

Université du Québec
Institut national de la recherche scientifique
Centre Énergie Matériaux Télécommunications

**PROTOTYPAGE RAPIDE ET ÉVALUATION EN CONDITIONS
D'OPÉRATION RÉELLES D'ALGORITHMES DE LOCALISATION POUR
LES RÉSEAUX DE CAPTEURS SANS FIL**

Par
Mohamed Guadane

Mémoire présenté pour l'obtention du grade de
Maître es Sciences, M.Sc.
en Télécommunications

Jury d'évaluation

Examineur externe	Prof. Mohamed Ibnkahla Carleton University
Examineur interne	Prof. Tarek Djerfi INRS-EMT
Directeur de recherche	Prof. Sofiène Affes INRS-EMT
Co-directeur de recherche	Dr. Abdelaziz Samet INRS-EMT

À l'âme de ma mère.

Remerciements

Ce travail a été élaboré au sein de l'équipe Wireless Lab du Centre Énergie, Matériaux et Télécommunications (ÉMT) de l'Institut National de la Recherche Scientifique (INRS).

C'est avec un grand plaisir que je réserve cette page en signe de gratitude et de profonde reconnaissance à tous ceux qui m'ont aidé à la réalisation de ce modeste travail.

J'adresse l'expression de ma très grande reconnaissance aux membres du jury avec mon respect et ma gratitude pour accepter d'évaluer mon travail.

Je tiens à exprimer ma gratitude et mes respects les plus sincères à mon directeur de recherche, le **Professeur Sofiène Affes** pour la confiance qu'il m'a investie d'intégrer l'équipe Wireless Lab. Je le remercie pour ses prodigieux conseils et directives qui m'ont guidé tout au long de l'accomplissement de ces travaux de recherche.

J'exprime également ma gratitude la plus profonde à **Dr. Abdelaziz Samet**, mon co-directeur de recherche, pour ses critiques constructives, ses suggestions pertinentes et l'attention qu'il m'a bien voulu apporté à mes travaux à divers stades de ses élaborations.

Mes remerciements vont aussi à tous les membres de l'équipe Wireless Lab et à toute personne qui a contribué de près ou de loin à ce travail.

Finalement, je voudrais dédier ce mémoire à ma famille et en particulier à ma mère. Je ne saurais pas exprimer mon grand chagrin en ton absence. J'aurais aimé que tu sois à mes côtés ce jour. Que ce travail soit une prière pour le repos de ton âme.

Résumé

Les réseaux de capteurs sans fil (RCSFs) sont l'un des domaines de recherche les plus populaires sur lesquels travaillent des chercheurs de différentes disciplines depuis plusieurs années. Un RCSF est constitué de petits capteurs à faible coût qui sont généralement déployés dans une zone d'intérêt et peuvent interagir avec l'environnement en détectant ou en contrôlant des paramètres physiques tel que la température, la lumière, la pression, etc. et offrent des possibilités de surveiller et de contrôler des maisons, des villes et des environnements. Ces dispositifs ont une capacité de calcul très limitée, et ils sont alimentés par de petites batteries ou un composant de captage d'énergie, et ils communiquent par des liaisons sans fil.

Les RCSFs collectent les données des capteurs et les transmettent à une station centrale (nœuds-puits). Cependant, ces données sont souvent totalement ou partiellement inutiles si l'emplacement à partir duquel elles ont été mesurées est inconnu, ce qui fait de la localisation des nœuds une tâche essentielle dans les RCSFs à saut multiple. Il est donc crucial de connaître l'emplacement des nœuds de capteurs sans fil, par exemple dans des applications telles que la surveillance des pipelines dans les zones arides, le suivi des animaux, la détection des incendies dans les forêts, etc.

Plusieurs algorithmes de localisation ont été proposés. Une de ces techniques, qui a été largement étudiée dans la littérature, est le DV-Hop. Afin de développer davantage cet axe de recherche, le travail présenté dans ce mémoire consiste à développer et analyser les performances d'une nouvelle version améliorée de cette technique de référence en utilisant une nouvelle expression pondérée de la taille du saut et son évaluation en conditions d'opération réelles sur des plateformes MICAz à l'aide de système d'exploitation open source Contiki et le simulateur COOJA. Les résultats de la simulation par Matlab montrent une amélioration de la performance en termes de précision de localisation. De plus, l'implémentation de cet algorithme en utilisant des nœuds capteurs réels présente un *proof-of-concept* valorisable aux yeux de la communauté scientifique et industrielle.

Keywords Contiki-Cooja; DV-Hop; localisation; MICAz; *range-free*; RCSF.

Abstract

Wireless sensor networks (WSN) are one of the most popular research domains that researchers from different disciplines have been working on for the last several years. A WSN consists of individual low-cost tiny sensor devices that are usually scattered in a sensor field and are able to interact with the environment by sensing or controlling physical parameters such as temperature, light, pressure, etc and provide unprecedented opportunities for monitoring and controlling homes, cities, and the environment. These devices have a very limited computational capability, are powered by small batteries or an energy harvesting component, and communicate over wireless links.

WSNs gather sensor data and forward it to a central station (sink). However, their data are often fully or partially meaningless if the location from where they have been measured is unknown, making the nodes localization an essential task in multi-hop WSNs. It is therefore crucial to know the locations of wireless sensor nodes location, for instance in applications such as pipeline monitoring in arid areas, animal tracking, forest fire detection, etc.

Various localization algorithms have been proposed. One of the widely investigated WSN localization techniques is the DV-Hop. In order to further develop this line of research, the work presented in this thesis consists to develop and analyze the performance of a new enhanced version of this reference technique using a novel weighted *hop-size* expression and its implementation in the real-world motes on MICAz platforms using the Contiki open source operating system and a COOJA simulator. The Matlab simulation results indicate an improvement of the performance in terms of accuracy, and the implementation on real-world motes makes a proof-of-concept of our proposed localization algorithm.

Keywords Contiki-Cooja; DV-Hop; localisation; MICAz; *range-free*; Wireless Sensor Networks.

Table des matières

Remerciements	iii
Résumé	v
Abstract	vii
Table des matières	ix
Liste des figures	xi
Liste des tableaux	xiii
Liste des abréviations	xv
Introduction générale	1
1 Généralités sur les réseaux de capteurs sans fil	5
1.1 Introduction	5
1.2 Les réseaux de capteurs sans fil (WSN)	6
1.2.1 Qu'est-ce qu'un nœud capteur?	6
1.2.2 Les composants d'un nœud capteur	6
1.2.3 Architecture d'un réseau de capteurs sans fil	7
1.2.4 Topologie d'un réseau de capteurs sans fil	8
1.3 Communications dans les WSNs	9
1.3.1 Architecture protocolaire de communication	9
1.3.2 Technologies de la communication dans les WSNs	10
1.4 Domaines d'application	10
1.5 Défis des réseaux de capteurs sans fil	11
1.6 Conclusion	12
2 Localisation dans les réseaux de capteurs sans fil	14
2.1 Introduction	14
2.2 Système de localisation	14
2.3 Techniques d'estimation de distance/angle	16
2.3.1 Temps d'arrivée (TOA)	16
2.3.2 Différence de temps d'arrivée (TDOA)	17
2.3.3 Angle d'arrivée (AOA)	17
2.3.4 Indicateur de puissance du signal reçu (RSSI)	18
2.4 Calcul de la position	19

2.4.1	Trilatération et multilatération	19
2.4.2	Triangulation	21
2.5	Algorithmes de localisation	21
2.6	Conclusion	23
3	Nouvel algorithme de localisation range-free pour les réseaux de capteurs sans fil	25
3.1	Introduction	25
3.2	La technique DV-Hop	26
3.2.1	Principe du DV-Hop	26
3.2.2	Étude d'améliorations de DV-Hop	28
3.3	Nouvel algorithme de Localisation	29
3.3.1	Calcul optimisé et pondérée de la taille moyenne du saut d'ancre	29
3.3.2	Méthode de localisation	31
3.4	Simulations et résultats	32
3.4.1	Erreur de localisation en fonction du pourcentage d'ancres	33
3.4.2	Erreur de localisation en fonction de la densité	35
3.4.3	Erreur de localisation en fonction de la portée	36
3.4.4	Distribution de l'erreur cumulée	37
3.5	Conclusion	37
4	Implémentation et évaluation en conditions d'opération réelles	40
4.1	Introduction	40
4.2	Environnement de travail	40
4.2.1	Contiki Operating System pour les RCSFs	41
4.2.2	Le simulateur Cooja pour les applications Contiki	44
4.2.3	La plateforme MICAz	45
4.2.4	Module programmeur MIB520 pour la famille MICA	45
4.3	Implémentation d'algorithmes de localisation sur la plateforme MICAz	46
4.4	Validation des performances sur la plateforme MICAz	49
4.5	Conclusion	52
	Conclusion générale	53
	Références	55

Liste des figures

1.1	Architecture d'un capteur sans fil.	6
1.2	Architecture d'un réseau de capteur sans fil.	8
1.3	Pile protocolaire de communication pour les réseaux de capteurs sans fil.	9
1.4	Les applications de RCSFs.	11
2.1	Genèse d'un système de localisation.	15
2.2	Principe du TOA.	16
2.3	Principe du TDOA.	17
2.4	Principe de la trilatération.	19
2.5	Principe de la triangulation.	21
3.1	Distribution des nœuds.	33
3.2	Erreur normalisée de localisation en fonction du nombre d'ancres.	34
3.3	Erreur normalisée de localisation pour chaque nœud avec $N = 100$ dont 10% d'ancres.	34
3.4	Erreur normalisée de localisation en fonction de la densité.	35
3.5	Erreur normalisée de localisation en fonction de la portée.	36
3.6	CDF de localisation NRMSE avec 100 nœuds dont 10% d'ancres	37
4.1	Architecture de Contiki.	43
4.2	Mécanismes de communication.	43
4.3	Interface graphique de Cooja.	44
4.4	Plateforme MICAz.	45
4.5	Module programmeur MIB520.	45
4.6	Étapes d'implémentation.	46
4.7	Organigramme d'un nœud ancre.	47
4.8	Organigramme d'un nœud inconnu.	48
4.9	Validation en conditions d'opération réelles.	49
4.10	Validation en conditions d'opération réelles du NRMSE de localisation en fonction de nombre d'ancres.	50
4.11	Validation en conditions d'opération réelles du NRMSE de localisation en fonction de la densité.	51
4.12	Validation en conditions d'opération réelles du NRMSE de localisation en fonction de la portée.	51

Liste des tableaux

4.1 Paramètres de simulation. 50

Liste des abréviations

6LoWPAN	IPv6 over Low-Power Wireless Personal Area Networks
ADC	Analog-to-Digital Converter
AOA	Angle Of Arrival
APS	Ad hoc Positioning System
BS	Base Station
COOJA	Contiki OS Java Simulator
CPU	Central Processing Unit
DC	Distance Vector
GPS	Global Positioning System
IP	Internet Protocol
LOS	line Of Sight
LPR	Low Power Radio
MCU	Micro Controller Unit
MEMS	(Microelectromechanical Systems
MSP	Mixed-Signal Microcontroller
OS	Operating System
PSO	Particle Swarm Optimization
RAM	Random Access Memory
RAM	Read Only Memory
RCSF	Réseaux de Capteurs Sans Fils
RMSE	Root Mean Square Error
RSSI	Received Signal Strength Indicator
TDOA	Time Difference of Arrival
TOA	Time of Arrival
WSN	Wireless Sensor Networks

Introduction générale

Les dernières recherches en technologie des systèmes micro-électro-mécaniques (microelectromechanical Systems (MEMS) en anglais), la télécommunication et l'électronique ont permis de créer des nœuds capteurs, multifonctionnels et de faible consommation d'énergie. De plus, le contrôle et la surveillance des paramètres physiques tel que la température, la pression, etc..., sont nécessaires pour plusieurs applications industrielles et scientifiques. Ce besoin a engendré la naissance d'un nouveau domaine qui offre des solutions économiques et faciles à déployer assurant la surveillance des environnements : les réseaux de capteurs sans fil : RCSFs (Wireless Sensor Networks (WSNs) en anglais) [1]-[4].

Un RCSF est composé d'un ensemble de petits nœuds électroniques autonomes coopérant entre eux via une communication sans fil [5]. Ce réseau est capable d'examiner une zone géographique, de fournir des informations par la combinaison des calculs par les différents nœuds capteurs et en suite les acheminer à travers une station de base. Ces capteurs sont dotés de capacité de calcul qui facilite la réalisation des applications infaisables avant quelques années. Ils ont pu être intégrés dans plusieurs applications surtout dans le domaine environnemental, militaire, médical et domotique. Le déploiement réel d'un réseau de capteurs sans fil pose plusieurs défis notamment la localisation de nœuds capteurs objet de ce travail.

La position de capteurs est une information primordiale pour l'exploitation des données fournies par le réseau [6][7]. En effet, de nombreux travaux ont proposé des solutions efficaces et optimisées. La localisation consiste à déterminer la position physique des nœuds au sein du réseau. Ce dernier n'aura pas intérêt si les données qu'il permet de collecter ne sont pas localisées.

Au cours des dernières années, les chercheurs ont travaillé sur une large gamme d'algorithmes de localisation [8][9]. Dans la plupart des cas, un petit ensemble de nœuds connus sous le nom d'ancres

et équipés d'une capacité de localisation intégrée telle qu'un module GPS serviront de référence pour l'estimation des autres emplacements de nœuds inconnus. Deux principales catégories d'algorithmes de localisation prévalent dans la littérature: *range-based* et *range-free*. Ces derniers exploitent les informations de connectivité des réseaux afin d'approximer les emplacements des nœuds inconnus sans recours à aucun matériel supplémentaire, tandis que les premiers accomplissent cela à l'aide d'équipements supplémentaires en mesurant certaines caractéristiques des signaux échangés entre les nœuds tels que AOA, TOA, RSS, etc...

Grâce à leurs faibles coûts et de leurs facilités d'implémentation, les algorithmes de type *range-free* jouissent d'une plus grande popularité et ils sont plus pratiques pour les RCSFs. L'un des représentants typiques de ce type d'algorithme et le largement étudié est le DV-Hop [10]. Ce dernier se caractérise par sa simplicité et son indépendance par rapport aux mesures de distances entre les nœuds grâce à sa technique qui ne repose que sur le nombre de sauts. Cependant, cette méthode n'est pas assez performante en terme de précision.

C'est dans ce contexte que le présent mémoire s'insère. Afin de rendre cette technique de localisation plus performante en terme de précision, nous présentons dans ce travail un nouvel algorithme basé sur la technique DV-Hop. Nous évaluons et confirmons ensuite cette nouvelle version par des simulations Matlab pour différents scénarios. De plus, nous implémentons cette nouvelle technique, ainsi que le DV-Hop sur des plateformes MICAz pour valider ses performances en conditions d'opération réelles.

Dans ce rapport, nous allons présenter notre travail en quatre volets comme suit:

- Dans un premier temps, nous présentons les réseaux de capteurs sans fil, les concepts de base de ces réseaux, leurs caractéristiques, leurs applications et les différents défis de ce type de réseaux.
- Dans le second chapitre, nous nous intéressons à la localisation dans les réseaux de capteurs sans fil. Nous présentons tout d'abord le concept du système de localisation, ainsi que les techniques utilisées pour la localisation en général.
- Puis, nous détaillons notre nouvel algorithme de localisation IWDV-Hop [11] basé sur la technique DV-Hop en expliquant son principe et l'acheminement de notre raisonnement. La confirmation se fera à l'aide d'évaluations par des simulations Matlab en en comparaison avec DV-Hop et HWDV-Hop [12].

