chaque nœud cause une surcharge surtout lorsque le nombre des nœuds est important.

Le tableau ci-dessous compare les protocoles de cette catégorie en se basant sur l'évolutivité, mobilité, métrique de routage, type de message périodique et robustesse.

Protocole	Evolutivité	Mobilité	Métrique de routage	Type de message périodique	Robustesse
SAR	Limitée	Non	Chemin minimisant la moyenne des métriques de qualité de service tout au long de la durée de vie du réseau	Messages <i>Hello</i>	Faible
SPEED	Limitée	Non	Chemin indéterminé sans état géographique	Messages <i>Hello</i>	Faible
MMSPEED	Limitée	Non	Chemin indéterminé sans état géographique	Messages <i>Hello</i>	Faible
MGR	Bonne	Bonne	Chemin minimisant les délais	Aucun	Faible

TABLE 1.10 – Comparaison entre certains protocoles basés sur la QoS [47].

## 1.5 Conclusion

Ce chapitre a été axé sur la présentation globale des réseaux de capteurs sans fil et leurs domaines d'application. Ensuite, le mécanisme du routage de données dans ces réseaux a été introduit ainsi que les contraintes liées à la conception de protocoles de routage. Enfin, une taxonomie des

protocoles de routage a été établie en se basant sur la classification proposée par *Al-Karaki et al.*. Une description détaillée du protocole LEACH et de certaines de ses améliorations feront l'objet du chapitre suivant.

# Chapitre 2

## Approche de clustering dans les réseaux de capteurs sans fil

### 2.1 Introduction

Le clustering a été largement abordé par la communauté de recherche académique en vue d'atteindre l'objectif d'évolutivité du réseau. L'approche du clustering repose sur la formation de groupes, appelés *clusters* et l'élection d'un chef par groupe ; souvent désigné comme *cluster head* (CH). Bien que de nombreux algorithmes de clustering ont été proposés pour les réseaux *ad-hoc* [10, 29] leur objectif principal était de générer des groupes stables dans des environnements avec des nœuds capteurs mobiles. Un bon nombre de ces algorithmes se soucient surtout de l'accessibilité des nœuds capteurs et la stabilité des chemins qu'aux contraintes liées à leur conception expliquées dans le chapitre précédent.

Récemment, plusieurs algorithmes de clustering ont été spécifiquement conçus pour les RCSF [7, 15]. Ces algorithmes varient largement en fonction de certains paramètres tels que le déploiement des nœuds capteurs, l'architecture du réseau, les caractéristiques des nœuds *CH*, etc. Un *CH* peut être élu par les capteurs dans un cluster ou pré-attribué par la station de base. Un *CH* peut aussi être l'un des nombreux nœuds capteurs ou le nœud le plus riche en ressources. L'appartenance à un cluster peut être fixe ou variable. Les *CHs* peuvent former un second réseau ou peuvent simplement transférer les données aux parties intéressées, par exemple une station de base ou un centre de commandement.

Dans ce chapitre, nous allons expliciter l'apport du clustering dans les protocoles de routage, ensuite définir les paramètres nécessaires au clustering pour enfin présenter en détails l'algorithme LEACH [23] ainsi que certaines de ses améliorations.

## 2.2 Apport du clustering

Les protocoles de clustering proposés dans la littérature diffèrent dans leurs objectifs. Généralement, le clustering vise à faciliter la réponse aux exigences des applications, par exemple si l'application est sensible à la latence des données. La connectivité intra et inter-cluster et la durée de l'acheminement des données sont généralement considérées comme des critères pour la formation de clusters. Les objectifs du clustering les plus connus sont [2] :

### 2.2.1 Équilibrage de charge

Il est intuitif d'équilibrer la charge entre les fonctions des CHs afin qu'ils puissent répondre aux objectifs de performances. L'équilibrage de charge est une question importante dans les RCSF où les CHs sont choisis à partir des nœuds capteurs du réseau. Dans ce cas, la construction des clusters de taille égale devient cruciale pour étendre la durée de vie du réseau. Lorsque les CHs exercent une agrégation de données, il est impératif d'avoir le même nombre de nœuds dans les clusters de sorte que le rapport de données combiné soit transmis presque en même temps pour un traitement ultérieur à la station de base (SB) ou au niveau suivant dans le réseau.

### 2.2.2 Tolérance aux pannes

Dans de nombreuses applications, les RCSF sont utilisés dans des environnements hostiles et donc sont généralement exposés à un risque accru de dysfonctionnement. Tolérer l'échec des CHs est généralement nécessaire dans de telles applications afin d'éviter la perte des données importantes. La façon la plus intuitive de récupération après l'échec d'un CH est de reconstruire les clusters à nouveau. Cependant, le clustering est non seulement une surcharge pour les nœuds capteurs, il est aussi perturbateur de l'opération en cours. Par conséquent, les techniques contemporaines de tolérance aux pannes seraient plus appropriées. Attribution de sauvegarde aux CHs est le régime le plus utilisé dans la littérature pour la récupération après l'échec du CH [17]. La rotation du rôle des CHs dans le cluster peut aussi être un moyen pour la tolérance aux pannes en plus à son avantage d'équilibrage de charge.

### 2.2.3 Augmentation de connectivité et réduction des délais

À moins que les CHs aient des capacités de communication à longues distances. Par exemple, une liaison satellite, connectivité inter-CH qui est une importante exigence dans de nombreuses applications. L'objectif de la connectivité peut être seulement limité à assurer la disponibilité d'un chemin à partir de chaque CH à la station de base ou être plus restrictif en imposant une limite sur la longueur du chemin. D'autre part, lorsque le temps d'attente des données est un point essentiel, la connectivité intra-cluster devient un objectif de conception. Le délai est généralement pris en compte par la fixation d'un nombre maximum de saut " $k$ " permis sur un chemin de données.

## 2.2.4 Nombre minimal de clusters

Cet objectif caractérise les RCSF possédant des CHs riches en ressources. Le concepteur du réseau emploie souvent moins de CHs puisqu'ils ont tendance à être plus chers et plus vulnérables que les autres nœuds capteurs. La limitation du nombre de CHs peut être due à la complexité du déploiement, mais aussi à la facilité de leur détection étant donné que leur taille a tendance à être beaucoup plus importante que les autres nœuds. Or, la visibilité des nœuds est hautement indésirable dans de nombreuses applications telles que la protection des frontières, les applications militaires, etc.

## 2.2.5 Longévité maximale du réseau

La durée de vie du réseau est une préoccupation majeure, en particulier pour les applications de RCSF dans les environnements difficiles. Lorsque les CHs sont plus riches en ressources que les autres nœuds capteurs, il est impératif de minimiser l'énergie pour la communication intra-cluster. Si possible, les CHs doivent être placés à proximité des capteurs dans le groupe. D'une autre part, lorsque les CHs sont des nœuds capteurs réguliers, leur durée de vie peut être étendue en limitant leur charge. Le clustering adaptatif est l'un des solutions les plus connues pour la maximisation de la durée de vie d'un réseau.

## 2.3 Paramètres du clustering

L'ensemble des paramètres qui peuvent être utilisés pour catégoriser et différencier les différents algorithmes de clustering dans les réseaux de capteurs sans fil sont énumérés dans cette section. Dans [2] les paramètres suivants ont été présentés :

### 2.3.1 Propriétés des clusters

Les systèmes de clustering cherchent souvent à assurer certaines caractéristiques (voir **figure 2.1**) pour les clusters générés. Celles-ci peuvent être liées à la structure interne du cluster ou à la façon dont ils sont reliés les uns aux autres. Les propriétés suivantes sont les plus pertinentes [2] :

#### 2.3.1.1 Nombre de clusters

Dans certains protocoles l'ensemble des CHs est prédéterminé et donc le nombre de clusters est prédéfini. Choisir au hasard des CHs à partir des capteurs déployés donne habituellement un nombre variable de clusters.

#### 2.3.1.2 Stabilité

Lorsque le nombre des clusters varie et les membres d'un groupe évoluent constamment, le régime du cluster est dit *adaptatif*. Dans le cas contraire, on le considère fixe puisque les capteurs



ne migrent pas d'un cluster à un autre et donc le nombre de clusters reste le même tout au long de la durée de vie du réseau.

### 2.3.1.3 Topologie intra-cluster

Certains systèmes de clustering sont basés sur la communication directe entre un capteur et son CH désigné. Cependant, la connectivité multi-sauts entre un capteur et son CH est parfois nécessaire, en particulier lorsque la portée des capteurs est limitée et/ou le nombre de CHs est borné.

### 2.3.1.4 Connectivité inter-cluster heads

Lorsque le CH n'a pas de capacités de communication à long terme, la connectivité des CHs à la station de base (SB) doit être provisionnée. Dans ce cas, le schéma de clustering doit veiller à la faisabilité de l'établissement d'un chemin inter-CH à partir de chaque CH à la SB. Certains travaux publiés supposent que le CH est en mesure d'accéder directement à la SB.

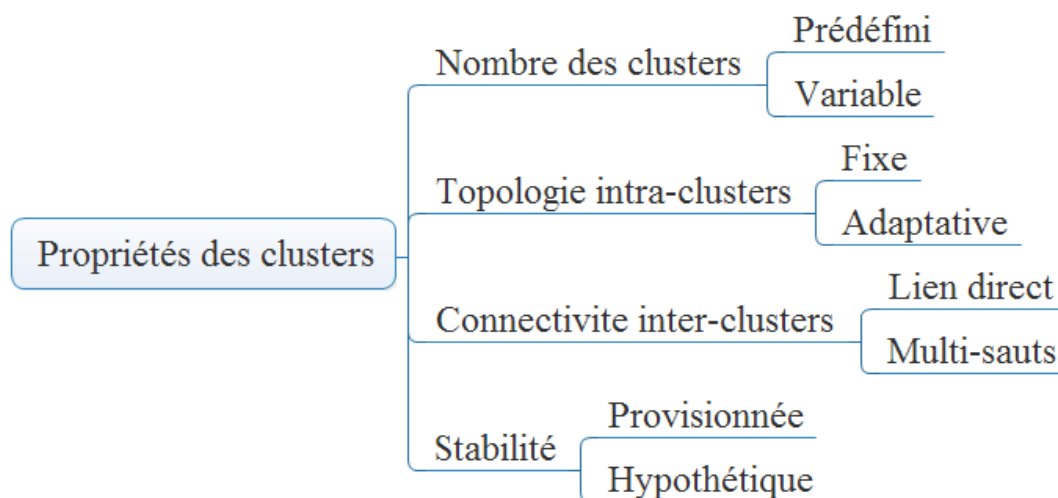


FIGURE 2.1 – Propriétés des clusters [2].

## 2.3.2 Capacités des cluster heads

Le modèle du réseau influence le clustering, particulièrement les capacités spécifiques de chaque nœud capteur comme la portée de transmission et le traitement de données dans le réseau. La **figure 2.2** illustre ces différents facteurs (**figure 2.2**) [2] :

### 2.3.2.1 Mobilité

Quand un CH est mobile, l'adhésion des capteurs change dynamiquement et les clusters auraient besoin d'être maintenus en permanence. D'autre part, un CH stationnaire tend à produire des

clusters plus stables et à faciliter l'intra et l'inter gestion du cluster. Parfois, les CHs peuvent migrer sur des distances limitées afin d'assurer une meilleure performance du réseau.

### 2.3.2.2 Types de nœuds

Dans certaines configurations, un sous-ensemble de capteurs déployés est désigné comme CHs. Tandis que dans d'autres, les CHs sont équipés de manière significative de plus de ressources (Énergie, puissance du calcul, etc.).

### 2.3.2.3 Rôle

Un CH peut simplement agir comme un relais pour le trafic généré par les capteurs dans son cluster ou effectuer l'agrégation/la fusion de données recueillies. Il peut aussi agir comme une station de base.

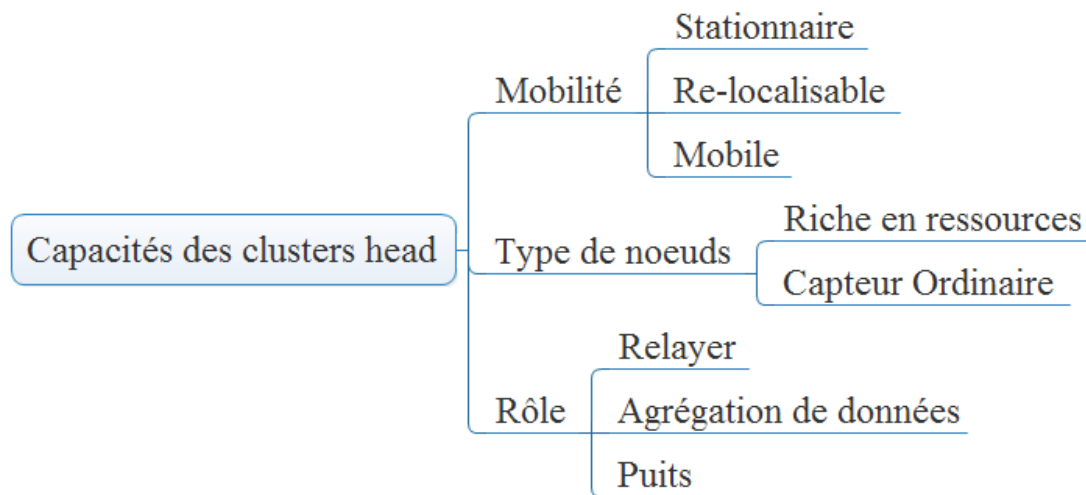


FIGURE 2.2 – Capacités des cluster heads [2].

## 2.3.3 Processus de clustering

La coordination du processus de clustering et les caractéristiques des algorithmes varient considérablement entre les régimes du clustering publiés. La **figure 2.3** illustre les facteurs influant le processus de clustering dans les réseaux de capteurs sans fil [2] :

### 2.3.3.1 Méthodologie

Lorsque Les CHs sont juste des nœuds capteurs réguliers, le clustering doit être effectué d'une manière distribuée. Dans quelques approches, une autorité centralisée partitionne les nœuds et contrôle l'appartenance au cluster. Les régimes hybrides peuvent être également trouvés, surtout quand les CHs sont riches en ressources.

### 2.3.3.2 Objectifs du clustering des nœuds

Comme indiqué dans la précédente section, plusieurs objectifs ont été poursuivis pour former des clusters. Les exemples incluent la tolérance aux pannes, l'équilibrage de charge, la connectivité du réseau, etc.

### 2.3.3.3 Sélection du Cluster Head

Les cluster heads peuvent être pré-attribués ou choisis au hasard parmi l'ensemble des nœuds déployés.

### 2.3.3.4 Complexité algorithmique

Selon l'objectif et la méthodologie, de nombreux algorithmes de clustering ont été proposés. La complexité et le taux de convergence de ces algorithmes peuvent être constants ou dépendant du nombre de CHs et/ou des nœuds capteurs. Il est également intéressant de noter que le clustering du réseau peut influencer ou être influencé par les protocoles réseau planifiés dans la couche liaison de données.

