



REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITE ABOU-BEKR BELKAID - TLEMCEM

THÈSE

Présentée à :

FACULTE DES SCIENCES – DEPARTEMENT D'INFORMATIQUE

Pour l'obtention du diplôme de :

DOCTORAT EN SCIENCES

Spécialité: Informatique

Par :

Mr BENAÏSSA Bedr-Eddine

Sur le thème

Détection et communication coopérative pour le déploiement des réseaux de capteurs

Soutenue publiquement le **05 Mars 2022** à Tlemcen devant le jury composé de :

Mr BENMAMMAR BADR	Professeur	Université de Tlemcen	Président
M ^{me} LAHFA FEDOUA	Professeur	Université de Tlemcen	Directrice de thèse
Mr. BOUCHIHA DJELLOUL	Professeur	Centre Univ. Naâma	Examinateur
Mr KHALFI FETHI	M.C.A.	Université Sidi Belabès	Examinateur
Mr MATALLAH HOCINE	M.C.A.	Université de Tlemcen	Examinateur
Mr BOUOUGADA BENAMAR	M.C.A.	Centre Univ. Naâma	Examinateur

*Laboratoire de recherche en informatique (LRI)
BP 119, 13000 Tlemcen – Algérie*

Je dédie cette thèse,

*A mes parents pour leur amour inestimable, leur confiance,
leur soutien, leurs sacrifices et toutes les valeurs qu'ils ont su
m'enseigner.*

*A la mère de mes enfants pour son soutien moral, et ses
encouragements.*

*A mes beaux-parents qu'ils trouvent ici le témoignage de
mon respect.*

*A mes enfants, mes frères et mes sœurs, j'espère que la vie
leurs réserve le meilleur.*

A toute ma famille ainsi qu'à mes amis.



Remerciements

Je remercie, tout d'abord, Dieu tout puissant de m'avoir donné le courage, la force et la patience d'achever ce modeste travail.

Je tiens tout d'abord à exprimer ma profonde gratitude au Professeur M^{me} **Lahfa Fedoua**, mon encadreur et enseignante à la Faculté des Sciences de l'Université de Tlemcen, de m'avoir encadré avec un intérêt constant et une grande compétence et patience ainsi pour l'intérêt qu'elle a bien voulu porter à mon travail.

J'exprime ma profonde reconnaissance à **Naima khatir, Kaid Noureddine**, enseignants au centre universitaire de Naâma pour leur soutien et les encouragements qui m'ont permis de mener à bien ce travail.

Que Messieurs les jurys, trouvent ici l'expression de mes remerciements les plus sincères d'avoir accepté d'examiner ce mémoire.

J'exprime également mes remerciements à Mr. Tadj Ahmed, Mr. Hadji Mohammed, Bouyahia S. mes étudiants dont la compagnie en contexte professionnel était réellement enrichissante.

Je remercie toutes les personnes qui ont participé de manière directe ou indirecte à la concrétisation de ce travail.



« On a souvent dit et répété que les livres se font avec les livres et que rien n'est plus facile que de copier. Mais, pour rendre utiles tous les matériaux rassemblés en vue d'une publication spéciale, avoir de la justesse dans l'esprit et ramener toutes ses pensées sur un même sujet, préciser toutes les opinions, quel est le fond et la forme que l'on doit adopter, il faut s'éclairer de toutes les divergences, et toutes les critiques, et ne publier que ce qui à un intérêt indiscutable pour créer un livre substantiel. Il ne s'agit plus seulement de copier, il faut du discernement, du goût, du travail et de l'expérience. »

Thomas Brunton, Esquisses morales et littéraires (1874)



Résumé

L'exploitation d'un réseau de capteurs soulève de nombreux problèmes et ce à plusieurs niveaux algorithmiques : localisation, déploiement, collecte de données, couverture et réduction de la consommation d'énergie des batteries, afin d'optimiser la durée de vie du réseau. Ce dernier point a suscité particulièrement l'attention des chercheurs. Dans un tel réseau, et plus particulièrement avec une politique de déploiement à saut unique, les mesures des capteurs contiennent beaucoup de redondance, soit dans les dimensions de mesure d'un seul capteur, soit entre les dimensions de mesure de différents capteurs en raison de la corrélation spatiale ou dans la dimension temporelle des mesures. L'objectif est d'étudier la détection et la coopération pour en tirer des conditions aidant à mieux positionner des capteurs dans une zone de déploiement donnée, tout en garantissant certaines contraintes liées à ce type de réseaux, telles que le coût de déploiement ainsi que la durée de vie du réseau. Deux approches ont été proposées. La première propose de minimiser la complexité en termes de communication et de calcul en se basant sur un système d'agrégation et de consensus pour réduire la dimension spatio-temporelle des données capturées et par conséquent le nombre de capteurs déployés. Les résultats montrent une performance visible par rapport à la méthode standard de transmission sur la plateforme libre du simulateur COOJA / Contiki permettant de simuler des connexions réseau de capteurs sans fil et d'interagir avec eux. La deuxième contribution minimise les fréquences de transmission des données mesurées vers la station de base en catégorisant les données captées dans des classes prédéfinies, numérotées au préalable, que nous nous nommerons « intervalles de confiance », de cette manière chaque valeur captée sera rangée dans une classe et seul son numéro sera envoyé à la station de base, si (et seulement si) un changement de classe par rapport à la valeur précédente est observé. Les résultats montrent que la collecte de données par intervalle de confiance réduit considérablement l'énergie des capteurs des nœuds. Ainsi, dans un réseau de capteur sans fil, un déploiement optimal est accentué par une bonne gestion de transmission de données vers le puits.

Mots clés : Déploiement des capteurs, agrégation de données, durée de vie d'un réseau, Indicateur de force du signal reçu (RSSI), protocole de routage IoT



Abstract

Operating a sensor network raises many issues at several algorithmic levels: localization, deployment, data collection, coverage, and reduction of battery power consumption to optimize network lifetime. This last point has been of particular interest to the researchers. In such a network, and especially with a single-hop deployment policy, sensor measurements contain a lot of redundancy, either in the measurement dimensions of a single sensor, or between the measurement dimensions of different sensors due to spatial correlation or in the temporal dimension of measurements. The goal is to study detection and cooperation to determine conditions that will help to better position sensors in a given deployment area, while guaranteeing certain constraints related to this type of network, such as the cost of deployment and the network's lifetime. Two approaches have been proposed. The first one proposes to minimize the complexity in terms of communication and computation by relying on an aggregation and consensus system to reduce the spatial and temporal dimension of the captured data and consequently the number of deployed sensors. The results show a visible performance compared to the standard transmission method on the open platform of the COOJA / Contiki simulator allowing to simulate wireless sensor network connections and to interact with them. The second contribution minimizes the transmission frequencies of the measured data to the base station by categorizing the captured data into predefined, pre-numbered classes, which we will call "confidence intervals". In this way, each captured value will be classified into a class and only its number will be sent to the base station, if (and only if) a class change with respect to the previous value is observed. The results show that interval-based data collection significantly reduces the energy of the motes' sensors. Thus, in a wireless sensor network, optimal deployment is accentuated by good data transmission management to the sink.

Keywords: Sensor Deployment, Data aggregation, Network Lifetime, Received signal Strength Indicator (RSSI), IoT Routing Protocol



ملخص

تثير تطبيقات شبكة الاستشعار العديد من المشاكل، وذلك على مستوى عدة خوارزميات من بينها: تحديد الموقع والنشر وجمع البيانات والتغطية وكذا تخفيض طاقة استهلاك البطاريات من أجل تحسين عمر تشغيل هذه الشبكة.

إنّ قياسات المستشعر تحتوي على الكثير من التكرار، إما في أبعاد القياس لمستشعر واحد أو بين أبعاد القياس لأجهزة الاستشعار المختلفة بسبب الارتباط المكاني أو في البعد الزمني للقياسات، فلقد كانت خوارزمية النشر وخاصة المتعلقة بالشبكات وحيدة القفزة محلّ انتباه واهتمام الباحثين في هذا المجال.

إنّ الهدف من هذه الدراسة هو الاكتشاف والتعاون من أجل اشتقاق الظروف التي تساعد على الانتشار الأمثل للمستشعرات، مع ضمان دقة في القياسات وأقصى عمر ممكن للمستشعرات المستخدمة. تم اقتراح مقاربتين لحل مشكلة الدراسة. تمثّلت الأولى في تقليل التعقيد من حيث الاتصال والحساب استنادًا إلى نظام التجميع والإجماع لتقليل البعد المكاني والزمني للبيانات التي تم التقاطها وبالتالي عدد أجهزة الاستشعار الموزعة. أظهرت نتائجها جودة في الأداء مقارنة بالطريقة القياسية للإرسال على النظام الأساسي المجاني لمحاكاة COOJA / Contiki حيث سمحت بمحاكاة اتصالات الشبكة لأجهزة الاستشعار اللاسلكية والتفاعل معها.

تمثّلت المقاربة الثانية في التقليل من ترددات إرسال البيانات المقاسة إلى المحطة الأساسية من خلال تصنيف البيانات المنقطة في فئات محددة ومرقمة مسبقًا، والتي أطلقنا عليها اسم "فترات الثقة"؛ وبهذه الطريقة سيتم تخزين كل قيمة تم التقاطها في قائمة، ثم إرسال الفئة ورقمها فقط إلى المحطة الأساسية، في حالة اختلاف قيمة الفئة الجديدة عن السابقة. أظهرت النتائج أنّ جمع بيانات فاصل الثقة يقلل بشكل كبير من طاقة أجهزة استشعار الحركة.

وبالتالي تم استخلاص أنّه في شبكات الاستشعار اللاسلكية، يتمّ تعزيز النشر الأمثل من خلال الإدارة الجيدة لنقل البيانات إلى المحطة الأساسية.

الكلمات المفتاحية: توزيع المستشعر، تجميع البيانات، حياة شبكة المستشعر، مؤشر قوة الإشارة المستقبلية (RSSI)، بروتوكول توجيه إنترنت الأشياء

Sommaire

Chapitre I. Introduction générale	1
I.1. CONTEXTE	1
I.2. PROBLEMATIQUE	3
I.3. CONTRIBUTION DE LA THESE.....	3
I.4. ORGANISATION DE LA THESE.....	4
Chapitre II. Architecture des réseaux de capteurs sans fil(RCSF)	5
II.1. INTRODUCTION	6
II.2. NŒUD : DESCRIPTION PHYSIQUE.....	7
II.3. MODELE DE NŒUDS.....	8
II.4. ROLE DES NŒUDS DE CAPTEURS	9
II.5. DESCRIPTION DES TOPOLOGIES DES RCSF	10
II.5.1. <i>Topologie peer to peer</i>	11
II.5.2. <i>Topologie en étoile</i>	11
II.5.3. <i>Topologie en maille (mesh)</i>	11
II.5.4. <i>Topologie hiérarchique</i>	12
II.5.5. <i>Éléments clés d'un réseau de capteurs</i>	12
II.6. STRUCTURE ET MODELES DANS UN RESEAU DE CAPTEURS.....	13
II.6.1. <i>Modèle de communication du RCSF</i>	13
II.6.1.1. Protocole de communication	13
a) Modèle coopératif	13
b) Modèle non coopératif	14
II.6.1.2. Communication d'infrastructure	14
II.6.2. <i>Modèle de collecte de données</i>	17
II.6.3. <i>Modèle de consommation d'énergie</i>	18
II.6.4. <i>Modèle de détection</i>	19
II.7. CONCLUSION.....	20
Chapitre III. Déploiement dans les Réseaux de capteurs sans fil (RCSF)	21
III.1. INTRODUCTION.....	22
III.2. PROBLEMATIQUE DE PLACEMENT DES NŒUDS.....	22
III.3. DIFFERENTS PROCESSUS DU DEPLOIEMENT	23
III.3.1. <i>Déploiement initial</i>	23
III.3.2. <i>Post-Déploiement</i>	25
III.3.3. <i>Redéploiement</i>	25
III.4. CONTRAINTES DU DEPLOIEMENT DANS LES RESEAUX DE CAPTEURS	25
III.4.1. <i>Couverture et connectivité</i>	25
III.4.2. <i>Energie dans les réseaux de capteurs</i>	31
III.4.3. <i>Déploiement et Routage</i>	31
III.5. CONCLUSION.....	32

Chapitre IV. Détection et coopération dans les Réseaux de capteurs sans fil (RCSF)	33
IV.1. INTRODUCTION	34
IV.2. DETECTION ET ROUTAGE	34
<i>IV.2.1. Routage et contraintes</i>	36
<i>IV.2.2. La détection dans les RCSF</i>	36
IV.2.2.1. Définition	37
IV.2.2.2. Méthodes de détection-coopération	37
a) L'agrégation	38
b) La détection et l'agrégation	41
c) Dépendance dimensionnelle et temporelle	41
IV.2.2.3. Méthodes de consensus.....	42
a) Consensus de la moyenne.....	43
b) Consensus des hypothèses	45
IV.2.2.4. Algorithmes de conversations entre voisins	46
a) Conversation entre paire de voisins.....	46
b) Conversation par inondation partielle	47
c) Conversation par triplet	49
d) Conversation entre voisins-à-un-saut	49
<i>IV.2.3. Cadre de comparaison entre les méthodes de détection</i>	50
IV.3. CONCLUSION	50
Chapitre V. Contribution à la détection et coopération des capteurs	52
V.1. INTRODUCTION	53
V.2. FIL CONDUCTEUR DE NOTRE CONTRIBUTION	54
V.3. SIMULATEUR CONTIKI/COOJA	55
<i>V.3.1. Protocole de routage - RPL</i>	55
<i>V.3.2. Force du signal (RSSI)</i>	56
V.4. EXPERIMENTATION.....	57
<i>V.4.1. Partie I : L'effet du nombre de capteurs sur d'un déploiement déterministe</i>	57
V.4.1.1. Introduction	57
V.4.1.2. Simulation faite pour 50, 24, 17 capteurs	58
V.4.1.3. Interprétation des résultats pour 50, 25, 17 capteurs	62
V.4.1.4. Expérimentation : 12 capteurs (Déploiement-4).....	63
V.4.1.5. Interprétation des résultats (pour 12 capteurs).....	64
V.4.1.6. Conclusion.....	65
<i>V.4.2. Partie II - La détection et la coopération des capteurs – 1^{ère} approche –</i>	65
V.4.2.1. Méthode de détection standard basé sur le Protocol RPL.....	67
V.4.2.1.1. Introduction	67
V.4.2.1.2. Simulation	67
V.4.2.1.3. Interprétation des résultats	73
V.4.2.1.4. Conclusion.....	74
V.4.2.2. La méthode proposée « Détection et coopération de 3-voisins»	74
V.4.2.2.1. Introduction	74
V.4.2.2.2. Simulation	76
V.4.2.2.3. Interprétation des résultats	81
V.4.2.2.4. Conclusion.....	83
<i>V.4.3. Partie III - Coopération des capteurs par classification – 2^{ème} approche –</i>	83

V.4.3.2. Compréhension de l'approche.....	84
V.4.3.3. Approche proposée.....	85
V.4.3.3.1. Introduction.....	85
V.4.3.3.2. Phase 1 : Classification.....	88
V.4.3.3.3. Phase 2 : Transmission et mise à jour des données.....	91
V.4.3.3.4. Résultat de l'expérimentation.....	94
V.4.3.3.5. Analyse et Interprétation des résultats.....	98
V.4.3.3.6. Conclusion.....	101
Conclusion générale.....	102
Bibliographie.....	105



Table des figures

Figure 1. – Architecture physique d'un nœud-capteur	7
Figure 2. Architecture détaillée d'un nœud-capteur	9
Figure 3. Topologie du réseau Zigbee	11
Figure 4. Modèle de communication « Disque Binaire »	15
Figure 5. Déploiement déterministe / Déploiement aléatoire.....	22
Figure 6. Classification des techniques de déploiement des nœuds de capteurs	23
Figure 7. Arroseur Centrifuge à Canon (CCS)	24
Figure 8. Nœuds de capteurs largués par un hélicoptère.....	24
Figure. 9. Capteurs avant/après application de la VFA	26
Figure 10. Modèles de déploiement des capteurs en 3D	27
Figure 11. Découpage de la zone d'intérêt	28
Figure 12. Champs de vision de la KINECT	29
Figure 13. La propagation hiérarchique de l'horloge dans le réseau.....	30
Figure 13. Trafic « de bas en haut »	35
Figure 14. Taxonomie sur les protocoles de routages dans les RCSF	36
Figure 15. Défis de la détection d'événements dans les RCSF].....	37
Figure 16. Architecture des approches d'agrégation multicritères	39
Figure 17. Classification des approches d'agrégation multicritères	40
Figure 18. Consensus de la moyenne.....	45
Figure 19. Exemple de conversation entre deux voisins.....	47
Figure 20. Exemple de conversation par inondation partielle	49
Figure 21. Exemple de conversation entre Triple points	49
Figure 22. Plate-forme de nœuds de capteurs TMote Sky	53
Figure 23. Déploiement 1 - 50 capteurs -.....	57
Figure 24. Déploiement 1, Plan 1 : Alarme en zone 1	58
Figure 25. Déploiement 1, Plan 2 : Alarme en zone 2	58

Figure 26. Déploiement 1, Plan 3 : Alarme en zone 3	59
Figure 27. Déploiement 1, Plan 4 : Alarme en zone 4	59
Figure 28. Déploiement 1, Plan 5 : Alarme en zone 5	60
Figure 29. Déploiement 2 - 25 capteurs –	61
Figure 30. Déploiement_3 - 17 capteurs.....	61
Figure 31. Représentation graphique des trois déploiements.....	62
Figure 32. Déploiement 4 - 12 capteurs.....	63
Figure 33. Représentation graphique des quatre déploiements	64
Figure 34. Déploiement 4 - 12 capteurs, avec le chemin emprunté vers le puits.....	64
Figure 35. Plan d'une topologie en détection standard.....	68
Figure 36. Consommation d'énergie du nœud 15 en détection standard.....	70
Figure 37. Taux du cycle effectif de la radio du nœud 15 en détection standard.	71
Figure 38. Consommation d'énergie du nœud 6 en détection standard.....	71
Figure 39. Cycle effectif de la radio du nœud 6 en détection standard	72
Figure 40. Consommation d'énergie en détection standard du nœud 6.....	72
Figure 41. Taux de cycle effectif de la radio du nœud 7	73
Figure 42. Taux de cycle effectif de la comparaison radio pour les trois nœuds,.....	73
Figure 43. Exemple de sélection de valeurs à transmettre par le nœud n°1.....	75
Figure 44. Topologie utilisée dans la méthode de 3-voisins	76
Figure 38. Consommation d'énergie du mote 2	77
Figure 46. Pourcentage du nombre de ticks pour la mote 2	78
Figure 47. Consommation d'énergie du mote 3	78
Figure 48. Pourcentage du nombre de ticks pour le mote 3	79
Figure 49. Consommation d'énergie de mote 4 par la méthode des 3-voisins	80
Figure 50. Taux du cycle effectif du mote 4	80
Figure 51. Comparaison du taux de cycle effectif radio des nœuds 2, 3 et 4	81
Figure 52. Plan de comparaison des méthodes	82
Figure 53. Comparaison de la méthode proposée sur un nouveau plan	83



Figure 53. Aperçu de la structure du DataSet d'Intel (2004)	86
Figure 54. Schéma résumant les statistiques descriptives.....	87
Figure 55. Nombre de transmissions par « Intervalle de Confiance – C.I.» (Capteur de Température)90	
Figure 56. Limites d'utilisation d'un capteur et étendue de mesure	90
Figure 57. Structure d'un paquet du Laboratoire Intel modifiée.....	91
Figure 58. Exemple de paquets avec le caractère température	92
Figure 59. Interface du programme « Sensoring »	93
Figure 60 : Détail sur la lecture des tables	98
Figure 61. Comparaison des résultats du Mote 1 avec les 4 capteurs.....	98
Figure 62. Comparaison des résultats du Mote 4 avec les 4 capteurs.....	99
Figure 63. Comparaison des résultats du Mote 7 avec les 4 capteurs.....	99
Figure 64. Comparaison des résultats du Mote 9 avec les 4 capteurs.....	99
Figure 65. Comparaison des résultats du Mote 18 avec les 4 capteurs.....	100
Figure 66. Comparaison des résultats du Mote 20 avec les 4 capteurs.....	100
Figure 67. Energie consommée sur chaque mote avec application des 4 algorithmes	101

Liste des tableaux

Tableau 1. Caractéristiques des capteurs ZigBee.....	6
Tableau 2 : Composants des différents capteurs Zigbee	10
Tableau 3 : Synthèse de quelques méthodes utilisées dans le déploiement des capteurs.....	31
Tableau 4. Avantages et Inconvénients des méthodes de détections.....	50
Tableau 5. Paramètres de simulation	57
Tableau 6. Résultat obtenu du plan1	58
Tableau 7. Résultat obtenu du plan 2	58
Tableau 8. Résultat obtenu du plan 3	59
Tableau 9. Résultat obtenu du 1er déploiement dans la Zone 4	59
Tableau 10. Résultat obtenu du 1er déploiement dans la Zone 5	60
Tableau 11. Récapitulatif des résultats obtenus des 5 plans du déploiement 1	60
Tableau 12. Récapitulatif des résultats obtenus des 5 plans du déploiement 2	61
Tableau 13. Récapitulatif des résultats obtenus des 5 plans du déploiement-3	62
Tableau 14. Récapitulatif des résultats obtenus des plans du déploiement-4.....	63
Tableau 15. Nombre de « ticks » pour le nœud 15.....	69
Tableau 16. Pourcentage du cycle effectif de la radio du nœud 15	69
Tableau 17. Consommation d'énergie du nœud 15.....	70
Tableau 18. Nombre de ticks du mote 2	77
Tableau 19. Nombre de ticks du mote 4.....	80
Tableau 20. Estimation du temps de réponse et de la convergence du réseau	82
Tableau 21. Intervalle de confiance – Mote 1: capteur de « Température »	89
Tableau 22. Représentation binaire du contenu des champs.....	91
Tableau 23 : Extension des classes en utilisant l'amplitude $A=1.8874$	92
Tableau 24 : Résultats Obtenus après l'exécution du programme « Sensoring »pour les motes 1, 4, 7, 9, 18, 20.....	97
Tableau 24. Consommation d'énergie par mote	100

Chapitre I. Introduction générale

I.1. Contexte

La prévision permet à l'homme d'anticiper les actions à prendre face à des phénomènes redoutés. C'est pourquoi les chercheurs en technologie ont développé des capteurs de petite taille à un prix raisonnable, dotés d'unités de traitement et de modes de communication sans-fil, pouvant relever des données sur un environnement géographique tout en communiquant de manière préprogrammée et instantanée avec d'autres capteurs avoisinants formant ainsi un réseau de type Ad-Hoc, nommé réseau de capteurs sans fil – RCSF (Wireless Sensor Network - RCSF).

La technologie RCSF est en pleine expansion, surtout dans la surveillance de l'environnement à distance où l'intervention humaine n'est plus quasi-nécessaire. Nous soulignons que par l'avènement de ces réseaux sans fil, l'informatique est devenue beaucoup plus dynamique et prévoyante qu'elle n'a été auparavant c'est-à-dire « réactive » et conditionné. Dans un RCSF, Les nœuds se réorganisent entre eux afin d'acheminer des données acquises sur un phénomène physique au moyen des ondes de radiofréquence (RF). Ces données peuvent être des relevées de température, de la lumière, de l'humidité, des fuites de gaz, d'intrusion, ou même des mouvements (Ponts, séismes, glissement de terrains, etc.).

Plusieurs facteurs et contraintes agissent sur l'architecture des réseaux de capteurs :

- La tolérance : une altération de certains capteurs n'affecte pas le réseau en entier.
- Le champ d'observation : les capteurs sont livrés à eux-mêmes ceci implique qu'ils doivent fonctionner sans l'intervention humaine dans les régions hostiles et surtout à risques.
- La topologie de réseau : une maintenance est recommandée – déploiement, post-déploiement (mobilité ou capteurs éteint,...), redéploiement de capteurs additionnels.
- Les contraintes physiques : la taille du capteur, la consommation avisée d'énergie pour la survie du réseau, résistance : acclimatation avec l'environnement (température élevée, eau,..).
- Les médias de transmission : la communication multi-saut se fait souvent par des fréquences radio : ZigBee.

– La durée de vie : dépend totalement de la consommation d'énergie. Dans un réseau de capteurs, chaque nœud collecte des données et les envoie/transmet à d'autres capteurs jusqu'au puits qui à son tour les envoie à la station. Toutes ces tâches demandent de l'énergie. C'est pourquoi tous les premiers travaux des chercheurs se sont infléchis particulièrement sur la question de minimiser cette consommation d'énergie.

Les chercheurs n'ont pas cessé de multiplier leurs efforts pour trouver de multiples perspectives quant à l'intégration de ces capteurs dans le domaine domotique, militaires, santé, agriculture, etc. Ceci les a incité à développer ces microsystèmes avec précision, maniabilité, à faible coût et d'une simplicité de déploiement. Néanmoins, l'attention de ces concepteurs s'est attachée beaucoup plus à l'aspect fonctionnel ou non de ce réseau, comme par exemple le temps de réponse (détection-traitement-envoi des données), et surtout, à la fiabilité des données collectées : cas de relevés de paramètres de santé. Ces réflexions les ont surtout mené à optimiser la durée de vie des capteurs avec leurs applications tout en mettant en place de nouveaux protocoles de routage, et aussi améliorer leurs mécanismes d'alimentation avec des systèmes de mises en veille tout en contrôlant la couverture du réseau.

Nous entendons par déploiement la disposition ou l'étalement des capteurs dans un milieu de contrôle ou de surveillance réel et qui, plus est, doivent être aussi fonctionnels que possible. En réalité, cette peine aussi laborieuse qu'elle soit, souligne couramment des erreurs et fausse les résultats obtenus depuis les nœuds ou les puits de données. Assurément, la précision de l'étendue de mesure d'un capteur s'exprime essentiellement par des grandeurs d'influence. Par exemple, un capteur de force ayant un intervalle de mesure de 0 à 10000 livres $\approx 4535.92\text{kg}$ doit respecter sa marge de température qui est de 0 à $+55^\circ\text{C}$. Cela explicite clairement que la précision et le bon fonctionnement de ce capteur sont affirmés seulement dans cette plage de température. D'autres grandeurs d'influences sont responsables de ces imprécisions, comme : les défaillances dues aux accidents, la pression et la corrosion, etc. Notons que la simulation de ces grandeurs est quasi-impossible du fait qu'elles ne peuvent-être modélisées.

L'architecture d'un réseau de capteurs dépend clairement du déploiement de ses nœuds. Par conséquent, le mode de déploiement des nœuds influence l'efficacité du réseau RSCF. Ce dernier doit être capable d'affronter des obstacles dans un environnement hostile. Un bon déploiement peut, non seulement, diminuer le nombre de nœud et le coût du réseau, mais aussi perpétuer l'opérabilité du réseau afin d'assurer une couverture maximale et un

temps de réponse de qualité. Dans la pratique, on peut placer intentionnellement les nœuds et les nœud-puits ; on parle alors de déploiement déterministe. Ce déploiement s'aperçoit surtout dans les applications de contrôle, les applications de surveillance vidéo et les applications relatives à la santé humaine, où les relevés doivent être de qualité et d'une précision accrue. En revanche le déploiement non déterministe ou souvent dénommé « aléatoire », où le nombre de nœuds est important et donc on n'a plus une pré-représentation des positions de chaque nœud, peut être préconisé dans les environnements hostiles.

I.2. Problématique

Les réseaux de capteurs soulèvent de nouveaux challenges réseaux tels que le routage, la gestion énergétique, l'auto-configuration, la dissémination et la collecte des données ou encore le déploiement, etc. Cependant, la majorité des études menées se sont focalisées essentiellement sur des problématiques réseaux liées au routage et/ou à la gestion de l'énergie.

Une très faible attention a été portée au déploiement. Malheureusement, un mauvais déploiement peut induire une mauvaise exploitation du réseau, telle qu'une consommation abusive de l'énergie ou encore à une mauvaise connectivité entre les capteurs. Ceci donna une motivation d'explorer cette problématique et à proposer de nouvelles recommandations pour le déploiement des réseaux de capteurs sans fil statiques. L'objectif est de faire une étude sur la détection et la coopération pour en tirer des conditions aidant à mieux positionner des capteurs dans une zone de déploiement donnée, tout en garantissant certaines contraintes liées à ce type de réseaux, telles que le coût de déploiement ainsi que la durée de vie du réseau.

Dans une première partie nous nous proposons d'étudier comment un tel modèle de détection collaborative peut être pris en considération pour minimiser le nombre de capteurs.

Un deuxième objectif de cette thèse est de proposer un modèle de dissémination optimale de l'information entre les capteurs et leurs voisins directs ou indirects dans le but de réduire l'impact de la coopération sur la durée de vie du réseau.

I.3. Contribution de la thèse

L'objectif du présent travail est d'exposer différentes réalisations pratiques et théoriques « de base » à la détection et à la coopération des nœuds de capteurs dans le but de contribuer à aider le bon déploiement d'un réseau de capteurs sans fil, et encourager les différents chercheurs de parts leurs disciplines, à envelopper leurs travaux autour de cette

problématique. Pour approcher cette compréhension d'optimisation, nous avons divisé notre recherche en trois parties. La première partie explicite l'impact du nombre de nœuds de capteurs dans une zone à surveiller. La seconde partie étudie l'effet de la détection et la coopération des nœuds de capteurs sur le nombre de capteurs déployés. La troisième partie propose une méthode originale de collecte sur des données réelles issues du laboratoire d'Intel Berkley qui permettra au réseau non seulement de minimiser la consommation d'énergie, mais aussi d'être tolérant aux fautes.

I.4. Organisation de la thèse

Cette thèse est composée de quatre chapitres. Après une introduction générale, on expose dans le premier chapitre le domaine de recherche de notre sujet de travail, i.e., l'architecture des réseaux de capteurs sans fil (RCSF), où on expose trois traits principaux : la description physique des nœuds, la notion de topologie et les modèles dans un réseau.

Le deuxième chapitre est consacré entièrement à la notion du déploiement des nœuds dans un réseau. L'accent est mis surtout sur la problématique de placement, et par la suite les différents processus de placement des nœuds. Le chapitre est clôturé par l'exposition de différentes contraintes de déploiement des nœuds.

Le troisième chapitre présente un état de l'art sur la détection et la coopération des nœuds de capteurs. Cette partie représente une assiette prépondérante pour notre travail, car l'exploration des différentes recherches nous a permis de tirer le model de notre approche. La dernière section relate un cadre de comparaison des différents systèmes existants, et elle est achevée par les différentes méthodes de détections étudiées.

Le quatrième chapitre est consacré à la présentation de l'approche choisie pour donner les concepts de base et les recommandations nécessaire afin de réussir un bon déploiement des nœuds et de faire perdurer un réseau de capteurs. Le chapitre est divisé en trois parties, la première partie est réalisée sur le simulateur Cooja/Contiki afin d'observer l'effet du nombre de capteurs et leur placement sur la perpétuité du réseau. La seconde partie introduit un algorithme de consensus pour régler les problèmes rencontrés auparavant ; une comparaison est faite entre l'approche présentée et la simulation standard. La troisième partie, donne un cadre générale de détection et de coopération en travaillant sur un échantillon réel, lui appliquant un deuxième algorithme basé sur les statistiques descriptives pour relayer de l'information jusqu'au puits. Nous terminons notre mémoire par une conclusion générale, dans laquelle on expose le jugement final de notre approche et de son efficacité.

Chapitre II
Architecture
des réseaux de capteurs sans fil(RCSF)

II.1. Introduction

Un réseau capteur possède une topologie dynamique. Un nombre important de capteurs sont parsemés dans une région et communiquent entre eux à travers une fréquence radio. Les capteurs déployés sont reliés à plusieurs ou, à défaut, à un seul puits qui à leurs tour sont connectés à une station de base ou à un satellite afin de transmettre les données mesurées à l'utilisateur (Figure 1.).

Plusieurs technologies radio sont utilisées dans ce sens utilisant l'un des protocoles : « Zigbee », « Z-wave », « UWB : Ultra-WideBand Technology », etc.

Accepter une technologie découle souvent des services qu'elle peut offrir aux réalisateurs du réseau. Certains paramètres doivent être pris en compte (voir Tableau n°1).

Néanmoins le protocole de haut niveau Zigbee appartient à une alliance de plus de 175 sociétés qui le développe vers un standard ouvert. Il est donc le plus utilisé, car il s'adresse plutôt aux applications qui émettent des messages courts et fiables dans un environnement bruité des radios fréquences, en formant un système robuste à séquences directes dans des réseaux en étoile, pairs à pairs ou maillés. Zigbee appartient à la norme IEEE 802.15.4 des réseaux Ad-hoc – nécessitant peu d'équipement dans un espace de 10m. (Wireless Personal Area Networks : WPAN).

Dans toute la suite du document, nous parlerons de nœuds de capteurs sans fil disposant du protocole Zigbee. Le tableau 1 ci-après présente les principales caractéristiques de ZigBee :

Propriété	Description
– Application	– Vigilance et contrôle
– Ressources systèmes (mémoire)	– 4Ko à 32Ko
– Longévité de la batterie (jours)	– > 1000
– Taille du réseau	– "illimité" en théorie, 32000 en réalité
– Bande passante théorique (KB/s)	– 20 à 250 KB/s
– Portée (mètres)	– 100 m
– Consommation en émission	– 25 à 35 mA
– Consommation en veille	– 3 μ A

Tableau 1. Caractéristiques des capteurs ZigBee

II.2. Nœud : Description physique

Un nœud est un appareil qui mesure des grandeurs physiques et qui, après les avoir convertis en valeurs numériques, les traite et les envoie via un réseau à une station de base.

On compte quatre unités relevant de la constitution d'un nœud de capteur, assurant une fonction bien précise :

- Module d'acquisition,
- Module de traitement,
- Module de transmission
- Module de source d'énergie

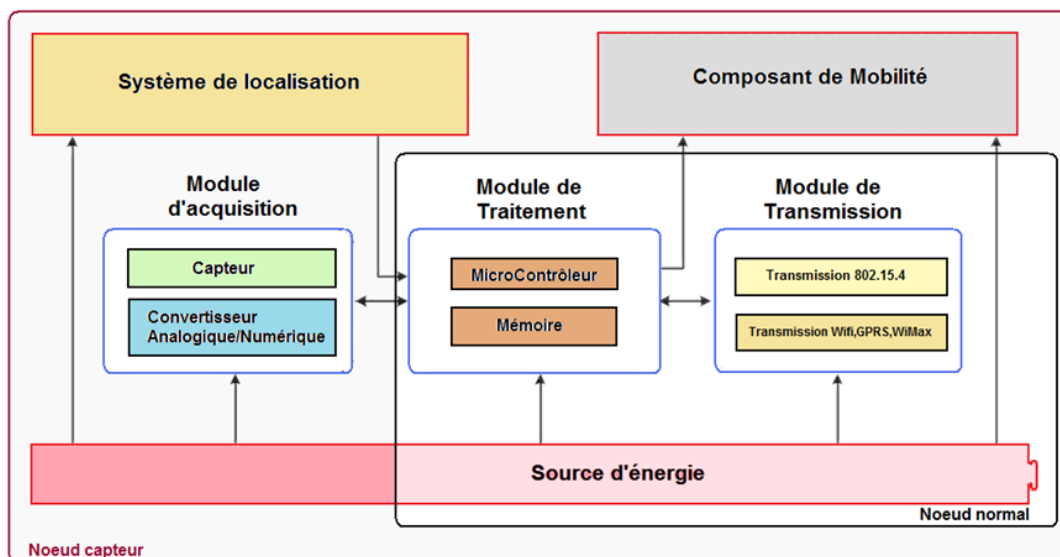


Figure 1. – Architecture physique d'un nœud-capteur.

a) Module d'acquisition

Ce module dispose d'un ou plusieurs capteur qui relève des grandeurs physiques de la zone à surveiller et possède un CAN (Convertisseur Analogique/Numérique) qui convertit la mesure relevée et la communique par la suite au module de traitement.

b) Module de traitement

Ce module est composé d'un microcontrôleur, d'une mémoire de capacité moyenne et d'un système embarqué propre aux nœuds. Le module de traitement reçoit les données en provenance du module d'acquisition et les communique au module de transmission.

c) Module de transmission

Ce module de communication assure toute émission/réception de données à travers un canal de diffusion radio fréquence (nœuds Stargate) ou optique (Smart Dust).

d) Module énergétique

Un capteur est alimenté par une batterie qui répartie son énergie à tous les autres modules (Voir Figure 1). Une contrainte majeure pour une multitude de capteurs est que cette batterie n'est ni remplaçable ni rechargeable, ce qui la rend une ressource critique, et par conséquent on doit l'utiliser avec parcimonie pour prolonger la survie du capteur.

e) Modules complémentaires

Nous avons la possibilité de voir d'autres modules qui viennent se coller aux modules existants, par exemple le GPS (Global Positioning System), un module de mobilité permettant le déplacement de nœud-capteur, une autre source d'énergie, un module de communication Rx/Tx (WIFI, autres), etc.

II.3. Modèle de Nœuds

Afin d'aller plus loin dans une architecture RCSF et la dimensionner par la suite, faisons une lecture transversale sur les modèles de nœuds existants. Un réseau de capteur est créé et structuré suivant le but de son application. On observe distinctement certains types de nœuds :

- Nœud simple ou standard : embarque un module de transmission et de traitement.
- Nœud-capteur : Nœud simple additionnant un module d'acquisition ou de détection.
- Un nœud actionneur : un nœud normal doté d'un module de contrôle exécutant certaines actions mécaniques (se déplacer, ...).
- Nœud puits : nœud simple additionnant un module de conversion de signal analogique-numérique relié par la suite à un autre module de communication.
- Nœud passerelle : nœud simple ayant la possibilité de transmettre les données dans le même réseau à l'aide d'une communication 802.15.4.

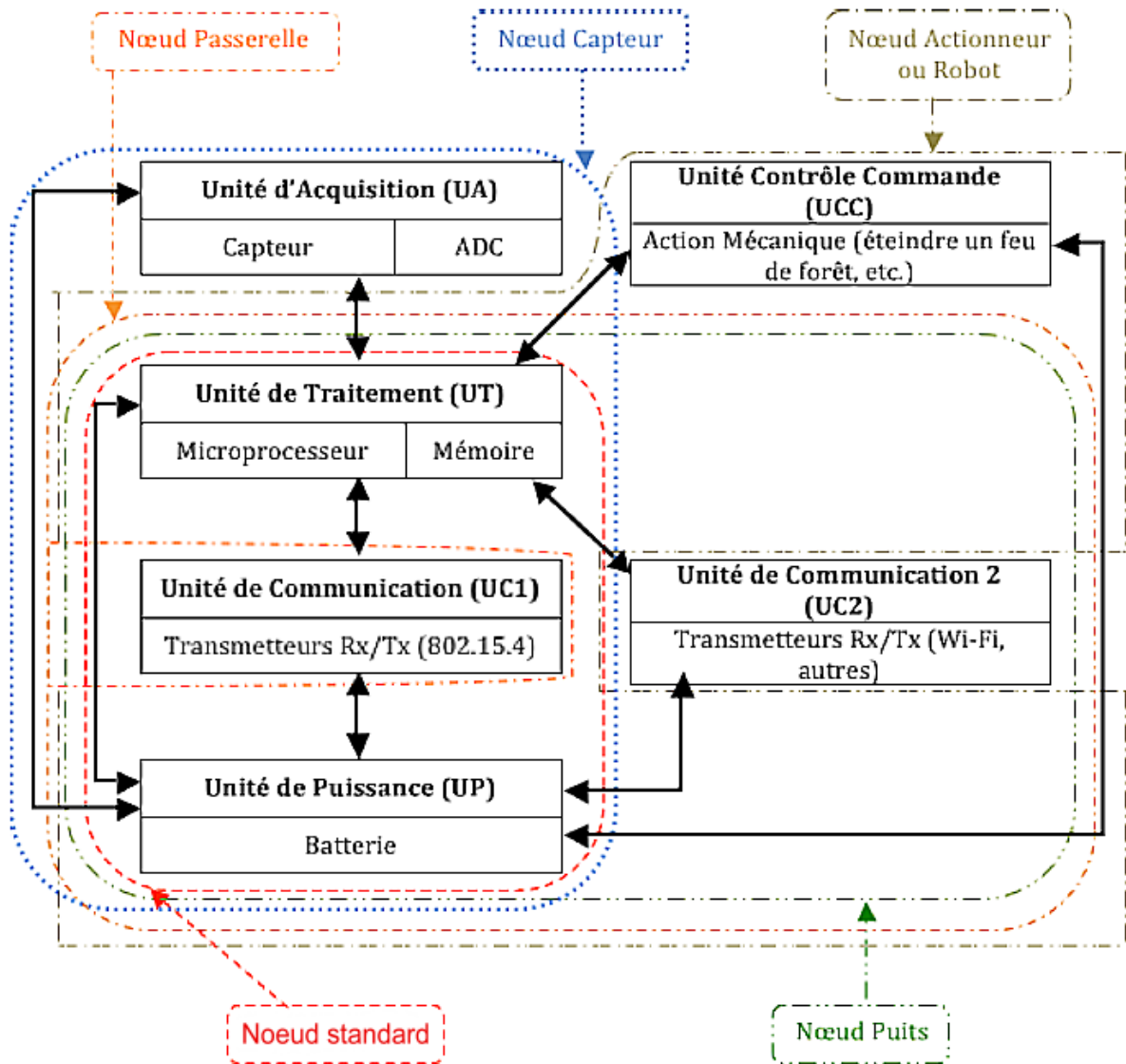


Figure 2. Architecture détaillée d'un nœud-capteur [100]

II.4. Rôle des nœuds de capteurs

Beaucoup de travaux ont défini les rôles joués par ces nœuds dans des différents types d'architectures : plat, hiérarchique ou multi-niveaux. Il existe alors trois sortes de rôles données aux nœuds dans un réseau.

1. **Nœud Coordinateur** (NC) : souvent appelé nœud collecteur. Il existe un seul pour chaque sous réseau (cluster) de capteur Zigbee. C'est lui qui initialise la constitution réseau et sert aussi de passerelle vers d'autres réseaux éventuels. Ce nœud agrège les données dans une architecture hiérarchique et il est appelé « Cluster-Head ».
2. **Nœud routeur** (NR) : il participe au routage des données qu'ils reçoivent, agrègent, relayent et donc augmentent la portée du signal. On le confond souvent avec le NC dans une architecture plate.

3. *Nœud Terminal (End Device) (NED)* : c'est un nœud simple. Il ne permet pas le routage ou d'association de nœuds. Il capte puis transmet ses données au nœud coordinateur en un saut.

Capteur		
NC	NR	NED
<ul style="list-style-type: none"> - 01 / réseau - Lance le réseau. - Se comporte en routeur - Contient tous les modules. 	<ul style="list-style-type: none"> - Optionnel. - Coordonne avec le ZC ou un autre ZR. - Joue le rôle d'un coordinateur. - Contient tous les modules. 	<ul style="list-style-type: none"> - Optionnel - Pas d'associations avec ZC/ZR. - Pas de routage - Minimum en module.

Tableau 2 : Composants des différents capteurs Zigbee¹

Pour beaucoup plus de détail, [1] fournit une spécification complète du standard IEEE 802.15.4 et dans [2] il y a une interprétation résumée.

II.5. Description des topologies des RCSF

Une topologie étale la manière dont les nœuds sont reliés géométriquement dans un réseau lui donnant ainsi une structure d'un graphe.

Dans ce sens, la norme IEEE 802.15.4 met en évidence les topologies réseau en étoile, arbres et maillé. Il est à noter que ce dernier repose sur le protocole IEEE 802.15.4, mais il peut fonctionner aussi par lui-même, et l'on peut développer des applications à ce niveau. Le protocole ZigBee gère les couches hautes, tandis que le protocole IEEE-802.15.4 gère les niveaux physique et logique (MAC).

La structure d'un réseau de capteurs sans fil (RCSF), dans sa forme standard, est composée d'un groupe de nœuds associé à une passerelle et une station de base.

La topologie physique des RCSF dépend beaucoup plus du déploiement aléatoire ou non aléatoire (déterministe) des capteurs. Le choix de l'un ou de l'autre modèle est réalisé en proportion de la complexité de ce déploiement, de la taille du réseau et de la surface de déploiement souvent hostile. Le déploiement physique cherche à déterminer souvent la couverture du réseau. Quant à la topologie logique, elle détermine souvent les qualités et performances du réseau. L'association de l'alliance ZigBee et le protocole 802.15.4. a donné la naissance à des réseaux Mesh suite au routage dynamique.

