



Mémoire de fin d'études

Pour l'obtention du diplôme

Master en Informatique

Spécialité : Systèmes d'Informations (SI)

Thème

Conception d'un Mécanisme de Localisation dans un réseau WSN (Basé sur l'amélioration de l'algorithme DV-Hop)

Présenté par:

- AMARA HAOUARI
- BOUZEROUATA MOHAMMED

Encadré par :

Mr. B. BENAÏSSA

Année universitaire 2020/2021

Remerciement

Avant tout nous remercions ALLAH ; le tout puissant, de nous avoir donné la force, la volonté et le courage afin d'accomplir ce modeste travail.

Nos reconnaissances et nos remerciements à notre encadreur, Mr B. BENAÏSSA pour son encadrement et sa disponibilité tout au long de ce projet.

Nos sincères remerciements à tous les professeurs, intervenants et toutes les personnes qui par leurs paroles, leurs écrits, leurs conseils et leurs critiques ont guidé nos réflexions et ont accepté de nous rencontrer et de répondre à nos questions tout au long de ce projet.

Nous nous permettons de remercier également tous ceux qui ont assistés et participé dans ce jury et être les témoins de ce modeste travail.

Enfin, nous voudrions exprimer nos reconnaissances envers nos proches et amis qui nous ont apporté leur soutien moral et de nous avoir encouragés durant cette période de travail.

Merci à tous et à toutes.

ملخص :

تعد شبكة المستشعرات اللاسلكية (WSN) مجال اهتمام مجتمع البحث. تتمثل الفائدة الرئيسية لهذا النوع من الشبكات في قدرتها على جمع البيانات الفيزيائية من بيئات معينة مثل درجة الحرارة والرطوبة والضوء وما إلى ذلك. لذلك، اكتسبت هذه الشبكات مكانتها في العديد من التطبيقات التي تتضمن عمليات المراقبة والبيوت الذكية والتنظيم الآلي للمدن، وهي مصممة خصيصًا للاتصالات منخفضة الطاقة و المتضمنة الأنظمة المحمولة. تتمتع هذه المستشعرات بقدرة معالجة للبيانات محدودة للغاية، ويتم تشغيلها بواسطة بطاريات صغيرة أو مكون لتجميع الطاقة، وتتواصل عبر الروابط اللاسلكية، و تتمثل إحدى سلبياتها الشائعة في أنها غير قادرة على التعرف على موقعها إلا إذا تم استعمال مستشعرات تستعمل نظام تحديد المواقع (GPS) القادرة على التعرف على موقعها من خلال الاتصال عبر الأقمار الصناعية وهي مكلفة وغير عملية.

في هذا السياق، يعد تحديد موقع المستشعرات في WSN موضوعًا بالغ الأهمية ومتطورًا باستمرار. في هذه المذكرة، قمنا أولاً بتطبيق خوارزمية حساب إحداثيات المواقع المعروفة باسم (DV-Hop) على منصة المحاكات المسماة بـ: "Contiki Cooja". بالإضافة إلى ذلك نقترح مساهمة جديدة حيث تقدر العقدة العادية (أي المستشعرات غير المعروفة الموقع) المسافة التي تفصلها عن جميع نقاط الارتساء (أي المستشعرات المعروفة المواضع - على سبيل المثال تلك المجهزة بنظام GPS) بالاعتماد فقط على المعلومات المتاحة محليًا.

Résumé

Le réseau de capteur sans fil (RCSF) est un domaine d'intérêt pour la communauté de recherche. L'utilité principale de ce type de réseaux est leur capacité à collecter des données physiques à partir d'environnements spécifiques tels que la température, l'humidité et la lumière, etc.

Par conséquent, les RCSF ont toujours gagné leur place dans la pratique dans de nombreuses applications impliquant des processus de surveillance, la domotique et l'automatisation des villes. Ils sont particulièrement conçus pour les communications embarquées de faible puissance. Ces capteurs ont une capacité de calcul très limitée, sont alimentés par de petites batteries ou un composant de récupération d'énergie, et communiquent par des liaisons sans fil. L'un des inconvénients populaires des RCSF est qu'ils ne sont pas capables de reconnaître leur propre position contrairement aux capteurs de positionnement global (GPS) qui sont capables de reconnaître leur position en communiquant par satellite.

Dans ce contexte, la localisation dans les RCSF est un sujet critique et en constante évolution. Dans ce mémoire, nous implémentons d'abord l'algorithme de localisation connu dans la littérature (DV-Hop) sur la plate-forme Contiki-Cooja. De plus, nous proposons une nouvelle contribution selon laquelle le noeud régulier (c.-à-d. noeud inconnu) estime sa distance à toutes les ancrs (c.-à-d. les positions connues - p. ex., équipées d'un GPS) du réseau en se fiant uniquement aux renseignements disponibles localement.

Abstract :

The Wireless Sensor Network (RCSF) is an area of interest to the research community. The main utility of this type of networks is their ability to collect physical data from specific environments such as temperature, humidity and light, etc.

Therefore, SCNs have always earned their place in practice in many applications involving surveillance processes, home automation and city automation. They are particularly designed for low power on-board communications. These sensors have very limited computing capacity, are powered by small batteries or an energy harvesting component, and communicate over wireless links. One of the popular drawbacks of RCSFs is that they are not able to recognize their own position unlike global positioning sensors (GPS) which are able to recognize their position by communicating via satellite.

In this context, localization in RCSF is a critical and constantly evolving subject. In this thesis, we first implement the localization algorithm known in the literature (DV-Hop) on the Contiki-Cooja platform. In addition, we propose a new contribution whereby the regular node (i.e. unknown node) estimates its distance to all anchors (i.e. known positions - e.g., equipped with 'a GPS) network by relying only on locally available information.

Table des matières

Liste des figures	iv
Liste des tableaux	v
 Introduction générale	 1
 Chapitre I : Généralités sur les réseaux capteurs	 3
1- Introduction	4
2- Architecture des nœuds capteurs	4
2-1- L'unité d'acquisition	5
2-2- L'unité de traitement des données	5
2-3- Unité de transmission	6
2-4- L'unité d'énergie	6
3- Les réseaux capteurs WSN	6
4- Domaines d'applications des réseaux de capteurs	7
4-1- Applications militaires	7
4-2- Applications à sécurité et surveillance	7
4-3- Applications environnementales	8
4-4- Applications médicales	8
4-5- La domotique (Home Automation)	8
4-6- Applications industrielles et commerciales	9
5- Conclusion	9
 Chapitre II : La localisation dans les réseaux WSN	 10
1- Introduction	11
2- Représentation du problème de localisation	11
3- Importance des systèmes de localisation	12
4- Contraintes pour un système de localisation	12
5- Composition d'un système de localisation	13
5-1- Technologies de mesure ou d'estimation de distance/angle	14
A- Indicateur de puissance du signal (RSSI)	14
B- Le temps d'arrivée (Time of Arrival)	16
C- Les mesures de différence de temps d'arrivée (TDOA)	17
D- L'angle d'arrivée (AOA)	18
5-2- Calcul de position	21
5-2-1- Techniques géométriques d'estimation de position	22
A- La Trilatération	22
B- La Triangulation	23
C- Distance MinMax	24
D- Centre de gravité (CENTROÏDE)	25
5-2-2- Tableau comparatif des méthodes géométriques	26
5-3- Algorithmes et Méthodes de localisation	27
5-3-1- Les méthodes RANGE-BASED	27
5-3-2- Les méthodes RANGE-FREE	28
A- DV HOP : (Distance Vector-Hop)	28

B- APIT : (Approximate Point In Triangulation)	28
6- Comparaisons et analyses	29
7- Conclusion	30
Chapiter III : Etat de l'art sur les améliorations de l'algorithme de localisation DV-HOP	31
1- Introduction	32
2- Fonctionnement de base de l'algorithme DV_HOP	32
2-1- Calcul le nombre de sauts minimum parmi les nœuds	32
2-2- Détermination de la taille moyenne des sauts et de la distance entre les nœuds	33
2-3- Calcul la position du nœud inconnu (U)	34
2-4- Analyse mathématique pour l'erreur avec DV-Hop	35
3- IDV-HOP basé sur la pondération du HOP-SIZE	37
3-1- La distance de saut moyenne pondérée	38
3-2- Correction de position	38
3-3- Résultat de la simulation	40
4- Algorithme de localisation amélioré RF-DVHop	40
4-1- RF-DVHop avec RSSI Reformé	40
4-2- RF-DVHop avec Mécanisme de rétroaction (Feedback Mechanism)	42
4-3- Résumer de RF-DVHop	43
4-4- Résultats des simulations	44
5- L'algorithme SD-DVHop : (Shortest Distances DV-Hop)	45
5-1- La distance du plus court chemin appliquée dans l'algorithme DV-Hop	46
5-2- Résultats de la simulation	47
6- Algorithme DEC-DVHop	48
6-1- Calcule du nombre de minimum de saut parmi les nœuds	48
6-2- Déterminer la moyenne de la taille du saut modifiée et la distance entre nœud	48
6-3- Estimation de la valeur de la position	50
6-4- Résultats et comparaisons	50
7- CHECKOUT_DV-Hop	50
8- Hybride DV-Hop Algorithme utilise RSSI	52
9- CWDV-Hop	53
10- NDLR_DV-Hop	55
11- RWDV-Hop (WEIGHTED DV-HOP ALGORITHM BASED ON RSSI)	58
11-1. Saut pondéré basé sur le RSSI	58
11-2. Amélioration du calcul de la distance moyenne des sauts	59
11-3. Optimisation sans contrainte	60
11-4. Les résultats de la simulation	62
12- SELECTIVE 3-ANCHOR DV-HOP	62
13- Analyse de l'état de l'art	65
14- Conclusion	68
Chapitre IV : Implémentation et Simulation.	69
1- Introduction	70
2- Objectifs de notre travail	70

3-Outils d'implémentation et de simulation	71
4- Le choix de Contiki	71
5- Le simulateur Cooja	72
6- Contraintes rencontrées dans cooja	74
7- Les principales étapes de notre Contribution	75
7-1. Organigramme de l'algorithme	77
7-2. L'algorithme DE_DV-Hop	78
8- Implémentation et Discussion	79
8-1. Méthode de calcul de précision:	79
8-2. Paramètres d'évaluation des deux algorithmes	79
8-3. Erreur de localisation en fonction du nombre total de nœuds.	79
8-4. Erreur de localisation en fonction du nombre d'ancres.	81
8-5. Détail des résultats	83
9- Evaluation énergétique de l'algorithme proposé	84
9-1. Paramètres de simulations	84
9-2. Topologie du réseau	84
9-3. Application POWERTRACE.	85
9-4. Evaluation énergétique de l'algorithme proposé	87
9-5. Calcul de la latence moyenne avec WIRESHARK	93
10- Conclusion	94
Conclusion Générale et perspectives	95
Références et Bibliographie	96

Liste des figures :

Figure	Titre	Page
1.1	Architecture interne d'un nœud capteur sans fil.	05
1.2	Architecture d'un réseau de capteur sans fil.	07
2.1	Composition d'un Système de localisation	13
2.2	L'indication de la force du signal reçu (RSSI)	14
2.3	Principe TOA	16
2.4	Schéma TOA Aller-Retour	16
2.5	Principe TDOA	18
2.6	Ensemble d'antenne de N éléments	19
2.7	LA TRILATERATION	22
2.8	La Triangulation	23
2.9	MinMax.	24
2.10	Distance MinMax	25
2.11	Méthode CENTROIDE	25
2.12	Schéma de localisation	27
3.1	Organigramme DV-HOP de base	33
3.2	Diagramme d'analyse de l'erreur pour l'algorithme DV hop	36
3.3	Graphe pondéré	45
3.4	Principe de Checkout DV-Hop	51
3.5	Schéma de la méthode hyperbola bidimensionnelle	54
3.6	Arbre Types des positions relatives	64
4.1	Interface graphique du simulateur Cooja	73
4.2	Organigramme de l'algorithme DE_DV-HOP	77
4.3	Erreur de localisation vs nombre total de nœuds	80
4.4	Erreur de localisation vs nombre des nœuds ancrés	82
4.5	Réseau de capteurs de l'évaluation énergétique	84
4.6	Echantillon sorties capteurs sur Cooja de l'algorithme proposé	87
4.7	Histogramme d'énergie consommée des ancrés.	88
4.8	histogramme d'énergie consommée du CPU	89
4.9	histogramme d'énergie consommée en mode sommeil	89
4.10	histogramme d'énergie consommée en TX	89
4.11	histogramme d'énergie consommée en RX	89
4.12	Connectivité réseau des ancrés N° 02 et N° 07	90
4.13	Connectivité réseau de l'ancre N° 01.	90
4.14	Histogramme d'énergie consommée des nœuds simples	92
4.15	Histogramme d'énergie consommée des nœuds simples détaillé	92
4.16	Des consoles WIRESHARK.	93

Liste des tableaux

Titre	Page
Tableau 1 : Avantages et inconvénients des techniques de mesure	20
Tableau 2 : Caractéristiques des différentes techniques de mesure	21
Tableau 3 : Récapitulatif des différentes techniques de combinaison de distance	26
Tableau 4 : Caractéristiques des différentes méthodes de localisation.	29
Tableau 5 : Distance et Nombre de sauts des ancrés	36
Tableau 6 : Tableau comparatif des différentes variantes DV HOP	66
Tableau 7 : Les erreurs de localisation vs nombre total de nœuds	80
Tableau 8 : Les erreurs de localisation vs nombre des nœuds ancrés	81
Tableau 9 : Liste de résultat de 20 nœuds capteur avec 5 ancrés	83
Tableau 10 : Caractéristiques du capteur SKYMOTE.	86
Tableau 11 : Echantillon des données sorties capteurs.	86
Tableau 12 : Total des TICK (s) CPU/RADIO des ancrés.	87
Tableau 13 : Consommation d'énergie des ancrés	88
Tableau 14 : Energie consommée des nœuds simples	91
Tableau 15 : Détail des énergies consommées par nœud.	91
Tableau 16 : Récapitulatif de la latence	92

Introduction Générale

Le réseau de capteurs sans fil (WSN) est un domaine d'intérêt pour la communauté des chercheurs. La principale utilité de ce type de réseaux est leur capacité à collecter des données physiques à partir d'environnements spécifiques tels que la température, l'humidité et la lumière, etc. Par conséquent, WSN a toujours gagné sa place dans la pratique dans de nombreuses applications impliquant des processus de surveillance, la domotique et l'automatisation de la ville. Ils sont particulièrement conçus pour les communications embarquées à faible consommation. Ces capteurs ont une capacité de calcul très limitée, sont alimentés par de petites batteries ou un composant de récupération d'énergie, et communiquent via des liaisons sans fil. L'un des inconvénients les plus courants des WSN est qu'ils ne sont pas capables de reconnaître leur propre position contrairement aux capteurs de positionnement global (GPS) qui sont capables de reconnaître leur position en communiquant via satellite. Dans ce contexte, la localisation dans WSN est un sujet critique et en constante évolution. Notre travail est d'implémenter dans un premier temps l'algorithme de localisation connu dans la littérature (DV-Hop) sur la plateforme Contiki Cooja. De plus, nous proposons une nouvelle contribution dans laquelle le nœud régulier (c'est-à-dire un nœud inconnu) estime sa distance à toutes les ancres (c'est-à-dire des positions connues - par exemple, équipées d'un «GPS») du réseau en s'appuyant uniquement sur les informations disponibles localement.

Aussi d'améliorer le DV HOP natif qui est basé uniquement sur le nombre de saut entre nœuds. La version standard de cet algorithme souffre d'importance déviations de position qui est une conséquence logique du cumul d'erreur propagée dans tous le réseau.

Notre objectif est de trouver une technique pour améliorer le DV HOP et réduire considérablement l'écart de distance entre la distance réelle et celle estimée.

L'idée de la contribution sera concrétisée sur l'émulateur Cooja en langage de programmation C, et se repose sur deux techniques :

- 1- Le calcul du taux d'erreur de la distance (distance Euclidienne/distance RSSI).
- 2- La conversion du type de nœud une fois localisé.

Le principal plan de ce mémoire s'étend sur une Introduction Générale et 04 Chapitres ainsi qu'une conclusion générale.

Le premier chapitre donne un petit aperçu sur les capteurs leurs compositions et leur champ d'utilisation dans le cadre des réseaux WSN.

Le deuxième chapitre expose le domaine de localisation des capteurs, son importance et ses contraintes ainsi que les différentes technologies mises en place pour l'estimation des distances entre capteurs.

Le troisième chapitre présente un état de l'art sur quelques contributions faites sur l'algorithme DV HOP natif pour l'améliorer.

Le quatrième chapitre comprend notre contribution et les simulations faites sur l'émulateur Cooja ainsi que les résultats analysés et commentés.

A la fin de ce mémoire, une conclusion générale exposant nos perspectives pour l'amélioration future de cet humble travail.

Chapitre I

Généralités sur les réseaux capteurs

1. Introduction

La miniaturisation des équipements électroniques embarqués ne cesse d'augmenter alors qu'ils sont destinés à accomplir des tâches de plus en plus complexes. En raison de leur petite taille et de la faible consommation d'énergie qui leur est imposée, les constituants des réseaux de capteurs ne peuvent pas rivaliser avec les ordinateurs, smart phones, tablettes ou autres terminaux de poche en termes de capacité de traitement de données, de stockage et de communication.

2. Architecture des nœuds capteurs

Les nœuds capteurs sont des dispositifs électroniques communicant entre eux par des liaisons sans fil, et dont les fonctions premières sont de collecter et de transmettre des données de manière totalement autonome en énergie, ces systèmes communicants constituent ainsi le réseau et sont appelés nœuds capteurs. Ils sont nécessairement équipés chacun d'une unité de captage afin d'observer l'environnement dans lequel ils sont placés. L'association de ces nœuds forme le réseau de capteurs sans fil ou WSN (Wireless Sensor Network). Les nœuds capteurs peuvent être fixes ou mobiles. Leur position dans le réseau n'est pas obligatoirement prédéterminée car ils peuvent être dispersés aléatoirement dans une zone géographique (avec un algorithme et un protocole d'auto-organisation) ou bien leurs positions peuvent être spécifiées et organisées dans une zone ciblée (topologie préconfigurée).

Ils sont composés de quatre composants de base comme représentée dans la figure 1.1 : une unité d'acquisition, une unité de traitement, une unité transmission et une source d'énergie.

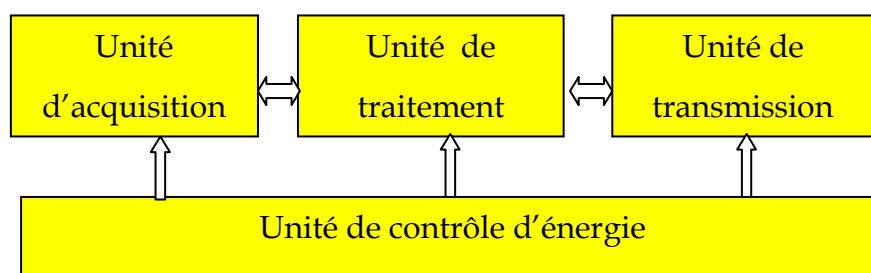


Figure 1.1 : Architecture interne d'un nœud capteur sans fil.

2.1. L'unité d'acquisition

Le rôle de l'unité d'acquisition est d'obtenir des mesures numériques des paramètres environnementaux. Elle se compose de deux sous unités, unité de captage et un convertisseur analogique « CAN ». Le capteur permet de mesurer des informations environnementales : Température, humidité, pression etc. Puis produit des signaux analogiques qui sont convertis par le convertisseur analogique-numérique pour pouvoir être traitées par l'unité de traitement.

2.2. L'unité de traitement des données

Cette unité comprend un processeur [2,3] avec une petite mémoire RAM de stockage et traitement des données et une ROM pour les programmes et une mémoire flash. Elle est gérée par un système d'exploitation spécialement conçu pour les capteurs (TinyOS par exemple). Cette unité coordonne le fonctionnement de l'ensemble des nœuds du réseau pour collaborer ensemble pour alléger la tâche du nœud puits (ou sink - Point de collecte de données).

Le rôle de l'unité de routage [2,3] est d'acquérir les informations en provenance de l'unité D'acquisition et de les envoyer à l'unité de radiocommunication afin que l'unité de contrôle Puisse les réceptionnées correctement par l'intermédiaire d'autres systèmes.

2.3. Unité de transmission

Le rôle de cette unité est d'émettre ou recevoir des données [2,3], appelée aussi unité de communication. La quantité d'énergie de la transmission est proportionnelle à la distance.

2.4. L'unité d'énergie

Le réseau WSN est constitué de capteurs alimentés par des batteries ou des piles à faible puissance, difficile à les faire remplacer quand leur autonomie atteignent zéro car généralement ils sont déployés dans des endroits inaccessibles voire même hostiles.

La raison pour laquelle, l'optimisation de consommation d'énergie est l'objectif de plusieurs études tenues sur les capteurs pour étendre la durée de vie des réseaux WSN.

3. Les réseaux capteurs WSN

Les réseaux de capteurs sans fil sont considérés comme un type spécial de réseaux ad-hoc. Les nœuds de ce type de réseaux consiste en un grand nombre de capteurs capables de récolter et de transmettre des données environnementales d'une manière autonome [4]. La position de ces nœuds n'est pas obligatoirement prédéterminée, ils peuvent être dispersés aléatoirement à travers une zone géographique, appelée champ de captage, qui définit le terrain d'intérêt pour le phénomène capté. Les données captées sont acheminées grâce à un routage multi-saut à un nœud considéré comme un "point de collecte", appelé nœud puits (ou sink) **Figure 1.2**. Ce dernier peut être connecté à l'utilisateur du réseau via Internet ou un satellite. Ainsi, l'utilisateur peut adresser des requêtes aux autres nœuds du réseau, précisant le type de données requises et récolter les données environnementales captées par le biais du nœud puits.

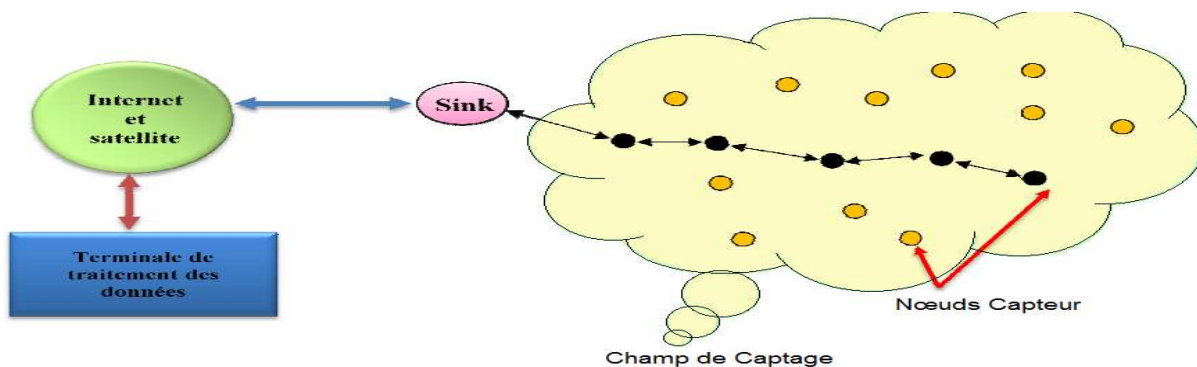


Figure 1.2. Architecture d'un réseau de capteur sans fil

4. Domaines d'applications des réseaux de capteurs

Les réseaux capteurs ont envahis plusieurs domaines d'applications[4], parmi ces domaines où ces réseaux se révèlent très utiles et peuvent offrir de meilleures contributions, on peut noter le domaine militaire, domaine sécuritaire, application médical, l'environnemental, et la domotique.

4.1. Applications militaires

La première utilisation des capteurs dans les applications militaires s'était en 1980 où il a été conçu un système de surveillance des mouvements des troupes ennemis. Ce système était connu sous le nom DISTRIBUTED SENSOR NETWORK

4.2. Applications de sécurité et surveillance

L'utilisation des réseaux capteurs dans le domaine de sécurité permet d'optimiser les dépenses réservées aux sécurités.

Parmi ces systèmes, on peut citer à titre d'exemple les détecteurs de mouvement qui constituent un système d'alarme contre toute intrusion.

D'autres systèmes de surveillance sont installés au niveau des grandes structures telles que les ponts ou les bâtiments dans le but de détecter les

altérations ou fissures d'une structure lors d'un sinistre ou vieillissement de la structure.

4.3. Applications environnementales

Les réseaux capteurs ont beaucoup aidés à l'apparition de nouvelles applications ayant relation avec l'environnement, par exemple les Thermo-capteurs qui sont largement utilisés dans la détection d'incendie dans les forêts pour faciliter l'intervention et éviter tout désastre.

D'autres types de capteurs sont conçu pour capter des évènements chimiques, par exemple contrôler le degré de pollution d'une zone ou de contrôler le degré pureté des liquides (eau).

Dans le domaine de l'industrie, les capteurs sont très utilisés que ce soit dans la chaine de production ou dans le contrôle de sécurité.

4.4. Applications médicale

Dans le domaine médical, les capteurs ne manquent pas d'être utilisés dans la surveillance des organes vitaux des patients ayant des pathologies, par exemple ceux des Cancéreux ou de contrôler le taux de saturation d'oxygène des gents atteint par le COVID-19.

4.5. La domotique (Home Automation)

Les capteurs et les Micro-capteurs ont fait apparaitre un nouveau domaine connu sous le nom IOT (Internet of Thing). L'utilisateur pourrait désormais contrôler à distance ces équipements ménagers (réfrigérateur, Climatiseur,...).

Aussi, les futures maisons vont profiter le maximum de cette opportunité qu'offre les Micro-capteurs en les rendre plus intelligents (SMART-HOME), en d'autre terme toutes les opérations domestiques seront automatisées.

4.6. Applications industrielles et commerciales

Des entreprises manufacturières, via des réseaux de capteurs pourraient suivre le procédé de production à partir des matières premières jusqu'au produit final livré. Grâce aux réseaux de capteurs sur la chaîne de fabrication, aussi les entreprises pourraient offrir une meilleure qualité de service tout en réduisant leurs coûts. Les produits en fin de vie pourraient être mieux démontés et recyclés ou réutilisés si les micro-capteurs en garantissent le bon état. Dans les chambres froides, en intégrant plusieurs micro-capteurs le système de refroidissement pourrait être déclenchée seulement c'est nécessaire.