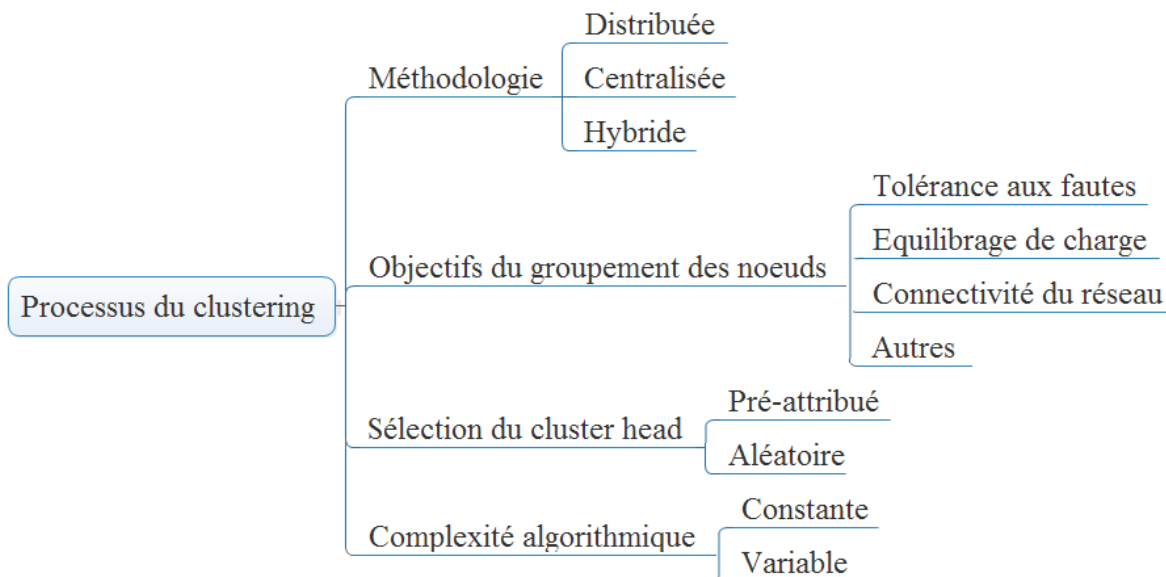


FIGURE 2.3 – Processus du clustering [2].

## 2.4 Protocole de routage hiérarchique LEACH

La hiérarchisation a permis d'étendre de façon considérable la durée de vie du réseau. L'un des premiers protocoles de cette catégorie est LEACH [23]. Cependant, il présente plusieurs points faibles auxquels des améliorations viennent pour les corriger. Dans ce qui suit, l'algorithme détaillé de LEACH sera décrit et certains de ses problèmes seront énumérés ainsi que les améliorations conçues pour les pallier.

## 2.4.1 Présentation du protocole LEACH

Le protocole LEACH est considéré comme étant l'un des premiers protocoles de routage hiérarchiques pour les RCSF, il a été conçu par Wendi B. Heinzelman et al. du MIT (Massachusetts Institute of Technology). C'est un protocole basé sur le clustering adaptatif qui utilise la randomisation pour distribuer équitablement la charge d'énergie entre les nœuds capteurs dans le réseau. Dans LEACH les nœuds s'auto-organisent en clusters et chaque *cluster* possède un *cluster-head* (CH) responsable du relais des données à la station de base (SB). A l'instar des autres protocoles de routage, LEACH utilise un algorithme de rotation du rôle de CH pour ne pas épuiser l'énergie d'un seul nœud (comme dans MTE (Minimum Transmission Energy)). De plus, afin de supprimer la redondance des données LEACH procède à une agrégation locale des données au niveau des CHs avant leur envoi à la SB [23].

LEACH se fait en rounds et chacun d'entre eux se décompose en deux phases, la première consiste à mettre en place les clusters. La seconde est pour l'acheminement des données à la station de base.

### 2.4.1.1 Construction des clusters

La construction des clusters se fait à l'aide d'un algorithme distribué qui permet aux nœuds d'être autonomes et indépendants de toute entité centralisée. Le rôle du CH nécessite une importante consommation énergétique. Dans le but d'éviter l'utilisation excessive d'un nombre fixe de nœuds comme CHs, qui épuiseront toute leur énergie, une rotation aléatoire de la fonction du CH est utilisée [22].

La construction des clusters débute par l'élection d'un certain nombre de nœuds *Cluster-head* (CH). Au début d'un round, chaque nœud  $n$  tire aléatoirement un nombre dans l'intervalle  $[0, 1]$ . Ce nombre est comparé à une probabilité  $T(n)$  calculée par l'équation (équation 2.1). Dans le cas où le nombre est inférieur à la probabilité  $T(n)$ , le nœud  $n$  est élu Cluster-Head [23].

$$T(n) = \begin{cases} \frac{p}{1-p*(r*\text{mod}\frac{1}{p})} & \text{Si } n \in G, \\ 0 & \text{Sinon,} \end{cases} \quad (2.1)$$

où  $p$ ,  $r$  et  $G$  représentent respectivement le pourcentage désiré de CHs, le round en cours et l'ensemble des nœuds n'ayant pas été élus CH lors des derniers  $\frac{1}{p}$  rounds. Durant le round 0, chaque nœud a une probabilité  $p$  de devenir CH. Les nœuds élus au round 0, deviennent illégitimes pour les prochains  $\frac{1}{p}$  rounds. Ainsi, la probabilité que le reste des nœuds soient CH augmente à mesure que le nombre de nœuds éligibles diminue. Chaque nœud élu CH pour le round en cours diffuse un message d'annonce *ADV* pour les nœuds se trouvant à sa portée. Ce message contient l'*ID* du nœud et une entête qui spécifie le type du message. Chaque nœud non-CH choisit le cluster auquel

il sera rattaché durant le round en cours et ceci en optant pour le CH qui requiert une énergie de communication minimale. Ce choix se base sur la force du signal perçu lors de la réception de l'ADV (RSSI). Après avoir fait le choix du cluster, chaque nœud devra en informer le CH par le biais d'un message de demande d'adhérence *Join - REQ*. Tout comme le message *ADV*, le *Join - REQ* contient, en plus de l'*ID* du nœud, l'*ID* du CH auquel il est rattaché [22].

Les CHs mettent en place un calendrier *TDMA* (Time Division Medium Access) pour partitionner le temps de transmission selon le nombre de ses membres (qui sera constant) [23]. Ce système permet d'éviter les collisions et de réduire la consommation énergétique de chaque nœud au moyen de la mise en veille de la radio en dehors des temps de transmissions, en plus d'utiliser de manière efficace la bande passante. L'organigramme de la **figure 2.4** est une synthèse de la phase de mise en place des clusters [22].

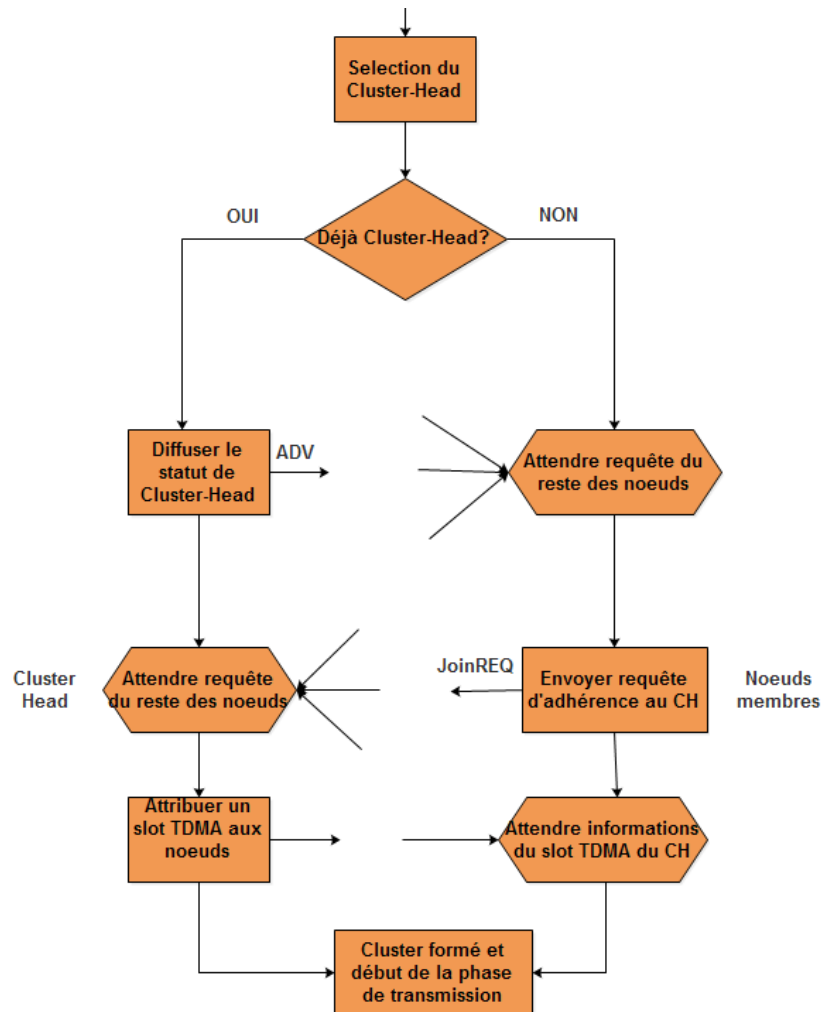


FIGURE 2.4 – Formation des clusters [22].

### 2.4.1.2 Acheminement des données

La fin de la mise en place des clusters laisse place à la seconde phase d'un round qui consiste en l'acheminement des données à la station de base. Les nœuds de chaque cluster transmettent leurs données au CH lors du temps qu'il leur est alloué. Suite à la réception des données de tous ses membres, le CH se charge de les fusionner en un unique signal qui sera transmis à la station de base [22].

La communication au sein du réseau se fait par l'intermédiaire de la radio et pour empêcher les interférences de perturber les différentes transmissions, chaque cluster dans LEACH se munit d'un code CDMA (*Code Division Multiple Access*) unique communément appelé *DSSS* (Direct Sequence Spread Spectrum) qui repose sur une approche d'étalement de spectre permettant de rendre les signaux plus résistants aux brouillages et aux interférences et le partage d'une fréquence porteuse entre plusieurs équipements [22].

### 2.4.2 Modèle radio du premier ordre

LEACH utilise un simple modèle de radio qui décrit la dissipation énergétique dans le circuit électronique de l'émetteur, du récepteur ainsi que dans l'amplificateur. Ce modèle est schématisé sur la **figure 2.5** [22]. L'équation 2.2 [23] donne la quantité d'énergie consommée lors de la transmission d'un message de  $k$  bits à une distance  $d$  du récepteur :

$$E_{Tx}(k, d) = \begin{cases} k * E_{Elec} + k * \epsilon_{fs} * d^2 & \text{Si } d < d_0, \\ k * E_{Elec} + k * \epsilon_{mp} * d^4 & d \geq d_0, \end{cases} \quad (2.2)$$

où  $E_{elec}$  est l'énergie utilisée pour exécuter le circuit émetteur,  $\epsilon_{fs}$  est l'énergie utilisée par l'amplificateur de transmission pour de courtes distances,  $\epsilon_{mp}$  est l'énergie utilisée par l'amplificateur de transmission pour de longues distances et  $d_0$  est le seuil de distance qui peut être obtenu par l'équation 2.3.

$$d_0 = \sqrt{\frac{E_{fs}}{E_{mp}}}. \quad (2.3)$$

L'équation 2.4 donne la quantité de l'énergie consommée lors de la réception d'un message de taille  $k$  bits où  $E_{elec}$  est l'énergie utilisée pour exécuter le circuit récepteur.

$$E_{Rx} = k * E_{Elect}. \quad (2.4)$$

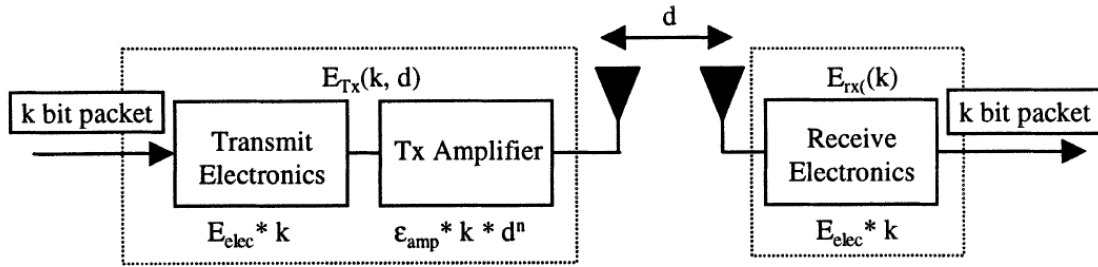


FIGURE 2.5 – Modèle radio du premier ordre [22].

### 2.4.3 Avantages et limites de LEACH

Le protocole LEACH a été l'un des premiers à introduire la notion du *clustering* dans les RCSF. Cette approche a apporté pléthore d'avantages tels que l'autonomie de la configuration des clusters puisqu'aucune entité centralisée n'intervient dans ce processus, la réduction de collisions lors des communications grâce au TDMA et l'agrégation des données qui permet de limiter le trafic au sein du réseau. Enfin, tous ces mécanismes ont conduit à répartir la consommation énergétique pour l'ensemble des nœuds du réseau. Cependant, les différentes études critiques ont soulevés un bon nombre de limites qui empêchent le protocole LEACH d'être performant à long terme telles que le problème de robustesse (la défaillance des CHs), l'épuisement plus rapide de l'énergie des CHs éloignés de la SB, l'utilisation du code DSSS qui peut nécessiter des coups supplémentaire en communication, le clustering dynamique qui peut mener à une surcharge du réseau ainsi que l'élection probabiliste des CHs, qui ne se base pas sur l'énergie résiduelle et donc peut donner lieu à des CHs qui ne durent pas dans le temps. En outre, les communications ne se font qu'à un saut augmentant ainsi la consommation énergétique.

## 2.5 Améliorations du protocole LEACH

LEACH [23] présente plusieurs défaillances qui empêchent les réseaux qui l'utilisent de travailler de façon efficace à mesure qu'ils évoluent dans le temps, il devient alors indispensable d'y remédier. A cet effet, plusieurs protocoles ont été proposés sur la base de LEACH et ont montré leurs efficacités en termes d'extension de la durée de vie du réseau par rapport à la version originale de LEACH.

### 2.5.1 LEACH-C : LEACH-Centralized

La formation distribuée des clusters présentée dans [23] présente beaucoup d'avantages (autonomie des nœuds capteurs), cependant elle ne garantit en aucun cas le bon placement des CHs au sein du réseau ni leurs nombres, il est donc difficile d'obtenir de bonnes performances. Une approche centralisée qui distribuerait d'une façon optimale les CHs, aiderait à mieux former les clusters et *ipso facto* étendre la durée de vie du réseau. Le protocole LEACH-C [22] à été proposé a cet effet, la SB intervient pour la mise en place des clusters et le choix des CHs pour chaque

round. Le protocole LEACH-C comporte un algorithme de clustering centralisé et la même phase de transmission de données que dans [23]. Initialement, la SB reçoit les données relatives à la position et au niveau d'énergie de tous les nœuds qui seront utilisées pour définir le nombre de CHs et configurer le réseau en clusters [21].

Trouver un nombre optimal de clusters est classé comme un problème *NP-Difficile* [3] et donc l'utilisation d'un algorithme approximatif est requise. L'algorithme du *Recuit simulé* [43] se base sur les principes de la *thermodynamique* pour introduire les changements des états du système. En effet, si le système entre dans un état moins énergétique que le précédent il ne changera pas d'état par contre s'il est énergétiquement plus élevé le changement est conditionné par une probabilité. Concernant le choix des CHs, la SB établit une pré-sélection en calculant la moyenne des énergies des nœuds et ceux dont l'énergie est inférieure à cette moyenne ne sont plus aptes à être CH pour le round en cours. Ainsi la SB appliquera l'algorithme du *Recuit simulé* afin de déterminer les  $k$  CHs parmi les nœuds restants, en minimisant la distance entre les CHs et les autres nœuds. A chaque itération un certain nombre de nœuds sont choisis pour être CHs pour le prochain état. Ceux dont les coordonnées sont proches d'un seuil sont élus [21].

La formation de clusters se fait à l'identique comme décrite dans LEACH [23], les CHs diffusent leurs statuts à l'aide d'un message *ADV* et les nœuds choisissent leurs CH selon la force du signal reçu (*RSSI*) puis envoient un message *Join-REQ* au CH pour l'informer de leurs adhérence. Après avoir reçus toutes les requêtes d'adhérence, chaque CH crée et diffuse un calendrier TDMA à ses membres [22].