¹ <https://wapiti.telecom-lille.fr/commun/ens/peda/options/st/rio/pub/exposes/exposesrio2006/descharles-waia/network.htm>

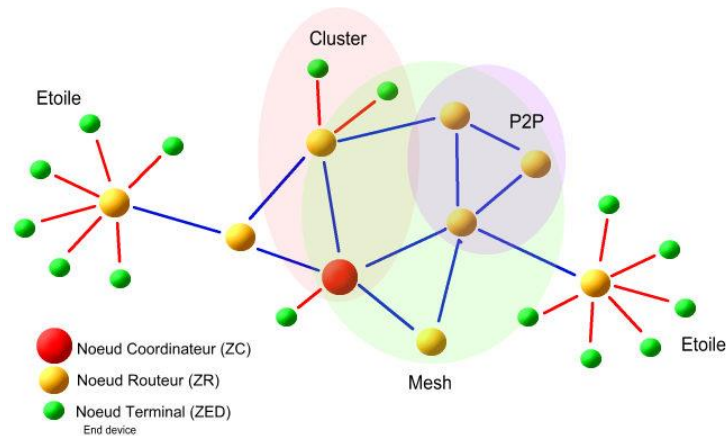


Figure 3. Topologie du réseau Zigbee¹

II.5.1. Topologie peer to peer

Dans les réseaux « peer to peer » (Figure 3), chaque nœud-capteur communique avec son voisin. Dans ce cas chaque nœud est autonome, c'est-à-dire, il est capable de fonctionner à la fois comme un ZC et ZR.

II.5.2. Topologie en étoile

La topologie étoile (Voir Figure 3), ressemble clairement à la topologie déjà étudiée dans les réseaux ad-hoc, où tous les messages passent par un commutateur – ici, il est question du nœud coordinateur ZC ou ZR qui garantira le rôle de relais des différents nœuds terminaux (ZED). Cette topologie reste assez simple du point de vue réalisation, mais son point faible est que le nœud collecteur doit faire toute la gestion, ce qui impliquera une certaine puissance de ce dernier.

II.5.3. Topologie en maille (mesh)

Le principe de la topologie maillée (Voir Figure 3) est de relier tous les nœuds entre eux (ou du moins, un maximum). Comme ça, le risque de panne générale du réseau est minimisé. Un certain nombre de nœuds routeurs (ZR) vont jouer le même rôle que le nœud coordinateur et assurent le relais d'informations, soit en un seul ou plusieurs sauts jusqu'à la destination. La communication multi-saut entre les capteurs est biaisée par la topologie du réseau utilisé, la distance entre les nœuds et la qualité du signal avoisinant.

La topologie maillée est bien adaptée aux applications dépendant du temps. Par exemple, si un nouveau capteur est ajouté au réseau, il lui faut 30ms pour qu'il fera partie du réseau, et moins 15ms pour passer de la mise en veille à l'état actif.

¹ <https://wapiti.telecom-lille.fr/commun/ens/peda/options/st/rio/pub/exposes/exposesrio2006/descharles-waia/network.htm>

Quant à l'apport de cette topologie, elle est soulignée par l'utilisation d'un grand nombre de nœuds jusqu'à 65536 capteurs, mais elle dépend du facteur d'énergie et de la durée de convergence (latence) perdue dans les liaisons multi-saut.

II.5.4. Topologie hiérarchique

Cette topologie nous fait beaucoup penser à un arbre (voir Figure 3), où les nœuds routeurs ZR communiquent leurs données à un nœud collecteur suivant un routage dynamique arborescent. Cette communication s'appuie sur un envoi d'une balise, souvent citée, « Beacons » – envoi périodique d'une balise afin de synchroniser les dispositifs avec le coordinateur ZC. Ce type de topologie permet d'étendre le réseau de plus 64000 nœuds au total !

Afin de fonctionner efficacement, cette topologie nécessite une synchronisation avec le réveil du coordinateur (ou d'un routeur), car il garantit le routage des données à travers le réseau.

II.5.5. Eléments clés d'un réseau de capteurs

Dans un réseau à base de capteurs, les grandeurs mesurées sur un phénomène sont relayés au puits moyennant un coût et un déploiement optimal. Ainsi, la structure du réseau peut-être observée comme suit [4]:

1. **Infrastructure**

L'une des composantes essentielles du réseau. L'infrastructure est une illustration abstraite d'un réseau qui fait intervenir, les capteurs et de leur placement actuel. Elle est guidée par les caractéristiques des capteurs (précision de la détection, taille de la mémoire, longévité de la batterie, et la plage de transmission) et la stratégie de déploiement (densité du capteur, localisation des capteurs, et mobilité des capteurs).

2. **Protocole réseau**

Le protocole assure la disponibilité des routes et le maintien d'un dialogue entre les capteurs et la station de base.

3. **Application / Observateur**

Les intérêts des observateurs sont, beaucoup plus, les requêtes que posent les observateurs sur le phénomène tel qu'il est estimé par les données distribuées que les capteurs sont capables de détecter.

Ces requêtes peuvent être statiques (nœuds préprogrammés) ou dynamiques. Le réseau peut participer à la synthèse de la requête (par exemple, en filtrant certaines données de capteur ou en fusionnant plusieurs mesures en une seule valeur); nous considérons

qu'une telle intelligence fait partie du processus d'une translation entre les intérêts des observateurs et un bas niveau d'implémentation.

II.6. Structure et Modèles dans un réseau de capteurs

La classification d'un réseau de capteurs repose sur leurs caractéristiques d'utilisation dans les différents domaines [5]. Effectivement, on les particularise par différents modèles.

II.6.1. Modèle de communication du RCSF

Un réseau efficace [6] doit rendre compte de sa précision et de son temps de réponse en surpassant les contraintes agissant sur son architecture, comme l'utilisation judicieuse de l'énergie, etc. Bien choisir son modèle de communication c'est bien choisir aussi son protocole de communication.

II.6.1.1. Protocole de communication

Ce dernier prend la communication d'application qui se charge du transfert des données relevées vers la station. Nous soulignons ici deux modèles : coopératif et non coopératif.

a) Modèle coopératif

Les nœuds coopèrent servent à traduire l'information à l'utilisateur. La topologie hiérarchique en est un exemple idéal de coopération de capteurs. Un exemple de communication réseau est donné dans [7] où les auteurs implémentent un protocole appelé SPIN (Sensor Protocols for Information via Négociation), qui est centré sur les données et basé sur la négociation est ajusté aux réseaux de capteurs sans fil. Heidemann et al. [8] affirment que la communication de bas niveau est basée sur des attributs externes à la topologie du réseau et pertinents pour une application. Ils montrent que des approches telles que l'agrégation en réseau et les requêtes imbriquées peuvent affecter de manière significative le trafic dans un réseau. Ainsi, J. Heidemann étale les résultats de son expérience et annonce que l'agrégation réduit le trafic jusqu'à 42% et les requêtes imbriquées réduisent les taux de perte de 30%.0

Dans sa thèse, J. Heidemann [9] mène une première étude et développe LEACH (Low-Energy Adaptive Clustering Hierarchy), une architecture pour les réseaux de capteurs distants qui combine les idées de conservation efficace d'énergie, le routage en cluster et l'accès aux médias avec l'agrégation des données spécifiques à l'application pour obtenir une bonne performance du réseau.

b) Modèle non coopératif

Dans ce type de modèle, les capteurs non coopératifs ne coopèrent pas à la diffusion de l'information. Ils agissent par eux-mêmes et déclenchent l'opération à suivre en cas de dépassement du seuil de capture. La détection de présence, pour allumer une lampe d'une chambre, en est un exemple.

II.6.1.2. Communication d'infrastructure

Toute la littérature énonce que parmi les facteurs et les contraintes d'une conception des RCSFs, sont l'absence d'infrastructure initiale et aussi de tout genre d'administration centralisée.

Dans cette section, la communication d'infrastructure fait référence à la communication nécessaire pour configurer, maintenir et optimiser le fonctionnement.

Plus précisément, les capteurs doivent être capables de trouver les routes optimales à travers d'autres capteurs voisins afin de livrer les informations à la station, bien sûr indépendamment de la mobilité ou de la défaillance du capteur. En résumé, la communication d'infrastructure représente la fonctionnalité du réseau, et peut être considérée comme une surcharge du protocole. Il est donc impératif d'optimiser cette communication tout en assurant une communication efficace d'application.

Dans les réseaux de capteurs, une phase initiale de communication d'infrastructure est nécessaire pour mettre en place le réseau. De plus, si les capteurs sont hors circuit, il y aura une communication supplémentaire pour la reconfiguration. Aussi, si les capteurs sont mobiles ou si on s'intéresse à la dynamique du réseau, une communication complémentaire est fondamentale pour la découverte ou la reconfiguration du chemin. Dans une topologie hiérarchique, le protocole de clustering puise sa robustesse de la communication d'infrastructure pour constituer les clusters et nommer ses têtes de nœuds, de plus si les nœuds du réseau sont mobiles, cette communication doit être renouvelée. Dans [10], les auteurs réalisent un montage de surveillance d'habitat et présentent un système nommé « Frisbee », faisant intervenir des capteurs hétérogènes pour assurer la couverture des zones à surveiller. Une fois un événement anormal est alerté, des nœuds *engagés* « *sentries* » déclenchent le réveil des autres nœuds en veille. Ainsi, seule la région de l'événement est active.

Enfin, la liaison sans fil dépend surtout du canal de communication. Dans un RCSF, Le modèle « Disque Binaire » est le modèle déterministe le plus usité (voir Figure 4).

Un nœud « x » relaye ses valeurs aux nœuds « y » qui sont situées dans son cercle de rayon de capture noté « r_x ».

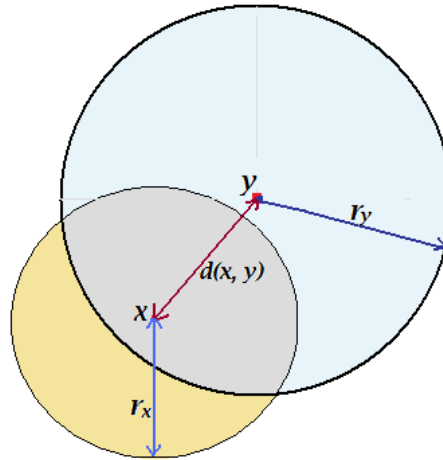


Figure 4. Modèle de communication « Disque Binaire »

Il est clair que la puissance du nœud restreint le rayon de diffusion. Donc, deux capteurs « x » et « y » sont constamment en communication s'ils vérifient l'équation suivante : $d(x, y) \leq \min \{r_x, r_y\}$.

Ajoutons à ce rayon l'affaiblissement de propagation¹ connu par : « **Path-loss** ». On définit aussi la zone d'ombre² comme étant un phénomène d'atténuation. Les zones d'ombre résultent de la fréquence utilisée et de la structure du site. La zone d'ombre est alors le champ où la détection n'est pas assurée.

Une formulation de l'affaiblissement de propagation « Path-loss » et celle de la zone d'ombrage sont données par les équations suivantes :

$$\frac{P_r}{P_t} = K \cdot \left[\frac{d_0}{d} \right]^n \cdot \psi \quad (1)$$

Ou en dBm (décibel),

$$P_{r_{dBm}} = P_{t_{dBm}} + K_{dBm} + 10 \cdot n \cdot \log_{10} \left[\frac{d}{d_0} \right] + \psi_{dBm} \quad (2)$$

$P_{r_{dBm}}$: Puissance du signal en réception

$P_{t_{dBm}}$: Puissance du signal transmis.

$\psi_{dBm} \sim N(0, n)$ (ψ_{dBm} suit une loi Normale $[0, \sigma]$ précisant la zone d'ombre)

d : La distance entre les nœuds émetteur-récepteur

n : Indice d'affaiblissement de propagation, variant selon la géométrie des lieux compris entre 2 et 6.

d_0 : Une distance de référence (entre 1 m ou 1 km)

K : paramètre d'affaiblissement par rapport à la distance de départ.

¹ https://fr.wikipedia.org/wiki/Affaiblissement_de_propagation

² [https://fr.wikipedia.org/wiki/Zone_d'ombre_\(t%C3%A9communications\)](https://fr.wikipedia.org/wiki/Zone_d'ombre_(t%C3%A9communications))

Les auteurs de [11] et [12] énoncent que la perte de puissance est une fonction déterministe $PL_0(d)$ et est de la forme :

$$PL_0(d) = PL(d_0) + 10 \cdot n \cdot \log_{10} \left[\frac{d}{d_0} \right] \quad (3)$$

Avec :

d : Distance entre l'émetteur et le récepteur,

n : Coefficient d'affaiblissement de propagation. Sa valeur est comprise [1.7, 5] : $n = 2$ en espace libre. [13]

d_0 : Distance initiale (1 mètre - 1 kilomètre)

$PL(d_0)$: Atténuation dû à l'affaiblissement à d_0 .

Pour les environnements les plus complexes, l'indice « n » peut être calculé suivant la hauteur de l'antenne d'émission ou de réception et la fréquence [13].

La puissance du signal reçu ($P_{r_{dBm}}$), est donnée par l'équation suivante :

$$P_{r_{dBm}} = P_{t_{dBm}} - PL_0(d). \quad (4)$$

Plusieurs modèle faisant apparaître certaines contraintes responsables de l'atténuation du signal dues aux problèmes de la structure de l'environnement d'un côté et de la dynamique des capteurs d'un autre côté lors de leurs déploiements. Citons, en premier, le modèle "Signal-to-Noise Ratio (SNR)" qui est décrit dans [12]. Il comptabilise les bruits de fond sur le signal. Ce bruit est généré par le changement de température sur les antennes des capteurs émetteurs et récepteurs. Notons que si cette température est constante, ce bruit est :

$$P_n = (F + 1)kT_0B \quad (5)$$

B : Bandwidth Bruit.

k : Valeur constante de Boltzmann

T_0 : T° environnante

F : Facteur de bruit

Dans le modèle de [14], les auteurs établissent l'équation (6) du bruit de fond à une distance « d » pour une puissance $P_{t_{dBm}}$, émis par un émetteur :

$$S_{NR}(d) = P_{t_{dBm}} - PL_0(d) - P_n \quad (6)$$

$PL_0(d)$: Identique au modèle "Path-loss" défini dans l'équation (3).

Un autre modèle « SINR » (Signal to Noise and Interference Ratio" prenant la qualité des liens en amont en s'appuyant sur les effets de bruits sur le signal reçu par le récepteur. Ce modèle est décrit dans [15] :

$$\frac{P_r(n_i)}{N + \sum_{n_k \in \eta \setminus n_i} P_r(n_k)} \geq SINR_\theta \quad (7)$$

n : Ensemble des nœuds.

$P_r(n_i)$: Force du signal reçu par n_j de n_i

N : Charge du bruit.

II.6.2. Modèle de collecte de données

Trois points sont à discuter selon l'interaction du réseau de capteurs et la station de base:

- Veille uniforme : ce type de modèle est le plus répandu, du fait que les capteurs envoient les mesures captées à des fréquences uniformes et régulières à la station de base, par exemple le suivi de la température ou d'humidité du sol dans une serre à plantes [16].
- Veille à la requête : ce type de modèle exige que les capteurs ne transmettent leurs mesures que s'ils sont sollicités par la station de base. Les demandes sont administrées à un nœud de capteur ou un cluster, ou bien parsemées à tous les nœuds comme une requête conditionnelle, par exemple à tous les capteurs vérifiant une certaine condition.
- Veille sur un événement : ce type de modèle est un cas typique d'intrusion où les capteurs réagissent seulement quand à l'observation d'un événement anormale.
- Modèle hybrides : une association de deux, ou plus, des modèles antérieures.

Deux approches pour construire et maintenir un chemin pris par les nœuds en destination de la station de base. Ceux-ci différeront en fonction de la dynamique du réseau statique ou mobile. D'autres sources entrent dans la dynamique, par exemple la panne d'un capteur ou le changement d'intérêt de l'expert.

a) Réseaux de capteurs statiques

Dans un tel réseau, il n'y a pas de mouvement entre les capteurs, l'assistant et le phénomène. Plusieurs études dans ce sens ont fait l'objet de recherche et ont prouvé que les algorithmes localisés donnent de meilleurs résultats [17].

b) Réseaux de capteurs dynamiques

Dans ce genre de réseau, les capteurs eux-mêmes, l'observateur ou le phénomène sont mobiles. Chaque fois que l'un de ces trois acteurs change de position, le chemin initial peut échouer. Dans ce cas, l'observateur ou le capteur concerné doit prendre en charge la reconstruction de la nouvelle route.

i) Approche réactive

Cette approche est initiée par l'observateur, où l'action de rétablissement du chemin est seulement prise après avoir observé un chemin échoué. Effectivement, dans la première phase de déploiement de capteurs, l'expert peut mettre en place plusieurs chemins entre lui et le phénomène observé, mais il pourra les cacher et préserver seulement le chemin le plus optimal à ce moment-là comme le chemin actuel. Toutefois, si le chemin optimal échoue, un autre des chemins mis en cache peut être utilisé. Si tous les chemins mis en cache ne sont pas valides, l'observateur doit reconstruire de nouveaux chemins.

ii) Approche proactive

Cette approche est initiée par le capteur, où l'opération commence par la prévision de chemin en voie de rupture. Dans cette procédure, le capteur, faisant partie du chemin actuel et demandant de se mettre hors portée du chemin actuel, diffuse une demande de participation à la reconstruction d'un nouveau chemin à tous ses capteurs voisins. L'un de ses derniers peut accuser réception par un message positif de participation au capteur initiateur, et alors une procédure de correction locale est entreprise pour créer le nouveau chemin. Dans le cas où aucun des capteurs voisins n'a accusé réception, le capteur initiateur remet la main à l'observateur afin de reconstruire lui-même le chemin. Un cas similaire est cité dans les réseaux IP mobiles pour qu'un nœud IPv4 ou IPv6 peut se déplacer de manière transparente [18].

II.6.3. Modèle de consommation d'énergie

Ce modèle a fait l'objet de plusieurs études [18]. Ainsi, la conclusion parvenue est que la communication de données (transmission, réception) est la plus gourmande en énergie par rapport au traitement et à la veille des capteurs :

a) Energie de capture

On sous-entend par capture, l'énergie nécessaire à l'initialisation du capteur et l'opération de numérisation : l'échantillonnage, la quantification et le codage du signal capté. Néanmoins, cette énergie consommée reste infime à l'énergie globalement utilisée.

b) Energie de traitement

Cette énergie se résume aux états de situations du capteur. Soit que le capteur est en écoute ou en commutation de message. Dans ce cas aussi, l'énergie fournie au traitement reste inférieure à celle de la communication.

c) Energie de communication

Elle dépend de la quantité d'information à transmettre et la taille du réseau. Elle regroupe l'énergie due à l'émission et l'énergie nécessaire à la réception.

Il est bon à noter que plus la puissance d'émission est élevée plus la portée du signal sera élevée, et par conséquent l'énergie consommée est grande. Dans ce cas, l'énergie nécessaire à la communication est la plus importante des énergies épuisées. Dans cette thématique, [19] énonce un modèle radio de consommation d'énergie.

II.6.4. Modèle de détection

La détection est le fait d'observer une modification de l'état normale de l'environnement surveillé, et avertir en un temps optimal la base afin de prendre les mesures nécessaires. Les auteurs de [20] rejettent les approches centrées sur les applications et soulignent l'importance de l'agrégation en tant qu'outil de réduction de données et mettent en avant leur agrégation guidée par une interface de style SQL à usage général qui peut exécuter des requêtes en offrant des opportunités d'optimisation significative.

Notons que la détection ne peut être efficace que si quelques paramètres de performances soient dégagés [21]:

- Durée de vie du système
- Taux de fausses alertes
- Estimation probabilistique de détection.
- Classification de l'évènement

- Etapes de la détection d'évènements

La détection d'évènements est constituée des quatre étapes suivantes :

1. Evènement basé sur des tâches :

- Soit que la détection est commandée par une requête provenant de la station en visant des captures bien définies à nœuds élus.
- Evènement préprogrammées dans les nœuds.

2. Evènement basé sur la nature de la capture : la détection d'intrusion en est un exemple.

3. Evènement basé sur l'agrégation : agréger des captures pour en vérifier la fiabilité.

4. Collecte des données par la station de base.

II.7. Conclusion

Il est à noter que la détection d'événements est motivée par la collecte de données la plus simple possible. L'information n'est relayée qu'après cette opération. De ce fait, le traitement assez complexe n'a pas sa place dans le réseau, car l'événement est contraint par le temps de remise de cette capture au sink.

Chapitre III
Déploiement
dans les Réseaux de capteurs sans fil (RCSF)

III.1. Introduction

De par leur importance dans la surveillance, la détection, et la localisation des coordonnées d'un phénomène bien déterminé, le rendement optimal des réseaux de capteurs dépend d'abord de leur bon déploiement assurant bien sûr une bonne couverture. Ce problème de déploiement optimal des nœuds est un problème qui a été prouvé NP-difficile pour la plupart des formulations de déploiement des nœuds [22]. Dans ce chapitre, nous allons nous focaliser sur les méthodes de déploiement d'un minimum de capteurs qui optimisent la couverture.

III.2. Problématique de placement des nœuds.

Nous noterons trois axes auxquels le déploiement peut être mis en évidence : le premier dans les réseaux à topologie invariable appelé souvent statique, le second où les nœuds sont dynamiques et le troisième axe pointe sur les réseaux hybrides (le cas où les capteurs sont fixes et le collecteur sont mobiles, ou bien l'inverse). Quant aux méthodes de déploiement, il existe trois familles de placement de nœuds : nous classons dans la première toute planification, de placement de nœuds, préprogrammée et aléatoire ; la deuxième famille regroupe tout placement par domaine ; la troisième famille regroupe tout déploiement basé sur une application.

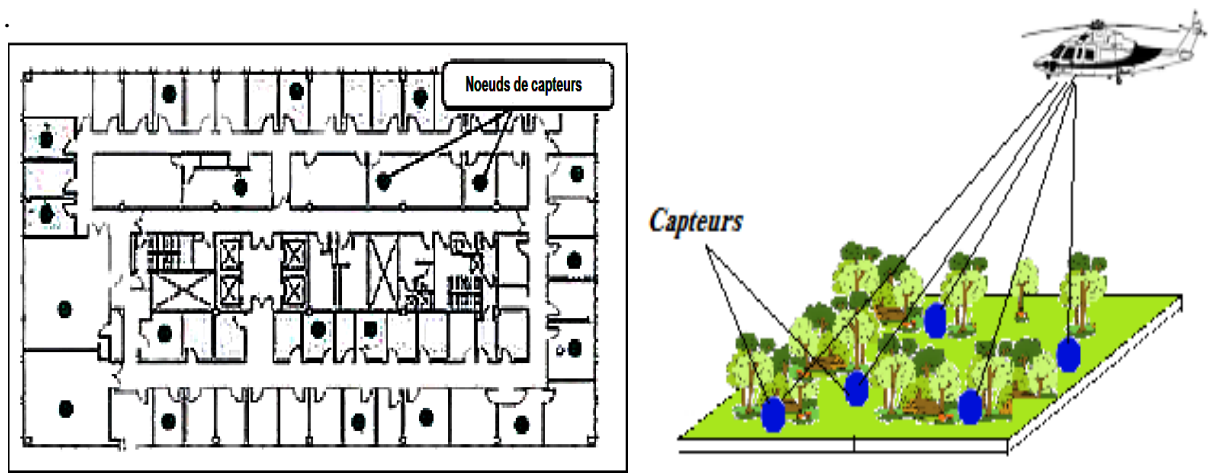


Figure 5. Déploiement déterministe / Déploiement aléatoire

La facilité d'implanter ces réseaux dans des régions rudes ou inaccessibles est due à la faculté d'autonomie énergétique des nœuds de capteurs. Ils peuvent être déployés dans des zones dépourvues d'électricité et de lignes de communication. Cependant, la faculté de miniaturisation et d'autonomie ont créé de nombreuses contraintes lors de la mise en place et l'exploitation d'un réseau de capteurs sans fil. L'une des préoccupations de toute conception

d'un réseau de capteurs est d'optimiser le nombre de nœuds mis en place dans un environnement surveillé, mais toutes la question est là, car plusieurs contraintes surgissent après un placement. Ainsi, il faut assurer que chaque position de la zone à contrôler soit couverte par au moins un capteur.

Aussi, plusieurs représentations ont été exposées pour le déploiement de nœuds de capteurs mobiles qui exigent un fonctionnement fiable avec une utilisation optimale des ressources (temps de déploiement et densité des nœuds mobiles).

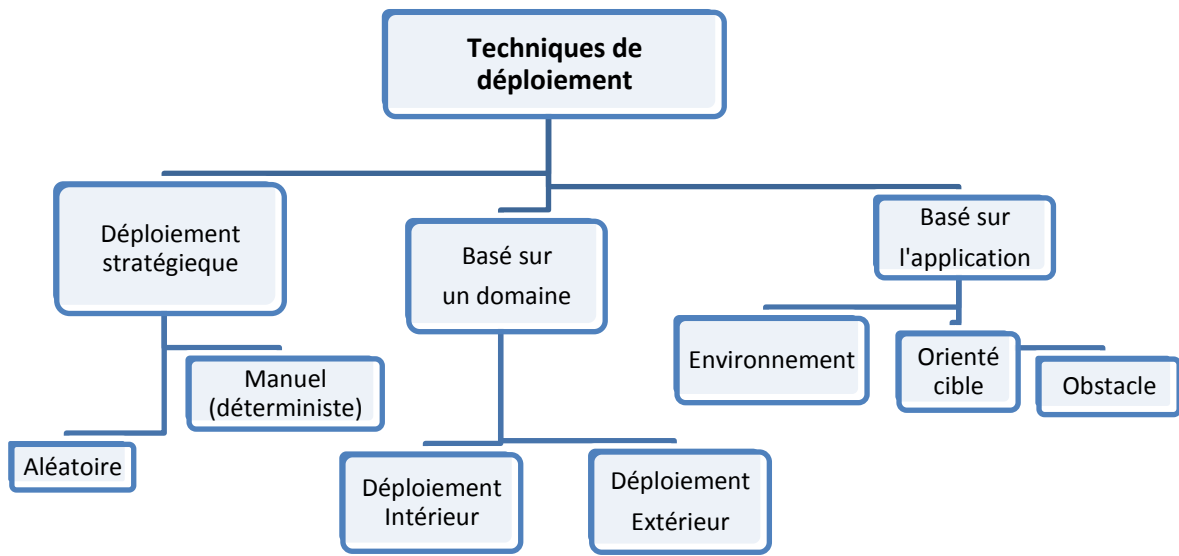


Figure 6. Classification des techniques de déploiement des nœuds de capteurs

III.3. Différents processus du déploiement

La méthode de mise en place des nœuds est préétablie souvent suivant l'attention et le type d'environnement à explorer.

III.3.1. Déploiement initial

Construire un réseau de capteurs commence d'abord par un placement initial des nœuds selon une géométrie prédéfinie (déterministe) comme, par exemple couvrir manuellement la zone, mais cette méthode n'est pas toujours possible à cause de l'hétérogénéité des zones (zones hostiles). On peut trouver aussi un placement initial dense où les nœuds sont lancés par-dessus d'un avion ou un projectile. Cette manière est souvent utilisée lorsque la zone est hostile et l'être humain ne peut y accéder, mais elle dégage des challenges plus complexes. Parmi les méthodes de déploiement aléatoire, l'utilisation d'un arroseur centrifuge à Canon (CCS) (voir Figure 7) issue des travaux de recherche de [23].

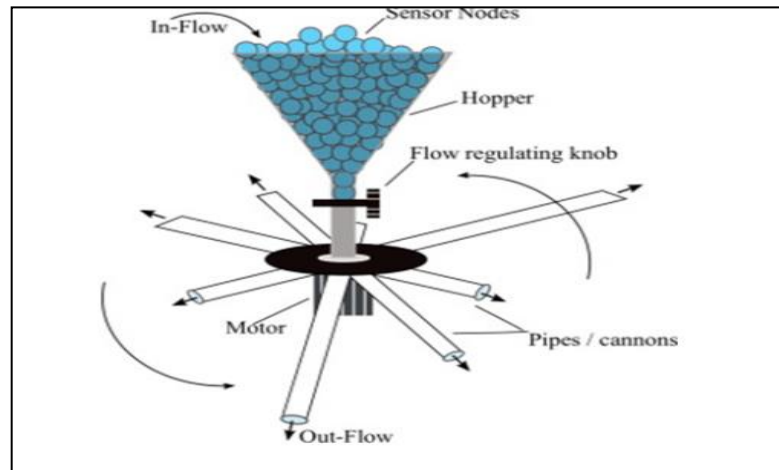


Figure 7. Arroseur Centrifuge à Canon (CCS)

L'objectif principal du modèle est de minimiser le nombre de balayages de l'hélicoptère de déploiement au-dessus de la région candidate à grande échelle pour une meilleure diffusion dans le temps. Ce modèle électromécanique est inspiré de l'arroseur centrifuge utilisé en agriculture comme système d'irrigation. Cette solution n'est plus utilisée car le nombre de défaillances de nœuds résultant de l'impact est très important.

Pour pallier au problème de perte de nœud, UAD (Uniform airdrop deployment method for large-scale wireless sensor networks) a été proposé par [24] ; les auteurs proposent que l'hélicoptère survole le centre d'une région circulaire postulante de rayon R situé à une altitude spécifique H , et que les nœuds de capteurs sont montés sur parachute et largués à partir de ce point ; le vent n'est pas pris en considération. Cette méthode amortit le risque de perte de nœuds, mais peut avoir une concentration de nœud dans une seule région (Figure 8).

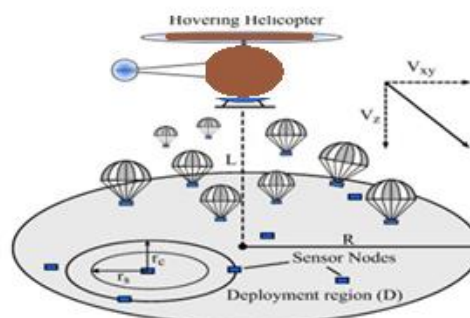


Figure 8. Nœuds de capteurs largués par un hélicoptère

En résumé, la densité des nœuds dans un réseau de capteurs peut causer un problème quant à leur déploiement selon une architecture attendue. Toutefois, un déploiement initial est quasi-obligatoire afin de :

- Réduire les coûts d'installation
- Augmenter la flexibilité d'arrangement des nœuds
- Assister l'organisation des nœuds.

III.3.2. Post-Déploiement

Une fois le déploiement initial établi, le réseau peut tisser une nouvelle topologie causée par des perturbations au niveau des nœuds comme :

- affaiblissement des batteries
- mobilité des nœuds,
- hostilité et difficulté de transmission.
- problème ou dysfonctionnement des nœuds
- etc.

III.3.3. Redéploiement

En général, un redéploiement n'est programmé que dans certaines applications qui regroupent certaines attentes. Ainsi, on peut reprendre la topologie présente et la restructurer en ajoutant ou en substituant les capteurs défectueux ou non fonctionnels par d'autres.

III.4. Contraintes du déploiement dans les réseaux de capteurs

Certains réseaux de capteurs requièrent une attention particulière, car leur déploiement sollicite une prise en compte de plusieurs contraintes. La contrainte de couverture précise et totale d'une zone est le premier objectif d'un déploiement optimisé, mais la détection, la localisation, la connectivité du réseau, la fiabilité des données, la consommation d'énergie, la tolérance aux pannes, la perpétuité du réseau et le routage sont aussi primordiaux qu'indispensables, et doivent être pris en compte pour toute conception optimale d'un réseau RCSF. Dans ce qui suit, nous allons donner un aperçu de quelques études faites dans ce domaine qu'est le déploiement.

III.4.1. Couverture et connectivité

Un des buts essentiels dans un réseau de capteurs sans fil est la couverture de toute la zone d'intérêt pour suivre un phénomène ou relever des données prépondérantes. L'optimisation du déploiement consiste à minimiser le nombre de capteurs et maximiser la couverture de ladite zone. Il est à noter que le problème de la maximisation de la couverture est connu pour être NP-difficile. En tant que telles, les méthodes dominantes reposent souvent sur des méta-heuristiques [25].

Les modèles de couverture évaluent l'aptitude et la qualité de détection en se servant des dépendances géométriques entre une cible et un capteur. Généralement, un tel modèle est présenté comme une distance euclidienne (éventuellement sous des angles) entre une cible et les capteurs. Presque toutes les approches se basent sur un déploiement qui garantit une couverture binaire.

Plusieurs approches sont présentes dans la littérature, depuis que Joseph O'Rourke a mis au point son algorithme, « Art Gallery Theorems and Algorithms » [26], relatif à la couverture de chaque coin d'une galerie d'art par au moins un vigile. Par exemple, l'idée de choisir les zones critiques d'une région plutôt que toute la surface à surveiller proposée par [27] afin de minimiser le nombre de capteurs. D'autres auteurs [28] [29] [30], se sont penchés sur le problème de couverture en trois dimensions soulignant la difficulté de surmonter les problèmes rencontrés par les problèmes de déploiement.

Kumar et al. [31] affirment aussi que pour un ensemble de points cibles, trouver un nombre minimum de positions pour placer des nœuds de capteurs remplissant à la fois la couverture et la connectivité est un problème NP-complet. Ces mêmes auteurs proposent alors une solution basée sur un algorithme génétique pour résoudre ce problème, tout en prenant en compte les différentes pannes révélées par quelques capteurs. Ainsi, le schéma proposé fournit une k -couverture à toutes les cibles et une m -connectivité à chaque nœud capteur. D'autres études ([32], [33] et [34]) se sont portées sur des algorithmes de force virtuels traitant la question de couverture et de déploiement des nœuds. Ils manipulent des nœuds flexibles qui possèdent une force d'attraction et d'éloignement pour assurer une mobilité et maintenir la connectivité. Divers autres approches se sont succédées. Par exemple, l'approche de Zhang et al. [35] se base sur l'intelligence collaborative des réseaux auto-organisés comme l'algorithme de colonies d'abeilles, de colonies de fourmis ou d'autres optimisations adaptées. Les auteurs appliquent des forces pour calculer le taux de mouvement des capteurs placés aléatoirement. Lorsque les capteurs sont localisés, un déplacement ponctuel est appliqué et les capteurs sont redéployés pour déterminer une meilleure position. Un inconvénient majeur de cette approche est qu'elle est coûteuse en temps de calcul. Au fur et à mesure que le nombre de capteurs augmente, chaque nouveau déploiement entraîne une quantité de calculs relativement importante (voir Figure 9).

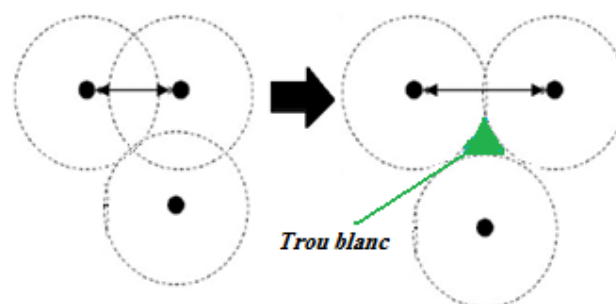


Figure. 9. Capteurs avant/après application de la VFA

Les auteurs de [36] proposent un autre modèle de déploiement aléatoire initiale suivi d'une auto-organisation des nœuds sur des régions adressables dont chacune comporte un nombre de capteurs concentrique autour d'un puits (Sink). Chaque région ou zone est identifiée de manière unique par sa distance au puits et son angle ainsi que tous les capteurs de cette région. Cette structuration optimise la distance entre les nœuds et contribue clairement à la réduction du nombre, mais elle peut générer beaucoup de trous blanc, ce qui nuit à la couverture. Dans [36bis] les auteurs proposent une stratégie de déploiement et une couverture en 3D. Ils articulent et annonce que le déploiement des nœuds de capteurs est un problème critique dans les RCSF, car il affecte la couverture et la connectivité du réseau, ajoutant à leurs propos que la couverture dans les réseaux de capteurs sans fil est une mesure de la qualité et de la durée d'observation de l'espace physique par les capteurs. Ils proposent différentes stratégies comparatives de déploiement de nœuds de capteurs pour les RCSF 3D, afin d'obtenir une couverture maximale. Ils mettent en place un algorithme d'ordonnancement pour minimiser le nombre de nœuds de capteurs utilisés dans la prédiction de couverture. L'étude donne également une comparaison entre les différents schémas de déploiement de nœuds de capteurs 3D, tel que (voir Figure 10) :

- Stratégie de déploiement de nœuds de capteurs de type Prisme
- Déploiement de nœuds de cube
- Déploiement de nœuds prismatiques hexagonaux
- Déploiement de nœuds pyramidaux

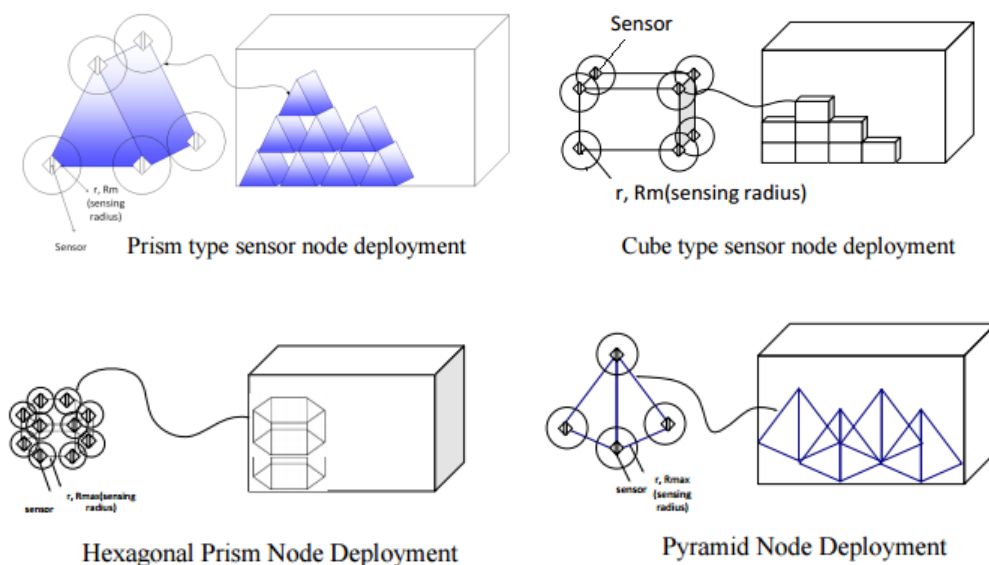


Figure 10. Modèles de déploiement des capteurs en 3D [36]

En conclusion, il résume sa théorie en mettant en avant leur observation relative à leur étude et souligne que le schéma de déploiement de nœuds en pyramide présente la prédiction de couverture la plus élevée et qu'il est presque égale au déploiement de type prisme, mais le type de prisme hexagonal est le moins prédictible en couverture.

Boualem et al. [37], quant à eux, annoncent que la couverture d'une zone est un problème difficile à résoudre avec un nombre minimal de nœuds de capteurs pour une durée maximale, surtout dans les zones étendues. Aussi, les auteurs critiquent le déploiement déterministe par sa limitation de la zone d'intérêt et la configuration du déploiement aléatoire par sa distribution non équitable des nœuds de capteurs. De ce fait, ils ont proposé une optimisation de la zone de couverture par un déploiement hybride (semi-aléatoire) qui accroît les avantages et minimise les inconvénients de l'application du déploiement aléatoire et du déploiement déterministe. Ce type comporte deux étapes : l'étape de configuration anticipée et l'étape du processus d'ordonnancement. La zone d'intérêt est divisée géographiquement en « n » sous-zones de dimension « $L \times L$ » et de diagonale « D ». Cette méthode a plusieurs caractéristiques, telles que (voir Figure 11) :

- Activez un seul nœud dans chaque sous-zone pour équilibrer le réseau en termes de consommation d'énergie, garantir la connectivité et augmenter la durée de vie du réseau.
- L'activation d'un seul nœud par sous-zone garantit une couverture complète de la zone d'intérêt pendant un temps maximal, sans avoir à mettre en œuvre de stratégie.
- Chaque nœud joue un double rôle : un rôle de surveillance et un rôle de communication.

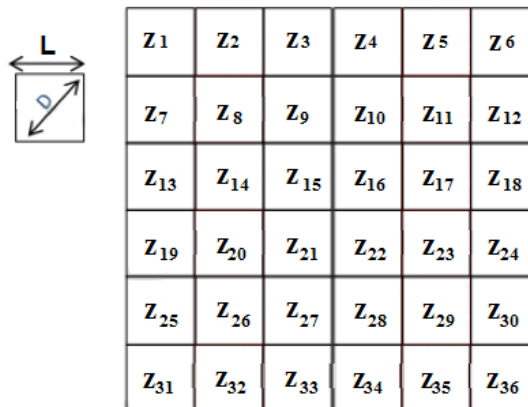


Figure 11. Découpage de la zone d'intérêt

Pour résoudre le problème des trous blancs après une auto-organisation, Shen et al. [38] mettent au point un algorithme adaptatif (Adaptive Hole Connected Healing), assurant une connectivité quasi-total du réseau. L'idée de base est de trouver une combinaison adaptative à partir du schéma initiale qui optimise les trous blancs. Par la suite, ils améliorent

leur heuristique et mettent en évidence deux autres algorithmes (Insufficient AHCH, General AHCH).

Un autre problème, relatif à l'effet des obstacles sur la couverture et la connectivité, est examiné par Wang, Hu et Tseng dans [39]. Les auteurs supposent un ensemble homogène de nœuds avec un déploiement déterministe, et tentent d'assurer la couverture et la connectivité avec un nombre minimal de nœuds. Ils divisent le champ d'intérêt en zones plus petites pour déterminer où déployer les capteurs. Dans ce même contexte, Zou et al. [40] font ressortir l'impact des forces exercées par les obstacles, et annoncent l'intérêt essentiel de la connaissance du terrain afin d'améliorer la couverture et le déploiement aléatoire initial.

Le choix de capteurs pour un déploiement optimal est très important afin couvrir une zone d'intérêt. Dans ce contexte, une étude est éclaircie par une mise en place d'un réseau de capteurs sans fil behavioriste pour l'assistance et le maintien à domicile des personnes âgées. Ce suivi dans un habitat intelligent a été donné par les travaux de [41]. L'auteur a proposé d'implanter un capteur vidéo KINECT (Figure 12) pour la détection des chutes et la surveillance des aînés à domicile .

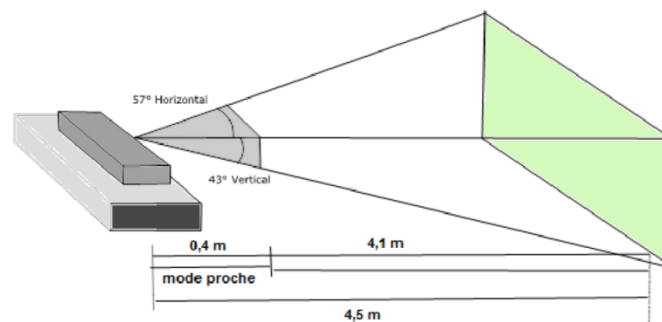


Figure 12. Champs de vision de la KINECT

Très vite, l'auteur s'est rendu compte que la surveillance nécessitait un équipement de plusieurs capteurs afin de garantir la sécurité des personnes âgées, et ce, de leurs comportements et gestes. Ainsi, elle souligne que l'inconvénient majeur de cette hybridation de capteurs relève de l'orientation unidirectionnelle des caméras vers la cible et le risque d'avoir des positions inexploitées. D'autres approches se sont succédé en apportant une flexibilité et une mobilité aux caméras.

Une étude particulière [42] a attiré notre attention ; il s'agit de mesurer l'expansion de polluants dans les cours d'eau afin de surveiller la croissance de bactéries dans les eaux de baignade. Ces réseaux de capteurs sans fil étendus et robustes exploitent une méthode de synchronisation orientée émetteur à récepteur. Cette technique de lien atypique (Figure 13) permet d'évaluer le délai de propagation de l'horloge dans une topologie en Arbre. L'horloge

du réseau est imposée par le puits, et celle-ci se propage de manière hiérarchique sur la topologie.

