

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université A/Mira de Béjaïa
Faculté des Sciences Exactes
Département d'Informatique



Mémoire de Master Professionnel

en Informatique

Option

Administration et Sécurité des réseaux/ Génie Logiciel

Thème

**Simulation d'un Réseau de capteur sans fil sous contiki "La
localisation d'un mobile"**

Présenté par

M^{elle} **BOUAMARA** Ouarda

M^{elle} **BOUAMARA** Rosa

Soutenu devant le jury composé de :

Présidente M^{elle} **ZIDANI** Ferroudja

Examinatrice M^{elle} **BOUADEM** Nassima

Encadreur M^r **ACHROUFENE** Achour

Promotion 2015/2016

Remerciements

Nous tenons à remercier tout d'abord ALLAH qui nous a donné toutes ces années, la santé, le courage, la volonté pour réaliser ce travail. Le plus grand merci lui revient de nous avoir guidées vers le droit chemin de nous avoir aidées tout le long de nos années d'études et pour toutes ses grâces qui nous entourent. Ah! les remerciements! Nous remercions vivement notre encadreur Monsieur ACHROUFENE Achour pour avoir accepté de nous encadrer et qui a toujours été à notre écoute et disponible tout au long de notre travail malgré ses charges ainsi que pour son aide et le temps qu'il a bien voulu nous consacré sans quoi ce mémoire n'aurait jamais vu le jour.

Nous tenons à remercier les membres du jury qui ont bien voulu accepter de juger notre modeste travail.

Nos remerciements vont également à nos parents qui nous ont encouragés, qui nous ont appris à travailler honnêtement et qui ont toujours supporté moralement et financièrement pendant toutes nos longues années d'études, Nous adressons également nos sincères remerciements à nos frères, soeurs.

A la fin nous tenons à remercier tous nos collègues d'étude, particulièrement notre promotion et pour tous ceux qui ne sont pas cités, qui sont si nombreux, on vous remercie de plus profond de nous-mêmes.

Dédicaces

À mes chères parents, mon très cher père pour sa patience et tous ses efforts et à ma mère
pour m'avoir encouragé à continuer mes études.

À ma soeure Fahima.

À toute ma famille : mes grand parents, mes frères, mes belles soeurs, mes cousin et cousines.

À ma binôme Rosa avec qui j'ai partagé de belles années d'études.

À tous mes professeurs ainsi que tous les étudiants de la promotion ASR, GL 2016.

À tous mes meilleurs amis, dont la liste est longue.

À tous ceux que j'aime.....

Ouarda

À mes parents. Tout au long de mon cursus, ils m'ont toujours soutenu, encouragé et aidé. Ils
ont su me donner toutes les chances pour réussir.

À ma famille et en particulier mes frères et soeurs pour m'avoir fait partager leur joie de vivre
et m'avoir ainsi soutenu dans mes efforts.

À toute ma famille : mes grand mère, mes belles soeurs, mes beau frères, mes petit neveux
adorés " Samy, Anya, Clara et Elina" mes cousin et cousines.

À ma binôme Ouarda avec qui j'ai partagé de belles années d'études.

À tous mes professeurs ainsi que tous les étudiants de la promotion ASR, GL 2016.

À tous mes meilleurs amis, dont la liste est longue.

À tous ceux que j'aime.....

Rosa

Liste des Abréviations

ADC	A nalog D igital C onverter
AoA	A ngle of A rrival
CSMA/CA	C arrier S ense M ultiple A ccess with C ollision A voidance
DV-Hop	D istance V ector- H op
GPS	G lobal P ositionning S ystème
IEEE	I nstitut of E lectrical and E lectronics E ngineers
ISO	O rganisation I nternationale de N ormalisation
LAN	L ocal A rea N etwork
LRPUWB	L ow R ate P ulse U ltra W ide B and
MAC	M edium A ccess C ontrol
MAN	M etropolitan A rea N etwork
NesC	N etwork E mbedded S ystem C
PAN	P ersonal A rea N etwork
RAM	R andom A ccess M emory
RCSF	R éseau de C apteurs S ans F il
RF	R adio F réquence
RFD	R educed F unction D evice
RSSI	R eceived S ignal S trength I ndicator
TDoA	T ime D ifference of A rrival
ToA	T ime of A rrival
UCT	U nité C entrale de T raitement
UDGM	U nit D isk G raph M edium
USA	U nité C entrale de T raitement
UWB	U ltra- W ide B and
WAN	W ide A rea N etwork
WI-FI	W ireless F idelity

WLAN	W ireless L ocal A rea N etwork
WMAN	W ireless M étropolitain A rea N etwork
WPAN	W ireless P ersonal A rea N etwork
WRAN	W ireless R egional A rea N etwork