L'avantage de LEACH-C est la réduction de la consommation énergétique des membres grâce à la minimisation de la distance carrée entre les CHs et leurs membres. De plus, l'approche centralisée utilisée permet d'avoir des clusters mieux formés et des CHs avec suffisamment d'énergie pour assurer leurs rôles [22]. Toutefois, le nombre de membres de chaque cluster n'est pas défini. En outre, l'élection des CHs qui se base uniquement sur l'énergie des nœuds capteurs et ne considère pas la distance de transmission avec la SB [30].

### 2.5.2 FZ-LEACH : Far-Zone LEACH

La présence de clusters de tailles différentes (grands et petits) au sein d'un même réseau dans LEACH [23] est un obstacle pour assurer la répartition de charge. En outre, les nœuds se trouvant loin des CHs épuisent plus rapidement leur énergie puisqu'ils transmettent sur de longues distances. Pour pallier à ce problème le protocole FAR-ZONE LEACH a été proposé. Dans ce protocole, la sélection du CH et la formation des clusters se font de la même manière que dans LEACH. De plus, la formation d'une *Far Zone* où zone lointaine dans les gros clusters formés par le protocole LEACH permet aux nœuds loin du CH de transmettre au chef de la zone [28].

Après la formation des clusters, les membres du cluster envoient leurs niveaux d'énergie ( $MinPwr_i$ )



au CH. Le CH calcule alors la valeur moyenne des  $MinPwr_i$  (noté  $AMRP$  (Average Minimum Reachability Power)). Le  $AMRP$  est calculé comme montré dans l'équation 2.5.

$$AMRP = \sum_{i=1}^N \frac{MinPwr_i}{N}, \quad (2.5)$$

où  $N$  est le nombre de membres du cluster. Le  $AMRP$  est une estimation du coût de la communication qui sera utilisé pour la formation de la Far-Zone. Une fois que la Far-Zone est formée, un membre de la zone est sélectionné aléatoirement comme  $ZH$ . Le  $ZH$  crée un calendrier TDMA et assigne des slots aux membres de la zone pendant lesquels ils peuvent transmettre les données au  $ZH$ . Ce dernier, agrège les données reçues et les transmet au CH (**figure 2.6**) [28].

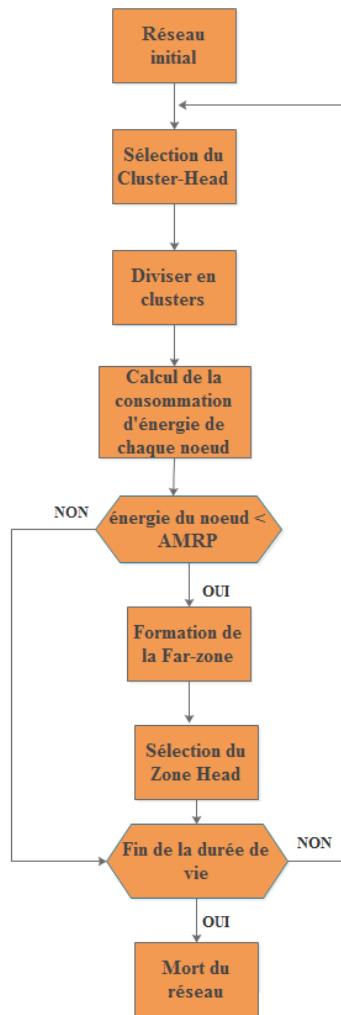


FIGURE 2.6 – Formation des FAR-ZONE [28].

### 2.5.3 MG-LEACH : Multi Group Based LEACH

Dans LEACH [23], les nœuds couvrant une même zone géographique peuvent recenser les mêmes événements mais cette redondance est éliminée par les CHs qui agrègent ces événements avant de les envoyer à la SB. La majeure partie des réserves énergétiques des nœuds s'épuisent à cause de la communication, ainsi le recensement multiples de mêmes événements est considéré comme un gouffre énergétique. L'idée de base du protocole MG-LEACH [18] est d'exploiter la redondance des nœuds dans une zone dans le but de réduire la consommation énergétique et donc prolonger la durée de vie du réseau [18].

MG-LEACH se fait en 3 phases : une première phase de pré-construction où la SB s'occupe de diviser les nœuds en plusieurs sous groupes selon leur densité et leurs positions et de mettre en place un cycle de travail. et deux autres phases de configuration des clusters et d'acheminement de données qui se font de la même manière que dans LEACH. Les nœuds de chaque sous groupe se mettent en veille hors de leur cycle de travail ce qui permet d'étendre de façon considérable la durée de vie du réseau [18].

L'avantage de MG-LEACH est qu'il permet de gagner en consommation énergétique mais aussi en trafic. Cependant, les CHs sont élus à la base d'une probabilité [18].

### 2.5.4 EC-LEACH : Equalized Cluster LEACH

Les clusters formés par LEACH [23] ne sont pas de tailles équivalentes, il peut exister des clusters plus grands ou plus petits que d'autres. Les nœuds assurant le rôle de CH dans les grands clusters épuisent plus rapidement leur énergie. Equalized Cluster LEACH [42] vise à former des clusters de tailles approximativement égales et ceci à l'aide de deux seuils qui régulariseront le nombre de nœuds dans chaque cluster. Le seuil minimal est représenté par *Thresh1* et le maximal par *Thresh2*. Lorsque le nombre de nœuds dans un cluster dépasse *Thresh2*, le CH retient un nombre plus proche de *Thresh1* et se défait du reste des nœuds (en envoyant une valeur négative de slot aux membres concernés). Dans le cas où le nombre est en dessous de *Thresh1* le CH renvoi un message d'annonce pour avoir plus de nœuds, particulièrement ceux qui ont été rejetés par les grands clusters. Dans le cas où la taille d'un cluster dépasse la somme des seuils *Thresh1* et *Thresh2*, dans ce cas le CH retient la valeur la plus proche de *Thresh2* et en informe les nœuds restants afin qu'ils forment un nouveau cluster dont le CH sera choisi par le CH original [42].

EC-LEACH permet d'avoir des clusters de tailles approximativement égales et donc d'assurer une bonne répartition de la consommation énergétique. Par contre, l'élection du CH se base sur une probabilité.

### 2.5.5 U-LEACH : Universal-LEACH

Cette approche combine les avantages des protocoles I-LEACH [34], PEGASIS [36] et DEEC [49] en plus de s'appuyer sur le modèle de radio du protocole LEACH [23]. Dans PEGASIS, chaque nœud  $i$  communique uniquement avec son voisin  $j$  en ajustant sa portée. Ce dernier est choisi à chaque round, en fonction de son énergie résiduelle, afin de relayer les données du nœud  $i$  à la Station de Base. A l'instar de LEACH, il n'y a qu'un seul nœud relais à la SB ce qui réduit la surcharge du réseau et l'utilisation de la bande passante pour la communication [33]. DEEC est une variante de LEACH aussi bien adaptée aux réseaux homogènes qu'hétérogènes [33].

Dans I-LEACH, les CHs sont sélectionnés en corrélation avec leurs énergies résiduelles et non à la base d'une probabilité. Dans U-LEACH, les clusters sont formés en premier lieu, en fonction des coordonnées horizontales des membres et les CHs sont élus sur la base de leurs énergies résiduelles. Parmi l'ensemble des CHs un *Master CH* est choisi. Le *Master CH* se charge de relayer à la station de base les données reçues des CHs. En second lieu, la formation de la chaîne est entamée depuis le membre le plus éloigné du cluster jusqu'au CH. Ce dernier se charge de fusionner les données de ses membres. La distribution de l'énergie se fait de manière symétrique de sorte que la moitié des nœuds ont une énergie plus élevée que l'autre moitié [33].

L'avantage de U-LEACH est l'élection des CHs qui se base sur leurs énergies et le regroupement des nœuds en utilisant les coordonnées minimisant ainsi les distances de transmissions. Cependant, le rôle de *Master CH* étant attribué à un seul membre est désavantageux dans le cas d'une densité importante des nœuds.

### 2.5.6 Q-LEACH : Quadrature-LEACH

L'un des nombreux challenges de la conception des protocoles de routage pour les RCSF est d'obtenir une bonne couverture des nœuds capteurs pour garantir un certain balancement énergétique. Dans LEACH [23], la formation dynamique des clusters épuisent plus rapidement l'énergie des membres se situant loin du cluster-head.

L'idée principale du protocole Quadrature LEACH (Q-LEACH) est de partitionner le réseau en quatre cadrans virtuels et d'y appliquer le processus de clustering [41]. Le protocole Q-LEACH s'effectue en deux phases : (1) phase de *mise en place des clusters* et (2) phase d'*acheminement des données*. Ces deux phases se font de la même manière que LEACH [23]. La mise en place des clusters se fait en deux étapes : la sélection des CHs dans les quatre zones (algorithme 1) et l'association des membres aux CHs (algorithme 2). L'organigramme montré par le **figure 2.7** explique en détails le principe de Q-LEACH [41].

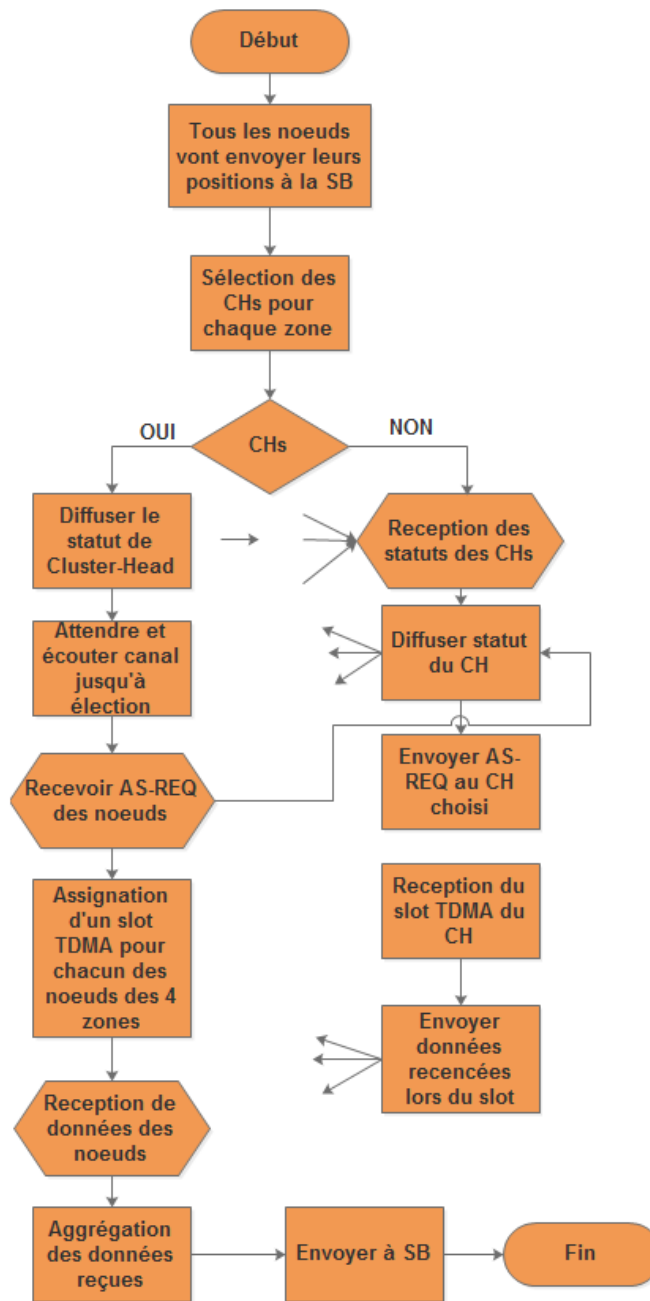


FIGURE 2.7 – Principe du protocole Q-LEACH [41].

L'algorithme 1 explique le processus de sélection des CHs pour chacune des zones (A, B, C, D), chaque nœud  $n$  voulant être CH pour un round tire un nombre aléatoire dans l'intervalle  $[0, 1]$ . Ce nombre est ensuite comparé à la probabilité  $T(n)$ . Dans le cas où le nombre est inférieur à  $T(n)$ , le nœud  $n$  devient CH. Le processus se répète dans les quatre zones jusqu'à obtention du nombre optimum de clusters [41].

**Algorithme 1** Sélection des cluster heads dans le protocole Q-LEACH [41].

---

```

1: Si (nœud  $\varepsilon G \rightarrow G =$  nœuds n'ayant pas été CH pendant le round) alors
2:   Si (ZoneNœud == 'ZoneA') alors
3:     Si ( $NOMBRECH_s \geq \frac{N}{K}$ ) alors
4:       VAR = Nombre Aléatoire (0-1)
5:       Si  $VAR \geq \frac{P}{1-P(r,mod1/P)}$  alors
6:         nœud = CH_A
7:         NOMBRE DE CHs = NOMBRE DE CHs+1
8:       Finsi
9:     Sinon
10:      Si (ZoneNoeud == 'ZoneB') alors
11:        RÉPÉTER ÉTAPES 3 :7
12:      Sinon
13:        Si (ZoneNoeud == 'ZoneC') alors
14:          RÉPÉTER ÉTAPES 3 :7
15:        Sinon
16:          Si (ZoneNoeud == 'ZoneD') alors
17:            RÉPÉTER ÉTAPES 3 :7
18:          Finsi
19:        Finsi
20:      Finsi
21:    Finsi
22:  Finsi
23: Finsi

```

---

L'algorithme 2 explique l'association des nœuds capteurs aux CHs de leurs zones. Tout d'abord, chaque nœud  $N$  se localise dans une zone  $(a_1, a_2, a_3, a_4)$  et trouve les CHs de cette zone. Il choisit et choisissent un selon le *RSSI*. Q-LEACH permet de distribuer les nœuds capteurs et de positionner les CHs d'une façon optimale afin de stabiliser le réseau et d'augmenter le débit. Néanmoins, l'élection des CHs est probabiliste et ne prend pas en compte l'énergie ou la distance des nœuds [41].

**Algorithme 2** Association des membres aux cluster heads dans le protocole Q-LEACH [41].

---

```

1:  $N \in$  Groupe de nœuds normaux
2:  $GC \in$  Groupe de CHs
3:  $A=(a1, a2, a3, a4)$ 
4: Si ( $N \in (A, a1)$ ) alors
5:   Recherche de tous les CH_A
6:   Vérifier RSSI de chaque CH_A
7:   Adhérer à CH_A
8:   Transférer données à CH_A
9: Finsi
10: Si ( $N \in (A, a2)$ ) alors
11:   Répéter étapes de 5 : 8 Pour CH_B
12: Finsi
13: Si ( $N \in (A, a3)$ ) alors
14:   Répéter étapes de 5 : 8 Pour CH_C
15: Finsi
16: Si ( $N \in (A, a4)$ ) alors
17:   Répéter étapes de 5 : 8 Pour CH_D
18: Finsi

```

---

### 2.5.7 ML-LEACH : Multi-Level LEACH

L'une des nombreuses limites de LEACH [23] est l'augmentation du nombre de nœuds déployés qui réduit considérablement la durée de vie du réseau. Les membres des clusters devront transmettre sur de plus longues distances et épuiseront plus rapidement leurs réserves d'énergies. En outre, la sélection aléatoire des Clusters-Head ne prend pas en considération le niveau d'énergie des nœuds [32]. Les protocoles Two-Level LEACH et 3 Level-LEACH restreignent le nombre de Cluster-Head (CH) qui communiquent avec la Station de Base en échelonnant les CHs en deux niveaux ou plus afin de minimiser les distances à parcourir pour les CHs se situant loin de la SB [31].