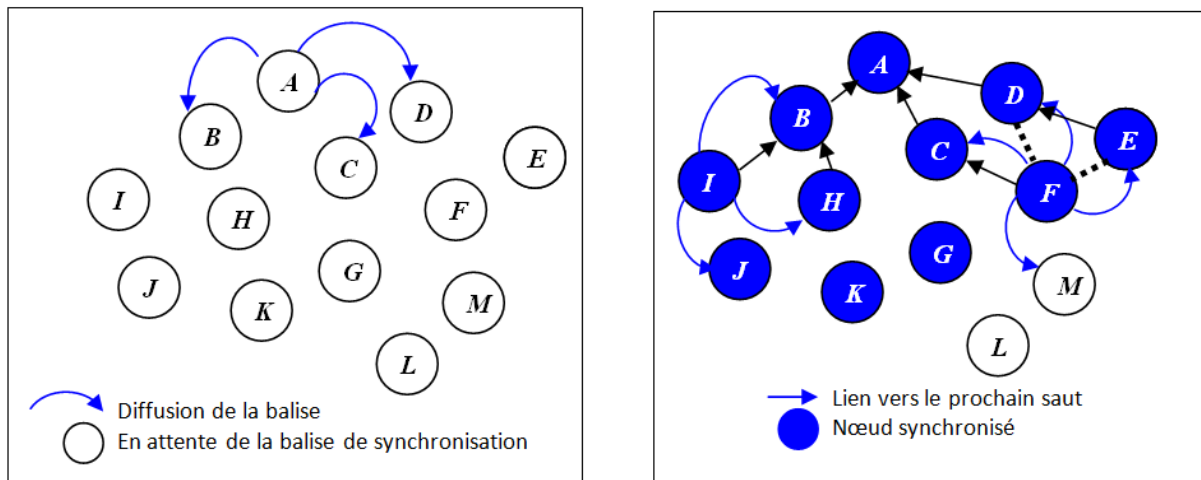


Figure 13. La propagation hiérarchique de l'horloge dans le réseau [20]

Un tableau comparatif des approches suscitées est donné ci-dessous :

Approche	Avantages	Inconvénients
Auto-organisation [32], [33], [34][36][37]	<ul style="list-style-type: none"> - Optimise le nombre des capteurs. - Gère mieux les placement des nœuds. -Dépoupage en zone d'intérêt. 	<ul style="list-style-type: none"> - Problème des capteurs défailants. - Trous Blancs (Pb. Couverture) - Temps de calcul élevé. - Coût élevé
Stratégie de déploiement 3D [36bis]	<ul style="list-style-type: none"> - Contrôle le duty-cle des capteurs. - Meilleure prédiction (prisme). 	<ul style="list-style-type: none"> - Les positions doivent être connues à l'avance. - Temps de calcul élevé.
Optimisation de la couverture [38][39][40]	<ul style="list-style-type: none"> - Optimise le déploiement initiale. - minimise les trous blancs. - Prise en considération des obstacles. 	<ul style="list-style-type: none"> - Coût élevé. - Temps de calcul élevé. - Présence des trous blancs.
Déploiement behavioriste [41]	<ul style="list-style-type: none"> - Facilite le traitement des images. - Utilise la technologie 3D. - Cadre la zone à surveiller [0,4- 4,5m] - Contrôle à distance. - Dépend du courant du domicile. 	<ul style="list-style-type: none"> - Dépend du courant. - Taille physique du capteur - Trous Blancs (Pb. Couverture).

Déploiement hiérarchique [42]	<ul style="list-style-type: none"> - Topologie hiérarchique. - Optimise le temps de réponse. - Conçu pour les grandes distances. 	<ul style="list-style-type: none"> - Augmentation de la taille des fenêtres de réceptions. - Consommation visible d'énergie.
-------------------------------------	---	--

Tableau 3 : Synthèse de quelques méthodes utilisées dans le déploiement des capteurs

A l'issue de cette étude, notre travail va se porter sur un modèle hybride mettant en fusion les modèles :

- L'idée de station de base centrée qui est mentionnée dans les travaux de [30].
- L'approche citée dans les travaux de [35], où ils utilisent un puits pour chaque région divisée. Seulement au lieu de prendre le sink, on utilisera un nœud de capteur agrégateur.
- Le découpage de la zone d'intérêt en zones de dimension $L \times L$ utilisé dans les travaux de [37].

III.4.2. Energie dans les réseaux de capteurs

La consommation énergétique d'un capteur peut être influencée par différents facteurs :

- a) la taille de la zone à couvrir par ce capteur,
- b) la distance séparant ce capteur des autres pour l'envoi d'information,
- c) le nombre de paquets de données transitant par ce capteur.

Dans certains cas, le dispositif d'acquisition peut être réglé pour que sa portée soit plus ou moins grande. Ce réglage aura un impact direct sur la consommation d'énergie. Pour ce qui est de la distance séparant le capteur du reste du réseau, l'envoi d'information nécessite une énergie non négligeable.

Notre étude, portera sur la consommation d'énergie dans les différents modules mis en actions: le module d'acquisition, de transmission, de la mise en veille et de traitement de données. Nous répondrons ainsi à la question suivante : « Quel module nous coûte le plus en énergie ? »

III. 4.3. Déploiement et Routage

Beaucoup d'études se sont focalisées sur le côté programmation pour résoudre le problème de déploiement des nœuds de capteurs dans les réseaux de capteurs sans fils. Dans cette perspective, une classification a été mise en évidence, reposant sur deux points essentiels : la topologie du réseau (plate ou hiérarchique) et le second point l'objectif de l'application. Mais un bon routage doit satisfaire certaines contraintes :

- limitation de l'énergie impliquant une topologie instable,
- mémoire de stockage limité,

- adressage global absent,
- réseau à sources multiples / destination unique
- bande-passante réduite,
- gestion des ressources stricte,
- données redondantes provenant de différentes sources,
- capacité de calcul faible,

De nombreux travaux ont été étalés dans ce sens ayant comme objectif d'approcher à une satisfaction de ces contraintes. L'inondation du réseau fût, parmi les premières approches dans la conception des réseaux de capteurs sans fil, sans doute pas la meilleurs à cause de sa consommation énergétique, son coût élevé en bande passante et pour couronné le tous, les interférences résultants de ces multiples transmissions. Une approche, très riche et très vite adoptée par la communauté scientifique, est celle des auteurs de [43] qui proposent un ensemble de protocoles modélisant des diffusions aléatoires pour localiser les nœuds de capteurs adjacents, suite à la portée de communication. L'étude suppose que les nœuds sont synchronisés et fixes (immobiles). Les auteurs mettent en application une théorie : les nœuds peuvent se situer dans l'un des trois états suivants :

1. en écoute,
2. en transmission
3. en veille

Ainsi, le réseau de capteurs se voit ses nœuds permuter entre les trois états possibles. Des algorithmes entrent en exécution pour optimiser les liens entre les nœuds et minimiser la consommation d'énergie.

III.5. Conclusion

Dans ce chapitre, nous voulions donner un bref aperçu sur les différentes suggestions de déploiements. Néanmoins, la réalisation des réseaux de capteurs sans fil doit se soumettre à quelques contraintes déjà citées ci-dessus, comme le changement de la topologie, la consommation d'énergie, la couverture, etc.

Chapitre IV
Détection et coopération
dans les Réseaux de capteurs sans fil (RCSF)

IV.1. Introduction

Du déploiement des capteurs résulte un réseau qui peut être exploité dans des domaines différents. Ces capteurs sont capables de détecter et mesurer approximativement des grandeurs physiques dans son environnement (humidité, température, lumière, pression, etc.) et de les transmettre, via des ondes radio, à une autre entité (capteur, agrégateur ou puits). De ce fait, il est nécessaire de connaître les positions géographiques exactes ou estimées des nœuds causant la détection des événements afin de fournir une assistance urgente en un temps record.

Nous allons prouver, dans ce travail, que les opérations de transmission et de réception de données consomment la plus grande partie de l'énergie des capteurs. Ainsi, un déploiement avec parcimonie des capteurs est plus qu'exigés, afin de minimiser les interférences et les redondances des émissions des mêmes valeurs captées vers les puits ou les nœuds agrégateurs. De plus, cette optimisation aura un impact sur la conservation d'énergie des capteurs, car responsabiliser les capteurs élus pour conduire l'information captée à la station de base, augmentera pleinement les performances du réseau qui influera positivement sur sa durée de vie. Cette dernière, reste tout de même, un point de recherche très convoité par la communauté des chercheurs qui déploient beaucoup d'effort afin de garder un rendement énergétique accrue des plates-formes en usant souvent des protocoles de communication.

Dans ce chapitre, nous allons recenser un aperçu de base des principales techniques usées pour la détection.

IV.2. Détection et routage

Notant que la communication, dans presque tous les réseaux de capteurs sans fil, est orientée « de bas en haut » du simple capteur vers le puits de données (Figure 13). Ainsi, le but de toutes les communications est de véhiculer l'information vers une passerelle. Dans cette perspective, des protocoles de routages ont été élaborés pour les réseaux de capteurs afin de résoudre la surconsommation de l'énergie. Cette idée est très vite élargie et orientée vers un nouveau domaine de recherche encadrant des techniques structurées de placement de nœuds. Il tente le plus souvent d'éviter la redondance d'information ou bien d'éviter le sur calcul, dans la seule préoccupation est de procurer une assez bonne qualité de service, de minimiser la consommation d'énergie et par conséquent de persister la durée de vie du réseau.

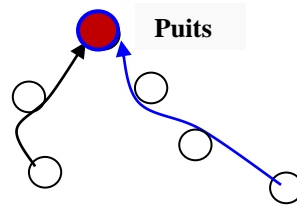


Figure 13. Trafic « de bas en haut »

Dans un réseau de capteur sans fil, la quantité de nœuds déployés peut dépasser des centaines voir des milliers ou plus. De ce fait, le réseau ne peut être doté d'une topologie fixe ou préétablie au départ du fait de la durée de vie limitée des capteurs. Ce nombre a incité les chercheurs à concevoir des approches adaptables et efficaces pouvant gérer un flux non négligeable de paquets de données et de paquets de contrôle. Ainsi, il faut signaler que plus le nombre de capteurs est grand plus le risque de saturation et de surcharge du réseau est plus que probable entraînant un étranglement de la bande passante, et par conséquent une baisse de fiabilité. C'est dans cette perspective qu'on a recensé beaucoup de protocoles de routages ayant comme but principale l'optimisation de la consommation d'énergie, même si ces protocoles empruntent d'autres méthodes pour s'en rapprocher, comme par exemple le contrôle du flux de données ou l'optimisation des traitements au niveaux des capteurs.

Notre travail se focalise sur cet axe de recherche, à savoir, la détection d'un évènement et un consensus de fusion ou de traitement de données, pour minimiser la redondance d'informations ainsi que le nombre de capteurs déployés, et maintenir une durée de vie aussi grande que possible de notre réseau.

Un algorithme de routage procède à l'acheminement des paquets de données soit par un seul saut, soit à travers un réseau multi-saut vers la station de base. Dans ce dernier cas, la couche de liaison prend la responsabilité de localiser les nœuds voisins afin de véhiculer les informations captées. La diffusion en Broadcast n'est pas possible car les études ont montrés que le réseau arrive à saturation en si peu de temps dès que le nombre de capteurs est important. Un routage dans les réseaux de capteurs sans fil diffère de celui des réseaux Ad-hoc du fait des ressources limitées (mémoire, puissance de calcul, énergie, etc.) et du canal de communication sans fil partagé et qui en plus est influencé par des interférences extérieures. Une représentation d'une taxonomie aux différents protocoles de routages existants est donnée dans la figure 14.

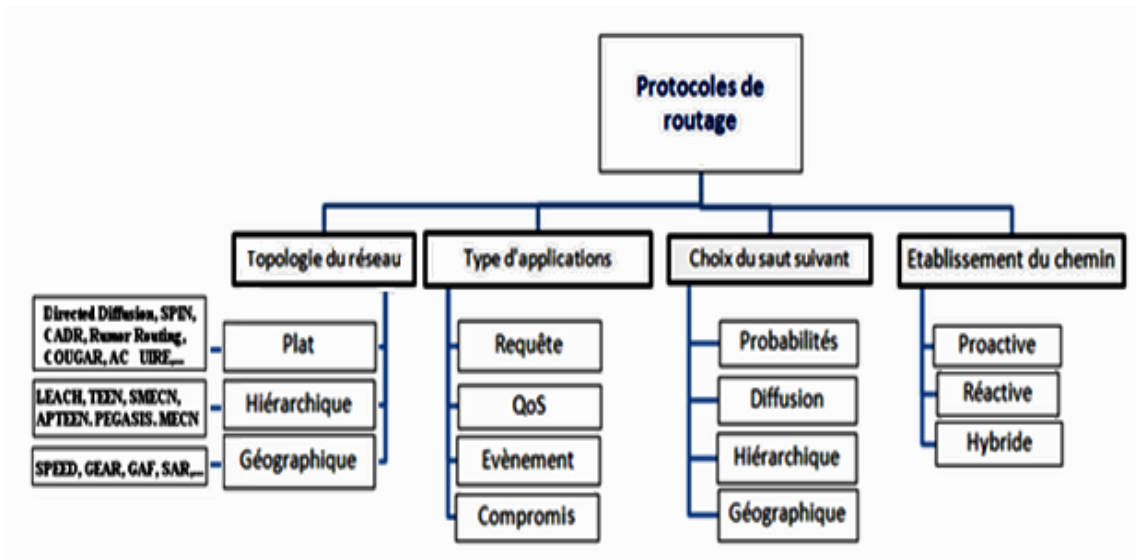


Figure 14. Taxonomie sur les protocoles de routages dans les RCSF

IV.2.1. Routage et contraintes

La conception d'un protocole de routage, dans les réseaux de capteurs sans fil, doit répondre à certaines contraintes comme :

1. Energie limitée.
2. Bande passante limitée.
3. Capacité de traitement limitée.
4. Ressource et capacité de stockage limitée.
5. Notion d'adressage absente.
6. Redondances d'informations.
7. Plusieurs sources et une seul destination.

La littérature, nous a éclaircis sur la méthode de placement des nœuds. Ainsi, la présentation la plus efficace d'un réseau est son organisation autour d'une structure en graphe orienté soutenu sans cycle en direction du puits. Cette représentation arborescente est appelée sur le simulateur COOJA/Contiki DODAG (Destination-Oriented Directed Acyclic Graph), où les arrêtes représentent les chemins par défaut empruntées dans le réseau. Le protocole RPL (Routing Protocol for Low-Power and Lossy Networks.), par exemple, garantie la construction de cet arbre de collecte, et s'adapte à la structure réseau variable.

IV.2.2. La détection dans les RCSF

Dans cette partie nous allons recenser quelques méthodes de détection connues jusqu'à aujourd'hui.

IV.2.2.1. Définition

La détection est le processus de capture des grandeurs physiques correspondant aux états physiques ou environnementales, comme : la température, la pression, la vitesse, les mouvements, le son, l'humidité, la lumière, la direction, le volume, etc., à l'aide de l'unité de captage du nœud capteur, composée d'un récepteur et un transducteur qui convertit le signal analogique du récepteur en un signal numérique compréhensible par l'unité de traitement.

IV.2.2.2. Méthodes de détection-coopération

Plusieurs capteurs enregistrent leurs données captées qui essaient de les relayer aux capteurs voisins. Ces valeurs sont le plus souvent les mêmes ou proche d'une certaine valeur. Une coopération de ces nœuds permettra la réduction de l'ampleur de ces données rassemblées par les capteurs. Ainsi, cette approche de coopération résoudra le problème de la précision des valeurs mesurées dans les régions à faible couverture.

Parallèlement, il est donc indispensable de prendre en compte les ressources limitées des nœuds de capteurs comme le stockage de l'information, la durée de communication et la complexité du calcul [44]. Néanmoins, les valeurs captées peuvent être mémorisées s'il n'y a pas une urgence d'évènement ou transmises à une station de fusion qui peut être aussi un capteur agrégateur. L'opération de détection ne peut prédire la quantité ou la taille des valeurs mesurées à un instant donnée, c'est pourquoi il faut réduire le maximum la taille de ces données par des traitements ou une classification [45].

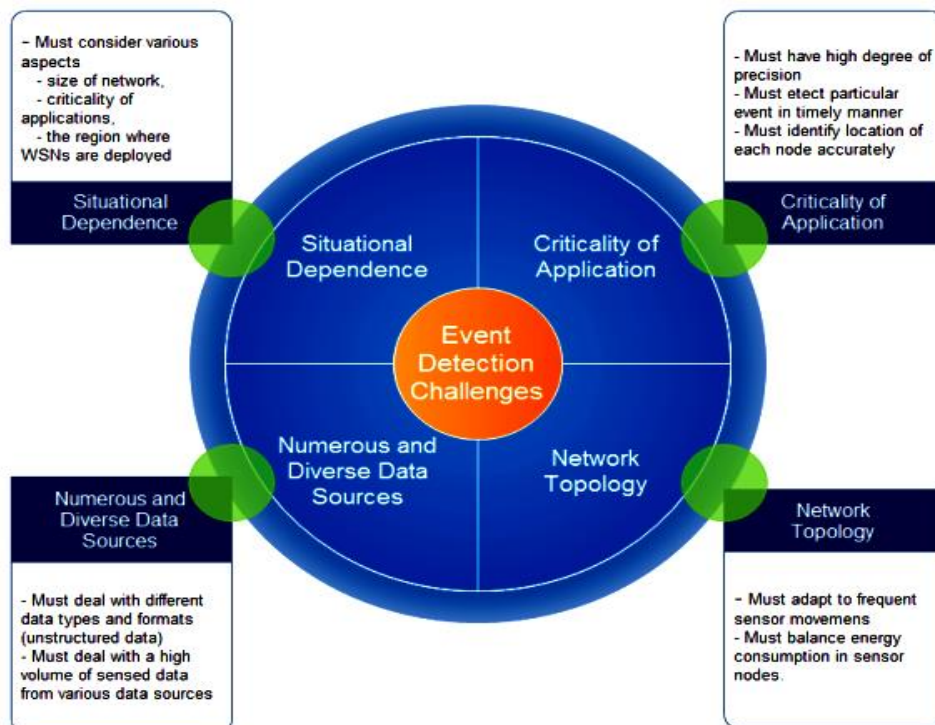


Figure 15. Défis de la détection d'évènements dans les RCSF [44].

La plupart des méthodes de détection d'événements fait partie de l'une des trois catégories suivantes : statistiques, probabiliste ou intelligence artificielle et apprentissage automatique. Kerman et al. [46] proposent une représentation de la détection d'événements avec ses défis (voir Figure 15).

Beaucoup d'auteurs se sont penchés sur les méthodes statistiques en détection. L'analyse des composantes principales (ACP) était toujours en amont, en explorant la détection d'un phénomène et les prédictions qui en découlent. D'autres essayent une détection d'évènement réductive, en mettant l'accent sur le fonctionnement d'un certain nombre de capteurs à un instant donnée, par rapport à la totalité des capteurs du réseau. Les auteurs [47] proposent deux heuristiques, la première est une optimisation de la vraisemblance marginale et la seconde est basée sur les statistiques dans un cadre bayésien. Les auteurs soulignent leur exploit avec une probabilité de détection plus élevée que la solution traditionnelle de programmation linéaire.

Vu et al. [48] étudient une nouvelle méthode capable de détecter des événements et de délivrer des alertes en temps voulu dans les RCSF. Un algorithme qui tient compte de la topologie du réseau et des capacités de routage a été proposé. L'algorithme présente plusieurs avantages dont la réduction du temps de notification, la conservation de l'énergie et la qualité acceptable des exigences de surveillance dans les applications d'alarme d'événements.

La complexité des problèmes de détection d'événements pose toute une série de défis indépendamment du problème de détection spécifique et du domaine à explorer. Il existe plusieurs défis communs et des vérités universelles, dans le développement et l'application des méthodes de détection d'événements.

a) L'agrégation

L'agrégation souvent appelée « clustering » est une approche qui trouve son origine dans le domaine de l'intelligence artificielle et d'aide à la prise de décision. Quand tous les critères d'appréciation d'une solution sont regroupés en un critère unique ; ce type de décision est appelé monocritère. Mais, lorsque le problème devient plus complexe et qu'on veut arbitrer entre plusieurs indicateurs, on parlera de décision multicritère. D'ailleurs, Bernard Roy, après avoir été confronté aux limites des modèles de décisions à rationalité limitée fonde son approche multicritère d'aide à la décision et distingue trois types de paradigmes :

- Le paradigme du choix : Choisir la solution optimale pour un problème donnée.
- Le paradigme du classement (Ranking) : mettre les solutions selon une représentation descendante de classement.
- Le paradigme du tri (Sorting) : Catégoriser les solutions.

La figure 16 illustre un organigramme d'une agrégation multicritères [49], qui permet de spécifier et classer l'agrégation en deux types :

- Grouper puis comparer.
- Comparer par paires.

Définition : Fonction d'agrégation.

Soit $\vartheta = \{v_1(a), \dots, v_n(a)\}$: Ensemble des valeurs partiels d'une alternative $a \in \mathring{A}$, qui sont obtenus suivant un ensemble de critères $\mathring{C} = \{c_1, \dots, c_n\}$, la valeur de préférence globale de a est donnée par une fonction $f : \vartheta \rightarrow \mathbb{R}$ obtenue suivant une agrégation de critères à l'aide d'une fonction d'agrégation $k : \mathbb{R}^n \rightarrow \mathbb{R}$:

$$f(a) = k(v_1(a), \dots, v_n(a)) \quad (8)$$

Notons qu'une relation de préférence globale entre deux alternatives a_1 et $a_2 \in \mathring{A}$, est définie comme suit [49] : On peut définir une relation de préférence globale « \ll_f » favorisant une alternative sur l'autre par rapport à l'ensemble global des critères.

- Dans les approches de type « grouper puis comparer », la préférence « a_1 » est mieux jugée que « a_2 », est symbolisée par :

$$\ll a_1 \ll_f a_2 \gg \Leftrightarrow k(v_1(a_1), \dots, v_n(a_1)) \geq k(v_1(a_2), \dots, v_n(a_2)) \quad (9)$$

- Dans l'approche de type "comparer par paires", la relation de préférences est symbolisée au moyen des critères. C'est-à-dire que les alternatives sont jugées suivant un seul critère préféré et non sur la totalité de l'ensemble \mathring{C} .

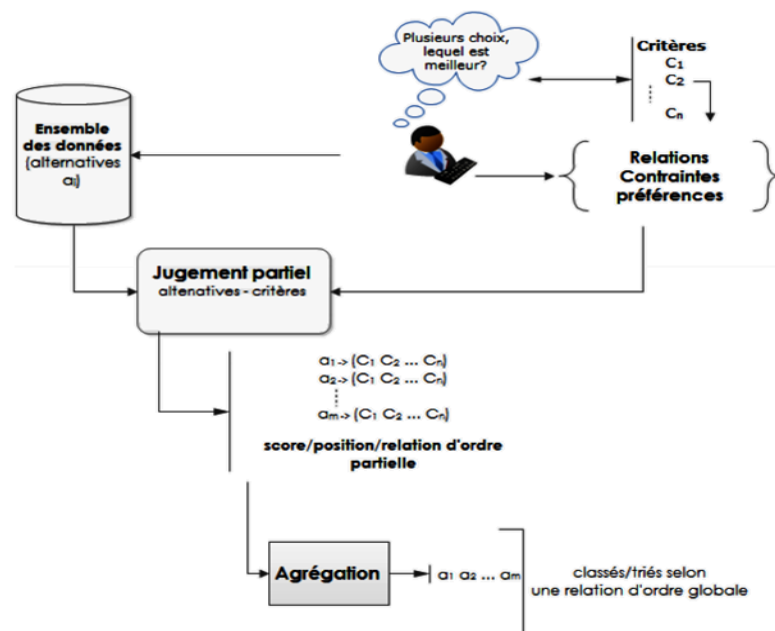


Figure 16. Architecture des approches d'agrégation multicritères [49]

- Classification des approches multicritères

Selon le type d'agrégation (préférence, résultat, signe, etc.) nous classons principalement les approches multicritères en deux catégories (voir Figure 17). Mais d'autres catégories restent envisageables (par compensation i.e., moyenne pondérée, conjonctives, disjonctives ou basés sur une pondération, etc.).

1. Agrégation de valeurs : la totalité des alternatives sont obtenues avec des résultats intermédiaires en appliquant tous les critères fixés. Ainsi, l'agrégation est soit une combinaison des valeurs numériques linéaires, soit une utilisation des opérateurs classiques d'agrégations.

2. Agrégation de listes : les critères sont remplacés par des listes d'ordonnements. L'agrégation peut concerner des positions plutôt que des valeurs. Ces positions représentent le classement des alternatives dans les listes.

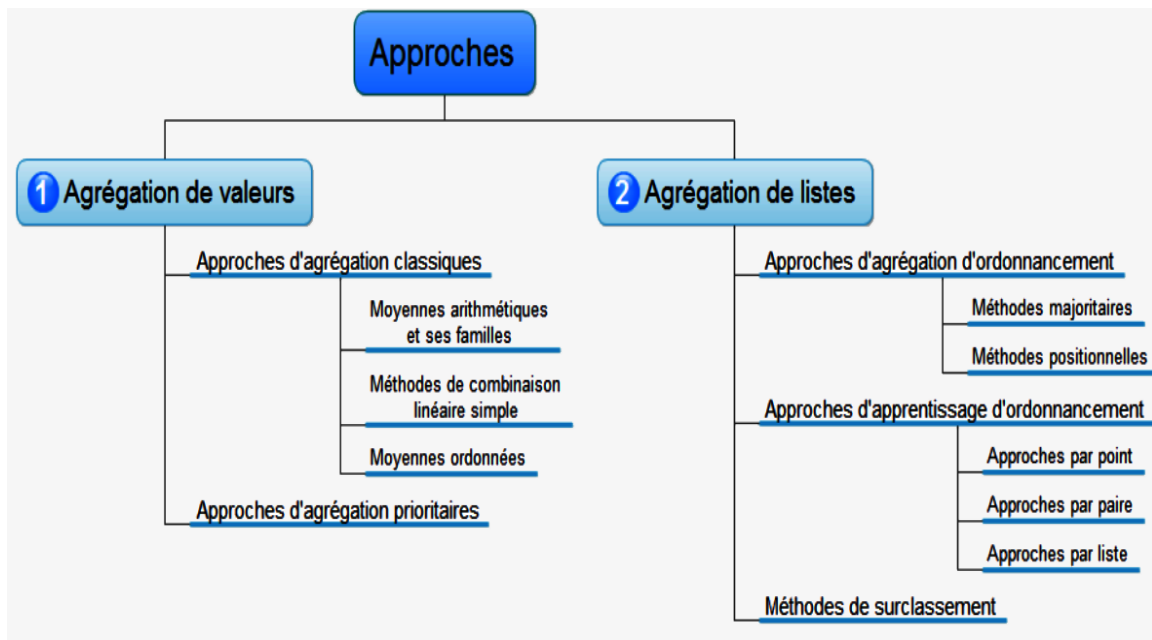


Figure 17. Classification des approches d'agrégation multicritères [49]

Une fois que les nœuds de capteurs sont déployés dans une zone d'intérêt de détection, ils peuvent nous aider à observer et à agréger les données. Beaucoup d'auteurs essaient encore de mettre en place des algorithmes plus efficaces, afin d'utiliser l'énergie limitée des nœuds de capteurs pour prolonger la durée de vie du réseau. Ils ont utilisé des protocoles de routages multi-sauts basés sur des architectures en grappes (LEACH) ou en chaînes (PEGASIS) ou sur des méthodes arborescentes (TREEPSI) [50].

L'introduction de la coopération dans un réseau de capteurs sans fil (RCSF) est exploitée à différents niveaux, allant d'une collection de nœuds collaborant pour transmettre

les données recueillies d'un environnement vers la station de base en utilisant des techniques efficaces d'agrégation de données et de clustering, aux nœuds collaborant pour aviser les changements évènementiels afin de contrôler la topologie [51].

b) La détection et l'agrégation

L'agrégation de valeurs captées reste le meilleur moyen pour économiser l'énergie et perdurer la vie du réseau, car le processus d'envoi d'une donnée consensuelle à la station de base évitera pleinement les transmissions inutiles des données redondantes ou des valeurs proches aux autres capteurs à destination du puits. Notre travail dans cette thèse est accentué autour d'une agrégation afin de structurer le déploiement en spécifiant chacun leurs fonctions. Tous les nœuds simples ne communiquent leurs valeurs qu'aux nœuds de capteurs agrégateurs qui transmettront eux-mêmes leur contenu au puits, réduisant ainsi le nombre de capteurs, et bien sûr les perturbations résultantes [52].

L'agrégation est couramment mise en évidence pour parvenir à une forme compressée des valeurs des capteurs, utilisant souvent des opérations comme la moyenne, le maximum ou une autre formule mathématique prouvant sa fonction d'agrégation. Malheureusement, en présence d'une densité accrue de nœuds, plusieurs nœuds peuvent avoir un dysfonctionnement (batteries faibles, choc, humidité, etc.) et collectent des fausses données, et relayent des valeurs fausses et nuisent au résultat final mis à la disposition du puits [53]. Cependant, il faut concevoir une meilleure méthode d'agrégation pour réduire non seulement les transmissions redondantes, mais aussi économise une grande quantité d'énergie des modules de transmission, de réception et de l'écoute radio et du processeur de calcul.

Notre contribution dans cette thèse met en avant un deuxième apport ; une méthode de discrétisation avec une agrégation des données par intervalle de confiance pouvant parvenir à catégoriser les valeurs en regroupant toutes les valeurs captées dans des classes pouvant ainsi détecter les valeurs aberrantes [45].

Les capteurs s'associent avec les nœuds qui leurs sont proches pour détecter et informer le puits central. Plusieurs recherches reposent sur la technique d'agrégation et proposent des applications afin de résoudre un problème donné, comme la qualité de service [54], optimisation de latence dans les réseaux de capteurs [55], et d'autres applications afin de prouver l'efficacité énergétique [56]-[61].

c) Dépendance dimensionnelle et temporelle

Le contexte de cette détection est fondée sur une agrégation de données relevant d'une collection spatio-temporelle de données dans un réseau de capteurs distribué où les nœuds

accumulent, dans leurs paquets d'attributs à un intervalle de temps régulier ou irrégulier, des données non corrélées, puis les livrent à la station de base. La redondance temporelle est traitée dans les nœuds de capteurs simples, et un filtrage spatial de données est traité dans les nœuds agrégateurs ou les Clusters-Head (CH). De ce fait, l'agrégation a contribué pleinement dans la question énergétique et a fait accroître visiblement la durée de vie du réseau. Diverses techniques spécifiques d'agrégation de données bien connues ont été étalées par des chercheurs. Une des applications de ce système, assez intéressante, est celle d'un balayage de l'espace du réseau pour identifier en amont les capteurs en bon état, considérées par la suite comme références pour la gestion futur, après une conversation mutuelle par envoi de paquets de contrôle (« Hello »). Néanmoins, il reste à contrôler l'échange élevé de communication qui influe sur la consommation d'énergie [62].

IV.2.2.3. Méthodes de consensus

Dans une architecture déconcentrée, les nœuds de capteurs simples ont les mêmes fonctions : ils s'auto-organisent en communiquant et en échangeant des informations pour effectuer des opérations consensuelles contenues dans des algorithmes distribués avant de transmettre leurs valeurs captées à la station de base. Jinping et al. [63] mettent en place un consensus adaptatif des réseaux de capteurs sans fils, où les informations reçues par chaque capteur sont altérées par un bruit additif, et les données peuvent être obtenues directement ou indirectement par un nœud récepteur. Sur la base de la mesure locale entre le nœud collecteur et ses voisins, un état consensuel est fourni. En fonction des erreurs entre l'état attendu et les états de tous les capteurs, les protocoles de consensus adaptatifs sont fournis par le théorème de Gershgorin et la méthode fonctionnelle de Lyapunov. Une autre approche, toujours dans le contexte du contrôle distribué, un consensus rapide et parcimonieux en énergie dans les réseaux de capteurs sans fil est présenté par Manfredi et al. [64], où ils proposent une méthode distribuée de la radio de transmission des nœuds dans les réseaux de capteurs sans fil avec une économie d'énergie pour faire face simultanément à la consommation d'énergie et à l'exigence de réactivité du consensus.

Le consensus a été toujours présent pour résoudre des problèmes moraux ou physiques. Le consensus est le processus d'une satisfaction sur l'une des nombreuses alternatives proposées par un certain nombre de sources. Ici dans la thèse, nous nous limiterons aux cas des systèmes distribués où les sources sont un ensemble de nœuds de capteurs coopérant pour parvenir à un consensus sur une seule valeur proposée. La conception des méthodes autour d'un consensus entre dans le système du calcul distribué.

En informatique et dans les systèmes distribués, Paxos est un ensemble de protocoles mis en place pour résoudre le consensus dans un réseau de nœuds pouvant être défaillants. Le consensus défini dans Paxos est l'accord mutuel des différents nœuds sur un résultat à valider et le transmettre. Cette démarche reste une tâche déterminante avec des nœuds susceptibles d'être défectueux. D'ailleurs, Ping Xu et al. [65] proposent une classe d'algorithmes distribués déclenchés par des événements résolvant le problème de consensus de la moyenne dans les systèmes multi-agents. En concevant des événements tels que qu'une fonction de Lyapunov spécifiquement choisie soit monotone décroissante, les algorithmes déclenchés par des événements réussissent à réduire les communications entre les agents tout en assurant que le système entier converge vers l'état désiré.

Pour réduire la consommation d'énergie et le nombre de données, Zhenyu Lu et al. [66] proposent une méthode d'optimisation de la topologie et une méthode d'identification itérative des paramètres pour estimer les facteurs communs du modèle dans les RCSF. La première méthode optimise la topologie décentralisée, de telle sorte que tous les nœuds feuilles d'une communauté se connectent directement au nœud de tête. Une topologie circulaire est construite pour permettre aux nœuds feuilles distants de se connecter au nœud de tête par l'intermédiaire de deux nœuds relais adjacents, afin de réduire la distance de communication totale et la consommation d'énergie. En prenant compte de la topologie optimisée, une méthode d'identification itérative est proposée pour minimiser la capacité d'information en transmettant les résultats traités au lieu des données brutes, afin d'optimiser le traitement d'une masse de données importante et son stockage.

a) Consensus de la moyenne

Les nœuds dans un réseau de capteurs sans fil ont une énergie limitée avec faible capacité de calcul. Leur défi est d'arriver à un consensus sur la valeur détectée, non seulement, en un minimum de message de communication, mais aussi en un temps optimal. Ceci résumera la question : « pourquoi ces algorithmes ont une complexité faible ? ».

Plusieurs algorithmes de coopération de détection se sont basés sur l'accord de moyenne. La littérature nous a éclairés sur deux types d'algorithmes : les algorithmes synchrones et les algorithmes asynchrones. En 2006, les auteurs dans [67] ont étudié la théorie du consensus de moyenne en optimisant le nombre d'itérations sur un graphe complet.

Ils énoncent : « *L'algorithme distribué de consensus est une règle d'interaction qui spécifie l'échange d'informations entre un capteur et ses voisins pour parvenir à un accord commun au sujet d'une fonction des mesures. Dans cet exemple, il s'agit de la moyenne des mesures* ».

La méthode de consensus de moyenne [68] est une suite d'itérations synchrones où un ensemble de capteurs transmettent régulièrement leurs valeurs dans un but de faire une moyenne des valeurs relevées avec les valeurs de leurs voisins directs.

De par sa simplicité, l'algorithme du consensus de moyenne est le plus souvent usité dans les réseaux de capteurs. Par exemple, les auteurs dans [69] et [70] examinent ce consensus dans sa forme la plus simpliste, totalement synchrone et en temps discret (voir equation1), en prenant en considération toutes les valeurs des nœuds voisins.

$$x_i(k+1) = x_i(k) + \varepsilon \sum_{j \in N_i} a_{ij} (x_j(k) - x_i(k)) \quad (10)$$

Avec :

- $x_i(k)$ désigne l'état du nœud i à l'instant k .
- N_i définit le voisinage, c'est-à-dire les nœuds qui sont reçus par le nœud i .
- ε un paramètre de mélange.
- a_{ij} de la matrice d'adjacence « M » est défini à « 1 » si le nœud i et le nœud j sont voisins. Dans le cas contraire, a_{ij} est égale à 0.
- a_{ii} les valeurs de la diagonale de « M » sont égales à 1.

La simulation a montré que le consensus converge rapidement, c'est-à-dire tous les nœuds ont acquis la même valeur de la moyenne en prenant un paramètre de mélange, appelé par la suite le seuil : $\varepsilon = 1/N$ (avec N taille du réseau, ou aussi, le nombre de récurrence afin d'arriver à une certaine qualité. Le consensus serait le même quelle que soit la taille N du réseau). O. Hlinka et al., dans leurs travaux [71] et [72], exploitent l'estimation d'état distribué dans un réseau de capteurs sans fil, plus vaste, en éliminant le puits, où chaque capteur effectue une tâche d'estimation globale, basée sur les ancienne mesures et actuelles de tous les capteurs, en utilisant uniquement un traitement local et des communications locales avec ses voisins. Cette estimation est réalisée grâce à la fonction de vraisemblance conjointe (tous capteurs). L'auteur dans son étude [73] s'intéresse de près au consensus de moyenne dans un réseau de capteurs. Il annonce que tous les algorithmes précédant sa présente recherche ne garantissaient pas une convergence totale mais plutôt asymptotique. De ce fait, il propose deux nouvelles méthodes de synthèse des matrices, afin d'obtenir le consensus de la moyenne en un nombre fini d'opérations.

Le consensus basé sur moyenne (Figure 18) a été beaucoup utilisé et appliqué dans divers champs de recherches, pour collecter, évaluer et traiter les informations, afin de pouvoir faire des recommandations dans des problèmes de décisions et/ou d'évaluations [74].

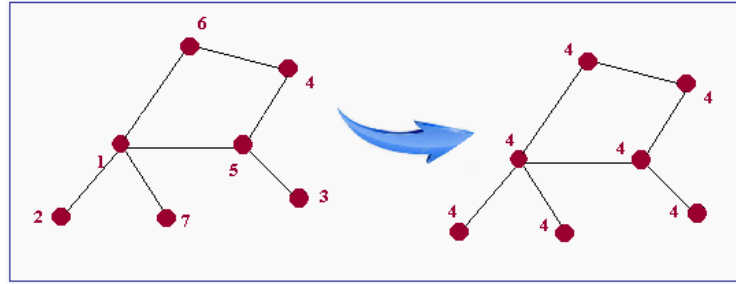


Figure 18. Consensus de la moyenne

Dans le cadre de notre travail, l'échange entre les nœuds de capteurs sera concrétisé par un ensemble de valeurs discrètes. Le travail s'est basé sur le choix des nœuds et le consensus de la moyenne. Plus de détail est explicité dans le prochain chapitre.

b) Consensus des hypothèses

Un cas particulier du consensus de la moyenne est le test d'hypothèse initié par Reza Olfati-Saber et al. [75]. Les auteurs mettent en exergue un algorithme heuristique permettant une classification probabiliste bayésienne d'un accord commun entre plusieurs capteurs distribués.

$$\Pr [h|z_1, z_2, \dots, z_N] = \frac{\Pr [h] \Pr [z_1, z_2, \dots, z_N|h]}{\Pr [z_1, z_2, \dots, z_N]} \quad (11)$$

$$= \frac{\Pr [h] \prod_{i=1}^N \Pr [z_i|h]}{\Pr [z_1, z_2, \dots, z_N]} \quad (12)$$

h : Une réalisation de l'espace des hypothèses H .

$\mathbf{Z} = \{z_1, z_2, \dots, z_N\}$: l'échantillon de valeurs captées des N capteurs.

Chaque probabilité de la règle de Bayes (Equation 11) à un rôle distinct :

Probabilité a priori $\Pr [h]$: indique la probabilité d'expertise.

Estimation de vraisemblance $\Pr[z_i|h]$: degré de similitude entre les données et l'hypothèse.

Probabilité a posteriori $\Pr[h|\mathbf{Z}]$: nouvelle hypothèse accompli comme des données z_i .

Bien que $\Pr[\mathbf{Z}]$ est indépendant, on peut le confondre à une constante afin de normaliser les probabilités résultantes. Ainsi, chaque capteur détient une hypothèse propre sur la probabilité de la réalisation « h » répartie sur les autres nœuds selon l'équation (11). Une estimation de vraisemblance (Equation 13) d'un capteur « i » dans le réseau, est alors énoncée:

$$\begin{aligned} \log \left(\prod_{i=1}^N \Pr [z_1, z_2, \dots, z_N|h] \right) &= \sum_{i=1}^N \log (\Pr [z_1, z_2, \dots, z_N|h]) \\ &= \sum_{i=1}^N \log (\Pr [z_i|h]) \end{aligned} \quad (13)$$

De ce fait, le programme du consensus de moyenne pourra estimer la moyenne des maximums de vraisemblance de l'ensemble des nœuds de capteurs « i » [75].

IV.2.2.4. Algorithmes de conversations entre voisins

Le consensus de la moyenne, bien qu'il est adapté et optimise l'énergie des capteurs, il risque comme même de mener une fausse information au puits. Ceci peut arriver visiblement, si un ou un ensemble de capteur(s) est/sont défaillant(s).

Dans cette partie, nous allons étaler quelques algorithmes de conversations locales pour trouver un accord commun sur la valeur à transmettre au nœud agrégateur dans un système distribué ou au puits dans un système centralisé. Deux modes de communications peuvent être observés, soit que les capteurs s'échangent leurs valeurs en même temps, et on parle alors d'une conversation synchrone, soit que l'échange des données est en différée, c'est-à dire que le capteur voisin est en mode veille, et dès qu'il se réveille, il reprend la conversation, et on parle alors d'une communication asynchrone. Cette dernière est la plus utilisée dans les réseaux de capteurs du fait qu'elle perdure la durée de vie du réseau en agissant sur le point de vue énergétique.

a) Conversation entre paire de voisins

Les algorithmes de conversations intermittentes ont suscité beaucoup d'intérêt en raison de leur capacité à calculer les statistiques globales en utilisant des communications locales entre deux nœuds voisins. Cependant, ces algorithmes tendent vers des valeurs statistiques globales après un certain nombre de communications. Seulement, il est presque impossible de prédire le temps de convergence pour les réseaux à grande échelle [76].

« La conversation par paire de voisins » est une forme de consensus permettant d'évaluer la moyenne globale des valeurs initiales des variables de comméragé. Le processus de comméragé peut être modélisé comme un système linéaire en temps discret [77] sous la forme suivante :

$$\mathbf{x}(t + 1) = \mathbf{M}(t) \cdot \mathbf{x}(t) \quad \text{avec } t = 1, 2, 3, \dots \quad (14)$$

Avec :

x : un vecteur de variables de nœud.

$M(t)$: matrice doublement stochastique. Si les nœuds i et j communiquent entre eux au moment t , alors les valeurs des nœuds au temps $(t + 1)$ seront mises à jour comme suit :

$$x_i(t + 1) = x_j(t + 1) = \frac{x_i(t) + x_j(t)}{2} \quad (15)$$

$M(t)$ est exprimé par $M(t)=P_{i,j}$, où $P_{ij} = [P_{lm}]_{n \times n}$ pour chaque étape (i, j) avec des entrées définies comme suit :

$$P_{lm} = \begin{cases} 1/2 & (l,m) \in (i,j); (j,i) \\ 1 & l=m \\ 0 & \text{sinon} \end{cases} \quad (16)$$

A chaque top d'horloge et aléatoirement, un capteur « i » estime un de ses voisins j avec une probabilité « P_{ij} ». L'algorithme suivant [78] montre la stratégie de comméragé par paire de voisins :

Algorithme_1 : Conversation par paire

1. **Tant que le consensus de la moyenne n'est pas atteint faire**
2. Activer au hasard un capteur « i » et favoriser un voisin j parmi ses voisins « N_i » (en utilisant la matrice de pondération « $W(k)$ »)
3. Envoi de la valeur du capteur j au capteur i
4. Mettre à jour les valeurs des deux capteurs i et j , par leur moyenne à l'étape précédente:

$$x_i(k) \leftarrow \frac{x_i(k-1) + x_j(k-1)}{2} ; \quad x_j(k) \leftarrow \frac{x_i(k-1) + x_j(k-1)}{2}$$

Fin.

Un exemple est donné par la figure 19 montrant comment mettre à jour la valeur des capteurs, selon l'algorithme de conversation entre deux voisins, dans un réseau composé de cinq capteurs déployés au hasard, et en faisant intervenir la matrice de pondération $W(k)$ qui joue un rôle important dans la convergence de l'algorithme_1 vers le consensus de la moyenne.

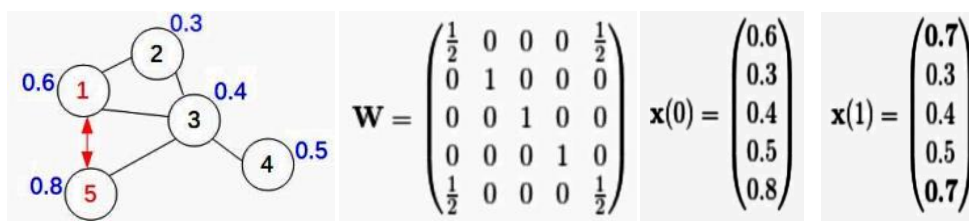


Figure 19. Exemple de conversation entre deux voisins [79]

b) Conversation par inondation partielle

La conversation par inondation peut être considérée une forme de diffusion par broadcast sur l'ensemble des voisins. En effet, l'algorithme s'appuie sur l'envoi aléatoire d'un nœud « i » choisi au hasard et communiquant sa valeur à l'ensemble de ces voisins « j » [79].