- Dans le dernier chapitre, nous présentons notre environnement de travail matériel et logiciel et les différentes étapes d'implémentation d'algorithmes IWDV-Hop et DV-Hop sur des plateformes MICAz ainsi que les résultats de la validation de ses performances.
- Finalement, nous achevons ce rapport par une conclusion générale sur le travail présenté de ce projet.

Chapitre 1

Généralités sur les réseaux de capteurs sans fil

1.1 Introduction

Un réseau de capteur sans fil [1]-[5] est composé d'un nombre très grand de nœuds capteurs, déployés de manière aléatoire coopérant entre eux via une communication sans fil en multi-saut afin d'adresser les informations collectés vers une unité à l'extérieur de la zone du réseau. Ce type de réseau est capable de s'auto-configurer et de s'autogérer de manière à améliorer son efficacité et sa fonctionnalité sans interventions humaines. Ceci est grâce aux capacités des capteurs ou des nœuds du réseau à coopérer entre eux. Ces réseaux sont intéressants particulièrement pour les applications militaires, environnementales, domestiques, médicales ainsi d'autres applications diverses.

Ce chapitre consiste à présenter les réseaux de capteurs sans fil, leurs architectures, leurs applications, les facteurs de conception et les défis lors de sa réalisation, ainsi que les problématiques imposées par ce type de réseau.

1.2 Les réseaux de capteurs sans fil (WSN)

1.2.1 Qu'est-ce qu'un nœud capteur?

Un nœud capteur sans fil est un petit dispositif électronique doté d'un ou plusieurs capteurs qui sont capables de mesurer plusieurs valeurs physiques environnementales (température, lumière, pression, etc.) ou encore physiologiques (glycémie, tension, etc.). Chaque nœud capteur garantit trois fonctions principales: la collecte de l'information, son traitement et son acheminement vers un ou plusieurs points de collecte appelés puits (station de base).

1.2.2 Les composants d'un nœud capteur

Un nœud capteur est composé de quatre unités [1]: l'unité de captage, l'unité de traitement, l'unité de transmission et l'unité de contrôle d'énergie. Il peut aussi contenir un système de localisation ou bien un système mobilisateur pour le rendre mobile en cas de nécessité grâce à une source d'alimentation, comme indique la figure 1.1.

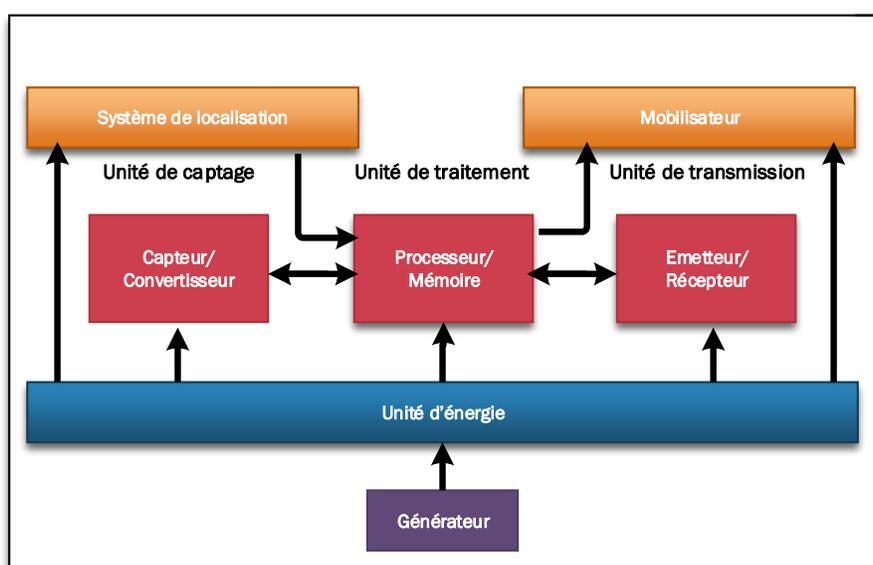


Figure 1.1 – Architecture d'un capteur sans fil.

- **L'unité de captage (Sensing unit):** est composée d'un ou plusieurs capteurs et d'un convertisseur Analogique/Numérique (ADC). Les capteurs obtiennent des mesures sur les pa-

ramètres environnementaux sous forme analogique et l'ADC convertit ces données analogiques en données numériques compréhensibles par l'unité de traitement.

- **L'unité de traitement (Processing unit):** contient deux interfaces, une interface avec l'unité d'acquisition et une interface avec l'unité de transmission. Cette unité est également composée d'un processeur qui supporte un système d'exploitation spécifique. Elle permette de traiter toutes les données résultant de l'unité de captage.
- **L'unité de transmission (Transceiver unit):** cette unité est responsable de toutes les émissions et les réceptions de données via un support de communication sans fil et une antenne.
- **L'unité d'énergie:** un capteur est muni d'une source d'énergie, généralement une batterie, pour alimenter ses composants. Les batteries utilisées peuvent d'être rechargeables par l'énergie solaire ou non dans le cas où les capteurs sont déployés à l'abri du soleil. Souvent, dans les environnements hostiles il est impossible de recharger ou changer une batterie. Pour cela, l'énergie est considérée comme une ressource très précieuse dans les WSNs puisqu'elle influe directement sur la durée de vie des capteurs.

Un nœud capteur peut contenir également, suivant son domaine d'application, des modules supplémentaires tels qu'un système de localisation (GPS), ou bien un système générateur d'énergie.

1.2.3 Architecture d'un réseau de capteurs sans fil

Un RCSF est un réseau de microsystemes autonomes déployés, dans une zone de contrôle bien déterminé communiquant entre eux via une liaison sans fil. L'espace où les capteurs agissent s'appelle zone de couverture (ou champ de captage).

Il est composé de deux types de nœuds:

- **Nœud capteur:** Le rôle de capteur est de surveiller son environnement. Ce nœud doit être alimenté par une pile qui représente une source d'énergie. Il est chargé pour transmettre les données collectées à un ou plusieurs nœuds puits.
- **Point de collecte:** Il joue le rôle d'une passerelle et responsable de l'acheminement des informations collectées provenant des nœuds capteurs vers l'ordinateur centrale à travers un réseau de communication.

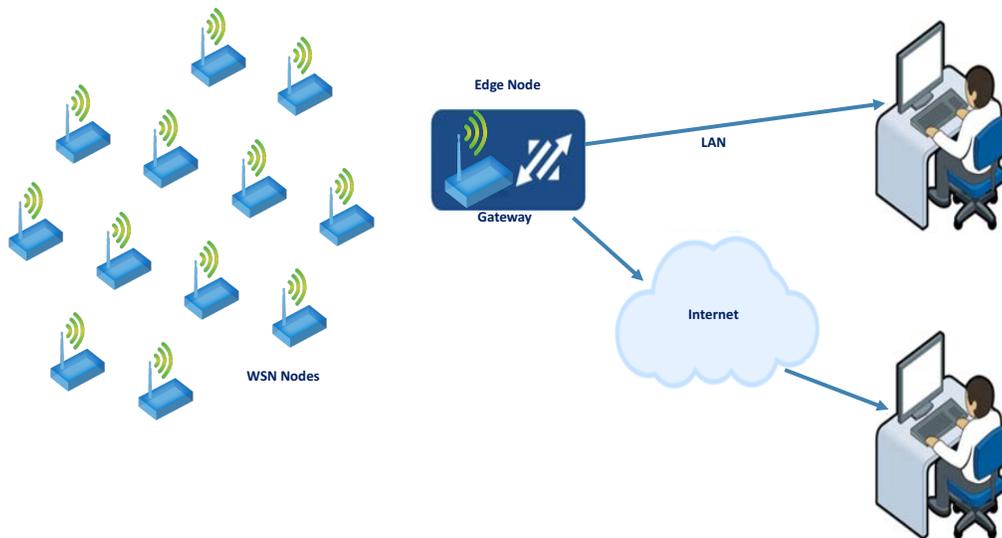


Figure 1.2 – Architecture d'un réseau de capteur sans fil.

1.2.4 Topologie d'un réseau de capteurs sans fil

L'acheminement des données dans les RCSFs se fait selon plusieurs schémas de routage qui sont sous-jacents à des topologies. Il existe plusieurs topologies pour les réseaux de capteurs:

- **Topologie en étoile:** est un système uni-saut dont les nœuds capteurs communiquent directement avec le point de collecte. Cette topologie est la plus simple et exige que la distance entre les nœuds et la passerelle soit limitée.
- **Topologie en toile:** est un système multi-sauts dont la communication entre les nœuds capteurs et la passerelle pourrait être directe ou via un schéma de routage multi-sauts. C'est le cas le plus fréquent où les nœuds servent, aussi plus que la transmission de leurs propres données de captage, de relais pour d'autres capteurs. Dans ce type de topologie, le problème de routage est l'un des défis les plus importants dans les RCSFs.
- **Topologie hybride:** est une combinaison des deux topologies ci-dessus. Les points de collectes (Stations de base) forment une topologie en toile et les nœuds autour d'elles sont en topologie étoile.

1.3 Communications dans les WSNs

1.3.1 Architecture protocolaire de communication

Les points de collectes et les nœuds capteurs utilisent une pile protocolaire pour la communication. Cette pile contient cinq couches qui ont les mêmes fonctions que celles du modèle OSI, plus trois couches pour la gestion d'énergie, de mobilité et des tâches comme le montre la figure 1.3 suivante:

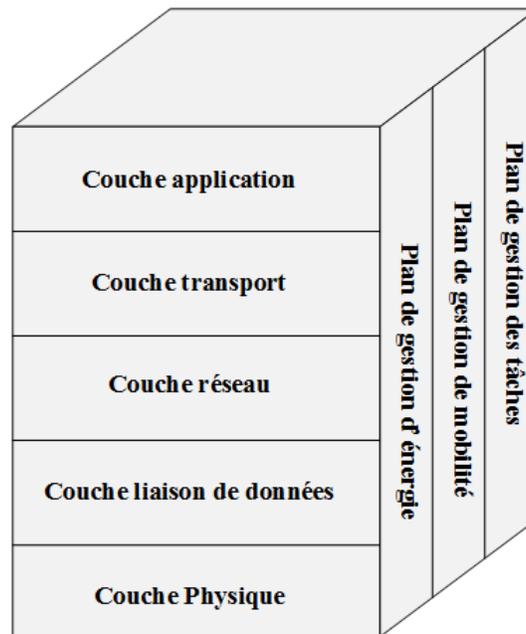


Figure 1.3 – Pile protocolaire de communication pour les réseaux de capteurs sans fil.

- **La couche physique:** permet de sélectionner des fréquences, la génération des ondes porteuses, la détection du signal, la modulation et le cryptage des données. Elle assure la transmission et la réception des données au niveau bit.
- **La couche liaison des données:** permette de déterminer comment les données sont adressées entre deux nœuds.
- **La couche réseau:** permet de garantir l'acheminement des données et de trouver le chemin le plus court entre le nœud et la passerelle de point de vue énergie, temps et coût.
- **La couche transport:** est nécessaire si le réseau est prévu d'être accessible à partir d'autres réseaux externes. Elle fournit un service de communication de bout en bout et fiable pour l'application.

- **La couche application:** garantie l'interface avec les applications.

1.3.2 Technologies de la communication dans les WSNs

Dans les RSCF, les nœuds sont souvent équipés de batteries. C'est qui rend la gestion de la consommation d'énergie une tâche difficile pour les RSCFs. Cela a conduit au développement de divers protocoles pour répondre aux besoins des réseaux des capteurs pour une faible consommation de puissance comme les protocoles ZigBee (IEEE 802.15.4), Bluetooth (802.15.1) et Dash7 (ISO/IEC 18000-7).

1.4 Domaines d'application

Les réseaux de capteurs envahissent divers domaines d'application suite aux derniers développements réalisés dans le domaine des techniques sans fil et au large gamme des types de capteurs disponibles. Parmi ces applications, on cite; les applications militaires, le processus industriels, l'agriculture de précision, le contrôle d'habitat, les applications environnemental etc...

Selon leurs modes d'utilisation, les applications des réseaux de capteurs sont classifiées en deux catégories: la surveillance (monitoring) et le suivi (tracking). Figure 1.4.

Les réseau de capteurs sans fil sont confrontés à des différents défis et contraintes selon l'environnement. Ils sont déployés sur la terre, sous la terre et sous eau.

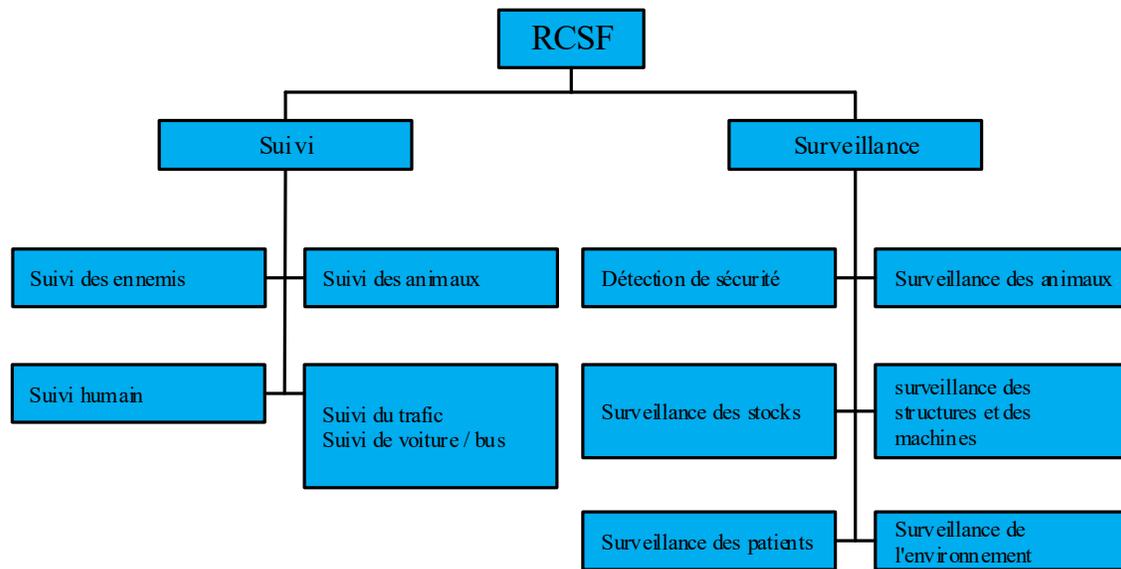


Figure 1.4 – Les applications de RCSFs.

1.5 Défis des réseaux de capteurs sans fil

Dans la littérature, de nombreux travaux de recherche se concentrent sur la proposition de solutions efficaces et optimales à des défis des réseaux de capteurs sans fil, parmi lesquels on cite :

- **Localisation:** Pour accomplir certaines applications des réseaux de capteurs sans fil comme le repérage des cibles et la détection des feux, il est nécessaire de connaître les positions des nœuds capteurs responsables de la détection de ces événements pour garantir une intervention dans le lieu et le temps convenables. Des contraintes comme la modification de la topologie et l'absence d'infrastructure influencent sur la localisation. Donc, il faut développer des algorithmes fiables et économes afin d'assurer une localisation efficace tout en minimisant la consommation énergétique.
- **Synchronisation:** La synchronisation temporelle est une problématique importante dans les réseaux de capteurs sans fil à communication multi-saut car certaines applications de WSNs demandent une synchronisation des horloges locales des nœuds.
- **Efficacité énergétique:** Les capteurs disposent des ressources énergétiques limitées (batterie). C'est qui fait de l'efficacité énergétique un facteur important dans la conception de tout capteur puisque le fonctionnement de WSN dépend fortement de la durée de vie des batteries

de nœuds de capteurs. Alors il est très important d'optimiser l'énergie pour pouvoir étendre et prolonger cette durée.