Le Two-Level LEACH (TL-LEACH) [32] introduit une hiérarchisation supplémentaire pour atteindre la SB, en subdivisant les CHs en deux niveaux. En effet, les CHs du niveau inférieur transmettent leurs données aux CHs du niveau supérieur. TL-LEACH se déroule en trois phases [32] :

1. **Élection des CHs** : pendant cette première phase, on peut distinguer trois sortes de nœuds [32] :
  - SN : *nœud simple* qui se trouve en bas de la hiérarchie dont la fonction est de collecter les données et les transmettre à son CH secondaire.

- $CH_i$  : CH *secondaire* qui se trouve au premier niveau de la hiérarchie et agrège les données recensées par les nœuds simples.
- $CH_{ij}$  : CH *primaire* qui se trouve dans le niveau supérieur de la hiérarchie dont la réserve énergétique est considérable et se trouve non loin de la SB qui reçoit et agrège les données provenant de ses CHs secondaires et les fait suivre à la SB.

L'élection des CHs *secondaires* est tout d'abord effectuée, elle se fait comme décrite dans LEACH [23]. Ensuite, les nœuds élus comme CHs *secondaires* diffusent leurs niveaux d'énergie. Les CHs *secondaires* ayant une importante réserve d'énergie seront choisis comme CHs *primaires* [32].

2. **Phase de configuration des clusters** : Chaque nœud simple choisit le CH *secondaire* auquel il va envoyer ses données en se basant sur la force du signal reçu. Ensuite, il envoie un message contenant l'identifiant du CH *secondaire* et son identifiant pour l'informer de son choix. Les CHs *secondaires* choisissent à leur tour le CH *primaire* auquel ils vont transmettre leurs données en envoyant un message contenant leurs identifiants mais aussi le nombre de membres qu'ils possèdent. Ainsi, les CHs *primaires* ont une connaissance globale de leurs hiérarchie (voir **figure 2.8**) [32].

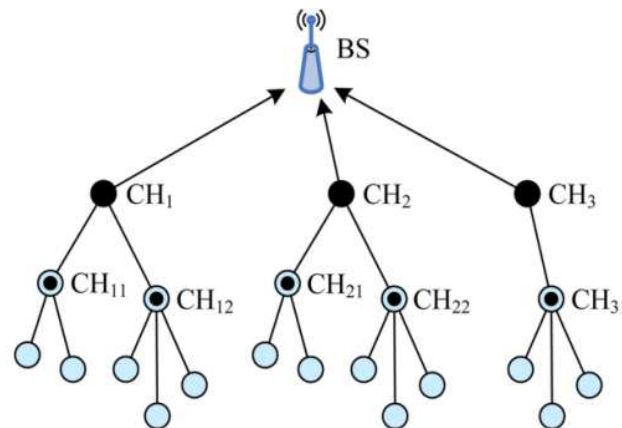


FIGURE 2.8 – Hiérarchisation dans TL-LEACH [37].

3. **Phase d'acheminement de données** : Dans un premier temps les nœuds simples se chargent de collecter les données directement sur le *terrain*. Ensuite, ils les transmettent à leurs CHs *secondaires*. Ces derniers agrègent les données de leurs membres et les transmettent aux CHs *primaires*. La fonction principale des CHs *primaires* consiste à agréger et à envoyer les données reçues à la Station de Base [32].

Le 3 Level-LEACH est une extension du TL-LEACH qui effectue la hiérarchisation de la même manière, à la différence que les CHs seront divisés en trois niveaux comme suit [31] :

1. **Élection des Cluster Heads** : les CHs des deux premiers niveaux (considérés comme *secondaires*) sont élus de la même façon que dans TL-LEAH. Le dernier niveau de la hiérarchie est formé par les CHs *secondaires* se trouvant à proximité de la Station de base et disposant d'un niveau d'énergie élevé sont choisis comme CHs *primaires* [31].
2. **Phase de configuration des clusters** : Les clusters se forment comme dans TL-LEACH. Cependant, les CHs du second niveau envoient leurs identifiants ainsi que le nombre de leurs membres aux CHs *primaires* créent et diffusent un calendrier TDMA [31].
3. **Phase d'acheminement de données** : L'acheminement de données se fait comme dans TL-LEACH. En plus, les CHs du second niveau agrègent et envoient les données à leurs CHs *primaires*. Ces données sont transmises à la SB après agrégation [31].

## 2.6 Conclusion

Le présent chapitre a permis de dépeindre le protocole LEACH et de présenter certaines de ses améliorations, principalement proposées pour les grands clusters, tout en énumérant les différentes contributions du processus de *clustering*.



# Chapitre 3

## RCLME : Réorganisation des Clusters à base de Localisation pour un routage Multi-sauts avec efficacité Energétique

### 3.1 Introduction

Le chapitre précédent était axé sur la description du protocole LEACH [23] et certaines de ses améliorations telle que FZ-LEACH. Ces dernières visent à résoudre certaines limites du protocole LEACH [23], telles que la surcharge des clusters, communication directe entre les CHs et la station de base (SB), la non prise en compte de l'énergie des nœuds dans le processus d'élection des CHs, etc.

Pour pallier ces limites, un nouveau protocole basé sur la localisation et la réorganisation des clusters (RCLME - **R**éorganisation des **C**lusters à base de **L**ocalisation pour un routage **M**ulti-sauts avec efficacité **E**nergétique) est proposé pour le routage multi-sauts dans les RCSF. Le protocole combine l'approche distribuée du protocole LEACH [23] avec le principe de partitionnement des clusters utilisé dans FZ-LEACH [28].

### 3.2 Problématique et motivations

L'un des objectifs majeurs des protocoles de routage dans les RCSF consiste à minimiser les dépenses énergétiques lors des échanges au sein du réseau ; en vue de maximiser sa durée de vie. Le protocole LEACH est l'un des protocoles de routage les plus populaires dans les RCSF. Néanmoins, plusieurs limites ont été recensées donnant lieu à maintes améliorations. Parmi ces limites, on peut citer :

- Les membres d'un cluster (MC) se trouvant à une distance lointaine du Cluster-Head (CH) voient leurs énergies s'épuiser plus rapidement que les autres membres. Cela engendre une

perte de la couverture d'une partie de la zone d'intérêt ;

- A chaque round, l'élection probabiliste des CHs se base sur la génération d'un nombre aléatoire mais ne tient pas compte de l'énergie résiduelle ni du nombre de voisins ;
- La communication directe entre les CHs et la station de base conduit, à l'épuisement rapide des ressources énergétiques des CHs se situant loin de la SB.

L'objectif de ce travail consiste donc à résoudre ces limites en proposant un algorithme de clustering utilisant : 1) *la réorganisation des clusters* (migration des CHs et formation des sous zones) et 2) *le routage multi-sauts* intra et inter-cluster.

### 3.3 Propriétés du réseau et hypothèses

On considère un réseau de capteurs sans fil composé de  $N$  nœuds capteurs statiques, déployés aléatoirement et uniformément afin de collecter continuellement les événements pouvant se produire sur la zone de déploiement. Les capteurs formant le réseau ont les mêmes capacités de couverture et de communication ainsi que la même quantité d'énergie initiale. Un nœud  $i$  est caractérisé par des coordonnées géographiques  $(x_i, y_i)$  et une zone de communication (zone de transmission ou portée) généralement modélisée par un cercle de centre  $(x_i, y_i)$  et de rayon  $RC_i$ . La solution proposée tient compte des hypothèses suivantes :

- La station de base se trouve en dehors de la zone de déploiement des nœuds capteurs ;
- Trois nœuds ancrés sont déployés aléatoirement. Ils possèdent assez d'énergie et sont équipés d'un GPS (Global Positioning System) leur permettant de calculer leurs coordonnées. Ces nœuds peuvent atteindre tous les nœuds du réseau ;
- Chaque nœud peut ajuster sa portée de transmission à une distance donnée afin d'atteindre une destination ;
- La défaillance d'un nœud ne peut être causée que par l'épuisement de son énergie ;
- Deux nœuds sont dits voisins s'ils se trouvent dans le rayon de communication l'un de l'autre.

### 3.4 RCLME : La solution proposée

L'algorithme proposé se déroule en trois phases : **1)** une phase de *configuration des clusters* similaire à celle de l'algorithme LEACH [23], **2)** une phase de *réorganisation des clusters* permettant la division des clusters en quatre zones et **3)** une phase d'*acheminement des données* vers les Cluster Heads, puis vers la station de base.

### 3.4.1 Phase de configuration des clusters

La première phase de l'algorithme RCLME s'effectue en trois étapes. La première consiste en la détermination des positions de chaque nœud du réseau, la seconde étant l'élection des CHs et enfin l'étape d'association des membres aux CHs.

#### 3.4.1.1 Localisation des nœuds capteurs

L'étape de localisation ne se fera qu'une seule fois après le déploiement des nœuds et avant le début du premier round. Les trois ancres diffusent leurs coordonnées à l'ensemble du réseau, les nœuds capteurs calculeront ensuite leurs coordonnées en utilisant la méthode de trilatération. De manière plus précise, soient trois ancres  $\mathbf{A1}(X_1, Y_1)$ ,  $\mathbf{A2}(X_2, Y_2)$ ,  $\mathbf{A3}(X_3, Y_3)$  et un nœud capteur  $\mathbf{B}(X, Y)$  qu'on souhaite localiser. Après avoir reçu les coordonnées des trois ancres, le nœud  $\mathbf{B}$  estime à l'aide du *RSSI* (Received Signal Strength Indication) les distances  $d_1$ ,  $d_2$  et  $d_3$  le séparant respectivement des ancres  $A_1$ ,  $A_2$  et  $A_3$  (**figure 3.1**) [27].

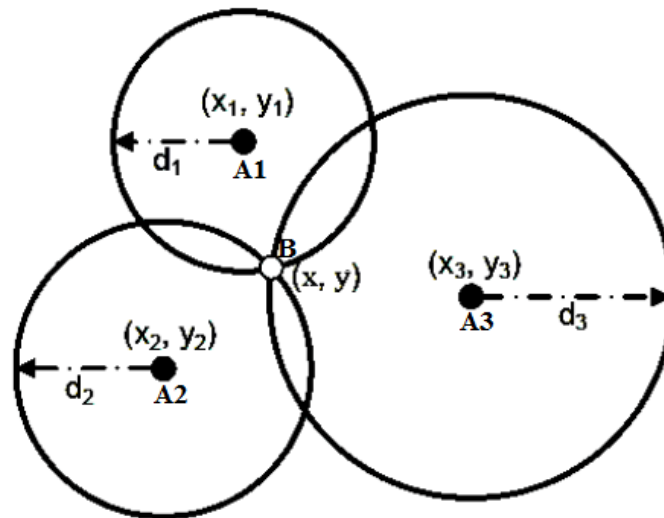


FIGURE 3.1 – Principe de la méthode de trilatération.

A partir des distances obtenues et des coordonnées reçues, les cercles ont pour équations :

$$\begin{cases} (X - X_1)^2 + (Y - Y_1)^2 = d_1^2 \\ (X - X_2)^2 + (Y - Y_2)^2 = d_2^2 \\ (X - X_3)^2 + (Y - Y_3)^2 = d_3^2 \end{cases} \quad (3.1)$$

Trouver les coordonnées du nœud  $\mathbf{B}$  revient à résoudre ce système d'équations à deux inconnus  $X$  et  $Y$ .

#### 3.4.1.2 Élection des Cluster-Heads temporaires

Les Cluster Heads sont élus à l'aide d'un algorithme distribué permettant aux nœuds d'être autonomes et indépendants de toute entité centrale. Au début d'un round, chaque nœud  $i$  tire

aléatoirement un nombre  $a$  dans l'intervalle  $[0, 1]$ . Ce nombre est comparé à une probabilité  $T(i)$  calculée par l'équation 3.2 [23].

$$T(i) = \begin{cases} \frac{p}{1-p*(r*\text{mod}(\frac{1}{p}))} & \text{Si } i \in G, \\ 0 & \text{Sinon,} \end{cases} \quad (3.2)$$

où  $p$ ,  $r$  et  $G$  représentent respectivement le pourcentage désiré de CHs, le round en cours et l'ensemble des nœuds n'ayant pas été élus CH lors des derniers  $\frac{1}{p}$  rounds. Dans le cas où  $a$  est inférieur à la probabilité  $T(i)$ , le nœud  $i$  est élu Cluster-Head. Autrement, il se met en attente des messages d'annonce (*ADV*) en provenance des CHs se trouvant dans sa portée de transmission. Le CH élu à cette étape est dit temporaire ( $CH_{Temporaire}$ ). La **figure 3.2** illustre l'organigramme de cette étape.

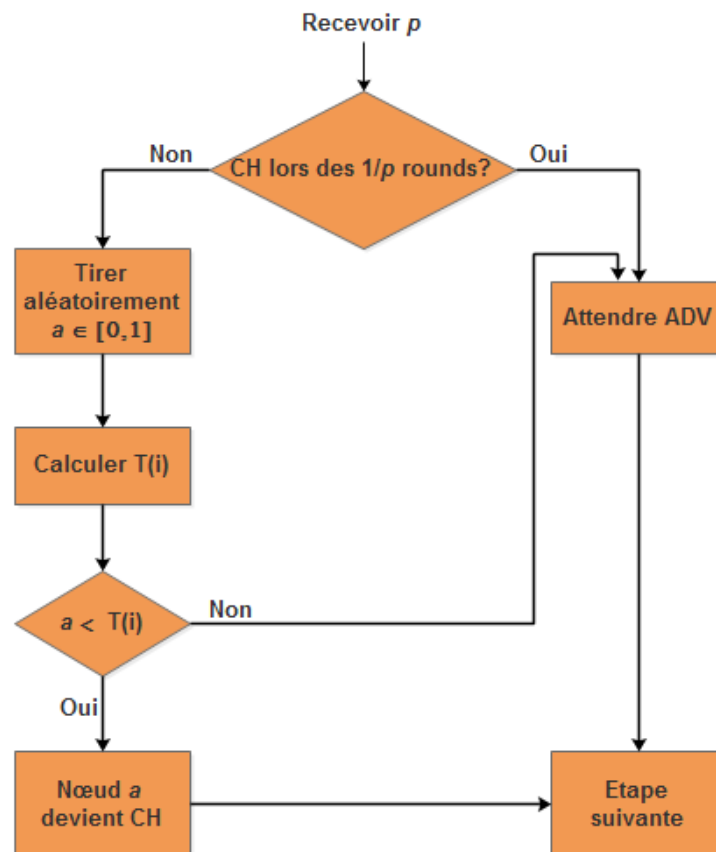


FIGURE 3.2 – Étape d'élection des Cluster Heads de l'algorithme *RCLME*.

### 3.4.1.3 Association des membres aux Cluster-Heads

Pour informer ses nœuds voisins de sa décision, chaque  $CH_{Temporaire}$  diffuse un message d'annonce  $ADV$ . Le contenu de ce message est montré dans la **figure 3.3**.

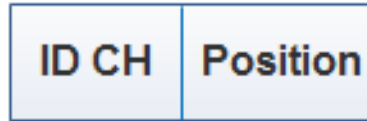


FIGURE 3.3 – Message  $ADV$  diffusé lors de l'association des membres.

Où **ID CH** représente l'identifiant du Cluster-Head ayant diffusé le message  $ADV$  et **Position** indique ses coordonnées.

Après avoir reçu le message  $ADV$ , chaque nœud non-CH  $j$  choisit le cluster auquel il sera rattaché durant le round en cours en optant pour le  $CH_{Temporaire}$  le plus proche. La distance étant déduite à partir des coordonnées géographiques. Pour rejoindre le  $CH_{Temporaire}$  choisi, le nœud  $j$  envoie un message de demande d'adhérence  $Join-REQ$  (voir le contenu dans la **figure 3.4**) et devient alors *nœud membre* du cluster.