Une pondération moyenne est calculée à chaque capteur « j » dès la réception de la valeur de « i », selon l'équation suivante :

$$x_j(k) = \varepsilon \cdot x_j(k-1) + (1 - \varepsilon) \cdot x_i(k-1) \quad (17)$$

Avec : ε : Paramètre de mélange. $\mu \in (0, 1)$

De ce fait, le capteur « i » et l'ensemble des capteurs non-voisins « nv » maintiennent leurs valeurs de l'étape précédente :

$$x_{nv}(k) = x_{nv}(k-1) \text{ et } x_i(k) = x_i(k-1) \quad (18)$$

Le calcul de la valeur moyenne des capteurs est représenté dans un vecteur $x(0)$, et les probabilités, à chaque étape k , sont représentées dans un vecteur $x(k)$. Ces probabilités sont actualisées d'une manière linéaire en appliquant l'équation de pondération :

$$x(k) = W(k) \cdot x(k-1) \quad (19)$$

Il est à noter que $W(k)$ nous permet de profiter de l'activité asynchrone qu'est dans les réseaux de capteurs. Les matrices $W(k)$, bien qu'elles sont aléatoires et proches de l'identité (capteur-dormant ; capteur-éveillé), elles diffèrent d'un algorithme à un autre. En incluant les états précédents, on obtient :

$$x(k) = W(k) \cdot W(k-1) \cdot \dots \cdot W(1) \cdot x(0) = P(k) \cdot x(0) \quad (20)$$

Avec $P(k) = \text{Produit } W(k) \cdot \dots \cdot W(1)$

En effet, la mise à jour des valeurs du vecteur $x(k)$ suit le processus linéaire de l'équation (19). La matrice de pondération de mise à jour $W(k)$ est formulée par :

$$W_{jk}^{(i)} = \begin{cases} 1 & j \notin Ni, k = j \\ \varepsilon & j \in Ni, k = j \\ 1 - \varepsilon & j \in Ni, k = i \\ 0 & \text{sinon} \end{cases} \quad (21)$$

Ici, la production de l'information est du ressort du nœud de capteur « i » (Figure 20) qui s'éveille à chaque itération k .

Algorithme_2 : Conversation par inondation partielle

1. **Tant que le consensus n'est pas atteint faire**
2. Activer un capteur « i » aléatoirement
3. Diffuser la valeur de « $x_i(k-1)$ » à l'ensemble de ses voisins « j »
4. Calculer : $x_j(k) \leftarrow \varepsilon \cdot x_j(k-1) + (1 - \varepsilon) \cdot x_i(k-1)$
5. Mémoriser la nouvelle valeur « $x_j(k)$ » dans chaque capteurs j ,

Fin.

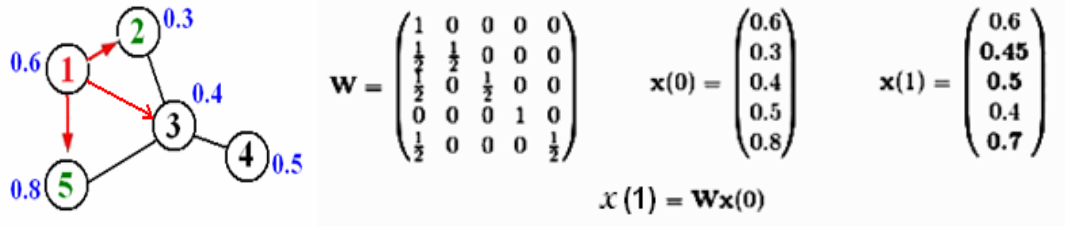


Figure 20. Exemple de conversation par inondation partielle [80]

c) Conversation par triplet

Au lieu d'une conversation d'une paire de voisin, les auteurs de [80] estiment que le résultat de la moyenne d'une conversation dans ce contexte (Triple points) améliore nettement la conversation par paire (Figure 21).

Algorithm_3 : Conversation par Triplet

1. **Tant que** l'accord sur la moyenne n'est pas obtenu **faire**
2. Activer au hasard un capteur « i »
3. Sélectionner aléatoirement deux voisins j_1, j_2
4. Récupérer les deux valeurs de « j_1 » et « j_2 » par le capteur « i ».
5. Calculer la moyenne et mise à jour des trois valeurs capteurs :
6.
$$x_i(k) \leftarrow \frac{x_i(k-1) + x_{j_1}(k-1) + x_{j_2}(k-1)}{3}; x_{j_1}(k) \leftarrow \frac{x_i(k-1) + x_{j_1}(k-1) + x_{j_2}(k-1)}{3};$$

$$x_{j_2}(k) \leftarrow \frac{x_i(k-1) + x_{j_1}(k-1) + x_{j_2}(k-1)}{3};$$

Fin.

$x_v(k) \leftarrow x_v(k-1)$. Les autres valeurs des capteurs non choisies du réseau gardent leurs anciennes valeurs, et dans ce cas, la matrice de pondération de l'équation (19) :

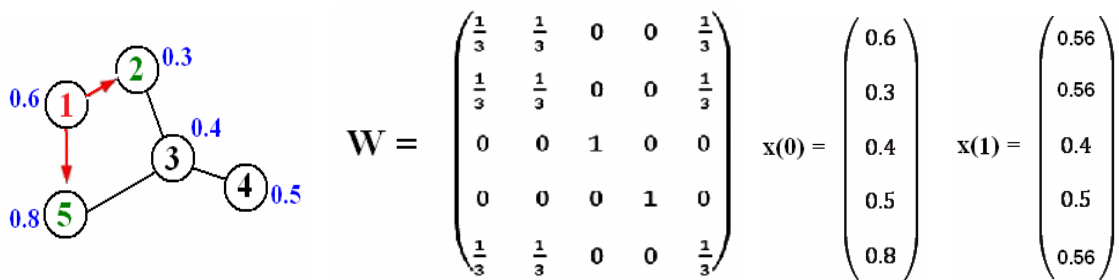


Figure 21. Exemple de conversation entre Triple points [80]

d) Conversation entre voisins-à-un-saut

Même conception d'algorithme que le précédent. Si « N_i » est le nombre de voisins alors la moyenne [81] serait :

$$x_i(k) \leftarrow \frac{1}{|N_i|} \sum_{j \in N_i(k)} x_j(k-1) \quad (22)$$

Algorithme_4 : Conversation entre voisins-à-un-saut

1. **Tant que** l'accord sur la moyenne n'est pas satisfait **faire**
2. Activer aléatoirement un capteur « i »
3. Diffuser un message au voisinage « Hello »
4. Manifestation de présence des capteurs « j » voisins par envoie du message: $\mathbf{m}_j = \{x_j(\mathbf{k} - 1)\}$ au capteur « i »
5. Mettre à jour la valeur du capteur « i » par : $x_i(\mathbf{k}) \leftarrow \frac{1}{|N_i|} \sum_{j \in N_i(\mathbf{k})} x_j(\mathbf{k} - 1)$
6. Diffusion de la valeur du capteur « i » au voisin par le message $\mathbf{m}_i = \{x_i(\mathbf{k}), N_i(\mathbf{k})\}$
7. Mise à jour des capteurs voisins à « i » par la valeur $\{x_i(\mathbf{k})\}$

Fin.

IV.2.3. Cadre de comparaison entre les méthodes de détection

Méthodes	Avantages	Inconvénients
Agrégation de la détection	- Fiable en présence de substitutions de capteurs défectueux.	- Ecart type 2.5 de la moyenne n'est pas un standard pour tous les types de grandeurs physique. - Calcul consomme de l'énergie.
Corrélation spatiale et temporelle de L. Myeong-Hyeon	- Identifie les capteurs qui sont en bon état.	- Erreur d'estimation de faux négatifs. - Surcharge : Communication, Energie.
Corrélation spatiale et temporelle de M. Ding	- Méthode Spéciale pour une architecture arborescente.	- Non applicable pour une autre architecture.
Consensus de la moyenne	- Synchronisation. - Absence de protocoles de routage. - Algorithmes itératifs de faible complexité.	- Suit un modèle de poisson de taux $N\lambda$. - Estimation de la pondération W .
Consensus des hypothèses	- Repose sur le modèle de classification de bayes.	- Complexité des calculs.
Conversation Par paire de voisins	- Simplicité des calculs. - évolutivité.	- Dépend de la topologie et de la pondération. - Consomme plus d'itérations pour converger.
Conversation entre Triple points	- Simplicité des calculs. - Diffusion contrôlée à 2 nœuds seulement. - Améliore la conversation par paire.	- Quatre messages par itération.
Conversation par inondation partielle	- Une seule communication pour la convergence. - Converge rapidement. - Evolutivité.	- Chaque nœud diffuse à tous les nœuds. (consomme les communications)
Conversation entre voisins-à-un-saut	- Performances élevées à celles de la conversation entre 2-voisins. - Evolutivité	- Nombre de messages à chaque itération. - Le coût de communication élevé.

Tableau 4. Avantages et Inconvénients des méthodes de détections

IV.3. Conclusion

Les algorithmes distribués de consensus synchrones et asynchrones réduisent, non seulement la consommation d'énergie, mais aussi nous aide à appréhender le rapport de ces derniers dans la détection. Ils sont simples et faciles à mettre en œuvre, et ne requièrent aucune connaissance au préalable sur le déploiement globale des nœuds du réseau. Les

communications et transferts de données se font de manière assez simple et sans l'apport de protocole de routage. Ils sont évolutifs quant au passage à l'échelle. Même les algorithmes de conversations dites (gossip en anglais) ne requièrent pas de changement ou d'action de récupération des données.

Le chapitre suivant explicitera notre intérêt à améliorer la méthode de conversation entre triple points.

Chapitre V.
Contribution
à la détection et coopération des capteurs

V.1. Introduction

Les recherches dans le domaine de la communication sans fil ont donné naissance à des équipements miniaturisés et intelligents comportant des unités de capture, de calcul et de communication sans fil dans un seul circuit, de dimension réduite, et à un coût raisonnable. Ces équipements, communément appelés micro-capteurs (Figure 22), ont favorisé l'idée de développer des réseaux de capteurs basés sur la collaboration d'un grand nombre de nœuds fonctionnant de manière autonome et communiquant entre eux par des transmissions à courte portée dans le but de signaler un événement selon le domaine d'application.



Figure 22. Plate-forme de nœuds de capteurs TMote Sky

Un capteur est capable de récolter des données relatives à son environnement, de les traiter et, si nécessaire, de les communiquer aux capteurs voisins via un support sans fil. Le déploiement d'un grand nombre de ce type de dispositif forme alors un réseau de capteurs sans fil qui peut être utilisé dans des domaines variés : scientifiques, militaires, industriels, santé, environnement, etc. Ce type de réseau est constitué de quelques centaines voire de milliers de capteurs autonomes, de taille extrêmement réduite, et qui fonctionnent avec des ressources très limitées en termes de mémoire, d'énergie et de capacités de traitement, et communiquent sans fil avec les autres dans leurs zones. La conception et la mise en œuvre du RCSF sont influencées par plusieurs contraintes, parmi lesquelles nous citons [81] :

1. l'environnement,
2. déploiement des nœuds,
3. hétérogénéité des nœuds,
4. tolérance aux pannes,
5. la mobilité,
6. mise à l'échelle,
7. contraintes matérielles,
8. la transmission,
9. la consommation d'énergie, et
10. qualité du service.

Dans un RCSF, la couverture totale de la zone à surveiller est l'une des questions les plus posées dans la recherche. Ainsi, en raison de la forte corrélation spatiale de la zone de détection, il y a un déploiement intensif de nœuds de capteurs qui détectent et relaient souvent

des informations similaires ou redondantes, et entraînant une analyse plus complexe des données et une perte importante de ressources pour les capturer, les traiter et les mémoriser, et par conséquent, une surconsommation des nœuds en énergie de transmission. Parmi les solutions à ce problème, la conception d'une topologie hiérarchique a été mise en œuvre en divisant les nœuds en plusieurs niveaux de responsabilité. L'une des méthodes les plus utilisées est le clustering, où le réseau est partitionné en groupes appelés "clusters". Un cluster est composé d'une tête de cluster et de ses membres.

V.2. Fil conducteur de notre contribution

Notre contribution est répartie en trois parties :

Partie I :

Dans la première partie, une compréhension du déploiement, et de la communication entre capteurs voisins et le puits de données est nécessaire. Dans ce sens, nous avons choisi des plans de simulations où nous avons fait varier la position d'un nœud déclencheur d'évènement dans cinq (05) zones différentes pour observer le comportement des nœuds de capteurs et relever le temps moyen de transmission des données au puits ainsi que la quantité d'énergie consommée de chaque capteur ayant participé à la transaction.

Partie II :

La seconde partie est catégorisée par l'élaboration d'un algorithme afin d'exploiter notre version sur consensus de la moyenne avec 3-voisins. L'idée, dans ce contexte, se base sur une architecture hiérarchique, en considérant le capteur élu pour le consensus comme cluster-Head qui demande les valeurs captées des voisins. Un plan témoin est réalisé dans le but de comparer notre méthode aux résultats standards du simulateur.

1. Plan-Témoin, appelé « Détection sans coopération », est notre point de comparaison où nous laissons le simulateur « Cooja » prendre l'initiative avec son propre protocole RPL (Routing Protocol for Low-power and Lossy Networks). Aucune intervention de notre part n'est accompli.
2. Algorithme proposé « Détection et coopération de 3-voisins ». Un algorithme de faible complexité pour minimiser la consommation d'énergie en traitement d'une part et organiser l'appartenance des nœuds à ses agrégateurs d'autre part. Il se base sur l'hypothèse suivante :

Hypothèse 1

« Si, au moins la valeur d'un capteur voisin est proche de la valeur captée d'un nœud « *i* » alors la moyenne de ces deux valeurs voisines sera la valeur-élue qui sera envoyée au puits ».

Partie III:

La troisième partie est réservée à une expérimentation avec des données réelles issues du laboratoire d'Intel Berkley. Une conception et une réalisation d'une méthode de classification des données captées avant d'envoyer la valeur captée au puits. L'algorithme ainsi élaboré fait abstraction des données collectées en communiquant seulement des informations implicites sur l'environnement. En se basant sur une discrétisation des données collectées ainsi que sur des statistiques. Cette dernière étude a une perspective assez large, car elle permettra aussi de détecter les fausses alertes ou des valeurs erronées natives de capteurs défectueux.

Le but de notre travail est d'étudier la détection et la coopération des capteurs et leur impact sur le déploiement. Dans notre travail, on s'est basé sur un déploiement déterministe, où nous avons mixé l'agrégation avec un système de consensus. Ce choix contribue pleinement à la réduction de la dimension des valeurs capturées, l'espace de stockage, la charge de transmission, l'énergie consommée et simplifie le processus d'exploration de données en cours, ce qui impliquera forcément une optimisation du nombre de capteurs et un déploiement réfléchi et judicieux dans un réseau de capteurs sans fil.

V.3. Simulateur Contiki/Cooja

Contiki OS Java Simulator est un émulateur de réseau de capteurs qui peut supporter un programme compilé afin de tester un plan avant de le charger dans la mémoire flash d'un nœud réel, comme dans la plateforme TI-MSP430¹.

V.3.1. Protocole de routage - RPL

Les principes de base du protocole sont [82, 83] :

1. Routage : la découverte du chemin est basée sur la qualité du signal, la densité des nœuds et le chemin sélectionné est quantifié ;
2. Sélection de la route : une route fiable repose sur la force du signal (RSSI) ;
3. Critères de qualité de la route : force du signal (RSSI), nombre de sauts, énergie disponible et taux d'erreur (aberrations) ;
4. Modification du chemin : le chemin initialement tracé peut être corrigé ultérieurement en fonction de la force du signal et du nombre total de nœuds visités en cas de défaillance de la route principale ; et

¹ https://fr.wikipedia.org/wiki/MSP430#/media/Fichier:MSP430FG461x_EVK.jpg

5. Routage basé sur les contraintes : la durée de vie d'un nœud dépend de la batterie et de la diversité des types de nœuds et de leurs caractéristiques (mémoire, emplacement, collecteur, etc.).

V.3.2. Force du signal (RSSI)

RSSI est un indicateur de l'intensité du signal capté Rx. C'est un paramètre essentiel pour les applications de radiolocalisation, en raison de sa simplicité et de son faible coût. Mais sa précision peut être altérée par le bruit causé par des objets, des obstacles stables, mobiles ou par la diffraction ou la réfraction des ondes radio [84].

$$RSSI = - (10 * n * \log d + A) \quad (23)$$

Avec :

- : Constante - propagation du signal.
- : Distance entre l'émetteur Tx et le récepteur Rx.
- : Intensité du signal reçu pour une distance de 1 mètre.

Notant qu'un réseau de capteurs est efficace si et seulement si les liens de communication sont fiables. Alrawashdeh [84] souligne et expérimente cette théorie sur les applications corporelles incorporés dans l'os, en appliquant une liaison standard avec une modulation par déplacement de phase (PSK).

Dans la seconde partie de notre travail, une méthode de détection est proposée afin de capturer des quantités physiques et de les transmettre à la station de base (Sink) dans un temps très court, basé sur le temps de traitement que chaque nœud peut effectuer selon un plan. Notre système est basé sur les trois tâches ordonnées suivantes :

1. Détecter les quantités physiques qui sont des valeurs entières de température et d'humidité prises au hasard, et qui changent toutes les 30 secondes.
2. Détecter les nœuds de capteurs voisins du nœud de capteur qui veut transmettre ces quantités.
3. Utiliser la topologie de routage DODAG construite par le protocole RPL.

DODAG (Destination Oriented Directed Acyclic Graph) : un graphe acyclique dirigé vers une destination qui est la racine du réseau (Sink). Il est à noter que tout au long de notre simulation, nous utiliserons la formule de calcul du RSSI donnée par l'équation (23), afin de nous assurer que le signal est suffisant pour garantir une connexion sans fil fiable. Cette formule a été testée en extérieur et donne des résultats satisfaisants et acceptables [85].

$$d = (- 0.5894 * RSSI) - (5.6768) \quad (24)$$

V.4. Expérimentation

V.4.1. Partie I : L'effet du nombre de capteurs sur d'un déploiement déterministe

V.4.1.1. Introduction

Comme déjà mentionné ci-dessus, cette partie est réservée à une étude du comportement d'un réseau de capteur sans fil sous le simulateur Cooja/contiki en utilisant une multitude de plan, dans le but de noter et de comparer des informations relatives au temps et à la consommation d'énergie par rapport à un déploiement déterministe.

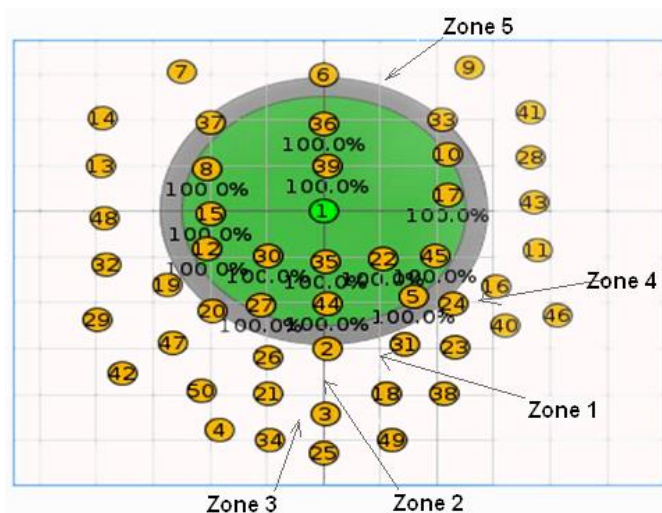


Figure 23. Déploiement 1 - 50 capteurs -

Note : Le choix des zones n'est pas au hasard, car ces zones représentent des positions critiques d'un plan architectural d'une résidence de personnes âgées.

Paramètre de simulation :

• Simulateur	Cooja
• Puits	01
• Capteurs relais	50, 24, 17
• Type de Nœuds	Sky Mote
• Couverture en transmission	25 m
• Couverture en réception	29 m

Tableau 5. Paramètres de simulation

Une série de simulations regroupées en 03 déploiements basés sur le nombre de nœuds.

Pour chaque déploiement cité ci-dessus, nous avons fait varier la position d'un nœud déclencheur d'évènement (Portant le N°51 dans le déploiement 1, le N°25 dans le déploiement 2 et le N°18 dans le déploiement 3) dans 5 zones différentes pour observer le comportement du réseau.

V.4.1.2. Simulation faite pour 50, 24, 17 capteurs

— Expérimentation : Déploiement 1 = 50 capteurs.

Plan 1 : Evènement déclencheur en zone1

Dans ce plan (Figure 24), nous avons positionné au choix 50 capteurs simples (couleur jaune), et capteur déclencheur d'évènement n°51 (couleur violet), le puits n°1 (couleur verte).

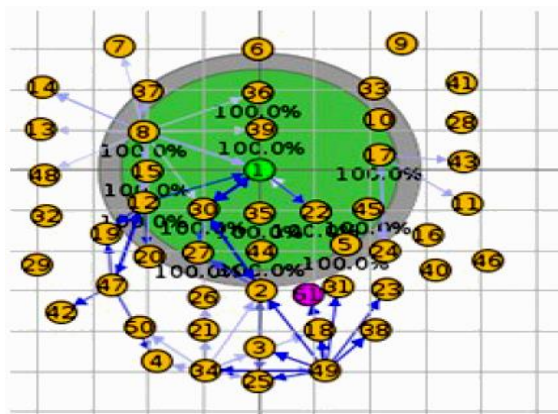


Figure 24. Déploiement 1, Plan 1 : Alarme en zone 1

Un résumé de consolidation des résultats du plan 1 est donné par le tableau 6 suivant :

Déploiement 1 : 50 nœuds	Zone 1
Moyenne Energie d'un nœud (mw)	2.44
Temps moyen de réponse au Puits (secondes)	5.24

Tableau 6. Résultat obtenu du plan1

Plan 2 : Evènement déclencheur en zone 2.

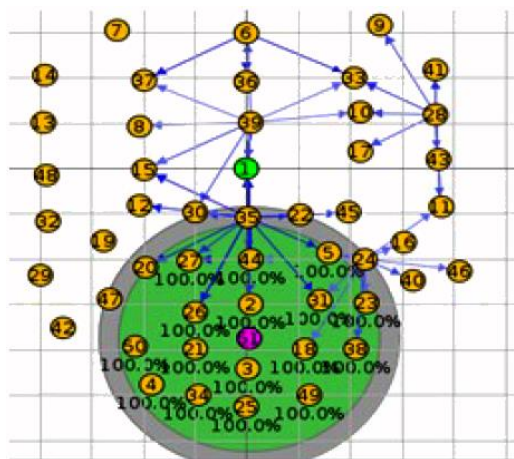


Figure 25. Déploiement 1, Plan 2 : Alarme en zone 2

Le tableau 7 résume le résultat issu du plan 2 :

Déploiement 1 : 50 nœuds	Zone 2
Moyenne Energie d'un nœud (mw)	2.16
Temps moyen de réponse au Puits (secondes)	1.69 s

Tableau 7. Résultat obtenu du plan 2

Plan 3 : Evènement déclencheur en zone 3

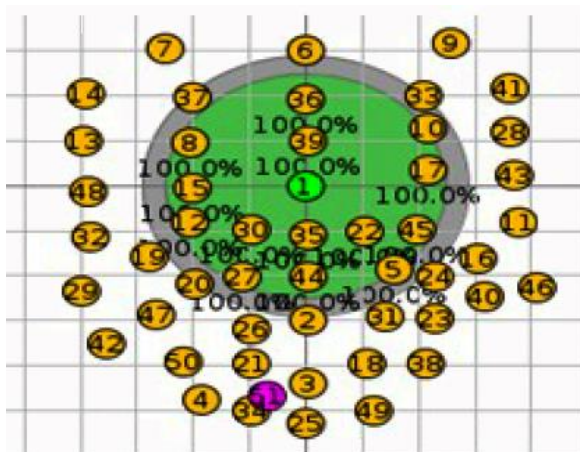


Figure 26. Déploiement 1, Plan 3 : Alarme en zone 3

Le tableau 8 résume le résultat issu du plan 3 :

Déploiement 1: 50 nœuds	Zone 3
Moyenne Energie d'un nœud (mw)	2.80
Temps moyen de réponse au Puits (secondes)	4.14 s

Tableau 8. Résultat obtenu du plan 3

Plan 4 : Evènement déclencheur en zone 4

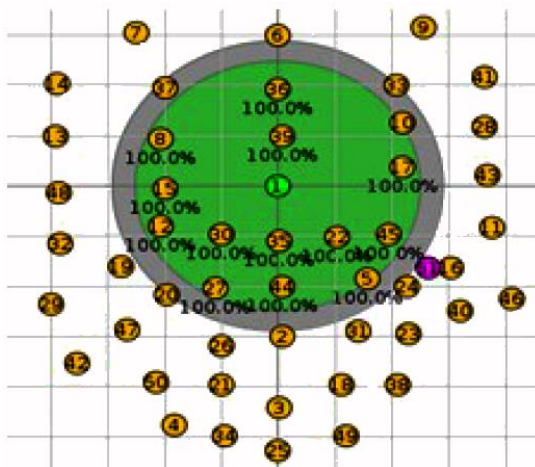


Figure 27. Déploiement 1, Plan 4 : Alarme en zone 4

Après un temps de convergence, nous obtenons le tableau 9 résumant le résultat du plan 4 dans la zone 4:

Déploiement 1	Zone 4
Moyenne Energie d'un nœud (mw)	1.63
Moyenne de temps réponse (s) d'un nœud	5.41 s

Tableau 9. Résultat obtenu du 1er déploiement dans la Zone 4

Plan 5 : Evènement déclencheur en zone 5

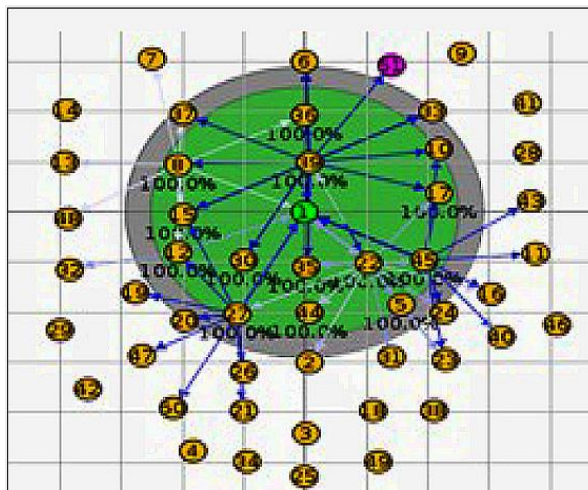


Figure 28. Déploiement 1, Plan 5 : Alarme en zone 5

Le résultat du plan 5 issu de la zone 5 est donné par le tableau 10 suivant :

Déploiement 1	Zone 5
Moyenne Energie d'un nœud (mw)	1.93
Moyenne de temps réponse (s) d'un nœud	13.13 s

Tableau 10. Résultat obtenu du 1er déploiement dans la Zone 5

Le tableau 11, résume l'expérimentation de tous les plans (de 1 à 5) du déploiement_1:

Type de déploiement	Plan	Energie consommée d'un nœud (mw)	Temps de réponse (s)
Déploiement 1 Avec 50 capteurs	1	2.44	5.24
	2	2.16	1.69
	3	2.8	4.14
	4	1.63	5.41
	5	1.93	13.03

Tableau 11. Récapitulatif des résultats obtenus des 5 plans du déploiement 1

— Expérimentation : Déploiement 2 = 25 capteurs.

La même expérience est réalisée pour le second déploiement, 24 capteurs au total répartie sur les cinq topologies. Le capteur n°25 va être utilisé comme capteur déclencheur d'évènement afin de suivre le comportement du réseau réduis en moitié.

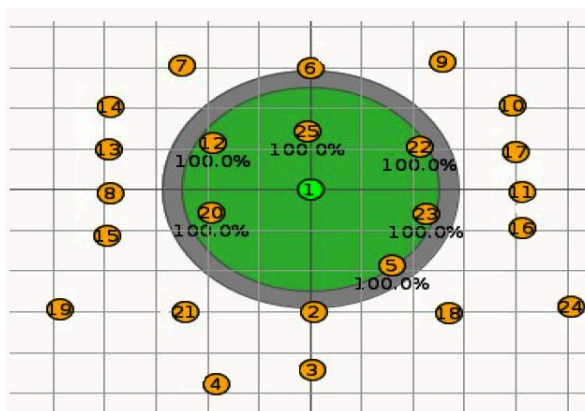


Figure 29. Déploiement 2 - 25 capteurs –

Après exécution des 5 plans sur le déploiement 2, nous résumons les résultats dans le tableau 12 suivant :

Type de déploiement	Zone	Energie consommée d'un nœud (mw)	Temps de réponse (s)
Déploiement-2 Avec 25 capteurs	1	1.85	4.40
	2	2.01	6.54
	3	1.58	4.78
	4	1.60	5.16
	5	1.57	4.24

Tableau 12. Récapitulatif des résultats obtenus des 5 plans du déploiement 2

— Expérimentation : Déploiement-3 = 17 capteurs.

La même expérience est réalisée, donnant ainsi le déploiement-3, 17 capteurs au total répartie sur les cinq plans de zones. Un capteur va être ajouté et utilisé comme capteur déclencheur d'évènement afin de suivre le comportement du réseau ainsi réduis (voir Figure 30) :



Figure 30. Déploiement_3 - 17 capteurs

Type de déploiement	Zone	Energie consommée d'un nœud (mw)	Temps de réponse (s)
Déploiement-3 Avec 17 capteurs	1	1.66	4.11
	2	1.61	6.26
	3	1.75	6.89
	4	1.69	3.87
	5	1.51	2.95

Tableau 13. Récapitulatif des résultats obtenus des 5 plans du déploiement-3

V.4.1.3. Interprétation des résultats pour 50, 25, 17 capteurs

En prenant les trois types de déploiements, on trace un histogramme pour observer l'impact du nombre de capteurs sur le temps de réponse et la consommation moyenne d'énergie qu'a été utilisée par le réseau. Il en ressort ce qui suit :

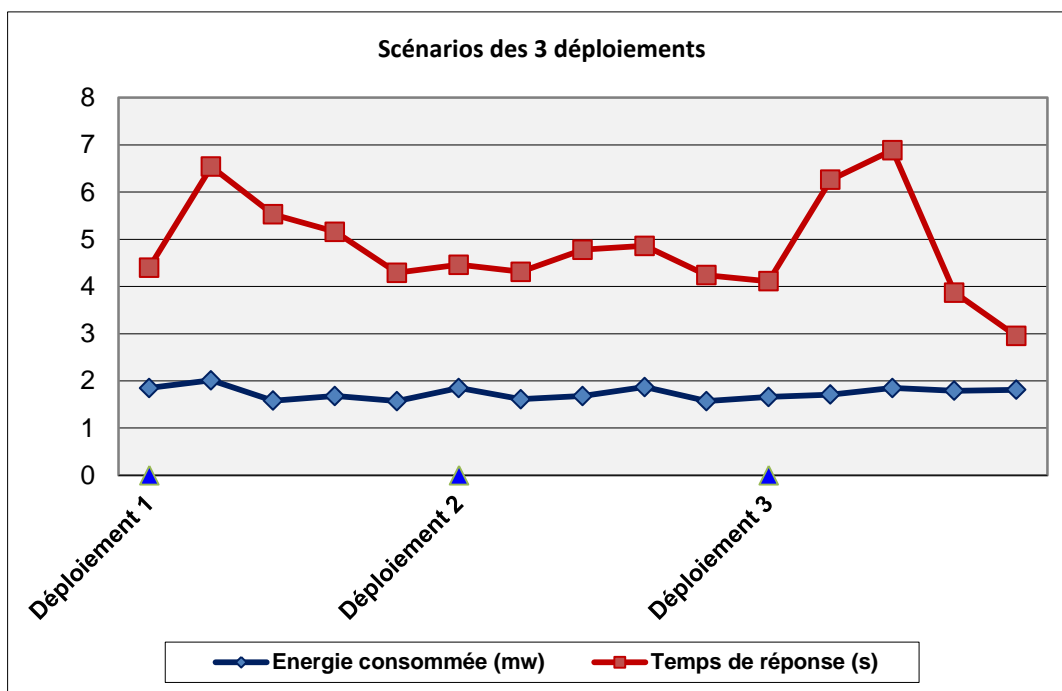


Figure 31. Représentation graphique des trois déploiements

On observant le graphe de la figure 31, on constate que l'énergie consommée reste au la limite inférieure des 2mw dans l'ensemble des trois déploiements, mais le temps de réponse augmente beaucoup plus dans le déploiement-1 (avec 50 capteurs), et le déploiement-3 (avec 17 capteurs), par contre le déploiement-2 (avec 24 capteurs) reste de près le meilleur résultat. Ainsi, l'expérience nous a montré que le déploiement d'un nombre assez important de capteurs peut coûter en temps de réponse et un peu en consommation d'énergie. Cela nous conduit à penser et tirer l'hypothèse que le choix de sélection de chemin nécessite un temps non négligeable vu les interférences et les redondances d'informations.

Pour éclaircir notre point de vue sur le nombre de capteurs déployés dans le réseau, nous ajoutons une quatrième expérimentation (déploiement-4) afin de prouver que le nombre de capteurs influe énormément sur le temps de réponse du réseau. Nous diminuons légèrement ce nombre à 12 capteurs au total. Ce nombre est aléatoire, mais dans un autre sens, nous voulions s'éloigner des résultats du déploiement 2. Sur quoi, nous avons noté un résultat assez satisfaisant en temps de réponse et en consommation d'énergie. Ainsi, en se basant sur le résultat des transmissions du déploiement-3, nous avons jugé de retirer les capteurs les plus loin du centre de fusion sans pour autant ôter la couverture de l'espace à surveiller.

V.4.1.4. Expérimentation : 12 capteurs (Déploiement-4)

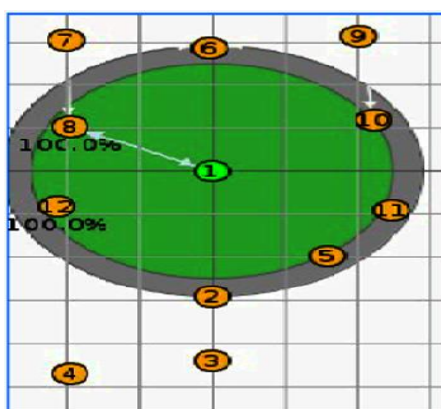


Figure 32. Déploiement 4 - 12 capteurs

Le tableau 14 suivant résume le résultat des plans des 5 zones où le capteur déclencheur d'évènement a été positionné.

Type de déploiement	Zone	Energie consommée d'un nœud (mw)	Temps de réponse (s)
Déploiement 4 Avec 12 capteurs	1	1,67	4,87
	2	1,79	6,18
	3	2,53	9,39
	4	1,89	5,22
	5	1,98	6,41

Tableau 14. Récapitulatif des résultats obtenus des plans du déploiement-4

V.4.1.5. Interprétation des résultats (pour 12 capteurs)

En dressant le graphe de la figure 33, nous observons une stabilité dans la consommation d'énergie, mais le temps de réponse souligne visiblement une augmentation par rapport au déploiement n°2.

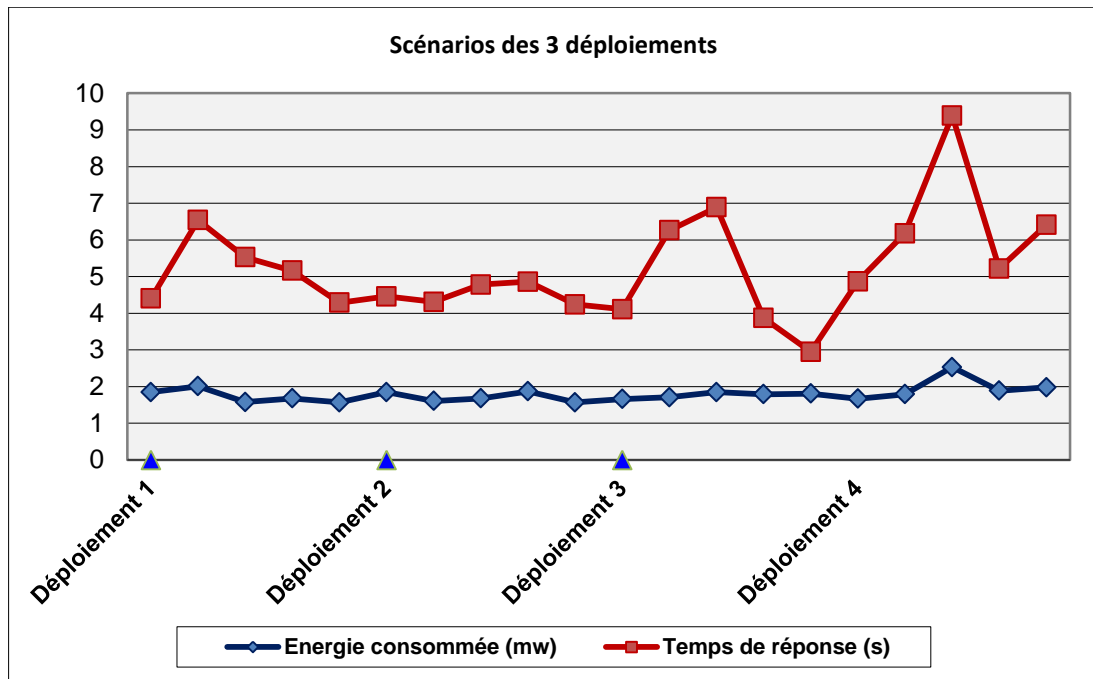


Figure 33. Représentation graphique des quatre déploiements

Cette courbe (Figure 33) est très expressive. Nous remarquons que le temps de transmission des données captées par le capteur n°13 responsable du déclenchement d'évènements au puits reste assez élevé. Ainsi, afin de mieux comprendre la corrélation existante entre un déploiement avec un nombre de capteurs assez réduit (12 capteurs) et le temps de convergence, nous allons explorer la trace des routes empruntées dans la zone 3 du déloiement-4 (voir Figure 34) :

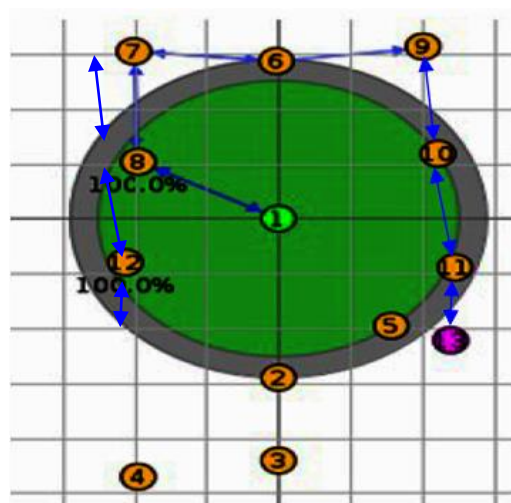


Figure 34. Déploiement 4 - 12 capteurs, avec le chemin emprunté vers le puits

De là, nous remarquons qu'effectivement, le choix du parent potentiel se fait par un calcul sur la qualité du signal (RSSI), mais un nœud mémorise aussi d'autres parents favoris en cas de défaillance d'un capteur participant à ce chemin.

V.4.1.6. Conclusion

Après l'étude de cette partie, nous pourrions dire que le nombre de capteurs influe beaucoup dans un réseau de capteurs sans fil, l'expérience nous a montré que :

- Plus on diminue du nombre de capteurs, plus on risque d'augmenter le temps de réponse au puits.
- Plus on augmente le nombre de capteurs, plus le temps de convergence et la consommation d'énergie s'amplifie.
- La diminution du nombre de capteurs peut influencer sur :
 - Le fonctionnement du réseau en cas de panne ou de défaillance d'un capteur.
 - La couverture globale du réseau.

Cette expérimentation était une entrée pour mieux connaître le phénomène de placement des nœuds et son impact sur le temps de réponse et la consommation d'énergie.

La section suivante expérimentera l'aspect coopération des capteurs et son impact sur le temps de réponse et la consommation de l'énergie du réseau.

V.4.2. Partie II - La détection et la coopération des capteurs – 1^{ère} approche –

Les capteurs sont intelligents et coopératifs. Ils surveillent un environnement et envoient leurs données à une station de base pour un traitement et éventuellement une prise de décision. Les données sont transmises par un simple saut, un multi-saut ou ils sont relayées par une passerelle. Les capteurs sont distribués et déployés en grand nombre offrant d'énormes possibilités. La coopération entre les nœuds d'un réseau augmentera les performances dans des tâches, telles que la détection, la localisation ou le suivi d'un phénomène afin de prévenir une inquiétude. La coopération est une méthode très utilisée dans la théorie des groupes ou des jeux, expliquant ainsi le comment et le pourquoi se forme un consensus entre les nœuds dans un réseau.

Dès lors, Hua et al. [86,87] ont affirmé que plusieurs nœuds peuvent coopérer pour rendre un réseau efficace et minimiser la consommation d'énergie en utilisant la nature de la diffusion du sans fil et le déploiement des nœuds de capteurs. Une telle stratégie est généralement appelée communication coopérative. Lorsque l'envoi des données se fait directement entre le nœud simple et la station de base, on parle d'une transmission directe. Dans ce cas, plusieurs nœuds voisins se mettent en état de veille afin d'économiser de

l'énergie. En choisissant un nœud pour le relais des données à partir d'un ensemble de candidats, la transmission coopérative permet de réduire considérablement l'énergie nécessaire pour une diffusion de données fiable [88]. En s'appuyant sur le RSSI, on pourrait par exemple augmenter la puissance du signal et diminuer le nombre de nœuds simples, ce qui influera sur leurs déploiements et réduira le nombre de sauts vers le puits. Le choix du chemin est basé sur ce dernier nombre, qui est un critère prépondérant dans la qualité de service d'une application réseau.

Dans ce qui suit, nous décrirons une nouvelle méthode de coopération hiérarchique utilisant un consensus de moyenne par triplet que nous allons comparer à un plan-témoin appelé méthode de détection standard de convergence utilisant le protocole RPL du simulateur Cooja, pour évaluer nos résultats.

La méthode proposée a été testée sur la plateforme CONTIKI/COOJA. Pour notre simulation, le mote (ou nœud) sélectionné est TmoteSky, une carte de capteurs sans fil de Moteiv [90]. Il s'agit d'une carte basée sur MSP430 avec une puce radio CC2420 compatible 802.15.4, une mémoire flash série externe de 1 mégaoctet et deux capteurs de lumière. Le port TmoteSky a été intégré au système Contiki en mars 2007. La fonctionnalité du module MSP430 F1611 est détaillée dans le guide de l'utilisateur de la série MSP430x1xx de Texas Instruments disponible à l'adresse [91].

Nous plaçons un nœud sink (nœud 1) à une position initiale fixe. Deux plans sont testés respectivement en déployant plusieurs nœuds simples-relais. Deux paramètres sont enregistrés : le premier est le temps de convergence de la valeur capturée au niveau du sink, et le second est la quantité d'énergie consommée. L'estimation de la consommation d'énergie des motes est basée sur le module 'Energest' de Contiki en évaluant certains états comme CPU, LPM, Tx et Rx, afin de calculer le temps écoulé à chaque nœud de capteur en temps réel. Ce module d'estimation d'énergie 'Energest' est appelé lorsque le composant est activé pour produire un horodatage. La consommation d'énergie des nœuds est calculée par l'équation suivante [92] :

$$E_c = (E_v \times E_p \times V_t) / R_{t_second} \times R_t \quad (25)$$

Avec :

E_c \equiv Consommation d'énergie (milli-watt),

E_v \equiv Valeur énergétique,

E_p \equiv Puissance électrique,

V_t \equiv Tension,

R_t \equiv $R_{timer} = 10$ secondes, et

R_{t_second} \equiv Fréquence du cristal basse fréquence = 32768 Hz.

Nous soulignons que la valeur d'Energest_Value est le reste de la différence entre le nombre de ticks (ALL_CPU) au temps actuel « t_2 » et au temps précédent « t_1 ». Un tick est un système unique correspond environ à 1 milliseconde.

$$eff_c (\%) = \frac{\Delta T_x + \Delta R_x}{CPU_e + LPMe} \quad (26)$$

Avec :

effc : cycle effectif de service (%), indique un taux d'un signal en service. (Duty-cycle)

ΔT_x : énergie de transmission,

ΔR_x : énergie de réception,

CPUe : énergie du processeur,

LPMe : énergie du mode de faible consommation, et

LPM : mode faible consommation, dispositif inactif en attente d'événements.

V.4.2.1. Méthode de détection standard basé sur le Protocol RPL

V.4.2.1.1. Introduction

Sur Contiki, la transmission standard est basée sur le principe que les nœuds peuvent transmettre les quantités physiques capturées au nœud voisin ou au nœud puits sans passer par un algorithme de sélection, de rumeur ou de consensus, mais elle utilise la topologie de routage DODAG (Destination Oriented Directed Acyclic Graph) construite par le protocole RPL, et qui est composée d'un ensemble de routes ou de chemins reliant les nœuds capteurs du réseau au nœud capteur racine (puits).