- **Routage:** Il s'agit de l'envoi des données sortant des nœuds sources jusqu'aux nœuds puits. Ces informations sont communiquées via un chemin multi-sauts pour atteindre leurs destinations. Il faut donc trouver le chemin optimal le plus court et adopter des algorithmes de routage les plus efficaces afin d'assurer le routage des données en respectant le délai de transfert des paquets et les contraintes d'énergie

1.6 Conclusion

Dans ce chapitre, nous avons présenté les réseaux de capteurs sans fil. Nous avons présenté quelques concepts de base telle que la définition et le principe de fonctionnement de ce type du réseau, l'architecture adopté, la pile protocolaire et ses différentes domaines d'applications.

Plusieurs défis des réseaux de capteurs sans fil pour lesquels de nombreux travaux de recherches se concentrent sur la proposition des solutions efficaces et optimales. Parmi ces défis, nous citons la localisation qui est l'objet de chapitre suivant.

Chapitre 2

Localisation dans les réseaux de capteurs sans fil

2.1 Introduction

Pour accomplir certaines applications des réseaux de capteurs sans fil comme le repérage des cibles et la détection des feux, il est nécessaire de connaître les positions des nœuds capteurs responsables de la détection de ces événements pour garantir une intervention dans le lieu et le temps convenables. Ainsi, sans connaître la position d'où elles proviennent, les informations ne racontent qu'une partie de l'histoire. Des contraintes comme la modification de la topologie et l'absence d'infrastructure influencent sur la localisation. Donc, il faut inventer des algorithmes fiables et économes afin d'assurer une localisation efficace tout en minimisant la consommation énergétique.

Ce chapitre vise à présenter le système de localisation ainsi que les techniques utilisées pour la localisation en général.

2.2 Système de localisation

Les coordonnées de nœuds capteurs peuvent être prédéterminées et préconfigurées. Cependant, le déploiement aléatoire des nœuds dans la région d'intérêt met en évidence la nécessité d'un système

de localisation [13] pour permet aux capteurs l'estimation de leurs emplacements. Un système de localisation est identifié impérativement comme un facteur clé pour le développement et l'utilisation des RCSFs.

Il peut être décomposé, comme illustré par la figure 2.1, en trois parties distinctes:

- **Estimation de distance:** permet l'estimation des distances et/ ou d'angles qui séparent les nœuds dans le réseau à partir des diverses caractéristiques des signaux radio échangés entre eux.
- **Calcul de la position:** permet d'estimer les coordonnées d'un nœud en exploitant les mesures disponibles de la première partie et les positions des nœuds de références (ancres).
- **Algorithme de localisation:** il définit la manière de manipulation des données disponibles (distances, angles, positions des ancres) afin que la totalité des nœuds puissent estimer leurs positions.

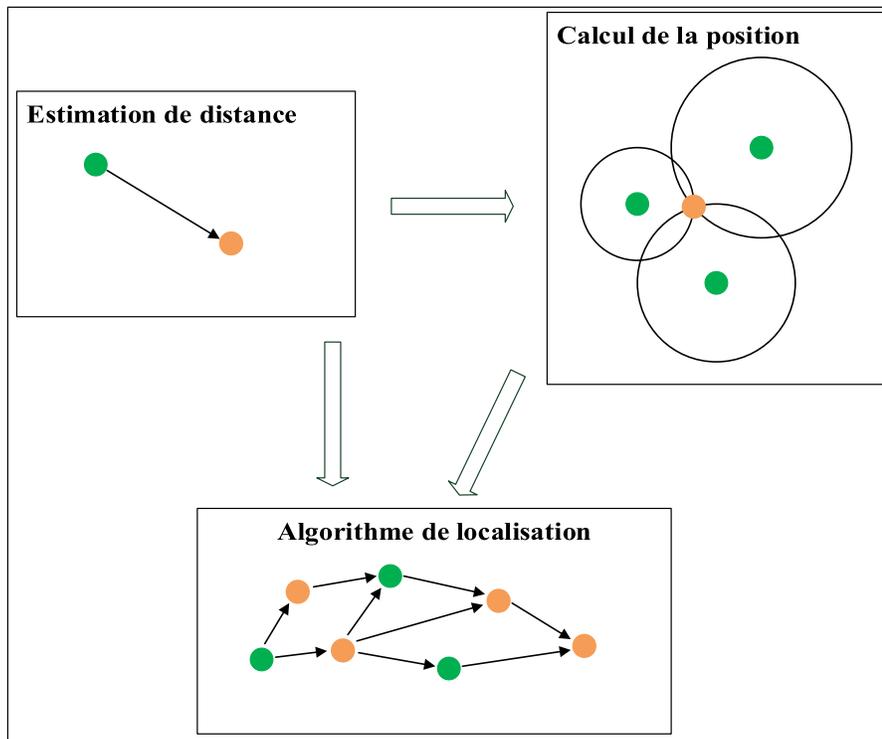


Figure 2.1 – Genèse d'un système de localisation.

2.3 Techniques d'estimation de distance/angle

Plusieurs méthodes d'estimation de distance entre les paires de nœuds sont utilisées. Ils sont classifiés principalement en deux catégories *range-based* et *range-free* [14][10]. Pour la première catégorie, les distances estimées sont obtenues par des mesures de certaines caractéristiques des signaux échangés entre les capteurs, tels que le temps de propagation des signaux, la puissance de signaux (*signal strength*), ou l'angle d'arrivée [6][15]. Tandis que dans la deuxième, elles sont obtenues en se basant sur des informations de connectivité.

2.3.1 Temps d'arrivée (TOA)

Pour la technique TOA (Time of Arrival), présentée par la figure 2.2, il suffit de connaître la vitesse et le temps de propagation du signal entre deux nœuds afin d'estimer la distance entre eux par la simple multiplication suivante:

$$Distance_{i,j} = Temps \times Vitesse \quad (2.1)$$

Où le temps est la différence entre t_1 et t_2 qui sont respectivement les instants d'envoi et de réception.

Cette méthode est utilisée par le système GPS (Global Positioning System) et elle nécessite que les nœuds du réseau soient synchronisés.

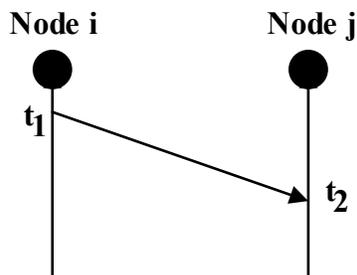


Figure 2.2 – Principe du TOA.

2.3.2 Différence de temps d'arrivée (TDOA)

La méthode TDOA (Time Difference of Arrival), présentée par la figure 2.3, se base sur la différence des temps d'arrivée de deux signaux qui ont des vitesses de propagation différentes. Cette technique requiert néanmoins des équipements supplémentaires au niveau de chaque capteurs. En général, le premier dispositif utilise des ondes radio (vitesse $V = 3.10^8$ m/s) et le deuxième utilise des ondes sonores (vitesse $V = 340$ m/s) ou ultrasonores. Le premier nœud émet les deux signaux simultanément ou séparés par un intervalle de temps t_{wait} . Par conséquent, la distance peut être calculée au niveau de récepteur avec la formule suivante:

$$Distance_{i,j} = (V_1 - V_2) \times (t_4 - t_2 - t_{wait}) \quad (2.2)$$

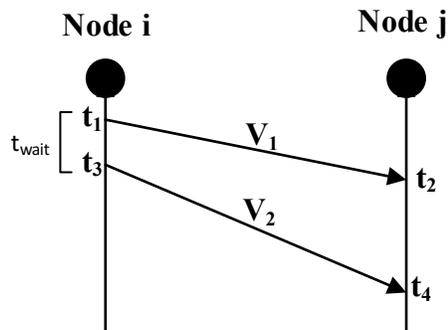


Figure 2.3 – Principe du TDOA.

Cette technique ne nécessite pas que les deux nœuds soient synchronisés. Mais pour une meilleur précision, les capteurs doivent être en visibilité directe (LOS).

2.3.3 Angle d'arrivée (AOA)

La technique AOA (Angle Of Arrival) [16] permet la détermination de l'angle d'arrivée du signal formé entre deux nœuds capteurs. Pour cela, chaque nœud doit être équipé d'un ensemble de petites antennes ou des émetteurs/récepteurs à ultrason. Pour déduire l'AOA du signal, le nœud utilise la phase ou le temps d'arrivée du signal au niveau de chaque antenne (ou récepteur ultrason).

Cette technique fournit des résultats précis (environ 5°). Mais, elle nécessite des matériels supplémentaires qui la rendent trop coûteuse.

2.3.4 Indicateur de puissance du signal reçu (RSSI)

La technologie RSSI (Received Signal Strength Indicator) exploite la puissance d'émission et de réception d'un signal pour estimer la distance entre deux nœuds capteurs. En effet, elle calcule la perte de propagation d'un signal entre son émission et sa réception. Par la suite, cette perte sera traduite en une distance.

Selon le modèle "binary disc model", la perte ou l'atténuation de puissance du signal noté $PL(d)$ (y compris les gains des antennes) est une fonction déterministe $PL_0(d)$ de la distance d entre les nœuds. $PL_0(d)$, exprimé en dB, est de la forme suivante:

$$PL_0(d) = PL(d_0) + 10n \log_{10} \left(\frac{d}{d_0} \right) \quad (2.3)$$

Où:

- d : la distance entre l'émetteur et le récepteur.
- n : le coefficient d'affaiblissement de propagation (entre 3 et 5).
- d_0 : distance de référence (généralement, 1 mètre ou 1 kilomètre).
- $PL(d_0)$: l'atténuation dû à l'affaiblissement à la distance de référence d_0 .

En pratique, la présence de certains éléments tels que le terrain, l'environnement, les obstacles ou encore la mobilité des nœuds (path loss, shadowing, fadings) peuvent engendrer plusieurs phénomènes physiques qui dégradent la qualité du signal. Ainsi, des modèles statistiques prenant en compte ses différents phénomènes physiques sont étudiés. Le modèle de signal le plus largement utilisé est le modèle log-Normal avec effet de masque (log-normal shadowing):

$$PL(d) = PL_0(d) + X_\sigma = PL(d_0) + 10n \log_{10} \left(\frac{d}{d_0} \right) + X_\sigma \quad (2.4)$$

Où X_σ est une variable aléatoire qui suit une loi normale de moyenne nulle et d'écart-type X_σ (déviation standard en dB).

Le grand avantage de cette technique est son faible coût car tous les nœuds sont déjà équipés de dispositif de communication sans fil. Cependant, elle est sensible aux bruits, interférences, asymétrie du lien de communication (RSSI non uniforme), obstacles et types d'antenne.

2.4 Calcul de la position

Quand un nœud a suffisamment d'information de distances/angles et des positions des nœuds de références (anchors), il peut calculer sa position. Pour cela, plusieurs méthodes sont utilisées. Parmi ces méthodes : la multilatération, la triangulation, la trilatération, etc...

2.4.1 Trilatération et multilatération

La trilatération est la méthode la plus simple et intuitive. Elle consiste au calcul de la position d'un nœud par l'intersection de trois cercles comme elle présente la figure 2.4.

Pour qu'un nœud peut estimer sa position (en 2D), il suffit qu'il connait les positions de trois "anchors" non colinéaires et sa distance par rapport à ces trois nœuds de références. La distance peut être estimée en utilisant l'une des méthodes décrites précédemment (par exemple RSSI).

Pour les "anchors", ses coordonnées sont connues à priori et peuvent être préenregistrées ou obtenues à l'aide d'un dispositif externe comme le GPS.

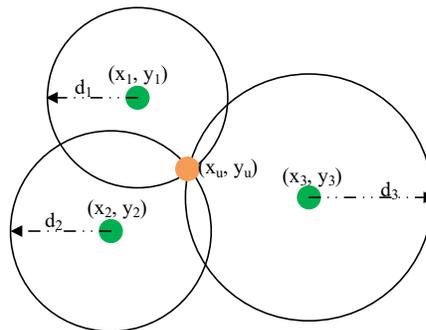


Figure 2.4 – Principe de la trilatération.

Supposons que les coordonnées des n nœuds d'ancrage sont donnés par $N_i = (x_i, y_i)$ ($i=1\dots n$) et que les distances entre un capteur inconnu $N_u = (x_u, y_u)$ et ces nœuds d'ancrage sont également

connus par d_{ui} . Ainsi, les relations entre les positions et les distances entre le nœud inconnu et les "anchors" seront exprimées par la matrice suivante:

$$\begin{bmatrix} (x_1 - x_u)^2 + (y_1 - y_u)^2 \\ (x_2 - x_u)^2 + (y_2 - y_u)^2 \\ \vdots \\ (x_n - x_u)^2 + (y_n - y_u)^2 \end{bmatrix} = \begin{bmatrix} d_{u1}^2 \\ d_{u2}^2 \\ \vdots \\ d_{un}^2 \end{bmatrix} \quad (2.5)$$

Ainsi, nous obtenons:

$$AP = B \quad (2.6)$$

Avec

$$A = 2 \times \begin{bmatrix} x_n - x_1 & y_n - y_1 \\ x_n - x_2 & y_n - y_2 \\ \vdots & \vdots \\ x_n - x_{n-1} & y_n - y_{n-1} \end{bmatrix}, \quad (2.7)$$

$$B = \begin{bmatrix} d_{u1}^2 - d_{un}^2 - x_1^2 + x_n^2 - y_1^2 + y_n^2 \\ d_{u2}^2 - d_{un}^2 - x_2^2 + x_n^2 - y_2^2 + y_n^2 \\ \vdots \\ d_{u(n-1)}^2 - d_{un}^2 - x_{n-1}^2 + x_n^2 - y_{n-1}^2 + y_n^2 \end{bmatrix}, \quad (2.8)$$

$$P = \begin{bmatrix} x_u \\ y_u \end{bmatrix} \quad (2.9)$$

Ce système peut être résolu avec une méthode standard comme la minimisation de la somme des erreurs quadratiques, dont la solution optimale est donnée par:

$$P = (A^T A)^{-1} A^T B \quad (2.10)$$

2.4.2 Triangulation

La triangulation ressemble à la trilatération, mais elle utilise les angles à la place des distances. Elle exploite les propriétés géométriques de triangles pour estimer la position de nœud inconnu en utilisant les lois de la trigonométrie (cosinus et sinus). Dans ce cas, au moins trois "anchors" sont nécessaires. Le nœud à localiser estime ses angles avec chaque ancre en utilisant la technique AOA par exemple. La figure 2.5 illustre le principe de l'auto-localisation en utilisant la triangulation.

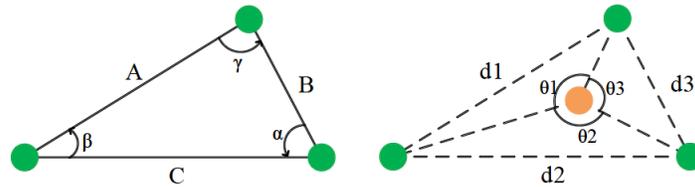


Figure 2.5 – Principe de la triangulation.