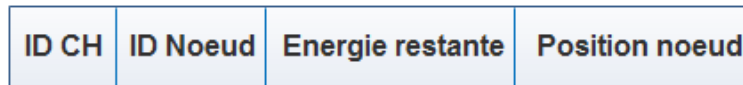


FIGURE 3.4 – Message  $Join-REQ$  envoyé par un nœud membre.

Où **ID CH** est l'identifiant du CH auquel le nœud  $j$  décide d'être rattaché. **ID nœud**, **Énergie restante** et **Position nœud** représentent respectivement l'identifiant, l'énergie résiduelle et les coordonnées du nœud membre  $j$ . Chaque  $CH_{Temporaire}$  met en place un calendrier  $TDMA$  pour partitionner le temps de transmission selon le nombre de membres du cluster et le répartir sur l'ensemble des nœuds du cluster.

L'algorithme 3 décrit les étapes d'élection d'un cluster head temporaire et d'association des membres. Il est à noter que  $Etat_i$ ,  $ERes_i$ ,  $position_i$  et  $V_i$  représentent respectivement l'état, l'énergie résiduelle, la position et l'ensemble des voisins du nœud  $i$ . En outre,  $C$  est l'ensemble des cluster heads et  $MC_i$  représente l'ensemble des membres du cluster head  $i$ .

**Algorithme 3** Élection du cluster head temporaire et association des membres dans *RCLME*.

---

```

1:  $Etat_i = NonDécidé$ 
2: Si ( $Etat_i <> CH$  lors des  $\frac{1}{p}$  round) alors
3:   Générer aléatoirement  $a \in [0, 1]$ 
4:   Calculer  $T(i)$  selon la formule 3.2
5:   Si ( $a < T(i)$ ) alors
6:      $Etat_i = CH_{Temporaire}$ 
7:      $C = C \cup i$  /* Mise à jour de l'ensemble  $C$  */
8:     Envoyer  $ADV(i, position_i)$  à  $V_i$ 
9:     Attendre  $Join - REQ$ 
10:    Si (Réception de  $Join - REQ$  de  $j/j \in V_i$ ) alors
11:       $MC_i = MC_i \cup j$  /* Mise à jour de l'ensemble  $MC_i$  */
12:    Finsi
13:  Sinon
14:    Attendre  $ADV$ 
15:    Si (Réception des  $ADV$  de  $V_i \cap C$ ) alors
16:      Considérer l' $ADV$  du nœud  $j$  le plus proche  $/j \in V_i \cap C$ 
17:      Envoyer  $Join - REQ(j, i, ERes_i, position_i)$  au cluster head  $j$ 
18:    Finsi
19:  Finsi
20: Sinon
21:   Répéter étapes 14 :17
22: Finsi

```

---

### 3.4.2 Phase de réorganisation des clusters

La réorganisation des clusters se fait en deux étapes. La première a pour objectif d'élire un Cluster-Head définitif ( $CH_{Définitif}$ ) en procédant au partitionnement du cluster en quatre zones. La seconde consiste à former une sous zone dans chaque zone.

#### 3.4.2.1 Élection des Cluster-Heads définitifs

Dans cette étape, chaque cluster est divisé en quatre zones virtuelles en utilisant deux axes X et Y où le  $CH_{Temporaire}$  est le point de repère (0, 0) (voir **figure 3.5**). Afin de déterminer l'appartenance des nœuds membres aux différentes zones, le  $CH_{Temporaire}$  calcule les nouvelles coordonnées de ses membres par rapport à l'origine considérée.

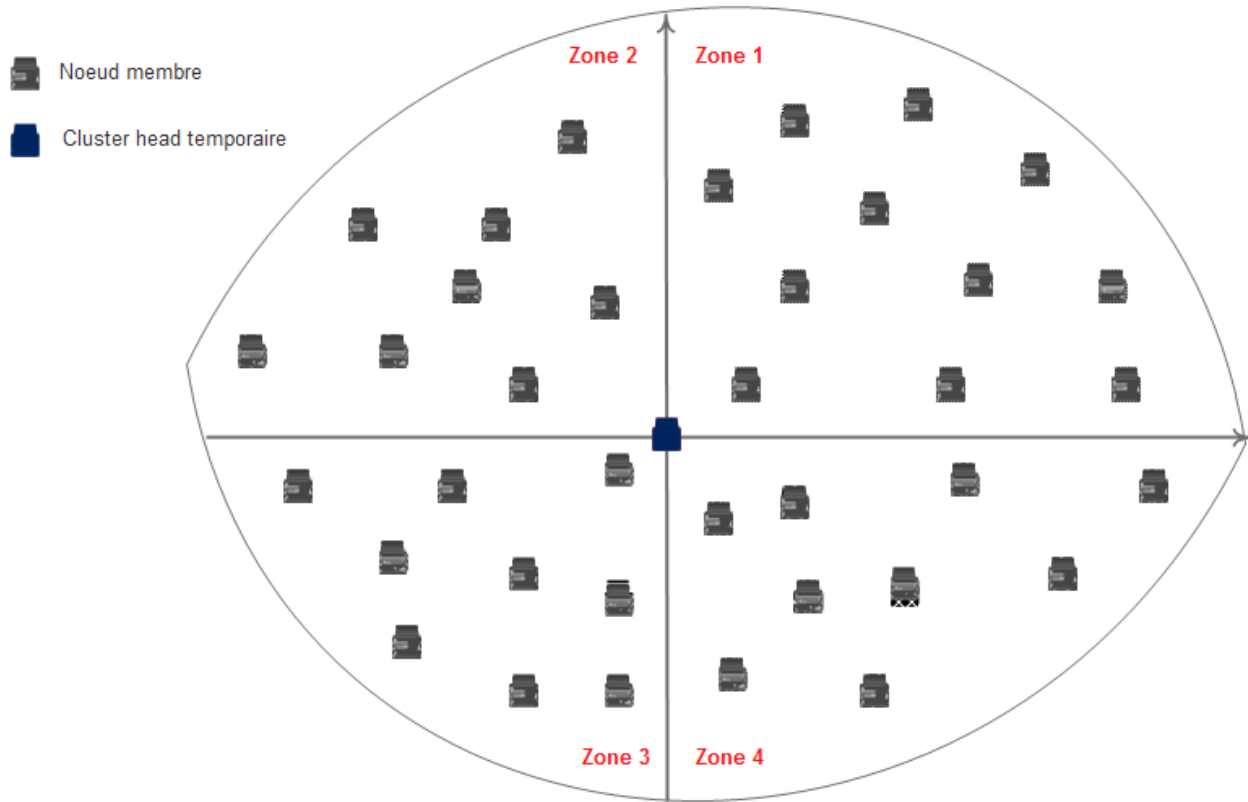


FIGURE 3.5 – Partitionnement d'un cluster en quatre zones.

Dans le but de déterminer la zone la plus dense, c.à.d. la zone qui possède le nombre maximal de nœuds membres, le  $CH_{Temporaire}$  calcule le nombre de membres dans chaque zone. En effet, la zone 1 est formée des membres ayant des abscisses et des ordonnées positives. La zone 2 se compose des membres possédant des abscisses négatives et des ordonnées positives. La zone 3 comporte des membres dont les abscisses et les ordonnées sont négatives. La zone 4 se caractérise par les membres ayant des abscisses positives et des ordonnées négatives (algorithme 4).

Après avoir déterminé la zone la plus dense, le  $CH_{Temporaire}$  élira un CH définitif ( $CH_{Définitif}$ ) parmi les membres de cette zone. Lors de cette élection, le  $CH_{Définitif}$  ne doit perdre la communication avec aucun autre membre du cluster. Dans le cas où plusieurs membres candidats existent, l'élection tiendra compte du nombre de voisins et de l'énergie résiduelle. La non perte de la communication signifie la satisfaction de la condition de migration suivante : "la distance entre le  $CH_{Définitif}$  et le reste des membres du cluster est inférieure ou égale au rayon de transmission du  $CH_{Définitif}$ ". La distance étant déduite par le  $CH_{Temporaire}$  grâce aux coordonnées géographiques des nœuds membre.

**Algorithme 4** Détermination de la zone la plus dense dans RCLME.

---

**Entrées** Un cluster composé de M membres  $m_i(x_i, y_i)$

**Sorties** ZoneDense

```

1: Zone1=0 ; Zone2=0 ; Zone3=0 ; Zone4=0 ;
2: Pour  $i$  allant de 1 à M faire
3:   Si  $((x_i \geq 0) \ \&\& \ (y_i \geq 0))$  alors
4:     Zone1++
5:   Sinon
6:     Si  $((x_i < 0) \ \&\& \ (y_i > 0))$  alors
7:       Zone2++
8:     Sinon
9:       Si  $((x_i \leq 0) \ \&\& \ (y_i \leq 0))$  alors
10:        Zone3++
11:       Sinon
12:         Si  $((x_i > 0) \ \&\& \ (y_i < 0))$  alors
13:           Zone4++
14:         Finsi
15:       Finsi
16:     Finsi
17:   Finsi
18: Finpour
19: ZoneDense = Max(Zone1, Zone2, Zone3, Zone4)

```

---

Pour chaque membre  $i$  du cluster susceptible de devenir  $CH_{Définitif}$ , le  $CH_{Temporaire}$  calcule  $EV(i)$  la combinaison entre le pourcentage de l'énergie résiduelle ( $\frac{E_{res}(i)}{E_{initiale}}$ ) et le nombre de voisins ( $|V_i|$ ) en utilisant la formule 3.3. Cette combinaison se base sur l'énergie et le nombre de voisins pour diverses raisons : (1) la consommation énergétique des cluster-heads est plus élevée que celle des membres (fonctions d'agrégation des données et de transmission à la station de base), (2) le choix de cluster-heads ayant le plus de voisins, permet de réduire les dépenses énergétiques lors des différents échanges.

$$EV(i) = |V_i| \times \frac{E_{res}(i)}{E_{initiale}}, \quad (3.3)$$

où  $E_{res}(i)$  et  $|V_i|$  représentent respectivement l'énergie résiduelle et l'approximation du nombre de voisins du membre  $i$  calculée par le  $CH_{Définitif}$ .

Le membre  $i$  ayant la valeur maximale de  $EV$  est élu comme étant  $CH_{Définitif}$ . Le  $CH_{Temporaire}$  envoie alors, aux membres du cluster, un message de notification *NOTIFY* contenant les identifiants du  $CH_{Temporaire}$  et du  $CH_{Définitif}$  ainsi que la liste des membres (**figure 3.6**). Les membres actualiseront l'identifiant du cluster head (CH) auquel ils sont désormais rattachés.



ID CH <sub>temporaire</sub>	ID CH <sub>définitif</sub>	Liste_Membre
-----------------------------	----------------------------	--------------

FIGURE 3.6 – Message *NOTIFY* envoyé aux membres du cluster à la suite de la migration.

### 3.4.2.2 Formation des sous zones

Après l'élection du  $CH_{Définitif}$ , chaque cluster est partitionné à nouveau en quatre zones virtuelles en utilisant deux axes X et Y mais cette fois-ci avec le  $CH_{Définitif}$  comme point de repère (0, 0). Pour chaque zone obtenue, les membres se trouvant au-delà de la distance  $\frac{d_{max}}{2}$  ( $d_{max}$  étant la distance séparant le  $CH_{Définitif}$  du membre le plus éloigné de la zone) forment une sous zone dont un sous CH (SCH) sera élu. Cette élection se base sur la distance avec le  $CH_{Définitif}$  et la valeur de  $EV$ . Cette étape est décrite par la **figure 3.7**.

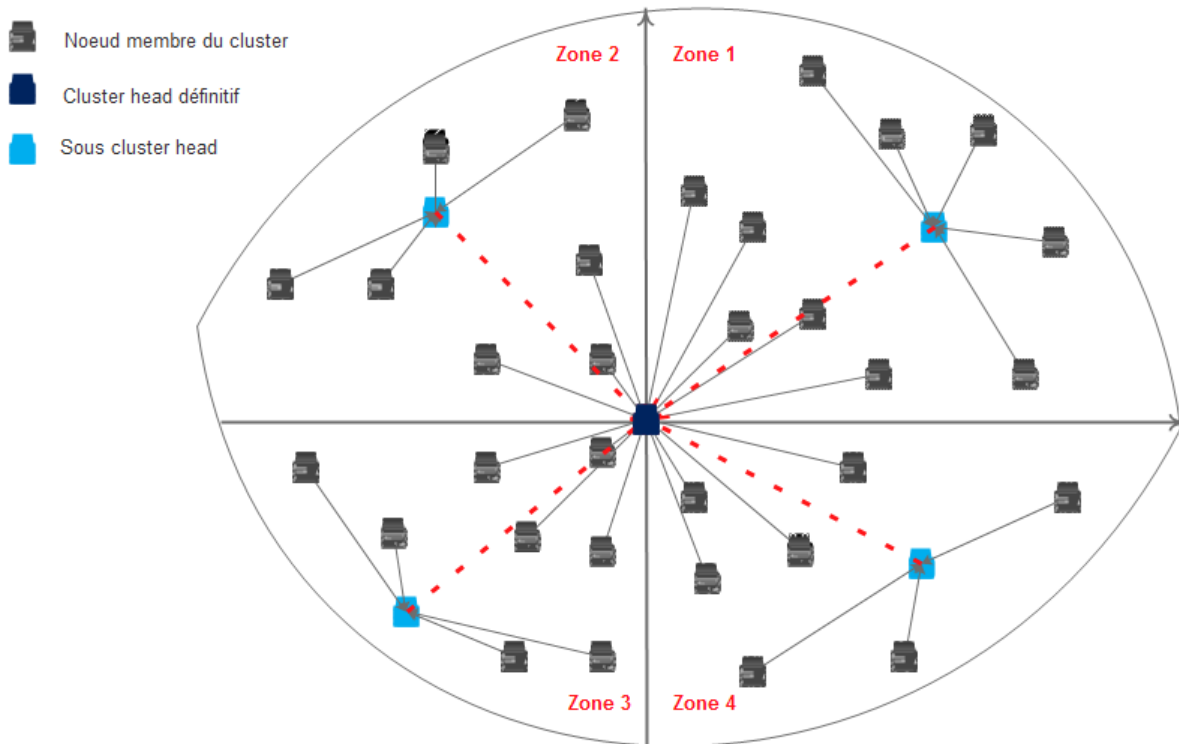


FIGURE 3.7 – Formation des sous zones dans un cluster.

La **figure 3.7** illustre la formation de quatre sous zones au sein d'un cluster. Pour chaque sous zone, le  $CH_{Définitif}$  choisit le membre le plus proche ayant la valeur maximale de  $EV$  comme étant sous CH. Ce choix est motivé par le fait que les communications n'épuisent pas de façon rapide les réserves énergétiques du nœud choisi. Après l'élection du SCH, le  $CH_{Définitif}$  diffuse un message de notification  $NOTIFY\_SCH$  à l'ensemble de ses membres (**figure 3.8**). Chaque sous CH met à jour son statut et les membres se trouvant dans une sous zone actualisent l'identifiant du sous CH auquel ils vont envoyer leurs données.

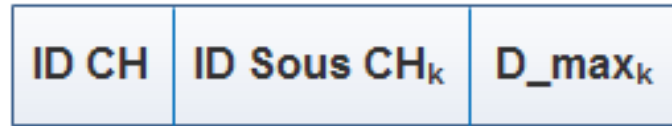


FIGURE 3.8 – Message  $NOTIFY\_SCH$  envoyé après le choix du sous cluster head.

Où  $ID\ CH$  est l'identifiant du  $CH_{Définitif}$  du cluster.  $ID\ Sous\ CH_k$  et  $D\_Max_k$  représentent respectivement l'identifiant du sous CH et la distance maximale de la zone k.