5. Conclusion

Dans ce chapitre nous avons donné un aperçu sur l'architecture des capteurs, les réseaux WSN et le domaine d'application des réseaux capteurs ainsi que leurs spécificités et les concepts nécessaires à la compréhension de ce type de réseau. Dans le prochain chapitre nous allons exposer le principe de localisation, ses composantes, ses contraintes et ses différents mécanismes et algorithmes.

Chapitre II

La localisation dans les réseaux WSN

1. Introduction

Différents champs d'application (militaires, environnementales, domestique, médicale ...) Utilisant les capteurs sans fils et s'appuient sur le déploiement aléatoire des capteurs qui nécessite alors une localisation précise de leurs positions dans un système de coordonnées fixes.

Les problématiques de la localisation dans les WSN qu'on peut citer : le routage, la qualité de service, la sécurité et la mobilité.

Différentes méthodes et techniques sont utilisées dans la localisation des capteurs, celles basés sur la télémétrie, connues sous le nom de « **RANGE-BASED** », c'est-à-dire basée sur la mesure de la distance, à titre d'exemple le RSSI, AOA, TOA...

D'autres techniques exploitent des informations de connectivités comme moyen d'aide pour les nœuds inconnus pour estimer leurs distances dans une topologie à distribution aléatoire, ces méthodes sont connues sous le nom « **RANGE-FREE** » (Libre plage), comme exemple le DV-HOP et APIT.

Des techniques géométriques qui complètent les deux précédentes méthodes RANGE-FREE et RANGE-BASED, comme la Multilatéraion, la Trilatération et la Triangulation dans le but d'apporter plus de précision à la position du nœud, connues sous le nom de « **COMBINAISON DE DISTANCE** », elles sont largement utilisées dans le système universel de calcul de position GPS.

2. Représentation du problème de localisation

Soit N un nombre de capteurs déployés dans un WSN, chacun ayant un emplacement S , avec $S = \{S_1, S_2, S_3, \dots, S_n\}$ soient S_{xi} , S_{yi} , S_{zi} , les coordonnées x , y , z de chaque nœud.

Le problème est de déterminer l'emplacement des nœuds dans ce réseau, sachant que certains nœuds connaissent leur propre position, appelés couramment 'Ancre' ou 'nœud balise'.

Le problème peut être formulé mathématiquement comme suit :

Etant données un réseau multi-saut représenté par un graphe $G=(V,E)$ et un ensemble de nœuds dont leurs positions $\{X_b, Y_b\}$ (coordonnées cartésienne) ; pour tous les \underline{b} appartiennent à \underline{B} , l'objectif est de trouver $\{X_u, Y_u\}$ pour tous les nœuds inconnus \underline{u} appartient \underline{U} .

3. Importance des systèmes de localisation

L'importance d'un système de localisation, c'est qu'il nous permette de savoir l'endroit où les mesures ont été prises par le système de captage d'évènements [5].

Plusieurs domaines d'application reposent sur les systèmes de localisation, parmi ces domaines :

- Suivi de la faune.
- La domotique.
- Le contrôle industriel.
- La surveillance permanente de la qualité de l'eau potable.

Par ailleurs, la disponibilité d'information de localisation permettrait à l'avenir l'apparition d'autres domaines d'application.

4. Contraintes pour un système de localisation

La spécification de la norme WSN impose des contraintes [5] à ce type de réseau. Compte tenu de leur limitation en ressources matérielles telle que la puissance du processeur, la capacité de la mémoire et la faible autonomie d'énergie, parmi ces contraintes :

Auto-organisation : implique que le système possède une certaine structure et des fonctionnalités associées au réseaux WSN. Ce qui signifie que les nœuds du réseau sont arrangés selon une logique définie et interagissent entre eux de façon identique et totalement distribuée pour un but commun.

Evolutivité (extensibilité, échelle) : les algorithmes de localisation sont sensibles à la densité du réseau WSN, si le réseau est moins dense alors très souvent les résultats sont imprécis.

Chapitre II : La localisation dans les réseaux WSN

Robustesse : L'algorithme de localisation doit être tolérant aux pannes de communication ainsi qu'aux imprécisions de distances et des positions conséquences des faits suivants :

Les liens de communications entre deux nœuds sont asymétriques.

La propagation des ondes radio diffèrent selon le milieu (herbe, sable ou sur l'asphalte phénomène réflexion et absorption).

Les obstacles et les irrégularités des terrains peuvent faussés les mesures et par conséquent une mauvaise localisation.

Topologie non convexe les coordonnées des nœuds qui se trouvent dans les extrémités sont souvent moins précises.

Efficacité dans l'utilisation des ressources du WSN bien que le système de localisation surveille et contrôle une zone d'intérêt et établit une cartographie complète du réseau, néanmoins son but principal est l'optimisation de l'utilisation des ressources. Les concepteurs des algorithmes de localisation doivent tenir en compte les coûts en termes d'énergie et rationaliser la consommation en matière de ressource.

5. Composition d'un système de localisation

Un système de localisation [5] est composé de trois (03) sous-systèmes distincts (voir figure 2.1.) :

- Technologies de mesure ou d'estimation de distance/angle
- Calcul de la position
- Algorithme de localisation

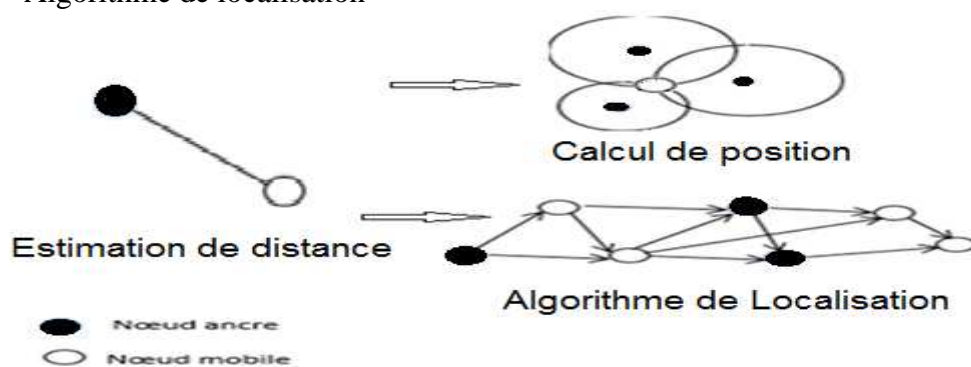


Figure 2.1. : Composition d'un Système de localisation

5.1. Technologies de mesure ou d'estimation de distance/angle

Permet à un capteur de mesurer la distance qui le sépare d'un capteur voisin ou de mesurer l'angle qu'il forme avec celui-ci. Cette mesure est très importante pour le calcul de la position.

A- Indicateur de puissance du signal (RSSI) : (Received Signal Strength Indicator)

RSSI : Est une mesure de niveau de puissance du signal en réception d'un signal radio provenant d'une antenne émetteur. Le RSSI permet de fournir une indication sur l'intensité du signal reçu selon une échelle de puissance de référence (figure 2.2).

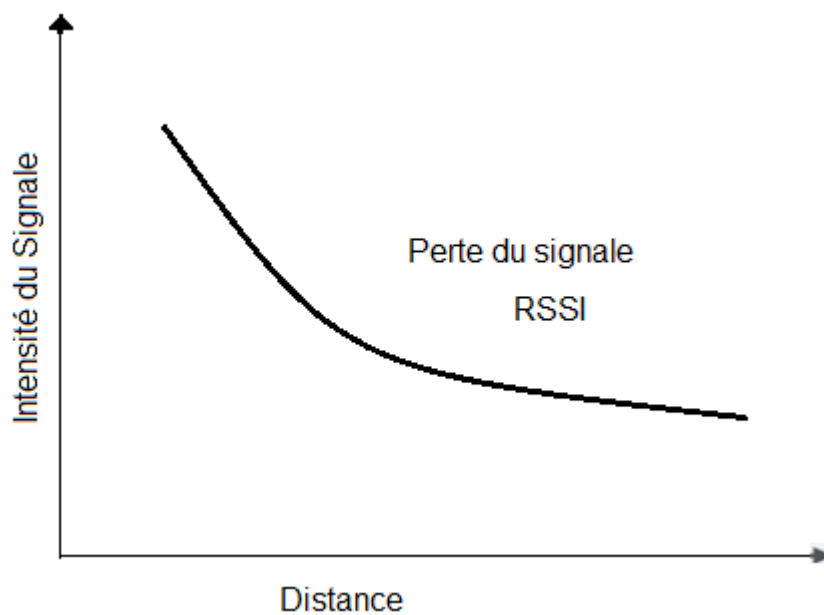


Figure 2.2.: L'indication de la force du signal reçu (RSSI).

La technique de la localisation RSSI [6] utilise la puissance de signal reçu comme moyen pour estimer la distance qui sépare un émetteur à un récepteur.

Il est à noter que la plupart des nœuds capteurs ont cette capacité de mesure qui est le RSSI.

L'intensité de signal reçu est une fonction monotone décroissante de leur distance.

Chapitre II : La localisation dans les réseaux WSN

Cette relation est formulée selon le modèle logarithmique suivant :

$$p_{\mu(d)}[dBm] = P_{0(d_0)}[dBm] - 10 n_p \log_{10}\left(\frac{d}{d_0}\right) + X_{\alpha} \quad (1)$$

Cas pratique :

$$p_r = p_{(d_0)} - 20 \log_{10}\left(\frac{d}{d_0}\right) \quad (2)$$

Où

d : est la distance entre les stations de base.

$P_{(d_0)}$: la puissance du signal reçu à distance d_0 de l'antenne mobile.

Avec $P_0 (d_0)[dBm]$ est une puissance de référence en Milliwatts dB à une distance de référence d_0 de l'émetteur, n_p est l'exposant de perte de trajet qui mesure la vitesse à laquelle l'intensité du signal reçu diminue avec la distance, X_{α} est une variable aléatoire de moyenne de distribution GAUSSIENE nulle avec l'écart type σ , et il représente l'effet aléatoire causé par l'ombrage. n_p et σ sont dépendant de l'environnement. L'exposant de perte de trajet n_p est généralement supposé être une constante.

❖ Avantages RSSI :

Peu coûteuse (aucun périphérique supplémentaire).

Mécanisme de calcul de position simple.

Disponibilité des modèles mathématiques d'atténuation.

❖ Inconvénients RSSI:

La perte de signal une fois les nœuds s'éloignent.

Sensibilité au bruit.

Difficultés de propagation dans des milieux INDOOR (obstacles).

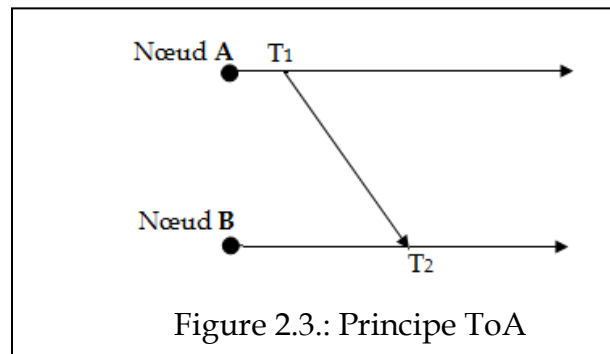
Mauvaise Performance dans un canal ayant un profil de propagation par trajet multiples sévère.

B- Le temps d'arrivé (Time of Arrival) :

Cette technique calcul la distance qui sépare l'émetteur au récepteur en se reposant sur le temps de parcours et la vitesse du signal radio transmis [5] (figure 2.3). La distance est calculée selon la formule suivante !

$$\text{Dist}_{AB} = (T_2 - T_1) * V$$

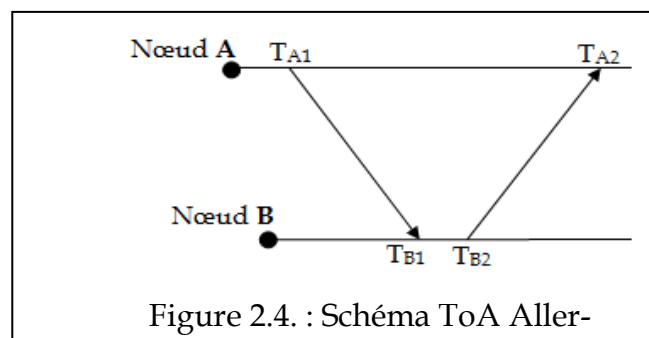
Pour une bonne précision de calcul de distance entre émetteur et récepteur la technique exige la synchronisation entre les horloges c'est-à-dire horloges à haute résolution du temps.



Le système de localisation universel GPS utilise cette technique. Néanmoins, lorsque les horloges de l'émetteur/récepteur ne sont pas synchrones alors l'envoi d'un message *aller-retour* est indispensable.

L'émetteur **A** envoie un signal au récepteur **B** au temps local T_{A1} , le signal arrive au capteur **B** au temps local de **B** T_{B1} .

Après certain délai, le capteur **B** envoie à son tour un signal au capteur **A** au temps local T_{B2} avec le temps de traitement du signal $T_{B2}-T_{B1}$. Le signal arrive au capteur A au temps locale de **A** T_{A2} . Ensuite le capteur **A** peut



calculer le temps de l'aller-retour à l'aide $(T_{A2}-T_{A1})-(T_{B2}-T_{B1})$. la distance est calculer par la formule suivante :

$$\text{DIST}_{AB} = \frac{(T_{A2}-T_{A1})-(T_{B2}-T_{B1})}{2} * V.$$

Chapitre II : La localisation dans les réseaux WSN

En plus de l'erreur de synchronisation tel que la dérive de l'horloge, la précision des deux mesures de temps de propagation à sens unique et aller-retour est affectée par le bruit, la bande passante du signal et les chemins multiples ce qui a donnée naissance à d'autre type de signaux connait sous l'acronyme UWB (Ultra Large Bandwidth).

L'UWB utilisé pour les mesures de temps de propagation précises.

❖ Avantages ToA :

Précision plus élevé au milieu confiné.

Technique de positionnement simple.

❖ Inconvénients ToA :

Nécessite une synchronisation parfaite entre émetteur et récepteur.

Coût très élevé.

Nécessite d'une résolution temporelle élevée au récepteur.

Nécessite d'avoir un trajet direct.

Sensibilité aux bruits.

C- **Les mesures de différence de temps d'arrivée (TDOA) :**

Cette méthode a deux variantes :

La premier variante est la différence de temps [5] pris par un signal envoyé par un nœud pour arriver au niveau de trois nœuds ou plus. Cette variante est connue dans les réseaux cellulaires comme les récepteurs de GPS pour calculer leur position sur terre elle nécessite une synchronisation parfaite des stations de base.

La deuxième variante se base sur le calcul de la différence des temps d'arrivés de deux signaux qui ont des vitesses de propagation différentes (figure 2.5), elle requiert des équipements supplémentaires au niveau de chaque capteur. Généralement, le premier dispositif utilise les ondes radio ($v=3 \times 10^8$ m/s) et le deuxième utilise des ondes sonores ($v=340$ m/s) ou Ultrason.

Chapitre II : La localisation dans les réseaux WSN

Le premier nœud émet les deux signaux simultanément ou en différés en un intervalle de temps T_{wait} . Le récepteur calcule la distance qui le sépare à l'émetteur avec la formule suivante :

$$D_{i,j}=(V_1-V_2)*(T_4-T_2-T_{wait})$$

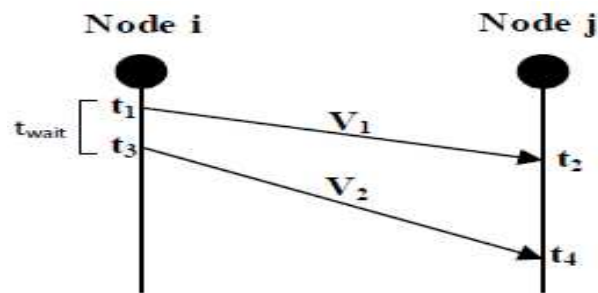


Figure 2.5.: Principe TDOA.

❖ Avantages :

Technique de positionnement simple.

Bonne précision en milieu confiné.

Pas besoin de synchronisation entre le nœud mobile et les stations de base.

❖ Inconvénients :

Moins de performance (le cas de trajet multiples, nécessite un trajet direct).

Une synchronisation d'horloge entre les paires de station de base.

Une résolution temporelle élevée (de l'ordre de la nanoseconde) est nécessaire au récepteur.

D- L'angle d'arrivée (AOA) :

Le AOA est utilisé dans le domaine de la navigation aérienne [7] ainsi dans tous les systèmes RADAR et les systèmes de téléphone mobile GSM, cette technique permet de mesurer l'angle formée entre deux capteurs par rapport à une référence donnée.

La figure 2.6 illustre un ensemble d'éléments d'antennes séparé par une distance déterminée (d).

Chapitre II : La localisation dans les réseaux WSN

La distance qui sépare un émetteur de la k ème antenne est approximativement obtenue par la formule ci-après :

$$R_k \approx R_0 - kd \cos \theta$$

Avec R_0 la distance entre l'émetteur et la 0ème antenne, θ est la direction de l'émetteur vu de l'ensemble d'antennes.

Donc l'angle d'arrivée de l'émetteur [7] à partir des mesures des différences de phase peut être dérivé à partir de $2\pi\left(\frac{d \cos \theta}{\lambda}\right)$ avec λ est la longueur d'onde du signal de l'émetteur.

Cette technique nécessite que le chemin soit en ligne directe de l'antenne (Line Direct Of Sight), elle est affectée par la présence de forte interférence et les trajets multiples des signaux, ce qui la rend moins précise.

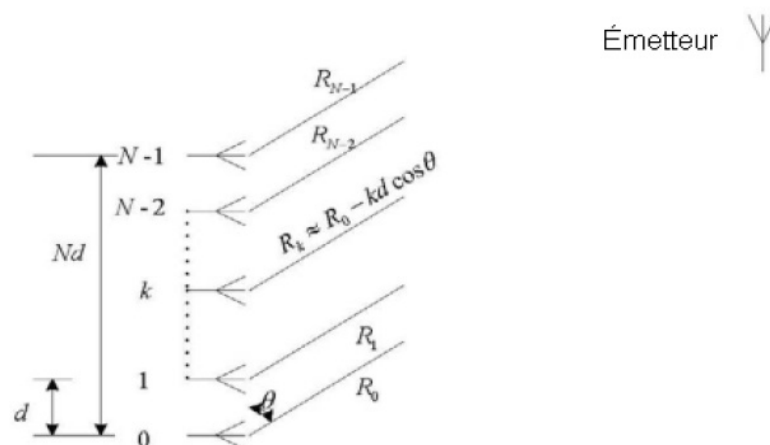


Figure 2.6.: Ensemble d'antenne de N éléments.

L'emplacement du nœud non-localisé est estimé en utilisant l'angle de deux signaux d'ancrage. Ceux-ci sont les angles auxquels les signaux d'ancrage sont reçus par le nœud non localisé. Les nœuds non-localisés utilisent la triangulation pour estimer leurs emplacements.

Chapitre II : La localisation dans les réseaux WSN

Ci-après un tableau récapitulant les avantages et les inconvénients des ces techniques de mesure.

Tableau 1 : Avantages et inconvénients des techniques de mesure.

Techniques de radio-localisation	Avantages	Inconvénients
Technique basée sur la puissance du signal (RSSI)	<ul style="list-style-type: none"> ○ Peu coûteuse (aucun périphérique supplémentaire) ○ Algorithme de positionnement simple. ○ Disponibilité des modèles mathématiques d'atténuation. 	<ul style="list-style-type: none"> ○ La perte de signal une fois les nœuds s'éloignent. ○ Sensibilité au bruit. ○ Difficultés de propagation dans des milieux INDOOR (obstacles). ○ Performance mauvaise dans un canal ayant un profil de propagation par trajet multiples sévère. ○ Faible précision
Techniques basée sur l'angle d'arrivé (AOA)	<ul style="list-style-type: none"> ○ Optimisation dans les stations de base ○ Algorithme de positionnement simple 	<ul style="list-style-type: none"> ○ Nécessite un trajet direct (chemin en ligne directe LOS) ○ Coût d'implémentation très élevé ○ Faible précision ○ Mauvaise performance dans le cas de trajet multiples
Technique basé sur le temps d'arrivé des signaux (TOA)	<ul style="list-style-type: none"> ○ Précision plus élevé en milieu confiné ○ Algorithme de positionnement simple 	<ul style="list-style-type: none"> ○ Nécessite une synchronisation parfaite entre émetteur et récepteur. ○ Coût très élevé. ○ Nécessite d'une résolution temporelle élevée au récepteur. ○ Nécessite d'avoir un trajet direct. ○ Sensibilité aux bruits.
Technique basé sur la différence du temps d'arrivé (TDOA)	<ul style="list-style-type: none"> ○ Technique de positionnement simple. ○ Bonne précision en milieu confiné. ○ Pas besoin de synchronisation entre le nœud mobile et les stations de base. ○ Moins de performance (le cas de trajet multiples). 	<ul style="list-style-type: none"> ○ Nécessite un trajet direct. ○ Une synchronisation d'horloge entre les paires de station de base. ○ Une résolution temporelle élevée est nécessaire au récepteur.

Chapitre II : La localisation dans les réseaux WSN

Ci-dessous un tableau récapitulant les caractéristiques de ces techniques.

Tableau 2 : Caractéristiques des différentes techniques de mesure

Méthode	Précision	Distance maximale	Matériel supplémentaire	Défis
RSSI	2-4 m	La porté	Aucun	Variation du puissance + interférence
TOA	2-3 cm	La porté	Aucun	Synchronisation des nœuds
TDOA	2-3 cm	Quelques mètres (2-10 m)	Capteur Ultrason	La distance maximale
AOA	Quelques degrés (5°)	La porté	Un ensemble de récepteurs	Travailler avec des nœuds de petite taille

5.2. Calcul de position :

La position d'un objet est son emplacement dans un repère de référence.

Tout système de localisation est caractérisé par les propriétés suivantes :

Une technique d'estimation de position - Multilatération, Trilatéraion ou triangulation et un repère permettant d'avoir les emplacements cohérents.

Une précision de position : Plus la précision est élevée plus la position est un point et si précision est moins importante la position devient une surface ouverte.

Une architecture particulière : Un milieu confiné (INDOOR) n'a pas les mêmes contraintes que celles de OUTDOOR.

L'estimation d'emplacement d'un point inconnu sur une surface donnée nécessite obligatoirement des points de références appelée couramment Ancre ou Balise (Beacon en anglais).

5.2.1. Techniques géométriques d'estimation de position :

Considérons un nœud A , pour estimer sa position au moins trois (03) points de référence, que nous appelons : N_1, N_2, N_3 sont nécessaires. Quelques méthodes géométriques sont utilisées pour estimer les positions, à savoir:

A- La Trilatération :

Soit M un nœud qu'on veut calculer sa position, soient N_1, N_2, N_3 trois nœuds Ancres. L'exécution de la trilatération (l'intersection des trois cercles) (voir figure 2.7) a produit le triplet (d_1, d_2, d_3)

Les équations (1.1) et (1.2) (1.3) décrivent la relation entre N_1, N_2, N_3 et A

$$(x - x_1)^2 + (y - y_1)^2 = (d_1)^2 \quad (1.1)$$

$$(x - x_2)^2 + (y - y_2)^2 = (d_2)^2 \quad (1.2)$$

$$(x - x_3)^2 + (y - y_3)^2 = (d_3)^2 \quad (1.3)$$

Les équations (1.1) et (1.2) décrivent la relation entre N_1, N_2 et A .

La position souhaitée est donc à l'intersection des cercles C_1 (N_1, d_1) et C_2 (N_2, d_2). Dans le cas général, C_1 et C_2 se rencontrent en deux points distincts M et M' . En utilisant les données de l'ancre N_3 par l'équations (1.3), l'un de ces deux points M et M' peut être retenu comme position du nœud A .

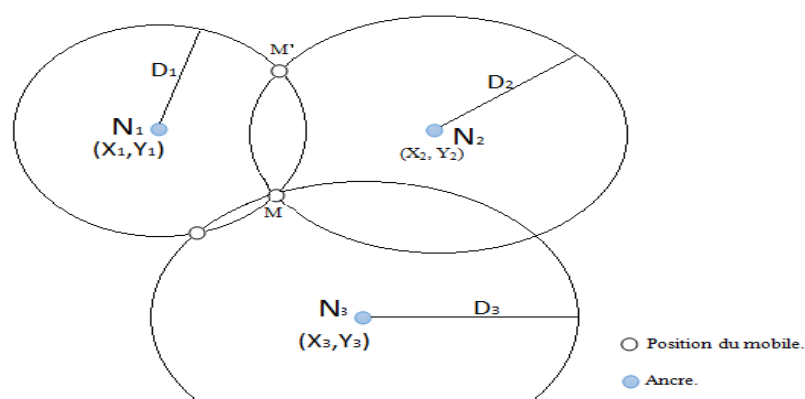


Figure 2.7.: La Trilatération.

B- La Triangulation :

M estime sa position en fonction des positions des nœuds d'ancrage avec le besoin de connaître l'angle d'arrivée du signal issu des nœuds de référence (Fig. 2.8). on utilisant les techniques basé sur l'angle d'arriver (AOA : Angle of Arrival).

Pour simplifier cette méthode de calcul par l'angle d'arrivée du signal, nous considérons deux ancres B1 (X1, Y1) et B2 (X2, Y2) à porter du nœud M (X,Y), chacun dispose des moyens physiques de mesurer l'angle auquel le signal est reçu d'après M. Supposons que α et β soient ces deux angles.