Pour la dérivation de la position, le théorème de Pythagore généralisé sera utilisé, dont les formules sont données par les équations suivantes:

$$\begin{cases} A^2 = B^2 + C^2 - 2BC \cos \alpha \\ B^2 = A^2 + C^2 - 2AC \cos \beta \\ C^2 = A^2 + B^2 - 2AB \cos \gamma \end{cases} \quad (2.11)$$

Ainsi, en connaissant les distances d_1 , d_2 , d_3 entre les ancres et les angles θ_1 , θ_2 , θ_3 , le capteur calcule sa position.

2.5 Algorithmes de localisation

L'algorithme de localisation est la partie principale du système de localisation. Il définit la méthode avec laquelle les informations disponibles sont exploitées afin que la grande partie, ou la totalité, des nœuds du RSCF estiment leurs positions.

En effet, plusieurs catégories d'algorithmes de localisation sont actuellement disponibles:

- **Centralisés ou Distribués:** pour les premiers, toutes les informations sont transmises à un nœud central. Ce dernier est appelé station de base ou nœud récepteur doté de capacités de traitement et de stockage bien plus grandes que celles des capteurs normaux. Il est responsable de prendre en charge le calcul de position de tous les nœuds. Tandis que pour les algorithmes distribués, chaque nœud de capteur est responsable du calcul et de l'estimation de sa position individuellement.
- **Avec ou sans infrastructures:** S'il y a besoin ou pas d'une infrastructure. Par exemple la position de quelques nœuds est connue à l'avance. Ces nœuds sont connus en tant que des ancrs qui servent comme des références pour déterminer l'emplacement des autres nœuds non localisés.
- **Position relative ou absolue:** Une position estimée est dite absolue lorsqu'elle est définie dans un repère global (latitude et longitude). Cependant une position relative est définie par rapport à un objet.
- **Déploiement en intérieur ou en extérieur:** Si l'algorithme est mieux adapté aux applications à l'intérieur (*indoor*) ou plutôt à l'extérieur (*outdoor*).

De nombreux paramètres doivent être pris en compte lors de la développement d'un algorithme de localisation. Ces paramètres aident à évaluer les performances de différentes approches de systèmes localisation. Parmi lesquels nous citons:

- **Précision:** C'est un paramètre clé pour la localisation. C'est la distance euclidienne entre les positions réelles des nœuds et celles estimées par l'algorithme.
- **Coût:** Le coût est un autre paramètre important et constitue le problème majeur de la plupart des techniques de localisation à cause des ressources très limitées de nœuds capteurs. Un algorithme de localisation doit être simple et non complexe et son développement n'exige pas de grands calculs ni de grande capacité du stockage de mémoire.
- **Nombre de nœuds localisés:** C'est le nombre de nœuds ayant réussi à estimer leur position à la fin de processus de localisation. L'idéal est que tous les nœuds parviennent à calculer leurs positions, mais, ce n'est pas toujours le cas dans plusieurs algorithmes.
- **Nombre d'ancres:** C'est le nombre de nœuds références nécessaires pour le bon fonctionnement d'un algorithme de localisation. Ces ancrs sont souvent plus chers que les nœuds normaux car ils sont équipés par des dispositifs comme des GPS.

2.6 Conclusion

Dans ce chapitre, nous avons abordé la problématique de la localisation de nœuds capteurs dans les RCSFs. Nous avons également présenté le système de localisation et les différents techniques utilisés pour l'estimation de la distance entre deux nœuds. Ainsi que les différentes catégories d'algorithmes de localisation tels que *range-based* et *range-free*. Ces derniers jouissent d'une plus grande popularité grâce à leurs faibles coûts et de leurs facilités d'implémentation et ils sont plus pratiques pour les WSNs. La précision et le faible coût sont les deux éléments clés de plusieurs études et les chercheurs leur vouent encore un réel intérêt.

Dans le chapitre suivant, nous allons présenter un nouvel algorithme de localisation en se fondant sur l'un des algorithmes *range-free* les plus étudiés (DV-Hop).

Chapitre 3

Nouvel algorithme de localisation range-free pour les réseaux de capteurs sans fil

3.1 Introduction

Les performances d'un système de localisation sont fortement liées aux performances de chacune de ses composantes. Et afin d'assurer une localisation efficace avec le meilleur compromis entre performances en termes de coûts et de consommation énergétique, nous nous sommes intéressés dans notre travail à la méthode d'estimation des distances sur laquelle se basera l'algorithme de localisation. L'idée principale consiste à déduire l'erreur d'estimation de la distance moyenne entre deux nœuds capteurs, à partir d'informations sur le nombre de sauts qui'il les sépare.

Dans ce chapitre, nous présentons l'algorithme DV-Hop et de travaux améliorés relatifs à cette technique. Puis, nous détaillons notre nouvel algorithme de localisation basé sur DV-Hop en expliquant l'idée de départ. Ensuite, nous évaluons et confirmons par des simulations les gains de précision qui s'avèrent significatifs pour l'état actuel de la technique.

3.2 La technique DV-Hop

A l'instar des techniques *range-based*, les méthodes *range-free* disposent d'un arsenal d'algorithmes permettant de calculer la position à partir de simples déductions. Un avantage majeur de cette famille relève du fait que les outils de traitement d'information ne sont pas généralement complexes. Dans la suite, la simple utilisation du nombre de sauts et de la connectivité proposée par cette technique illustrera cet aspect. Il convient néanmoins de signaler que cette réduction de la complexité s'accompagne d'une perte significative de précision des positions estimées.

DV-Hop est l'un des représentants typiques des algorithmes de localisation *range-free*. Sa version originale ainsi que plusieurs améliorations sont présentées dans la section suivante.

3.2.1 Principe du DV-Hop

Similaire à un schéma de routage de base utilisant le concept de l'échange de vecteur de distance (DV), Niculescu et Nath ont développé un algorithme de localisation de distance de vecteur (DV-hop) [10]. DV-hop ne nécessite aucun équipement supplémentaire car il utilise uniquement l'estimation de la distance multi-sauts. Nous pouvons le diviser en trois étapes principales:

- **Étape 1:** Le nombre minimum de sauts séparant chaque nœud inconnu d'un nœud ancre est compté. Chaque nœud ancre inonde un paquet dans tout le réseau où il indique ses coordonnées et un attribut *hop-count* dédié au comptage de sauts initialisé à 1. Un nœud de réception encaisserait alors seulement les coordonnées et le *hop-count* d'ancre ayant la valeur minimum. Ensuite, chaque nœud qui reçoit le paquet incrémentera le nombre de sauts de 1. À la fin de cette étape, tous les nœuds du réseau (ancres et inconnus) auront une table contenant le nombre de sauts minimum à chaque ancre plus ses coordonnées géographiques.
- **Étape 2:** Une fois que tous les ancres ont récupéré les positions et les nombres de sauts les séparant de leurs homologues, chaque nœud ancre estime alors la taille moyenne du saut *hop-size* en utilisant la formule (3.1) et la diffuse. L'étape suivante consiste à calculer la taille moyenne du saut \bar{h}_i^s par chaque nœud ancre i en tant que rapport entre la distance physique réelle à leurs homologues sur la valeur du nombre de sauts. Cette distance moyenne est estimée

par la formule suivante:

$$\bar{h}_i^s = \frac{\sum_{j \neq i} \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}}{\bar{h}_{ij}^c \times (N_a - 1)}, \quad (3.1)$$

Avec $(x_i, y_i), (x_j, y_j)$ sont les coordonnées des nœuds ancres i et j respectivement, la valeur du nombre de sauts *hop-count* $\bar{h}_{i,j}^c$ est le nombre minimum de sauts entre ces derniers, et N_a est le nombre total d'ancres.

Chaque ancre estime sa taille moyenne de saut et l'inonde à l'ensemble du réseau. Les nœuds inconnus reçoivent les informations de taille moyenne de saut et sauvegardent la valeur reçue de l'ancre le plus proche en diffusant cette valeur à leurs voisins. À la fin de cette étape, un nœud inconnu avec des coordonnées (x_u, y_u) estime sa distance à chaque ancre j comme suit:

$$\hat{d}_{uj} = \bar{h}_j^s \times \bar{h}_{uj}^c. \quad (3.2)$$

- **Étape 3:** Dans le cas où les distances séparant un nœud inconnu, avec des coordonnées (x_u, y_u) , vers plus que deux ancres sont connues, nous utilisons la méthode d'estimation du maximum de vraisemblance afin de déterminer son emplacement:

$$\begin{bmatrix} (x_1 - x_u)^2 + (y_1 - y_u)^2 \\ (x_2 - x_u)^2 + (y_2 - y_u)^2 \\ \vdots \\ (x_n - x_u)^2 + (y_n - y_u)^2 \end{bmatrix} = \begin{bmatrix} \hat{d}_{u1}^2 \\ \hat{d}_{u2}^2 \\ \vdots \\ \hat{d}_{un}^2 \end{bmatrix}. \quad (3.3)$$

L'équation (3.3) peut être transformée en une matrice sous la forme suivante:

$$AX_u = B, \quad (3.4)$$

Avec:

$$A = 2 \times \begin{bmatrix} x_n - x_1 & y_n - y_1 \\ x_n - x_2 & y_n - y_2 \\ \vdots & \vdots \\ x_n - x_{n-1} & y_n - y_{n-1} \end{bmatrix}, \quad (3.5)$$

$$B = \begin{bmatrix} \hat{d}_{u1}^2 - \hat{d}_{un}^2 - x_1^2 + x_n^2 - y_1^2 + y_n^2 \\ \hat{d}_{u2}^2 - \hat{d}_{un}^2 - x_2^2 + x_n^2 - y_2^2 + y_n^2 \\ \vdots \\ \hat{d}_{u(n-1)}^2 - \hat{d}_{un}^2 - x_{n-1}^2 + x_n^2 - y_{n-1}^2 + y_n^2 \end{bmatrix}, \quad (3.6)$$

$$X_u = \begin{bmatrix} x_u \\ y_u \end{bmatrix}. \quad (3.7)$$

Selon la méthode des moindres carrés (*Least Square*), la solution à l'équation (3.4) est donnée par:

$$\hat{X}_u = (A^T A)^{-1} A^T B. \quad (3.8)$$

L'algorithme DV-Hop se caractérise par sa simplicité et son indépendance par rapport aux mesures de distances grâce à sa technique qui ne repose que sur les nombres de sauts pour estimer les distance entre les capteurs. Cependant cette méthode n'est pas assez performante en terme de précision puisque elle considère que la taille d'un saut est toujours la même au sein du réseau. Ce qui n'est pas le cas dans la réalité. Ainsi, de nombreux algorithmes basés sur DV-Hop ont été proposés pour améliorer sa précision.

3.2.2 Étude d'améliorations de DV-Hop

L'algorithme DV-Hop est toujours très populaire en raison de sa simplicité, de sa rentabilité et de sa robustesse. Par conséquent, les chercheurs continuent d'améliorer la précision en utilisant plusieurs approches (par exemple en modifiant la taille moyenne du saut *hop-size*, le déploiement spécial des ancrs, correction des coordonnées estimées, etc...).

Chen et al. [17] ont proposé une méthode pour améliorer la précision de localisation du DV-Hop en utilisant une seule taille de saut pour l'ensemble des nœuds. Cette taille est la moyenne des tailles par saut de tous les ancrs. Ils ont également adopté la méthode hyperbolique 2D au lieu de la trilatération traditionnelle.

Une autre version améliorée nommée WDV-Hop a été développée dans [18]. Là, contrairement au DV-hop classique où les nœuds ne considèrent que la taille moyenne du saut la plus proche, chaque nœud inconnu calcule leur propre taille de saut comme une moyenne pondérée de toutes les tailles de saut d’ancres.

Dans [19], une version améliorée de DV-Hop a été proposée en trois étapes de traitement. Certains ancres sont d’abord déployés aux frontières du champ d’intérêt. Ensuite, chaque nœud inconnu utilise une taille moyenne du saut pondérée pour estimer sa position par une méthode hyperbolique 2D. Enfin, les positions estimées sont corrigées par PSO (*Particle Swarm Optimization*).

Dans une autre variante de DV-Hop développée dans [12], la taille moyenne du saut à chaque ancre est calculée comme une somme pondérée de toutes les distances moyennes à tous les ancres. Le poids de chaque ancre est la somme inverse appliquée à la différence de distance entre son estimation initiale de la taille moyenne du saut et celles calculées aux autres ancres.

3.3 Nouvel algorithme de Localisation

Les erreurs de localisation dans le cas de DV-Hop sont dues d’une part au calcul du nombre de sauts et de la taille moyenne du saut de chaque ancre, et d’autre part à la méthode d’estimation de localisation. Afin de rendre la technique DV-Hop plus précise, nous proposons une version améliorée en travaillant sur les deux dernières étapes, à savoir le calcul de la taille moyenne du saut et la méthode d’estimation de la localisation. Nous gardons la première phase de l’algorithme DV-hop où tous les nœuds dans le réseau obtiennent le nombre minimal de sauts à chaque ancre, alors que nous utilisons un calcul pondéré pour le *hop-size* et nous modifions la méthode d’estimation des coordonnées.

Dans ce qui suit, nous détaillons la méthode de calcul du *hop-size*, puis nous présentons notre nouvelle approche d’estimation de l’emplacement.

3.3.1 Calcul optimisé et pondérée de la taille moyenne du saut d’ancre

Comme il est illustré pendant la première phase de DV-hop original, chaque ancre calcule son *hop-size* en utilisant la formule (3.1), et estime les distances avec chaque autre ancre en utilisant

l'équation (3.2). Par la suite, une erreur quadratique moyenne (RMSE) est obtenue. Cette erreur est déterminée comme suit:

$$E = \frac{1}{N_a - 1} \sum_{i \neq j} (\|\hat{X}_i - \hat{X}_j\| - d_{ij}) , \quad (3.9)$$

Avec d_{ij} est la distance entre les ancres i et j Selon le critère de MSE, les erreurs d'estimation de *hop-size* à chaque ancre i peuvent donc être réduites comme suit:

$$\bar{h}_i^s = \frac{\sum_{i \neq j} \bar{h}_{ij}^c \|\hat{X}_i - \hat{X}_j\|}{\sum_{i \neq j} \bar{h}_{ij}^c{}^2} . \quad (3.10)$$

En DV-Hop, chaque nœud inconnu exploite uniquement le *hop-size* calculé par l'ancre le plus proche pour estimer son propre emplacement. Cependant, dans la situation réelle les nœuds dans le WSN sont disposés aléatoirement, ce qui signifie que le chemin entre les nœuds peut ne pas être droit et la taille moyenne du saut d'un seul ancre n'est pas suffisante pour refléter les propriétés du réseau entier. Par conséquent, la précision de *hop-size* sera affectée et nous n'arrivons pas à obtenir des coordonnées encore plus précises de l'emplacement des nœuds. C'est pourquoi, dans notre version améliorée, nous avons choisi d'exploiter tous les *hop-size* calculés par les ancres. Cependant, accorder le même niveau d'importance ou de fiabilité à tous ces derniers serait infructueux. En fait, après avoir calculé la taille moyenne du saut en utilisant (3.10), nous pondérons chaque nœud ancre comme suit:

$$W_i = \frac{\frac{1}{\bar{h}_{iu}^c}}{\sum_{k=1}^{N_a} \left(\frac{1}{\bar{h}_{ku}^c}\right)} , \quad (3.11)$$

Ce poids dépend du nombre de sauts puisque l'ancre le plus proche fournirait une localisation plus précise que les autres.