L'algorithme 5 décrit les étapes du partitionnement ainsi que la formation des sous zones dans un cluster.  $C_i$  représente le cluster dont le nœud  $i$  est cluster head,  $ZD_i$  est l'ensemble des membres de la zone la plus dense du cluster  $C_i$  et  $CE_i$  étant l'ensemble des membres du cluster  $C_i$  susceptibles d'être élus  $CH_{Définitif}$ . Les ensembles  $Z_{im}$ ,  $SZ_{im}$  représentent respectivement les nœuds de la zone  $m$  et les membres appartenant à la sous zone  $m$  du cluster  $i$ .  $Dmax_{im}$  est la distance entre le  $CH_{Définitif}$   $i$  et le membre le plus éloigné de la zone  $m$ .

---

**Algorithme 5** Partitionnement d'un cluster  $C_i$  et formation des sous zones dans *RCLME*.

---

```

1:  $Dmax_{im} = 0$ 
2:  $CE_i = \emptyset$ 
3: Diviser  $C_i$  en quatre zones
4: Déterminer l'ensemble  $ZD_i$ 
5: Pour chaque ( $j \in ZD_i$ ) faire
6:   Pour chaque ( $k \in MC_i$ ) faire
7:     Si ( $Dist(j, k) \leq RC_j$ ) alors
8:        $CE_i = CE_i \cup j$           /* Mise à jour de l'ensemble  $CE_i$  */
9:     Finsi
10:  Finpour
11: Finpour
12: Si ( $|CE_i| = 0$ ) alors
13:    $Etat_i = CH_{Définitif}$ 
14:   Diviser  $C_i$  en quatre zones  $Z_{i1}, Z_{i2}, Z_{i3}$  et  $Z_{i4}$ 
15:   Association des  $MC_i$  à chaque zone
16:   Pour ( $m=1$  à  $4$ ) faire
17:      $SZ_{im} = \emptyset$ 
18:     Calculer  $Dmax_{im}$ 
19:     Pour chaque ( $t \in Z_{im}$ ) faire
20:       Si ( $Dist(t, i) \geq \frac{Dmax_{im}}{2}$ ) alors
21:          $SZ_{im} = t \cup SZ_{im}$ 
22:       Finsi
23:     Finpour
24:     Élection du  $SCH_{im}$  le plus proche ayant la valeur maximale de EV
25:     Envoyer NOTIFY_SCH( $i, SCH_{im}, Dmax_{im}$ )
26:   Finpour
27: Sinon
28:   Pour chaque ( $s \in CE_i$ ) faire
29:     Calculer  $EV_s$ 
30:   Finpour
31:   Élire  $s$  le plus proche du  $CH_{Temporaire}$  ayant le valeur maximale de EV comme  $CH_{Définitif}$ 
32:   Envoyer NOTIFY( $i, s, MC_i$ ) à  $MC_i$ 
33:   Répéter les étapes 14 :25 par le CH définitif  $s$ 
34: Finsi

```

---

### 3.4.3 Phase d'acheminement des données

L'acheminement des données se fera au moyen d'un routage multi-sauts intra et inter-cluster. L'intérêt de ce type de routage est de minimiser les coûts de communications des nœuds se situant loin du CH ou de la station de base [14]. Dans chaque sous zone, les membres transmettent les données collectées aux sous CHs. Ces derniers agrègent les données reçues de leurs membres et les envoient à leur tour au  $CH_{Définitif}$ . Les membres d'un cluster ne se trouvant dans aucune des sous zones formées envoient leurs données directement au  $CH_{Définitif}$  (figure 3.9).

Après L'agrégation des données reçues, les différents  $CH_{SDéfinitifs}$  effectuent un routage multi-sauts en envoyant leurs données aux  $CH_{SDéfinitifs}$  se trouvant à mi-chemin de la station de base.

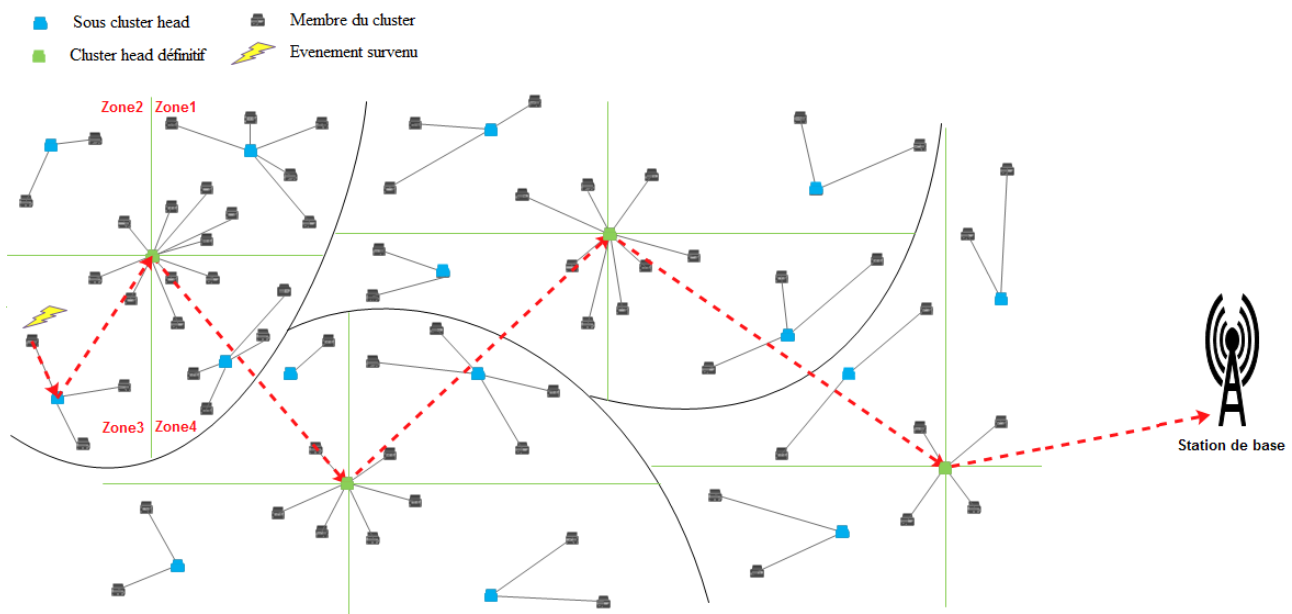


FIGURE 3.9 – Acheminement des données dans le protocole RCLME.

## 3.5 Conclusion

Dans ce chapitre, un protocole de clustering baptisé RCLME a été introduit. Le principe de ce protocole consiste en la réorganisation des clusters et le routage multi-sauts intra et inter-clusters. Dans le but de réaliser l'équilibrage de charges, RCLME utilise deux mécanismes : le premier consiste à migrer le rôle des CHs et à former des sous zones, le second se base sur un routage multi-sauts pour réduire les distances séparant les nœuds du réseau.

# Chapitre 4

## Simulation et évaluation de performances

### 4.1 Introduction

L'objectif de ce travail était de proposer une solution pour la réorganisation des clusters en vue d'une répartition équitable de la charge entre les différents nœuds du réseau. Le processus de conception de l'algorithme se base sur deux mécanismes à savoir : la réorganisation des cluster et le routage multi-sauts.

Le présent chapitre est dédié à l'évaluation de performances de l'algorithme proposé RCLME (**R**éorganisation des **C**lusters à base de **L**ocalisation pour un routage **M**ulti-sauts avec efficacité **E**nergétique) dans le chapitre 3. Le but de cette évaluation est de montrer l'intérêt et le bon fonctionnement de la solution proposée à travers des scénarios de simulation. Pour cela, nous avons développé un simulateur écrit en MATLAB afin de comparer les résultats obtenus avec ceux des protocoles LEACH [23] et FZ-LEACH [28] en matière de : nombre de nœuds vivants, énergie consommée et longévité du réseau.

### 4.2 Description de l'environnement de simulation

MATLAB (MATrix LABoratory) est un langage de programmation, calcul scientifique et de visualisation graphique développé par la société The MathWorks. Il permet de manipuler des matrices, d'afficher des courbes et des données, de mettre en œuvre des algorithmes, etc. Les utilisateurs de MATLAB sont de milieux très différents allant de l'ingénierie à l'économie dans un contexte aussi bien industriel que pour la recherche. MATLAB peut s'utiliser à l'aide de *toolboxes* (boîte à outils) [1].

### 4.3 Résultats de simulation

Le déploiement réel d'un réseau de capteurs est de nos jours très coûteux, complexe et difficile à mettre en œuvre. C'est pourquoi une simulation sous MATLAB se voit imposée. Les résultats

obtenus dans le cas du protocole RCLME sont comparés à ceux des protocoles LEACH [23] et FZ-LEACH [28]. Il est à noter que l'énergie consommée en émission des données, voire en réception, est calculée en utilisant le modèle radio décrit dans le chapitre 2.

Les nœuds sont supposés être homogènes, statiques et possèdent la même quantité d'énergie initiale. Les liens de transmission sont symétriques, c.à.d. l'énergie nécessaire pour transmettre un message d'un nœud  $i$  à un nœud  $j$  est la même que celle requise pour transmettre un message du nœud  $j$  au nœud  $i$ . La station de base dispose de ressources énergétiques, capacité mémoire et puissance de calcul théoriquement illimitées.

### 4.3.1 Paramètres de simulation

Deux scénarios de simulation sont considérés lors du déploiement aléatoire des capteurs. Le premier consiste à fixer le nombre de capteurs à 100 tout en variant le nombre de rounds. Le second scénario a pour objectif d'évaluer la scalabilité de la solution proposée en variant le nombre de capteurs de 100 à 600. Les résultats obtenus sont la moyenne de 10 exécutions. Le tableau ci-dessous récapitule les paramètres de simulation :

Paramètre	Valeur
Dimension de la zone de déploiement	100x100 $m^2$
Coordonnées de la station de base	(50, 175)
Énergie d'amplification de l'espace libre $\epsilon_{fs}$	10 pJ/bit/ $m^2$
Énergie d'amplification du multi-chemins $\epsilon_{mp}$	0.0013 pJ/bit/ $m^4$
Énergie électronique $E_{elec}$	50 nJ/bit
Énergie d'agrégation $E_{DA}$	50 nJ/bit
Taille du paquet de contrôle	25 bit
Taille du paquet de données	500 bit
Énergie initiale	1J
Pourcentage désiré de cluster heads	0,1

TABLE 4.1 – Paramètres de simulation.

### 4.3.2 Discussion et comparaison des résultats

Afin de montrer la valeur ajoutée de l'approche de clustering proposée, deux protocoles sont évalués pour les raisons suivantes :

- *LEACH* est le premier protocole qui fut introduit le concept du *clustering* dans les réseaux de capteurs.
- *FZ-LEACH* en raison de la formation des sous zones. Ce principe est aussi utilisé dans le protocole proposé.

Les performances des algorithmes susmentionnés sont évaluées en se basant sur les mesures suivantes :

- *Nœuds vivants* : représente le nombre moyen de nœuds disposant d'une quantité d'énergie suffisante pour servir au fonctionnement du réseau.
- *Quantité d'énergie* : mesure, d'une part, l'énergie résiduelle en joules de tout le réseau et, d'autre part, l'énergie moyenne consommée par un nœud durant un round.
- *Durée de vie* : représente le nombre moyen de rounds durant lesquels le réseau assure la fonctionnalité pour laquelle il a été déployé.

#### 4.3.2.1 Nœuds vivants

La figure 4.1 illustre l'évolution du nombre de nœuds vivants dans le réseau en fonction du nombre de rounds.

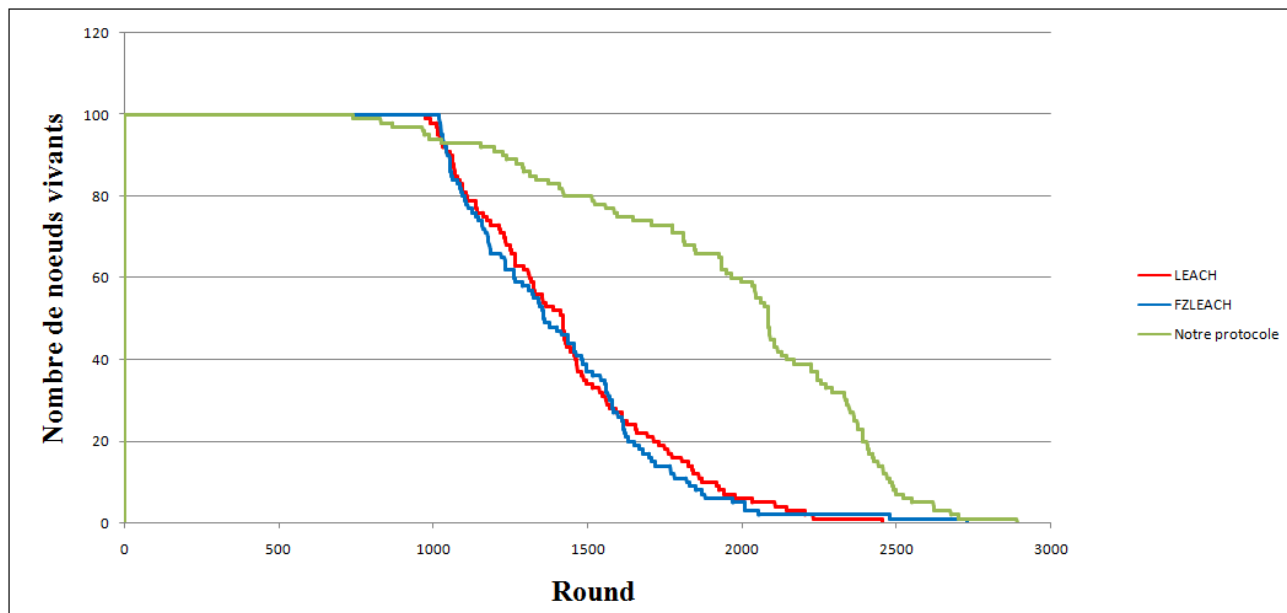


FIGURE 4.1 – Nombre de nœuds vivants *versus* nombre de rounds.

La figure montre que le protocole proposé engendre un nombre plus important de nœuds vivants le long de la durée de vie du réseau en comparaison avec LEACH et FZ-LEACH. En effet, le protocole proposé permet, grâce à la migration du CH et au routage multi-sauts, d'équilibrer la charge au sein du réseau et de réduire la défaillance des nœuds. En revanche, la politique d'élection probabiliste (partage non-équitable des tâches du réseau) adoptée par LEACH et FZ-LEACH, ainsi que la communication directe entre les CHs et la station de base, sont les causes principales de la dégradation continue du nombre de nœuds vivants dans le réseau.

Bien que, la défaillance du premier nœud dans le cas du protocole proposé, est plus rapide à cause des différentes configurations (localisation, partage du cluster et migration du CH), le nombre de nœuds vivants diminue légèrement au cours des rounds tout en restant supérieur en

comparaison aux protocoles LEACH et FZ-LEACH.

#### 4.3.2.2 Quantité d'énergie

La figure 4.2 montre l'énergie résiduelle totale dans le cas des trois protocoles en fonction du nombre de rounds.