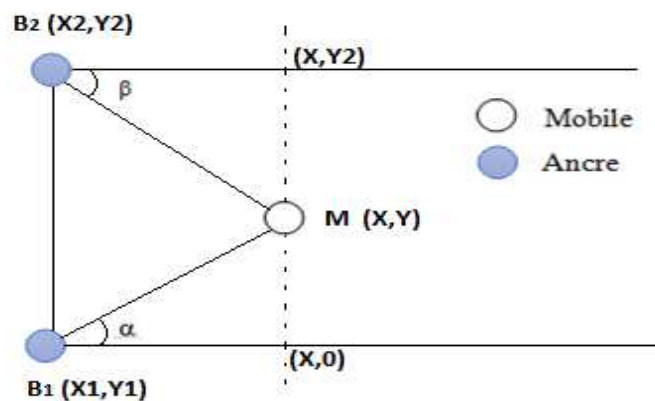


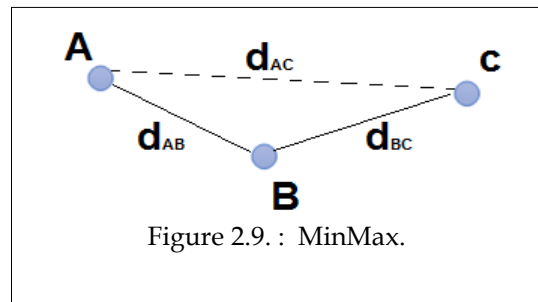
Figure 2.8. : La Triangulation

Le point M se trouve à l'intersection des droites passant par les couples (B1, M) et (B2, M) (Fig. 2.8.) En définissant un nouveau système de coordonnées d'origine B1 et où (B1, B2) ont fusionné à l'aide l'axe des y, on obtient La position du nœud M en calculant les formules suivantes :

$$\begin{cases} \tan \alpha = \frac{y}{x} \\ \tan \beta = \frac{y_2 - y}{x} \end{cases} \Rightarrow \begin{cases} x = \frac{y_2}{\tan \alpha + \tan \beta} \\ y = \frac{y_2 \tan \alpha}{\tan \alpha + \tan \beta} \end{cases} \quad \text{Donc } \left\{ M \left(\frac{y_2}{\tan \alpha + \tan \beta}, \frac{y_2 \tan \alpha}{\tan \alpha + \tan \beta} \right) \right.$$

C- Distance MinMax :

La méthode MinMax, consiste à ajouter les distances mesurées entre chaque paire de capteurs voisins séparant l'ancre et le capteur qui cherche à estimer sa position (figure 2.9).



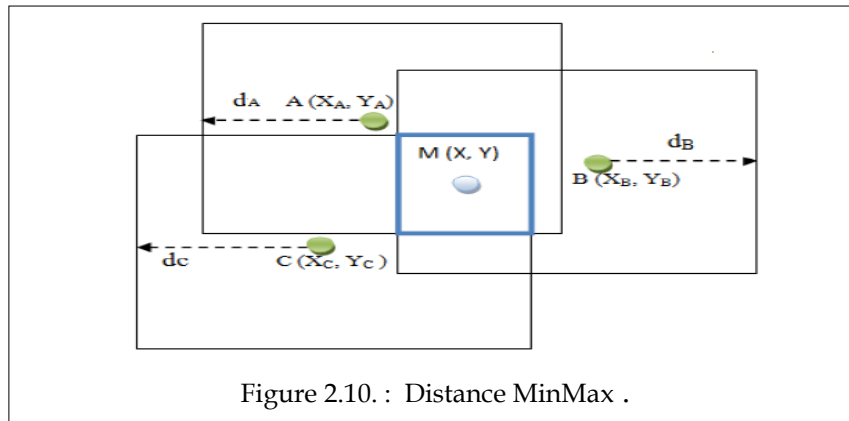
Lorsqu'une ancre envoie sa position, chaque capteur voisin mesure la distance avec l'émetteur, l'enregistre, ajoute cette distance à celle cumulée dans le message et fait suivre la position de l'ancre avec la distance mise à jour. Le même processus est répété pour tous les capteurs.

Au final, chaque capteur obtient la position de chacun des ancres et estime une distance qui le sépare de celle-ci.

La figure 2.10. illustre le fonctionnement de SumDist, la distance estimée entre les noeuds A et C est égale à $d_{AB} + d_{BC}$, sachant que de par l'inégalité triangulaire $d_{AC} \leq d_{AB} + d_{BC}$.

Après cette phase d'estimation des distances avec les ancres, les capteurs calculent leurs positions estimées en utilisant la méthode MinMax.

Le principe de cette méthode est de déterminer, pour chaque capteur, un « cadre » le contenant dont le centre de gravité correspond à sa position estimée. Dans la figure 2.10; le capteur M associe un " cadre " à chacune des ancres A, B, C. Chaque "cadre " est centré en la position de l'ancre (X_i, Y_i) et dépend de la distance estimée (d_i) avec le capteur M.



Par exemple, le " cadre " centré en A est construit comme suit :

$(X_A - d_A, Y_A - d_A)$ et $(X_A + d_A, Y_A + d_A)$, l'intersection de ces « cadres » forme un nouveau cadre définie par :

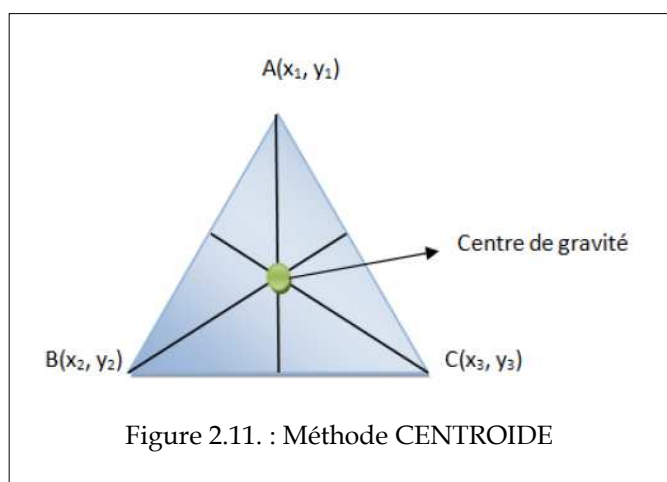
$(\max(X_i - d_i), \max(Y_i - d_i))$ et $(\min(X_i + d_i), \min(Y_i - d_i))$ avec $i \in \{A, B, C\}$

Le capteur M estime sa position comme étant le centre gravité de ce cadre, elle est calculée comme suit :

$$(X, Y) = \left(\frac{\max(X_i - d_i) + \min(X_i + d_i)}{2}, \frac{\max(Y_i - d_i) + \min(Y_i + d_i)}{2} \right)$$

D- Centre de gravité (CENTROÏDE) :

CENTROÏDE est un point où tous les trois médianes du triangle se coupent. Le centre de gravité de triangle peut être obtenu en trouvant la moyenne de la coordonnée « X » et la moyenne de la coordonnée « Y » de tous les sommets du triangle.



Chapitre II : La localisation dans les réseaux WSN

Formule CENTROÏDE : Dans la figure 2.11, les trois sommets du triangle sont : $A(X_1, Y_1)$, $B(X_2, Y_2)$ et $C(X_3, Y_3)$. Le CENTROÏDE de points, A, B et C est :

$$(x, y) = \left(\frac{(X_1 + X_2 + X_3)}{3}, \frac{(Y_1 + Y_2 + Y_3)}{3} \right)$$

Ci-dessous quelques propriétés de la méthode CENTROÏDE :

- CENTROÏDE est un terme qui décrit le centre d'un objet, appelé aussi centre de gravité, le géo-centre ou le barycentre.
- Le centre de gravité est toujours à l'intérieur du triangle.

5.2.2. Tableau comparatif des méthodes géométriques :

Tableau 3 : Récapitulatif des différentes techniques de combinaison de *distance*.

Méthode	Nombre de références	Distance	Angle	Défis
Trilatération	3	Oui	Non	Sensible aux imprécisions des distances.
Triangulation	3	Non	Oui	Besoin de matériel supplémentaire.
MinMax	$n \geq 2$	Oui	Non	Erreur de la position finale.
Centroïde	1	Non	Non	Erreur de la position finale.

5.3. Algorithmes et Méthodes de localisation:

Il existe plusieurs classification des méthodes de localisation, on cite parmi eux Algorithmes sans plage et basés sur la plage (Range-Free et Range-based) [5]. (voir figure 2.12)

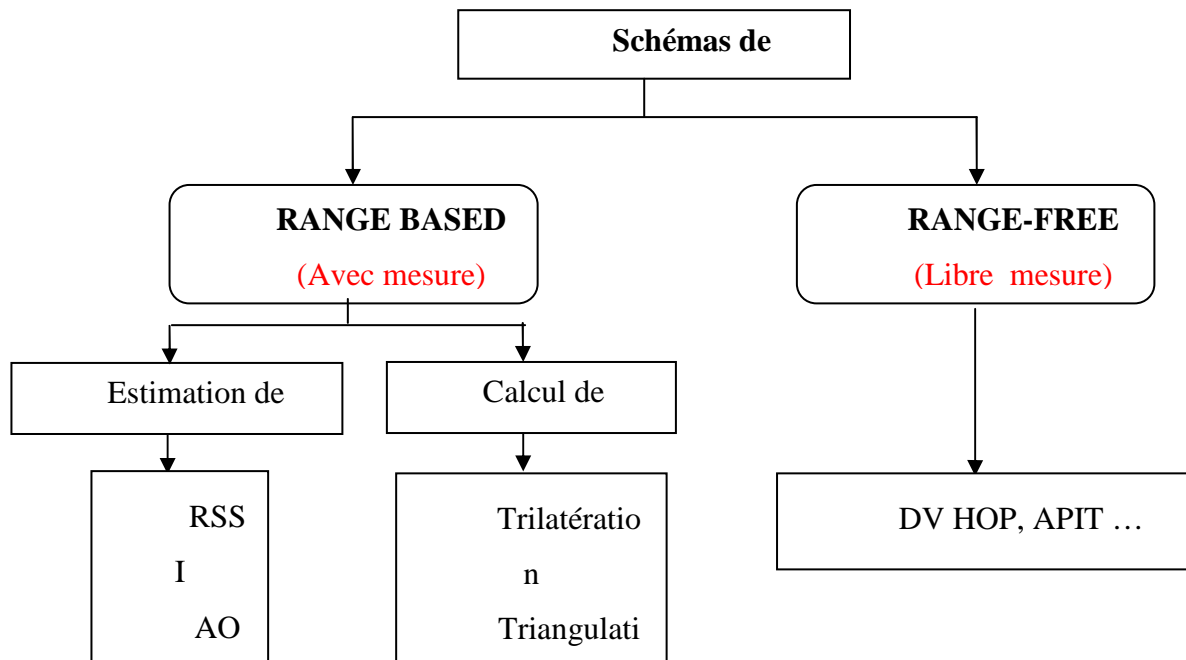


Figure 2.12 : Schéma de localisation.

5.3.1. Les méthodes RANGE-BASED :

Exigent de mesurer la distance entre les noeuds à localiser et les noeuds ancrés et utilisent différentes techniques de mesure de distance (RSSI, AOA etc.) pour estimer la position d'un noeud.

Ces techniques sont très précises pour la localisation, mais nécessitent du matériel supplémentaire pour TDoA et AoA. Module acoustique ou ultra-son pour TDoA et matrices radio ou microphones pour AoA. Le matériel AoA est plus cher que TDoA car en AoA chaque noeud a un haut-parleur et plusieurs microphones.

5.3.2. Les méthodes RANGE-FREE :

Algorithme Range-Free : les algorithmes de cette classe n'exigent pas la mesure de la distance ou de l'angle entre les nœuds, mais estiment la distance entre deux nœuds suivant l'information de connectivité ou celle de l'énergie consommée lors de la transmission d'un signal.

Ces algorithmes incluent les techniques de comptage de voisinage et de saut(Hop), et sont des techniques rentables car ils n'ont pas besoin de matériel supplémentaire, les résultats ne sont pas très précis. Parmi ces méthodes on peut citer : APIT et DV-HOP...

A- DV-HOP : (Distance Vector-Hop)

Estimation entre nœuds en utilisant le nombre de sauts [5,8,9] au moins trois (03) nœuds d'ancrage diffusent sur le réseau des coordonnées avec le nombre de saut, cette information se diffuse de voisin au voisin à travers le réseau, une fois reçu par le nœud voisin le nombre de saut est incrémenté à un.

De cette façon le nœud non-localisé peut trouver le nombre de saut qui le sépare du nœud ancre.

Ensuite tous les nœuds d'ancrage calculent le chemin le plus court à partir d'autre nœuds

Les nœuds non-localisés eux aussi calculent le chemin le plus court à partir de tous les nœuds ancrés. La formule de distance moyenne de saut est calculée comme suit : *Distance entre deux nœuds/nombre de saut.*

Cet algorithme objet de notre étude sera détaillé dans le chapitre III.

B- APIT : (Approximate Point In Triangulation)

Dans le schéma APIT [10] les nœuds d'ancre obtiennent les informations de localisation à partir du GPS ou bien des émetteurs. Le nœud non-localisé obtient les informations de localisation à partir des triangles qui se chevauchent. La zone est divisée entre les triangles chevauchés. On distingue quatre étapes :

Chapitre II : La localisation dans les réseaux WSN

Les nœuds non-localisés maintiennent la table après réception des messages balises des nœuds ancrés. La table contient l'information ID de l'ancre, l'emplacement et la puissance du signal.

Les nœuds non-localisés sélectionnent trois (03) nœuds ancre parmi ceux de la zone puis vérifient si forment-elles un triangle, ce type de test est appelé PIT (point de triangulation).

Ce test se poursuit jusqu'à avoir une précision sur le nœud estimé (non-localisé) par combinaison de n'importe quel trois nœuds d'ancre.

A la fin, le centre de gravité (COG) est calculé, c'est l'intersection de tous les triangles où le nœud non-localisé est placé pour trouver sa position estimée.

6. Comparaisons et analyses :

- Le système basé sur GPS et TOA sont plus coûteux à celle basé sur DV hop ou RSSI.
- Le mécanisme de localisation équipé d'un système GPS est très précis.
- Les mécanismes de localisation basés sur le système GPS sont moins en efficacité énergétique (moins économes). Cependant les mécanismes basés sur le RSSI sont très éco-énergétique (plus économe).

Tableau 4 : Caractéristiques des différentes méthodes de localisation.

Technique	coût	Précision	Efficacité énergétique	Taille du matériel
GPS	Elevé	Haute	Moins	Grande
GPS - Free	Bas	Moyenne	Moyenne	Petite
A Base centralisée	/	Haute	Moins	/
A Base Décentralisé	/	Basse	Haute	/
RSSI	Bas	Moyenne	Haute	Petite
TOA (onde ultrason)	Elevé	Moyenne	Moins	Grande
TDOA	Bas	Haute	Haute	Moins complexe peut être grande
AOA	Elevé	Basse	Moyenne	Grande
DV hop	Bas	Moyenne	Haute	Petite
APIT	Moyen	Moyenne	Haute	Moyenne

7. Conclusion :

Dans ce chapitre, nous avons présenté la problématique de localisation dans les WSN ainsi que les différentes méthodes (avec mesure, libre mesure et les méthodes de combinaison de distance) utilisées pour l'estimation de l'emplacement. Ces méthodes dépendront des exigences de l'application spécifique sur la précision de localisation, le coût et la complexité des algorithmes...

Il est évident que l'utilisation de telle ou telle méthode dépend de :

- *l'environnement du réseau* : obstacle, degré, d'humidité, vitesse du vent...etc. A titre d'exemple l'utilisation des technologies AOA ou RSSI sont fortement déconseillé dans les milieux confinés INDOOR.

- *le coût d'implémentation*, le cas du AOA.

Dans le prochain chapitre nous allons se focalisé sur un algorithme de la famille RANGE-FREE qui s'appelle le DV- HOP et qui vas nous donner une perspective intéressante pour localiser un nœud inconnu sans estimation directe de la distance.

Chapitre III

Etat de l'art sur les
améliorations de l'algorithme
de localisation DV-HOP

1. Introduction

DV-Hop est parmi ces algorithmes de localisation de la classe Range-Free, il est très populaire en raison de sa simplicité, de sa rentabilité et de sa robustesse. Il utilise l'estimation de la distance de saut, l'une des méthodes les plus largement utilisées pour la localisation. Toutefois, cette méthode engendre des erreurs de localisation dues d'une part au calcul du nombre de sauts et d'autre part à la taille moyenne des sauts de chaque ancre. Pour surmonter ses insuffisances et améliorer ainsi sa précision de la localisation, beaucoup de modifications et améliorations ont été faites sur le DV_HOP de base, Ci-dessous quelques approches et corrections pour rendre cet algorithme plus efficace.

- *IDV HOP basé sur la pondération du HOP SIZE.*
- *RFDV HOP (RSSI FEEDBACK DV HOP).*
- *SDDV-HOP (Shortest Distances DV-Hop).*
- *DEC-DV HOP amélioré basé sur la correction d'erreur de distance.*
- ...

2. Fonctionnement de base de l'algorithme DV_HOP

Cet algorithme [13,14] à base de saut fonctionne en différentes phases. La première phase calcule le nombre minimum de sauts entre tous les nœuds de capteur. Après cela, dans la deuxième phase, la taille moyenne de sauts est déterminée et la distance entre les nœuds est calculée. Enfin, la troisième phase calcule la position du nœud inconnu en utilisant la méthode de la multilatération[12].

2-1. Calcul le nombre de sauts minimum parmi les nœuds

Pendant cette phase, chaque ancre commence à inonder le paquet contenant les informations $\{(X_i, Y_i), h_i, id\}$, où *id* est l'identificateur de l'ancre, (X_i, Y_i) sont les coordonnées de ce $i^{\text{ème}}$ ancre et h_i est une variable qui contient le nombre de sauts qui est initialement mis à zéro. Chaque nœud inconnu préserve une table contenant les détails de $\{(X_i, Y_i), h_i, id\}$ reçus sous forme de paquets de chaque nœud d'ancrage avec le nombre minimum de sauts.

Maintenant, lorsque un nœud reçoit le paquet, il compare le nombre de sauts enregistré avec le nombre de sauts du paquet reçu pour la $i_{\text{ème}}$ ancre, si le nombre de sauts est inférieur au nombre de sauts du paquet reçu, alors le nœud rejette le paquet. Sinon, il mettra à jour sa table de saut en ajoutant un saut dans le nombre de sauts (HOP COUNT) du paquet puis il retransmettra ce paquet mise à jour.

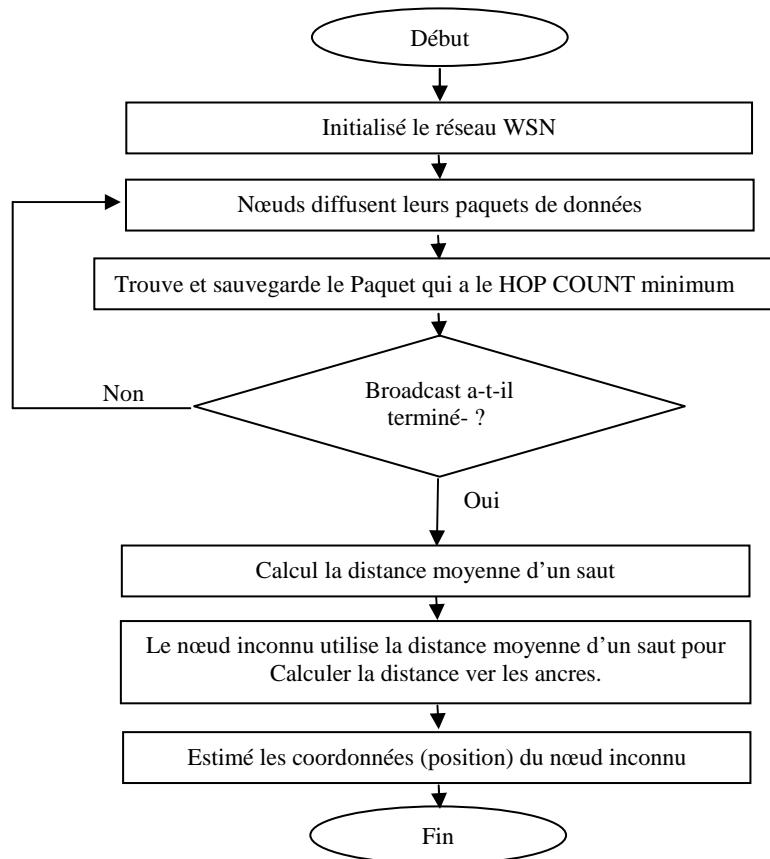


Figure 3.1 Organigramme DV-HOP de base

2-2. Détermination de la taille moyenne des sauts et de la distance entre les nœuds

Ici, chaque ancre détermine la distance moyenne des sauts par rapport au autres ancres du réseau, à l'aide de l'Equation (1) comme spécifié ci-dessous.

$$AHS_i = \frac{\sum_{i \neq j}^m \sqrt{(X_i - X_j)^2 + (Y_i - Y_j)^2}}{\sum_{i \neq j}^m h_{ij}} \quad (1)$$

AHS : AVERAGE HOP SIZE

Dans l'équation ci-dessus (X_i, Y_i) , les coordonnées du $i^{\text{ème}}$ nœud d'ancrage et (X_j, Y_j) sont les coordonnées d'un autre nœud d'ancrage, $h_{i,j}$ est le nombre de sauts entre ces nœuds d'ancrage. Après cela, chaque nœud d'ancrage diffuse la valeur calculé de la taille moyenne du saut ci-dessus Eq(1) Dans l'ensemble du réseau.

Maintenant, chaque nœud inconnu U va calculer la distance entre le $i^{\text{ème}}$ ancre en utilisant la taille moyenne de saut de l'ancre le plus proche par l'équation (2).

$$\text{Dist}_U = AHS_i \times h_{iU} \quad (2)$$

2-3. Calcul la position du nœud inconnu (U)

Nous calculons ici la position de U en utilisant les distances estimées dans la technique de multilatération. Ici (X_u, Y_u) sont les coordonnées de U , (X_{T_n}, Y_{T_n}) sont les coordonnées des nœuds d'ancrage et T_n est le nombre total de nœud d'ancrage, $Dist_i$ est défini comme la distance calculée à l'étape 2 ci-dessus et enfin, obtenez les coordonnées en utilisant Equation (3) ci-dessous.

$$\begin{bmatrix} (X_u - X_1)^2 + (Y_u - Y_1)^2 = dist_1^2 \\ (X_u - X_2)^2 + (Y_u - Y_2)^2 = dist_2^2 \\ \vdots \\ (X_u - X_{T_n})^2 + (Y_u - Y_{T_n})^2 = dist_{T_n}^2 \end{bmatrix} \quad (3)$$

L'équation (3) peut également être exprimée comme ci-dessous en l'élargissant sous forme d'équation (4).

$$\begin{bmatrix} X_1^2 - X_{T_n}^2 + Y_1^2 - Y_{T_n}^2 - dist_1^2 - dist_{T_n}^2 = 2 \times X_u \times (X_1 - X_{T_n}) + 2 \times Y_u \times (Y_1 - Y_{T_n}) \\ X_2^2 - X_{T_n}^2 + Y_2^2 - Y_{T_n}^2 - dist_2^2 - dist_{T_n}^2 = 2 \times X_u \times (X_2 - X_{T_n}) + 2 \times Y_u \times (Y_2 - Y_{T_n}) \\ \vdots \\ X_{T_n-1}^2 - X_{T_n}^2 + Y_{T_n-1}^2 - Y_{T_n}^2 - dist_{T_n-1}^2 - dist_{T_n}^2 = 2 \times X_u \times (X_{T_n-1} - X_{T_n}) + 2 \times Y_u \times (Y_{T_n-1} - Y_{T_n}) \end{bmatrix} \quad (4)$$

L'équation ci-dessus (4) peut en outre être présentée sous la forme $\boxed{AX_u = b}$, où

$$A = \begin{bmatrix} 2(X_1 - X_m) & 2(Y_1 - Y_m) \\ 2(X_2 - X_m) & 2(Y_2 - Y_m) \\ \vdots & \vdots \\ 2(X_{m-1} - X_m) & 2(Y_{m-1} - Y_m) \end{bmatrix} \quad (5)$$

$$b = \begin{bmatrix} X_1^2 - X_m^2 + Y_1^2 - Y_m^2 - dist_1^2 - dist_m^2 \\ X_2^2 - X_m^2 + Y_2^2 - Y_m^2 - dist_2^2 - dist_m^2 \\ \vdots \\ X_{m-1}^2 - X_m^2 + Y_{m-1}^2 - Y_m^2 - dist_{m-1}^2 - dist_m^2 \end{bmatrix} \quad (6)$$

$$X_u = \begin{bmatrix} X_u \\ Y_u \end{bmatrix} \quad (7)$$

$$X_u = (A^T A)^{-1} A^T b \quad (8)$$

2-4. Analyse mathématique pour l'erreur avec DV-Hop

Dans l'algorithme de base de DV Hop, il existe deux principales raisons de l'accumulation d'erreur, une est due au processus de sélection du nombre de saut minimum (HOP COUNT) et l'autre au calcul de la taille moyenne d'un saut (AHS_i). Dans l'algorithme standard, tous les nœuds inconnus utilisent le AHS pour calculer sa distance avec les nœuds d'ancrage selon l'étape 2. Il est supposé que tous les nœuds voisins sont à un pas de saut indépendamment de leur déploiement dans l'environnement réel et c'est l'une des causes de l'erreur.

Comme dans la Fig.3.1, trois nœuds d'ancrage (A₁, A₂, A₃) et un nœud inconnu (U). Selon l'approche traditionnelle de l'algorithme DV Hop, le saut entre A₁ et U est un (01) est le même entre U et C. Toutefois la réalité est autre chose selon la Fig. 3.1.

La distance de saut moyenne calculée de A₁, A₂ et A₃ est représentée ci-dessous. Ici, dans la partie dénominateur, on définit le nombre de sauts entre une ancre et une autre. Dans ce cas, il y a quatre sauts (4 m) à prendre par A₁ à A₂ : A₁-U-C-D-A₂ selon la Fig. 3.1 en utilisant l'équation (1).

Chapitre III : ETAT DE L'ART SUR LES AMELIORATIONS DE L'ALGORITHME DE LOCALISATION DV-HOP

Chaque ancre calcul sa taille moyenne de saut (AHS_i) avec les autres ancres selon la formule (1).