TABLE DES MATIÈRES

Dédicaces	2
Liste des Abréviations	i
Table des Matières	iii
Liste des tableaux	vii
Table des figures	vi
Introduction générale	1
1 Généralités sur les réseaux de capteurs sans fil	3
1.1 Introduction	3
1.2 Capteurs sans fil	3
1.2.1 Qu'est-ce qu'un capteur ?	3
1.2.2 Architecture d'un noeud capteur sans fil	4
1.2.2.1 Architecture matérielle	4
1.2.2.2 Architecture logicielle	6
1.3 Réseaux de capteurs sans fil (RCSF)	6
1.3.1 Qu'est-ce qu'un réseau de capteurs sans fil ?	6
1.3.2 Architecture des RCSF	6
1.3.3 Différentes topologies dans les RCSF	7
1.3.4 La durée de vie d'un RCSF	9
1.3.5 Caractéristiques des réseaux de capteurs sans fil	9
1.3.6 Systèmes d'exploitation pour les RCSF	10
1.3.7 Domaines d'application	11

1.4	Conclusion	13
2	Localisation dans les RCSF	14
2.1	Introduction	14
2.2	Problématique de la localisation dans les RCSF	14
2.3	Localisation dans les RCSF	15
2.3.1	Technologies de mesure	15
2.3.2	Méthode de calcul de positions	20
2.3.3	Algorithmes de localisations	23
2.3.4	Forme d'implémentation	24
2.3.4.1	Les méthodes centralisées	24
2.3.4.2	Les méthodes distribuées	25
2.4	conclusion	25
3	Implémentation et Simulation	26
3.1	Introduction	26
3.2	Problématique	27
3.3	Objectifs de notre travail	28
3.4	Système de localisation proposé	28
3.5	Outils d'implémentation et de simulation	29
3.5.1	Le choix de Contiki	29
3.5.2	Le simulateur Cooja	29
3.5.3	Contraintes rencontrées dans cooja	31
3.6	Simulation	31
3.6.1	Scénarion de simulation	31
3.6.2	Résultats de simulation	31
3.6.3	Analyse des résultats	36
3.6.4	Interprétation et évaluation des résultats	39
3.7	Amélioration du système de localisation proposé	40
3.7.1	Solution proposée	40
3.7.2	Rappel sur CSMA/CA-adapté	40
3.7.3	Implémentation des méthodes	41
3.7.4	Méthode Centroïde	42
3.7.5	Méthode MinMax	42
3.7.6	Algorithmes des programmes à mettre dans les noeuds	45
3.8	Simulation du système de localisation après l'amélioration	46
3.9	Comparaison des résultats	49

3.9.1	Evaluation et interprétation	49
3.10	Conclusion	51
	Conclusion générale	52
	Bibliographie	viii
	A <i>Annexe</i>	x
	Résumé	xiii

LISTE DES TABLEAUX

1.1	Principales différentes entre les quatre systèmes d'exploitation.	11
2.1	liste des caractéristiques essentielles des méthodes décrites dans les paragraphes précédents.	19
2.2	les principales caractéristiques des méthodes de calcul de position.	23
3.1	Les résultats de simulation avec la méthode centroide	33
3.2	Les résultats de simulation avec la méthode MinMax	35
3.3	Tableaux représentant les mesures RSSI sur différentes distances	43
3.4	Les résultats de simulation avec la méthode centroid après l'amélioration	47
3.5	Les résultats de simulation avec la méthode MinMax après la simulation	48

TABLE DES FIGURES

1.1	Noeud capteur MicaZ.	4
1.2	Architecture d'un capteur sans fil.	4
1.3	Architecture d'un réseau de capteurs sans fil.	6
1.4	Topologie en étoile.	8
1.5	Topologie maillée.	8
1.6	Topologie cluster-tree.	9
1.7	Quelques domaines d'applications des RCSF.	12
2.1	Composition d'un système de localisation.	16
2.2	Principe de l'utilisation des RSSI.	16
2.3	Principe de ToA.	17
2.4	Principe de AOA.	19
2.5	Principe de Trilateration.	20
2.6	Angle d'incidence et position.	21
2.7	Principe de MinMax	21
2.8	Principe de la méthode Centroïde	22
3.1	Interface graphique du le simulateur Cooja.	30
3.2	La suite des messages recus par le mobile	32
3.3	Comparaison d'erreur de précision entre Controïde et MinMax pour trois capteurs.	36
3.4	Comparaison d'erreur de précision entre Controïde et MinMax pour cinq capteurs.	37
3.5	Comparaison d'erreur de précision entre Controïde et MinMax pour dix capteurs.	38
3.6	Comparaison d'erreur de précision entre Controïde et MinMax pour vingt capteurs.	39
3.7	La durée d_t	41
3.8	Les mesures RSSI sur différentes distances	43