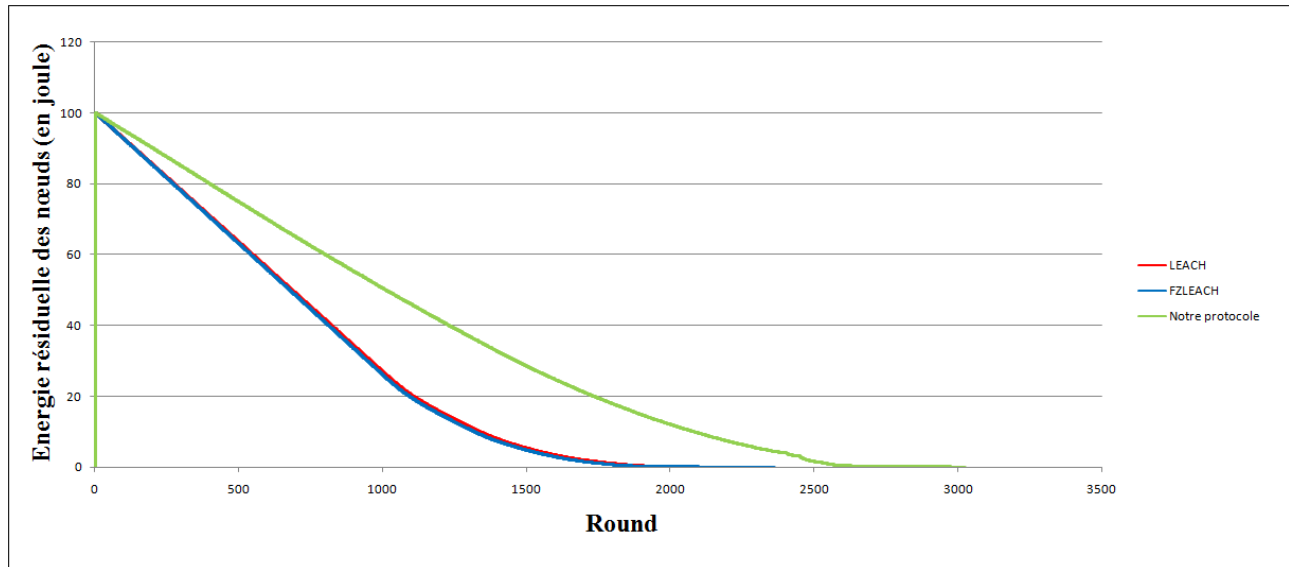
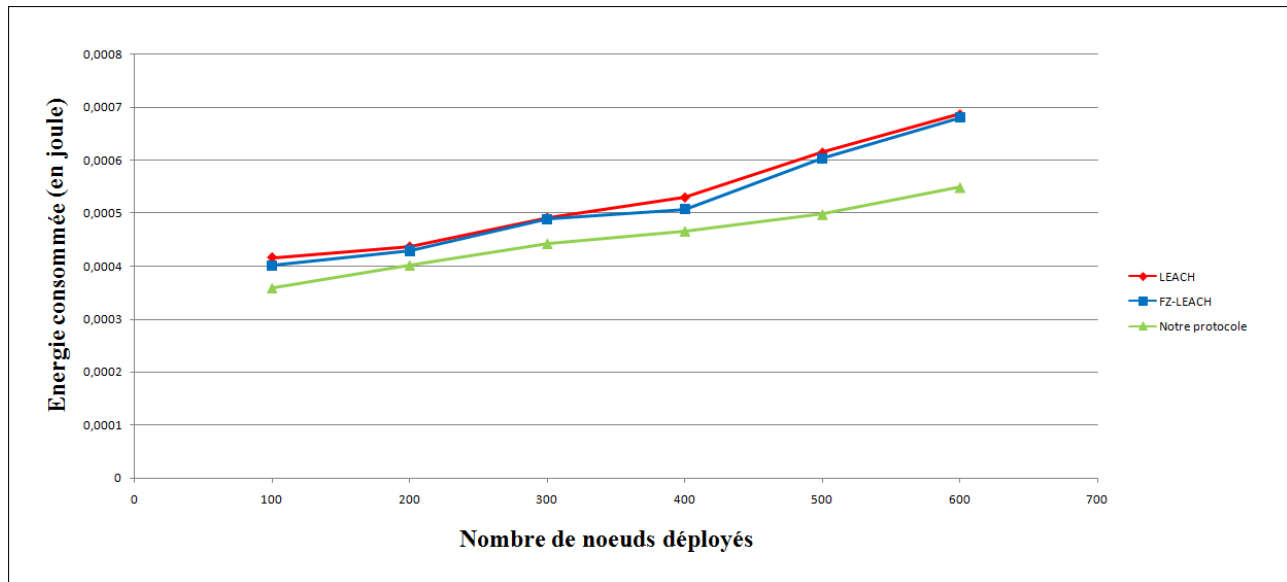


FIGURE 4.2 – Énergie résiduelle des nœuds *versus* nombre de rounds.

Les résultats indiquent que l'énergie résiduelle engendrée par le protocole proposé est nettement supérieure à celle de LEACH et FZ-LEACH, allongeant ainsi la longévité du réseau. En effet, RCLME utilise un routage multi-sauts (intra et inter-clusters) ce qui réduit l'énergie consommée par le réseau. En revanche, la communication directe sur une longue distance dans le cas des protocoles LEACH et FZ-LEACH, augmente les dépenses énergétiques et, par conséquent, entraîne la défaillance rapide des nœuds.

La figure 4.3 et le tableau 4.2 illustrent une comparaison entre les trois protocoles en termes d'énergie moyenne consommée par les nœuds du réseau en fonction du nombre de nœuds déployés.



FIGURE 4.3 – Consommation d'énergie *versus* nombre de nœuds déployés.

Le protocole proposé consomme moins d'énergie en comparaison avec les protocoles LEACH et FZ-LEACH et ce malgré l'augmentation du nombre de nœuds déployés. En matière d'énergie, le protocole proposé conserve lorsque le nombre de nœuds varie de 100 à 600, entre 16.14% et 25.22% (respectivement entre 12.26% et 23.95%) de l'énergie par rapport à LEACH (respectivement FZ-LEACH). En effet, le routage multi-sauts utilisé dans le protocole proposé, a permis de réduire les distances de transmission et, de ce fait, diminuer les dépenses énergétiques. A mesure que la densité du réseau augmente, les distances de transmission s'allongent et vu l'absence du routage multi-sauts, la consommation énergétique s'accroît dans le cas de LEACH et FZ-LEACH.

Protocole \ Nombre de nœuds	100	200	300	400	500	600
LEACH	$4,15973 \times 10^{-4}$	$4,36491 \times 10^{-4}$	$4,91159 \times 10^{-4}$	$5,29661 \times 10^{-4}$	$6,14628 \times 10^{-4}$	$6,87285 \times 10^{-4}$
FZ-LEACH	$4,02091 \times 10^{-4}$	$4,29553 \times 10^{-4}$	$4,88998 \times 10^{-4}$	$5,07872 \times 10^{-4}$	$6,035 \times 10^{-4}$	$6,80272 \times 10^{-4}$
RCLME	$3,58166 \times 10^{-4}$	$4,01284 \times 10^{-4}$	$4,42478 \times 10^{-4}$	$4,65983 \times 10^{-4}$	$4,97512 \times 10^{-4}$	$5,48847 \times 10^{-4}$

TABLE 4.2 – Énergie moyenne consommée *versus* nombre de nœuds déployés.

### 4.3.2.3 Durée de vie

La défaillance du dernier nœud du réseau est l'une des métriques de durée de vie les plus utilisées [61]. La figure 4.4 et le tableau 4.3 représentent une comparaison entre les trois protocoles en

termes de durée de vie (mesurée en nombre de rounds) séparant le déploiement du réseau jusqu'à la défaillance du dernier nœud.

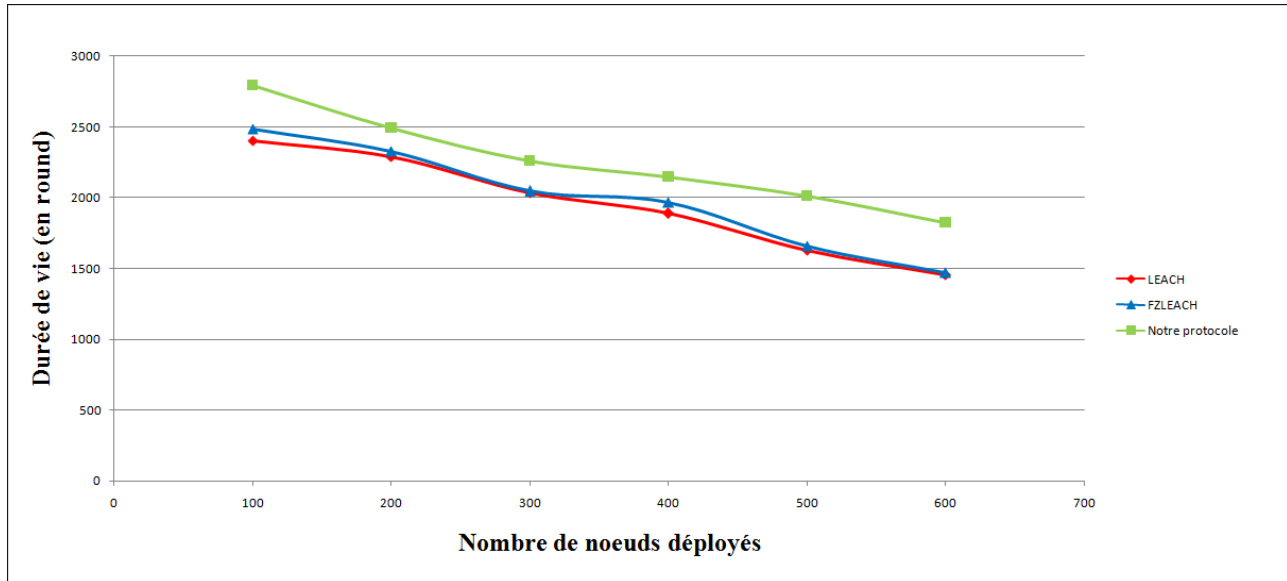


FIGURE 4.4 – Durée de vie jusqu'à la défaillance du dernier nœud *versus* nombre de nœuds déployés.

La figure 4.4 montre que la durée de vie engendrée par les algorithmes RCLME, LEACH et FZ-LEACH diminue lorsque le nombre de nœuds déployés augmente. Toutefois, l'algorithme proposé est largement plus efficace, et ce lorsque le nombre de nœuds varie de 100 à 600, avec une amélioration de 16.14% (respectivement 25.22%) par rapport à LEACH et de 12.26% (respectivement 23.95%) en comparaison à FZ-LEACH. Cette amélioration est due à la consommation équitable de l'énergie dans les clusters grâce à la formation des sous zones et au routage multi-sauts qui réduisent les distances de transmission. Ce résultat est attendu du fait que les dépenses énergétiques de l'algorithme proposé sont réduites par rapport à LEACH et FZ-LEACH (voir figure 4.3).

Protocole \ Nombre de nœuds	100	200	300	400	500	600
LEACH	2404	2291	2036	1888	1627	1455
FZ-LEACH	2487	2328	2045	1969	1657	1470
RCLME	2792	2492	2260	2146	2010	1822

TABLE 4.3 – Défaillance du dernier nœud du réseau *versus* nombre de nœuds déployés.

## 4.4 Conclusion

Ce chapitre a été consacré à l'évaluation de performances et à l'étude de la scalabilité du protocole proposé. Pour cela une comparaison aux protocoles LEACH et FZ-LEACH en termes de

durée de vie et de consommation énergétique a été réalisée.

Les résultats de simulation obtenus montrent que le protocole proposé réalise des économies d'énergie importantes grâce à l'équilibrage de charges qu'il assure. Cela a permis d'étendre la durée de vie du réseau de façon considérable.

# Conclusion générale et perspectives

L'essor des réseaux de capteurs sans fil (RCSF) au cours des dernière années est principalement dû aux fonctionnalités qu'ils offrent à divers domaines (militaire, médical, etc.). Toutefois, les RCSF doivent faire face à de nombreux défis de conception en raison de leurs capacités restreintes notamment l'énergie. Cette dernière est une ressource critique qui constitue souvent un obstacle majeur lors de l'acheminement de données. Il devient alors primordial de concevoir des protocoles de routage permettant de minimiser la consommation énergétique. L'un des mécanisme utilisés pour résoudre ce problème est la réorganisation des clusters et le routage multi-sauts afin d'assurer un répartition équitable de la charge au sein du du réseau. Dans cette optique, le protocole RCLME (**R**éorganisation des **C**lusters à base de **L**ocalisation pour unroutage **M**ulti-sauts avec efficacité **É**nergétique) a été proposé.

Le protocole RCLME vise à résoudre certaines limites des protocoles LEACH et FZ-LEACH telles que l'élection probabiliste des cluster heads, la communication directe avec la station de base, ect. Ce protocole se déroule en trois phases : **1**) une phase de *configuration des clusters* similaire à celle de l'algorithme LEACH [23], **2**) une phase de *réorganisation des clusters* permettant le partitionnement des clusters en quatre zones et **3**) une phase d'*acheminement des données*. La réorganisation des clusters consiste en la migration du rôle du cluster head vers un meilleur candidat en termes d'énergie, de distance avec les membres du cluster et du nombre de voisins. Cela pour diverses raisons : (1) la consommation énergétique des cluster-heads est plus élevée que celle des membres, (2) le choix du cluster-heads ayant le plus de voisins permet de réduire les dépenses énergétiques lors des différents échanges. Dans RCLME, chaque cluster est partitionné en quatre zones virtuelles. Dans chaque zone est formée une sous zone où un sous cluster head est élu pour l'agrégation et l'acheminement des données au CH. Un routage multi-sauts inter-clusters est effectué afin de réduire la distance de transmission des CHs et ainsi minimiser la consommation énergétique.

L'évaluation de performance du protocole RCLME a été faite sous MATLAB. Les résultats de simulations obtenus ont montré que RCLME offrent de meilleures performances en termes du nombre de nœuds vivants, de consommation énergétique et de durée de vie du réseau en comparaison avec les protocoles LEACH et FZ-LEACH.

En guise de perspectives, il sera judicieux :

- D'adapter la solution aux réseaux de capteurs mobiles. Cependant, la connaissance des po-

sition des capteurs est requise ;

- d'utiliser plusieurs stations de base qui permettrait de réduire les distances de transmission (par exemple, les nœuds envoient leurs données à la station de base la plus proche) ;
- De tester le support de l'hétérogénéité par RCLME. En effet, les nœuds d'un RCSF peuvent être dotés de capacités différentes (rayon de transmission, énergie, etc.) ;
- De simuler sur des simulateurs tels que OMNET, OPNET, NS3, etc. ;
- D'expérimenter RCLME sur une plate-forme réelle de capteurs.

# Bibliographie

- [1] [http://www.mathworks.com/help/matlab/learn\\_matlab/product-description.html](http://www.mathworks.com/help/matlab/learn_matlab/product-description.html), (Consulté le 30 mai 2016).
- [2] A. A. Abbasi and M. Younis. A survey on clustering algorithms for wireless sensor networks. *Computer communications*, 30(14) :2826–2841, 2007.
- [3] P. K. Agarwal and C. M. Procopiuc. Exact and approximation algorithms for clustering. *Algorithmica*, 33(2) :201–226, 2002.
- [4] K. Akkaya and M. Younis. A survey on routing protocols for wireless sensor networks. *Ad hoc networks*, 3(3) :325–349, 2005.
- [5] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. Wireless sensor networks : a survey. *Computer networks*, 38(4) :393–422, 2002.
- [6] J. N. Al-Karaki and A. E. Kamal. Routing techniques in wireless sensor networks : a survey. *Wireless communications, IEEE*, 11(6) :6–28, 2004.
- [7] S. Bandyopadhyay and E. J. Coyle. An energy efficient hierarchical clustering algorithm for wireless sensor networks. In *Twenty-Second Annual Joint Conference of the IEEE Computer and Communications*, volume 3, pages 1713–1723. IEEE, 2003.
- [8] S. Basagni, I. Chlamtac, V. R. Syrotiuk, and B. A. Woodward. A distance routing effect algorithm for mobility (dream). In *the 4th annual ACM/IEEE international conference on Mobile computing and networking*, pages 76–84. ACM, 1998.
- [9] D. Čevizović, S. Galović, S. Zeković, and Z. Ivić. Boundary between coherent and noncoherent small polaron motion : Influence of the phonon hardening. *Physica B : Condensed Matter*, 404(2) :270–274, 2009.
- [10] M. Chatterjee, S. K. Das, and D. Turgut. Wca : A weighted clustering algorithm for mobile ad hoc networks. *Cluster Computing*, 5(2) :193–204, 2002.
- [11] M. Chen, S. Gonzalez, Y. Zhang, and V. C. Leung. Multi-agent itinerary planning for wireless sensor networks. In *Quality of Service in Heterogeneous Networks*, pages 584–597. Springer, 2009.
- [12] M. Chen, C.-F. Lai, and H. Wang. Mobile multimedia sensor networks : architecture and routing. *EURASIP Journal on Wireless Communications and Networking*, 2011(1) :1–9, 2011.
- [13] P. Ding, J. Holliday, and A. Celik. Distributed energy-efficient hierarchical clustering for wireless sensor networks. In *Distributed computing in sensor systems*, pages 322–339. Springer, 2005.