Suit au graphe de la figure 3.2 les distances et les sauts entre ancres sont reportés sur le tableau ci-dessous :

Tableau 5 : Distance et Nombre de sauts des ancres

	A1-A2	A1-A3	A2-A3
distance	50	50	40
Nbre de saut	4	4	4

Les distances de sauts moyennes calculés de A₁, A₂ et A₃ sont mentionnées ci-dessous.

$$A1 = (50 + 50)/(4 + 4) = 12.5 \text{ m}$$

$$A2 = (50 + 40)/(4 + 4) = 11.25 \text{ m}$$

$$A3 = (50 + 40)/(4 + 4) = 11.25 \text{ m}$$

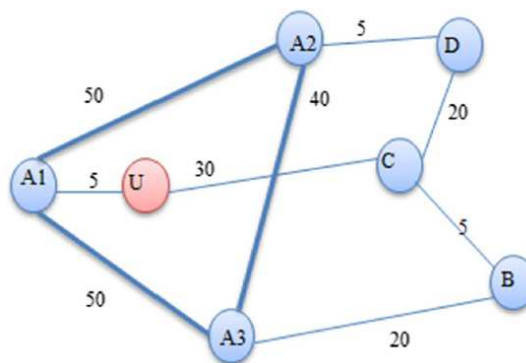


Figure 3.2 : Diagramme d'analyse de l'erreur pour l'algorithme DV hop

Le nœud U utilise la distance calculée du nœud ancre le plus proche (A1) est 12,5 m ci-dessus. Mais réellement la distance est de 5 m comme indiquée sur la Fig.3.1. Par conséquent, l'erreur de positionnement pour A1 est de 12,5 - 5 = 7,5 m.

Ainsi, tous les nœuds inconnus utilisent la valeur de saut d'ancre la plus proche qui est ajoutée à l'erreur estimée.

C'est ainsi que l'erreur augmente et la précision est réduite. Afin d'augmenter l'efficacité de l'algorithme DV hop, un nouvel algorithme basé sur la correction d'erreur est proposé plus bas.

3. IDV-HOP basé sur la pondération du HOP-SIZE

Cet algorithme (IDV-Hop) dans l'article [11] est une amélioration de DV-HOP de base, il est basé sur la pondération du saut moyenne.

La deuxième étape de l'algorithme DV-Hop, chaque nœud d'ancrage obtient l'emplacement des autres nœuds d'ancrage avec le nombre minimum de sauts, puis il calcule la distance moyenne des sauts et diffuse un paquet contenant l'information de cette distance moyenne au réseau.

Cependant, il y aura une grande erreur en utilisant la distance de saut moyenne du nœud d'ancrage le plus proche au lieu de la distance de saut moyenne de tous les nœuds d'ancrage impliqués dans la localisation.

Par conséquent, dans cet algorithme IDV-HOP, pour estimer la distance moyenne de saut, il n'utilise qu'une partie des nœuds d'ancrage qui ont un effet important sur la distance de saut moyenne du nœud inconnu, en introduisant un seuil (M). les ancrés qui ont le nombre de sauts avec le nœud inconnu inférieur a ce seuil (M), leur distance moyenne sont utilisées avec une méthode pondérée pour calculer la distance moyenne de saut qui sera utilisée par ce nœud inconnu. Les autres nœud d'ancrage qui ont le nombre de saut supérieur a M sont ignoré.

Sur cette base, la valeur de M devrait essayer de répondre au besoin de localiser tous les nœuds inconnus et de choisir le minimum, de sorte que la communication réseau soit relativement petite. Il convient de noter que la formule (3.1) est la plage de (M) dans le cas idéal.

$$\frac{1}{R} \sqrt{\frac{A * L_h}{T_n * L_p}} < M < H_{max} \quad (3.1)$$

Dont A est la zone du réseau; L_h est le nombre moyen de nœuds d'ancrage pour la nécessité de localiser chaque nœud inconnu, cela dépend du nombre total de nœuds dans le réseau; T_n est le nombre total de nœuds dans le réseau; L_p est le pourcentage de nœuds d'ancrage; H_{max} est le nombre maximum de sauts minimum entre le nœud inconnu et le nœud d'ancrage.

3-1. La distance de saut moyenne pondérée

Par exemple, la distance de saut moyenne du nœud inconnu est S, le nombre de sauts entre le nœud d'ancrage i et les nœuds inconnus est H_i . En supposant que le nœud inconnu a reçu les informations des nœuds d'ancrage, chaque valeur pondérée de nœud d'ancrage de la distance de saut moyenne est notée W_i , formule (3.2). A savoir, la valeur pondérée du nœud d'ancrage i est la valeur selon laquelle l'inverse de saut du nœud inconnu au nœud d'ancrage i divise la somme des inverses des sauts du nœud inconnu à chaque nœud d'ancrage. Ces poids ont ensuite été normalisés pour obtenir:

$$W_i = \begin{cases} \frac{1/H_i}{\sum_{j=1}^n 1/H_j}, & H_i \leq M \\ 0, & H_i > M \end{cases} \quad (3.2)$$

En fonction de la distance de saut moyenne de chaque nœud d'ancrage on calcule la distance de saut moyenne des nœuds inconnus selon la formule (3.3).

$$S = \sum_{i=1}^n W_i S_i \quad (3.3)$$

À savoir, la distance de saut moyenne du nœud inconnu est égale à la somme des produits de la valeur pondérée multipliée à la distance de saut moyenne de chaque nœud d'ancrage. Ainsi, la distance de saut moyenne de chaque nœud inconnu sera estimée avec plus de précision grâce au traitement pondéré. Il est plus précis et meilleur de refléter la distance de saut moyenne du réseau.

3-2. Correction de position

Afin d'améliorer encore la précision, cet algorithme doit estimer l'emplacement de chaque nœud ancre. Par rendre les nœuds inconnus qui viennent d'être estimés en tant que nœuds d'ancrage et rendre les nœuds ancrés comme nœuds inconnus, afin

de ré-estimées leurs coordonnées. Par exemple, un nœud d'ancrage B, utilisant sa distance de saut moyenne multiplier par le nombre de sauts qui le sépare du nœud inconnu pour obtenir la distance entre le nœud d'ancrage et le nœud inconnu, puis en utilisant la méthode d'estimation du maximum de vraisemblance pour estimer la position du nœud ancre. Ensuite, en comparant ce résultat au réel emplacement de B, et on obtient le facteur de correction suivant la formule (3.4) suivante:

$$\begin{aligned} x_i &= x_a - x_b \\ y_i &= y_a - y_b \end{aligned} \quad (3.4)$$

(x_a, y_a) est l'emplacement réel du nœud d'ancrage B, (x_b, y_b) est l'emplacement estimé pour B. quand le facteur de correction du nœud d'ancrage est obtenu, ce dernier le diffuse à travers le réseau. Le nœud inconnu reçoit tout les facteurs de correction des nœuds ancrés, ces facteurs sont pondérés puis la moyenne de facteur de correction est calculée. Le processus est le suivant (supposant que le nombre de nœuds d'ancrage dans le réseau est n:

$$W_j = \frac{1/H_j}{\sum_{i=1}^n \frac{1}{H_i}} \quad (3.5)$$

Ces poids ont ensuite été normalisés pour obtenir W'_j , puis le calcul de la moyenne du facteur de correction :

Utilisez ensuite l'équation suivante pour la correction de position du nœud inconnu:

$$\begin{aligned} x &= \sum_{j=1}^n (W'_j \times x_j) \\ y &= \sum_{j=1}^n (W'_j \times y_j) \end{aligned} \quad \Rightarrow \quad \begin{aligned} x' &= x + x \\ y' &= y + y \end{aligned}$$

(x,y) est la moyenne du facteur de correction du réseau, (X, Y) est l'emplacement estimé d'origine pour les nœuds inconnus, (X',Y') est l'emplacement corrigé pour les nœuds inconnus.

3-3. Résultat de la simulation.

Les expériences de simulation montrent que l'erreur de positionnement de ce algorithme amélioré est plus petit que l'algorithme DV-Hop traditionnel ou l'erreur moyenne de positionnement est réduite d'environ 15%. Nous pouvons donc dire que l'algorithme amélioré est plus prépondérant que l'algorithme DV-Hop traditionnel.

Cependant, la diminution de l'erreur de positionnement dépend des différentes valeurs de seuil. Dans cette expérience, nous pouvons voir que lorsque $M=6$, l'erreur de positionnement est la plus petite. Donc, il faut bien choisir le seuil M « pas trop grand et pas trop petit », le choix du seuil M dépend de la situation réelle du réseau.

4. Algorithme de localisation amélioré RF-DVHop

Cette approche [15] améliore l'algorithme de localisation basé sur DV-Hop avec RSSI et un mécanisme de rétroaction (RF-DVHop : RSSI et Feedback Mécanisme Based DV-Hop). RF-DVHop peut utiliser le RSSI et la rétroaction du biais de localisation pour augmenter la précision de localisation. Il peut être divisé en deux étapes.

-Utiliser le RSSI pour remplacer les nombres de sauts et puis calculer l'emplacement estimé du nœud inconnu.

-Création d'un canal de rétroaction entre l'ancre et le nœud inconnu, le nœud inconnu utilise le biais qui est la différence entre l'emplacement réel et l'emplacement estimé de l'ancre comme facteur d'ajustement et ajoute ce facteur d'ajustement dans le calcul du maximum de vraisemblance pour obtenir l'emplacement optimisé.

4-1. RF-DVHop avec RSSI Reformé

Les méthodes basées sur le nombre de sauts permettent facilement une déviation de distance. L'ajout de RSSI dans le calcul de la distance éliminera considérablement cet écart de distance qui résulte du fait que le nombre de sauts ne peut pas bien représenter l'état réel du réseau.

L'ajout du RSSI pour remplacer le nombre de sauts dans le calcul de la distance, doit d'abord connaître le modèle d'atténuation du signal. On choisit un modèle de distribution log-normal [16] dans RF-DVHop.

Le modèle de distribution log-normale fournit également un ensemble de paramètres pour l'ajustement, comme indiqué dans la formule (4.1) :

$$PL(d) = \overline{PL}(d_0) + 10\eta \lg\left(\frac{d}{d_0}\right) + X\sigma \quad (4.1)$$

$PL(d)$ est la perte de l'intensité pour distance d , $PL(d_0)$ est la perte pour d_0 est de 1 m, η est l'indice de perte sur le chemin, $X\sigma$ est la fonction de distribution aléatoire de Gauss tandis que la moyenne est de 0.

Pour utiliser plus précisément le RSSI pour calculer la distance entre le nœud inconnu et les balises, RF-DVHop utilise les deux mesures suivantes, la valeur RSSI pour chaque nœud récepteur et la moyenne de tous les RSSI sur l'itinéraire de routage appelé $RSSI_{avg}$.

Les ancres diffusent (Broadcast) leurs messages de localisation où le format est $\{IDB, (X_i, Y_i), Hops, RSSI_{pre}, RSSI_{total}\}$, où IDB est l'identifiant de la balise, (X_i, Y_i) est l'emplacement de la balise i , Hops est le nombre de sauts qui est la somme des nœuds sur lesquels le message a été transmis, $RSSI_{pre}$ est l'intensité du signal de réception du message provenant des voisins, $RSSI_{total}$ est la somme de tous les $RSSI_{pre}$.

Ainsi, nous pouvons obtenir le $RSSI_{avg}$, comme indiqué dans la formule (4.2):

$$RSSI_{avg} = RSSI_{total} / Hops \quad (4.2)$$

RSSI peut être plus puissant lorsque les nœuds se rapprochent et à l'inverse il s'affaiblit. Sur cette base, nous définissons W_R comme poids pour le calcul de la distance, comme indiqué dans la formule (4.3) :

$$W_{Rij} = \varepsilon \times \frac{RSSI_{avg}}{RSSI_{pre}} \quad (4.3)$$

Où W_{Rj} est le poids de la balise (j) qui se trouve dans l'itinéraire de routage, ϵ est le facteur d'ajustement du poids, et définissez $\epsilon = 1$ pour plus de simplicité. L'utilisation de RSSI pour remplacer le nombre de sauts pour le calcul de la distance, comme indiqué dans la formule (4.4) :

$$d_{Bi} = \sum_{j=1}^i (W_{Rj} \times C_{Bi}) \quad (4.4)$$

d_{Bi} est la distance entre la balise i et le nœud inconnu, C_{Bi} est la taille moyenne du saut entre la balise i et le nœud inconnu.

Cet algorithme utilise la méthode de maximum de vraisemblance pour calculer l'emplacement estimé du nœud inconnu en fonction des distances entre le nœud inconnu et chaque balise.

4-2. RF-DVHop avec Mécanisme de rétroaction (Feedback Mechanism)

Dans RF-DVHop, un nœud inconnu utilise le DV-Hop reformé par RSSI pour obtenir l'emplacement estimé. Il Construit un nouveau message avec l'emplacement et la taille du saut au format $\{C, (X, Y), Id\}$, et diffuse vers les balises. En ce qui concerne le nœud inconnu, C est la taille du saut, (X, Y) est l'emplacement estimé, (Id) est le numéro du nœud. Une fois que les balises ont reçu le message d'un nœud inconnu, les balises effectuent une opération de recherche dans la table de routage, trouvent le (Id) correspondant de saut minimum, considèrent le nœud inconnu comme une balise et utilisent le DV-Hop reformé pour calculer leur emplacement d'estimation.

Les balises définissent la différence entre l'emplacement précis intégré et l'emplacement estimé en tant que facteur d'ajustement P ($\Delta X, \Delta Y$), construisent un nouveau message $\{(\Delta X, \Delta Y), IdB\}$ et un retour vers un nœud inconnu. Enfin, le nœud inconnu ajoute P dans le DV-Hop reformé et calcule l'emplacement optimisé du nœud inconnu.

Ajout (P) dans la formule de localisation multilatérale, comme indiqué dans les formules (4.5) et (4.6), où (d) est le résultat de la distance réformée par RSSI.

$$\begin{cases} A_1x + B_1y = d_1^2 - d_n^2 + C_1 + D_1 \\ \vdots \\ A_{n-1}x + B_{n-1}y = d_{n-1}^2 - d_n^2 + C_{n-1} + D_{n-1} \end{cases} \quad (4.5)$$

$$\begin{aligned} A_i &= (2\Delta x_i - 2\Delta x_n + 2x_n - 2x_i), \\ B_i &= (2\Delta y_i - 2\Delta y_n + 2y_n - 2y_i) \\ C_i &= -(x_i^2 - x_n^2 - 2x_i\Delta x_i + 2x_n\Delta x_n + \Delta x_i^2 - \Delta x_n^2) \\ D_i &= -(y_i^2 - y_n^2 - 2y_i\Delta y_i + 2y_n\Delta y_n + \Delta y_i^2 - \Delta y_n^2) \end{aligned} \quad (4.6)$$

Sachant que $i = 1, 2, \dots, n - 1$.

Construire la matrice $JX = K$, comme indiqué dans la formule (4.7):

$$\begin{aligned} J &= \begin{bmatrix} A_1 & B_1 \\ \vdots & \vdots \\ A_{n-1} & B_{n-1} \end{bmatrix}, \\ K &= \begin{bmatrix} d_1^2 - d_n^2 + C_1 + D_1 \\ \vdots \\ d_{n-1}^2 - d_n^2 + C_{n-1} + D_{n-1} \end{bmatrix}, X = \begin{bmatrix} x \\ y \end{bmatrix} \end{aligned} \quad (4.7)$$

Enfin, nous pouvons obtenir l'emplacement du nœud inconnu en utilisant la méthode des moindres carrés pour résoudre la matrice (4.7), comme indiqué dans la formule (4.8) :

$$X^T = (J^T J)^{-1} J^T K \quad (4.8)$$

4-3. Résumer de RF-DVHop

RF-DVHop est principalement divisé en quatre étapes suivantes:

- Le nœud inconnu utilise RSSI pour remplacer le nombre de sauts dans DV_Hop, calcule l'emplacement estimé et le diffuse au format {C, (X, Y), ID}.
- Utilisation du nœud inconnu comme une balise, les balises utilisent la taille de saut moyenne C et le DV-Hop reformé par RSSI pour calculer leur emplacement estimé.
- Les balises définissent la différence entre l'emplacement précis intégré et l'emplacement estimé comme facteur d'ajustement (P), et la rétroaction (feedback) au nœud inconnu.
- Le nœud inconnu ajoute l'ajustement P dans le DV-Hop réformé par RSSI et obtient l'emplacement optimisé.

4-4. Résultats des simulations

En utilisant MATLAB comme plateforme de simulation pour RF-DVHop. Les simulations se concentrent sur l'erreur moyenne de localisation entre l'emplacement estimé et l'emplacement réel. un environnement représentatif est choisi pour les simulations: les nœuds sont disposés au hasard en aire carrée et le réseau est isotropique. La comparaison des performances des algorithmes se concentre sur RF-DVHop, DV-Hop.

Dans un réseau carré, RF-DVHop obtient une précision de localisation plus élevée que DV-Hop de base. L'erreur moyenne de localisation de RF-DVHop est inférieure sur 20 unités. Tant que le nombre total de balises est inférieur entre 3 et 5 balises, RF-DVHop peut toujours conserver une bonne précision de localisation. Lorsque le nombre total de balises a été augmenté à 15, les erreurs de localisation commencent à se stabiliser et l'erreur de localisation de RF-DVHop est toujours la plus petite.

Étant donné que le nouveau nombre de sauts a été réformé par RSSI, une nouvelle méthode de calcul de distance et le mécanisme de rétroaction basé sur un facteur d'ajustement, RF-DVHop réduit efficacement l'erreur de localisation. Mais, il y aura un petit retard dans une certaine mesure.

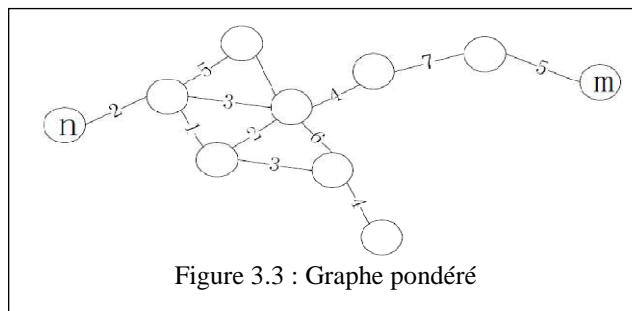
En outre, en comparaison avec la capacité de la régulation de rétroaction du facteur d'ajustement, l'impact du RSSI sur l'erreur de localisation est beaucoup plus important. On constate que le RF-DVHop contenant la fonction de reformation RSSI seulement donne une erreur de localisation beaucoup plus petite que le RF-DVHop contenant seulement le facteur d'ajustement.

En fin, les simulations montrent que RF-DVHop obtient moins d'erreur de localisation, une plage de variation plus petite de l'erreur de localisation, une vitesse de convergence plus rapide et une meilleure robustesse dans les réseaux carrés.

5. L'algorithme SD-DVHop (Shortest Distances DV-Hop)

Dans cette approche [14], l'idée provient du rapport entre la distance du plus court chemin et la distance droite entre les nœuds. En théorie des graphes, le problème du chemin le plus court est le problème de trouver un chemin entre deux nœuds dans un graphe afin que la somme des poids de ses arêtes constitutives soit minimisée [17].

Il existe plusieurs variantes selon que le graphe donné est non orienté, dirigé ou mixte. Pour les graphiques non orientés, le problème de chemin le plus court peut être formellement défini comme suit. Étant donné un graphe pondéré de la fig.3.3 (c'est-à-dire un ensemble N de nœuds, un ensemble E d'arêtes et une fonction de pondération à valeur réelle $f: E \rightarrow R$), et les éléments n et m de N et de trouver un chemin P de n à m de N tel que $\sum_{p \in P} f(p)$ est minime parmi tous les chemins qui lient n à m .



Parfois, le problème est également appelé problème de chemin le plus court à paire unique pour le distinguer du problème de chemin le plus court à toutes les paires. Le problème de la paire unique est toujours résolu par l'algorithme de Dijkstra. Le principe du chemin le plus court de toutes les paires est en fait l'approche consistant à exécuter un algorithme de chemin le plus court à paire unique sur tous les paires de nœuds. Cependant, afin de réduire la complexité de l'algorithme, nous adoptons l'algorithme de **JOHNSON** pour construire le chemin le plus court parmi tous les nœuds. Les poids de ses constituantes arêtes sont les distances obtenues à partir du modèle de distance RSSI entre les nœuds.

5-1. La distance du plus court chemin appliquée dans l'algorithme DV-Hop

Pour améliorer la précision de localisation de l'algorithme DV-HOP, cet algorithme SD-DVHop (Shortest Distances DV-Hop) modifie la distance de saut moyenne du réseau.

Dans l'algorithme primitif, la taille du saut moyenne est égale à la distance en ligne droite divisée par le nombre de sauts minimum entre les nœuds d'ancrage. En fait, la distance en ligne droite est assez différente de la somme des distances entre les nœuds sur le plus petit chemin de nombre de sauts. La taille de saut obtenue à partir de (fig.3.3) ne peut pas obtenir la taille de saut la plus appropriée à la nature du réseau. Afin de résoudre ce problème, cet approche choisit la distance du chemin le plus court plutôt que le HOP COUNT minimum et **affichez le rapport entre la distance du plus court chemin et la distance en ligne droite comme degré de contribution de la distance en ligne droite pour modifier la taille moyenne du réseau entier lors de l'obtention.**

Et par conséquent, Hopsiz_e (i) peut être défini comme (5.1):

$$Hopsiz_{e_i} = \frac{\sum_{i \neq j} D_{ij} * \frac{1}{w_{ij}}}{\sum_{i \neq j} h_{ij} * \frac{1}{w_{ij}}} \quad (5.1)$$

Où D_{ij} est la distance de chemin le plus court entre les nœuds de référence i et j .

Lorsque nous construisons le chemin le plus court, nous utilisons la distance estimée obtenue à partir du modèle de distance RSSI comme poids des arêtes.

Le processus de cet algorithme amélioré suit ces étapes :

- les nœuds d'ancrage diffusent des paquets d'informations comprenant leurs identifiants, leur localisation, le nombre de sauts initialisé à 0 et la distance estimée aux nœuds voisins selon le modèle de distance RSSI
- Selon l'algorithme JOHNSON pour construire le chemin le plus court et compter le nombre de sauts entre tous les nœuds
- Les nœuds inconnus doivent avoir reçu au moins trois informations de liaison, sinon le nœud ne peut pas être localisé.

- *Les nœuds ancres calculent leurs sauts moyens selon l'algorithme SDDV- HOP et les diffusent dans le réseau.*
- *Après avoir reçu le saut moyen, les nœuds inconnus calculent la distance aux nœuds ancres*

Pour chaque nœud inconnu, choisissez trois nœuds d'ancrage qui ont la distance la plus proche du nœud inconnu comme nœuds de référence pour calculer ses coordonnées selon la méthode des moindres carrés

5-2. Résultats de la simulation

Dans la littérature [14], pour Le réseau est un carré avec une longueur de côté de 100 m. La distribution de nœuds est Laissée aléatoirement avec une portée de transmission de 20 m dans le réseau, en utilisant MATLAB comme simulateur pour mesurer la précision de la localisation.

L'algorithme SD-DVHOP atteint de meilleures performances que l'algorithme DV-HOP. Par exemple, à 20% des nœuds d'ancrage, IL présente une erreur de localisation moyenne d'environ 28%, soit 22% plus précise que l'algorithme DV-HOP. Au fur et à mesure que le nombre de nœuds d'ancrage augmente, la taille moyenne se conforme mieux à la nature du réseau. Donc, comme prévu, l'erreur de localisation diminue avec l'augmentation du pourcentage de nœuds d'ancrage.

Pour une densité des nœuds varie de 50 à 300 nœuds capteurs et nous fixons le nombre de nœuds d'ancrage à 15. Les résultats montrent que l'erreur diminue avec l'augmentation de la densité des nœuds. L'erreur de localisation atteint le stade de stabilité quand le nombre de nœuds de capteur continue d'augmenter est atteint une valeur fixe.

Lorsque le nombre de nœuds de capteurs est beaucoup plus élevé, l'algorithme SD-DVHOP est 15% ~ 25% plus précis que l'algorithme DV-HOP. Lorsque le nombre est petit, l'algorithme SD-DVHOP est toujours 4% ~ 10% plus précis que l'algorithme DV-Hop.

6. Algorithme DEC-DVHop

(Distance Error Correction Based Hop Localization Algorithm for WSN)

Cette amélioration proposée dans [12] est basé sur la correction d'erreur de distance ou elle se concentre sur la réduction de l'erreur de distance et par conséquent une réduction d'erreur de localisation dans son ensemble.

Des modifications ont été faites sur l'algorithme de base de DV Hop comme indiqué plus bas.

6-1. Calcule du nombre de minimum de saut parmi les nœuds

Cet algorithme a le même fonctionnement pour cette phase, comme l'algorithme DV HOP de base déjà définit. Dans cette phase, chaque nœud d'ancrage calcule la valeur minimale du nombre de sauts par rapport aux autres nœuds d'ancrages présents dans la zone déployée.

6-2. Déterminer la moyenne de la taille du saut modifiée et la distance entre nœuds

Quelques modifications ont été apportées dans cette phase par rapport à l'algorithme de base. Après le calcul de la taille du saut, on calcule la valeur moyenne du saut selon la formule Eq.(6.1).

$$AHS_A = \frac{\sum_{A \neq T}^T \sqrt{(X_A - X_j)^2 + (Y_A - Y_j)^2}}{\sum_{A \neq T}^T h_{A_j}} \quad (6.1)$$

Le saut utilisé est celui du réseau et non pas de l'ancre le plus proche. Ensuite on trouve la moyenne du résultat calculé par EQ. (6.1) pour chaque ancre selon la formule Eq.(6.2) :

$$\Delta NHD = \frac{\sum_{A \neq T}^T AHS_A}{T} \quad (6.2)$$

L'utilisation de la taille du saut de l'ancre le plus proche génère une erreur de distance comme il a été mentionné dans l'algorithme DV Hop de base. Normalement, la taille du saut varie en fonction de la distance réelle de l'environnement.

Chapitre III : ETAT DE L'ART SUR LES AMELIORATIONS DE L'ALGORITHME DE LOCALISATION DV-HOP

Où (X_a, Y_a) les coordonnées de l'ancre A, et (X_j, Y_j) d'autres nœuds d'ancrage. H_{AJ} définit les sauts entre ancrés et T est le nombre total des ancrés.