3.9	Tableau comparative de taux d'erreu des deux méthode avant et après l'amélioration.	49
3.10	Différence des taux d'erreur avant et après l'amélioration avec MinMax	50
3.11	Différence des taux d'erreur avant et après l'amélioration avec Centroide	50
A.1	l'exécution du programme hello-world sous Contiki.	xii

Depuis une dizaine d'années, le besoin de contrôler son environnement, que soit à l'intérieur ou à l'extérieur de la maison, a donné naissance aux réseaux de capteurs sans fil (RCSF). Ces derniers permettent : à l'homme d'accéder à distance et de façon automatique aux informations telles que la température ou le taux d'humidité de sa maison ; à une unité de pompiers de surveiller une zone forestière et ainsi de prévenir les feux de forêt ; à un médecin de suivre l'évolution du rythme cardiaque de son patient.

La tâche première d'un capteur est de détecter un événement (par exemple, un changement de température, des mouvements, des vibrations, ...). Il est donc capable de récolter des données relatives à son environnement, de les traiter, puis, si nécessaire, de les communiquer à des capteurs voisins via un médium sans fil. Le déploiement de ce type d'appareils forme alors un réseau qui peut être utilisé dans des domaines différents. Les problématiques qu'engendrent les réseaux de capteurs sans fil n'en sont pas moins nombreuses. Nous pouvons citer le problème du routage qui consiste à acheminer un message d'un capteur vers un autre et celui de la localisation qui cherche à attribuer une position géographique exacte ou estimée à un capteur. L'objectif principal de notre travail s'insère dans cette dernière problématique. Ainsi, nous allons étudier le fonctionnement des systèmes de localisation dans les réseaux de capteurs pour contribuer à l'amélioration des systèmes existants. Ce genre de système cherche à estimer les positions des mobiles qui se déplacent dans un environnement dans lequel est déployé un réseau de capteurs sans fil. Les informations sur la position de la cible mobile doivent être mises à jour régulièrement à des intervalles de temps bien précis, ce qui signifie que la contrainte du temps est aussi importante que la précision de la localisation.

Généralement les systèmes de localisation sont composés de trois éléments principaux qui sont : a) les technologies utilisées pour la communication telles que WIFI, ZigBee, b) les mé-

triques de mesures de distance telles que la puissance du signal reçu ou le temps du vol du signal et c) les de méthodes de localisation qui calculent la position à partir des distances mesurées auparavant comme la trilatération et MinMax. Dans ce travail, nous allons proposer un système de localisation qui utilise la technologie Zigbee pour la communication et la puissance du signal reçu pour l'estimation de la distance entre deux noeuds, ainsi que les méthodes de localisation MinMax ou Centroïde pour calculer la position du noeud inconnu. Afin d'implémenter et de valider le système proposé, nous utilisons le système d'exploitation Contiki dédié aux réseaux de capteurs et le simulateur Cooja intégré dedans. Ce mémoire est organisé en quatre chapitres :

Le premier chapitre donne une description globale des réseaux de capteurs sans fil. Il présente également les architectures et les systèmes d'exploitation dédiés aux réseaux de capteurs ainsi que leurs différents domaines d'applications.

Le deuxième chapitre porte sur la problématique de la localisation dans les réseaux de capteurs. Ce chapitre détaille les composantes d'un système de localisation à savoir les technologies utilisées, les métriques ainsi que les méthodes de localisation.

Le troisième chapitre constitue le coeur de notre travail. Dans ce chapitre nous présentons l'objectif de notre travail, notre problématique, les outils matériels et logiciels nécessaires à l'implémentation et la simulation de nos propositions. Ces dernières consistent en l'amélioration de la précision de la localisation dans les réseaux de capteurs en prenant en compte la contrainte du temps. A la fin de ce chapitre, nous donnons une comparaison des résultats obtenus avant et après l'amélioration du système de localisation proposé.

Nous terminons ce mémoire par une conclusion qui résume l'essentiel de notre travail et quelques perspectives.

CHAPITRE 1

Généralités sur les réseaux de capteurs sans fil

1.1 Introduction

Les progrès récents dans la technologie des systèmes micro-électromécaniques, les communication sans fil, et l'électronique numérique ont permis le développement de petits dispositifs peu coûteux, de faible puissance, qui peuvent communiquer entre eux, appelés capteurs. Ces derniers communiquent entre eux via une communication sans fil pour le partage d'information et le traitement coopératif, ils sont déployés aléatoirement dans une zone d'intérêt pour superviser ou surveiller des phénomènes divers (température, humidité, vibration, luminosité, ...etc.). Ces dispositifs coopèrent entre eux pour former une infrastructure de communication appelée réseau de capteurs.