Pour refléter le poids de chaque ancre, chaque nœud inconnu utilise une somme pondérée des tailles de sauts de tous les ancres pour calculer sa propre *hop-size*. Ce dernier est déterminé par la formule suivante:

$$\bar{h}_u^s = \sum_{i=1}^{N_a} W_i \bar{h}_i^s . \quad (3.12)$$

3.3.2 Méthode de localisation

Au lieu d'appliquer la méthode ML, utilisée par DV-Hop, pour calculer les positions des nœuds inconnus, nous adoptons l'algorithme de localisation hyperbolique [20] afin d'obtenir des localisations plus précises.

Supposons que les coordonnées d'un ancre i sont données par (x_i, y_i) et les coordonnées d'un nœud inconnu u sont (x_u, y_u) . La distance d_{iu} entre i et u est donnée par:

$$d_{iu}^2 = (x_i - x_u)^2 + (y_i - y_u)^2. \quad (3.13)$$

Si $A_i = x_i^2 + y_i^2$ et $B_u = x_u^2 + y_u^2$, nous obtenons la formule suivante:

$$d_{iu}^2 - A_i = -2x_i x_u - 2y_i y_u + B_u. \quad (3.14)$$

Cette équation peut être réécrite sous forme matricielle comme suit:

$$JU = I, \quad (3.15)$$

Avec,

$$J = \begin{bmatrix} -2x_1 & -2y_1 & 1 \\ -2x_2 & -2y_2 & 1 \\ \vdots & \vdots & \\ -2x_n & -2y_n & 1 \end{bmatrix}, \quad (3.16)$$

$$I = \begin{bmatrix} d_{1u}^2 - A_1 \\ d_{2u}^2 - A_2 \\ \vdots \\ d_{nu}^2 - A_n \end{bmatrix}, \quad (3.17)$$

$$U = \begin{bmatrix} x_u & y_u & B_u \end{bmatrix}^T \quad (3.18)$$

Selon l'équation (3.15), U peut être obtenue en utilisant la méthode des moindres carrés comme suit:

$$\hat{U} = (J^T J)^{-1} J^T \hat{I} \quad (3.19)$$

Ainsi, les coordonnées du nœud inconnu sont:

$$\hat{x}_u = \hat{U}(1),$$

$$\hat{y}_u = \hat{U}(2).$$

3.4 Simulations et resultas

Plusieurs simulations à différents scénarios sont présentées dans cette section afin d'évaluer notre algorithme IWDV-Hop en terme de précision de localisation par rapport aux variations de la portée de communication, des pourcentages d'ancres distribués dans le réseau et du nombre de nœuds.

Les performances de la nouvelle technique sont comparées à celles de DV-Hop [10] ainsi que la version améliorée HWDV-Hop [12]. Les simulations ont été réalisées avec le logiciel MATLAB en calculant chaque moyenne de performance sur 100 réalisations Monte-Carlo. Comme illustré dans la figure 3.1, tous les nœuds sont aléatoirement distribués au sein d'une surface de $2500m^2$ ($50 \times 50m^2$). La portée R de chaque nœud ancre ou inconnu est fixée à 15 m.

Cette distribution uniforme des nœuds au sein de la zone d'intérêt a pour objectif de garantir que les propriétés du réseau (à savoir la connectivité, la portée ainsi que la densité) soient quasiment les mêmes partout, c'est le cas de la topologie isotropique [21].

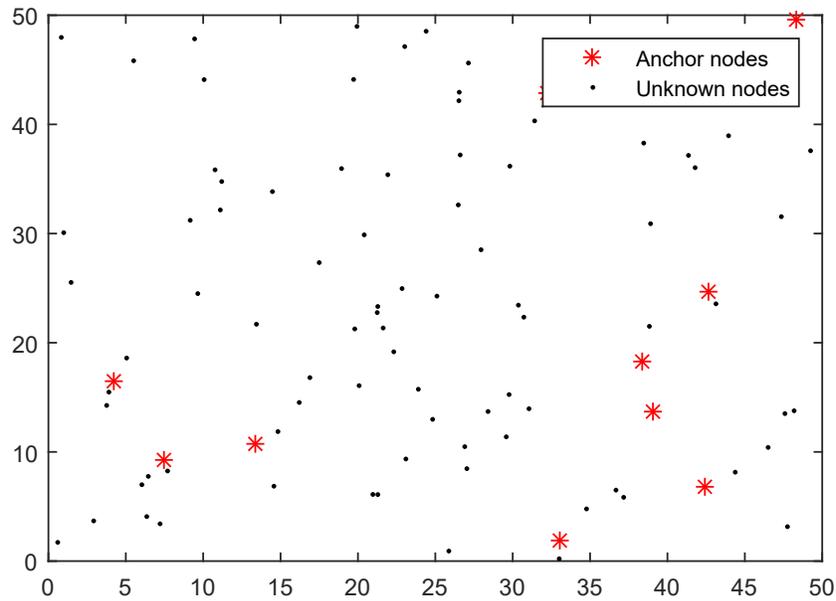


Figure 3.1 – Distribution des nœuds.

3.4.1 Erreur de localisation en fonction du pourcentage d’ancres

La figure 3.2 montre les performances des trois algorithmes avec différents nombres d’ancres allant de 4% à 22% du nombre total de nœuds $N = 100$ avec $R = 15m$.

En tant que mesure de performance, nous proposons l’erreur quadratique moyenne normalisée (NRMSE) qui est définie comme suit:

$$NRMSE = \frac{1}{N_u} \sum_{i=1}^{N_u} \frac{\sqrt{(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2}}{R}. \quad (3.20)$$

Avec N_u est le nombre de nœuds inconnus.

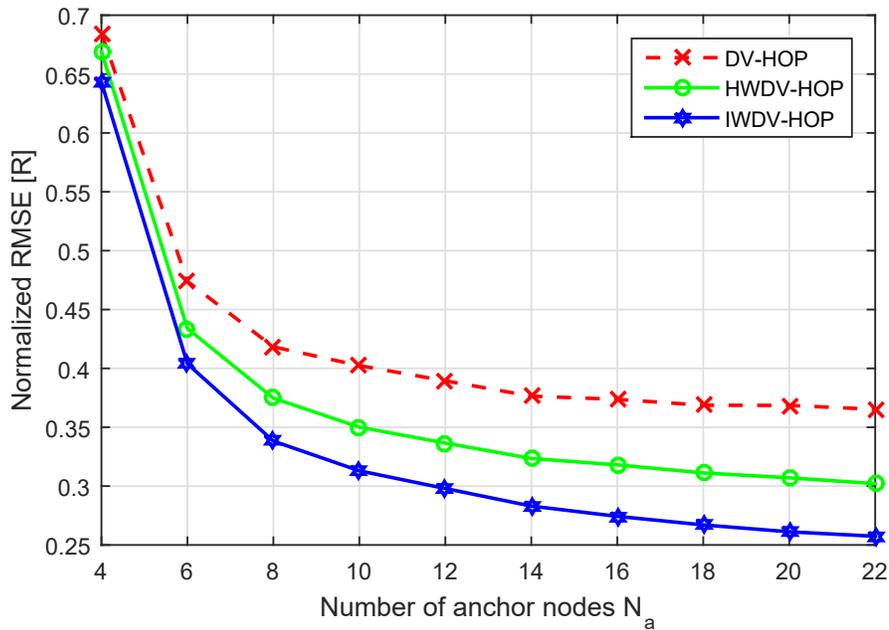


Figure 3.2 – Erreur normalisée de localisation en fonction du nombre d’ancres.

Nous remarquons que la précision des trois algorithmes s’améliore avec l’augmentation du nombre d’ancres. Pourtant, sans ambiguïté, notre version proposée est constamment la plus précise et surpasse les deux autres références.

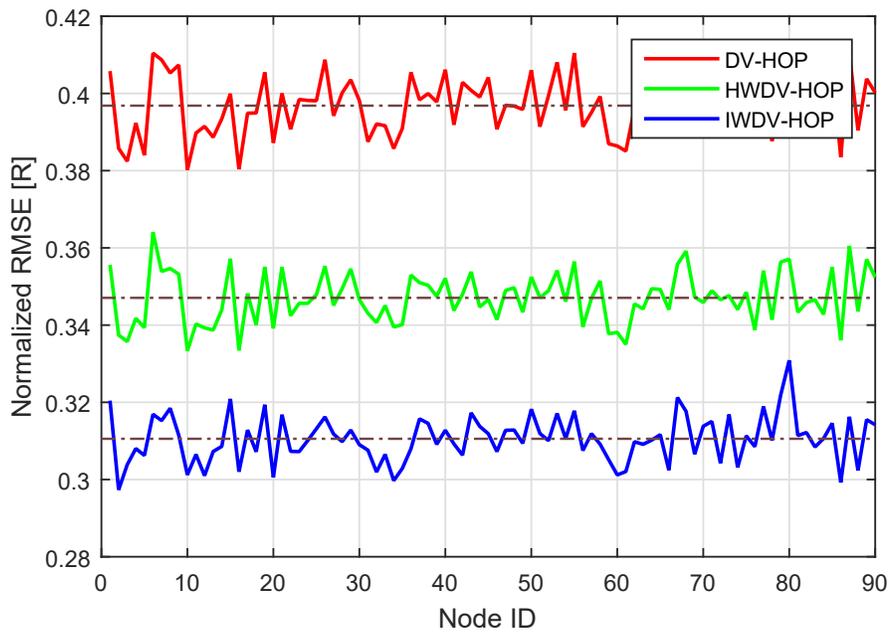


Figure 3.3 – Erreur normalisée de localisation pour chaque nœud avec $N = 100$ dont 10% d’ancres.

Pour mieux montrer les gains de précision de la nouvelle technique, nous illustrons dans la figure 3.3 l'erreur normalisée de localisation pour chaque nœud sur un réseau simulé de 100 nœuds avec 10 ancres.

3.4.2 Erreur de localisation en fonction de la densité

Cette simulation consiste à calculer l'erreur quadratique moyenne normalisée de nombres différents de nœuds ayant la même portée R (15m). Pour ce faire, nous avons choisi de changer ce nombre de 100 à 500, avec un taux d'ancres fixé à 10% du nombre total de nœuds. L'objectif ici est d'étudier l'impact de la densité sur l'erreur de localisation.

La figure 3.4 illustre la variation de l'erreur de localisation en fonction du nombre de nœuds déployés.

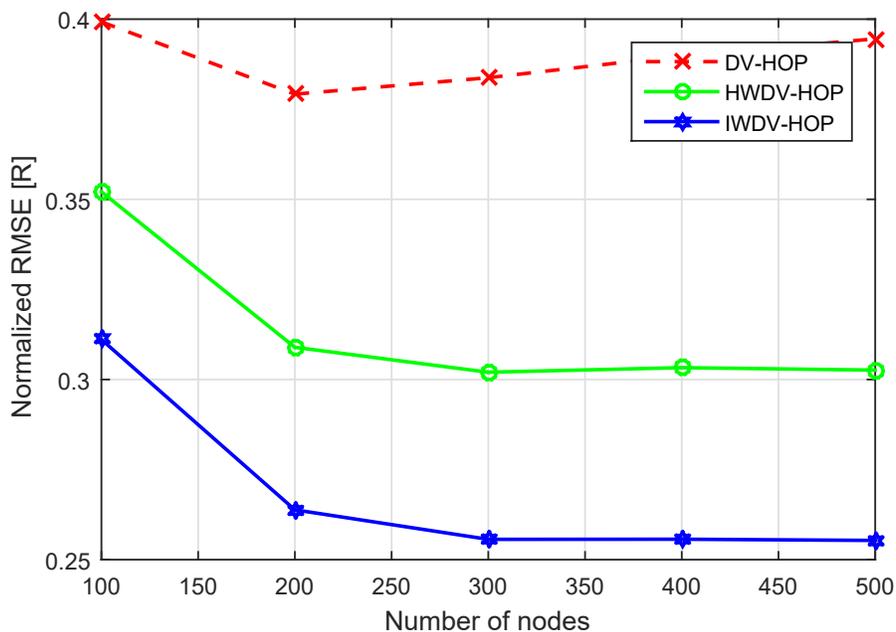


Figure 3.4 – Erreur normalisée de localisation en fonction de la densité.

Nous remarquons l'amélioration prévue et surtout la supériorité de la nouvelle technique de localisation proposée par rapport au DV-Hop et sa variante améliorée. C'est que prouve que notre méthode est plus efficace en cas de fortes densités.

3.4.3 Erreur de localisation en fonction de la portée

La figure 3.5 montre l'erreur quadratique moyenne normalisée de localisation lorsque la portée de communication R augmente progressivement de 10 m à 20 m. Dans ce scénario, nous avons 20% d'ancres parmi 100 nœuds en gardant toujours la même surface de simulation de $2500m^2$ ($50 \times 50m^2$).

Et encore une fois, comme prévu, la précision de localisation des trois algorithmes s'améliore avec l'augmentation des valeurs de la portée R . Cependant, notre nouvelle technique IWDV-Hop est plus précise comparée aux deux algorithmes DV-Hop et HWDV-Hop.

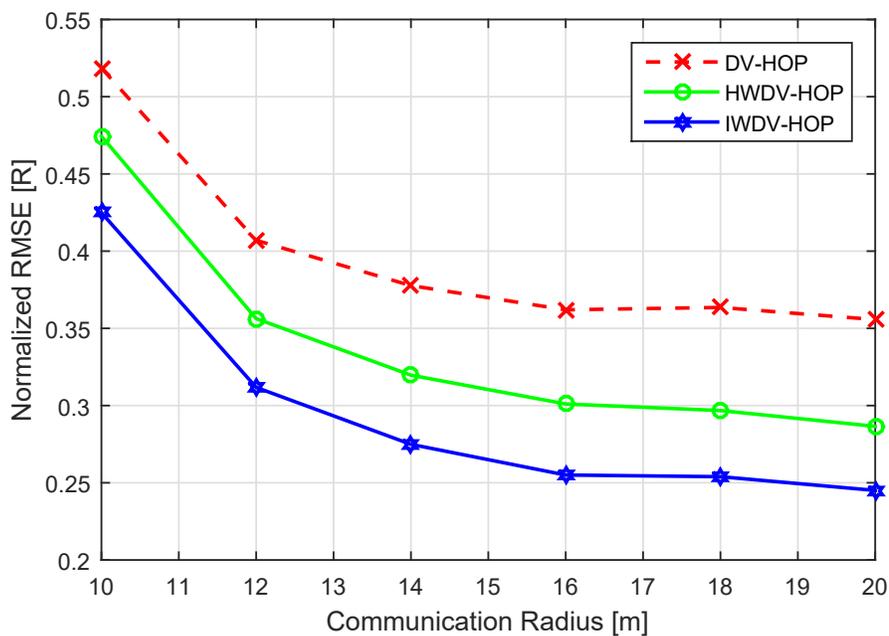


Figure 3.5 – Erreur normalisée de localisation en fonction de la portée.