En outre, l'erreur de distance (DE_A) est calculée à l'aide d'Eq. (6.3) Qui est le résultat principal de cette amélioration. Ici, nous trouvons l'erreur de la distance du saut produit lors de l'utilisation de la distance de saut moyenne en cas du DV HOP.

$$DE_A = (ANHD \times h_{AJ}) - \sqrt{(X_A - X_j)^2 + (Y_A - Y_j)^2} \quad (6.3)$$

Ensuite, nous soustrayons la distance réelle de la distance basée sur le saut calculée en DV-Hop. Cette valeur est diffusée par chaque ancre dans tout le réseau afin que les mêmes informations soient connues de chaque ancre présente dans le réseau.

Maintenant, nous trouvons l'erreur moyenne (AEA_A) de chaque ancre avec les autres ancrés à selon Eq. (6.4). Si pour la même erreur de distance du nombre de sauts est différent alors prendre la moyenne des différentes erreurs correspondant au nœud d'ancrage pour le même nombre de sauts.

$$AEA_A = \frac{\sum_{A \neq 1}^T DE_A}{T} \quad (6.4)$$

Ensuite, le nœud inconnu trouve la distance ($D_{U, A}$) entre les ancrés avec Eq.(6.5). Comme dans l'algorithme DV Hop, la distance du saut moyenne du réseau est obtenu est multiplié avec le nombre de sauts pour trouver la distance entre le nœud inconnu et l'ancre.

$$D_{UA} = DE_A - AEA_A \quad (6.5)$$

Mais dans cet algorithme, pour réduire l'erreur de distance qui s'est accumulée, on multiplie l'erreur totale avec le nombre de sauts et la soustraire de la distance basée sur le nombre des sauts.

6-3. Estimation de la valeur de la position

Les coordonnées du nœud U doivent être calculées à l'aide des mêmes méthodes que celles utilisées par l'algorithme DV HOP de base, à savoir la méthode des moindres carrés ou multilatération.

6-4. Résultats et comparaisons

Cet algorithme conduit à moins de génération d'erreur par rapport à DV Hop. Ceci est principalement dû au facteur de correction d'erreur introduit. De cette manière, la réduction des erreurs de distance est effectuée par le nœud inconnu. Cette distance joue un rôle important dans le calcul des coordonnées.

Ainsi, avec la réduction de l'erreur de distance, l'erreur de localisation globale est réduite.

De plus, cet algorithme est comparé à des algorithmes comme DV-Hop[18], IDV-Hop[19] et PSO [20] dans l'environnement MATLAB. Dans toutes les simulations, on ose dire que cet algorithme amélioré a une précision supérieure par rapport aux autres algorithmes.

7. CHECKOUT_DV-Hop

Dans la littérature [21], DV-HOP est une méthode de localisation fréquemment utilisée, l'idée clé de DV-HOP est de calculer la distance approximative entre le nœud inconnu N_x et chaque ancre A_i , en multipliant le nombre de sauts minimal par la distance moyenne par saut. Cela signifie $D_{i,N_x} = \text{hop}_{i,N_x} \times d_{phi}$ où d_{i,N_x} est la distance approximative entre N_x et A_i hop_{i,N_x} est le nombre de sauts minimum entre N_x et A_i , d_{phi} est la distance approximative moyenne par saut sur A_i . Ici, i appartient à l'ensemble $[1,2,\dots,m]$, si le nombre total des ancrs est m .

Comme d_{i,N_x} est un paramètre fondamental pour le calcul de la position du nœud normal N_x il a une influence considérable sur la précision de DV-HOP. On note la distance réelle entre N_x et A_i : $d_{i,N_x\text{true}}$ est la différence entre $d_{i,N_x\text{true}}$ et d_{i,N_x} :

$\Delta_{di,Nx}$ Evidemment, $\Delta_{di,Nx}$ influence directement l'imprécision de DV HOP. On note la différence entre d_{hp} et sa valeur réelle : Δd_{hp} , puis nous obtenons : $\Delta d_{i,Nx} = \text{hop}_{i,Nx} * \Delta d_{phi}$. donc, lorsque $\text{hop}_{i,Nx}$ augmente, Δd_{hp} augmente également, et la précision DV HOP devient plus faible. Si A_{near} est l'ancre la plus proche de N_x parmi toutes les ancres possibles, corrélativement, $\text{hop}_{near, Nx}$ est la plus petite valeur. Finalement, $\Delta d_{near, Nx}$ est plus précise. En fonction de cette déduction, cet algorithme CHECKOUT DV-HOP [21] tente de profiter au maximum de $d_{near, Nx}$ qui est la valeur relativement la plus faible.

Dans la figure 3.4, nous illustrons le principe de CHECKOUT DV-HOP. Cette méthode n'ajoute qu'une étape à DV HOP. En utilisant DV-HOP, le nœud normale N , obtient sa position estimée notée N_{DV-HOP} avec les coordonnées (x',y') . Ensuite, N_x calcule la distance entre N_{DV-HOP} et A_{near} , notée d_{DV-HOP} . Notons que N_x a évalué sa distance à A_{near} notée $d_{near,Nx}$. Par la suite, N_x exécute l'étape « checkout ». Le but de cette étape est de déplacer la position estimée de N_{DV-HOP} vers une nouvelle position $N_{checkout}$ dont la distance à A_{near} est $d_{near, Nx}$. Pour aboutir à cet objectif, la méthode la plus facile et la plus rapide est de déplacer la position le long de la droite reliant N_{DV-HOP} et A_{near} . Et sur cette droite, la distance entre N_{DV-HOP} et A_{near} est $d_{near, Nx}$.

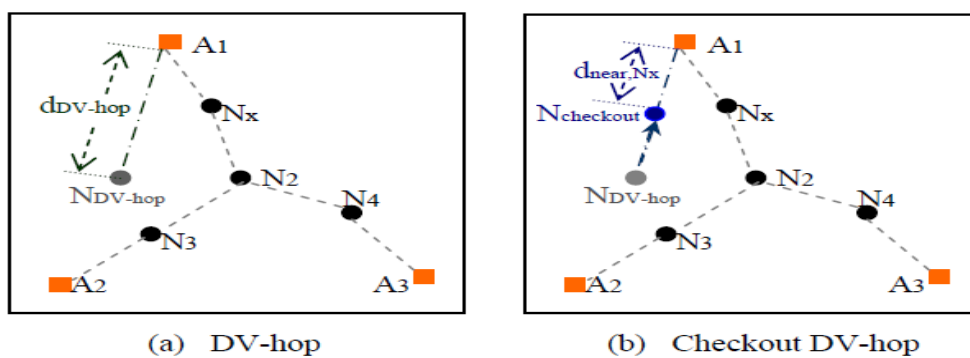


Figure 3.4 : Principle of Checkout DV-hop

8. Hybride DV-Hop Algorithme utilise RSSI

Dans l'article [22], un algorithme DV-Hop amélioré nommé algorithme Hybrid DV-Hop est proposé. Il utilise les mesures RSSI entre nœuds voisins (à un seul saut) pour estimer la distance au lieu que les distances moyennes par saut. De plus, il permette également d'exploiter les nœuds capteurs déjà localisés les faisant devenir des nœuds d'ancrage supplémentaires pour la localisation ultérieure des nœuds capteurs restants.

Cet algorithme hybride utilise deux étapes supplémentaires lors de la localisation les nœuds on utilisant l'algorithme DV-Hop.

Dans la première étape, il utilise les valeurs RSSI pour estimer les distances entre les nœuds d'ancrage et leurs nœuds capteurs voisins à un saut, puis utiliser ces distances ainsi estimée plutôt que d'utiliser la distance de saut moyenne comme dans l'algorithme DV-Hop original.

Les expériences, indiquent que, pour les nœuds qui sont à un saut d'une ancre (hop-count = 1), la distance estimée à l'aide du RSSI est plus précise que la distance estimée à l'aide de la distance moyenne par saut utilisée dans l'algorithme DV-Hop.

Deuxièmement, lorsqu'un nœud est directement connecté (un saut) au moins trois ancres différentes, son emplacement est calculé en utilisant la Multilatération basée uniquement sur les distances estimées basées sur le RSSI à partir de ces ancres.

une fois qu'un nœud soit localisé il est promu pour agir comme un point d'ancrage supplémentaire, qui est par la suite utilisé pour localiser ses nœuds de capteurs voisins. Cette méthode fournit des ancres supplémentaires a fin d'améliorer la précision en localisant les nœuds de capteurs restants. Cela est particulièrement utile dans réseaux sans fil avec une densité de nœuds d'ancrage inférieure.

De plus, Afin d'exploiter cet avantage, les nœuds de capteurs sans fil soient localisés dans une manière telle que les nœuds proches des ancres soient localisés en premier.

Plus précisément, chaque ancre envoie un message `LocStartMsg` à ses voisins afin de démarrer la localisation. Le message `LocStartMsg` contient principalement le champ suivant : `LocStartMsg = { fnodeID, Type, nodeCoordinate }`, où `nodeID` : représente l'identité du nœud qui envoie le message, `Type` : spécifie que le nœud est ancre ou non, et `nodeCoordinate` les coordonnées du nœud.

Chaque nœud qui reçoit ce message mémorise l'identité du nœud voisin et sa position estimée. Une fois qu'un nœud a au moins trois ancres et/ou voisins, il peut estimer sa position, puis transmet le message à ses voisins. Par cette technique, on s'assure que les nœuds les plus proches des nœuds d'ancrage sont localisés en premier et que la localisation se fait progressivement.

Notez que, chaque nœud maintient un `LocTable` qui initialement, contient la liste des ancres accessibles avec leurs coordonnées et `Hopsiz`. Cette table est mis à jour chaque fois que le nœud reçoit un message `LocStartMsg` d'un voisin en stockant le `nodeID` et le `nodeCoordinate` du nœud voisin dans cette table.

Dans cet article [22] les performances de cet algorithme sont comparées avec l'algorithme DV-Hop de base et deux variantes récemment publiées de celui-ci, à savoir RSSI Auxiliary Ranging [23] et l'algorithme Selective 3-Anchor DV-hop [24].

L'évaluation des performances des quatre algorithmes par le biais de simulations a indiqué que cet algorithme surpasse considérablement ses homologues. Plus précisément, il améliore la précision de localisation dans certains scénarios de près de 95 %, 90 % et 70 % par rapport aux algorithmes DV-Hop de base, Selective 3-Anchor et RSSI DV-Hop, respectivement.

9. CWDV-Hop

(A Hybrid localization algorithm with Distance Weight DV-Hop and CSO for WSN)

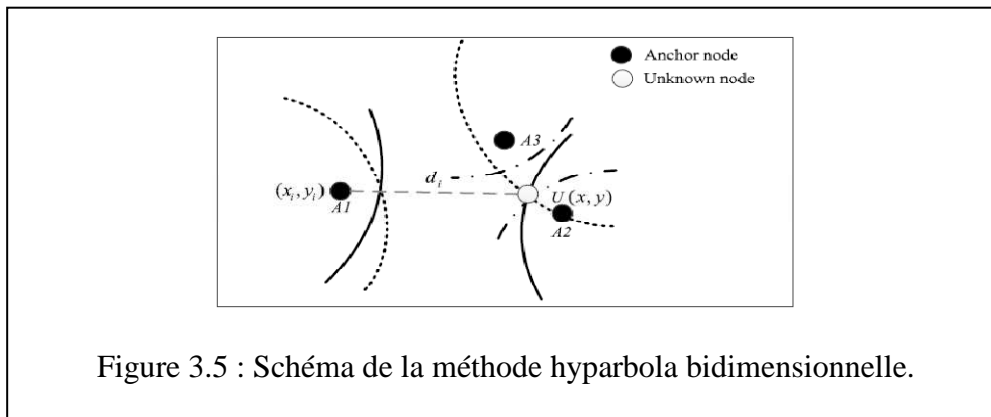
L'approche CWDV-HOP a été proposée [25] pour améliorer la précision de la localisation. La première est celle de DV-HOP de base, la deuxième étape est de

calculer le « Distance-weighted hop distance » la *distance pondéré de distance de saut* en introduisant le nouveau facteur de pondération de distance de saut à un seul nœud en utilisant la formule (9.1) et la moyenne de la distance de saut en utilisant la formule (9.2).

$$H'_i = \sum_{j=1}^m \left(\frac{d_{ij}}{\sum_{i \neq j} d_{ij}} \times \frac{d_{ij}}{h_{ij}} \right) = \sum_{j=1}^m (\omega_{ij} \times H_{ij}) \quad (9.1)$$

$$H_i = \frac{\sum_{i \neq j} \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}}{\sum_{i \neq j} h_{ij}} \quad (9.2)$$

La troisième étape, estimation de l'emplacement du nœud inconnu, une approche hyperbolique bidimensionnelle est utilisée pour obtenir les coordonnées estimées des nœuds inconnus.



La quatrième étape ; consiste à optimiser l'estimation des coordonnées obtenues lors de l'étape 3 pour trouver le meilleur résultat. La mise en œuvre de l'optimiseur CSO est comme suit:

Initialisation : Les paramètres du CSO (taille de l'essaim de poules, les itérations maximales, l'intervalle de temps de mise à jour de l'état de l'essaim de poule, le nombre de coqs, les poules et les poussins).

Calcul les valeurs de fitness de toutes les poules: Dans WSN, le problème de localisation est formalisé comme un problème d'optimisation où le l'erreur de localisation est considérée comme **une fonction objectif**.

Mise à jour l'emplacement des coqs, poules et poussins.

Fin de processus de mise à jour l'itération s'arrête lorsque les conditions de fin sont remplies.

le meilleur emplacement trouvé actuellement par l'essaim de poules sera considéré comme le résultat final de chaque nœud inconnu.

Il convient de noter que le processus d'estimation d'emplacement de chaque nœud inconnu est indépendant les uns des autres. Cela signifie que chaque nœud inconnu nécessite une particule CSO solitaire pour obtenir ses coordonnées.

10. NDLR_DV-Hop

(Nodes Distribution Law and Revised HOP-Size of DV-HOP)

Cette approche dans la littérature [26] s'appuie sur des règles de distribution des nœuds que DV_HOP en dispose, ces règles de distribution qui sont stables autant pour les emplacements des nœuds inconnus ayant la même erreur de localisation qu'aux emplacements des nœuds de balise ayant la même taille de saut. Sur cette base un nouvel algorithme de localisation s'est proposé pour améliorer la précision de localisation. L'idée principale est de faire une révision de la taille de saut selon la règle de distribution et appliquer un algorithme approprié en fonction de l'emplacement de la zone de surveillance.

L'algorithme de cette littérature [26] se **base sur la règle de distribution des nœuds (réguliers et ancrés) et sur la taille de sauts révisée**. Selon les résultats des expériences, il y en a des règles stables, les nœuds inconnus avec une erreur importante se répartissent principalement sur la frontière de la zone WSN et ceux avec une erreur plus petite principalement distribuer au milieu de la zone WSN.

D'après plusieurs simulations, une comparaison est faite aux coordonnées réelles des nœuds inconnus avec celle estimées qui sont calculés par DV-HOP. La méthode de calcul des erreurs de localisation est selon l'expression (10.1), le calcul de l'erreur moyenne de la localisation est selon l'expression (10.2). Les nœuds inconnus sont classés en deux classes selon l'erreur moyenne de localisation. Une classe contient les erreurs les plus importantes que l'erreur moyenne tandis que l'autre correspond aux erreurs qui sont inférieures à l'erreur moyenne.

$$LE = \frac{\sqrt{(Xr - Xe)^2 + (Yr - Ye)^2}}{ComR} \quad (10.1)$$

$$LEa = \frac{\sum_{i=1}^n \sqrt{(Xr_n - Xe_n)^2 + (Yr_n - Ye_n)^2}}{n * ComR} \quad (10.2)$$

Dans DV-HOP de base, les nœuds inconnus obtiennent la taille du saut à partir des nœuds de balise voisins alors on déduit que la distribution de l'erreur coïncide avec l'emplacement des nœuds inconnus, de sorte que la raison qui provoque l'erreur accompagnant l'emplacement c'est bien l'erreur d'accumulation de la taille du saut du nœud balise dans une zone particulière.

L'estimation par le maximum de vraisemblance (MLE) est utilisée dans le calcul d'estimation des coordonnées, les nœuds inconnus calculent leurs distances avec tous les nœuds balises. Les nœuds inconnus au frontière de la zone conduisent à des erreurs importantes par l'expression de la taille du saut multiplié par le nombre de saut en raison de l'accumulation de sauts très grands avec les nœuds balises de l'autre côté de la zone que provoque la règle de distribution des nœuds inconnus avec une grande erreur de localisation.

Les étapes de l'algorithme amélioré

Les auteurs déterminent la portée de l'anneau de distribution des positions des nœuds balises en calculant les coordonnées.

Les nœuds balises hors de l'anneau demandent la taille du saut de ceux de l'anneau afin de modifier leur taille de saut et réduire ainsi l'accumulation de

l'erreur produite. Un nouvel algorithme est proposé selon la théorie ci-dessus pour améliorer la précision de localisation.

Étape 1 : Calculer les sauts minimaux entre nœuds inconnus et chaque nœud de balise. Les nœuds de balise diffusent leurs informations de position à l'ensemble du réseau et enregistrent le **hop-count** qui est 0 comme valeur par défaut.

Le nœud de réception enregistre les informations de position et les sauts minimaux d'un nœud de balise et ignore les informations avec plus de sauts du même nœud de balise. Et puis le nœud de réception ajoute 1 à le hop-count et transmet les informations au prochain nœud. Après cette étape, chaque nœud acquiert les informations de positions et les sauts minimaux de chaque nœud de balise dans le réseau.

Étape 2 : Chaque nœud de balise calcule sa taille de saut selon les informations de position et de saut minimum enregistrées à partir de l'étape 1.

Selon la distribution des nœuds balises, l'étendue de la distribution peut être déterminée. Les nœuds balises hors de la zone anneau demandent la taille du saut de ceux dans la zone anneau pour modifier l'accumulation de l'erreur afin de trouver une taille de saut plus précise, et alors la taille du saut est plus similaire à la taille moyenne du saut.

Les nœuds de balise diffusent la nouvelle taille de saut avec champ de cycle de vie au réseau voisin. Les nœuds inconnus reçoivent et enregistrent les premières informations qui arrivent au plus tôt. Après cela, les nœuds inconnus calculent la distance avec les nœuds de balise sélectionnés par la nouvelle taille de saut et le nombre de saut.

Étape 3 : Calculer les coordonnées estimées du nœud inconnu en utilisant MLE.

Étape 4 : Calculez les coordonnées des nœuds inconnus en bordure de la zone par multilatération.

11. RWDV-Hop (WEIGHTED DV-HOP ALGORITHM BASED ON RSSI)

Un algorithme DV-Hop basé sur RSSI et avec l'optimisation sans contrainte pour la localisation dans WSN.

Dans la littérature [27], Pour améliorer la précision de la localisation, un algorithme DV-Hop pondéré basé sur l'indicateur d'intensité du signal reçu (RSSI) était proposé, il comporte trois étapes :

Premièrement, le RSSI est utilisé pour pondérer le nombre de sauts entre les nœuds. Ensuite, une distance de saut moyenne corrigée est obtenue. Enfin, l'optimisation sans contrainte est utilisée pour améliorer la précision de la localisation. Les résultats de simulation présentent de performance de localisation qui montre que l'algorithme de cette littérature est supérieur au DV-Hop et a un autre DV-Hop pondéré connus appelé (WDV-Hop).

11-1. Saut pondéré basé sur le RSSI

Le modèle de canal le plus populaire pour la transmission du signal est le modèle d'affaiblissement du trajet d'ombrage log-normal [16] qui est donné par formule(11.1) :

$$[P_r(d)]_{dBm} = [P_r(d_0)]_{dBm} - 10\eta \log\left(\frac{d}{d_0}\right) + X_\sigma \quad (11.1)$$

où $P_r(d_0)$ est la puissance du signal reçu à une distance de référence (d_0) entre l'émetteur et le récepteur, $P_r(d)$ est la puissance du signal reçu à une distance (d) .

η est l'exposant de l'affaiblissement sur le trajet.

X_σ est une variable aléatoire gaussienne d'une moyenne nulle avec écart type σ

η est typiquement compris entre 1,5 et 5, et σ est compris entre 4 et 12 dBm.

Dans [29], il a été montré que pour une erreur RSSI donnée, une erreur d'estimation de localisation plus importante se produit avec une plus grande distance entre les nœuds. Ainsi à la premier étape de l'algorithme, $RSSI(j)$ est ajouté au message diffusé qui correspond à la puissance du signal reçu après la jème

diffusion du nœud d'ancrage A_k au nœud inconnu U_i . Comme DV-Hop, un message est rejeté par un nœud si le nombre de sauts est supérieur à un autre message pour le même nœud d'ancrage, mais le nombre de sauts est maintenant :

$$h_j = h_{j-1} + H_j h_1 \quad (11.2)$$

où h_1 est le nombre initial de sauts donné par

$$h_1 = 10^{\frac{RSSI_0 - RSSI_1}{10\eta}}$$

Et

$$H_j - d_j / d_1 = 10^{\frac{RSSI_1 - RSSI_j}{10\eta}}$$

avec $RSSI_0$ la référence RSSI obtenue à partir de (11.1) en utilisant la plage de communication R de nœud.

11-2. Amélioration du calcul de la distance moyenne des sauts

Ici, un facteur de correction est utilisé pour améliorer la distance de saut moyenne comme dans l'algorithme WDV-Hop. Ce facteur est donné par

$$\delta = \frac{\sum_{k \neq l} |d_{kl} - \hat{d}_{kl}| / h_{kl}}{\sum h_{kl}}$$

Où d_{kl} est la distance réelle entre les deux nœuds d'ancrage A_k et A_l ,

\hat{d}_{kl} est la distance estimée obtenue en utilisant le nombre de sauts minimal multiplier par le Hopsizemoyenne comme le DV-Hop, puis le h_{kl} est obtenu de (11.2). Ensuite, la distance moyenne de saut corrigée est :

$$c = \bar{c} + \lambda \delta \quad (11.3)$$

où $\bar{c} = \sum_{k=1}^N Hopsizemoyenne_k / N$ est la distance moyenne des tous les sauts moyenne de tous le réseaux,

et λ est un paramètre utilisé pour équilibrer le distance de saut moyenne dans le réseau [28]. Elle concernait l'environnement du réseau, elle est déterminée

expérimentalement. Les expériences indiquent que les meilleurs résultats sont obtenus lorsque $\lambda = 0,6$.

Donc en résumé, la distance entre le nœud inconnu U_i et le nœud d'ancrage A_k est estimée en utilisant le saut moyen du réseau obtenu par (11.3) multiplié par le nombre minimal des sauts obtenu itérativement en utilisant (11.2).

11-3. Optimisation sans contrainte :

L'optimisation des moindres carrés peut accumuler des erreurs, de sorte que la méthode d'optimisation sans contrainte de [30] est utilisée pour l'estimation de la localisation à troisième étape.

Premièrement, les erreurs dans les distances estimées entre le nœud inconnu U_i et les nœuds d'ancrage A_k , $k = 1, 2, \dots, N$ sont séparés, ce qui donne

$$\begin{cases} (x - x_1)^2 + (y - y_1)^2 = (d_1 + e_1)^2 \\ (x - x_2)^2 + (y - y_2)^2 = (d_2 + e_2)^2 \\ \vdots \\ (x - x_N)^2 + (y - y_N)^2 = (d_N + e_N)^2 \end{cases}$$

où (x, y) et (x_k, y_k) sont les positions de U_i et A_k respectivement,

d_k , $k = 1, 2, \dots, N$ est la distance, et e_k est l'erreur correspondante. Soustraire la dernière équation des $N-1$ premières équations et en simplifiant donne un système d'équations surdéterminé. Définir les constantes a_i , b_i , c_i , c et D_i , $i = 1, 2, \dots, N-1$, comme

$$a_i = \frac{2(x_N - x_i)}{d_N^2}, b_i = \frac{2(y_N - y_i)}{d_N^2}, c_i = \frac{2d_i}{d_N^2}, c = \frac{2}{d_N}$$

$$D_i = \frac{(x_N^2 + y_N^2) - (x_i^2 + y_i^2) + d_i^2}{d_N^2} - 1$$

Ensuite, le système d'équations surdéterminé peut être écrit sous forme matricielle comme

$$\mathbf{Gz} = \mathbf{d} \quad (11.4)$$

Ou

$$\mathbf{G} = \begin{pmatrix} a_1 & b_1 & c_1 & 0 & \dots & 0 & c \\ a_2 & b_2 & 0 & c_2 & \dots & 0 & c \\ \vdots & \vdots & \vdots & \vdots & \dots & \vdots & \vdots \\ a_{N-1} & b_{N-1} & 0 & 0 & \dots & c_{N-1} & c \end{pmatrix}$$

$$\mathbf{z} = [x \quad y \quad e_1 \quad \dots \quad e_N]^T$$

$$\mathbf{d} = [D_1 \quad D_1 \quad \dots \quad D_{N-1}]^T$$

L'équation (11.4) peut être écrite comme un problème de programmation quadratique

$$\begin{cases} \min \mathbf{e}^T \mathbf{W} \mathbf{e} \\ \text{s.t.} \\ \mathbf{Gz} = \mathbf{d} \end{cases} \quad (11.5)$$

où $\mathbf{e} = [e_1 \ e_2 \ \dots \ e_N]$ et \mathbf{W} est une matrice de pondération liée au nombre de sauts h_{IK} défini comme

$$\mathbf{W} = \begin{pmatrix} 1/h_{i1} & 0 & \dots & 0 \\ 0 & 1/h_{i2} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1/h_{iN} \end{pmatrix}$$

La position inconnue du nœud peut être estimée en résolvant (11.5), mais ce problème est trop complexe pour les nœuds capteurs avec une énergie limitée. Ainsi, (11.5) est transformé en le problème d'optimisation sans contrainte [30] suivant :

$$\min \left(\mathbf{e}^T \mathbf{W} \mathbf{e} + \nu \|\mathbf{Gz} - \mathbf{d}\|^2 \right) \quad (11.6)$$

où $v > 0$ est une constante qui peut être obtenue expérimentalement. les meilleurs résultats sont obtenus lorsque $v = 200$. La résolution (11.6) donne

$$\mathbf{z} = (\mathbf{P} + v\mathbf{G}^T\mathbf{G} + \Delta\mathbf{I})^{-1} v\mathbf{G}^T\mathbf{D}$$

où $\Delta\mathbf{I}$ est un terme de régularisation avec $0 < \Delta$ et \mathbf{I} est la matrice identité d'ordre $n+2$, et

$$\mathbf{P} = \begin{pmatrix} \mathbf{0}_{2 \times 2} & \mathbf{0}_{2 \times n} \\ \mathbf{0}_{n \times 2} & \mathbf{W}_{n \times n} \end{pmatrix}$$

11-4. Les résultats de la simulation

Les performances des algorithmes DV-Hop, WDV-Hop et ce RWDV-Hop-U sont évaluées et comparées par simulation. De plus, DV-Hop et WDV-Hop ont été implémentés à l'aide d'une optimisation sans contrainte, et sont appelés respectivement DV-Hop-U et WDV-Hop-U.