Le RPL (Routing Protocol for Low-power and Lossy Networks) est un protocole de vecteur de distance proactif : lorsqu'un parent préféré a été sélectionné, tout le trafic sera transféré via ce parent préféré, tant qu'il est accessible, sans aucune tentative d'équilibrage de charge entre les autres candidats parents disponibles. Cet avantage peut épuiser la puissance des parents surchargés, ce qui entraîne des problèmes de défaillance du réseau et de manque de fiabilité, car les nœuds surchargés sont susceptibles de périr assez rapidement. De nombreux articles expliquent le fonctionnement du protocole RPL, par exemple les auteurs José Sobral et al. [89] présentent plus en détail le protocole RPL dans les applications de l'Internet des objets.

V.4.2.1.2. Simulation

La topologie retenue pour le plan-témoin comprend 20 nœuds simple-relais et un Sink (mote1). Ces 20 nœuds peuvent détecter et transmettre les températures, l'humidité et les niveaux de lumière au sink sans aucun calcul et sans conditions requises.

Le cycle effectif (%) est calculé pour comparer la consommation d'énergie exprimée en milli-joule d'un nœud déployé à proximité du Sink (ici nœud 15), d'un autre déployé loin du Sink mais dans sa zone de couverture (mote 6) et d'un troisième nœud déployé en dehors de la zone de couverture du Sink (mote 7) après une période de 10 secondes et dans un intervalle de temps compris entre 0 et 200 secondes (figure 35).

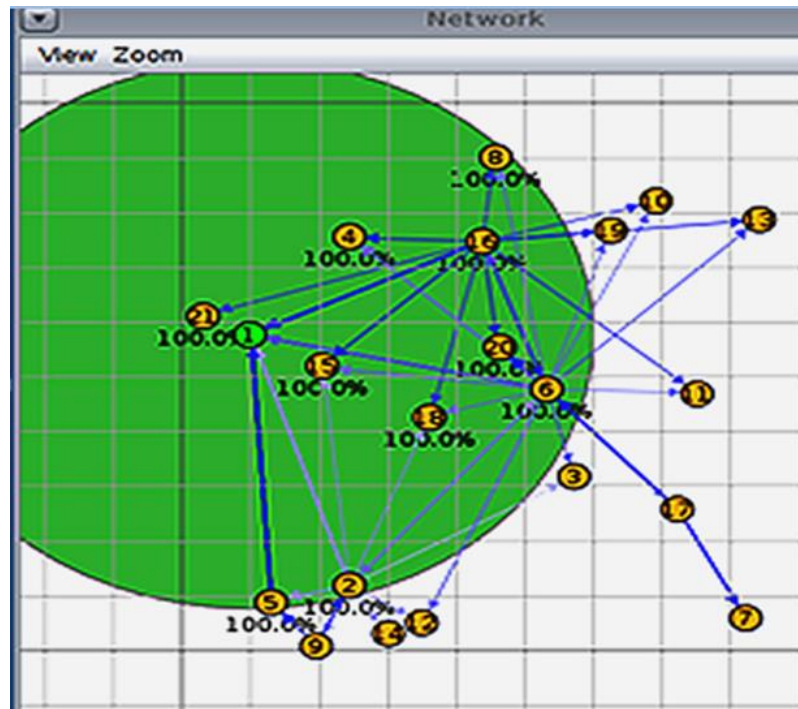


Figure 35. Plan d'une topologie en détection standard

En réglant l'affichage des données sur 10 secondes et en utilisant l'outil POWERTRACE de Contiki, tout en sachant que sa précision est de 94% par rapport à la valeur réelle [93], nous résumons l'expérimentation comme suit :

1. Nœud 15: L'analyse, faite sur le nœud le plus proche du puits de données, nous a donné le tableau 15 suivant :

Time	ALL CPU	ALL LPM	ALL TX	ALL RX
0	14902	312578	3070	6686
10	34244	620853	6061	11922
20	52817	929943	6836	16040
30	64626	1245800	9056	18395
40	76011	1562087	9795	20365
50	94331	1871430	12145	24386
60	106827	2186599	14337	27172
70	118474	2502616	16545	29142
80	130156	2818601	18795	31168
90	148018	3128405	20045	33581
100	161477	3442793	22595	37137
110	173940	3757821	25047	39887
120	185565	4073865	27735	41857

130	197266	4389828	30051	43827
140	208931	4705830	32655	45797
150	220532	5021897	35140	47767
160	232200	5337896	37625	49737
170	244925	5652837	41045	51821
180	256569	5968862	43210	53791
190	268565	6284531	45140	56050
200	280938	6599823	47354	58723

Tableau 15. Nombre de « ticks » pour le nœud 15

Le tableau 15 présente un retour de valeurs de nœud 15 dans quatre états différents sous la forme d'un nombre de ticks d'horloge.

- ALL_CPU : le total (haut) du CPU (CPU en mode actif).
- ALL_LPM : le nombre total de ticks dans un état LPM (low power mode).
- ALL_Tx : le nombre total de ticks dans l'état Tx (Transmit).
- ALL_Rx : le nombre de ticks dans l'état Rx (Receive).

En appliquant l'équation 23 sur les valeurs du tableau 15, nous obtenons le tableau 16 suivant du cycle effectif radio :

Time	Tx(%)	Rx(%)	Total(%)
10	0.91	0.02	0.93
20	0.24	0.01	0.25
30	0.68	0.01	0.68
40	0.23	0.01	0.23
50	0.72	0.01	0.73
60	0.67	0.01	0.68
70	0.67	0.01	0.68
80	0.69	0.01	0.69
90	0.38	0.01	0.39
100	0.78	0.01	0.79
110	0.75	0.01	0.76
120	0.82	0.01	0.83
130	0.71	0.01	0.71
140	0.79	0.01	0.80
150	0.76	0.01	0.76
160	0.76	0.01	0.76
170	1.04	0.01	1.05
180	0.66	0.01	0.67
190	0.59	0.01	0.60
200	0.68	0.01	0.68

Tableau 16. Pourcentage du cycle effectif de la radio du nœud 15

Exemple :

Pour calculer Tx-Eff_c(%) = ?

Tx-Eff_c(%) = (6061-3070) / ((34244-14902) + (620853- 312578)) = 0.009130 = 0.91%
(voir Tableau 15 et Tableau 16).

Après avoir activé POWERTRACE sur Cooja, connecté sa bibliothèque et configuré son horloge à 10 secondes dans notre programme source, nous obtenons un ensemble de valeurs imprimées dans le tableau 17 suivant représentant une estimation de la quantité d'énergie consommée :

Time	CPU	LPM	Tx	Rx
10	0.088541	0.007338	0.533977	1.045027
20	0.085020	0.007357	0.138359	0.821891
30	0.054057	0.007519	0.396332	0.470023
40	0.052116	0.007529	0.131932	0.393182
50	0.083862	0.007364	0.419540	0.802531
60	0.057202	0.007502	0.391333	0.556044
70	0.053316	0.007522	0.394189	0.393182
80	0.053476	0.007522	0.401688	0.404359
90	0.081766	0.007374	0.223160	0.481599
100	0.061610	0.007484	0.455246	0.709724
110	0.057051	0.007499	0.437750	0.548859
120	0.053215	0.007523	0.479883	0.393182
130	0.053563	0.007521	0.413470	0.393182
140	0.053398	0.007522	0.464886	0.393182
150	0.053105	0.007524	0.443642	0.393182
160	0.053412	0.007522	0.443642	0.393182
170	0.058250	0.007497	0.610565	0.415935
180	0.053302	0.007523	0.386513	0.393182
190	0.054913	0.007514	0.344559	0.450862
200	0.056639	0.007505	0.395261	0.533491

Tableau 17. Consommation d'énergie du nœud 15

Note :

Voir la fiche technique de TmoteSky pp.9 [80]

1. Courant actif à $V_{cc} = 500\mu A/s = 0.50mA/s$, Tension = 3v et $1s = 32768$ horloges.
 2. Energie consommée (CPU)= $(34244-14902) \times 0.50 \times 3 / (32768 \times 10) = 0,088541$ mw.
- Nous calculons la consommation d'énergie en utilisant la formule de l'équation 19.

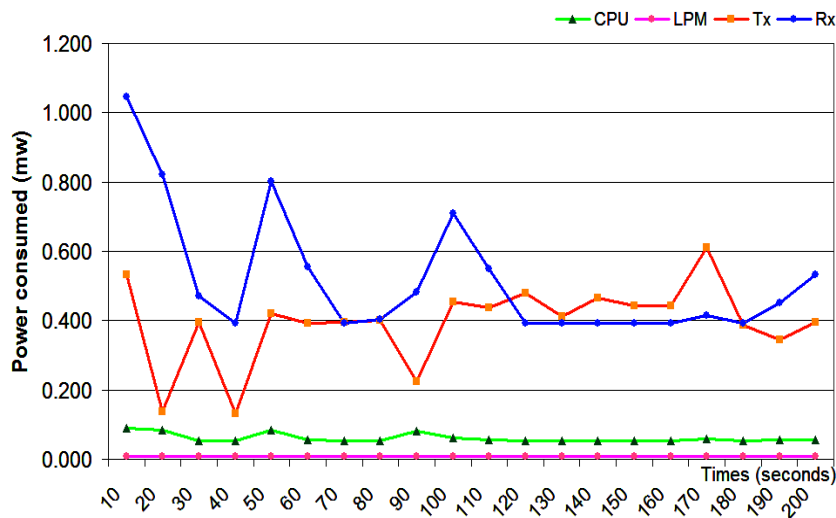


Figure 36. Consommation d'énergie du nœud 15 en détection standard

Le modèle standard montre dans la figure 36, la variation de la consommation d'énergie dans les quatre états mentionnés précédemment. On voit clairement que l'augmentation de l'énergie concerne la transmission, mais surtout la réception. La figure 37 confirme notre affirmation en additionnant les énergies de transmission et de réception pour représenter le cycle effectif du service radio du nœud 15.

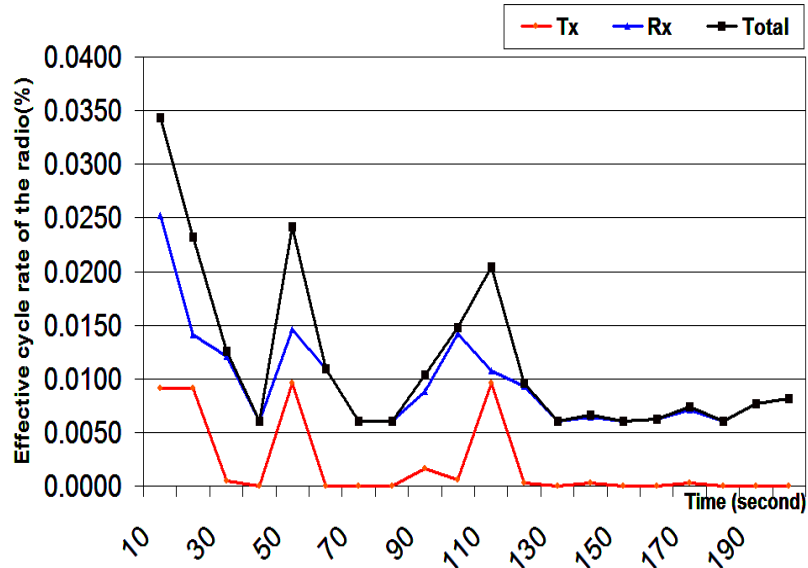


Figure 37. Taux du cycle effectif de la radio du nœud 15 en détection standard.

2. Nœud 6 :

Les données de nœud 6 en nombre de ticks capturées sont affichées dans le graphique suivant :

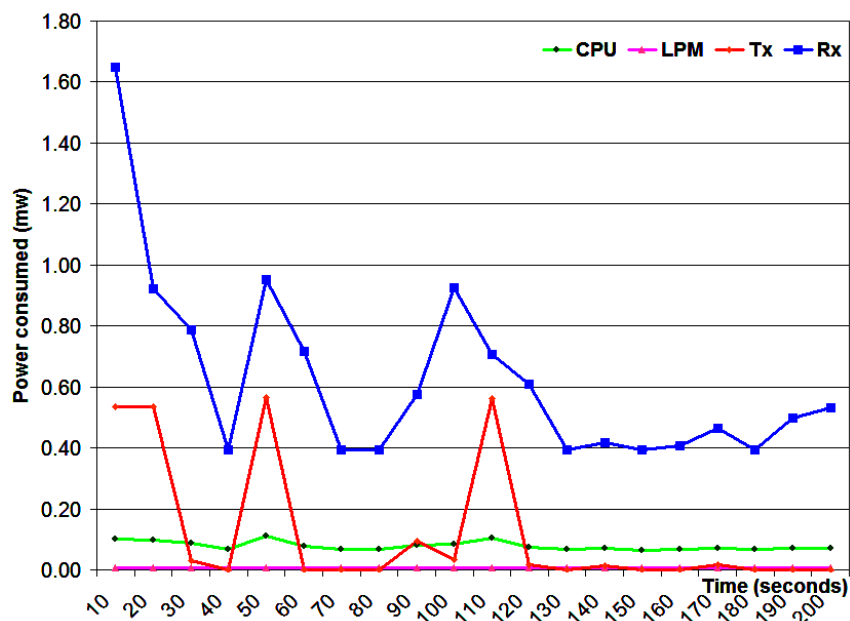


Figure 38. Consommation d'énergie du nœud 6 en détection standard

Comme pour la figure 36, la figure 38 représente la variation de la consommation énergétique dans les quatre états (CPU, LPM, Tx, Rx) du mote 6 dans le modèle standard. La figure 39 représente le cycle radio effectif afin de montrer la différence entre l'émission et la transmission.

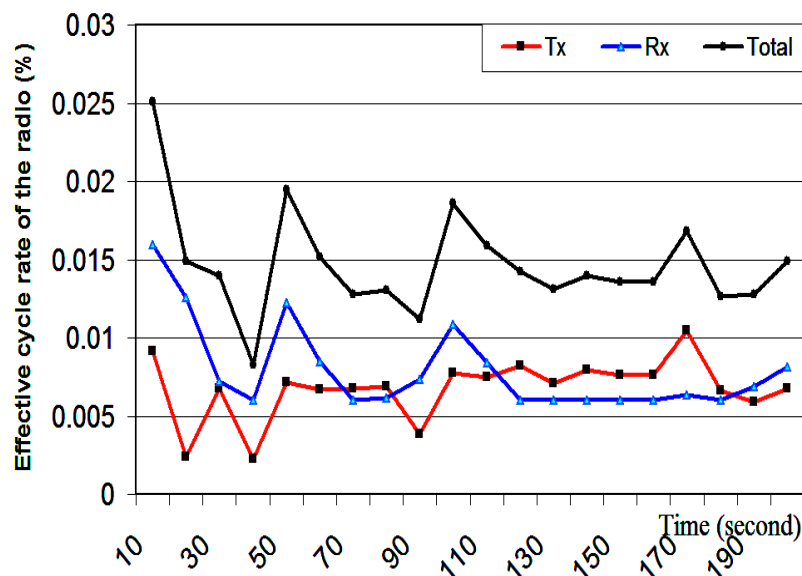


Figure 39. Cycle effectif de la radio du nœud 6 en détection standard

3. Nœud 7:

Maintenant, nous allons étudier le comportement énergétique du nœud 7, y compris son cycle radio effectif, afin de comprendre l'influence de la distance d'un nœud par rapport au sink afin de prédire préalablement un déroulement optimal avec la collaboration des nœuds entre eux. Les données du nœud 7 en nombre de ticks capturés sont affichées dans la figure 40.

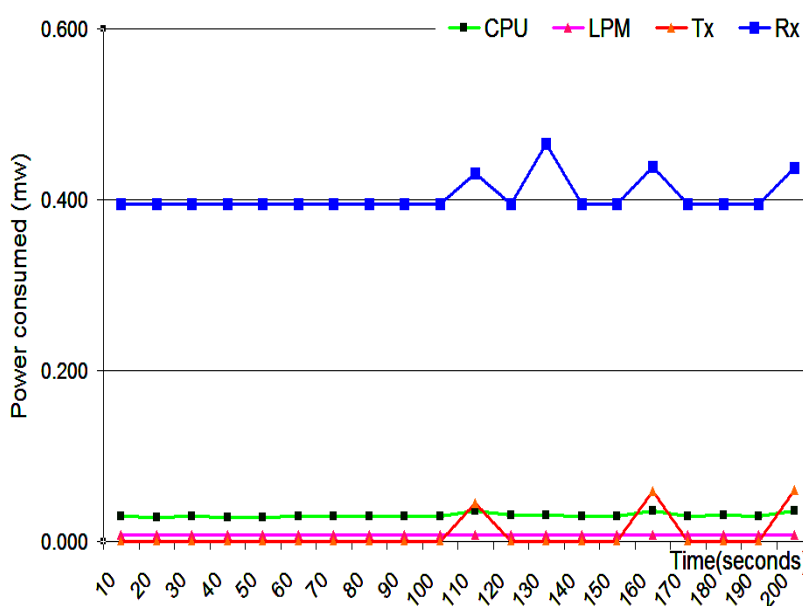


Figure 40. Consommation d'énergie en détection standard du nœud 6

Tout comme les figures 36 et 38, la figure 40 montre la variation de la consommation d'énergie dans les quatre états (CPU, LPM, Tx, Rx) du nœud 7 dans le modèle standard.

En analysant la figure 35 du plan de la topologie du modèle de détection standard, nous remarquons que le nœud 7 est en dehors de la couverture du sink d'un côté et qu'il n'a presque aucun voisin. La figure 41 indique une quasi-absence de transmission, mais le nœud 6 est toujours à l'écoute.

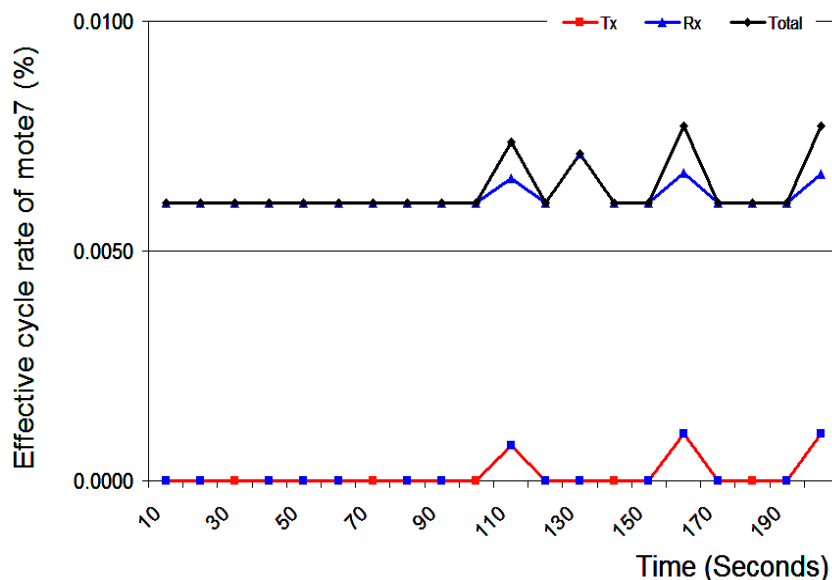


Figure 41. Taux de cycle effectif de la radio du nœud 7

V.4.2.1.3. Interprétation des résultats

Notez que le nœud 15 et le nœud 6 sont dans la même couverture du puits (nœud 1). Par contre le nœud 7 est en dehors de la zone de couverture du puits.

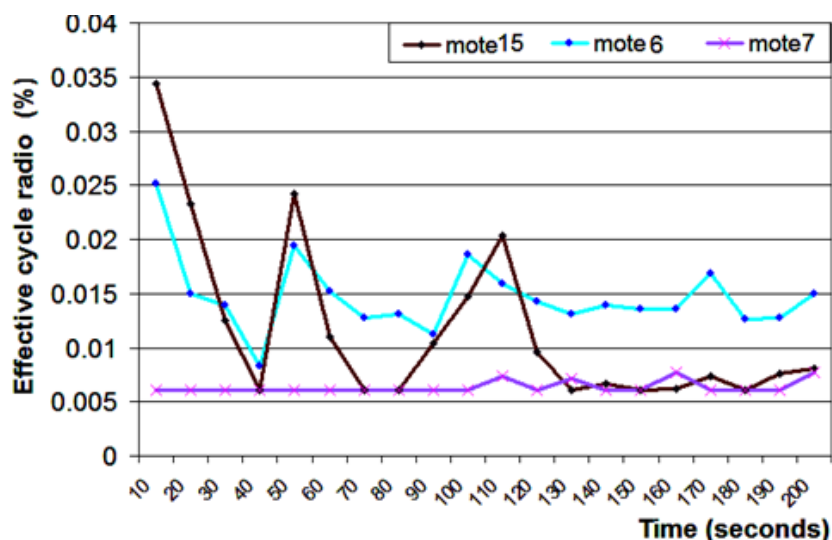


Figure 42. Taux de cycle effectif de la comparaison radio pour les trois nœuds,

On peut voir, sur la figure 42, que la consommation d'énergie des trois nœuds 15, 6 et 7 est plus marquée sur les nœuds en mode de réception de paquets (Rx) (figures 36, 38 et

40). Cela pourrait s'expliquer par le fait que les nœuds sont toujours à l'écoute du canal de transmission, et cela est également prouvé par les résultats trouvés par le cycle effectif de chaque mote (pourcentage élevé de Rx par rapport à Tx).

Les données correspondantes sur la mise en veille de ces trois nœuds (nœud 15, nœud 6, et nœud 7) sont représentées par le paramètre LPM qui souligne une consommation d'énergie. Cela signifie que les nœuds ne consomment presque pas d'énergie lorsqu'ils sont en mode veille.

D'après la figure 42, l'utilisation de la radio est plus accentuée lorsque les nœuds sont sur le couvercle du puits, mais elle l'est beaucoup plus lorsqu'elle se rapproche de ce dernier. Cela pourrait s'expliquer par le fait que le nœud 15 est plus susceptible de relayer des données vers le sink que les autres. Cette déduction est également vérifiée en analysant la quantité d'énergie consommée par le CPU pendant le traitement. Par exemple, le nœud 15 imprime (Figure 29) une utilisation moyenne entre [0.3 mw - 0.5 mw].

V.4.2.1.4. Conclusion

Une conclusion importante est dégagée à partir de ces résultats : Quand un nœud (ici nœud 6) a plusieurs voisins, l'énergie consommée à la réception augmente considérablement (Figure 37). Ceci peut être vérifié avec le nœud 7 qui n'a qu'un seul voisin (nœud 17).

Ainsi, pour un déploiement déterministe optimal des capteurs, il faut éviter la concentration des nœuds autour d'un seul nœud, car cela induira une utilisation excessive de l'énergie, et aboutira à la perte du nœud et à une restructuration de la topologie ou à la perte définitive du réseau de capteurs sans fil.

V.4.2.2. La méthode proposée « Détection et coopération de 3-voisins »

V.4.2.2.1. Introduction

Cette approche est basée sur une méthode hybride, combinant l'algorithme de consensus de moyenne et l'algorithme d'agrégation de rumeur de triplets, afin de se mettre d'accord sur la valeur détectée (moyenne entre paire de voisins dont les valeurs sont les plus proches) à relayer au puits.

A un instant " t " de l'horloge, un nœud chef de grappe N_i (CH) est choisi aléatoirement parmi tous les nœuds (CHs). « N_i » sélectionne deux nœuds voisins « N_j », « N_k » parmi tous les nœuds de sa grappe en respectant la condition : les nœuds élus doivent avoir les valeurs les plus proches de la valeur de « N_i ».

Une exception à cette règle est faite lorsqu'un des nœuds voisins est hors de portée ; la communication par 3-voisins est alors réduite à une discussion par paire dans ce créneau temporel. Quant au coût de transmission, la notification du nœud « N_i » aux nœuds « N_j » et « N_k » est évaluée à une transmission, grâce à la fonction de diffusion sans fil.

Les nœuds « N_j » et « N_k » transmettent leurs propres valeurs locales « $V_{N_j}(t)$ » et « $V_{N_k}(t)$ » respectivement au nœud « N_i » (pas de collision, le nœud « N_i » peut apporter un consensus à sa première notification) et au nœud « N_k ». Cette étape coûte 2 transmissions. Le nœud « N_i » fait la moyenne des valeurs des trois nœuds sélectionnés localement et met à jour son estimation locale selon l'algorithme Coop1 :

Algorithme Coop1

```

// Un capteur  $N_1$  est activé aléatoirement. Il choisit deux voisins les plus proches  $N_2$  et  $N_3$ 
// parmi les autres //
// { $V_{d1}, V_{d2}, V_{d3}$  valeurs détectées par  $N_1, N_2$  et  $N_3$  respectivement} //
// { $d_{12}, d_{23}, d_{13}$  différences respectives entre les valeurs  $V_{d1}, V_{d2}, V_{d3}$ } //
Lire ( $V_{d1}, V_{d2}, V_{d3}$ )
//
min ← abs( $V_{d1}-V_{d2}$ )
Moy ← ( $V_{d1}+V_{d2}$ )/2
//
Si min > abs( $V_{d1}-V_{d3}$ ) alors
    min ← abs( $V_{d1}-V_{d3}$ )
    Moy ← ( $V_{d1}+V_{d3}$ )/2
Finsi
//
Si min > abs( $V_{d2}-V_{d3}$ ) alors
    min ← abs( $V_{d2}-V_{d3}$ )
    Moy ← ( $V_{d2}+V_{d3}$ )/2
Finsi
//
Envoyer (Moy)
Fin
    
```

La méthode de détection et coopération de 3-voisins (figure 43) résume la valeur moyenne prise par la comparaison entre trois valeurs des nœuds voisins, par 3 transmissions radio dans un contexte de consensus par conversation.

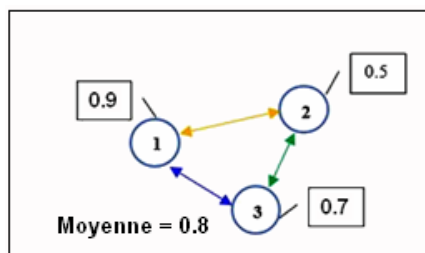


Figure 43. Exemple de sélection de valeurs à transmettre par le nœud n°1

Remarque :

Une exécution de l'algorithme consomme soit :

- a. 10 opérations de simple structure de données : 01 addition, 02 soustraction, 01 appels de fonction (Abs), 01 division, 01 affectation, 02 comparaisons et 01 opération d'écriture.
- b. 14 opérations de simple structure de données : 02 addition, 03 soustraction, 02 appels de fonction (Abs), 02 division, 02 affectation, 02 comparaisons et 01 opération d'écriture.

Ce nombre d'opérations bénéficie d'une consommation énergétique négligeable. Dans ce contexte, plus l'algorithme est moins complexe plus la consommation en énergie est faible en traitement.

V.4.2.2.2. Simulation

La configuration choisie est donnée par la figure 44 ci-après. La simulation de notre méthode est basée sur 7 nœuds. Un puits, 4 nœuds de capture standard et deux nœuds agrégateurs chargés de coopérer pour envoyer une valeur unique, selon un consensus, au puits. Les nœuds standards 4, 5, 6 et 7 détectent et transmettent les valeurs de température, d'humidité et de lumière en broadcast. Les nœuds 4 et 5 agrègent leurs données au nœud 2 et les nœuds 6 et 7 vers le nœud 3. Les nœuds 2 et 3 sont les agrégateurs.

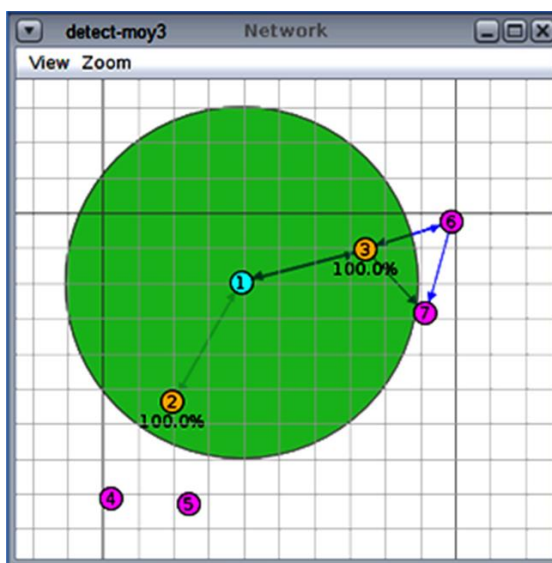


Figure 44. Topologie utilisée dans la méthode de 3-voisins

Notre méthode est basée sur les agrégateurs pour optimiser la transmission radio et surmonter les problèmes posés par la configuration standard. Une fois la simulation déclenchée, l'outil POWERTRACE retrace les résultats dans le tableau 18 suivant :

1. Mote 2 :

Sur ce mote nous notons un nombre de ticks après chaque 10 secondes :

Time	ALL CPU	ALL LPM	ALL TX	ALL RX
0	15584	312155	5765	3422

10	31396	623959	9080	7586
20	44257	938711	12237	10370
30	52227	1258393	12317	13175
40	58490	1579789	12317	15155
50	72257	1893695	15468	18589
60	79606	2213977	15548	20708
70	85543	2535618	15548	22688
80	91483	2857264	15548	24668
90	97890	3178514	15548	26648
100	111115	3492951	18705	29548
110	118191	3813542	18705	32537
120	126278	4133033	18875	34795
130	132433	4454461	18875	36938
140	138398	4776082	18875	38918
150	144656	5097483	18875	40898
160	150987	5418815	18875	43092
170	157191	5740282	18875	45072
180	163593	6061546	18875	47290
190	169786	6383022	18875	49270
200	176559	6703892	18965	51391

Tableau 18. Nombre de ticks du mote 2

En respectant la même approche du modèle standard mais en utilisant cette fois-ci notre algorithme, nous obtenons le tableau 18 représentant le nombre de ticks dans les quatre états (CPU, LPM, Tx, Rx) pour le mote 2. Ensuite, nous calculons la consommation d'énergie en utilisant la formule de l'équation 25, alors nous obtenons la figure 38, une représentation graphique qui visualise la consommation d'énergie par notre méthode des 3-voisins. Cette courbe met en évidence une augmentation de la consommation d'énergie autour de la radio (Tx : transmission, Rx : réception).

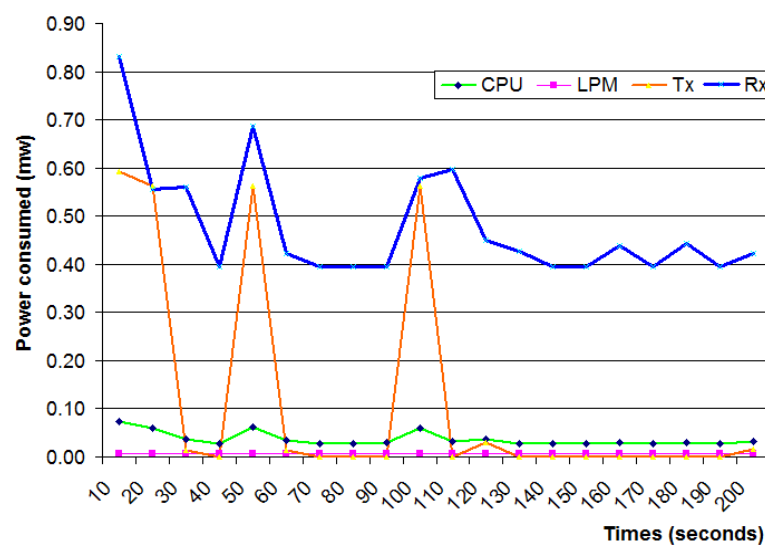


Figure 38. Consommation d'énergie du mote 2

De même, la figure 46 souligne une utilisation beaucoup plus importante de la transmission que de la réception.

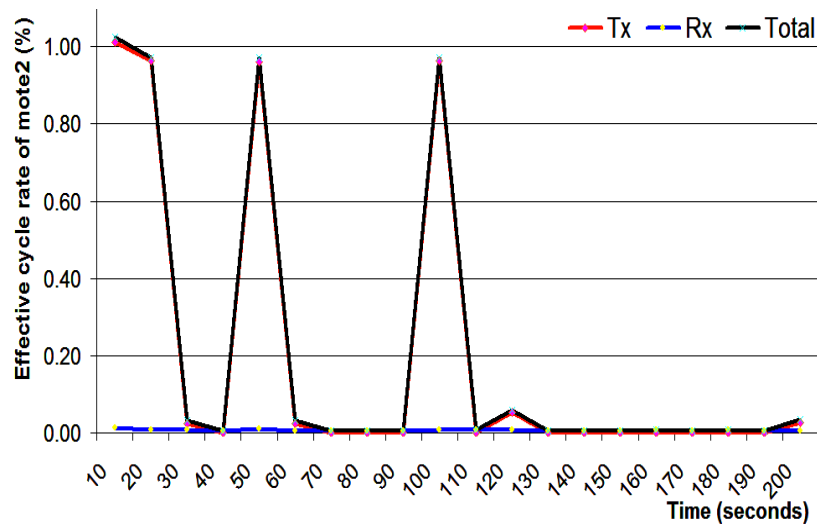


Figure 46. Pourcentage du nombre de ticks pour la mote 2

2. Mote 3 :

Sur ce mote agrégateur, nous notons également le nombre de ticks après chaque 10 secondes. De la même manière que pour Mote 2 et en utilisant l'outil POWERTRACE, les données collectées par l'agrégateur Mote 3 sont représentées sur la figure 47 marquant l'énergie consommée en transmission et en réception ainsi que par le processeur.

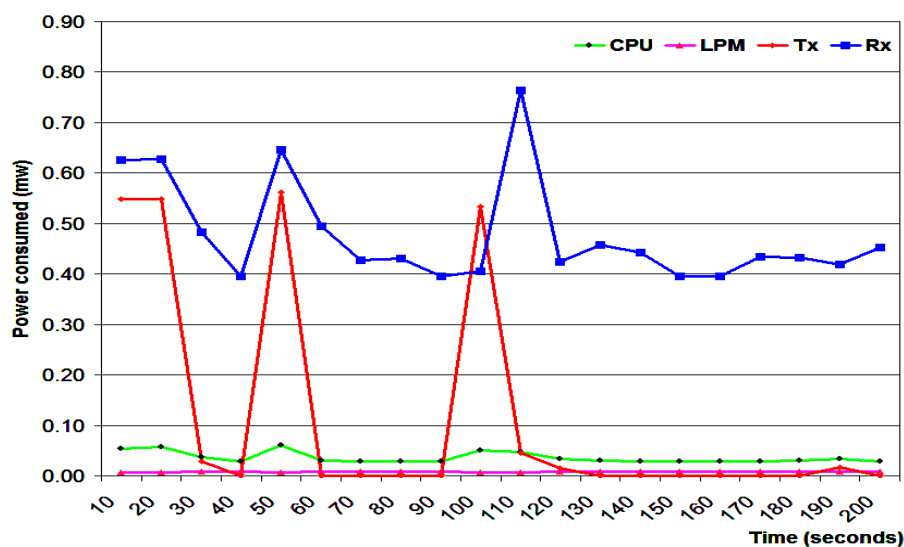


Figure 47. Consommation d'énergie du mote 3

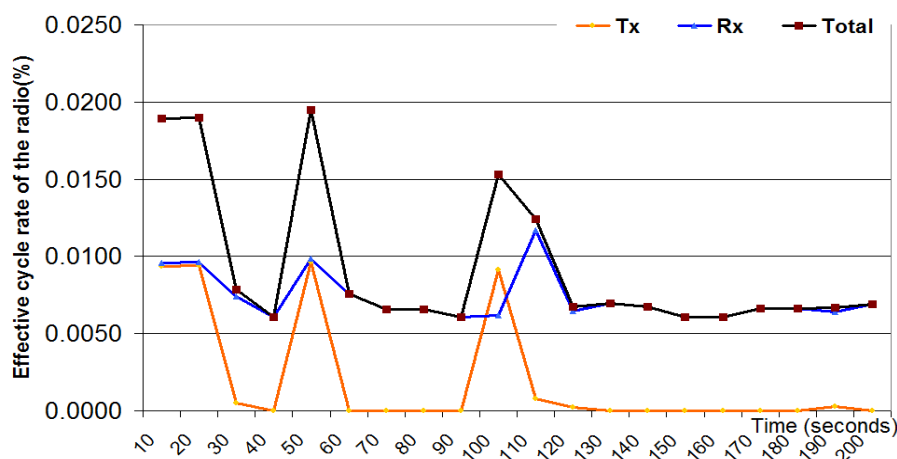


Figure 48. Pourcentage du nombre de ticks pour le mote 3

Nous obtenons la figure 48 qui visualise la consommation d'énergie par notre méthode des 3-voisins. Cette courbe met en évidence une augmentation de la consommation d'énergie autour de la radio (Tx, Rx). La figure 48 montre le cycle réel d'utilisation de la radio du mote 3 et donne une représentation graphique de l'utilisation élevée de l'écoute par rapport à la transmission qui semble occasionnelle.

3. Mote 4:

Il s'agit d'un simple capteur situé en dehors de la zone de couverture du puits et nous voulons suivre son comportement et sa consommation d'énergie tout au long du processus avec la méthode des 3-voisins. Le tableau 19 note également le nombre de ticks après chaque 10 secondes.

Time	ALL CPU	ALL LPM	ALL TX	ALL RX
0	9403	318366	2992	3044
10	23207	632223	7426	6281
20	28738	954356	7426	8527
30	39955	1270802	10658	10943
40	45199	1593220	10658	12923
50	56917	1909170	13971	15644
60	62611	2231064	13971	17624
70	68188	2553069	13971	19604
80	73776	2875067	13971	21584
90	79342	3197086	13971	23564
100	91118	3512923	17060	26005
110	101343	3830365	20053	28421
120	107040	4152255	20053	30401
130	112700	4474244	20053	32605
140	118168	4796442	20053	34585
150	123620	5118657	20053	36565

160	133746	5436194	22820	38591
170	139037	5758573	22820	40571
180	149221	6076023	25590	42598
190	154507	6398399	25590	44578
200	159742	6720829	25590	46558

Tableau 19. Nombre de ticks du mote 4

La figure 49 et la figure 50 montrent respectivement la consommation d'énergie et le cycle effectif des principaux modules en utilisant la méthode des 3-voisins. L'observation est mise en évidence sur le mode de réception (Rx) en premier et la radio transmission en second.

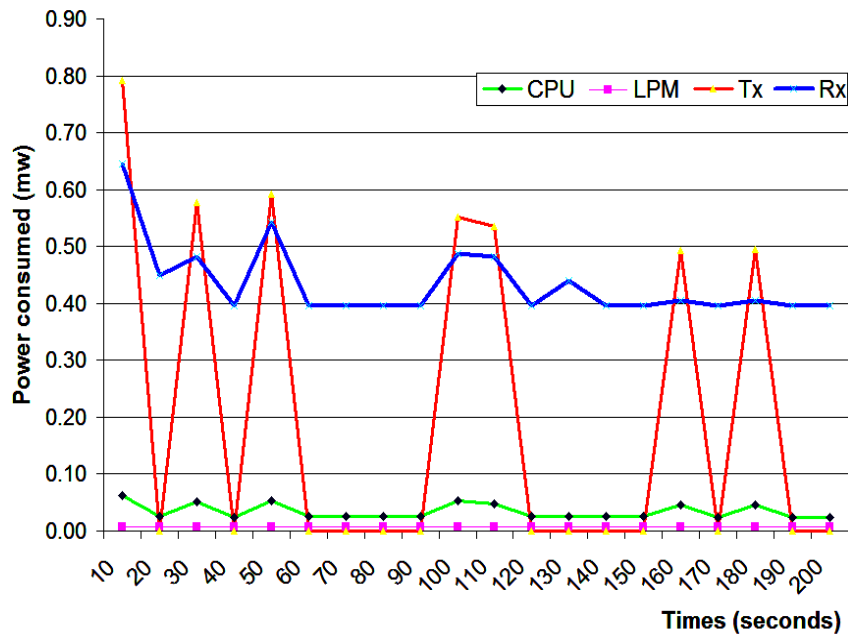


Figure 49. Consommation d'énergie de mote 4 par la méthode des 3-voisins

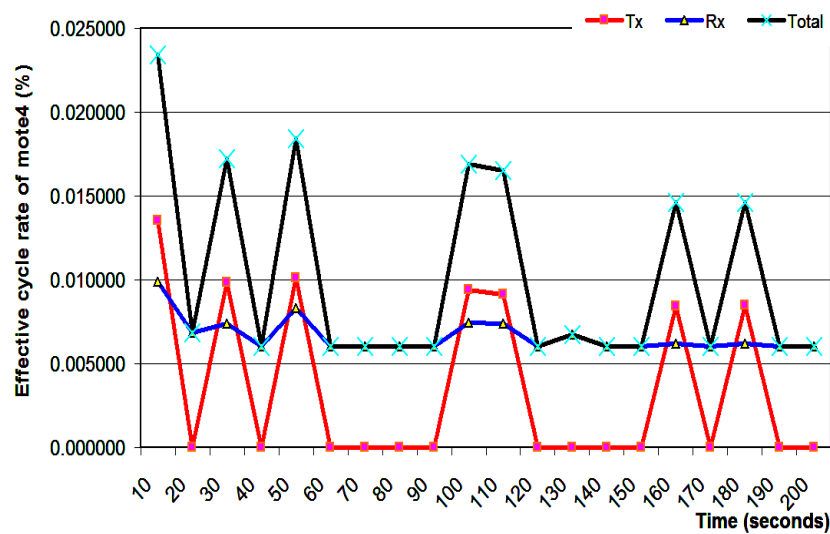


Figure 50. Taux du cycle effectif du mote 4

V.4.2.2.3. Interprétation des résultats

D'après les résultats ci-dessus, on remarque que la consommation d'énergie pour ces trois nœuds, lorsqu'ils sont en mode transmission (TX) ou réception (RX), est plus importante que si ces derniers sont en mode calcul (CPU) ou en mode veille (LPM). Cela signifie que la majeure partie de l'énergie est souvent consommée en émission et en réception, et qu'elle est plus consommée en réception qu'en émission.

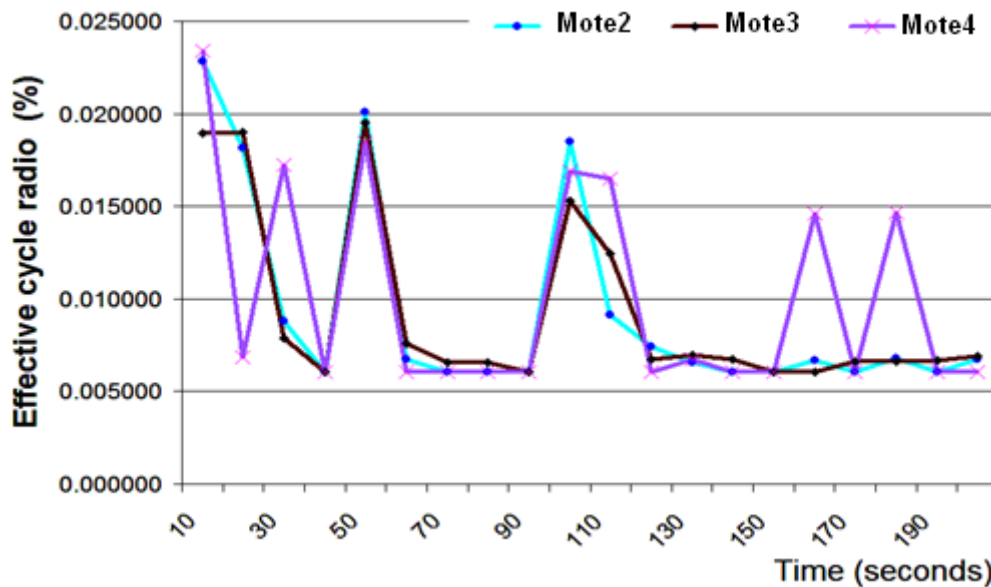


Figure 51. Comparaison du taux de cycle effectif radio des nœuds 2, 3 et 4

Dans la figure 51, le déploiement du nœud 4, un nœud simple, est en dehors de la zone de couverture du puits. L'observation que l'on peut faire du cycle effectif radio est que le nœud 4 consomme une énergie importante par rapport aux nœuds 2 et 3 qui sont des agrégateurs. Nous expliquons cette consommation excessive d'énergie des nœuds 4 et 7, entre autres, par la transmission en Broadcast des températures, taux d'humidité et taux de lumière tous les 30 secondes d'une part, et leur déploiement rapproché (nœud 4 et nœud 5) qui donne un avantage au RSSI de prendre le dessus de dialogue. Par contre, si nous analysons le traitement des données, nous affirmons que les nœuds 2 et 3 consomment plus d'énergie que le nœud 4, et elle est faible par rapport à celle consommée dans les réceptions.