- 
- [14] M. Feldman, J. Chuang, I. Stoica, and S. Shenker. Hidden-action in multi-hop routing. In *the 6th ACM conference on Electronic commerce*, pages 117–126. ACM, 2005.
- [15] S. Ghiasi, A. Srivastava, X. Yang, and M. Sarrafzadeh. Optimal energy aware clustering in sensor networks. *Sensors*, 2(7) :258–269, 2002.
- [16] S. Giannoulis, C. Antonopoulos, E. Topalis, and S. Koubias. Zrp versus dsr and tora : A comprehensive survey on zrp performance. In *10th IEEE Conference on Emerging Technologies and Factory Automation*, volume 1, pages 8–15. IEEE, 2005.
- [17] G. Gupta and M. Younis. Fault-tolerant clustering of wireless sensor networks. In *Wireless Communications and Networking*, volume 3, pages 1579–1584. IEEE, 2003.
- [18] M. Haneef, Z. Wenxun, and Z. Deng. Mg-leach : Multi group based leach an energy efficient routing algorithm for wireless sensor network. In *14th International Conference on Advanced Communication Technology (ICACT)*, pages 179–183. IEEE, 2012.
- [19] T. He, J. Stankovic, C. Lu, T. Abdelzaher, et al. Speed : A stateless protocol for real-time communication in sensor networks. In *the 23rd International Conference on Distributed Computing Systems*, pages 46–55. IEEE, 2003.
- [20] S. M. Hedetniemi, S. T. Hedetniemi, and A. L. Liestman. A survey of gossiping and broadcasting in communication networks. *Networks*, 18(4) :319–349, 1988.
- [21] W. Heinzelman. Application-specific protocol architectures for wireless networks [ph. d. thesis]. *Boston : Massachusetts Institute of Technology*, 2000.
- [22] W. B. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan. An application-specific protocol architecture for wireless microsensor networks. *IEEE Transactions on Wireless Communications*, 1(4) :660–670, 2002.
- [23] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan. Energy-efficient communication protocol for wireless microsensor networks. In *the 33rd annual Hawaii international conference on System sciences*, pages 10–27. IEEE, 2000.
- [24] W. R. Heinzelman, J. Kulik, and H. Balakrishnan. Adaptive protocols for information dissemination in wireless sensor networks. In *the 5th annual ACM/IEEE international conference on Mobile computing and networking*, pages 174–185. ACM, 1999.
- [25] X. Hou, D. Tipper, and J. Kabara. Label-based multipath routing (lmr) in wireless sensor networks. In *The International Symposium on Advanced Radio Technologies (ISART)*, 2004.
- [26] C. Intanagonwiwat, R. Govindan, and D. Estrin. Directed diffusion : a scalable and robust communication paradigm for sensor networks. In *the 6th annual international conference on Mobile computing and networking*, pages 56–67. ACM, 2000.
- [27] H. Karl and A. Willig. *Protocols and architectures for wireless sensor networks*. John Wiley & Sons, 2007.
- [28] V. Katiyar, N. Chand, G. C. Gautam, and A. Kumar. Improvement in leach protocol for large-scale wireless sensor networks. In *International Conference on Emerging Trends in Electrical and Computer Technology (ICETECT)*, pages 1070–1075. IEEE, 2011.

- 
- [29] V. Kawadia and P. Kumar. Power control and clustering in ad hoc networks. In *Twenty-Second Annual Joint Conference of the IEEE Computer and Communications*, volume 1, pages 459–469. IEEE, 2003.
- [30] A. Khamiss, C. Senchun, Z. Baihai, and C. Lingguo. Combined metrics-clustering algorithm based on leach-c. In *27th Chinese Control and Decision Conference (CCDC)*, pages 5252–5257. IEEE, 2015.
- [31] R. K. Kodali and N. K. Aravapalli. Multi-level leach protocol model using ns-3. In *Advance Computing Conference (IACC)*, pages 375–380. IEEE, 2014.
- [32] R. K. Kodali and N. Sarma. Energy efficient routing protocols for wsn’s. In *International Conference on Computer Communication and Informatics (ICCCI)*, pages 1–4. IEEE, 2013.
- [33] N. Kumar, P. Bhutani, P. Mishra, et al. U-leach : A novel routing protocol for heterogeneous wireless sensor networks. In *International Conference on Communication, Information & Computing Technology (ICCICT)*, pages 1–4. IEEE, 2012.
- [34] N. Kumar and J. Kaur. Improved leach protocol for wireless sensor networks. In *7th International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM)*, pages 1–5. IEEE, 2011.
- [35] H. Lim and C. Kim. Multicast tree construction and flooding in wireless ad hoc networks. In *Proceedings of the 3rd ACM international workshop on Modeling, analysis and simulation of wireless and mobile systems*, pages 61–68. ACM, 2000.
- [36] S. Lindsey and C. S. Raghavendra. Pegasis : Power-efficient gathering in sensor information systems. In *Aerospace conference proceedings*, volume 3, pages 3–1125. IEEE, 2002.
- [37] X. Liu. A survey on clustering routing protocols in wireless sensor networks. *Sensors*, 12(8) :11113–11153, 2012.
- [38] H. Luo, F. Ye, J. Cheng, S. Lu, and L. Zhang. Ttdd : Two-tier data dissemination in large-scale wireless sensor networks. *Wireless Networks*, 11(1-2) :161–175, 2005.
- [39] Z. Manap, B. M. Ali, C. K. Ng, N. K. Noordin, and A. Sali. A review on hierarchical routing protocols for wireless sensor networks. *Wireless personal communications*, 72(2) :1077–1104, 2013.
- [40] A. Manjeshwar and D. P. Agrawal. Teen : a routing protocol for enhanced efficiency in wireless sensor networks. page 30189a. IEEE, 2001.
- [41] B. Manzoor, N. Javaid, O. Rehman, M. Akbar, Q. Nadeem, A. Iqbal, and M. Ishfaq. Q-leach : A new routing protocol for wsns. In *Procedia Computer Science*, volume 19, pages 926–931. Elsevier, 2013.
- [42] R. Mehta, A. Pandey, and P. Kapadia. Reforming clusters using c-leach in wireless sensor networks. In *International Conference on Computer Communication and Informatics (ICCCI)*, pages 1–4. IEEE, 2012.
- [43] T. Murata and H. Ishibuchi. Performance evaluation of genetic algorithms for flowshop scheduling problems. In *the First IEEE Conference on Evolutionary Computation*, pages 812–817. IEEE, 1994.



- 
- [44] S. Murthy and J. J. Garcia-Luna-Aceves. An efficient routing protocol for wireless networks. *Mobile Networks and Applications*, 1(2) :183–197, 1996.
- [45] J. Newsome and D. Song. Gem : Graph embedding for routing and data-centric storage in sensor networks without geographic information. In *the 1st international conference on Embedded networked sensor systems*, pages 76–88. ACM, 2003.
- [46] R. Ogier, F. Templin, and M. Lewis. Topology dissemination based on reverse-path forwarding (tbrpf). Technical report, 2004.
- [47] N. Pantazis, S. A. Nikolidakis, and D. D. Vergados. Energy-efficient routing protocols in wireless sensor networks : A survey. *Communications Surveys & Tutorials, IEEE*, 15(2) :551–591, 2013.
- [48] M. Patil and R. C. Biradar. A survey on routing protocols in wireless sensor networks. In *18th IEEE International Conference on Networks(ICON)*, pages 86–91. IEEE, 2012.
- [49] L. Qing, Q. Zhu, and M. Wang. Design of a distributed energy-efficient clustering algorithm for heterogeneous wireless sensor networks. *Computer communications*, 29(12) :2230–2237, 2006.
- [50] H. Raghunandan, G. and N. Lakshmi, B. A comparative analysis of routing techniques for wireless sensor networks. In *National Conference on Innovations in Emerging Technology (NCOIET)*, pages 17–22. IEEE, 2011.
- [51] J. Raju and J. J. Garcia-Luna-Aceves. A new approach to on-demand loop-free multipath routing. In *Eight International Conference on Computer Communications and Networks*, pages 522–527. IEEE, 1999.
- [52] P. Rawat, K. D. Singh, H. Chaouchi, and J. M. Bonnin. Wireless sensor networks : a survey on recent developments and potential synergies. *The Journal of Supercomputing*, 68(1) :1–48, 2014.
- [53] N. Sadagopan, B. Krishnamachari, and A. Helmy. The acquire mechanism for efficient querying in sensor networks. In *IEEE International Workshop on Sensor Network Protocols and Applications*, pages 149–155. IEEE, 2003.
- [54] T. Sharma, R. Joshi, and M. Misra. Gbdd : Grid based data dissemination in wireless sensor networks. In *the 16th International Conference on Advanced Computing and Communications*, pages 234–240. IEEE, 2008.
- [55] E. Shih, S.-H. Cho, N. Ickes, R. Min, A. Sinha, A. Wang, and A. Chandrakasan. Physical layer driven protocol and algorithm design for energy-efficient wireless sensor networks. In *the 7th annual international conference on Mobile computing and networking*, pages 272–287. ACM, 2001.
- [56] C. P. Singh, P. Vyas, O., and M. K. Tiwari. A profound survey of sensor networks & related routing protocols. In *4th International Conference on Wireless Communications, Networking and Mobile Computing.*, pages 1–5. IEEE, 2008.
- [57] S. P. Singh and C. Sharma, S. A survey on cluster based routing protocols in wireless sensor networks. *Procedia Computer Science*, 45 :687–695, 2015.

- 
- [58] S. Sirsikar and S. Anavatti. Issues of data aggregation methods in wireless sensor network : A survey. *Procedia Computer Science*, 49 :194–201, 2015.
- [59] K. Sohrabi, J. Gao, V. Ailawadhi, and G. J. Pottie. Protocols for self-organization of a wireless sensor network. *IEEE personal communications*, 7(5) :16–27, 2000.
- [60] K. Sohrabi, J. Gao, V. Ailawadhi, and G. J. Pottie. Protocols for self-organization of a wireless sensor network. *IEEE personal communications*, 7(5) :16–27, 2000.
- [61] D. Tian and N. D. Georganas. A coverage-preserving node scheduling scheme for large wireless sensor networks. In *Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications*, pages 32–41. ACM, 2002.
- [62] S. Varshney, C. Kumar, and A. Swaroop. A comparative study of hierarchical routing protocols in wireless sensor networks. In *2nd International Conference on Computing for Sustainable Global Development(INDIACom)*, pages 1018–1023. IEEE, 2015.
- [63] H. K. A. Willig and H. Karl. Protocols and architectures for wireless sensor networks. *England : John Wiley & Sons*, pages 17–57, 2005.
- [64] Y. Xu, J. Heidemann, and D. Estrin. Geography-informed energy conservation for ad hoc routing. In *the 7th annual international conference on Mobile computing and networking*, pages 70–84. ACM, 2001.
- [65] Y. Yao and J. Gehrke. The cougar approach to in-network query processing in sensor networks. *ACM Sigmod Record*, 31(3) :9–18, 2002.
- [66] F. Ye, G. Zhong, S. Lu, and L. Zhang. Gradient broadcast : A robust data delivery protocol for large scale sensor networks. *Wireless Networks*, 11(3) :285–298, 2005.
- [67] O. Younis and S. Fahmy. Heed : a hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks. *IEEE Transactions on Mobile Computing*, 3(4) :366–379, 2004.
- [68] F. Yu, Y. Li, F. Fang, and Q. Chen. A new tora-based energy aware routing protocol in mobile ad hoc networks. In *3rd IEEE/IFIP International Conference in Central Asia on Internet*, pages 1–4. IEEE, 2007.
- [69] Y. Yu, R. Govindan, and D. Estrin. Geographical and energy aware routing : A recursive data dissemination protocol for wireless sensor networks. Technical report, UCLA Computer Science Department, 2001.

## RÉSUMÉ

Les récents progrès technologiques qu'à connu l'univers de la communication sans fil en termes de supports de communication et de miniaturisation des dispositifs ont facilité la collaboration de plusieurs capteurs pour former un réseau de capteurs sans fil. Pour régir les différentes fonctionnalités, les protocoles conçus pour ces réseaux doivent faire face aux capacités restreintes des capteurs telles que les ressources énergétiques. La problématique traitée vise à résoudre certains limites des protocoles LEACH et FZ-LEACH telles que la surcharge des clusters, la communication directe entre les clusters head (CHs) et la station de base (SB) et l'élection probabiliste des CHs qui ne tient pas compte de l'énergie résiduelle des nœuds. Le protocole proposé baptisé **RCLME** (**R**éorganisation des **C**lusters à base de **L**ocalisation pour un routage **M**ulti-sauts avec efficacité **E**nergétique) est une solution se basant sur la réorganisation des clusters (migration du cluster head et formation des sous zones) et le routage mutli-sauts intra et inter cluster. La réduction de la consommation énergétique dans **RCLME** est la conséquence de : **i**) la migration du rôle de cluster head vers un meilleur candidat, **ii**) la formation des sous zones pour les nœuds membres se situant loin de leurs CHs ainsi que **iii**) l'utilisation d'un routage multi-sauts lors de l'acheminement de données. L'évaluation de performances a montré l'efficacité et la scalabilité du protocole proposé en termes de nombre de nœuds vivants et en matière d'énergie consommée et durée de vie du réseau en comparaison avec d'autres protocoles proposés dans la littérature.

**Mots clés :** Réseau de capteurs sans fil, multi-sauts, clustering, migration du cluster head, réorganisation des clusters, distances de transmission, LEACH, FZ-LEACH.

## ABSTRACT

The recent technological advances that experienced the wireless world communication in terms of communication media and miniaturization of devices, have facilitated the cooperation of many sensor to form a wireless sensor network. To control the various features, the designed protocols for this networks have to face the limited capacity of sensors such as energy resources. This work has been proposed on the idea of solving some of the limits of the protocols LEACH and FZ-LEACH such as clusters overhead, the direct communication between cluster head (CHs) and the base station (SB) and probabilistic election of the CHs that ignores residual energy of the nodes. The proposed protocol called **CRLEM** (**C**lusters **R**eorganization based on **L**ocalization for an **E**nergy efficient **M**ulti-hops routing protocol) is a solution for cluster reorganisation (cluster head migration and sub-zones formation) and multi-hops routing intra and inter-cluster. The reduction of energy consumption in **CRLEM** is the consequence of : **i**) cluster head migration to a better candidate, **ii**) sub-zones formation for the nodes member that are far from their CHs and **iii**) use of multi-hops routing. performances evaluation has shown the efficiency and the scalability of the proposed protocol in terms of number of alive nodes regarding energy consumption and network lifetime in comparison with other proposed protocol in litterature.

**Key words :** Wireless sensor network, multi-hops, clustering, cluster head migration, cluster reorganization, transmission distances, LEACH, FZ-LEACH.