Les résultats des simulations montrent que l'algorithme RWDV-Hop-U fournit des erreurs de localisation les plus bas et que les algorithmes qui utilisent une optimisation sans contrainte surpassent ceux qui utilisent les moindres carrés.

En outre, l'utilisation du RSSI dans RWDV-Hop-U entraîne une amélioration par rapport à WDV-Hop-U d'environ 0,05 avec 150 nœuds et de 0,10 avec 400 nœuds.

12. SELECTIVE 3-ANCHOR DV-HOP

Cet algorithme [21] peut augmenter la précision en détriment de la complexité de calcul.

Son principe est : Le nœud régulier sélectionne toutes les trois ancrs possibles afin de former des « 3-anchor groups » puis il calcule les positions estimées grâce à ces « 3-anchor groups ». Enfin, en fonction de la relation entre les positions estimées et les connectivités, le nœud régulier choisit la position la plus précise.

Considérons un réseau avec m_d ancrs A_1, A_2, \dots, A_{m_d} . En utilisant l'algorithme DV-HOP, un nœud normal N_x peut calculer sa position estimée N_{DV-HOP} en fonction

de ses distances estimées aux ancres. Ainsi, la précision de ces distances estimées a une influence importante sur la précision de DV-HOP.

En fait, au lieu d'utiliser toutes les m_d distances estimées, trois (03) distances sont suffisantes pour N_x pour calculer sa position. Par exemple, nous pouvons utiliser d_{i,N_x} , d_{j,N_x} , d_{k,N_x} qui sont les trois distances estimées entre N_x et les trois (03) ancres A_i , A_j , A_k . Puis en se basant sur la méthode MLE (Estimation par le Maximum de Vraisemblance), nous pouvons obtenir une « 3-anchor estimated position » de N_x notée comme $N_{\langle i,j,k \rangle}$.

Le principe de *selective 3-anchor DV-HOP* est de sélectionner la plus précise « 3-anchor estimated position ». Ici, le critère de sélection est la connectivité. Dans l'algorithme DV-HOP, la connectivité de N_x est définie comme les nombres de sauts minimum entre N_x et des ancres. Par exemple, si dans un réseau, il y a m_d ancres au total, et si le nombre de sauts minimum entre N_x et chaque ancre A_i est hop_{i,N_x} alors la connectivité de N_x est $[hop_{1,N_x}, hop_{2,N_x}, hop_{3,N_x}, \dots, hop_{m_d,N_x}]$. La relation entre la connectivité et la distance est donnée comme suit :

Alors plus la différence de connectivité entre deux nœuds est petite, plus la distance entre ces deux nœuds est petite.

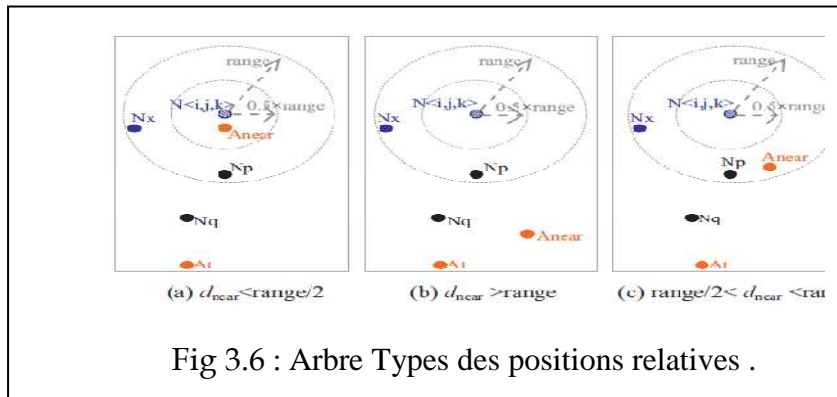
Selon cette relation, la « 3-anchor estimated position » ayant la connectivité la plus similaire à N_x devait être la plus proche de N_x .

Ainsi, le principe de base de cet algorithme « *Selective 3-Anchor DV-HOP* » est de choisir la « 3-ancre position estimée » qui a la connectivité la plus semblable à N_x .

Cependant, la connectivité de « 3-anchor estimated position » $N_{\langle i,j,k \rangle}$ est encore inconnue. Par conséquent, on a proposé la méthode suivante pour calculer le nombre de sauts entre $N_{\langle i,j,k \rangle}$ et chaque ancre, notée $hop_{\langle i,j,k \rangle, t}$. On note la distance entre $N_{\langle i,j,k \rangle}$ et chaque ancre A_t : $d_{\langle i,j,k \rangle, t}$. Ainsi, si N_x connaît la distance par saut entre $N_{\langle i,j,k \rangle}$ et A_t comme étant $hop_{\langle i,j,k \rangle, t} = \frac{d_{\langle i,j,k \rangle, t}}{dph_{\langle i,j,k \rangle, t}}$. Il faut donc trouver comment estimer $dph_{\langle i,j,k \rangle, t}$.

Chapitre III : ETAT DE L'ART SUR LES AMELIORATIONS DE L'ALGORITHME DE LOCALISATION DV-HOP

En fait, N_x connaît seulement les distances par saut de chaque ancre : $dph_1, dph_2, dph_3, \dots, dph_{md}$, y compris la distance par saut de A_t notée dph_t . Ainsi on doit estimer $dph_{\langle i,j,k \rangle, t}$ basé sur $dph_1, dph_2, dph_3, \dots, dph_{md}$. Pour cela, trois (03) types de relations entre $N_{\langle i,j,k \rangle}$ et sa plus proche ancre A_{near} sont considérés, en fonction de leur distance. Dans le premier cas, la distance entre $N_{\langle i,j,k \rangle}$ et A_{near} est si petite que nous pouvons utiliser la distance par saut sur A_{near} (notée dph_{near}) comme une approximation de $dph_{\langle i,j,k \rangle, t}$. En revanche, dans le deuxième cas, la distance entre $N_{\langle i,j,k \rangle}$ et A_{near} est si grande qu'on ne peut utiliser que dph_t comme l'approximation de $dph_{\langle i,j,k \rangle, t}$. Le troisième cas est entre les deux cas ci-dessus, donc, la valeur de $dph_{\langle i,j,k \rangle, t}$ peut être définie comme la moyenne des dph_{near} et dph_t . Ces trois (03) cas sont présentés dans la figure (3.6)



La procédure de l'algorithme *Selective 3-anchor DV-HOP* se résume comme suit. Les première et seconde étapes sont les mêmes que DV-HOP de base. Dans la troisième étape, N_x calcul d'abord ses « 3-ancre position estimée », puis N_x calcul la connectivité de chaque « 3-ancre position estimée ». Enfin, N_x choisit la meilleure « 3-ancre position estimée » qui a la connectivité la plus semblable avec lui.

13. Analyse de l'état de l'art

L'utilisation de la distance moyenne calculée par l'ancre le plus proche a été critiquée par tous ceux qui ont essayé d'améliorer DV-Hop, car elle affecte négativement la précision de localisation. Pour cette raison tous les efforts ont été concentrés sur la minimisation de l'erreur de calcul de la distance entre les ancres et les nœuds.

A cet effet plusieurs approches et diverses contributions avaient pour objectif d'améliorer l'algorithme DV HOP de base pour réduire l'erreur de localisation et donner plus de précision aux nœuds non encore localisés ainsi que d'apporter un meilleur temps de convergence pour l'ensemble du réseau WSN.

Parmi ces contributions, il y en a celles qui reposent sur la pondération de la taille de saut (CWDV-HOP, IDV -HOP) ou sur la correction d'erreur de distance (DEC-DV HOP), d'autres s'appuient sur le calcul de la distance par le RSSI (RF-DV HOP ou Hybride DV-HOP).

Quant à d'autres approches se basent sur les modèles statistiques, comme par exemple le calcul de l'estimateur de vraisemblance ou sur les méthodes heuristiques d'optimisation comme PSO et CSO.

Par ailleurs, plusieurs autres techniques sont connues et qui font appel à des concepts de recherche opérationnelles notamment la théorie de graphe comme l'algorithme (SD-DV HOP) ou à des concepts géométriques (diagramme de VORONOI) comme le DV-LOC, parfois même les lois de distribution sont invoquées, comme dans NDLR-DV HOP.

Le grand enjeu qu'affronte toutes ces approches c'est le coût de communication réseau c'est-à-dire la quantité de messages circulant à l'intérieur du réseau et qui a un impact direct sur la consommation d'énergie, la complexité temporelle ou spatiale de ces algorithmes sur ces équipements à ressources limitées, ainsi que le temps de convergence du réseau.

- Tableau 6 : Tableau comparatif des différentes variantes DV HOP :

Algorithme de localisation	Précision	Complexité de l'algorithme	Nbre de paquet communiqués dans le réseau	Coût énergétique	Temps de convergence	L'impact de l'augmentation du nombre de balise sur la réduction l'erreur de localisation	L'impact de l'augmentation du nombre des nœuds réguliers sur la réduction de l'erreur de localisation
DV HOP de base [13, 14]	Moyenne	Faible	Moyen	Faible	Long	Très important	Augmente l'erreur
DV HOP Distance Error Correction (DEC- DV HOP) [5]	Bonne	Élevé	moyen	moyen	Long	Important	Important
IDV HOP amélioré basé sur la pondération du HOP SIZE. [11]	Dépend du seuil M (Très bonne si M est bien choisi)	Dépend de la densité des nœuds	elevé	Moyenne	Très Long	Se stabilise une fois le nombre d'ancres dépasse un certain nombre.	Se stabilise une fois le nombre de nœuds dépassent un certain nombre.
SDDV HOP [14]	Moyenne	Faible	Faible	Faible	Long	important	Se stabilise une fois le nombre de nœuds dépassent un certain nombre.
RFDV HOP (RSSI FEEDBACK DV HOP). [12]	Plus élevée – sauf dans le cas d'un réseau irrégulier	Elevé	Elevé	Elevé	Rapide	Se stabilise une fois le nombre d'ancres dépasse un certain nombre.	Se stabilise une fois le nombre de nœuds dépassent un certain nombre.
Hybride DV-Hop Algorithme utilise RSSI [22]	Bonne	Faible	Elevé	Elevé	Très Long	Important	/

.../...

- Tableau 6 : Tableau comparatif des différentes variantes DV HOP : (Suite)

Algorithme de localisation	Précision	Complexité de l'algorithme	Nbre de paquet communiqués dans le réseau	Coût énergétique	Temps de convergence	L'impact de l'augmentation du nombre de balise sur la réduction l'erreur de localisation	L'impact de l'augmentation du nombre des nœuds réguliers sur la réduction de l'erreur de localisation
CWDV-Hop Optimiseur CSO [25]	Moyenne	Très élevé	Très élevé	Elevé	Très Long	/	/
Checkout DV_Hop [21]	Bonne	Faible	Moyen	Moyen	Long	Très important	Augmente l'erreur
SELECTIVE 3_ANCHOR DV-HOP [21]	Bonne	Très élevé	Très élevé	Elevé	Très Long	Important	/
RWDV-Hop [25]	Bonne	Elevé	Moyen	Moyen	Long	/	Améliore la précision
NDLR-DV-Hop [26]	Bonne	Elevé	Moyenne	Faible	Long	A L'extérieur de l'agneau, pas d'influence A l'intérieur de l'agneau l'influence est importante	A L'extérieur de l'agneau, l'erreur augmente A l'intérieur de l'agneau l'erreur se réduit

Tableau 6 : Tableau comparatif des différentes variantes DV HOP

14. Conclusion

Dans ce chapitre nous avons présenté un état de l'art sur des améliorations de l'algorithme de localisation DV-Hop. L'étude et l'analyse de ces améliorations nous ont permis de se pencher sur ce problème et de chercher une meilleure approche de localisation pour viser les objectifs suivants :

- *Obtenir un positionnement fiable des nœuds capteurs dans un réseau WSN.*
- *Gestion efficace d'énergie pour prolongement du temps de vie d'un réseau WSN.*

On peut remarquer que les techniques de pondération effectuées sur les distances des sauts moyens génèrent des erreurs de localisation surtout quand le nombre de sauts est grand ce qui rendra inutile de recourir à des calculs de distances pondérées.

Le calcul de la distance d'un nœud avec un ancre dans le cas d'un seul saut (nœud voisin) est plus exacte dans le cas d'un RSSI que celle qui repose sur la distance du saut moyen.

Par ailleurs, Plusieurs approches utilisent cette technique avec une multitude d'équations très compliquées voire même incompréhensibles pour une large communauté intéressée à trouver des méthodes précises est simples.

Recourir à des méthodes statistiques pour calculer la position des nœuds capteurs tels que par exemple l'estimation par le maximum de vraisemblance conduit souvent à des résultats aléatoires, cette méthode n'est toujours pas stable est parfois difficile à trouver la loi statistique correspondant.

Les méthodes basées sur la rétroaction pour calculer le facteur de correction nécessaire à la correction des positions estimées, comme c'est le cas de RFDV_HOP compliquent les localisations des nœuds et pourrait être une source de cumule d'erreurs. Ces méthodes ne permettent pas une économie d'énergie des capteurs ce qui pourrait diminuer le LIFE-TIME des réseaux WSN.

Afin d'améliorer le calcul de position des nœuds, notre approche s'est basée sur le calcul RSSI d'un seul saut voisin et exploite les nœuds récemment localisé et les considérés comme nœuds ancres pour le reste de calcul de localisation des nœuds non encore localisés.

Chapitre IV :

Implémentation et Simulation.

1. Introduction

Comme nous l'avons évoqué dans les chapitres précédents. Le réseau WSN est étroitement lié à la connaissance de la localisation de ses nœuds capteurs, car les données collectées par ce réseau doivent être liées à l'emplacement réel à partir duquel elles ont été collectées afin que les informations reçu de la zone cible puisse être analysée, et parce que ces capteurs sont généralement déployés de manière aléatoire dans des zones où est difficiles de calculer manuellement les coordonnées de chaque nœud, les remplacés ou encore recharger sa batterie. De plus, équiper chaque nœud d'un GPS coûte cher et consomme de l'énergie, ce qui réduit la durée de vie du réseau. Sachant que l'unité qui consomme le plus d'énergie est l'unité radio (la transmission d'un bit, consomme une énergie nécessaire pour exécuter environ 1000 instructions dans un capteur).

Par conséquent, nous devons nous appuyer sur des algorithmes qui calculent les coordonnées de localisation avec la plus grande précision et la plus petite marge d'erreur possible, tout en contrôlant le nombre de messages échangés et en éliminant ceux qui ne sont pas nécessaires pour réduire le flux de messages dans le réseau.

2. Objectifs de notre travail

L'objectif de notre travail consiste à proposer une amélioration de l'algorithme de localisation DV-Hop basé sur plusieurs critères pour localiser les nœuds inconnus dans un endroit précis à base d'un réseau de capteurs sans fil.

Après avoir examiné un certain nombre d'algorithmes améliorés du DV-Hop vu précédemment, l'algorithme amélioré que nous proposons est basé sur la distance des sauts pondérés en fonction des mesures RSSI entre les nœuds et exploite également les nœuds dont les coordonnées ont déjà été calculées afin qu'ils deviennent des nœuds d'ancrage supplémentaires qui seront utilisés pour localiser les nœuds restants. Cet algorithme amélioré sera abordé dans la section 07 de ce chapitre.

3. Outils d'implémentation et de simulation

Les capteurs disposent d'un processeur à basse fréquence de calcul d'une faible mémoire et d'une alimentation électrique très limitée. Par conséquent, il est devenu nécessaire de concevoir des programmes légers qui s'adaptent à ces capacités limitées en termes de coût énergétique et d'occupation de la mémoire. Pour résoudre ce problème, des systèmes logiciels ont été créés pour cibler spécifiquement les applications dans les réseaux WSN. Nous mentionnons Contiki, MANTIS et TinyOS.

Dans notre projet, nous allons travailler avec le système d'exploitation Contiki-OS pour implémenter et simuler notre algorithme de localisation. Ce système nous fournit la plupart des éléments de base nécessaires pour l'émulation d'un réseau de capteurs.

Nous le présentons ci-dessous avec le simulateur Cooja qui fonctionne sous contiki.

4. Le choix de Contiki

Contiki est système qui a été développé par une groupe de développeurs et par les contributions d'Atmel, Cisco, ETH, Redwire LLC, SAP, Thingsquare, et beaucoup d'autres, menés par Adam Dunkels de Thingsquare [31].

Il est considéré comme un système d'exploitation complet et réputé, il est un système open source, léger, flexible et générique qui s'appuie sur un modèle de fonctionnement hybride [27], Le noyau CONTIKI se base nativement sur le mode évènement. Néanmoins il supporte le mode multitâche. Avec Contiki OS, le développement d'application est facile et rapide, l'implémentation est en langage C.

Contiki a introduit l'idée d'utiliser les communications IP dans des réseaux de capteurs généralement à faible consommation et à ressources limitées. Les applications exécutées sur CONTIKI peuvent se connecter à n'importe quel appareil prenant en charge le protocole IP. L'accès IPV6 et UIPv6 sur CONTIKI offre une nouvelle interopérabilité avec les appareils prenant en charge ce protocole. La prise en charge de 6LoWPAN permet à CONTIKI de communiquer avec l'appareil via un

réseau sans fil basé sur la norme 802.15.4. CONTIKI est connu pour être un système d'exploitation puissant et mature, fournissant IPv4 et IPv6 pour les réseaux de capteurs sans fil.

D'autre part, Contiki offre un type de connectivité sous la couche RIME qui est un ensemble de protocoles en couches pour la communication en réseau. Rime est basé sur le principe du best effort et d'une transmission fiable avec qualité de service (QoS). Elle permet un dialogue avec les capteurs voisins pour établir des schémas de routage.

De ce qui précède, nous résumons les principales caractéristiques du système d'exploitation CONTIKI OS, à savoir :

Simple, dynamique et open source, et se basant sur les Protothread. Ce système modulaire est largement disponible sur le web et il a été conçu pour les équipements ayant des ressources limitées, tels que les capteurs.

5. Le simulateur Cooja

Un émulateur est un sous-ensemble des appareils et environnements virtuels qui permettent d'analyser le code des logiciels embarqués. Il simule de manière réaliste le comportement de ces logiciels. Ceci est particulièrement utile pour tester les programmes avant de les mettre en mémoire flash pour les capteurs réels.

Cooja est un acronyme pour "Contiki OS Java Simulator" qui est un simulateur/émulateur de réseau inclus avec Contiki. Il simule un réseau de capteurs. On peut tester rapidement du code écrit en C, avant de les flasher dans des vrais capteurs. De plus, nous pouvons déployer n'importe quel nombre de nœuds dans une zone donnée et visualiser la topologie du réseau en temps réel.

Pour les mesures basées sur RSSI, Cooja est limité car il ne reflète pas l'environnement réel pour les réseaux WSN, de plus il utilise un modèle linéaire pour simuler les valeurs de distance alors que le modèle est en réalité logarithmique.

Cooja peut visualiser et enregistrer les données provenant de la station de base dans des fichiers ou lues par des logiciels qui peuvent par la suite traiter et présenter

les données à l'utilisateur. Par exemple l'application collect-view intégrée dans Contiki qui permet de visualiser les valeurs des capteurs et des données de supervision du réseau.

L'interface utilisateur de l'application Cooja contient plusieurs fenêtres :

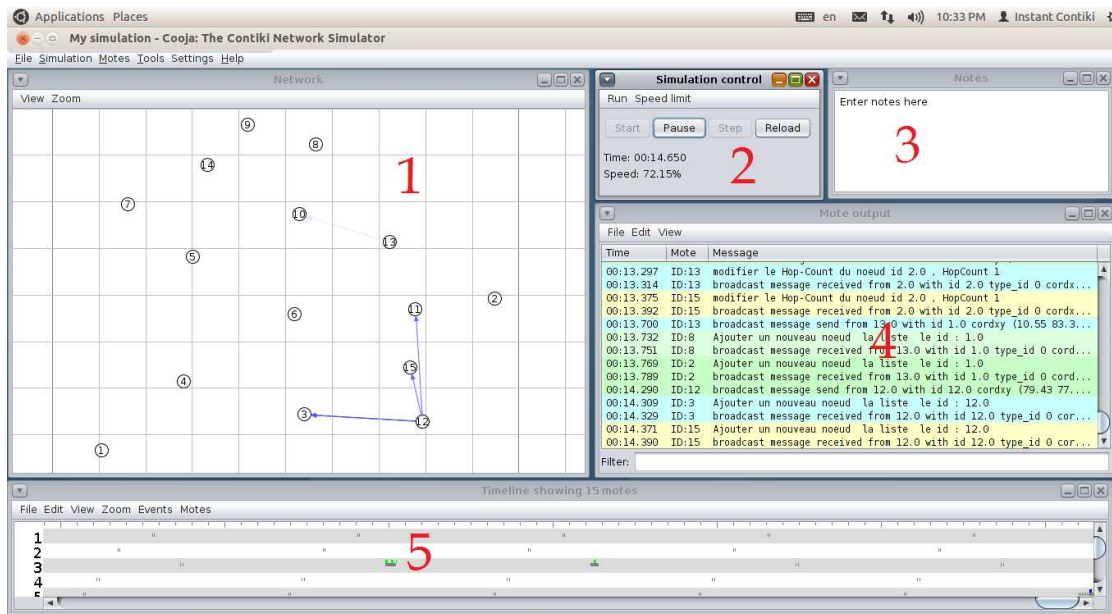


Fig. 4.1 – Interface graphique du le simulateur Cooja.

1. La fenêtre Network : Affiche la représentation graphique du réseau, nous montre tous les nœuds dans le réseau simulé, les positions, le radio trafic, les attributs des nœuds ...

2. Le panneau de simulation control : responsable du contrôle de la simulation où nous pouvons démarrer, mettre en pause et recharger la simulation. Aussi La vitesse de simulation est contrôlée via le menu Limite de vitesse.

3. La fenêtre Notes : En haut à droite est l'endroit où nous pouvons mettre des notes concernant notre simulation.

4. La fenêtre Mote Output : Elle est très important car c'est là que les sorties de journal pour tous les capteurs sont imprimées. Le champ de texte en bas nous permet de contrôler un filtre pour cibler un capteur ou un message en particulier.

5. La fenêtre Time line : Affiche tous les événements de la simulation dans le temps, très pratique pour comprendre ce qui se passe dans le réseau. La chronologie peut être utilisée pour inspecter les activités de nœuds individuels ainsi que les interactions entre les nœuds.

Lorsqu'on démarre la fenêtre de simulation on obtient l'affichage suivant :

- L'adresse Rime du noeud.
- L'adresse MAC du noeud ainsi que son identifiant.
- Le mécanisme par défaut Contiki_Mac, La couche MAC est responsable pour éviter les collisions au moyen de la radio et de retransmettre les paquets si il y avait une collision.
- Le nom du processus qu'a démarré.

6. Contraintes rencontrées dans cooja

Les problèmes avec les fonctions mathématiques se rapportant généralement aux nombres réels sont définis dans "math.h". C'est au programmeur de les programmer et de s'occuper des opérations et faire attention plus à la précision. Il fallait modifier dans le code source de Contiki pour apporter une solution avec transtypage.

Afin d'imprimer les nombres réels, ils doivent être convertis en nombres entiers puis gérée la réservation mémoire. Cette restriction n'est peut-être pas un défaut de contiki mais une exigence des capteurs afin de réduire le temps de calcul et l'occupation mémoire.

La récupérant les coordonnées des capteurs déployés dans une zone géographique représentée par une interface graphique n'est pas programmée surtout pour les nœuds ancrés et après avoir parcouru les forums dédiés au système Contiki, nous avons constaté qu'elles peuvent être obtenues à partir d'un fichier écrit en JAVA après avoir enregistré la simulation sur le disque.

7. Les principales étapes de notre Contribution sont :

Pour réaliser notre proposition nous avons déployé un ensemble de capteurs dans un environnement carré de 100m x 100m. Dans notre cas nous avons cinq pourcent de capteurs (ancres) dont leurs positions est connues auparavant (X_i, Y_i) parmi N nœuds inconnues qui cherchent à estimer leur positions à l'aide de ces ancres.

Notre travail se résume dans les étapes suivantes :

1ère Etape : calcul du taux d'erreur de distance:

Les nœuds ancres inondent le réseau avec des messages contenant les informations de localisation et d'identification $\{ID, (X_i, Y_i), RSSI_N, DR_N, DR_TOT, HOP-COUNT\}$.

ID : identification du nœud, (X_i, Y_i) : Coordonnées des nœuds, *HOP-COUNT* : nombre de sauts, initialisé à 0, *DR_N* : distance réelle calculée par *RSSI_N*, *DR_TOTAL* : total de la distance cumulé.

- Chaque nœud inconnu reçoit ces messages. Si un message reçu d'un ancre dont la valeur du champ HOP-COUNT est inférieure à celle du nœud alors le nœud mettre à jour sa table par la nouvelle valeur et incrémente son HOP-COUNT, calcule le distance avec le nœud d'où il provient ce message puis le diffuse à travers le réseau, sinon le message reçu est ignoré.

- Les nœuds ancres sauvegardent les informations ayant un HOP-COUNT minimum.