Le protocole standard du RPL de Cooja permet de relayer la valeur capturée d'un nœud à un autre en se basant simplement sur le RSSI. Cette collecte d'information peut générer une altération de données et relayer de fausses informations, ce qui conduira à de mauvaises décisions. Nous avons introduit une nouvelle méthode de coopération appelée « moyenne des 3-voisin » inspiré de la méthode du bavardage par triplet [94], basée sur un

consensus de fiabilité administré par un nœud agrégateur. Ce dernier évalue la moyenne, des valeurs de deux nœuds voisins ayant les valeurs les plus proches, afin de l'envoyer au Sink. Cette perspective assure d'une part la fiabilité des données et d'autre part la perpétuité du réseau. Pour mieux comparer la méthode proposée à la méthode standard, il est impératif de choisir la même topologie et de lancer la simulation sur les deux méthodes. Le puits de données étant le nœud 1, les nœuds 2 et 3 sont des agrégateurs (voir Figure 52). Par contre les autres nœuds sont des nœuds de captures simples permettant aussi de relayer l'information.

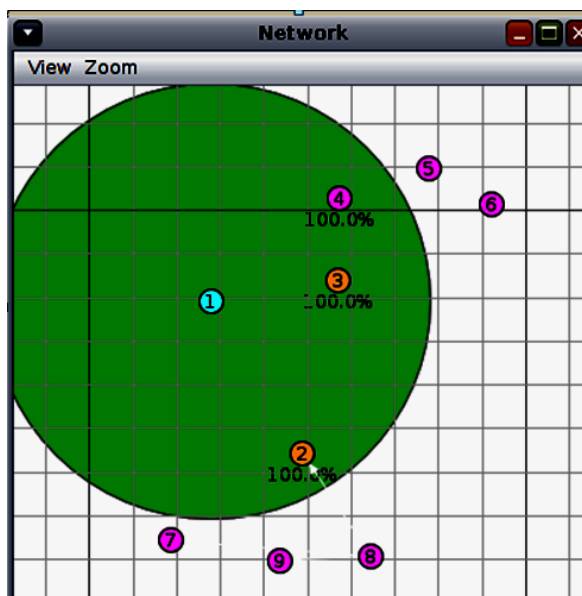


Figure 52. Plan de comparaison des méthodes

Après le lancement de la simulation et en générant des données intermédiaires avec la même méthode suivie que précédemment, nous déduisons le tableau 20 suivant.

Méthode	Temps d'arrivée des données au sink	Consommation d'énergie milliWatts
Méthode standard basée sur RPL	18.187 s	0.537 mw par nœud
Méthode proposée des 3-voisins	46.473 s	0.469 mw par nœud

Tableau 20. Estimation du temps de réponse et de la convergence du réseau

Ce résultat est très important, car il montre que la méthode, ainsi élaborée, réalise une optimisation unilatérale reposant, beaucoup plus, sur la consommation d'énergie que sur le temps d'arrivée des données au puits (voir Figure 53) :

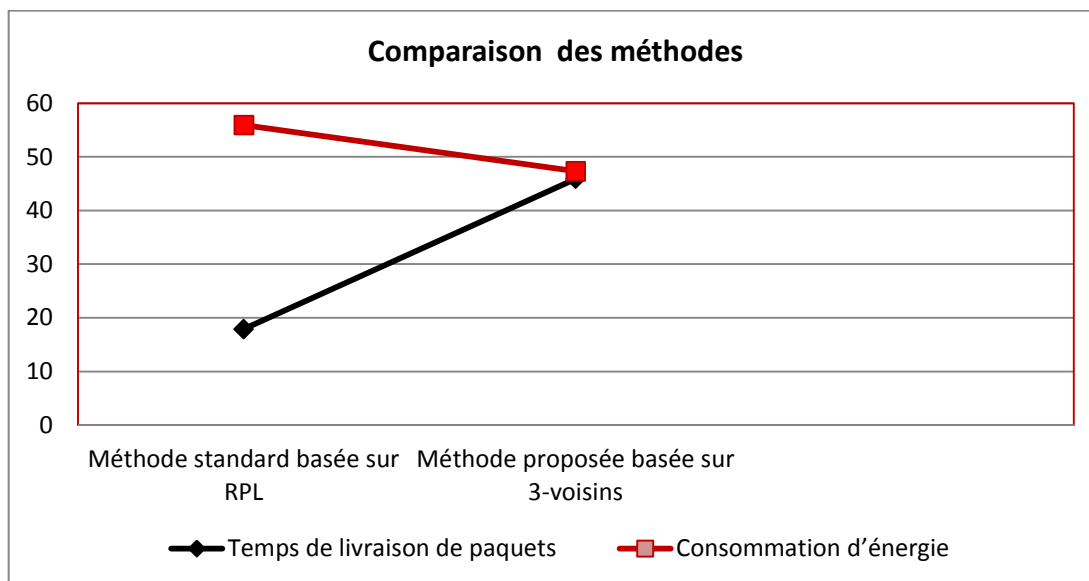


Figure 53. Comparaison de la méthode proposée sur un nouveau plan

Sur cette figure nous constatons que le temps de livraison de paquet est meilleur par rapport à la méthode standard. Mais l'optimisation en consommation d'énergie reste nettement visible sur le graphe favorisant ainsi notre méthode.

V.4.2.2.4. Conclusion

Dans cette partie, nous avons introduit une nouvelle méthode de coopération appelée « moyenne des 3-voisin » inspiré de la méthode du bavardage par triplet [94], mais le choix s'est porté sur un déploiement en cluster avec un choix adéquat des nœuds, ce qui a donné des résultats prometteurs, dans lequel l'accent est mis sur la consommation d'énergie dans les trois modules essentiels Tx, Rx, CPU.

Nous remarquons que notre solution appelée « méthode des 3-voisins » est évidemment la meilleure solution en matière de consommation d'énergie et donc de pérennité du réseau. La méthode standard souligne un temps de délivrance de paquet plus intéressant que la nôtre, qui est sûrement dû à notre programme de traitement du consensus moyen. Mais, elle consomme plus d'énergie. Il convient donc de choisir la méthode à utiliser selon le type de l'application convoitée.

V.4.3. Partie III - Coopération des capteurs par classification – 2^{ème} approche –

Cette partie serait ainsi de considérer une nouvelle méthode de coopération des nœuds de capteurs afin d'optimisation d'énergie d'un réseau de capteurs en minimisant les fréquences de transmission des données mesurées vers la station de base. Le nœud (mote) possède plusieurs capteurs et mesure différentes valeurs environnementales. La méthode

proposée reviendrait à catégoriser les données captées dans des classes prédéfinies, numérotées au préalable, que nous la nommerons « Intervalles de confiance ». De cette manière chaque valeur captée sera rangée dans une classe, et seul son numéro sera envoyé à la station de base, si (et seulement si) un changement de classe, par rapport à la valeur précédente, est observé. L'expérimentation est faite avec des données réelles issues du laboratoire d'Intel Berkley.

V.4.3.2. Compréhension de l'approche

Dans les applications de réseaux de capteurs, il y a nécessairement une collecte d'informations sur l'état d'un environnement afin de se prévenir des risques. Cela suppose donc que les données provenant de ces réseaux de capteurs aient un système de communication efficace et optimisé. L'optimalité, ici, signifie que le capteur doit pouvoir prendre une décision appropriée d'envoi ou non des données au puits. Beaucoup de travaux proposent de réduire le nombre de transmissions et diminuer la quantité de données, ce qui va prolonger la durée de vie du réseau. On peut donc parler de la notion de changements relatifs et celles des différences relatives entre les valeurs mesurées à l'instant « t » du capteur et les valeurs déjà mesurées à l'instant « $t-1$ » du même capteur. Les auteurs de l'article [95] exposent leur approche, avec des résultats concurrents, pour pallier aux travaux de [96] dont les auteurs proposent un algorithme de consensus, avec le principe qu'aucun des nœuds ne puisse envoyer réellement ses données si la valeur captée « $Sv(t)$ » de l'état courant (Curent value measured by the Sensor) de la détection, est identique (algorithme ES : Exactly Same) à la valeur du précédent état « $(Sv(t-1))$ ». En d'autres termes, la valeur du changement absolu entre « $Sv(t)$ » et « $Sv(t-1)$ » est nulle.

Ce travail est centré sur la détection et collecte des données. Nous faisons le point sur une application de prise de décision des transmissions de données captées qui introduit la notion « d'intervalle de confiance ». Cet algorithme décidera ainsi de déclencher ou non la transmission des données vers la station de base pour participer à l'optimisation, la plus efficace, de la transmission, et par conséquent en minimiser l'énergie.

La réduction du nombre de transmissions de données, par requêtes et par intervalle de temps, en vue de collecter périodiquement les mises à jours des données d'une zone sous observation, est un moyen efficace de réduire le le Duty-cycle (On/Off) des capteurs, et par conséquent accroître la durée de vie du réseau IoT. Les auteurs de [96] proposent une agrégation de données vers le nœud collecteur ; un traitement est fait à ce niveau. Ils recourent à « l'égalité absolue » ; aussi, si la valeur de l'état actuel « $Sv(t)$ » est « égale exactement » –

au sens absolu – à la valeur de l'état précédent « $Sv(t-1)$ » – précédemment enregistrée et sauvegardée dans un tampon du collecteur – il n'y aura pas de transfert de données à l'observateur. De même les auteurs de [95] reprennent ces travaux et soulignent la même idée avec un algorithme amélioré : un nœud ne transmet ces données que s'il y a une « *différence relative* » entre la valeur mesurée à l'instant « t » et la dernière valeur mesurée transmise à « $t-1$ ».

Définition 1

La valeur des données captée est relative, selon que l'on se trouve du côté du réseau ou de celui de l'observateur. Ce dernier s'intéresse beaucoup plus aux changements de l'environnement surveillé qu'aux valeurs des données elles-mêmes.

Par exemple, si les capteurs surveillent une zone de forêt pour prévenir le déclenchement du feu, l'observateur n'a pas besoin de voir les données numériques de tous les capteurs individuels, mais doit savoir s'il y a eu augmentation notable de température indiquant le démarrage d'un incendie dans la zone surveillée. C'est cette question primordiale qui va déclencher des prises de décisions efficaces.

– Notions mathématiques

La « *différence réelle* » entre deux valeurs numériques, v_t et v_{t-1} , est présentée par la formule : $\mathfrak{X} = v_t - v_{t-1}$. Cette dernière est dite « *changement réel* » lorsque la valeur v_{t-1} est une valeur de départ ou de référence à laquelle v_t est comparée. Par contre, si v_{t-1} n'est pas une valeur de référence, la valeur de \mathfrak{X} serait alors nommée « *différence absolue* », soit : $|\mathfrak{X}| = |v_t - v_{t-1}|$, où l'ordre de v_t , v_{t-1} n'a pas d'importance. Cependant, une autre observation suscite notre attention : à savoir que la différence absolue entre deux valeurs n'est pas toujours le bon moyen de comparer les nombres.

V.4.3.3. Approche proposée

V.4.3.3.1. Introduction

Dans un réseau de capteur sans fil, le déploiement de capteurs a pour but de surveiller une zone quasi- autonome ; chaque capteur étant concerné par une sous-zone. Un mote spécial est chargée de rassembler les informations en provenance des capteurs et d'en informer l'observateur humain si un phénomène inhabituel survient. Ainsi, minimiser la consommation d'énergie reviendrait à minimiser, entre autres actions, la quantité de données transmises dans le réseau. En effet, selon les études menées dans ce contexte, l'énergie consommée dans un nœud capteur serait due aux transmissions de données.

De ce fait, notre approche consiste à agréger les données des différents capteurs en une information significative appelée « Intervalle de confiance », en éliminant la redondance par le recours à une nouvelle méthode basée sur la notion de tolérance (d'appartenance à des intervalles de confiance). Ceci résoudra nécessairement le problème d'excès de transmission de valeurs collectées, et par conséquent augmentera la durée de vie des capteurs.

Les modèles de Markov Cachés (Hidden Markov Models ou HMM) modélisent des phénomènes aléatoires composés d'un processus aléatoire de transition entre des états inobservables dans un premier niveau (les états cachés) et, à un second niveau, d'un autre processus aléatoire qui, dans chaque état, engendre des valeurs observables (appelées observations). Notre idée repose sur le principe du niveau d'observation des HMM.

Nous construisons une discrétisation des données de transitions d'un ensemble de jours, nous relevons une classification par intervalle des données, et nous l'appliquons sur l'ensemble des données des jours à venir. De ce fait, les valeurs seront vérifiées par rapport à un intervalle de données préétabli dans les conditions normales. Si une valeur n'est pas classée dans ces intervalles de confiance, elle l'est donc dans une classe spéciale appelée « OutOff range ». L'application de la méthode « *Exacly Same* » – Valeur à l'instant « $S_{v_{t-1}}$ » est égale la valeur captée à l'instant « S_{v_t} », et sera appliquée pour comparer les appartenances aux classes. Ainsi, toutes les valeurs normales recueillies seront dans une classe spécifiée et connues à l'avance, et les valeurs d'alertes seront quant à elles représentées par la classe « 0000 ». A noter que l'ensemble des tests est effectué sur la base de données des informations (Figure 53) collectées par 54 capteurs déployés dans le laboratoire Intel Berkeley Research entre le 28 février et le 5 avril 2004. [97]

N°	Date	Heure	epoch	MotelID	Temp	Humidity	Lumière	Tension
1	28/02/2004	00:59:16.028	3	1	19.9884	37.0933	45.0800	2.6996
2	28/02/2004	01:03:16.334	11	1	19.3024	38.4629	45.0800	2.6874
3	28/02/2004	01:06:16.013	17	1	19.1652	38.8039	45.0800	2.6874
4	28/02/2004	01:06:46.778	18	1	19.1750	38.8379	45.0800	2.6996
5	28/02/2004	01:08:45.993	22	1	19.1456	38.9401	45.0800	2.6874
6	28/02/2004	01:09:22.324	23	1	19.1652	38.8720	45.0800	2.6874
7	28/02/2004	01:09:46.110	24	1	19.1652	38.8039	45.0800	2.6874
8	28/02/2004	01:10:16.679	25	1	19.1456	38.8379	45.0800	2.6996
9	28/02/2004	01:10:46.251	26	1	19.1456	38.8720	45.0800	2.6874
10	28/02/2004	01:11:46.941	28	1	19.1456	38.9401	45.0800	2.6996
11	28/02/2004	01:12:46.251	30	1	19.1358	38.9061	45.0800	2.6874
12	28/02/2004	01:14:16.631	33	1	19.1162	38.8039	45.0800	2.6996
13	28/02/2004	01:14:46.569	34	1	19.1162	38.8720	45.0800	2.6996
	28/02/2004	01:15:16.650	35	1	19.1064	39.0082	45.0800	2.6996
	28/02/2004	01:16:16.344	37	1	19.1064	38.8720	43.2400	2.6996
	28/02/2004	01:16:46.509	38	1	19.0966	38.8039	43.2400	2.6996
	28/02/2004	01:17:16.672	40	1	19.0966	38.7357	43.2400	2.6996
	28/02/2004	01:18:16.837	41	1	19.0868	38.8039	43.2400	2.6996
	28/02/2004	01:19:16.999	42	1	19.0672	38.9061	43.2400	2.6874
	28/02/2004	01:20:17.164	43	1	19.0672	38.8720	43.2400	2.6996

Figure 53. Aperçu de la structure du DataSet d'Intel (2004)

Hypothèse

Dans les conditions normales d'observation, les relevés de données issues des capteurs représentent les valeurs les plus fiables. Un intervalle de rangement de ces valeurs suffit ainsi à prendre connaissance de l'état de cet environnement. Par contre, au-delà de ces intervalles, les valeurs sont sujettes à conjecture.

Une discrétisation par intervalle de données agrégera les transitions en observation. La décision de mise à jour est une fonction booléenne « F_s » basée sur un intervalle de confiance ; on utilisera la méthode « Exactly same » des travaux de [98], sur des intervalles et non sur des valeurs observables :

- (F_s) ne doit, en aucun cas, être indéterminée.
- Dans toutes les situations (F_s) doit fournir une solution $\in \{0,1\}$.

L'objectif de la station de base est de s'informer des valeurs relevées dans le réseau à intervalles réguliers afin de détecter un éventuel changement. Sur les données issues du dataSet d'Intel, l'intervalle des relevés est de trente et une seconde (31s), et le relevé d'une journée effectuée par l'équipe du laboratoire variant entre 1316 et 2800 valeurs.

Une perspective de transmettre ce nombre, pendant une année, consisterait donc à envoyer un nombre de valeurs d'environ $\cong (3600/30) \times 24 \times 365 = 10\,512\,000$ valeurs. Ce nombre est calculé pour un seul paramètre : la température ; toutefois, si nous prenons en considération l'humidité, la lumière et la pression, la taille du paquet augmentera forcément, ce qui influera obligatoirement sur la durée de transmission. Il en résulte que notre approche est basée sur la réduction de la taille du paquet et la transmission des données en utilisant une méthode de discrétisation issue des statistiques descriptives (figure 54), ce qui permettra de réunir des données captées sur des ensembles.



Figure 54. Schéma résumant les statistiques descriptives

Définition 2

Une série statistique se définit par une population et un caractère :

La population est l'ensemble des individus (ici : les capteurs)

Le caractère est la qualité étudiée chez ces individus. (Ici : température, humidité, lumière, tension, etc.). De plus, un caractère peut-être :

- Qualitatif, (valeurs non numériques : couleur, forme, etc.).
- Quantitatif : à savoir
 - Discret (nombre fini de valeurs naturelles)
 - Continu (infinité de valeurs) : notre cas d'études.

La discrétisation met en évidence la question du nombre de classes (k) à créer. Par définition, « *son objectif est de perdre le moins d'information possible tout en retranscrivant au mieux l'organisation statistique des données* ».

Nos données constituent de la sorte un caractère quantitatif continu. Ainsi, l'établissement du tableau de fréquences implique d'abord une répartition en classes de ces données. Cela nécessite de définir, en premier, le nombre de classes puis l'amplitude de chacune d'elles. En règle générale, le choix des classes doit être de même amplitude. Il est à noter cependant qu'une distribution en fréquences est explicative si – et seulement si – chaque classe comprend un nombre suffisant de valeurs captées « v_i ». Plusieurs méthodes expérimentales permettent d'établir le nombre de classes pour un échantillon de taille « n ».

V.4.3.3.2. Phase 1 : Classification

1^{ère} étape : Identifier le nombre de classes (k)

Définition 3

- Le nombre de classes (k) et la formule qui y est associée dépendent du type de variables. Quel que soit le cas, deux règles s'imposent :
- Le nombre de classes est arrondi à l'entier le plus proche.
 - Le nombre de classes doit s'inscrire dans l'intervalle [5,8].

Diverses formules empiriques existent afin de classer les données; ainsi, pour les variables continues :

- Formule de Sturge : $k = \text{Log}2(n)$
- Formule de Sturge-Huntsberger : $k = 1 + (10/3) \cdot \text{Log}10(n)$
- Formule de Brooks-Carruthers : $k = 5 \cdot \text{Log}10(n)$
- Formule de Scott: $k = (\text{valeur min} - \text{val max}) / 3,5 \cdot \sigma \cdot n^{(-1/3)}$
(cas de distribution asymétrique)

Application

Notre choix s'est porté sur la formule de Sturge-Huntsberger¹, étant donné la distribution de l'échantillon élu :

¹ La règle de Sturges : formule mathématique proposée par Herbert Sturges (1882-1958).

$$k = 1 + (10/3) * \text{Log}_{10}(n) \quad (27)$$

Si nous prenons en exemple un échantillon de valeur $n = 1000$, le nombre de classes significatif serait :

$$k = 1 + 3.33 \cdot \text{Log}_{10}(1000) = 10.99 \approx 11 \text{ classes (en application de la définition 2).}$$

De cette formule, nous déduisons la méthode de classification en choisissant $k = 5$ classes pour un contrôle supervisé et une meilleure visibilité.

2^{ème} étape : Analyse descriptive de l'échantillon

Notre échantillon est pris dans la base de données du laboratoire d'Intel Berkley [97]. Un intervalle de confiance est calculé sur la base de 5 Classes ($k=5$) en déterminant à titre d'exemple, le niveau de risque associé au danger. Nous relevons les valeurs minimale et maximale admises parmi les valeurs d'une journée jugées acceptables (valeurs prises dans des conditions normales d'observation) : exemple :

- $\text{Min}(vt_i) = 17,5090$
- $\text{Max}(vt_i) = 26,6328$
- Calculons l'écart par la formule de Sturge :
- $E = \text{Max}(vt_i) - \text{Min}(vt_i) = 26,6328 - 17,1954 = 9.4374$
- Calculons l'amplitude : $A = E/k = 9.4374 / 5 = 1.8874$

Une analyse statistique descriptive des données de température de notre échantillon « vt_i », nous révèle ainsi les données quantitatives suivantes (voir tableau 21):

Classes	Borne inf.	Borne Sup.	Effectif	Fréquence	Densité
1	[17.1954;	19.0829[2575	0.291	0.154
2	[19.0829;	20.9704[2106	0.238	0.126
3	[20.9704;	22.8579[1687	0.191	0.101
4	[22.8579;	24.7454[1618	0.183	0.097
5	[24.7454;	26.6328[857	0.097	0.051

Tableau 21. Intervalle de confiance – Mote 1: capteur de « Température »

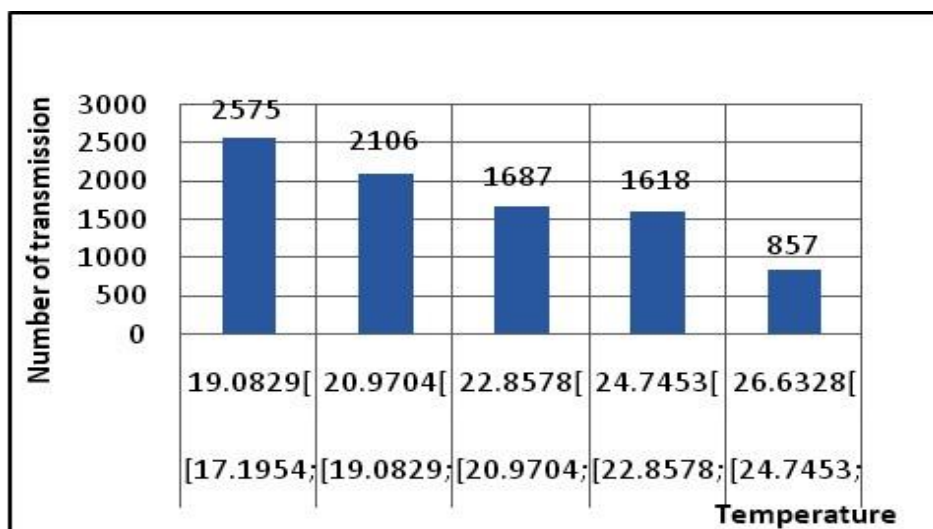


Figure 55. Nombre de transmissions par « Intervalle de Confiance – C.I.» (Capteur de Température)

3^{ème} étape :

De cette étape, les valeurs de température sont établies par intervalles de confiance (classes) ; exemple : Classe1 = [17,1954 ; 19,0829[, Classe1 = [19,0829; 20,9704[, etc. Sachant que l'on a recensé 5 classes, on met en marche les capteurs, et chaque température relevée (niveau caché) sera attribuée à la classe correspondante (niveau observable). De ce fait, si la valeur d'une température captée est cadrée dans une classe, le numéro de classe est transmis à la station de base pour une mise à jour. Si, par contre, la température relevée n'appartient à aucun intervalle de confiance, un niveau de classe « OutOf » (Out of range) sera transmis pour une mise à jour, soulignant ainsi un état d'alerte. De cette façon, la station de base pourra observer le mote et prévenir le niveau de danger, avant d'arriver au seuil de détérioration (Figure 56).

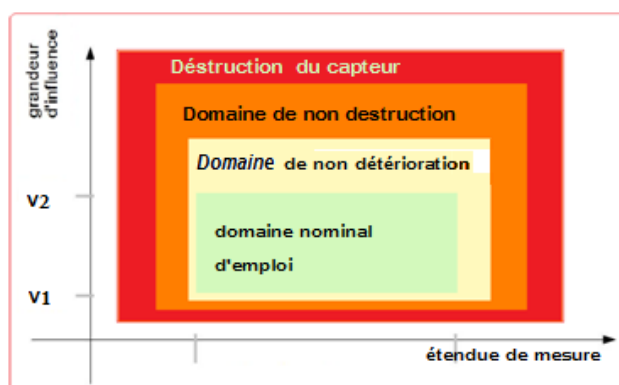


Figure 56. Limites d'utilisation d'un capteur et étendue de mesure [99]

Remarque

Le nombre de classes (k) peut être augmenté, ce qui élargira l'étendue des valeurs captées.

V.4.3.3.3. Phase 2 : Transmission et mise à jour des données

1^{ère} Etape : Structure du paquet transmis

Les données sont la partie la plus importante dans la transmission d'un capteur et le nœud capteur waspmote peut supporter jusqu'à 16 différents capteurs en même temps (voir caractéristiques). Pour ce type de mote, nous pourrions réserver 16 champs de données. Aussi, étant donné l'entête du paquet, si nous avons « n » capteurs nous aurons « n » champs de données en plus (Figure 55).

Notre paquet de données sera donc le même que le paquet décrit par le groupe du laboratoire Intel [97], mais en substituant un champ « N° classe » de chaque type de caractère du mote au champ des valeurs des capteurs. Le paquet se représentera comme suit :

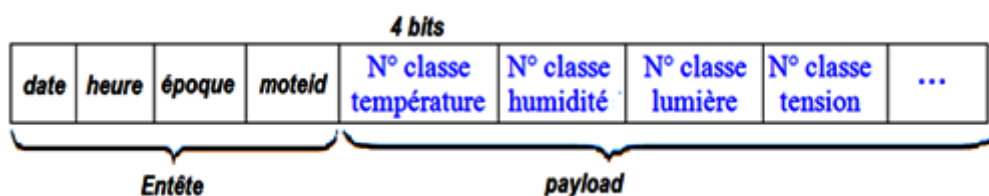


Figure 57. Structure d'un paquet du Laboratoire Intel modifiée [20]

On peut en déduire que pour un mote de 4 capteurs, nous avons besoin de $4\text{bits} \times 4 = 16$ bits.

- Le champ « époque » est la période de capture de ces grandeurs ; c'est un numéro de séquence croissant de façon monotone à partir de chaque mote.
 - Le champ « Moteid » : Un nombre séquentiel de capteurs.
 - Le champ N° classe contient « m » bits et dépend du nombre de classes k.
- Ici, $k = 5 = 2m$. Ceci donne $m = 3$ bits pour codifier 8 classes.

Nbr. Classes	N° classe température	Min (v_{t_i})	Max (v_{t_i})
0	0000	"OutOff"	
1	0001	[17.1954;	19.0829 [
2	0010	[19.0829;	20.9704 [
3	0011	[20.9704;	22.8579 [
4	0100	[22.8579;	24.7454 [
5	0101	[24.7454;	26.6328 [
7	0110	Réservés pour d'autres classes	
8	0111		

Tableau 22. Représentation binaire du contenu des champs

Connaissant l'amplitude de la température ($A= 1.8874$) dans le tableau ci-dessus, il est possible de rajouter d'autres classes (voir Tableau 23). On pourra alors choisir le nombre de bits y correspondant. Par exemple, pour $m = 4$, on peut avoir plus de 8 classes : $2^4 = 16$ classes.

N° classe température	Min (vt_i)	Max (vt_i)
0110	[26.6328;	28.5202 [
0111	[28.5202;	30.4076 [
1000	[30.4076;	32.2950 [
1001	[32.2950;	34.1824 [
1010	[34.1824;	36.0698 [
1011	[36.0698;	37.9572 [
1100	[37.9572;	39.8446 [
1101	[39.8446;	41.7320 [
1110	[41.7320;	43.6194 [
1111	[43.6194;	45.5068 [

Tableau 23 : Extension des classes en utilisant l'amplitude $A=1.8874$

NB. Il reste que les classes ajoutées sont prises dans les conditions d'acceptabilité des valeurs captées.

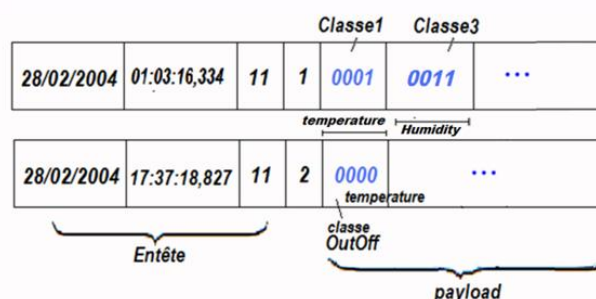


Figure 58. Exemple de paquets avec le caractère température

Remarque :

Dans notre cas d'étude, le choix des intervalles de confiance (au lieu des valeurs) contribue pleinement à réduire la taille du paquet : pour 4 capteurs par mote, nous aurons 16 bits de données. Ceci entraînera nécessairement une diminution des délais de transmission et de propagation dudit paquet, et par conséquent l'énergie consommée pour une transmission.

2^{ème} Etape : Algorithme de mise à jour

Algorithme mono-sensor

```

Inputs: classes(k), Sv(t), Sv(t-1), A //k=1..5; A : amplitude = 1,8874;
Outputs: Fs,i // Fs: Binary function to send data or Not send
Begin
Set Fs ← 0
Set Tf ← OFF
Set Classe(t-1) ← the last classed Sensor (S) value // Classe(t-1) : Class sensor at time "t-1"
Read Sv(t) // Sv(t) a sensor value at time t
// j: a counter of (k) classes
For j = 1 to k=5
    If Sv(t) in classe(j) then
        If classe(t-1) = classe (j) then // "Exactly-Same" comparison class method,
            Set Fs ← 0 // already in class
        Else
            Set Fs ← 1
        End if
    Else
        Set Classe(t-1) ← "0000" // "OutOff" range class - ALARM -
        Set Fs ← 1
    End if
End For
// sending data
IF Fs = 1 then
    RF-Transmit ← ON // sending a new class of data
Else
    RF-Transmit ← OFF
End if
End
    
```

3^{ème} Etape : Réalisation de l'interface

Nous avons mis en place un programme nommé « Sensoring » illustrant les quatre algorithmes décrits précédemment, à savoir « Exactly same - ES », « Absolute Difference - AD », « Relative Difference - RD » et naturellement notre algorithme « Confidence Interval - CI ».

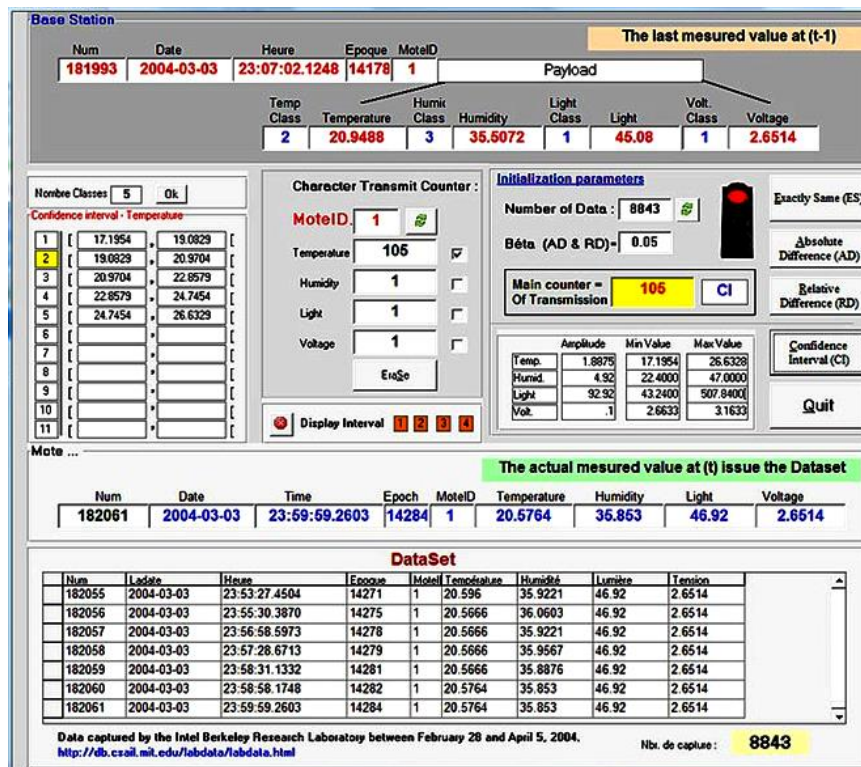


Figure 59. Interface du programme « Sensoring »

V.4.3.3.4. Résultat de l'expérimentation

L'évaluation de l'algorithme est effectuée sur un échantillon de données d'Intel Berkeley Research Lab (IBRL) [97]. L'expérience est prise au hasard sur plusieurs notes. Ainsi, on en a retenu 6 (sur 54) : soit les notes 1, 4, 7, 9, 18, 20 pour la présentation, mais les tests ont été effectués sur l'ensemble, et les résultats ont été très satisfaisants.

Notons que tous les notes comportent quatre (04) capteurs, et que les tests ont été pris en considérant toutes les combinaisons possibles des capteurs.

Les plans de tests sont représentés dans le tableau ci-dessous ; on choisira les dénominations suivantes :

Temp. (T): Température; Humid. (H): Humidité; Light (L): ; Lumière; Volt. (V) : Voltage

MoteID	1	(a)		
DataSet	8843 paquets			
Algorithmes...	ES	AD	RD	CI
Paramètres				
Temp. (T)	6679	1706	62	105
Humid. (H)	6596	4498	54	91
Light. (L)	1592	1592	777	163
Volt. (V)	2916	12	1	1
Temp.+ Humid.	8147	4718	80	193
Temp.+ Light.	6894	2678	802	261
Temp.+ Volt.	7363	1706	62	105
Humid.+ Light	6922	5095	794	249
Humid.+ Volt.	7365	4498	54	249
Light. + Volt.	4000	1592	777	163
Temp. + Humid. + Light	8207	5249	802	345
Temp.+Light+Volt	7512	2678	802	261
Humid. + Light + Volt	7575	5095	794	249
Temp.+Humid.+Light+Volt	8411	5249	802	345

MoteID	4	(b)		
DataSet	8754 paquets			
Algorithmes...	ES	AD	RD	CI

Paramètres				
Temp. (T)	6109	1291	51	67
Humid. (H)	6380	4275	51	96
Light. (L)	1043	1043	587	113
Volt. (V)	3053	9	9	2
Temp.+ Humid.	7893	4411	69	153
Temp.+ Light.	6316	1954	597	175
Temp.+ Volt.	7021	1291	51	68
Humid.+ Light	6595	4713	602	204
Humid.+ Volt.	7235	4275	51	204
Light. + Volt.	3692	1044	587	114
Temp. + Humid. + Light	7952	4821	602	259
Temp.+Light+Volt	7149	1954	597	176
Humid. + Light + Volt	7364	4713	602	205
Temp.+Humid.+Light+Volt	8234	4821	602	260

MoteID	7	(c)
DataSet	11969 paquets	

Algorithmes.....	ES	AD	RD	CI
Paramètres				
Temp. (T)	8128	1334	48	77
Humid. (H)	8180	5246	47	80
Light. (L)	1414	1414	754	109
Volt. (V)	4390	7	7	1
Temp.+ Humid.	10676	5423	64	153
Temp.+ Light.	6894	2678	802	184
Temp.+ Volt.	7363	1706	62	77
Humid.+ Light	6922	5095	794	187
Humid.+ Volt.	7365	4498	54	187
Light. + Volt.	4000	1592	777	109
Temp. + Humid. + Light	8207	6083	770	259
Temp.+Light+Volt	9814	2332	763	184
Humid. + Light + Volt	9792	5969	765	187

Temp.+Humid.+Light+Volt	11238	6083	770	259
-------------------------	-------	------	-----	-----

MoteID	9	(d)		
DataSet	9113 paquets			

Algorithmes...	ES	AD	RD	CI
Parametres				
Temp. (T)	6659	1371	59	78
Humid. (H)	6334	4298	57	118
Light. (L)	1343	1343	848	57
Volt. (V)	2378	10	1	260
Temp.+ Humid.	8264	4482	77	193
Temp.+ Light.	6906	2276	872	134
Temp.+ Volt.	7291	1371	59	334
Humid.+ Light	6690	4925	872	171
Humid.+ Volt.	7076	4298	57	171
Light. + Volt.	3362	1345	848	316
Temp. + Humid. + Light	8342	5042	884	246
Temp.+Light+Volt	7470	2276	872	390
Humid.+Light+Volt	7336	4925	872	429
Temp.+Humid.+Light+Volt	8529	5042	884	502

MoteID	18	(e)		
DataSet	8923 paquets			

Algorithmes...	ES	AD	RD	CI
Paramètres				
Temp. (T)	6082	1007	54	47
Humid. (H)	5734	3361	41	63
Light. (L)	1668	1600	914	161
Volt. (V)	2371	12	3	2
Temp.+ Humid.	7786	3493	63	108
Temp.+ Light.	6582	2239	939	205
Temp.+ Volt.	6821	1009	55	48

Humid.+ Light	6272	4262	936	221
Humid.+ Volt.	6583	3361	42	221
Light. + Volt.	3604	1603	914	162
Temp. + Humid. + Light	7986	4359	942	264
Temp.+Light+Volt	8093	3493	64	264
Humid.+Light+Volt	6999	4262	936	222
Temp.+Humid.+Light+Volt	8250	4359	942	265

MoteID	20	(f)
DataSet	9582 paquets	

Algorithmes...	ES	AD	RD	CI
Paramètres				
Temp. (T)	6831	1393	64	84
Humid. (H)	6510	4171	73	137
Light. (L)	1823	1823	1266	111
Volt. (V)	2125	17	10	12
Temp.+ Humid.	8467	4297	84	217
Temp.+ Light.	7256	2735	1290	194
Temp.+ Volt.	7415	1393	70	93
Humid.+ Light	6957	4953	1288	246
Humid.+ Volt.	7191	4171	75	246
Light. + Volt.	3561	1825	1255	122
Temp. + Humid. + Light	8603	5053	1295	326
Temp.+Light+Volt	7749	2735	1290	203
Humid.+Light+Volt	7543	4953	1288	255
Temp.+Humid.+Light+Volt	8820	5053	1295	335

Tableau 24 : Résultats Obtenus après l'exécution du programme « Sensoring »

pour les notes 1, 4, 7, 9, 18, 20

Nous représenterons ci-après les graphiques du tableau 24, afin d’avoir une comparaison claire des résultats obtenus dans les 6 mote avec les différents algorithmes comprenant toutes les combinaisons possibles entre les capteurs d’un mote.

		MoteID = 1			
DataSet		8843			
Sensor on Mote: 1- Temperature	Algorithm	ES	AD	RD	CI
	Temp. (T)	6679	1706	62	105
	Humid. (H)	6596	4498	54	91

Total number of data captured (Without any algorithm)
Type of algorithm applied
Number of data transmitted with the "ES" algorithm
Number of data transmitted by applying the "CI" algorithm

Figure 60 : Détail sur la lecture des tables

V.4.3.3.5. Analyse et Interprétation des résultats

Lorsque l’un des capteurs du mote a des relevés assez élevés (exemple : le capteur de lumière), l’écart des algorithmes de la Différence Relative « RD » et ceux de la Différence Absolue « AD » n’est pas important. Ceci est d’autant plus vrai lorsque l’on sait qu’une valeur de 1 lux correspond au clair de lune, 400 lux à un bureau lumineux et 100 000 lux au plein soleil [97]. (Voir les histogrammes ci-dessous)

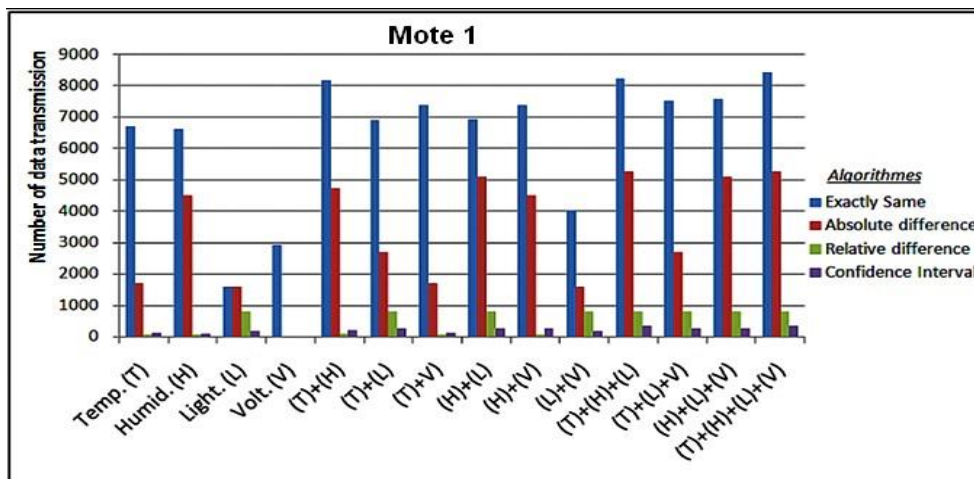


Figure 61. Comparaison des résultats du Mote 1 avec les 4 capteurs.