3.4.4 Distribution de l'erreur cumulée

Cette simulation consiste à calculer le CDF de localisation NRMSE. Pour ce scénario, nous avons fixé le nombre d'ancres au sein du réseau à 10% du nombre total de nœuds $N = 100$ en gardant la même surface de simulation $2500m^2$ ($50 \times 50m^2$) et la même portée $R = 15m$.

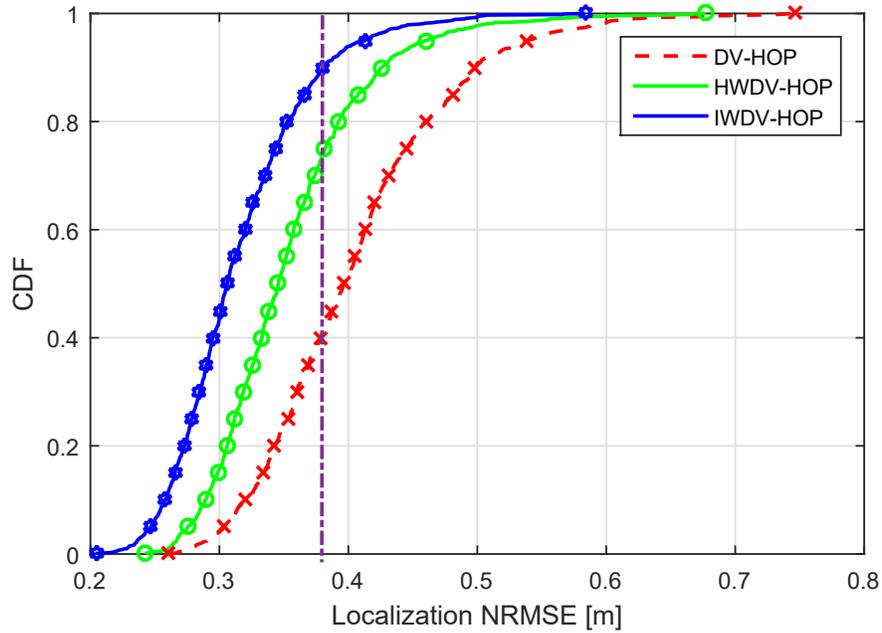


Figure 3.6 – CDF de localisation NRMSE avec 100 nœuds dont 10% d'ancres .

Nous remarquons dans la figure 3.6 qu'avec l'utilisation de la technique IWDV-Hop proposée, 90% des nœuds inconnus pourraient estimer leurs positions à moins de $38\%R$ tandis que seulement $40\%R$ et $75\%R$ des nœuds inconnus atteignent la même précision avec DV-Hop et HWDV-Hop respectivement. C'est que prouve encore l'efficacité de notre nouvel algorithme.

3.5 Conclusion

Au cours de ce chapitre, nous avons étudié l'algorithme de localisation de type range-free DV-Hop pour les réseaux de capteurs sans fil, ainsi que de travaux relatifs à cet algorithme. Nous avons également présenté notre méthode, en expliquant son principe et l'acheminement de notre raisonnement. Cette nouvelle approche utilise les poids des différents ancres dans le réseau pour une meilleure estimation de *hop-size* dans chaque nœud inconnu, et applique la méthode hyperbo-

lique pour le calcul des positions. Les résultats des simulations indiquent une amélioration de la performance en termes de précision.

De nombreux algorithmes de localisation intéressants existent dans la littérature, mais ils ne sont pas sortis des laboratoires de recherche, et ne sont pas implémentés et testés dans des conditions réelles. Dans ce contexte, l'objet du chapitre suivant est d'implémenter notre nouvel algorithme IWDV-Hop, ainsi que le DV-Hop sur des plateformes MICAz afin de valider leurs performances en terme de précision.

Chapitre 4

Implémentation et évaluation en conditions d'opération réelles

4.1 Introduction

La dernière étape dans notre projet a consisté à passer de la théorie à la pratique afin de tester et valider notre algorithme en conditions d'opération réelles. Cette étape est très importante, et nous permet de savoir si l'implémentation matérielle de notre nouvelle approche a le même comportement et donne les mêmes résultats que celles qui sont obtenus par MATLAB.

Dans ce chapitre, nous commençons par la présentation de notre environnement de travail matériel et logiciel. Ensuite, nous évaluons le comportement des algorithmes implémentés.

4.2 Environnement de travail

Les RCSFs sont considérés parmi les systèmes à ressources limitées à cause des caractéristiques des nœuds capteurs telles qu'une mémoire réduite, une ressource d'énergie très limitée, un processeur de faible fréquence de calcul, etc. De ce fait, les outils logiciels traditionnels conçus aux ordinateurs comme Linux ou Windows ne sont plus adaptables à ce type de systèmes. Pour cela, plusieurs

systèmes d'exploitation légers ont été développés [22]-[24] pour répondre aux attentes des systèmes embarqués et qui sont dédiés spécialement aux RCSFs.

Dans cette section, nous allons présenter le système d'exploitation Contiki qui est un système *open-source* embarqué, le simulateur Cooja et la plateforme MICAz.

4.2.1 Contiki Operating System pour les RCSFs

Présentation

Contiki [25][26] est un système d'exploitation *open-source*, intégré, modulaire, flexible, générique qui s'appuie sur un modèle de fonctionnement hybride, destiné aux réseaux de capteurs. Ce système a été développé par un groupe de développeurs de l'industrie et du monde universitaire par Adam Dunkels de l'institut suédois d'informatique en 2004. Il fonctionne sur les microcontrôleurs de faible puissance et permet de développer des applications qui font une utilisation efficace du matériel tout en fournissant une communication standardisée sans fil de faible puissance pour une gamme de plateformes des microcontrôleurs tels que le Atmel AVR et TI MSP430, qui sont utilisés dans les familles Telos, Tmote et MICA.

Caractéristiques de Contiki

L'évolution de Contiki a été déterminée essentiellement par les caractéristiques suivantes:

- **Normes Internet:** Contiki a présenté l'idée d'utiliser la communication IP dans des réseaux de capteurs basse consommation. En plus il supporte les protocoles IPV6 et 6LOWPAN. Cela s'avère particulièrement utile dans la mesure où les nœuds communiquent en IPV6 et utilisent le standard 802.15.4 définie par l'IEEE.
- **Développement rapide:** Les applications à base de Contiki sont écrites en C standard. Ces applications peuvent être émulées avec le simulateur Cooja avant d'être injectées sur des capteurs réels.
- **Une sélection du matériel:** Il fonctionne sur une gamme de différentes plates-formes matérielles de faible puissance tels que la famille de MICA et la famille de Telos et est conçu pour être facile à porter sur les nouveaux matériels.

- **Logiciel Open Source:** Contiki est un système open source dont le code est toujours disponible. Et il peut être exploité librement dans les deux systèmes commerciaux et noncommerciaux sans restrictions.
- **Couche de protocole Rime:** Rime est une légère couche de communication [27] conçue pour des radios basse puissance et supporte des opérations simples. Elle fournit à la couche applicative un jeu d'instructions de communication, permettant les inondations de réseau, de collecte de données, et les différentes connexions avec les capteurs voisins, pour établir des schémas de routage.
- **Allocation de mémoire:** Contiki est conçu pour les systèmes légers. Il a une empreinte mémoire limitée ayant seulement quelques kilo-octets de mémoire disponible (2 kilo-octets de RAM et 40 kilo-octets de ROM). De plus, il fournit un ensemble de mécanismes d'allocation dynamique de mémoire.
- **Consommation d'énergie:** Contiki a été conçu pour réduire au maximum la consommation en énergie du capteur. Ainsi, lorsqu'aucune tâche n'est active, il se met automatiquement en veille.

Architecture de Contiki

Contiki OS est basé sur une architecture modulaire. Il est constitué d'un noyau, de bibliothèques, d'un ordonnanceur et d'un gestionnaire de processus. Le rôle de Contiki, est de gérer les ressources physiques telles que le processeur, la mémoire, les périphériques (d'entrée/sortie). Le noyau se compose d'un planificateur pour la gestion des événements synchrones et asynchrones. Il utilise un mécanisme qui lui permet de fonctionner comme un système multitâche.

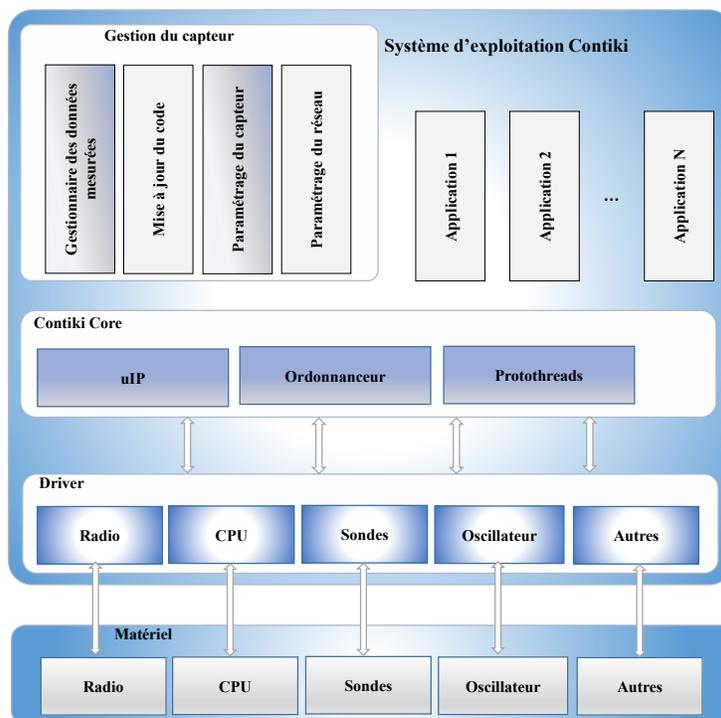


Figure 4.1 – Architecture de Contiki.

La communication dans Contiki

La communication est une tâche principale dans les RCSFs. Ainsi, Contiki implémente deux mécanismes de communication:

- **La couche de protocole Rime:** elle permet un dialogue vers les capteurs voisins ainsi que le routage.
- **La couche uIP:** est une petite pile de TCP/IP orientée Internet, elle offre les services essentiels du protocole IP mais nécessite plus de ressources que Rime.

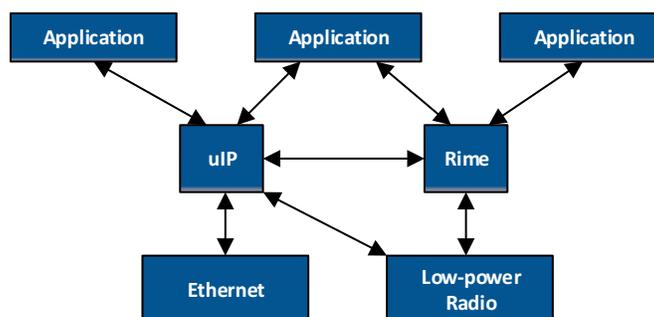


Figure 4.2 – Mécanismes de communication.

4.2.2 Le simulateur Cooja pour les applications Contiki

COOJA [28] est l'acronyme de Contiki OS Java Simulator. Ce simulateur permet l'émulation de manière réaliste de différents nœuds sur lesquels seront chargés le système d'exploitation Contiki et les applications. Cet outil permet de tester des applications à moindre coût avant de les charger dans la mémoire flash des capteurs réels. La figure 4.3 présente une capture écran du simulateur Cooja.

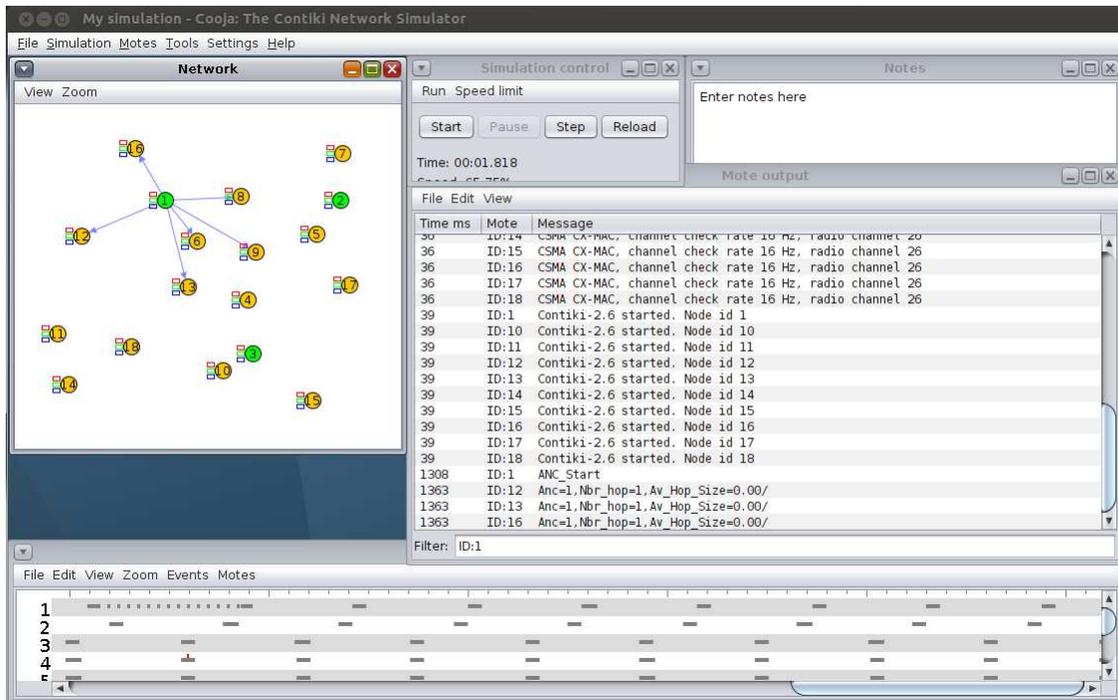


Figure 4.3 – Interface graphique de Cooja.

4.2.3 La plateforme MICAz

Parmi les capteurs supportés par Cooja, on peut citer la plateforme MICAz de MEMSIC. Elle est principalement composée d'un microcontrôleur ATmega128 8MHz et doté de 4 ko de RAM, de 512 ko de mémoire flash et d'une puce radio Chipcon CC2420 qui est devenu le standard au niveau des modules de transmission utilisant la norme IEEE 802.15.4 sur une bande de fréquence 2.4 GHz. Pour fonctionner, un capteur MICAz doit être alimenté par deux piles de 1.5V et il peut être équipé de plusieurs types de circuits intégrés permettant d'effectuer des mesures de phénomènes naturels.



Figure 4.4 – Plateforme MICAz.

4.2.4 Module programmeur MIB520 pour la famille MICA

Afin de charger un programme compilé dans la mémoire du microcontrôleur ATmega128, nous avons utilisé l'outil matériel MIB520. Ce dernier fournit une connectivité USB à la famille de capteurs MICA pour la communication et la programmation interne. Il permet aussi à un nœud MICAz de fonctionner en tant que station de base lorsqu'il est connecté avec lui.

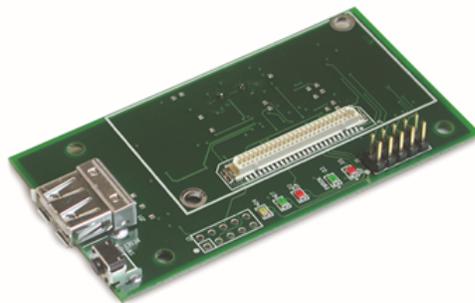


Figure 4.5 – Module programmeur MIB520.