- Une fois la première phase de diffusion est terminée, les ancres procèdent aux calculs des taux d'erreurs de distance suite au fonction (1) ci-dessus qui seront diffusés dans de nouveaux messages. Le format du nouveau message est $\{ID, Type_Id, (X_i, Y_i), RSSI_N, DR_N, DR_TOT, HOP-COUNT, TauxE_i\}$.

$$\text{TauxE}_i = \frac{\sum_{i=1}^N \left[1 - (D_{ui} / Dr_Tot_i) / \text{Hop-Count}_i \right]}{N} \quad \dots(1)$$

D_{ui} : distance Euclidienne

N : Nbre total des ancrs sur la table.

2 ème étape : Calcule de la position des nœuds :

- Même scénario que la première étape, seule les messages reçus et ayant des HOP-COUNT minimum seront sauvegardés et/ou incrémentés et diffusés.
- Si un nœud inconnu reçoit trois messages ou plus provenant des nœuds ancrs dont parmi eux au moins un ancre voisin alors le nœud inconnu calcul ses coordonnées conformément aux règles suivantes :
- *Si la distance entre le nœud inconnu et l'ancre est d'un saut alors le nœud calcul directement la distance réelle via RSSI (DR_N).*
- *Si la distance entre le nœud inconnu et l'ancre est plus d'un saut alors la distance sera calculée en multipliant la distance total par le taux d'erreur de distance*

$$\text{Dist}_{UA} = Dr_Tot (1 - (\text{TauxE}_i * \text{HOP-COUNT}))$$

- Apres le calcul de la distance, le nœud inconnu estime sa position puis change son type (Type_Id =ancre) et il devient à son tour un ancre.
- Si un nœud inconnu reçoit des messages envoyés par des ancrs non voisins alors le nœud ne calcule pas sa position.

3 ème étape : Mise à jour des positions des nœuds :

Si un nœud reçoit un message d'un autre nœud qui vient d'être ancre et la distance qui leur sépare est d'un saut (HOP-COUNT=1) alors le nœud recalcule sa position et mettre à jour sa table, sinon le message est ignoré.

7-1. Organigramme de l'algorithme :

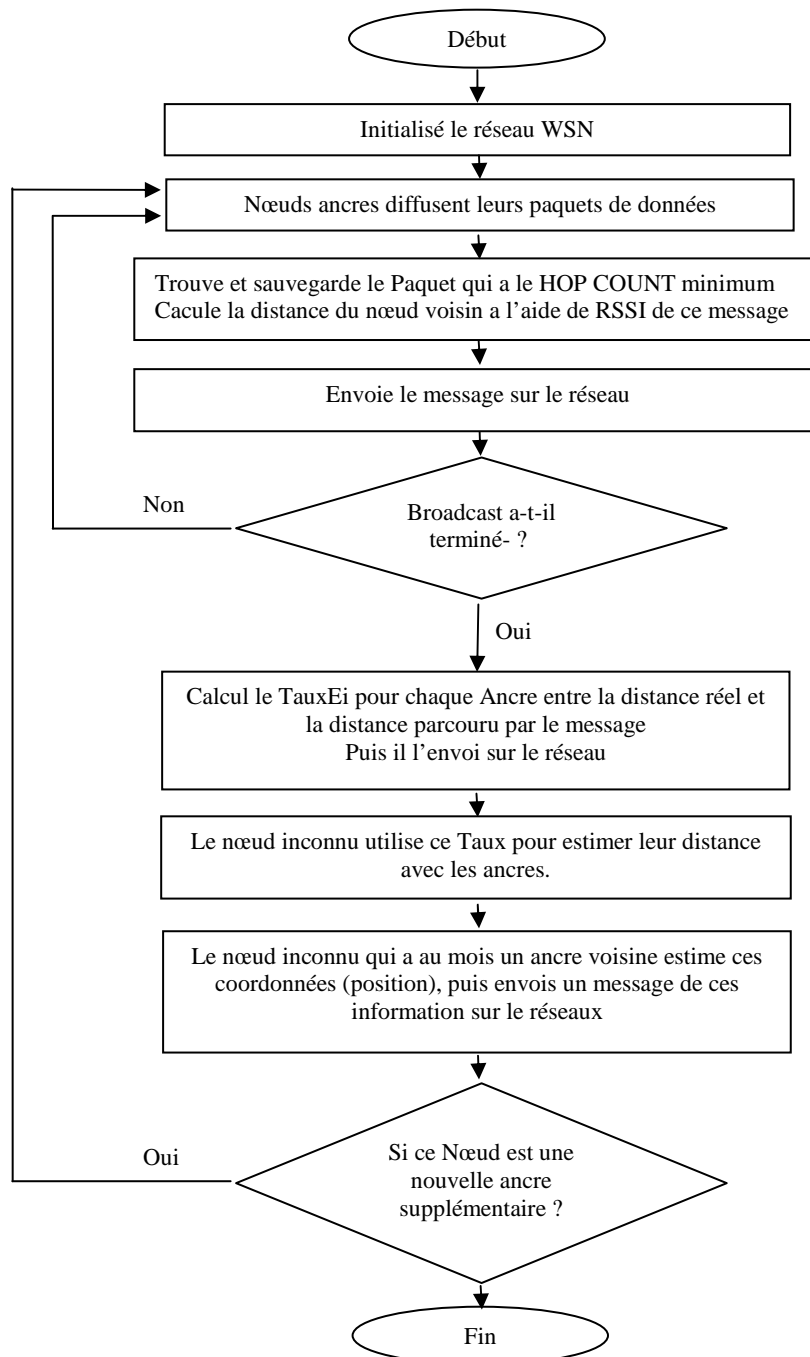


Figure 4.2 Organigramme de l'algorithme DE_DV-HOP.

7-2. L'algorithme DE_DV-Hop

- 1: **Initialization** de l'algorithme.
- 2: **for** i=1 to N ;
- 3: Les ancres préparent les messages de (hop-count=0, coordonnées)
- 4: Les Ancres diffusent leurs messages ;
- 5: **Fin For**
- 6: Un nœud inconnu reçoit le message et sélectionne le hop-Count minimum ;
- 7: **Si** Hop-Count = minimum **alors**
- 8 : Incrément le hop-count ;
- 9 : Calcule la distance du message à l'aide du RSSI
- 10 : Calcule le cumule de la distance parcouru par le message.
- 11 : Mise à jour puis renvoi le nouveau message sur le réseau.
- 12 : **Si non** ne rien faire
- 13 : **fin-si.**
- 14 : Chaque Ancre Calcule TauxEi ;
- 15 : Diffuse le tauxEi sur le réseau ;
- 16 : **Si** nœud inconnu a au moins une ancre voisin **Alors**
- 17 : Sélectionne les ancres et nœud les plus proche (hop-count , dr_tot) ;
- 18 : Nœud inconnu calcule sa position ;
- 19 : Ce nœud devient ancre supplémentaire ;
- 20 : Retour au début 3:
- 21 : Fin si ;
- 22 : Fin ;

8. Implémentation et Discussion :

8-1. Méthode de calcul de précision:

Afin d'évaluer la précision de positionnement de l'algorithme amélioré on peut calculer l'erreur moyenne produite par cette méthode. On doit calculer l'erreur moyenne comme étant la moyenne des différences entre les positions réelles et les positions estimées. L'erreur moyenne est calculée comme selon la formule suivante:

L'erreur moyenne de localisation

$$ALE = \frac{\sum_{i=1}^M \sqrt{(x'_i - x_i)^2 + (y'_i - y_i)^2}}{M}$$

Où (x,y) représente les positions réelles du nœud capteur, (x',y') les positions estimées du nœud, M représente le nombre des nœuds inconnus.

8-2. Paramètres d'évaluation des deux algorithmes :

Paramètres	Valeurs
Dimension de la topologie	100x100 m
Nombre de capteurs	40
Nombre d'ancres	5
Rayon de communication	30 m
Nombre d'itération	5

8-3. Erreur de localisation en fonction du nombre total de nœuds.

Cette simulation consiste à calculer la déviation de localisation en mètre pour un nombre de 10,20,30,40 et 50 nœuds tout en fixant le nombre d'ancres à 5. Les résultats sont illustrés par le tableau et le graphe ci-dessous :

Tableau 7 : Les erreurs de localisation vs nombre total de nœuds.

Nombre de nœuds	10	20	30	40	50
Erreur de localisation en mètre de (DV-HOP)	16,13	16,91	11,84	10,64	9,2
Erreur de localisation en mètre de (DE DV-HOP)	11,35	9,65	7,48	4,39	4,68

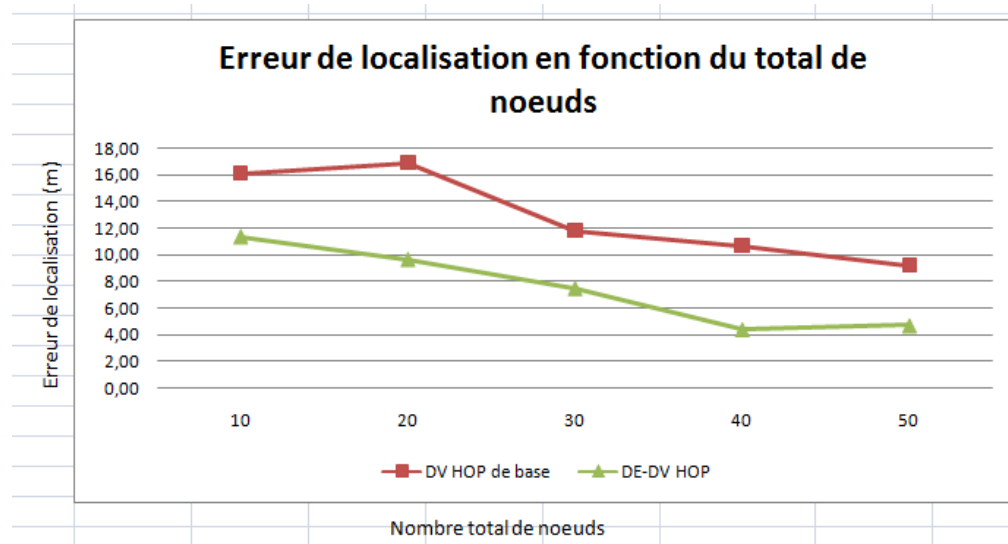


Figure 4.3 : Erreur de localisation vs nombre total de nœuds.

- **Analyse et explication:**

Quand le nombre de nœuds directement connectés est petit (entre 10 et 20 nœuds) alors le nombre de sauts du réseau augmente ce qui augmente également le taux d'erreur, le cas de DV HOP de base et dans une zone de surveillance 100x100m (petite superficie).

Cependant, les performances et la précision de l'algorithme proposé (DE-DV HOP) se détériorent si la distribution des nœuds est peu dense ou non uniforme.

Les écarts de positionnement des deux algorithmes diminuent au fur et à mesure de l'augmentation du nombre de nœuds (entre 20 et 50 nœuds).

Donc, comme il est illustré sur la figure 4.3, à mesure que le nombre de nœuds augmente, les erreurs montrent une tendance à la baisse c'est parce

qu'avec l'augmentation de la densité de nœuds, chaque nœud peut contenir plus d'un saut et par conséquent la connectivité de nœuds devient également plus forte.

L'algorithme proposé (DE DV-HOP) a une amélioration sur le taux d'erreur supérieur à celle de DV-HOP.

Dans les WSN peu dense, il existe une forte probabilité de nœuds mal connectés ou isolés et, dans de tels cas, la précision de la localisation peut être considérablement dégradée. Il est toujours avantageux d'augmenter la densité de nœuds (des informations de connectivité plus élevées signifient une probabilité plus faible de réseaux mal connectés) pour améliorer la précision de la localisation.

Ainsi, l'algorithme donne une meilleure précision pour un réseau dense.

8-4. Erreur de localisation en fonction du nombre d'ancres.

Cette simulation consiste à calculer la déviation de localisation de 40 nœuds avec respectivement 5,8,11 et 14 ancres.

Tableau 8 : Les erreurs de localisation vs nombre des nœuds ancres. :

Nombre d'ancres	5	8	11	14
Erreur de localisation en mètre de (DV-HOP)	10,64	8,80	7,49	6,20
Erreur de localisation en mètre de (DE DV-HOP)	4,39	4,03	3,94	3,23

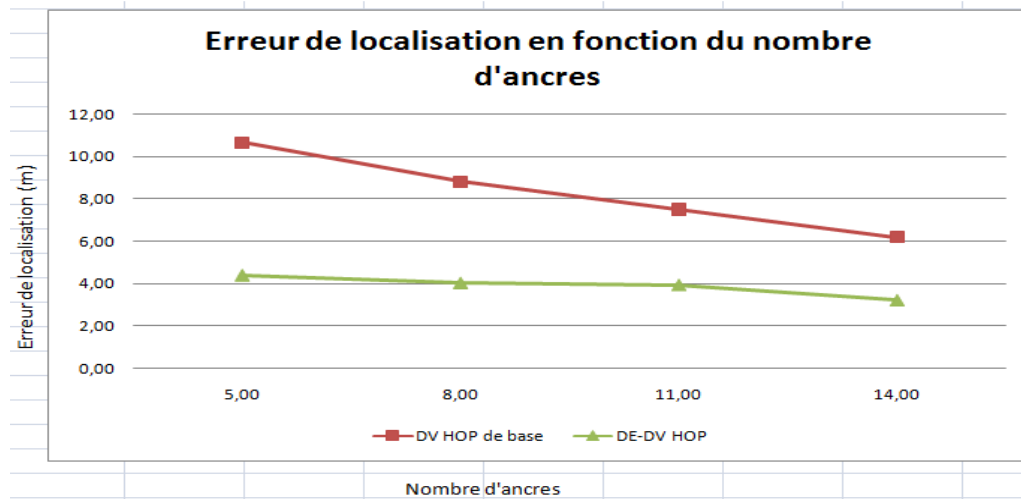


Figure 4.4 : Erreur de localisation vs nombre des nœuds ancrés.

- **Analyse et explication :**

Les positions des nœuds d’ancrages sont connues d’avance, le rayon de communication R est fixé à 30 mètres. L’augmentation du nombre des nœuds ancre en maintenant le nombre total des nœuds constant, la précision s’améliore.

Les écarts de positionnement des deux algorithmes diminuent au fur et à mesure que l’augmentation du rapport des nœuds d’ancrages. Mais l’algorithme proposé à une déviation plus faible.

Quand il ya moins de nœuds d’ancrages, le taux de déviation est plus élevé car les nœuds ne peuvent pas obtenir des informations de position suffisantes.

Lorsque le nombre de nœuds d’ancrages est 5 du nombre de nœuds, le taux de déviation est élevé car le manque d’ancre oblige les nœuds inconnus de passer plusieurs sauts pour atteindre l’ancre le plus proche. Donc, l’écart de distance augmente avec l’augmentation du nombre de saut.

La réduction de déviation est la conséquence de l’augmentation du nombre d’ancres. L’algorithme proposé a une amélioration sur le taux d’écart par rapport au DV HOP natif.

8-5. Détail des résultats :

Le tableau 9 ci-dessous montre les résultats extraits des simulations que nous avons menées sur le simulateur Cooja, où les capteurs ont été identifiés par des chiffres de 1 à 20, dont les cinq premiers sont des ancres. Après avoir exécuté l'algorithme que nous avons conçu, nous avons comparés les coordonnées calculées avec les données réelles que nous avons prises du simulateur. Ensuite nous avons calculé la différence de distance pour chaque capteur entre les coordonnées réelles et celles obtenues par notre algorithme. La moyenne d'erreur de localisation a été calculée est porté à la fin du tableau :

Tableau 9 : Liste de résultat de 20 nœuds capteur avec 5 ancres

Id des nœuds	coordonnées réels			Coord, Calculées par l'algorithme		Distance d'erreur
	Xi	Yi		Xi	Yi	
1	43,2	46,09		Ancres		
2	88,6	91,85				
3	10,54	95,51				
4	11,3	10,02				
5	82,31	4,73				
6	20,85	39,61		16,33	40,31	4,57
7	19,94	72,06		18,52	67	5,26
8	40,76	15,81		48,9	16,38	8,16
9	55,12	76,85		69,43	70,11	15,82
10	65,44	66,42		61,8	52,68	14,21
11	47,81	60,13		45,16	53,28	7,34
12	16,28	83,25		14,62	74,83	8,58
13	54,86	28,65		61,15	33,7	8,07
14	64,95	36,18		71,92	42	9,08
15	23,82	54,46		24,44	52,47	2,08
16	68,66	22,88		76,07	29,3	9,80
17	19,25	25,9		18,52	24,91	1,23
18	72,49	46,28		77,96	41,18	7,48
19	52,61	88,74		86,49	84,87	34,10
20	63,42	14,05		68,69	21,37	9,02
Total des distances d'erreur						144,81
La Moyenne						9,65

9. Evaluation énergétique de l'algorithme proposé :

9-1. Paramètres de simulations :

Paramètre	Valeur
Modèle radio	UDGM
Plateforme équipement	Skymote
Nombre de nœuds inconnus	11
Nombre de balises	04
Zone de transmission	30 m
Zone d'interférence	30 m
Ratio de transmission	30 %
Ratio de réception	30 %
Déploiement	Aléatoire
Temps de simulation	20 min
Dimension réseau	100m x100m

9-2. Topologie du réseau :

La figure ci-dessous illustre la topologie réseau de l'évaluation énergétique.



Figure 4.5 : Réseau de capteurs de l'évaluation énergétique.

Remarque importante : Pendant toute la simulation, nous considérons que les capteurs sont en mode radio ON.

Supports utilisés pour collecter les informations de consommation en énergie :

9-3. Application POWERTRACE.

- **Caractéristiques capteur SKYMOTE**

CPU : Nombre de TICKS d'horloge du processeur en état actif, sans les modules radios.

LPM : Nombre de TICKS d'horloge du CPU en état de veille.

TX : Nombre de TICKS radio en état de transmission.

RX : Nombre de TICKS radio en état de réception.

RTIMER_SECOND : nombre de TICKS par seconde. (1s= 32768 TICKS).

TICKS: Temps d'horloge.

Runtime : Temps d'exécution du POWERTRACE.

- **Méthodologie de travail :**

- ✓ Activation de l'application POWERTRACE (l'inséré dans le code source des nœuds ancres et nœuds inconnus).
- ✓ Calcul la moyenne de consommation énergétique (POWER CONSUMPTION) du nœud simple et des ancres.
- ✓ Calcul le temps de latence, c'est le temps qui sépare l'émission d'un paquet de données de l'arrivée à sa destination mesuré en seconde ou un de ses ordres de grandeurs avec l'analyseur WIRESHARK.

Remarque:

La consommation d'énergie d'un nœud capteur (P) est la somme de toutes les consommations suivantes :

$$E = CPU_P + LPM_P + TX_P + RX_P$$

Exemple : Les caractéristiques du capteur SKYMOTE figure sur le tableau10.

Tableau 10 : Caractéristiques du capteur SKYMOTE.

Voltage	Ampérage				RTIMER- SECOND	Temps d'exécution (Runtime POWER TRACE)
	CPU	LPM	TX	RX		
3 V	330 uA	1.1 uA	17.4mA	18.8 mA	32768 Tick /s	5 secondes

Tableau 11 : Echantillon des données sorties capteurs.

Temps de simulation	N° ID	Taille des paquets transmis	N° de séquence	ALL_CPU	ALL_LPM	ALL_TX	ALL_RX
19:55.391	ID:8	152964,00	238,00	1831659,00	37324602,00	94331,00	286020,00
20:00.391	ID:8	153604,00	239,00	1837780,00	37482316,00	94331,00	287010,00

Les informations du tableau 11 représentent les sorties capteur N° 08 de notre simulation notamment les TICKS d'horloge des différents états du capteur.

On calcule la différence du nombre de TICKS entre deux intervalles de temps.

$$CPU_P = (1837780,00 - 1831659,00) = \mathbf{6121 \text{ TICKS}}$$

$$LPM_P = (37482316,00 - 37324602,00) = \mathbf{157 \ 714 \ \text{TICKS}}$$

$$TX_P = (94331,00 - 94331,00) = \mathbf{0 \ \text{TICKS}}$$

$$RX_P = (287010,00 - 286020,00) = \mathbf{990 \ \text{TICKS}}$$

$$PC = ((CPU_P * C_{cpu} + LPM_P * C_{lpm} + TX_P * C_{tx} + RX_P * C_{rx}) * V) / (RTIMER_SECOND * Runtime) \quad \dots (01)$$

PC : Power consumption (énergie consommée)

C_{xx} : Ampérage capteur en Milli-Amper (mA).

V : Voltage en volt.

RTIMER_SECOND : nombre de TICKS par seconde.

Runtime : Temps d'exécution POWERTRACE.

En appliquant la formule (01) nous obtenons le résultat suivant :

$$PC = \left(((6121 * 0,33) + (157714 * 0,0011) + (0 * 17,4) + (990 * 18,8)) * 3 \right) / (32768 * 5) = 0,3809585339355 \text{ mW}$$

9-4. Evaluation énergétique de l’algorithme proposé

Ci-dessous la figure (4.6) qui concerne l’affichage des informations utiles pour le calcul d’énergie et des informations de position.

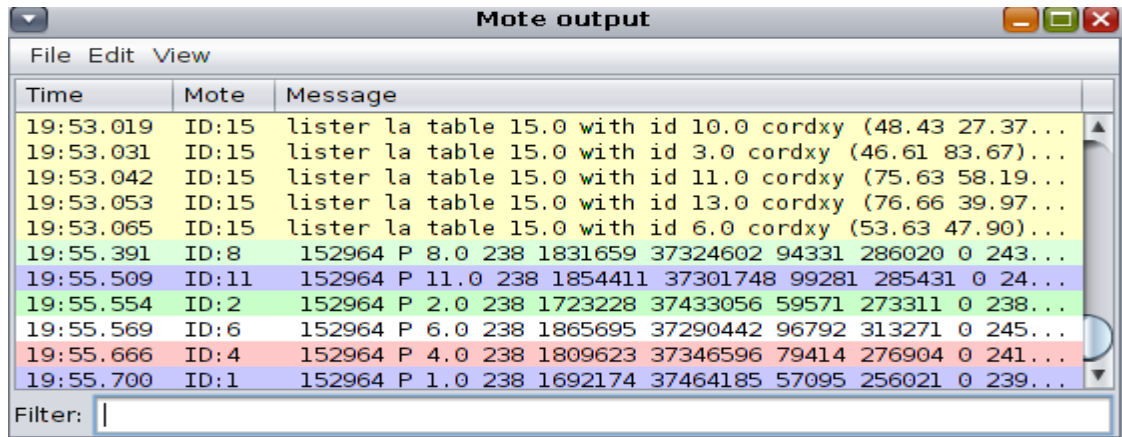


Figure 4.6 : Echantillon sorties capteurs sur Cooja de l’algorithme proposé.

- **Calcul de la consommation moyenne des quatre ancrés :**

Tableau 12 : Total des TICK (s) CPU/RADIO des ancrés.

Temps de simulation	N° capteur	ALL_CPU	ALL_LPM	ALL_TX	ALL_RX
00:05.697	ID:1	11009	152837	2487	1015
20:00.700	ID:1	1698293	37621901	57095	257011
00:05.551	ID:2	11830	152016	2490	1187
20:00.554	ID:2	1729347	37590772	59571	274301
00:05.716	ID:7	11003	152843	2483	1014
20:00.720	ID:7	1746391	37573718	57115	281705
00:05.909	ID:12	11101	152735	2479	1148
20:00.913	ID:12	1725736	37594358	56984	275828

L’énergie consommée par les ancrés est calculée sur la base des TICKS selon la formule (01). Les résultats obtenus sont reportés sur le tableau 13 ci-dessous :

Tableau 13 : Consommation d'énergie des ancrés

N° ancrés	Id :1	Id :2	Id :7	Id :12
PC_CPU	0,042481	0,0432419	0,043691867	0,0431694
PC_LPM	0,003145	0,003142	0,003140485	0,0031422
PC_TX	0,072493	0,0757759	0,07252478	0,0723562
PC_RX	0,367182	0,3917346	0,402602448	0,3939807
Total Energie en mW	0,4853	0,5138943	0,52195958	0,5126485

Puis les résultats sont présentés sur un Histogramme voir figure 4.7 :

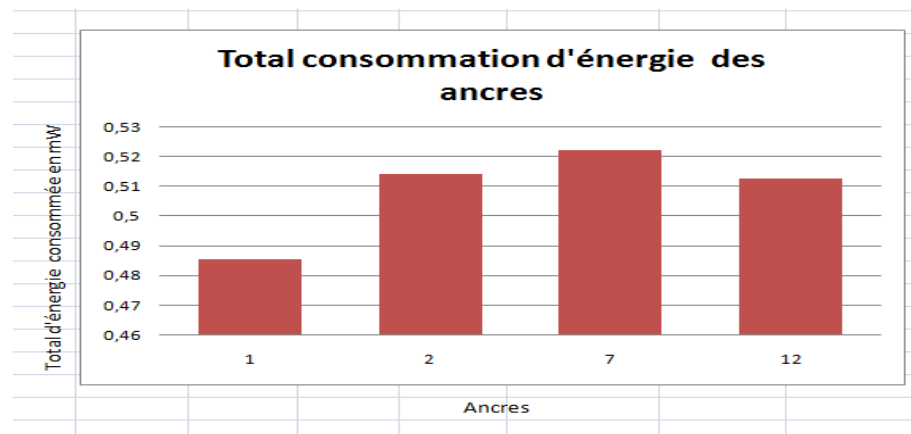


Figure (4.7) Histogramme d'énergie consommée des ancrés.

Observation : D'après l'histogramme de la figure (4.7), l'ancre N° 07 a eu une légère augmentation de consommation d'énergie en le comparant avec ses homologues (1-2-12). Cette augmentation est à l'ordre de 0,00807 par rapport à son suivant l'ancre N° 2.