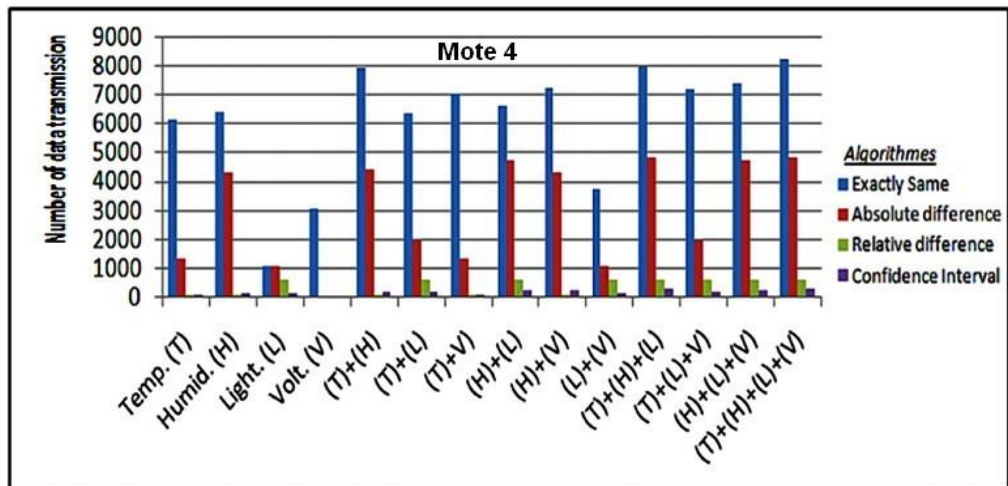


Figure 62. Comparaison des résultats du Mote 4 avec les 4 capteurs

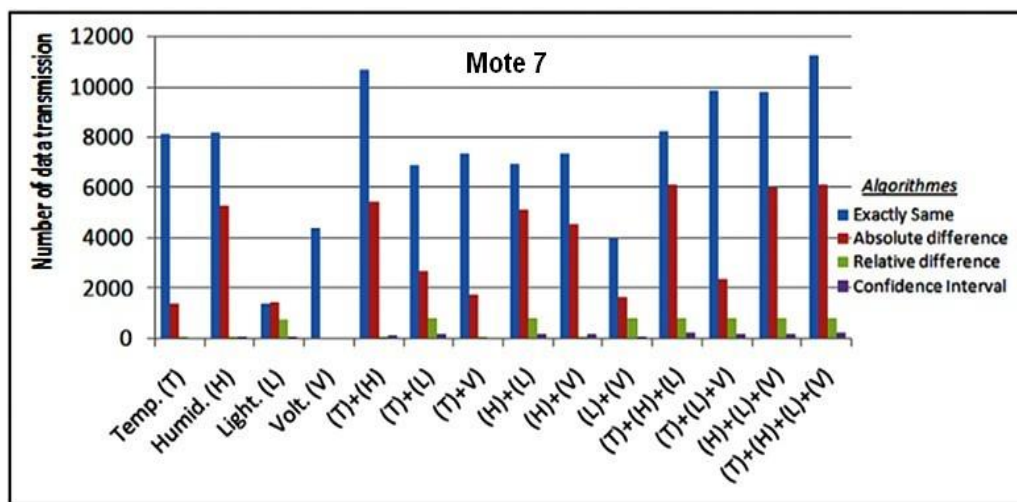


Figure 63. Comparaison des résultats du Mote 7 avec les 4 capteurs

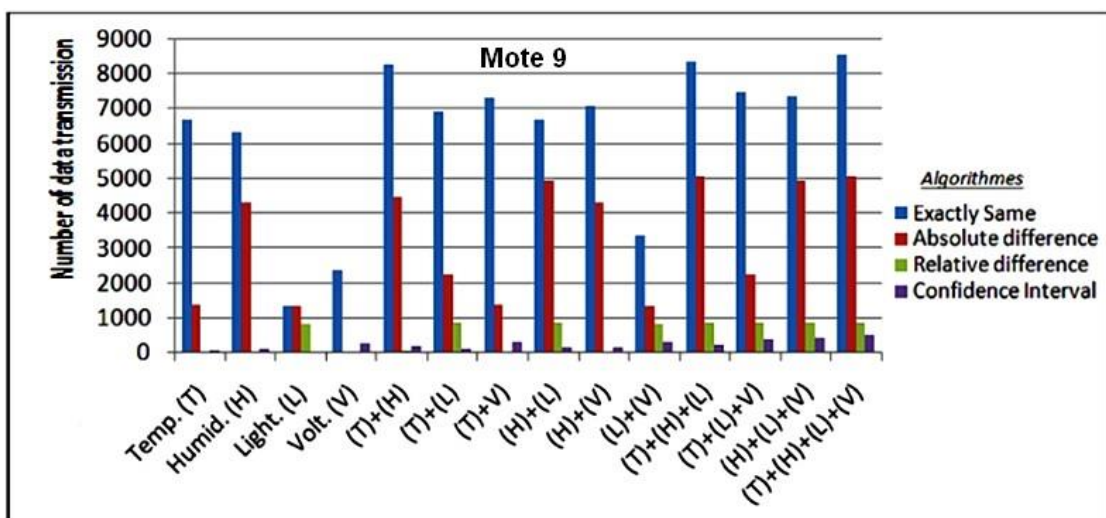


Figure 64. Comparaison des résultats du Mote 9 avec les 4 capteurs

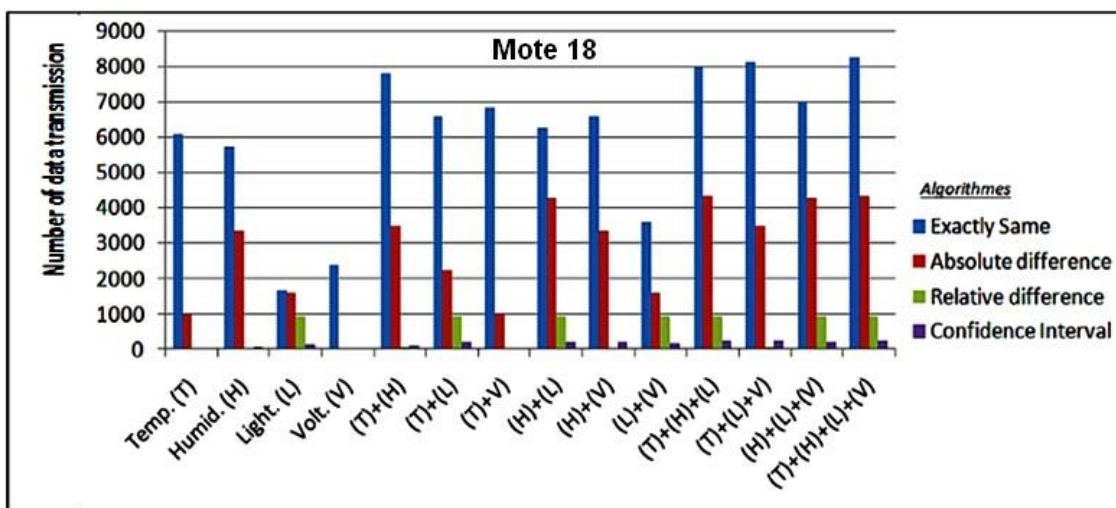


Figure 65. Comparaison des résultats du Mote 18 avec les 4 capteurs

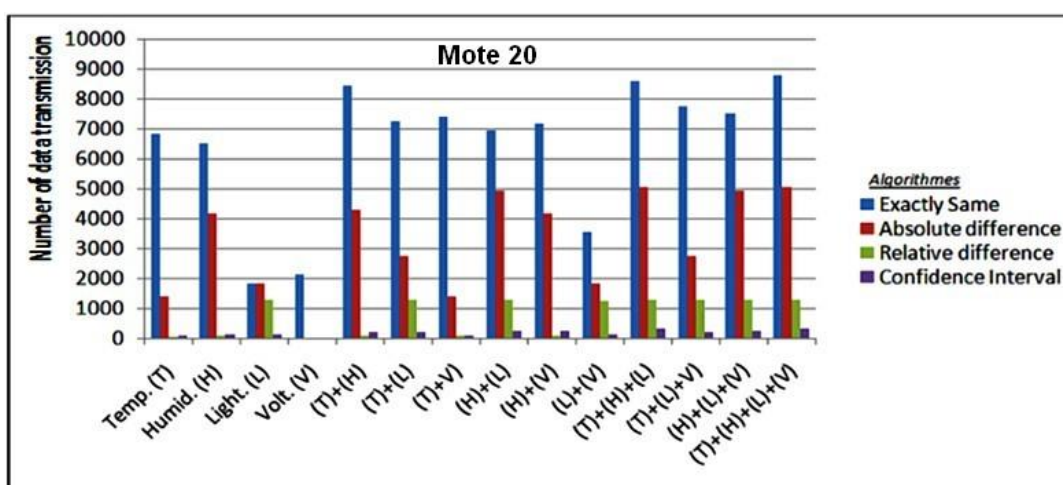


Figure 66. Comparaison des résultats du Mote 20 avec les 4 capteurs

Le tableau 24 ci-dessous engendre des valeurs avec des motes comprenant 4 capteurs réunis, déployés sur une architecture basée sur l’IoT, illustrant ainsi la consommation d’énergie pendant une durée de deux mois. Ces valeurs calculées se présentent à titre d’exemple :

Mote 1 (avec l’algorithme « ES ») : nous enregistrons 8411 transmissions, soit :

$$E = 8411 \times 59.2 \times 2 = 9,958624 \times 10^5$$

Algo. \ Motes	ES	AD	RD	CI
Mote 1	9,958624	3,107408	0,474784	0,204
Mote 4	4,874528	2,854032	0,356384	0,154
Mote 7	6,652896	3,601136	0,45584	0,153
Mote 9	10,098336	5,969728	1,046656	0,594
Mote 18	9,768	5,161056	1,115328	0,314
Mote 20	10,44288	5,982752	1,53328	0,397

Tableau 24. Consommation d’énergie par mote

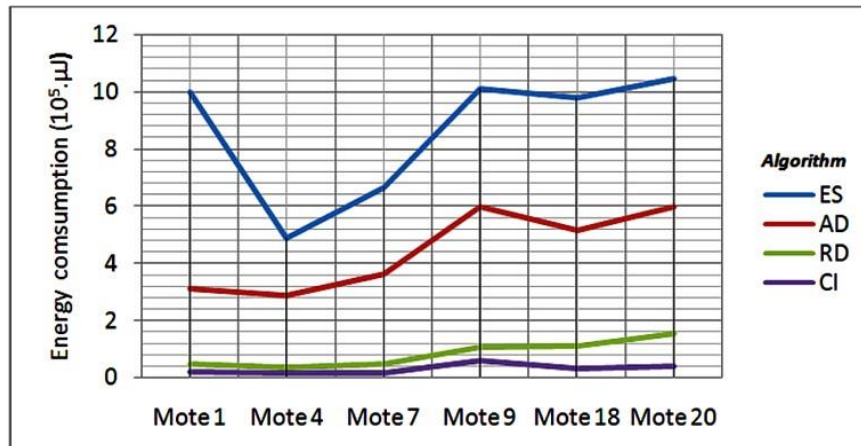


Figure 67. Energie consommée sur chaque mote avec application des 4 algorithmes

L'algorithme « Sensoring », basé sur les intervalles de confiances, affiche une grande performance dans les résultats, et par conséquent une économie d'énergie beaucoup plus importante. Pour ce faire, laissons la valeur de la taille de l'entête inchangée et reprenons la valeur du coût d'envoi d'un octet à 59,2 μJ (comme défini dans [95] pour le mote MICA2Dot). Ainsi la taille des données obtenues (payload) vaudra : 2 octets = 2 x 8 = 16 bits pour les 4 capteurs d'un mote (voir Figure 55), face aux 64bits de [95]. Cette taille influera largement et positivement sur la durée d'émission du paquet pour une mise à jour.

V.4.3.3.6. Conclusion

L'analyse des résultats confirme donc l'hypothèse que nous avons proposée, à savoir que le programme « Intervalle de confiance » consomme moins d'énergie, comparée aux autres algorithmes. Cette performance peut être exploitée pour une coopération optimale des nœuds, sur un déploiement déterministe afin de prolonger au maximum la durée de vie d'un réseau.

Conclusion générale

Ce travail met en évidence quelques règles de déploiement des nœuds en s'appuyant sur la détection et les communications coopératives des nœuds dans les réseaux de capteurs sans fil, car la plus part des recherches se sont concentrées sur le routage qui reste sans doute important dans un tel réseau. Par nature, un nœud de capteurs est limité en énergie, en bande passante et possédant une petite mémoire de stockage. Ainsi, minimiser la consommation d'énergie est un défi majeur. L'étude spatio-temporelle du meilleur chemin en est une solution.

Dans cette thèse, nous nous sommes focalisés plutôt sur la façon dont les données sont collectées et traitées par les nœuds de capteurs afin d'augmenter la fiabilité de la masse d'informations et réduire la communication global des données, car les nœuds étant souvent déployés de manière dense et les données collectées par les nœuds voisins sont : soit redondantes, soit avoisinantes. Une bonne gestion de donnée permettra de minimiser le nombre de capteurs afin d'optimiser un déploiement déterministe. Notre travail est composé de trois parties principales réalisé sur un simulateur:

1. Une étude observatoire du nombre de capteurs sur un déploiement déterministe, cette première partie vise à atteindre une corrélation spatiale entre les nœuds et d'en tirer les règles optimales de déploiement.
2. La seconde est une tâche de détection et de communications coopératives des données visant à trouver des conditions optimales de collecte.
3. Une troisième étude, toujours centrée sur les données, mais appliquée sur des données réelles permettant une détection et une coopération des nœuds et visant à minimiser la taille des paquets et la fréquence d'envoi des données aux capteurs voisins ou au puits.

En effet, nous avons utilisé Contiki/Cooja pour réaliser la simulation de la partie d'observation, où une multitude de plans avec une variation spatiale des nœuds sur des zones bien définie et précises a été appliquée (déploiement statique et déterministe). Une fois la simulation lancée et s'appuyant seulement sur le protocole RPL, nous soulignons que :

- Plus le déploiement des nœuds est dense plus le temps de remise de paquet au sink augmente. Cela est dû au trafic de donnée élevé causant une congestion du réseau et par conséquent un retard ou perte de paquets.
- La consommation d'énergie est proportionnelle à la densité de nœuds déployés dans le réseau.
- Un faible déploiement des nœuds dans un grand espace minimis

- e la couverture du réseau et accroît le temps de remise de paquet voir même une absence de réponse.
- Augmenter la qualité joue un rôle important dans le placement des nœuds mais risque de diminuer la durée de vie d'un réseau de capteurs.

La deuxième partie de notre contribution repose sur une architecture de RCSF à deux niveaux. Un niveau master représenté par le puits, et un niveau inférieur basé sur des capteurs agrégateurs. Nous proposons un nouvel algorithme, de faible complexité, de coopération utilisant un consensus entre trois nœuds voisins où la moyenne optimale par paire est prise comme valeur élue à relayer au puits. Le déploiement des nœuds est statique. La valeur ajoutée à notre topologie est la nature des nœuds utilisés qui influe considérablement sur la communication coopérative des nœuds du réseau. Ainsi, des nœuds agrégateurs sont introduits formant des clusters-Head donnant une structure hiérarchique à notre réseau.

Cet apport permet de partitionner le réseau principal en plusieurs sous-réseaux utilisant des nœuds ayant des radios fréquences bidirectionnelle vers leur agrégateur. Les résultats obtenus soulignent les confirmations suivantes :

- La structure hiérarchique basée sur les cluster-Head résout énormément le problème de congestion du réseau, en minimisant les interférences entre les nœuds.
- Le déploiement est d'autant plus structuré, en se basant sur des agrégateurs.
- La solution aux problèmes de déploiement est principalement menée par les exigences physiques (distance entre les nœuds, l'espace à contrôler...) et les besoins de l'application convoitée (Nombre de nœuds à positionner, complexité de l'application, diffusion de paquets, optimiser le nombre de sauts,...).
- Le coût financier est proportionnellement lié au nombre de nœuds agrégateurs utilisé dans le réseau.

Une troisième et dernière contribution, marquant la communication et coopération des nœuds de capteurs sur des données réelles issues du laboratoire d'Intel Berkley ou nous avons conçu un algorithme « Intervalle de confiance » où un consensus d'alerte est prédéfini par la station de service. Une classification statistique de données utilisant une analyse descriptive renfermant une marge inférieure et une marge supérieure formant un intervalle ou classe de données permettant d'optimiser :

- Les fréquences de transmission,
- La congestion en minimisant la quantité de données circulant dans le réseau.
- Réduction de la taille des paquets.
- Le coût financier du réseau.
- La durée de vie.

La méthode ainsi conçu est prometteuse car en amplifiant le nombre de classe, l'approche donne, aussi, de très bon résultat en termes de tolérance de fautes. En rapport avec de

déploiement, cette méthode peut être utilisée dans le contexte de l'internet des objets où les nœuds sont concentriques autour d'un seul puits de données afin de surveiller une zone d'intérêt.

Dans cette thèse, nous avons décrit la manière de déployer un réseau de capteurs opérationnel en se basant sur une architecture hiérarchique statique utilisant des agrégateurs et moyennant un consensus centré sur les données qui nous permis de dégager des recommandations afin d'optimiser la mise place d'un réseau.

Perspectives

Notre proposons de continuer travail avec l'utilisation de plusieurs puits, car le nombre puits pourrait influencer le résultat de notre approche. Le choix d'un déploiement en sous-zones de surveillance optimisera sans doute les résultats. Mais cette démarche sera marquée sûrement par un coût financier élevé. Une autre perspective est de voir le fonctionnement du réseau avec un sink mobile sur un circuit prédéfini. De ce fait les nœuds sont contraints de diffuser les données en Broadcast et cette méthode n'est pas conseiller, car par cette inondation du réseau on augmentera la probabilité de congestion et par conséquent la durée de vie du réseau.

Bibliographie

- [1] IEEE 802.15.4TM, IEEE Standard for local and metropolitan area networks – Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs), 2011.
Available from: <http://standards.ieee.org/getieee802/download/802.15.4-2011.pdf>
- [2] C. ZARADER, « Le protocole sans-fil ZigBee/IEEE 802.15.4 et ses applications », REE N°10, pp. 91-98, Revue de l'Electricité et de l'Electronique, novembre 2004.
- [3] A. Mainwaring, d. Culler, j. Polastre, r. Szewczyk, j. “Anderson. Wireless sensor networks for habitat monitoring. In RCSFA'02”. Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications, pp. 88–97 New York, NY, USA, 2002. ACM
- [4] S. Tilak, N. B. Abu-Ghazaleh and W. Heinzelman, “A Taxonomy of Wireless Microsensor Network Models”. ACM Mobile Computing and Communications Review, Vol. 6(2), 2002, pp. 28-36. Doi: 10.1145/565702. 565708
- [5] J. Yick, B. Mukherjee, D. Ghosal. “Wireless sensor network-survey”. Comput. Netw., 52, pp. 2292-2330, August 2008.
- [6] T. Taleb, F. Naït-Abdesselam, A. Jamalipour, K. Hashimoto Nei Kato, and Y. Nemoto. Design Guidelines for a Global and Self-Managed LEO Satellites-based Sensor Network. In GLOBECOM, 2006.
- [7] W. Heinzelman, J. Kulik, and H. Balakrishnan, “Adaptive Protocols for Information Dissemination in Wireless Sensor Networks,” in Proceedings of the Fifth Annual ACM/IEEE International Conference on Mobile Computing and Networking (Mobi-Com '99), pp. 174-185, 1999.
- [8] J. Heidemann, F. Silva, C. Intanagonwiwat, R. Govindan, D. Estrin, and D. Ganesan, “Building efficient wireless sensor networks with low-level naming.,” in Proceedings of the Eighteenth ACM Symp. on Operating Systems Principles [21], pp. 146–159, 2001.
- [9] W. Heinzelman, Application-Specific Protocol Architectures for Wireless Networks, Ph.D. thesis, Massachusetts Institute of Technology, 2000.
- [10] A. Cerpa, J. Elson, D. Estrin, L. Girod, M. Hamilton, and J. Zhao. “Habitat monitoring: Application driver for wireless communications technology,” In Proceedings of the 2001 ACM SIGCOMM Workshop on Data Communications, April, 2001.
- [11] D. Dhoutaut, A. Regis, F. Spies. “Impact of radio propagation models in vehicular ad hoc networks simulations”. In Proceedings of the 3rd international workshop on Vehicular ad hoc networks, pp. 40-49. ACM, 2006.
- [12] T.S. Rappaport. “Wireless Communications”, 1st edn. Upper Saddle River, NJ: Prentice Hall, 1996.

- [13] P. Tuset-Peiro, A. Angles-Vazquez, J. Lopez-Vicario, X. Vilajosana Guillen, “On the suitability of the 433 Mhz band for m2m low-power wireless communications”. Propagation aspects. Transactions on Emerging Telecommunications Technologies, 25(12), pp. 1154-1168, 2014.
- [14] M. Zuniga, B. Krishnamachari. «Analyzing the transitional region in low power wireless links”. In Proc. First Annual IEEE Communications Society Conf. Sensor and Ad Hoc Communications and Networks IEEE SECON 2004. pp. 517-526, 2004.
- [15] M. Thomas, R. Wattenhofer, A. Zollinger. Topology control meets sinr : the scheduling complexity of arbitrary topologies”. In Proceedings of the 7th ACM international symposium on Mobile ad hoc networking and computing. MobiHoc’06. ACM. New York, NY, USA. pp. 310-321, 2006.
- [16] R.J. Quails, J.M. Scott, W.B. DeOreo. “Soil moisture sensors for urban landscape irrigation: Effectiveness and reliability”. JAWRA Journal of the American Water Resources Association, 37(3), pp. 547-559, 2001.
- [17] C. Intanagonwiwat, R. Govindan, and D. Estrin, “Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks,” Proc. 4th ACM International Conference on Mobile Computing and Networking (Mobicom’98), 2000.
- [18] “IETF MobileIP Working Group Internet Draft”, 1996.
Available from: <http://www.ietf.org/rfc/rfc2002.txt>.
- [19] W. Heinzelman, A. Chandrakasan, H. Balakrishnan. “Energy-Efficient Communication Protocol for Wireless Micro sensor Networks”. In proceeding of the Hawaii International Conference on Systems Science, vol. 8, pp. 8020, 2000.
- [20] S. Madden, R. Szewczyk, M.J. Franklin, D. Culler. “Supporting aggregate queries over ad-hoc wireless sensor networks”. In WMCSA’02: Proceedings of the Fourth IEEE Workshop on Mobile Computing Systems and Applications, page 49, Washington, DC, USA, 2002. IEEE Computer Society.
- [21] Y. Benmaissa. “Contribution à la modélisation et à la vérification de réseaux sans fil”. Thèse de doctorat, Université Mohammed V Agdal, Rabat, 2013.
- [22] X. Cheng, DZ Du, L. Wang, and B. Xu, “Relay sensor placement in wireless sensor networks,” ACM/Springer Journal of Wireless Networks, 14: pp. 347-355, 2008.
- [23] Y. Wu, M. Li, Z. Cai, and E. Zhu. “A distributed algorithm to approximate node-weighted minimum-connected [k]-coverage in dense sensor networks”. In Proceedings of the 2nd annual international workshop on Frontiers in Algorithmics, pp. 221–232, 2008.
- [24] H.M. Ammari, S.K. Das. “Coverage, connectivity, and fault tolerance measures of wireless sensor networks”. In Proceedings of the 8th International Symposium on Stabilization, Safety, and Security of Distributed Systems, pp. 35-49, 2006.
- [25] N.T. Hanh, H.T. Binh, N.X. Hoai, M..S. Palaniswami. “An efficient genetic algorithm for maximizing area coverage in wireless sensor networks”. Journal of information sciences, 488, pp. 58-75, 2019.

- [26] J.O. Rourke. Art gallery theorems and algorithms. Oxford University Press, Inc, Oxford, 1092, 1987.
- [27] C. Liu, K. Wu, Y. Xiao, B. Sun. "Random coverage with guaranteed connectivity: Joint scheduling for wireless sensor networks". IEEE Transactions on Parallel and Distributed Systems, 17, pp. 562-575, 2006.
- [28] S. Mnasri, A. Van Den Bossche, N. Nasri, T. Val, "The 3D Deployment Multi-objective Problem in Mobile RCSF: Optimizing Coverage and Localization" International Research Journal of Innovative Engineering. ISSN 2395-0560, vol. 1(5), 2015.
- [29] S. Mnasri, N. Nasri, A. Van Den Bossche, T. Val. "Simulation d'un réseau sans fil d'intérieur et des algorithmes NSGA-II et NSGA-III modifiés pour la résolution de la problématique de couverture et de localisation 3D". 2016.
- [30] A. Farhad, S. Farid, Y. Zia and F. B. Hussain, "A delay mitigation dynamic scheduling algorithm for the IEEE 802.15.4 based WPANs". International Conference on Industrial Informatics and Computer Systems (CIICS), Sharjah, pp. 1-5, 2016. doi: 10.1109/ICCSII.2016.7462430.
- [31] S.K. Gupta, P. Kuila, P.K. Jana. "Genetic algorithm approach for k-coverage and m-connected node placement in target based wireless sensor networks". Computers & Electrical Engineering. Volume 56, pp. 544-556, 2016. Available from: <https://doi.org/10.1016/j.compeleceng.2015.11.009>.
- [32] Zou, Yi & Chakrabarty, Krishnendu. "Coverage-Oriented Sensor Deployment". 2004.
- [33] N. Ait Saadi, "thesis: Multi-Objective Wireless Sensor Network Deployment," PIERRE & MARIE CURIE UNIVERSITY, Ecole Doctorale Informatique, Télécommunication et Electronique (EDITE de Paris), defense: March, 11th 2010.
- [34] X. Yu, N. Liu, W. Huang, X. Qian, and T. Zhang, "A Node Deployment Algorithm Based on Van Der Waals Force in Wireless Sensor Networks," Hindawi Publishing Corporation; International Journal of Distributed Sensor Networks; Volume 2013. Article ID 505710; <http://dx.doi.org/10.1155/2013/>
- [35] Z. Zhang, K. Long, J. Wang, F. Dressler, "On Swarm Intelligence Inspired Self-Organized Networking: Its Bionic Mechanisms, Designing Principles and Optimization Approaches," In IEEE Communications Surveys and Tutorials, 16(1), pp. 513-537, 2014. Doi: 10.1109/SURV.2013.062613.00014.
- [36] A. Katti, D.K. Lobiyal. Node Deployment Strategies and Coverage Prediction in 3D Wireless Sensor Network, Scheduling School of Computer and System Sciences, Jawaharlal Nehru University, New Delhi, India, 2017
- [37] A. Boualem, Y. Dahmani, A. Maatoug, C. De-runz. "Area Coverage Optimization in Wireless Sensor Network by Semi-random Deployment". In Proceedings of the 7th International Conference on Sensor Networks - Volume 1: SENSORNETS, ISBN 978-989-758-284-4, pp. 85-90, 2018. Doi: 10.5220/0006581900850090.

- [38] Y. Shen, D. T. Nguyen, and M. T. Thai, "Adaptive Approximation Algorithms for Hole Healing in Hybrid Wireless Sensor Networks". In Proceedings of the 32nd Int Conference on Computer Communications (INFOCOM), 2013.
- [39] Y. Wang, C. Hu, and Y. Tseng, "Efficient Deployment Algorithms for Ensuring Coverage and Connectivity of Wireless Sensor Networks". WICON Proceedings of the First International Conference on Wireless Internet, pp. 114-121, 2005.
- [40] Y. Zou and K. Chakrabarty. Sensor deployment and target localization in distributed sensor networks. Accepted for publication in ACM Transactions on Embedded Computing Systems, 2003.
- [41] A. Ben Hadj Mohamed. « Réseau de capteurs sans fil comportementaux pour l'aide au maintien à domicile par la surveillance en habitat intelligent ». Ordinateur et société [cs.CY]. Université Toulouse le Mirail - Toulouse II; Université de Sfax (Tunisie), 2015.
- [42] H. Bizagwira, « Réseau de capteurs sans fil étendu robuste exploitant des liens Atypiques ». Thèse de Doctorat en Informatique, Université Blaise Pascal Clermont, 2016.
- [43] M. J. McGlynn and S. A. Borbash. "Birthday protocols for low energy deployment and flexible neighbor discovery in ad hoc wireless networks". Proceedings of the 2nd ACM International Symposium on Mobile Ad Hoc Networking Computing, pp.137-145, 2001.
- [44] M. Cetin, L. Chen, J. W. Fisher, A. T. Ihler, R. L. Moses, M. J. Wainwright, et A. S. Willsky. "Distributed fusion in sensor networks. Signal Processing Magazine". IEEE, 23(4), pp. 42–55, July 2006. ISSN 1053-5888.
- [45] B. Benaissa, F. Lahfa. « Diffusion avisée après une collecte efficace de données par Intervalle de Confiance dans RCSF basé sur l'IoT ». International Journal of scientific Research & engineering technology (IJSET), 7(1), pp. 6-15, 2019.
- [46] M.C. Kerman, W. Jiang, A.F. Blumberg, S.E. Buttrey. "Event Detection Challenges, Methods, and Applications in Natural and Artificial Systems". In Proceeding of the International Command and Control Research and Technology Symposium, pp. 1-19, 2009.
- [47] J. Meng, H. Li, Z. Han. "Sparse event detection in wireless sensor networks using compressive sensing". In Proceeding of Conference on Information Sciences and Systems, pp. 181-185, 2009.
- [48] CT. Vu, R.A. Beyah, Y. Li. "Composite Event Detection in Wireless Sensor Networks". In Proceeding of the IEEE International Performance, Computing, and Communications. Conference, pp. 264–271, 2007.
- [49] B. Moulahi. "Définition et évaluation de modèles d'agrégation pour l'estimation de la pertinence multidimensionnelle en recherche d'information". Recherche d'information [cs.IR]. Université Toulouse III Paul Sabatier, 2015. Français. fftel-01249652f

- [50] Y. Yen, K. Huang, H. Chao, J.H. Park. "Tree-clustered data collecte protocol (TCDGP) for wireless sensor network". *Journal of the Chinese Institute of Engineers*, 32(7), 1025-1036, 2009. Doi : 10.1080/02533839.2009.9671589.
- [51] H. Mousannif, I. Khalil. "Cooperative Data Management in Wireless Sensor Networks", *International Journal of Computational Intelligence Systems*, 5:3, pp. 403-412, 2012. Doi:10.1080/18756891.2012.696890
- [52] Jin Cui. "Data Aggregation in Wireless Sensor Networks". *Networking and Internet Architecture [cs.NI]*. INSA Lyon, 2016. English. fftel-01428983f
- [53] Hida Yozo, Huang Paul, Nishtala Rajesh. "Aggregation Query Under Uncertainty in Sensor Networks", 2003.
- [54] A. Fapojuwo and A. Cano-Tinoco, "Energy Consumption and Message Delay Analysis of QoS Enhanced Base Station Controlled Dynamic Clustering Protocol for Wireless Sensor Networks," *IEEE Trans. Wireless Comm.*, vol. 8(10), pp. 5366-5374, 2009.
- [55] H. Li, C. Wu, Q.-S. Hua, and F. C. Lau, "Latency-minimizing data aggregation in wireless sensor networks under physical interference model," *Ad Hoc Networks*, vol. 12, pp. 52-68, 2014.
- [56] E. Shih, S. 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*. ACM SIGMobile, 2001.
- [57] S.H. Cho, A. Wang, C.G. Sodini, A.P. Chandrakasan. "Energy-Efficient Modulation and MAC for Asymmetric RF Microsensor Systems". *ACM ISLPED*. pp. 106-111, 2001.
- [58] S. Lindsey, C. Raghavendra. "Pegasis: Power-Efficient Gathering in Sensor Information Systems". *Int Conf Comm*, 2001.
- [59] A. El Gamal, C. Nair, B. Prabhakar, E. Uysal-Biyikoglu and S. Zahedi, "Energy-efficient scheduling of packet transmissions over wireless networks". *Proceedings. Twenty-First Annual Joint Conferences of the IEEE Computer and Communications Societies*, vol. 3, pp. 1773-1782, 2002. Doi: 10.1109/INFCOM.2002.1019431.
- [60] R. Shah, S. Roy, S. Jain and W. Brunette, "Data mules: modeling a three-tier architecture for sparse sensor networks," *IEEE SNPA*, 2003
- [61] J. Jeong, D. Culler, and J. Oh. "Empirical Analysis of Transmission Power Control Algorithms for Wireless Sensor Networks". *IEEE INSS*, 2007.
- [62] L. Myeong-Hyeon, C. Yoon-Hwa. "Distributed diagnosis of wireless sensor networks". *IEEE Region 10 Conference, Hongik, Seoul* pp. 1-4, 2007.
- [63] J. Mou, H. Ge. "Adaptive consensus of wireless sensor networks under local measurement by a sink node". *International Journal of Control*, 88(2), pp. 256-263, 2015. Doi: 10.1080/00207179.2014.948916.

- [64] S. Manfredi, E. Di Tucci. “Distributed control for energy-efficient and fast consensus in wireless sensor networks”. *International Journal of Control*, 90(5), pp. 1090-1103, 2017. Doi: 10.1080/00207179.2016.1201218.
- [65] P. Xu, C. Nowzari, Z. Tian. “A class of distributed event-triggered average consensus algorithms for multi-agent systems”. *International Journal of Control*. 2020. Doi: 10.1080/00207179.2020.1799244.
- [66] Zhenyu Lu, Ning Wang & Chenguang Yang. A novel iterative identification based on the optimized topology for common state monitoring in wireless sensor networks, *International Journal of Systems Science*, 2021. Doi: 10.1080/00207721.2021.1936275
- [67] D. Kingston, R. Beard, “Discrete-time average consensus under switching network topologies”. In *Proceedings of American Control Conference (ACC)*. Minneapolis, Minnesota, USA, 2006.
- [68] A. Hanaf. « Algorithmes distribués de consensus de moyenne et leurs applications dans la détection des trous de couverture dans un réseau de capteurs. Traitement du signal et de l’image [eess.SP] ». Université de Reims Champagne-Ardenne, 2016.
- [69] R. Olfati-Saber, R.M. Murray. “Consensus problems in networks of agents with switching topology and time-delays. *IEEE Transactions On Automatic Control*, 49(9), pp.1520–1533, 2004. (Cité en pages 1et 36.)
- [70] R. Olfati-Saber, J.A. Fax, R.M. Murray. “Consensus and Cooperation in Networked Multi-Agent Systems , ” *Proceedings of the IEEE*, Vol. 95(1), pp. 215-233, 2007.
- [71] O. Hlinka, O. Sluciak, F. Hlawatsch, P.M. Djuric, M. Rupp. “Likelihood Consensus: Principles and Application to Distributed Particle Filtering,” *Asilomar Conference on Signals, Systems, and Computers*, Pacific Grove (CA), USA, 2010.
- [72] O.Hlinka, O.Sluciak, F.Hlawatsch, P.M.Djuric, and M.Rupp, ”Distributed Gaussian particle filtering using likelihood consensus,” *Proc. of ICASSP 2011*, Czech Republic, 2011.
- [73] Alain Kibangou. « Consensus de moyenne dans un réseau de capteurs sans fil sécurisé ». *CIFA 2012 - 7ème Conférence Internationale Francophone d’Automatique*, Grenoble, France, 2012.
- [74] D. Bouyssou, T. Marchant, M. Pirlot, A. Tsoukiàs, P. Vincke. “Evaluation and Decision Models with Multiple Criteria”. *International Series in Operations Research & Management Science*. Kluwer Academic Publishers; 2006; Available from: <http://dx.doi.org/10.1007/0-387-31099-1>
- [75] R. Olfati-Saber, E. Franco, E. Frazzoli, J.S. Shamma, “Belief consensus and distributed hypothesis testing in sensor networks”. *Network Embedded Sensing and Control*, pp.169–182, 2006.
- [76] S. Kouachi, S. Dhuli, Y.N. Singh, “Convergence rate analysis for periodic gossip algorithms in wireless sensor networks. *arXiv preprint arXiv:1805.02430*, 2018.

- [77] J. Liu, S. Mou, A.S. Morse, B.D. Anderson, C. Yu, "Deterministic gossiping," *Proceedings of the IEEE*, vol. 99(9), pp. 1505-1524, 2011.
- [78] Stephen Boyd, Arpita Ghosh, Balaji Prabhakar, and Devavrat Shah. Randomized gossip algorithms. *IEEE Transactions On Information Theory*, 52(6) : pp. 2508–2530, 2006.
- [79] T.C. Aysal, ME. Yildiz, A.D. Sarwate A. Scaglione. "Broadcast gossip algorithms for consensus". *IEEE Transactions on Signal Processing*, 57: pp. 2748-2761, 2009.
- [80] B. Nazer, A.G. Dimakis, M. Gastpar. "Neighborhood gossip: Concurrent averaging through local interference". In *ICASSP*, 2009.
- [81] J.N. Al-Karaki, A.E. Kamal, "Routing techniques in wireless sensor networks: a survey". In *IEEE wireless communications*, 11(6), pp. 6-28, 2004.
<https://doi.org/10.1109/MWC.2004.1368893>
- [82] T. Winter, P. Thubert, A. Brandt, J.W. Hui, R. Kelsey, P. Levis, K. Pister, R. Struik, J.P. Vasseur, R. Alexander. "RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks". *RFC 6550*, pp.1-157, 2012.
- [83] J. Ren, S. Huang, W. Song, J. Han. "A Novel Indoor Positioning Algorithm for Wireless Sensor Network Based on Received Signal Strength Indicator Filtering and Improved Taylor Series Expansion". *Traitement du Signal*, 36(1), pp.103-108, 2019.
 Available from: <https://doi.org/10.18280/TS.360113>
- [84] R. Alrawashdeh, "Path Loss Estimation for Bone Implantable Applications". *Jordanian Journal of Computers and Information Technology (JJCIT)*, 4(2) pp. 94-101, 2018.
 Available from: <https://doi.org/10.5455/jjcit.71-1515153943>
- [85] R. Javaid, R. Qureshi, R.N. Enam. "RSSI based node localization using trilateration in wireless sensor network". *Bahria University Journal of Information & Communication Technologies (BUJICT)*, 8(2), pp. 58-64, 2015.
- [86] C. Hua, T.P. Yum, "Optimal routing and data aggregation for maximizing lifetime of wireless sensor networks". *IEEE ACM Trans. Netw.*, vol. 16(4), pp. 892-903, 2008.
- [87] H. Zhang, H. Shen, "Balancing energy consumption to maximize network lifetime in data-gathering sensor networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 20(10), pp.1526-1539, 2009.
- [88] I. S. AlShawi, L. Yan, W. Pan and B. Luo, "Lifetime Enhancement in Wireless Sensor Networks Using Fuzzy Approach and A-Star Algorithm". In *IEEE Sensors Journal*, vol. 12(10), pp. 3010-3018, 2012, Doi: 10.1109/JSEN.2012.2207950.
- [89] J.V.V. Sobral, J.J.P.C. Rodrigues, R.A.L. Rabêlo, J. Al-Muhtadi, V. Korotaev. "Routing Protocols for Low Power and Lossy Networks in Internet of Things Applications". *Sensors*, 19(9), pp.2144, 2019.
 Available from: <https://doi.org/10.3390/s19092144>
- [90] Contiki OS: Using Powertrace to estimate power consumption.
 Accessed on Jan. 04, 2021.

- [91] Instruments, T. Msp430f5438 datasheet. Reference SLAS655B, 2010.
- [92] Available from: <http://thingschat.blogspot.mk/2015/04/contikios-using-powertrace-and.html>. Accessed on Feb. 25, 2021
- [93] A Dunkels, J Eriksson, N Finne, N Tsiftes. “Powertrace: Network-level power profiling for low-power wireless networks”. 2011.
- [94] B. Yang, W. Wu, G. Zhu, “Distributed averaging in wireless sensor networks with triplewise gossip algorithms”. IEEE 2013 Tencon–Spring, pp. 178-182, 2013. Doi: 10.1109/TENCONSpring.2013.6584436.
- [95] N.A.M. Alduais, J. Abdullah, A. Jamil and L. Audah, “An efficient data collection and dissemination for IOT based RCSF”. In: IEEE 7th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON), pp. 1-6, 2016.
- [96] J.Y. Kim, C.H. Chu, and S.M. Shin, “ISSAQ: An Integrated Sensing Systems for Real-Time Indoor Air Quality Monitoring”, IEEE Sensors Journal, vol. 14(12), pp. 4230-4243, 2014.
- [97] Intel Lab Data website. Last modified: 06/2014.
Available from: <http://db.csail.mit.edu/labdata/labdata.html>. Accessed: 2018- 02- 13
- [98] A. Ghaddar, T. Razafindralambo, I. Simplot-Ryl, D. Simplot-Ryl, S. Tawbi and A. Hijazi, “Investigating Data Similarity and Estimation Through Spatio-Temporal Correlation to Enhance Energy Efficiency in RCSFs”. Ad Hoc & Sensor Wireless Networks, PKP Publishing Services Network, vol. 16(4), pp. 273-295, 2012.
- [99] Wikiversity website, "Capteur : Introduction ", 2017,
[Online].Available: <https://fr.wikiversity.org/wiki/Capteur/Introduction>.
Last Update. 01/08/2017. Accessed: 27 Jan 2018.
- [100] [100] : Cheick-Tidjane Kone. Conception de l’architecture d’un réseau de capteurs sans fil de grande dimension. Réseaux et télécommunications [cs.NI]. Université Henri Poincare - Nancy I, 2011.

Publication

Catég. A

Lien :

<https://www.iieta.org/journals/ts/paper/10.18280/ts.380303>



Traitement du Signal

ISSN:0765-0019 (print); 1958-5608 (online)

Indexing & Archiving: Web of Science, Science Citation Index Expanded, Journal Citation Reports, Scopus, SCImago (SJR), Ei Compendex, DBLP, MIAR, EBSCOhost, Publons, Science Open, Google Scholar, CrossRef, Portico, Microsoft Academic, CNKI Scholar, Baidu Scholar

Subject: Computer Sciences, Engineering



2020
Impact
Factor
2.589

JOURNAL METRICS	
Impact Factor (JCR) 2020:	2.589
5-Year Impact Factor:	1.947
CiteScore 2020:	2.1
SCImago Journal Rank (SJR) 2020:	0.279
Source Normalized Impact per Paper (SNIP) 2020:	1.414



Lien : <https://www.ieta.org/journals/ts/paper/10.18280/ts.380303>

Page:555-564 DOI:<https://doi.org/10.18280/ts.380303>

Received:15 April 2021 Accepted:3 June 2021 Published:30 June 2021

Detection and Cooperative Communications for Deployment Sensor Networks

Bedr-Eddine Benaissa, Fedoua Lahfa & al.

Detection and Cooperative Communications for Deployment Sensor Networks



Bedr-Eddine Benaissa^{1,2}, Fedoua Lahfa¹, Khatir Naima², Giulio Lorenzini³, Mustafa Inc^{4,5,6*}, Younes Menni⁷

¹ Department of Computer sciences, Abou Bekr Belkaid University, Tlemcen 13000, Algeria

² Department of Technology, University Centre of Naama - Salhi Ahmed, P.O. Box 66, Naama 45000, Algeria

³ Department of Engineering and Architecture, University of Parma, Parco Area delle Scienze, 181/A, Parma 43124, Italy

⁴ Department of Computer Engineering, Biruni University, Istanbul 34025, Turkey

⁵ Department of Mathematics, Science Faculty, Firat University, Elazig 23119, Turkey

⁶ Department of Medical Research, China Medical University Hospital, China Medical University, Taichung 40402, Taiwan

⁷ Unit of Research on Materials and Renewable Energies, Faculty of Sciences, Department of Physics, Abou Bakr Belkaid University, P.O. Box 119, Tlemcen 13000, Algeria

Corresponding Author Email: minc@firat.edu.tr

<https://doi.org/10.18280/ts.380303>

ABSTRACT

Received: 15 April 2021

Accepted: 3 June 2021

Keywords:

Wireless Sensor Network (WSN), clustering, Received Signal Strength Indicator (RSSI), IoT routing protocol

In a sensor network, and more specifically with a single-hop deployment policy, sensor measurements contain a lot of redundancy either in the measurement dimensions of a single sensor, or between the measurement dimensions of different sensors due to of the spatial correlation either in the temporal dimension of the measurements. The goal is to reduce this redundancy by deploying fewer sensors, while ensuring high measurement accuracy and maximizing service life. The proposed method minimizes the complexity in terms of communication and calculation and maximizes the lifetime of the network based on an aggregation and consensus system to reduce the spatio-temporal dimension of the data captured and consequently the number of sensors deployed. The results show a visible performance compared to the standard method of transmission on the free platform of the COOJA/Contiki simulator allowing to simulate network connections of wireless sensors and to interact with them.

1. INTRODUCTION

Searching in the field of wireless communication has given rise to miniaturized and intelligent equipment with capture, calculation and wireless communication units in a single circuit, with reduced dimension, and with a reasonable cost. These equipments, commonly known as micro-sensors, have favored the idea of developing sensor networks based on the collaborative effort of a large number of nodes operating autonomously and communicating with each other via short-range transmissions in the purpose of reporting an event according to the field of application (Figure 1).



Figure 1. TMote Sky sensor node platform [1]

A sensor is able to reap data relating to its environment, process them, and if necessary, communicate them to neighboring sensors via a wireless medium. The deployment

of this type of device then forms a network of wireless sensors that can be used in a variety of areas: scientists, military, industrial, health, environment, etc. This type of network consists of a few hundred or even thousands of autonomous sensors, of extremely reduced size and that work with very limited resources in terms of memory, energy and treatment capacities and communicate wirelessly with others in their communication zones.

The design and implementation of WNS are influenced by a several constraints among them we quote [2]:

1. Environment,
2. Deployment of nodes,
3. Heterogeneity of nodes,
4. Failure tolerance,
5. Mobility,
6. Scaling,
7. Material constraints,
8. Transmission,
9. Energy consumption, and,
10. Quality of service.

In a WSN, total coverage of the area to be monitored is one of the most asked questions in research. Thus, due to the strong spatial correlation of the detection area, there is an intensive deployment of sensor nodes which often detect and relay similar or redundant information and leading to more complex data analysis and a significant loss of resources for capturing, processing and memorizing them and consequently causing

the nodes to over-consume their transmission energies. Among the solutions to this problem, the design of a hierarchical topology was implemented by dividing the nodes into several levels of responsibility. One of the most widely used methods is clustering, where the network is partitioned into groups called "clusters". A cluster is made up of a cluster-head and its members.

In this perspective, our contribution lays out an algorithm based on an aggregation and consensus system to reduce the dimension of the values captured. This new type of mean consensus leads clustering-based routing to analyse and classify detection data and extract data characteristics to infer similar data, then decrease storage space, transmission load, and power consumption energy and simplify the ongoing data mining process.

2. RELATED WORK

Depending on the number of nodes of the network and the surface of the cover field, some nodes cannot directly transmit their messages to the collector node. Thus, the collaboration between the nodes to guarantee this transmission becomes a requirement. In this way, the messages are relayed by the intermediate nodes by establishing multi-jumping paths between a remote source and the well. This process of routing messages from a source node of the network to a recipient node is called routing. Thus, for a sensor network to be effective, the setting up of a routing algorithm becomes inevitable [3].

2.1 Routing concepts

There are two classes of routing protocols: distance vector routing and link state routing, so a routing protocol can be proactive or reactive [4].

2.1.1 Distance vector routing

This protocol class is based on an exchange between neighboring nodes and the node sending information about the list of destinations available to it and the corresponding cost. Routing protocols based on the most famous distance vector for ad hoc networks are: DSR, DSDV6, AODV and the new RPL protocol.

2.1.2 Link state routing

This protocol is known under the name of 'Shortest Path First'. This routing is based on the information collected and disseminated periodically on the state of the links in the network to update its description base describing a general view of the topology of the network. The main routing protocols in ad hoc networks that belong to this class are as follows: TORA, OLSR, etc.).

2.1.3 Proactive / reactive routing protocol

The proactive protocol builds its routing table before the request is made. A system of periodic exchange of control packets is put in place so that each node can build the network topology in a distributed fashion. The main proactive protocols are OLSR, FSR. On the other hand, a node using the reactive protocol completely ignores the topology of its network it updates its routing table only after having received a routing request from a neighboring node.

2.2 Detection in wireless sensor network

The trivial meaning of the term 'detection' is the process of capturing physical quantities related to environmental conditions (movement, temperature, humidity, light, etc.). The processing is based on the compatibility of the data captured by deploying different algorithms, either by comparing the previous state with the current state or using a consensus in order to eliminate the outliers and transmit a single common value to the sink.