4.3 Implémentation d'algorithmes de localisation sur la plateforme MICAz

Le développement d'un tel système nécessite une étape préliminaire hors ligne qui consiste à valider l'ensemble du scénario de localisation à travers une expérience réaliste. À partir d'une version Matlab de la technique de localisation, nous convertissons l'algorithme décrit en un code de langage C exécutable dans notre simulateur Cooja. Une fois la simulation avec ce dernier est validée, nous passons à l'évaluation en condition d'opération réelles. Ainsi, l'algorithme du langage C sera chargé dans nos nœuds MICAz via un fichier exécutable dans les microcontrôleurs intégrés dans ces dispositifs.

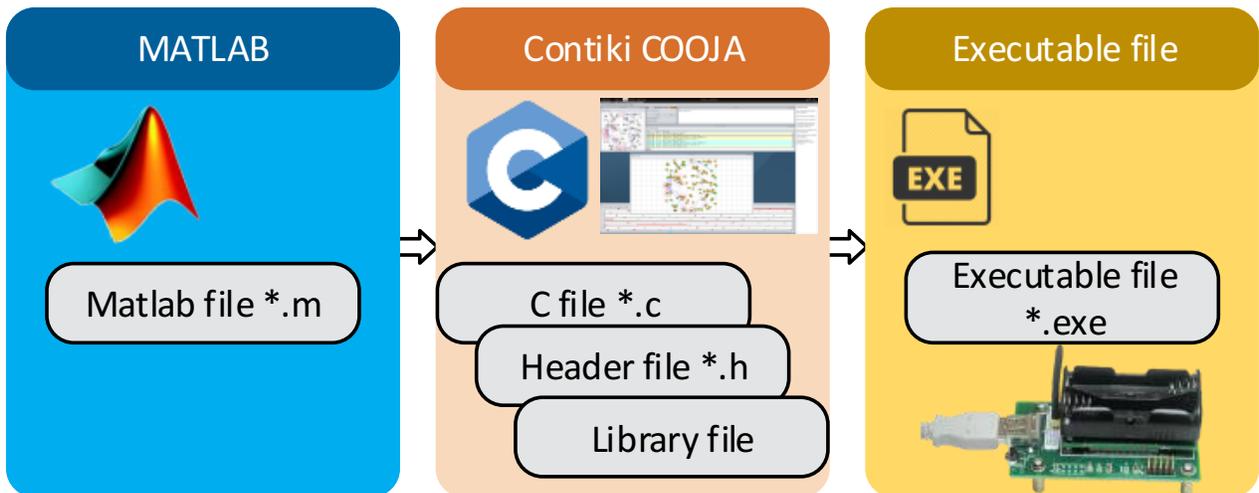


Figure 4.6 – Étapes d'implémentation.

Selon le type de fonctionnement des nœuds ancrés et inconnus, les deux algorithmes à implémenter peuvent être considérés comme des algorithmes composites:

- **Ancrés:** Chaque ancre fait; la diffusion d'une structure contenant son ID, sa position et un attribut *hop-count* initialisé à 1, stockage des coordonnées d'autres ancrés dans une BD et la participation à leurs inondation dans le réseau, calcul et diffusion de son *hop-size* (avec les équations; (3.1) pour DV-Hop et (3.10) pour IWDV-Hop) et la participation à l'inondation des *hop-size* d'autres ancrés, et l'acheminement des positions estimées par les nœuds inconnus à une station de base.

- **Nœuds inconnus:** Chaque nœud inconnu fait; la création d'une BD contenant les informations reçues sur chaque ancre (ID, (x,y), *hop-count*, *hop-size*), la participation à la diffusion de ces informations dans le réseau, et l'estimation de sa position (en utilisant les méthodes; ML pour DV-hop et hyperbolique pour IWDV-hop) afin de l'envoyer à la BS. De plus, pour IWDV-Hop, chaque nœud inconnu fait le calcul de poids de tout ancre en utilisant (3.11) et ses propres *hop-size* avec (3.12).

Les deux figures 4.7 et 4.8 suivantes présentent respectivement les deux organigrammes des nœuds ancrés et inconnus du DV-Hop.

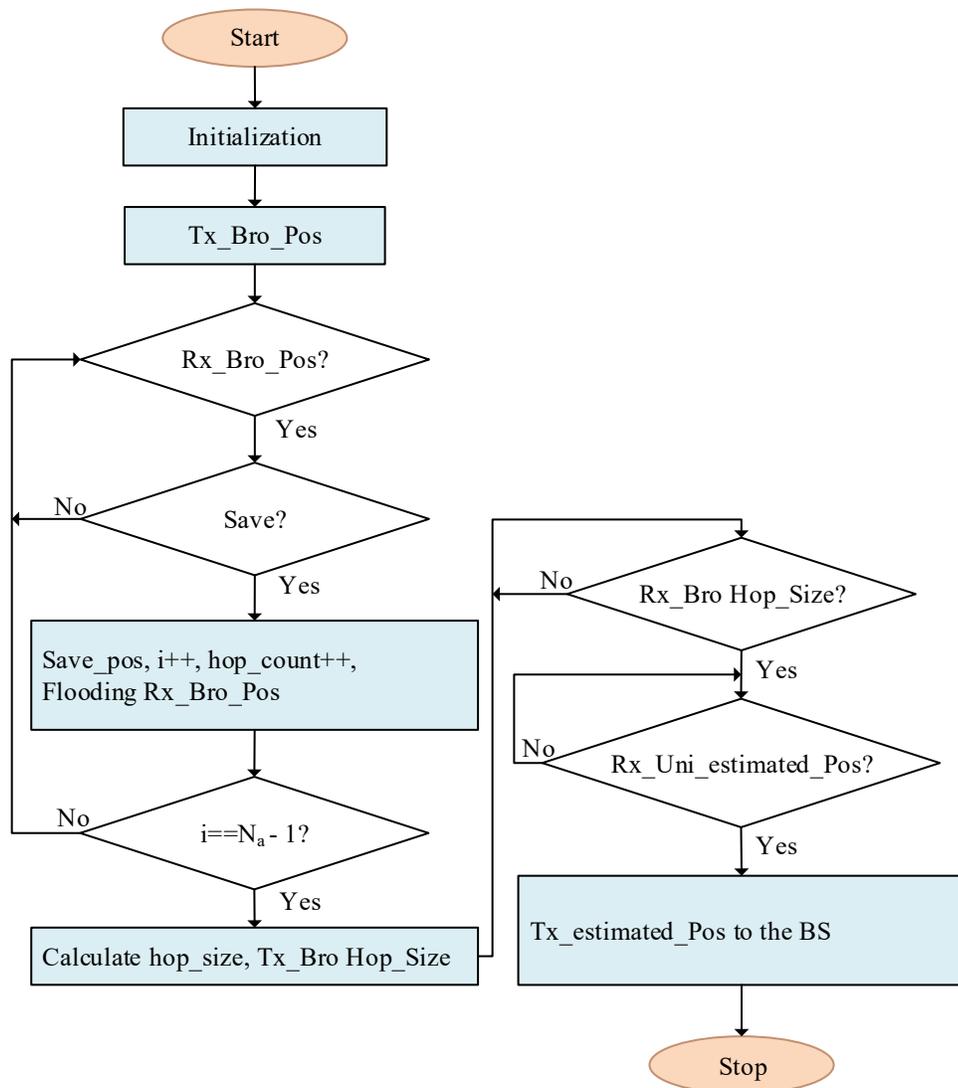


Figure 4.7 – Organigramme d'un nœud ancre.

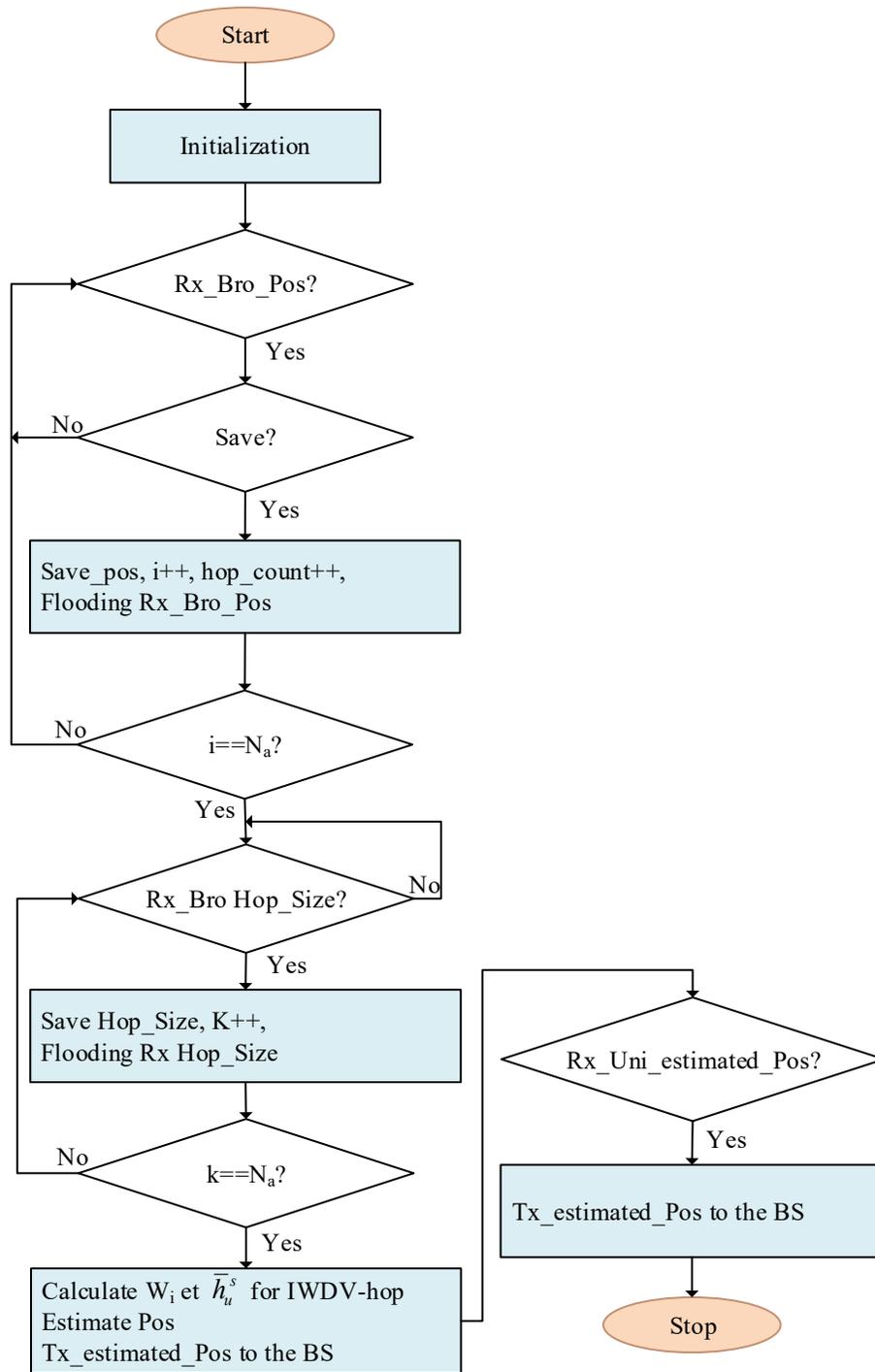


Figure 4.8 – Organigramme d'un nœud inconnu.

4.4 Validation des performances sur la plateforme MICAz

Après le développement d'algorithmes en langage C embarqué avec le système d'exploitation Contiki, le simulateur COOJA a été utilisé pour donner un aperçu des nœuds et les différents paquets qu'ils transmettent afin d'analyser le bon fonctionnement d'algorithmes. L'environnement de simulation COOJA permet de porter la simulation de manière réaliste sur les différents nœuds.

Nous évaluons, dans cette section, les performances de notre algorithme en effectuant une étude comparative des résultats de simulations générés par Matlab avec ceux obtenus par l'outil Contiki Cooja, et par des tests en conditions d'opération réelles sur les plateformes MICAz. Nous avons utilisés 20 plateformes MICAz dans un surface S de $(22 \times 22\text{cm}^2)$ en configurant la portée R de chaque nœud à 15 cm. Tous les résultats de simulations sont obtenus en procédant à 20 itérations.

La figure 4.9 suivante illustre une distribution uniforme de plateformes MICAz pour un test réel.

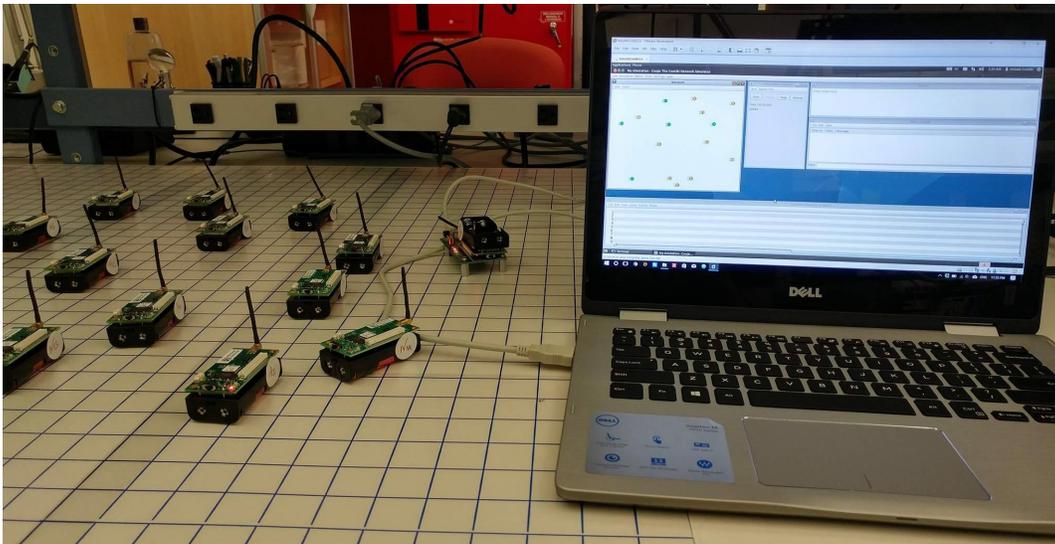


Figure 4.9 – Validation en conditions d'opération réelles.

Les paramètres de simulation sont donnés par le tableau 4.1 suivant:

Tableau 4.1 – Paramètres de simulation.

Paramètre	Valeur
MCU	ATmega128
Radio	CC2420
Protocole de communication	Rime
Taux de réussite TX/RX	100%
Portée de transmission R	15cm
Nombre de nœuds N	20
Surface de simulation S	$(22 \times 22cm^2)$

Les trois figures 4.10, 4.11 et 4.12 illustrent respectivement l'erreur quadratique moyenne normalisée de la localisation en fonction de nombre d'ancres, de la densité et de la portée. Dans le premier scénario, nous avons varié la densité d'ancres dans le réseau de 4 à 7 avec un nombre total de nœuds fixé à 20. Pour le deuxième scénario, nous avons aléatoirement distribué 5, 10, 15 puis 20 plateformes MICAz avec un nombre d'ancres allant de 4 à 7. Et finalement, nous avons fait varier la portée des nœuds de 10 à 16cm en fixant le nombre d'ancres à 5 et le nombre total des nœuds à 20.