Ci-après les détails d'énergie des l'ancres :

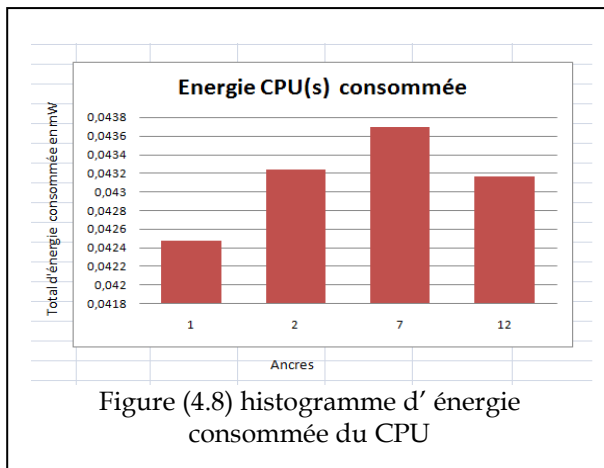


Figure (4.8) histogramme d'énergie consommée du CPU

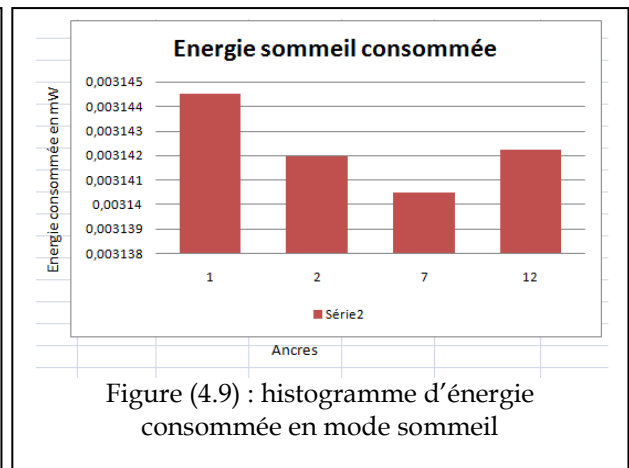


Figure (4.9) : histogramme d'énergie consommée en mode sommeil

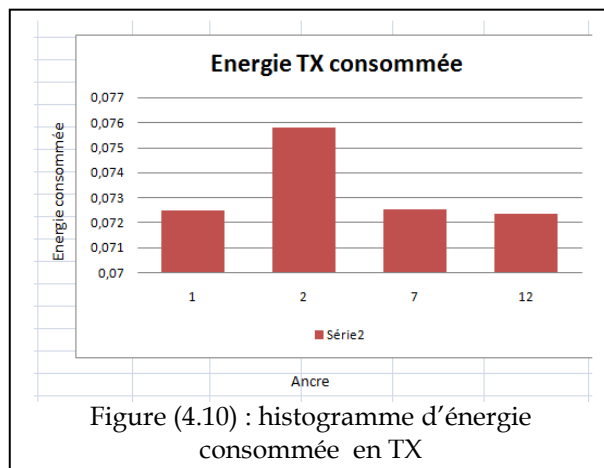


Figure (4.10) : histogramme d'énergie consommée en TX

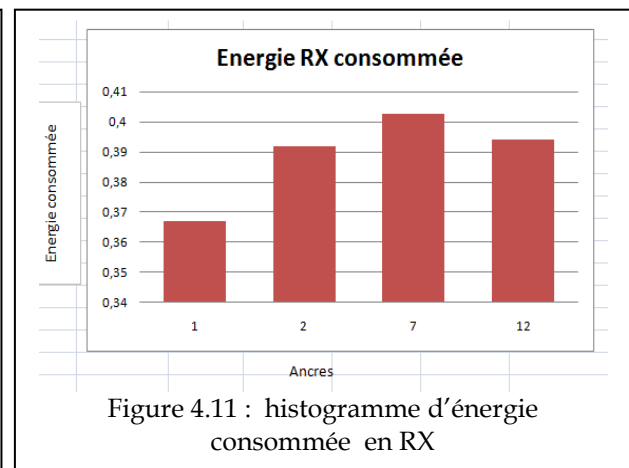


Figure 4.11 : histogramme d'énergie consommée en RX

- **Analyse et explication :**

Cette légère augmentation en énergie consommé qu'a eu l'ancre 07 pendant 20 minutes de simulation bien qu'elle est très minime environ 0,008 mW, néanmoins elle pourrait être justifiée :

1- C'est l'énergie qu'a consommé le module radio soit en réception de données ou simplement en écoute active (IDLE LISTENING) et elle est supérieure aux énergies des autres ancres.

Cela peut être expliqué par :

- **L'inactivité** du protocole CONTIKI-MAC qui fonctionne au niveau de la couche RDC située entre la couche physique et la couche liaison de données et qui est conçu pour faire fonctionner le cycle radio afin

d'optimiser la consommation d'énergie du module radio dans le cas où aucune réception n'est détectée sur le canal.

- L'ancre N° 07 et l'ancre N° 02 figure (4.12) ayant une connectivité réseau avec 03 nœuds au voisinage, donc leurs énergies d'écoute sont plus élevées.

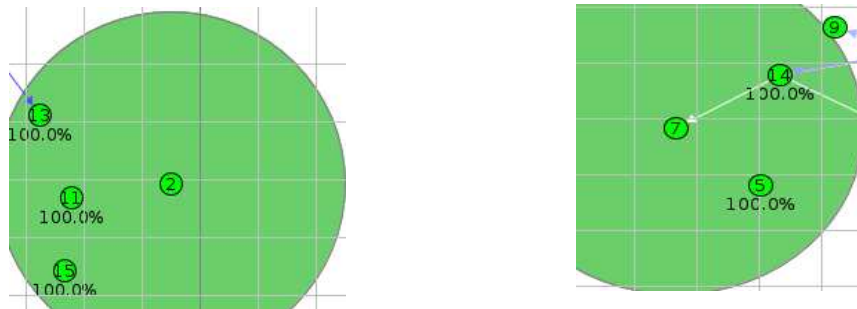


Figure 4.12 : Connectivité réseau des anres N° 02 et N° 07.

- L'ancre N° 01: figure 4.13 a la plus petite consommation d'énergie en réception parmi les anres car sa connectivité est limitée uniquement à un (01) nœud voisin ce qui fait que son énergie d'écoute est la plus faible par rapport aux autres anres.

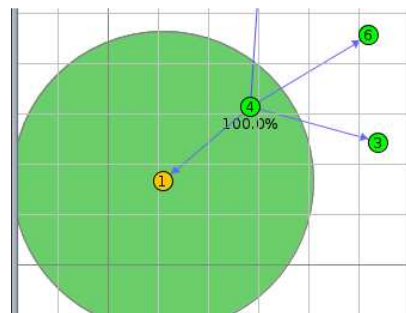


Figure 4.13 : Connectivité réseau de l'ancre N° 01.

- L'absence dans ce réseau simulé du mécanisme FAST SLEEP qui détecte les faux WAKE UP généré par les interférences et ayant un effet négatif sur la consommation d'énergie.

2- L'énergie consommée dans le calcul au niveau du CPU de l'ancre N° 7 est supérieure de l'ordre de 0,000449938 mW aux autres anres voir la figure 4.8.

Remarque : Pour éviter la perte d'énergie, le module radio d'un capteur doit s'activer en cas de nécessité. Le reste du temps, il doit rester en état de sommeil (alternance entre mode active et mode veille).

- **Consommation moyenne d'énergie des nœuds simples :**

Tableau 14 : Energie consommée des nœuds simples.

N° ID du Nœud	3	4	5	6	8
Energie consommée en mW	0,59873319	0,55119927	0,59457338	0,62749058	0,58450365
N° ID du nœud	9	10	11	13	14
Energie consommée en mW	0,56832134	0,61346923	0,5906028	0,61553113	0,57420594
N° ID du nœud	15				
Energie consommée en mW	0,57598521				

Tableau 15 : Détail des énergies consommées par nœud.

N° nœud	CPU	LPM	TX	RX	Total énergie
ID:3	0,046837633	0,003129995	0,131601837	0,408287903	0,598733191
ID:4	0,045562921	0,003134243	0,105423248	0,388630371	0,55119927
ID:5	0,046581985	0,003130841	0,121824692	0,414035095	0,594573377
ID:6	0,046951332	0,003129609	0,128492798	0,439365417	0,627490579
ID:8	0,046118955	0,003132394	0,125225784	0,401302551	0,584503653
ID:9	0,045559346	0,003134264	0,118564316	0,392530151	0,568321341
ID:10	0,04678456	0,0031301	0,121908325	0,432249512	0,613469226
ID:11	0,046667109	0,003130559	0,131796982	0,400305847	0,590602798
ID:13	0,047068758	0,003129216	0,13181955	0,424289917	0,615531134
ID:14	0,045865096	0,003133232	0,118363861	0,3981875	0,574205939
ID:15	0,046584881	0,003130781	0,128340134	0,38946283	0,575985208

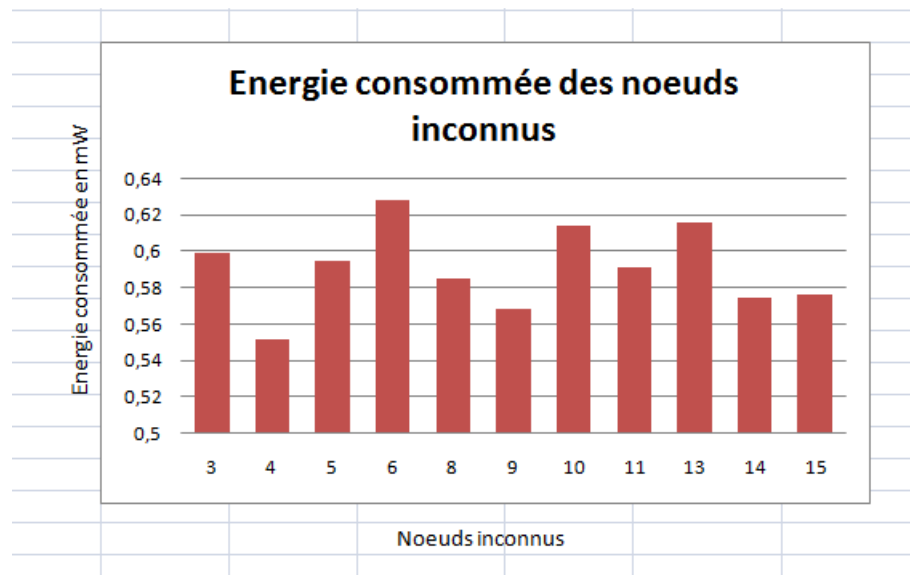


Figure 4.14 : Histogramme d'énergie consommée des nœuds simples.

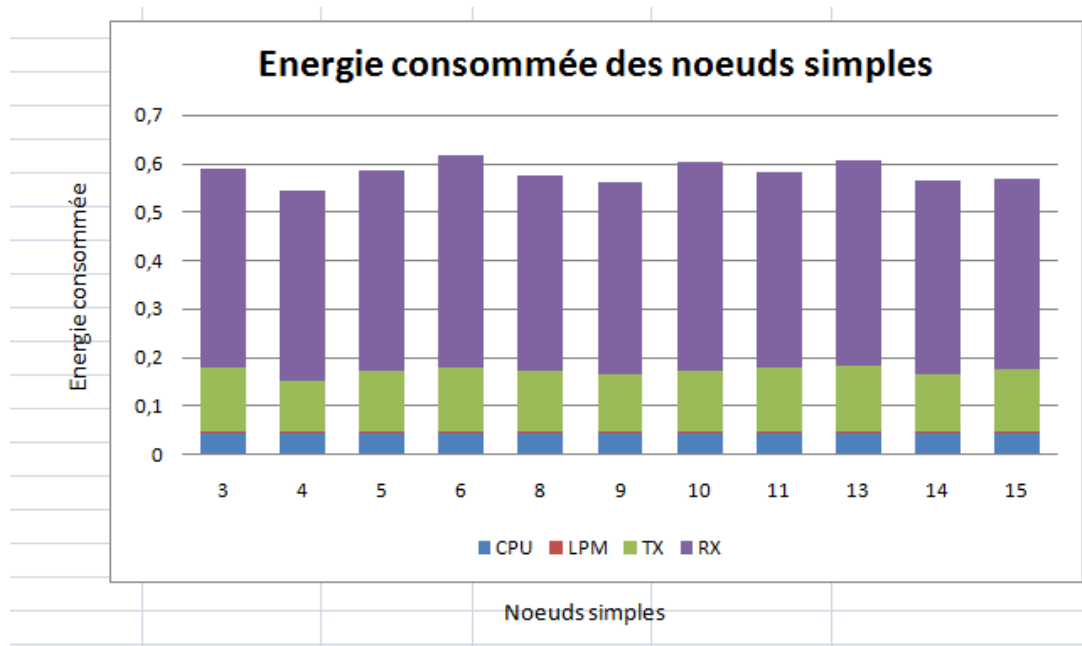


Figure 4.15 : Histogramme d'énergie consommée des nœuds simples détaillé.

Observation : L'histogramme de la figure 4.14 indique que le nœud N° 6 a consommé un peu plus d'énergie par rapport aux autres nœuds.

- **Analyse et explication :**

Le nœud N° 06 a consommé un surplus d'énergie en réception à cause de:

1- Sa position, car ce nœud est au centre du réseau et il sert aussi comme un nœud relai pour le reste du réseau.

2- Il est tout le temps en écoute active (IDLE LISTENING) avec l'absence du mode sommeil durant les périodes d'inactivité du module radio. Comme il a été dit précédemment la fonctionnalité du cycle radio n'a pas été activée au cours de cette simulation et qui a effectivement augmenté la consommation d'énergie sur tout le réseau.

3- **L'absence** du mécanisme FAST SLEEP qui détecte les faux WAKE UP généré par les les interférences mutuelles des capteurs de proximités.

9-5. Calcul de la latence moyenne avec WIRESHARK :

Latence (délai) : c'est le temps qui sépare l'émission d'un paquet de données de l'arrivé à sa destination ; il est mesuré en seconde ou un de ses ordres de grandeurs.

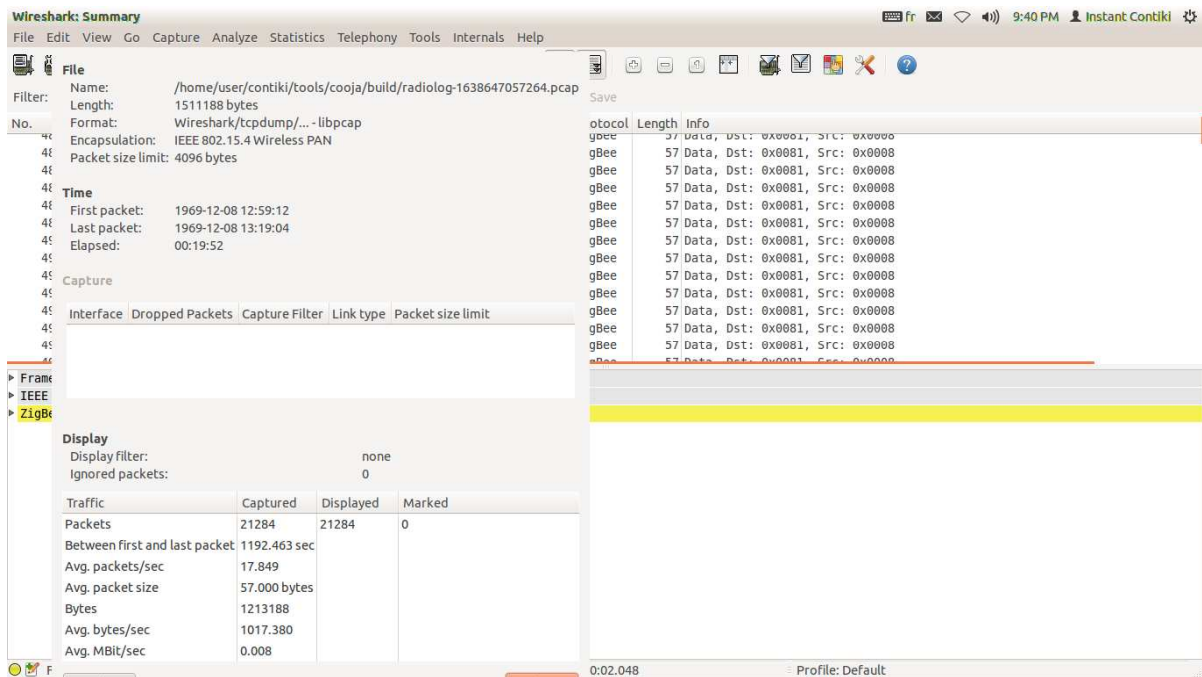


Figure 4.16 : des consoles WIRESHARK.

Tableau 16 : Récapitulatif de la latence :

Total noeuds	Temps de simulation	Nombre de paquets diffusés	Taille paquet (Byte)	Moyenne Paquets/s	Latence Moyenne (seconde)
15	20 min	21284	57.00	17.849	0.056 s

Explication : Pendant 20 minutes de simulation figure 4.16 on a pu constater un important sur coût de communication environ 21284 paquets diffusés dans le réseau simulé cela aura une influence directe sur la consommation d'énergie des nœuds. La taille moyenne d'un paquet est de 57 Octets, la latence moyenne de réseau est définie comme la durée de temps prise par un paquet entre le nœud expéditeur et le nœud récepteur, dans notre cas la latence est environ à 56 ms.

10. CONCLUSION

Dans ce chapitre Nous avons proposé notre approche, version améliorée de l'algorithme natif DV-Hop, nommée DE DV HOP qui se base sur les mesures RSSI pour la localisation des nœuds inconnus dans le voisinage immédiat des balises. Étant donné que le nœud capteur offre la puissance de signal RSSI dans les paquets de données reçus alors il n'entraîne aucun coût supplémentaire en termes de nouveaux composants matériels. De plus, cet algorithme localise progressivement les nœuds inconnus et les rendre par la suite comme des références c'est-à-dire des ancres pour compléter le processus de localisation. Nous avons étudié et comparé les performances de l'algorithme avec celui de base DV HOP natif, l'évaluation des performances ont clairement indiqué que cette version améliorée surpasse son homologue. Plus précisément, en précision et a déviation de localisation.

Conclusion générale et perspectives :

Les réseaux WSN ; au cours de ces dernières années, ont été l'objet de plusieurs recherches aussi bien dans l'industrie que dans les milieux universitaires. Cela est du à l'ampleur sans précédent des possibilités qu'offre cette technologie. Toutefois, les réseaux de capteurs sans fil doivent aussi faire face à d'importants défis de localisation des nœuds aléatoirement répartis dans des endroits hostiles ou inaccessible.

Dans ce projet, on a essayé de présenter une approche hybride de localisation des nœuds capteurs dans un WSN basé sur les deux mécanismes de calcul de position le RSSI et le DV-HOP pour donner plus de précision à nos calcul de positions.

Tout au long de ce travail qui a commencé par une étude bibliographie, on a découvert un champ de recherche très intéressant qui touche presque tous les domaines d'application. Cela nous a permis de nous initier à la recherche, et nous espérons avoir apporté une contribution, aussi petite qu'elle soit, à ce domaine qui en pleine évolution.

Comme perspective, nous nous fixons :

A améliorer cet algorithme en corrigeant les problèmes survenus au moment de la simulation et qui concernent l'écart de positionnement des nœuds inconnus surtout ceux situés au bord (frontière) de la zone de surveillance.

Références et Bibliographie :

- [1]. CAYIRCI, E. (2004). "Wireless sensor networks". In : D. Katsaros et al. (éd), Wireless information highways (pp. 273-301). Hershey : Idea group inc.
- [2] Ahmed Habbani, Olivier Romain, Jamal El Abbadi et Patrick Garda article " système d'acquisition et de routage d'informations" e-TI, Revue électronique des Technologies de l'Information. <http://www.revue-eti.net>, Numéro 2. 2006. ISSN 1114-8802.
- [3]. S. Tilak, N. Abu-Ghazaleh, and W. Heinzelman, "A taxonomy of Wireless Micro-sensor Network Models," ACM SIGMOBILE, Mobile Computing and Communications Review, vol.6, issue: 2, pp. 28-36, April, 2002. [Available from the World Wide Web (WWW).
<http://www.cs.binghamton.edu/nael/research/papers/taxonomy.pdf>.
- [4]. Adel CHOUHA « Traitement et transfert d'images par réseau de capteur sans fil » mémoire Magister en Informatique - Ingénierie des systèmes d'informatiques année 2011.
- [5] Mlle REZKI YASMINA & BOURDJAH SIHAM « Etude des mécanismes de localisation dans les réseaux de capteurs sans fil » mémoire de Master recherche en Informatique - Réseaux et systèmes distribués Université A/Mira de Béjaia - 2014
- [6] Jiuqiang Xu, Wei Liu, Fenggao Lang, Yuanyuan Zhang, Chenglong Wang Distance Measurement Model Based on RSSI in WSN School of Information Science & Engineering Northeastern University, Shenyang, China E-mail: xujiuqiang@ise.neu.edu.cn Received May 18, 2010; revised June 9, 2010; accepted June 18, 2010.
- [7] Mr Messaoud BELLOULA "La géolocalisation dans les réseaux de capteurs sans fils » mémoire Magister - Ingénierie des systèmes informatiques - Université Hadj Lakhdar - Batna.
- [8] Yu Hu, Xuemei Li « An improvement of DV-HOP localization algorithm for wireless sensor networks » Published online : 28 June 2013.
- [9] Fei Liu and Guang-zeng Feng Research on improved DV-HOP Localization Algorithm Based on RSSI and Feedback Mechanism - college of Telecommunication and information Engineering, Nanjing of Post and Telecommunications, Nanjing 210003, China.
- [10] CHETOUAH Karima & BADDUOU Saliha « Application de l'essaim de poulet pour la localisation dans les RCSFs » mémoire Master - Système intelligent - Année 2019.

- [11] Yu Hu · Xuemei Li An improvement of DV-Hop localization algorithm for wireless sensor networks Published online: 28 June 2013 © The Author(s) 2013. This article is published with open access at Springerlink.com.
- [12] DeepakPrashar, kiranjyoti, Distance Error Correction Based Hop Localization Algorithm for Wireless Springer Science + Business Media, LLC, part of Springer Nature March 2019.
- [13] Xueguang Wang, Jieke Lin , Research and Improvement of DV-Hop Localization Algorithm For Wireless Sensor NetworkGuiqi Liu, ZhihongQian, XueWang An improved DV-Hop localization algorithm based on hop distances correction for the publisher version.06-12-2010
- [14] Yan H, Zhilong Shan, and Hua Yu “ Research on improved DV HOP Localization Algorithm Base don the Ratio of Distances » School of Computer, South China Normal University. Guangzhou 510631, China sunnysz1@163.com. School of Electronic and information Engineering south China University of Technologie Guangzhou 510640, China Yuhua@scut.edu.cn- 2015
- [15] Fei Liu, and Guang-zeng Feng, Research on Improved DV-Hop Localization Algorithm Based on RSSI and Feedback Mechanism 1 College of Telecommunications and Information Engineering, Nanjing University of Post and Telecommunications, Nanjing 210003, China 2 College of Mechanical Engineering,Guizhou University, Guiyang 550025, China 981890733@qq.com - 2015
- [16] G. Zhou, T. He, S. Krishnamurthy, and J. A. Stankovic, “Models and solutions for radio irregularity in wireless sensor networks,” ACM Trans. Sensor Networks, vol. 2, no. 2, pp. 221-262, Mai 2006.
- [17] http://en.wikipedia.org/wiki/Dijkstra's_algorithm
- [18]. Niculescu D., & Nath B. (2003). DV-Hop based positioning in ad-hoc networks. *Telecommunication Systems*, 22(1-4), 267-280.
- [19] Singh, S. P., & Sharma, S. C. (2018). A PSO based improved localization algorithm for wireless sensor network. *Wireless Personal Communications*, 98(1), 487-503.
- [20] Chen X. & Zhang B. (2012). Improved DV-Hop node localization algorithm in wireless sensor networks. *International Journal of Distributed Sensor Networks*, 8(8), 213980.
- [21] LIQING GUI thèse doctorat université de TOULOUSE “ Improvement of range-free localization systems in Wireless Sensor Networks “ école doctorale (EDSYS) Directeur de thèse Thierry VAL et Anne WEI. Année 2013.
- [22] O. Cheikhrouhou , GM. Bhatti and R. Alroobaea “A Hybrid DV-Hop Algorithm Using RSSI for Localization in Large-Scale WSN ” *Sensors*, Mai 2018 - mdpi.com.

- [23] Xie H.; Li W.; Li S.; Xu B. An improved DV-Hop localization algorithm based on RSSI auxiliary ranging. In Proceedings of the 2016 35th Chinese Control Conference (CCC), Chengdu, China, 27–29 July 2016; pp. 8319–8324.
- [24] Gui L.; Val T.; Wei A.; Dalce R. Improvement of range-free localization technology by a novel DV-Hop protocol in wireless sensor networks. *Ad Hoc Netw.* 2015, 24, 55–73. [CrossRef]
- [25] JIAXING CHEN, WEI ZHANG, ZHIHUA LIU, RIU WANG AND SHUJING ZHANG “A Hybrid Localization Algorithm With Distance-Weight DV-Hop and CSO for WSN” college of Engineering-Electronics and Information – computer and cyber security – China. Publication December 17 2020.
- [26] Xin FENG, Zhuang LIU, Xin DING, Su Peng SUN « An improved Localization Algorithm in WSN based on Nodes Distribution Law and Revised HOP-Size of DV-HOP » Changchun University of Science and Technology, China, November 2013.
- [27] H. Xiao, H. Zhang, Z. Wang, and T. Aaron Gulliver “An RSSI Based DV-Hop Algorithm for WSN” IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (PACRIM), août 2017.
- [28] J. Mass-Sanchez, E. Ruiz-Ibarra, J. Cortez-González, A. Espinoza-Ruiz, and L. A. Castro, “Weighted hyperbolic DV-Hop positioning node localization algorithm in WSNs,” *Wireless Pers. Commun.*, 2017.
- [29] P. Tarrio, A. M. Bernardos, J. A. Besada, and J. P. Cesar, “A new positioning technique for RSS-based localization based on a weighted least squares estimator,” in *Proc. IEEE Int. Symp. on Wireless Commun. Systems*, Reykjavik, Iceland, Oct. 2008, pp. 633-637.
- [30] S. Kunar and D. K. Lobiyal, “Novel DV-Hop localization algorithm for wireless sensor networks,” *Telecomm Systems*, vol. 64, no. 3, pp.509-524, Mars 2017.
- [31] S. Abdessamad, M. Wafa “ Développement d’une application orientée surveillance pour les réseaux de capteurs sous Contiki ” Mémoire de fin d’études pour l’obtention du diplôme de Licence en Informatique, Université Abou Bakr Belkaid- Tlemcen Mai 2015