2.2.1 Detection methods

1. Detection aggregation: based on statistical tests such as Maximum, Average [5] or a discretized classification of data [6] to aggregate the values captured over time;

2. Spatial and temporal correlation: Periklis et al. [7] list all the sensors in good condition and use them as references to test the other sensors. The simulation results show that this method consumes a lot of landfill and congestion. The use of cluster architecture is adopted to reduce this overhead [8];

3. Average consensus: The algorithms based on this principle conform to calculate the average between the recorded measurements and distribute it between neighboring sensors [9]; and

4. Consensus of beliefs: this consensus [10] is an algorithm derived from Bayes rule where a set of sensors try to harmonize on the most probable classification of an event.

2.2.2 Detection methods comparison

In Table 1, we present a classification of the protocols described in the previous section 2.1.

Table 1. Advantage and disadvantage of detection methods

Method	Advantage	Disadvantage
Detection aggregation	Reliable in the presence of faulty sensor readings.	Consumes any energy
Spatial and temporal correlation	Identifies sensors in good condition. For a tree architecture	Worn sensors. Communication and energy.
Consensus of the mean	No protocol. Low complexity algorithms.	Faulty sensors problem.
Consensus of beliefs	Classification model.	Complexities of calculations
Pair gossiping	Simple model. Scalability.	Converges slowly More iterations to converge.
Broadcast gossiping	Converges quickly. Scalability.	Consumes communications.
Gossip by triplet	Simplicity of calculations.	Controlled broadcast at 2 nodes only.
Gossip between neighbourhood	High performance Scalability.	Very high cost communication.

2.3 Simulator Cooja

Contiki OS Java Simulator is a sensor network emulator that can house a compiled program in order to test a scenario before loading it into flash memory of real nodes as in the TI MSP430 platform (Figure 2).

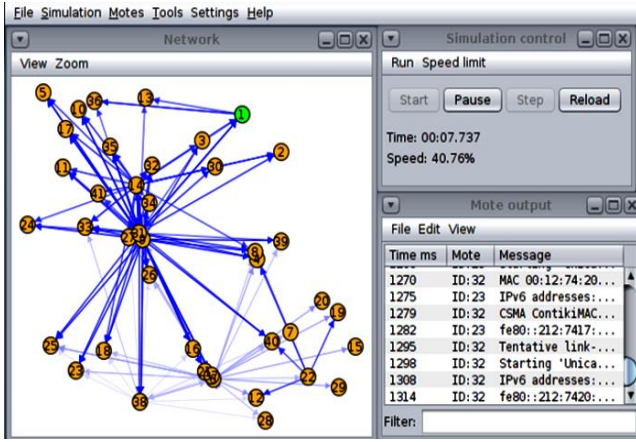


Figure 2. Interface graphique Cooja

2.4 Routing protocol - RPL

The basic principles of the protocol are [11, 12]:

1. Routing: the path discovery is based on the quality of the signal, the density of nodes and active path is represented by its costs;
2. Route selection: a reliable route relies on signal strength (RSSI);
3. Road quality criteria: signal strength (RSSI), number of hops, available energy and error rate (outliers);
4. Changing the path: the path initially traced can be corrected later depending on the strength of the signal and the total number of nodes visited in the event of failure of the main road; and
5. Constraint-based routing: The life of a node depends on the battery and diversity of node types and their characteristics (memory, location, collector, etc.).

2.5 Signal Strength (RSSI)

Received Signal Strength Indication informs us by a measurement on the intensity of the received signal Rx. It is an essential parameter for radiolocation applications, due to its simplicity and low cost, but its accuracy can be degraded by noise caused by stable, moving objects and obstacles or by diffraction or refraction of radio waves [13].

$$RSSI = - (10 * n * \log d + A) \quad (1)$$

n : Constant - signal propagation.

d : distance between the transmitter Tx and the receiver Rx.

A : Signal strength received for 1 meter distance.

So the distance is calculated from the RSSI:

$$d = 10^{((Abs(RSSI) - A) / 10 * n)} \quad (2)$$

According to Eq. (1), if A increases then RSSI increases. Thus the relation between the power of the transmitted signal Tx and the distance of the signal traveled " d " is exponential; but in reality this relationship is only as implicit in theory.

In this article we focus on a well-chosen configuration to collect values by exploiting the received signal power (RSSI).

3. PROPOSED WORK

Noting that a network of sensors is effective if is only if the

communication links are reliable. Alrawashdeh [13], underlines and experiments this theory on the implantable bodily applications in the bone, by applying a standard binding with a phase shift modulation (PSK).

In this paper, a detection method is proposed in order to capture physical quantities and transmit them to the base station (Sink) in a very short time, based on the processing time that each node can perform according to a scenario. Our system is based on the following three ordered tasks:

1. Detect physical quantities which are whole values of temperature and humidity taken at random and which change every 30 seconds.
2. Detect the sensor nodes neighboring the sensor node which wants to transmit these quantities.
3. Use the DODAG routing topology built through the RPL protocol.

DODAG (Destination Oriented Directed Acyclic Graph): A directed acyclic graph directed to a destination which is the root of the network (Sink).

It should be noted that throughout our simulation we will use the RSSI calculation formula given in Eq. (3) in order to ensure that the signal is sufficient to guarantee a reliable wireless connection. Said formula is tested outdoors giving acceptable results [14].

$$d = (-0.5894 * RSSI) - (5.6768) \quad (3)$$

3.1 Triple point method

This approach is based on a hybrid method, which combines the mean consensus algorithm and the triplet gossip algorithm in order to agree on the detected value (mean between pairs whose values are closer) to relay to the well (Figure 3).

At a time " t " of the clock, a cluster leader node N_i (CH), is randomly selected among all nodes (CHs). N_i selects two neighboring nodes N_j , N_k among all the nodes of its cluster respecting the condition: the elected nodes must have the closest values to the value of N_i respecting a proximity threshold quantified at the value S verifying for example the condition ($|VN_i - VN_j| < S$) between node N_i and node N_j .

An exception to this rule is made when one of the neighboring nodes is out of range, the triple gossip is then reduced to one chat per pair in this time slot.

As for the transmission cost, the notification from node N_i to nodes N_j and N_k is evaluated at one transmission, thanks to the wireless broadcast function.

Nodes N_j and N_k transmit their own local values $VN_j(t)$ and $VN_k(t)$ respectively to node N_i (no collision, node N_i can bring a consensus to its first notification). This step costs 2 transmissions.

Node N_i averages the values of the three locally selected nodes and updates its local estimate according to the Election1 algorithm:

Algorithm Election1

```
//A sensor is randomly activated N1 and chooses two closest
neighbors N2 and N3 among the others//
{v1, v2, v2 values detected by N1, N2 and N3 respectively}
min = abs(v1-v2)
If min > abs(v1-v3) then min = abs(v1-v3)
If min > abs(v2-v3) then min = abs(v2-v3)
If min = abs(v1-v2) then ElectedValue=(v1+v2)/2
If min = abs(v1-v3) then ElectedValue=(v1+v3)/2
```

```

If min = abs(v2-v3) then ElectedValue =(v2+v3)/2
Send (ElectedValue)
End

```

The triple point method summarizes the average value of three nodes per 3 radio transmissions in a gossip context.

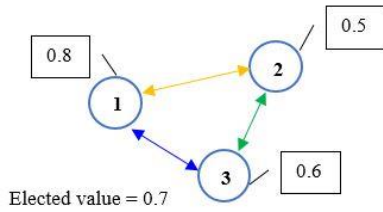


Figure 3. Example of the value selection to transmit

Note:

An execution of the algorithm consumes a maximum of: 6 subtraction operations and 6 comparison tests, 1 addition, 1 division, 2 assignments (benefit from negligible energy consumption).

3.2 Standard detection method

Standard transmission is based on the principle that the nodes can transmit their physical quantities captured to the neighbor node or to the sink node without going through a selection or gossip or consensus algorithm, but it uses the DODAG (Destination Oriented Directed Acyclic Graph) routing topology built by the RPL protocol and which is composed of a set of routes or paths connecting the sensor nodes of the network to the root sensor node (Sink).

The RPL (Routing Protocol for Low-power and Lossy Networks) is a proactive distance vector protocol: When a preferred parent has been selected, all traffic will be transferred via this preferred parent, as long as it is accessible, without any attempt at load balancing between the other available parent candidates. This benefit can drain the power of overloaded parents, prompting network failure and unreliability issues, as overloaded nodes are likely to die soon enough. Many articles explain how the RPL protocol works, for example the authors, Sobral et al. [15] which present in more detail the RPL protocol in Internet of Things applications.

4. SIMULATION AND RESULT ANALYSIS

The proposed methods were tested by a simulation on the CONTIKI / COOJA platform. For our simulation the mote selected is TmoteSky is a wireless sensor board from Moteiv [16]. This is an MSP430 based board with an 802.15.4 compatible CC2420 radio chip, 1megabyte external serial flash memory and two light sensors. TmoteSky port was integrated into the Contiki building system in March 2007. The functionality of the MSP430 F1611 module is detailed in the Texas Instruments MSP430x1xx Series User Guide available in Ref. [17].

We place the sink node (called mote1) at a fixed initial position. The three scenarios are tested respectively by applying the three methods by deploying several anchor, aggregator or simple nodes according to the chosen algorithm. Two parameters are recorded, the first one is the convergence time of the value captured at the sink and the second the

quantity of energy consumed.

The estimation of the energy consumption of the motes is based on the 'Energest' module of Contiki by evaluating certain states like CPU, LPM, Tx and Rx, in order to calculate the elapsed time at each sensor node in real time. This energy estimation 'Energest' module is called when the component is activated to produce a timestamp. The energy consumption of the nodes is calculated with the following equation [18]:

$$E_c = (E_v \times E_p \times V_t) / R_{t_second} \times R_t \quad (4)$$

E_c ≡ Energy consumption (milli-watt),

E_v ≡ Energy value,

E_p ≡ Electric power,

V_t ≡ Voltage,

R_t ≡ $R_{timer} = 10$ second, and

R_{t_second} ≡ Low frequency crystal frequency = 32768 Hz.

We underline that the value of $E_{nergest_Value}$ is the remainder of the difference between the number of ticks (ALL_CPU) at the current time 't2' and the previous time 't1'.

A single system tick is approximately 1 millisecond.

$$eff_c (\%) = \frac{\Delta T_x + \Delta R_x}{CPU_e + LPM_e} \quad (5)$$

eff_c ≡ Effective cycle (%),

ΔT_x ≡ Transmission energy,

ΔR_x ≡ Reception energy,

CPU_e ≡ CPU energy,

LPM_e ≡ Low Power Mode energy, and

LPM ≡ Low Power Mode, device inactive waiting for events.

4.1 Simulation with a standard detection method

The topology taken for this first method comprises 20 anchor mote and 01 Sink (mote1). These 20 motes can sense and transmit temperatures, humidity and light levels to the sink without any calculations and without requiring conditions.

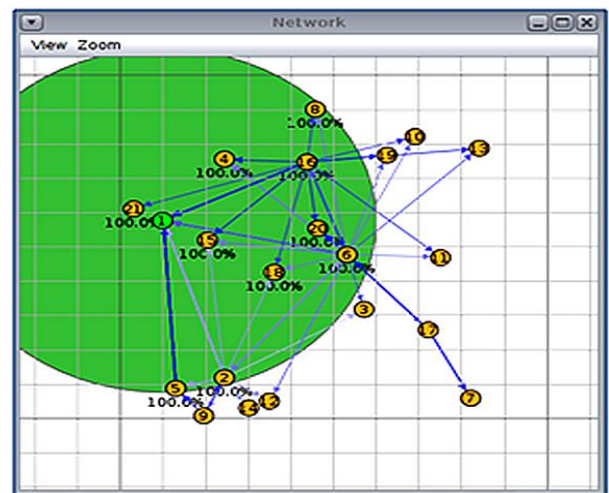


Figure 4. Scenario of a topology in standard detection

The effective cycle is the energy consumption expressed in milli-joule of a mote close to Sink (here mote 15) and another which is far from Sink but is in its coverage zone (mote 6) and a third mote is outside the Sink's coverage area (mote 7) after every 10 seconds and in a time interval between 0 and 200 seconds (Figure 4).

By setting the data display to 10 seconds and using the POWERTRACE tool from Contiki knowing that it is accurate to 94% compared to the real value.

1. Mote 15: The analysis made on the nearest mote gave us the following Table 2:

Table 2. Number of ticks for anchor mote 15

Time	ALL CPU	ALL LPM	ALL TX	ALL RX
0	14902	312578	3070	6686
10	34244	620853	6061	11922
20	52817	929943	6836	16040
30	64626	1245800	9056	18395
40	76011	1562087	9795	20365
50	94331	1871430	12145	24386
60	106827	2186599	14337	27172
70	118474	2502616	16545	29142
80	130156	2818601	18795	31168
90	148018	3128405	20045	33581
100	161477	3442793	22595	37137
110	173940	3757821	25047	39887
120	185565	4073865	27735	41857
130	197266	4389828	30051	43827
140	208931	4705830	32655	45797
150	220532	5021897	35140	47767
160	232200	5337896	37625	49737
170	244925	5652837	41045	51821
180	256569	5968862	43210	53791
190	268565	6284531	45140	56050
200	280938	6599823	47354	58723

By applying the Eq. (5), we obtain the following Table 3 of the radio effective cycle:

Table 3. Percent of effective cycle of node 15 radio

Time	Tx(%)	Rx(%)	Total(%)
10	0.91	0.02	0.93
20	0.24	0.01	0.25
30	0.68	0.01	0.68
40	0.23	0.01	0.23
50	0.72	0.01	0.73
60	0.67	0.01	0.68
70	0.67	0.01	0.68
80	0.69	0.01	0.69
90	0.38	0.01	0.39
100	0.78	0.01	0.79
110	0.75	0.01	0.76
120	0.82	0.01	0.83
130	0.71	0.01	0.71
140	0.79	0.01	0.80
150	0.76	0.01	0.76
160	0.76	0.01	0.76
170	1.04	0.01	1.05
180	0.66	0.01	0.67
190	0.59	0.01	0.60
200	0.68	0.01	0.68

i.e. $Tx- Eff_c = (6061-3070) / ((34244-14902) + (620853-312578)) = 0.009130 = 0.91\%$ (see Table 2 and Table 3).

After activating Power trace on Cooja and connecting its library and configuring its clock to 10 seconds in our source program, we obtain a set of values printed in the following table representing an estimate of the amount of energy consumed:

Table 4. Power consumption on TmoteSky 15

Time	CPU	LPM	Tx	Rx
10	0.088541	0.007338	0.533977	1.045027
20	0.085020	0.007357	0.138359	0.821891
30	0.054057	0.007519	0.396332	0.470023
40	0.052116	0.007529	0.131932	0.393182
50	0.083862	0.007364	0.419540	0.802531
60	0.057202	0.007502	0.391333	0.556044
70	0.053316	0.007522	0.394189	0.393182
80	0.053476	0.007522	0.401688	0.404359
90	0.081766	0.007374	0.223160	0.481599
100	0.061610	0.007484	0.455246	0.709724
110	0.057051	0.007499	0.437750	0.548859
120	0.053215	0.007523	0.479883	0.393182
130	0.053563	0.007521	0.413470	0.393182
140	0.053398	0.007522	0.464886	0.393182
150	0.053105	0.007524	0.443642	0.393182
160	0.053412	0.007522	0.443642	0.393182
170	0.058250	0.007497	0.610565	0.415935
180	0.053302	0.007523	0.386513	0.393182
190	0.054913	0.007514	0.344559	0.450862
200	0.056639	0.007505	0.395261	0.533491

Note: See datasheet for TmoteSky pp.9 [1] 1. Active courant at $V_{cc} = 500\mu A/s = 0.50mA/s$, Voltage = 3v and $1s = 32768$ clocks. 2. i.e. Energy consumed (CPU) = $(34244-14902) \times 0.50 \times 3 / (32768 \times 10) = 0,088541$ mw.

Table 4 displays a return of Tmote Sky values in four states in the form of a number of clock ticks.

- ALL_CPU: The total (high) CPU (CPU in active mode).
- ALL_LPM: The total number of ticks in an LPM state (low power mode).
- ALL_Tx: The total number of ticks in the Tx (Transmit) state.
- ALL_Rx: The number of ticks in the Rx (Receive) state.

We calculate the energy consumption using the formula of Eq. (5).

The standard model displays in Figure 5 the variation of the energy consumption in the four states mentioned previously. We can clearly see that the increase in energy concerns the transmission but above all the reception. Figure 6 confirms our assertion by summing the transmission and reception energies to represent the effective radio service cycle.

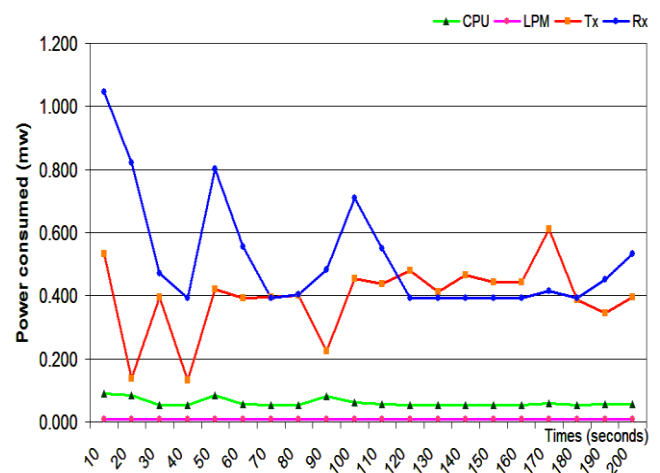


Figure 5. Skymote 15 power consumption in standard detection

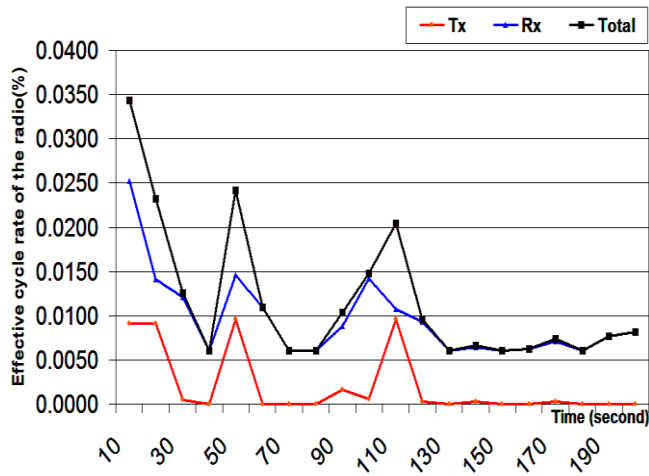


Figure 6. Effective cycle rate of the radio after every 10 seconds from mote 15 in standard detection

2. Mote 6: The data of Mote 6 in number of captured ticks are displayed in the following graph:

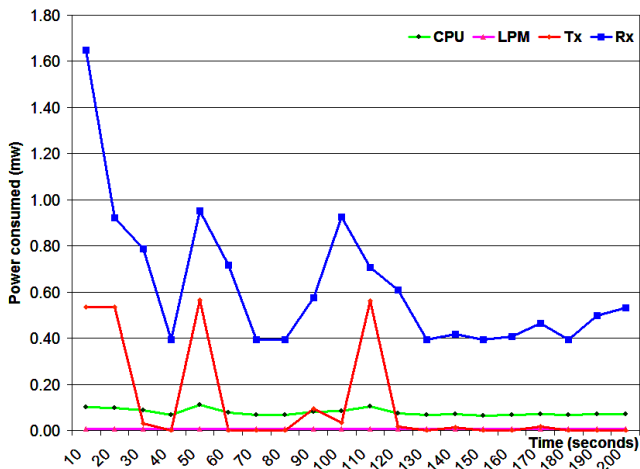


Figure 7. Skymote 6 power consumption in standard detection

As for Figure 5, Figure 7 represents the variation of the energy consumption in the four states (CPU, LPM, Tx, Rx) of the mote 6 in the standard model. Figure 8 shows the effective radio cycle in order to show the difference between emission and transmission.

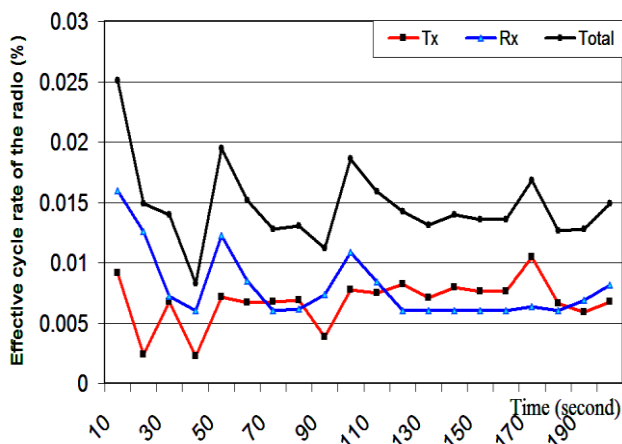


Figure 8. Effective cycle of the radio after every 10 seconds from mote 6 in standard detection

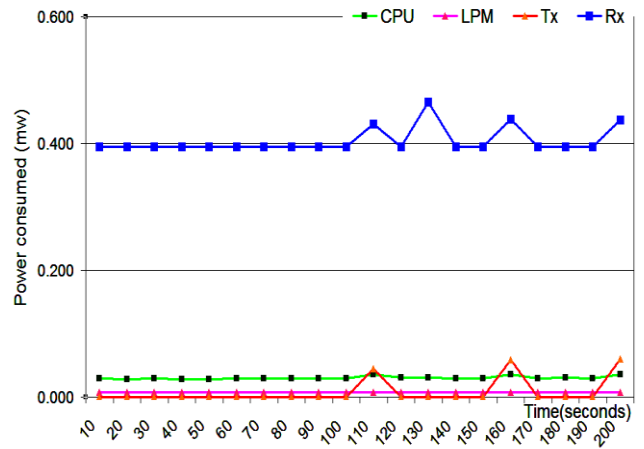


Figure 9. Skymote 7 power consumption in standard detection

3. Mote 7: Now we are going to study the energetic behavior of mote 7.

Including its effective radio cycle in order to understand the influence of the distance of a mote from the sink in order to previously predict an optimal unfolding with collaboration the nodes between them. The data of word 7 in number of captured ticks are displayed in Figure 9. Just as Figures 5 and 7, Figure 9 shows the variation of the energy consumption in the four states (CPU, LPM, Tx, Rx) of the mote 7 in the standard model.

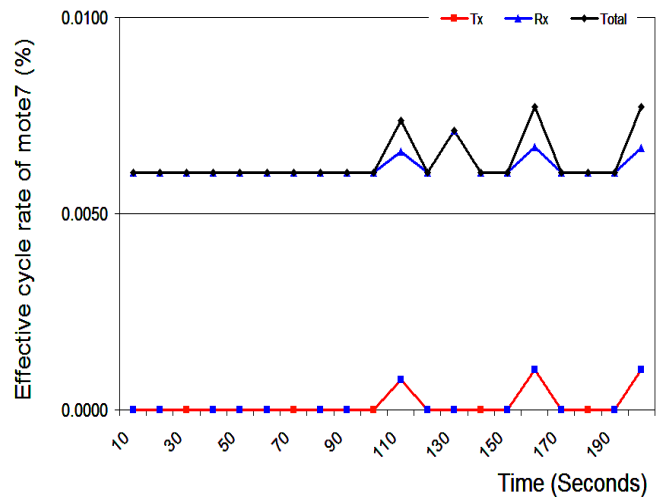


Figure 10. Effective cycle rate of the radio after every 10 seconds from mote7 in standard detection

By analyzing Figure 4 of the scenario of the topology of the standard detection model, we notice that the mote7 is outside the coverage of the sink on one side and that it has almost no neighbor. Figure 10 denotes a virtual absence of transmission but the mote6 is still listening.

4.2 Standard detection method discussion

Note that mote 15 and mote 6 are in the same coverage of the sink (mote1), on the other hand mote 7 is outside the coverage area of the sink.

It can be seen in Figure 11 that the energy consumption by the three motes 15, 6 and 7 is more marked on the motes in packet reception mode (Rx) (Figures 5, 7 and 9), this could be explained by the fact that the motes are always listening to the

transmission channel, and this is also proved by the results found by the effective cycle of each mote (high percentage of Rx compared to Tx).

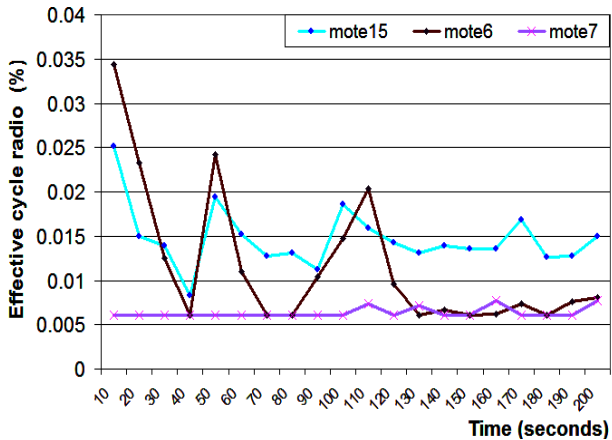


Figure 11. Effective cycle rate of the radio comparison for of the three motes, after every 10 seconds in standard detection

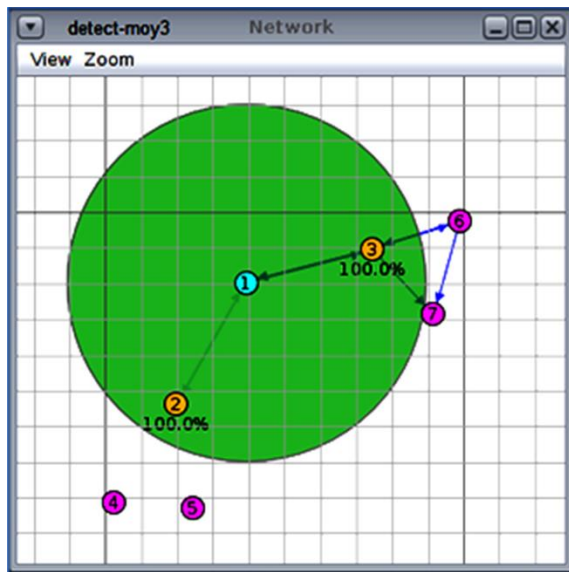


Figure 12. Scenario of a topology in the triple point method

The corresponding data on the eve of these three motes (mote 15, mote 6, mote 7) is represented by the parameter LPM which underlines an energy consumption. This means that the motes consume almost no power when they are in standby mode.

According to Figure 11, the use of the radio is more accentuated when the motes are on the cover of the sink, but it is much more when it approaches the latter. This could be explained by the fact that mote 15 is more likely to relay data to the sink than others. This deduction is also verified by analyzing the amount of energy consumed by CPU during processing. For example, mote 15 prints (Figure 5) an average usage between [0.3 mw - 0.5 mw].

An important remark is also verified from these results is the fact that when a mote (here mote6) has several neighbors, the energy consumed at reception increases considerably (Figure 6), this can be verified with mote 7 which has only one neighbor (mote17).

So, for an optimum deployment of the sensors, it is necessary to avoid the concentration of the nodes around a single mote because this could induce an excessive use of

energy and arrive at its loss and a re-structuring of the topology or to permanently lose the network of sensors wireless.

4.3 Simulation with the triple point method

The configuration chosen is given by the following Figure 12. The simulation of our method is based on 7 motes. One sink, 4 standard catch motes and two aggregator motes responsible for cooperation to send a single value, according to a consensus, to the Sink.

Standard motes 4, 5, 6, 7 detect and transmit the values of temperature, humidity and light in broadcast. Motes 4 and 5 aggregate their data to word 2 and words 6 and 7 to word 3. Motes 2 and 3 are the aggregators.

Our method is based on aggregators to optimize radio transmission and overcome the problems posed by the standard configuration.

Once the simulation is triggered, the POWERTRACE tool gives the following results in Table 5:

1. Mote 2: On this mote we note the number of ticks after every 10 seconds:

Table 5. Number of ticks for a mote 2

Time	ALL CPU	ALL LPM	ALL TX	ALL RX
0	15584	312155	5765	3422
10	31396	623959	9080	7586
20	44257	938711	12237	10370
30	52227	1258393	12317	13175
40	58490	1579789	12317	15155
50	72257	1893695	15468	18589
60	79606	2213977	15548	20708
70	85543	2535618	15548	22688
80	91483	2857264	15548	24668
90	97890	3178514	15548	26648
100	111115	3492951	18705	29548
110	118191	3813542	18705	32537
120	126278	4133033	18875	34795
130	132433	4454461	18875	36938
140	138398	4776082	18875	38918
150	144656	5097483	18875	40898
160	150987	5418815	18875	43092
170	157191	5740282	18875	45072
180	163593	6061546	18875	47290
190	169786	6383022	18875	49270
200	176559	6703892	18965	51391

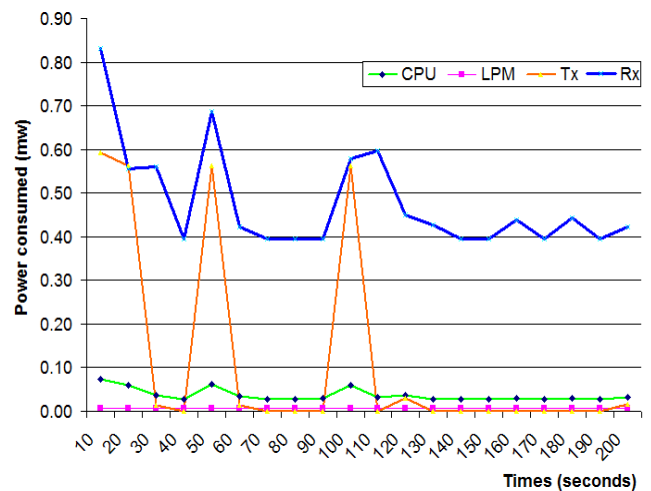


Figure 13. Skymote 2 power consumption in the triple point method

By respecting the same approach of the standard model but using our algorithm this time, we obtain Table 5 representing the number of ticks in the four states (CPU, LPM, Tx, Rx) for the mote 2. Next, we calculate the energy consumption using the formula of the Eq. (4).

Next, we calculate the energy consumption using the formula of the Eq. (5) and after a graphical representation we obtain Figure 13 which visualizes the energy consumption by our triple point method. This curve highlights an increase in energy consumption around the radio (Tx, Rx).

Likewise, Figure 14 underlines a use much of the transmission than that of reception.

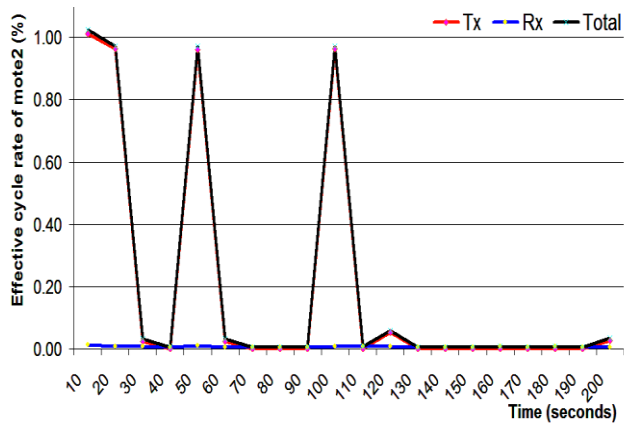


Figure 14. Percent of number of ticks for mote 2 in the triple point method

2. Mote 3: On this aggregator mote we note also the number of ticks after every 10 seconds:

Similarly to Mote 2 and by using the POWERTRACE tool, the data collected by the Mote 3 aggregator is shown in Figure 15 marking the energy consumed in transmission and reception as well as by the processor.

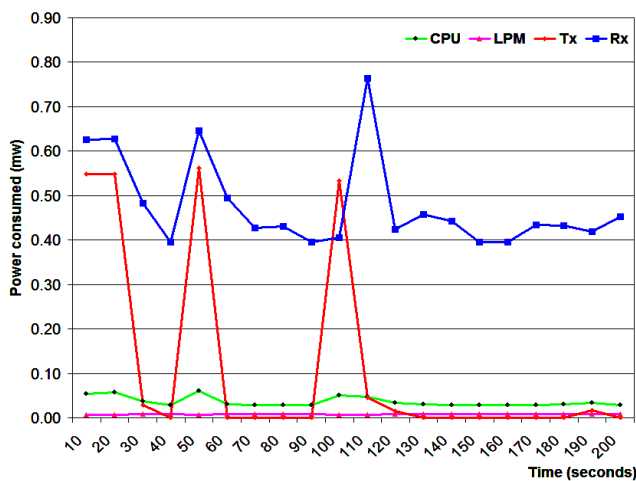


Figure 15. Skymote 3 power consumption in the triple point method

The energy consumption using the formula of the Eq. (5) and after a graphical representation we obtain Figure 16 which visualizes the energy consumption by our triple point method. This curve highlights an increase in energy consumption around the radio (Tx, Rx). Figure 16 displays the actual cycle of radio use and gives a graphical representation of high listening use versus transmission which appears occasional.

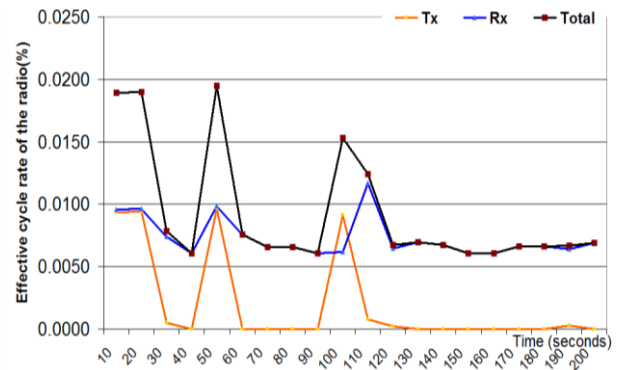


Figure 16. Percent of number of ticks for mote 3 in the triple point method

3. Mote 4: This is a simple sensor outside the sink coverage area and we want to follow its behavior and energy consumption throughout the process with the triple point method. Table 6 note also the number of ticks after every 10 seconds.

Figures 17 and 18 show the energy consumption of the main modules in Module 4 using the triple point method, respectively. The observation is highlighted on the receive (Rx) mode in front and the transmission thereafter.

Table 6. Number of ticks for a mote 4

Time	ALL CPU	ALL LPM	ALL TX	ALL RX
0	9403	318366	2992	3044
10	23207	632223	7426	6281
20	28738	954356	7426	8527
30	39955	1270802	10658	10943
40	45199	1593220	10658	12923
50	56917	1909170	13971	15644
60	62611	2231064	13971	17624
70	68188	2553069	13971	19604
80	73776	2875067	13971	21584
90	79342	3197086	13971	23564
100	91118	3512923	17060	26005
110	101343	3830365	20053	28421
120	107040	4152255	20053	30401
130	112700	4474244	20053	32605
140	118168	4796442	20053	34585
150	123620	5118657	20053	36565
160	133746	5436194	22820	38591
170	139037	5758573	22820	40571
180	149221	6076023	25590	42598
190	154507	6398399	25590	44578
200	159742	6720829	25590	46558

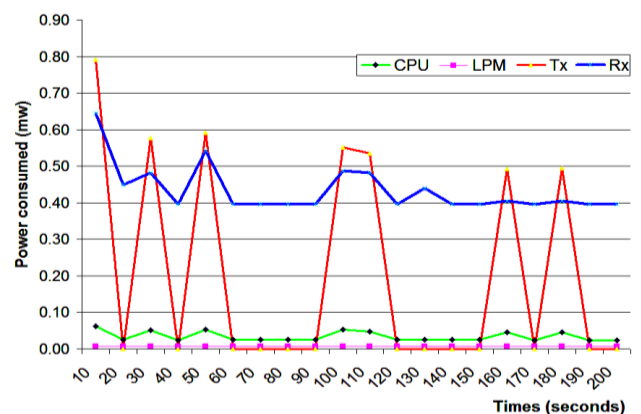


Figure 17. Skymote 4 power consumption in the triple point method

4.4 Triple point method discussion

According to the above results, we notice that the energy consumption by the three motes when these motes are in transmission (TX) or reception (RX) mode is greater than that if these same motes are in calculation (CPU) or standby (LPM). This means that most of the energy is often consumed in transmitting and receiving, and it is consumed more in receiving than in transmitting.

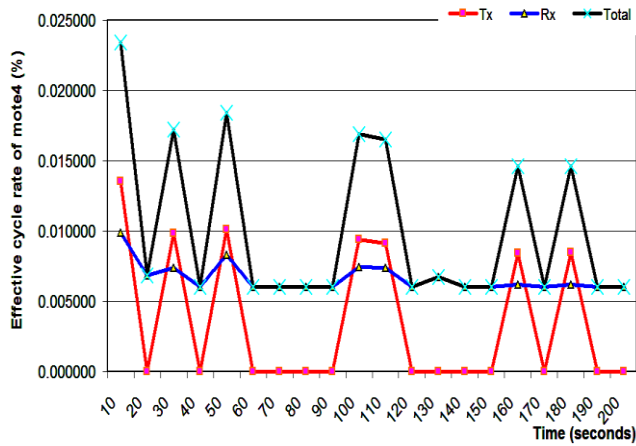


Figure 18. Percent of number of ticks for mote 4 in the triple point method

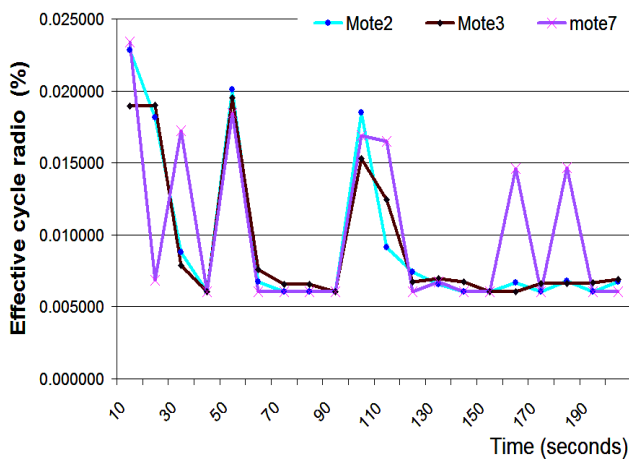


Figure 19. Effective cycle rate of the radio comparison for of the three motes, after every 10 seconds in in the triple point method

In Figure 19, we compare the two motes 2 and 3 which serve as aggregators for a mote 4, we note that the latter consumes a lot of energy than the aggregators and this is explained by the transmissions of temperatures, humidity rates and humidity rates broadcast light every 10 seconds.

4.5 Comparison of the two methods

The standard protocol of the RPL Cooja makes it possible to relay the value captured from one mote to another in a normal way. This method is simple and does not require any calculation. But the values captured are not always reliable and can cause a false alarm and a waste of time and resources for an audit. We have suggested a new method based on a reliability consensus governed by an aggregator node. It is based on a collector which calculates the average, at most, of

the first three values received, which then sends them to the Sink; this one we named it "the method of the triple points". This perspective ensures data reliability on one side and network perpetuity on the other side.

By taking the two plots of the effective radio cycles (Figures 11 and 19) side by side, we can draw the following conclusion:

(i) The maximum energy consumption by the standard method is 3.5% compared to our method which is 2.5%.

(ii) In our method the average consumption is around 1.1% compared to the standard method which exceeds 1.5% in energy.

(iii) It can also be noted that listening to the Rx reception channel is expensive in energy, but Tx transmission is more occasional.

(iv) In the standard method, mote 7 is not covered by the sink, so it is not overly stressed (no neighbor) if its reactivity will visibly increase the energy curve.

We notice that the standard RPL protocol gives very good results in response time but in power consumption it is by far the best. Our solution called 'triple points' is obviously the best solution in energy consumption and therefore the perpetuity of the network.

The standard method underlines a more interesting convergence time than our method that is surely due to our program for processing the average consensus but on the other hand it consumes more energy than ours.

5. CONCLUSIONS

The objective outlined in this article is the study of detection and cooperation for a good deployment of sensor networks. Detection can be summed up in two points: the first, detecting an event, but this operation is fortuitous because it is the pure function of the sensor. Agree on a consensus of the value to be transmitted in order to avoid false alarms on one side and to minimize network saturation by relaying the same values captured.

Our contribution was to improve the standard method of packet relay at the base station by other methods taking into account response time, data reliability and network lifetime. A method has been developed: Triple point method.

Our expectations thus verified, the simulations prove that the aggregation and the use of a cluster-head (aggregator) gives very good results in terms of network energy consumption and indeed a more interesting longevity. The trade-off is therefore emphasized: if the network relies on response time only then the standard method is the preferred one. But if we are talking about energy consumption and data reliability in order to avoid false alarms our triple point method is the best.

REFERENCES

- [1] Tmote, S. (2006). Ultra low power IEEE 802.15. 4 compliant wireless sensor module. <http://www.moteiv.com>
- [2] Al-Karaki, J.N., Kamal, A.E. (2004). Routing techniques in wireless sensor networks: A survey. *IEEE Wireless Communications*, 11(6): 6-28. <https://doi.org/10.1109/MWC.2004.1368893>
- [3] Wang, F., Hu, H. (2019). An energy-efficient unequal

- clustering routing algorithm for wireless sensor network. *Revue d'Intelligence Artificielle*, 33(3): 249-254. <https://doi.org/10.18280/ria.330311>
- [4] Yamparala, R., Perumal, B. (2019). Secure data transmission with effective routing method using group key management techniques-A survey. *Journal Européen des Systèmes Automatisés*, 52(3): 253-256. <https://doi.org/10.18280/jesa.520305>
- [5] Gorbachov, V., Sytnikov, D., Ryabov, O., Batiaa, A.K., Ponomarenko, O. (2020). Dimension reduction for network systems using structure model aggregation. *International Journal of Design & Nature and Ecodynamics*, 15(1): 13-23. <https://doi.org/10.18280/ij dne.150103>
- [6] Benaissa, B.E., Lahfa, F. (2019). Diffusion avisée après une collecte efficace de données par Intervalle de Confiance dans WSN basé sur l'IoT. *International Journal of Scientific Research & Engineering Technology (IJSET)*, 7(1): 6-15.
- [7] Periklis, N., Evangelia, P., Ozdemir, A. Barshan, B., Megalooikonomou, V. (2017). Investigation of sensor placement for accurate fall detection. In: Perego P., Andreoni G., Rizzo G. (eds) *Wireless Mobile Communication and Healthcare. MobiHealth 2016. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, vol 192. Springer, Cham. https://doi.org/10.1007/978-3-319-58877-3_30
- [8] Khan, S.M., Khan, W.M., Faraz, F.U., Khan, S.M. (2018). Incremental voting based spectrum sensing model for cognitive radio networks. *Review of Computer Engineering Studies*, 5(2): 27-33. <https://doi.org/10.18280/rces.050201>
- [9] Guyeux, C., Haddad, M., Hakem, M., Lagacherie, M. (2020). Efficient distributed average consensus in wireless sensor networks. *Computer Communications*, 150: 115-121. <https://doi.org/10.1016/j.comcom.2019.11.006>
- [10] Seam R.M., McCourt, M.J. (2020). Distributed estimation of an uncertain environment using belief consensus and measurement sharing. *IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp. 1362-1367. <https://doi.org/10.1109/SMC42975.2020.9283404>
- [11] Winter, T., Thubert, P., Brandt, A., Hui, J.W., Kelsey, R., Levis, P., Pister, K., Struik, R., Vasseur, J.P., Alexander, R. (2012). RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks. *rfc*, 6550: 1-157.
- [12] Ren, J., Huang, S., Song, W., Han, J. (2019). A novel indoor positioning algorithm for wireless sensor network based on received signal strength indicator filtering and improved Taylor series expansion. *Traitement du Signal*, 36(1): 103-108. <https://doi.org/10.18280/TS.360113>
- [13] Alrawashdeh, R. (2018). Path loss estimation for bone implantable applications. *Jordanian Journal of Computers and Information Technology (JJCIT)*, 4(2): 94-101. <https://doi.org/10.5455/jjcit.71-1515153943>
- [14] Javaid, R., Qureshi, R., Enam, R.N. (2015). RSSI based node localization using trilateration in wireless sensor network. *Bahria University Journal of Information & Communication Technologies (BUJICT)*, 8(2): 58-64.
- [15] Sobral, J.V.V., Rodrigues, J.J.P.C., Rabêlo, R.A.L., Al-Muhtadi, J., Korotaev, V. (2019). Routing protocols for low power and lossy networks in Internet of Things applications. *Sensors*, 19(9): 2144. <https://doi.org/10.3390/s19092144>
- [16] http://contiki.sourceforge.net/docs/2.6/a01784.html#_details, accessed on Jan. 04, 2021.
- [17] Instruments, T. (2010). Msp430f5438 datasheet. Reference SLAS655B.
- [18] Contiki OS: Using Powertrace to estimate power consumption. [Online]. Available: <http://thingschat.blogspot.mk/2015/04/contikios-using-powertrace-and.html>, accessed on Feb. 25, 2021.