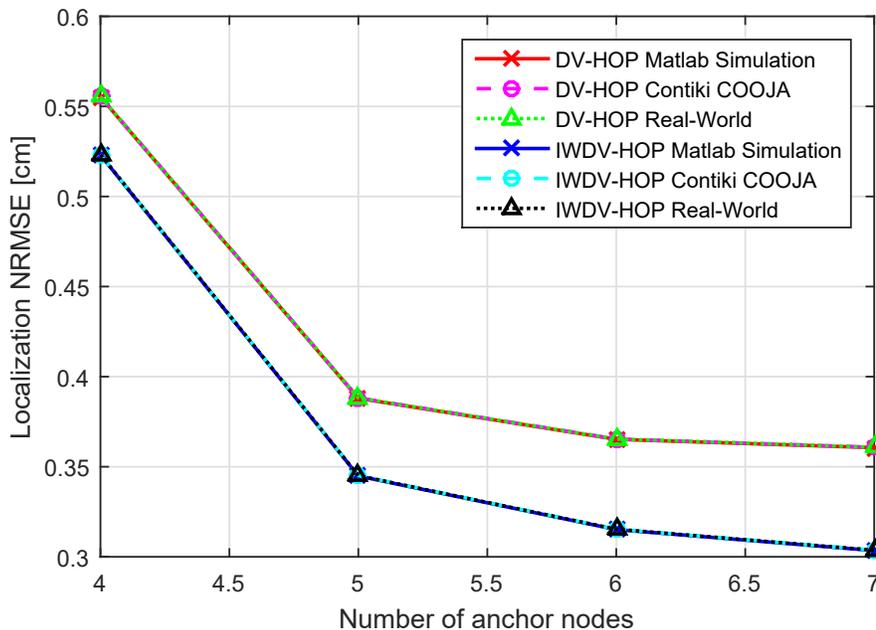


Figure 4.10 – Validation en conditions d'opération réelles du NRMSE de localisation en fonction de nombre d'ancres.

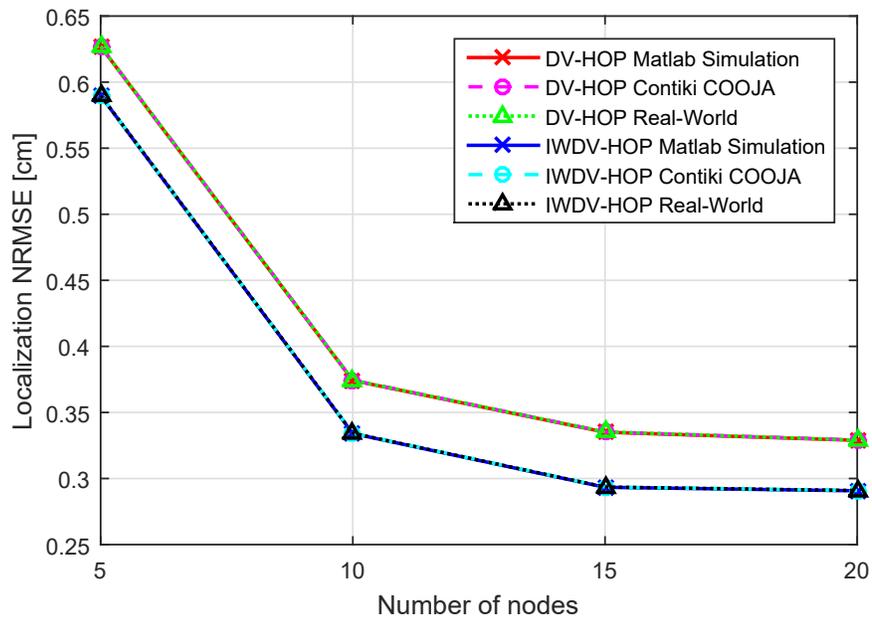


Figure 4.11 – Validation en conditions d'opération réelles du NRMSE de localisation en fonction de la densité.

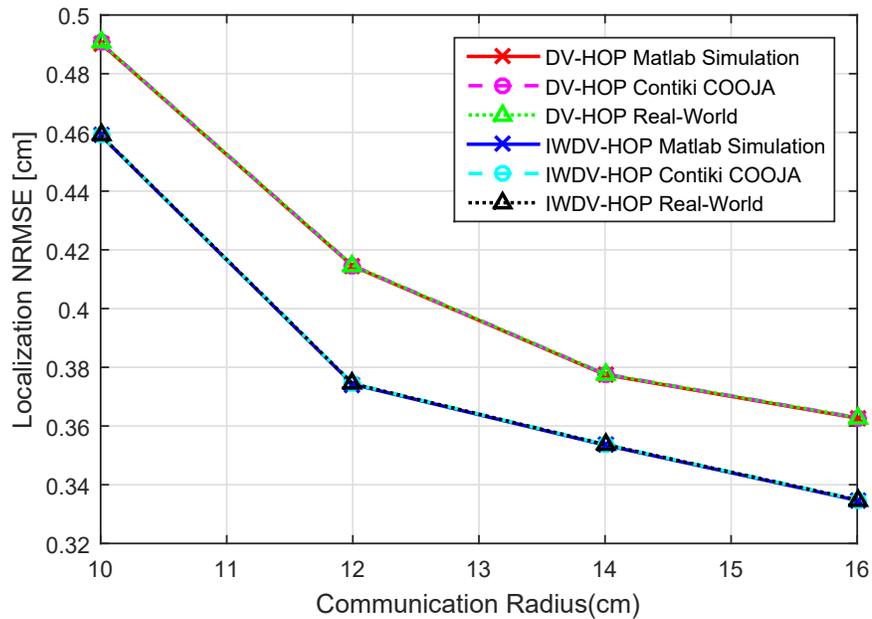


Figure 4.12 – Validation en conditions d'opération réelles du NRMSE de localisation en fonction de la portée.

Nous remarquons bien que les trois courbes obtenues par Matlab, Contiki Cooja et de tests réels sont pratiquement confondues et suivent la même tendance dans les trois figures. Ceci confirme et

valide l'amélioration de la performance de notre algorithme en terme de précision et ses comportements dans les différents scénarios.

4.5 Conclusion

Un élément central de ce travail est la comparaison des résultats expérimentaux sur des plateformes MICAz, en conditions d'opération réelles, avec ceux qui sont obtenus par Matlab.

Dans ce chapitre, nous avons présenté notre environnement de travail à savoir le système d'exploitation open-source Contiki, le simulateur Cooja, la plateforme MICAz et le module programmeur MIB520. Ensuite, nous avons décrit les étapes d'implémentation et le fonctionnement de nœuds ancrés et inconnus. Enfin, nous avons évalué et validé l'amélioration de la performance de notre nouvelle méthode de localisation.

Conclusion générale

Grâce à leurs importance et leurs utilisation à grande échelle dans plusieurs domaines, les réseaux de capteurs sans fil ont connu une évolution technologique ces dernières années. Et ils sont devenus une solution efficace pour plusieurs problèmes industriels et environnementaux. Cependant, nombreux défis majeurs demeurent à résoudre dans ces réseaux tel que la localisation des nœuds capteurs.

C'est dans ce contexte que le travail présenté dans ce mémoire s'inscrit. Il consiste à développer un nouvel algorithme de localisation en se basant sur la technique de DV-Hop afin d'améliorer sa précision d'estimation des positions de capteurs. Ainsi que la validation de cet algorithme sur des plateformes réelles.

Dans ce rapport, nous avons étudié les réseaux de capteurs sans fil d'une manière générale. Nous avons commencé par présenter l'état de l'art pour ce type de réseaux, et une description du principe de fonctionnement. Nous avons également présenté les domaines d'utilisation tels que la santé, le domaine militaire et environnemental, les grands défis qui influent sur la conception et la performance d'un réseau de capteurs sans fil. Puis nous avons mis l'accent sur l'ensemble des composants d'un système de localisation, les différentes méthodes d'estimation de distance entre deux nœuds capteurs et les techniques de calcul des positions. Ensuite, nous avons proposé notre nouvelle approche de localisation des capteurs. Les simulations Matlab de cette approche ont montré une amélioration de la performance en termes de précision par rapport à DV-Hop. Enfin, nous avons validé la précision de notre nouvel algorithme en conditions d'opérations réelles sur des plateformes MICAz.

Bien que les résultats de l'algorithme proposé représentent une bonne contribution au domaine de localisation pour les RCSFs, cette technique ne s'applique pas dans le cas des milieux anisotropiques.

En fait, il est fort probable que la communication via une visibilité indirecte (NLOS) entraîne une surestimation des distances entre les ancrés et les nœuds inconnus. Et par la suite, la localisation deviendra moins précise. Pour remédier à ce problème, plusieurs solutions, développées au sein du Wireless Lab, peuvent être intégrées dans ce travail. À titre d'exemple, nous pouvons citer la stratégie de sélection d'ancres fiables qui assure une estimation précise de la distance [29].

Références

- [1] Ian F Akyildiz, Weilian Su, Yogesh Sankarasubramaniam, and Erdal Cayirci. A survey on sensor networks. *IEEE Communications magazine*, 40(8):102–114, 2002.
- [2] David Culler, Deborah Estrin, and Mani Srivastava. Guest editors' introduction: Overview of sensor networks. *Computer*, 37(8):41–49, 2004.
- [3] Waltenegus Dargie and Christian Poellabauer. *Fundamentals of wireless sensor networks: theory and practice*. John Wiley & Sons, 2010.
- [4] Bhaskar Krishnamachari. *Networking wireless sensors*. Cambridge University Press, 2005.
- [5] Jennifer Yick, Biswanath Mukherjee, and Dipak Ghosal. Wireless sensor network survey. *Computer networks*, 52(12):2292–2330, 2008.
- [6] Neal Patwari, Joshua N Ash, Spyros Kyperountas, Alfred O Hero, Randolph L Moses, and Neiyer S Correal. Locating the nodes: cooperative localization in wireless sensor networks. *IEEE Signal processing magazine*, 22(4):54–69, 2005.
- [7] PK Singh, Bharat Tripathi, and Narendra Pal Singh. Node localization in wireless sensor networks. *PK Singh et al/(IJCSIT) International Journal of Computer Science and Information Technologies*, 2(6):2568–2572, 2011.
- [8] Nabil Ali Alrajeh, Maryam Bashir, and Bilal Shams. Localization techniques in wireless sensor networks. *International Journal of Distributed Sensor Networks*, 9(6):304628, 2013.
- [9] Ronald Beaubrun. Methods for node localization in wireless sensor networks. In *Mobile Ad Hoc and Sensor Systems (MASS), 2014 IEEE 11th International Conference on*, pages 521–522. IEEE, 2014.
- [10] Dragos Niculescu and Badri Nath. Ad hoc positioning system (aps). In *Global Telecommunications Conference, 2001. GLOBECOM'01*, volume 5, pages 2926–2931. IEEE, 2001.
- [11] Abdelaziz Samet Sofiene Affes Mohamed Guadane, Wassima Bchimi. Enhanced range-free localization in wireless sensor networks using a new weighted hop-size estimation technique. In *International Symposium on Personal, Indoor and Mobile Radio Communications. IEEE PIMRC 2017*. IEEE, 2017.
- [12] Abdelali Hadir, Khalid Zine-Dine, Mohamed Bakhouya, and Jamal El Kafi. An optimized dv-hop localization algorithm using average hop weighted mean in wsns. In *Codes, Cryptography and Communication Systems (WCCCS), 2014 5th Workshop on*, pages 25–29. IEEE, 2014.

- [13] Anthony LaMarca and Eyal De Lara. Location systems: An introduction to the technology behind location awareness. *Synthesis Lectures on Mobile and Pervasive Computing*, 3(1):1–122, 2008.
- [14] U Nazir, N Shahid, MA Arshad, and SH Raza. Classification of localization algorithms for wireless sensor network: A survey. In *Open source systems and technologies (ICOSST), 2012*, pages 1–5. IEEE, 2012.
- [15] Guoqiang Mao, Barış Fidan, and Brian DO Anderson. Wireless sensor network localization techniques. *Computer networks*, 51(10):2529–2553, 2007.
- [16] Peng Rong and Mihail L Sichitiu. Angle of arrival localization for wireless sensor networks. In *Sensor and Ad Hoc Communications and Networks, 2006. SECON'06. 2006*, volume 1, pages 374–382. IEEE, 2006.
- [17] Kai Chen, Zhong-hua Wang, Mei Lin, and Min Yu. An improved dv-hop localization algorithm for wireless sensor networks. In *Industrial Electronics and Applications, 2008. ICIEA 2008*, pages 1557–1561. IEEE, 2008.
- [18] Jian Li, Jianmin Zhang, and Liu Xiande. A weighted dv-hop localization scheme for wireless sensor networks. In *Scalable Computing and Communications; Eighth International Conference on Embedded Computing, 2009. SCALCOM-EMBEDDEDCOM'09*, pages 269–272. IEEE, 2009.
- [19] Xiao Chen and Benliang Zhang. Improved dv-hop node localization algorithm in wireless sensor networks. *International Journal of Distributed Sensor Networks*, 8(8):213980, 2012.
- [20] Yiu-Tong Chan and KC Ho. A simple and efficient estimator for hyperbolic location. *IEEE Transactions on signal processing*, 42(8):1905–1915, 1994.
- [21] Hyuk Lim and Jennifer C Hou. Localization for anisotropic sensor networks. In *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE*, volume 1, pages 138–149. IEEE, 2005.
- [22] Muhammad Omer Farooq and Thomas Kunz. Operating systems for wireless sensor networks: A survey. *Sensors*, 11(6):5900–5930, 2011.
- [23] Shah Bhatti, James Carlson, Hui Dai, Jing Deng, Jeff Rose, Anmol Sheth, Brian Shucker, Charles Gruenwald, Adam Torgerson, and Richard Han. Mantis os: An embedded multithreaded operating system for wireless micro sensor platforms. *Mobile Networks and Applications*, 10(4):563–579, 2005.
- [24] Philip Levis, Sam Madden, Joseph Polastre, Robert Szewczyk, Kamin Whitehouse, Alec Woo, David Gay, Jason Hill, Matt Welsh, Eric Brewer, et al. Tinyos: An operating system for sensor networks. *Ambient intelligence*, 35:115–148, 2005.
- [25] Adam Dunkels, Bjorn Gronvall, and Thiemo Voigt. Contiki-a lightweight and flexible operating system for tiny networked sensors. In *Local Computer Networks, 2004. 29th Annual IEEE International Conference on*, pages 455–462. IEEE, 2004.
- [26] Alok Ranjan, HB Sahu, and Prasant Misra. A survey report on operating systems for tiny networked sensors. *arXiv preprint arXiv:1505.05269*, 2015.

- [27] Adam Dunkels. Rime-a lightweight layered communication stack for sensor networks. In *Proceedings of the European Conference on Wireless Sensor Networks (EWSN), Poster/Demo session, Delft, The Netherlands*. Citeseer, 2007.
- [28] Joakim Eriksson, Fredrik Österlind, Niclas Finne, Nicolas Tsiftes, Adam Dunkels, Thiemo Voigt, Robert Sauter, and Pedro José Marrón. Cooja/mspsim: interoperability testing for wireless sensor networks. In *Proceedings of the 2nd International Conference on Simulation Tools and Techniques*, page 27. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2009.
- [29] Ahmad El Assaf, Slim Zaidi, Sofiene Affes, and Nahi Kandil. Range-free localization algorithm for anisotropic wireless sensor networks. In *Vehicular Technology Conference (VTC Fall), 2014 IEEE 80th*, pages 1–5. IEEE, 2014.