Ce chapitre a pour objectif de décrire les caractéristiques générales des noeuds et des réseaux de capteurs sans fil, nous aborderons les notions d'architectures et de topologies, leurs systèmes d'exploitations ainsi que leurs domaines d'applications variés. Finalement, ce chapitre s'achève par une conclusion.

1.2 Capteurs sans fil

1.2.1 Qu'est-ce qu'un capteur ?

Un noeud capteur est un dispositif qui transforme une grandeur physique observée (température, pression,..) en une grandeur utilisable [1] et de la communiquer à un centre de contrôle via une station de base. Parmi les différents types de mesures enregistrées par les noeuds capteurs, on peut citer entre autres : la température, l'humidité, la luminosité, l'accélération, la distance, les mouvements, la positions, la pression, la présence du gaz, la vision (capture d'image), le

son, etc. La figure 1.1 montre un exemple de noeud capteur MicaZ [2] :



FIG. 1.1 – Noeud capteur MicaZ.

Les noeuds capteurs déployés en grand nombre, sont capables de récolter et de transmettre d'une manière autonome des données de leur environnement immédiat. La localisation géographique de ces noeuds dans le réseau n'est pas obligatoirement prédéterminée [3]. Ils sont déployés dans une zone géographique appelée champ de capture, définissant le terrain d'intérêt pour le phénomène capturé.

1.2.2 Architecture d'un noeud capteur sans fil

Dans cette section, nous distinguons les deux parties qui composent un capteur :

1.2.2.1 Architecture matérielle

Suivant les besoins, il existe un grand nombre des noeuds capteurs, avec des fonctionnalités diverses et variées, les noeuds capteurs de température, d'humidité, de pression, etc. Cependant, malgré cette diversité apparente, ils restent dotés d'une architecture similaire. Des composants additionnels peuvent être ajoutés selon le domaine d'application. La figure 1.2 représente l'architecture d'un capteur sans fil [4] :

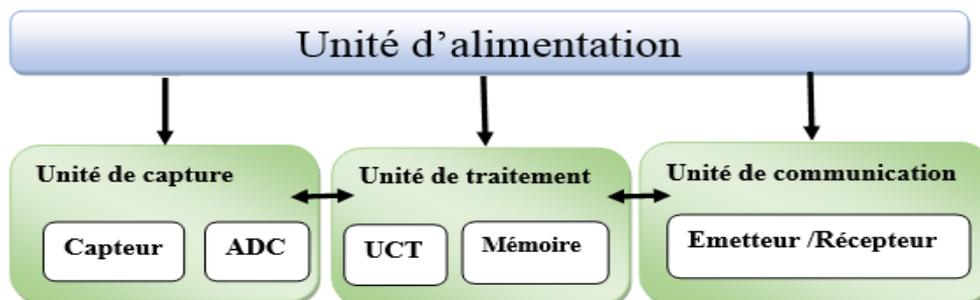


FIG. 1.2 – Architecture d'un capteur sans fil.

Un capteur est composé principalement des unités de : captage, traitement, communication et d'énergie.

- **Unité de capture d'informations** : elle est composée du capteur proprement dit et du convertisseur Analogique/Numérique (ADC : Analog/Digital Converter) qui transforme les signaux analogiques en signaux numériques[5]. En effet, le capteur observe un phénomène et fournit les signaux analogiques correspondant au phénomène au convertisseur analogique /numérique, le ADC, qui les transforme en signaux numériques compréhensible par l'unité de traitement.
- **Unité de traitement d'information (processeur)** : cette unité constitue l'élément central d'un noeud capteur UCT (Unité central de traitement), elle est composée d'un processeur et d'une mémoire intégrant un système d'exploitation spécialement conçu pour les capteurs. Cette unité possède deux interfaces, une interface pour l'unité d'acquisition et une interface pour l'unité de communication[5]. Cette unité est chargée aussi d'exécuter les protocoles de communication qui permettent de faire collaborer le noeud capteur avec d'autres noeuds, comme elle peut aussi analyser les données captées [6].
- **Unité de communication** : cette unité est responsable de toutes les émissions et les réceptions des données via un support de communication sans fil [7]. Les différents choix de média de transmission incluent la Radio Fréquence(RF), le laser et l'infrarouge.
- **Unité d'alimentation** : c'est un élément primordial de l'architecture d'un noeud capteur, c'est elle qui fournit l'énergie à toutes les autres unités, elle correspond le plus souvent à une batterie ou une pile alimentant le noeud capteur, dont les ressources limitées en font une problématique propre à ce type de réseau puisque ces derniers sont généralement déployés dans des zones non accessibles [8]. La réalisation récente d'unité d'alimentation à base de panneaux solaires tente d'apporter une solution pour prolonger sa durée de vie.

Par ailleurs, un noeud capteur peut être doté d'autres unités. Citons, entre autres, la possibilité d'ajouter une unité de localisation, tel qu'un GPS (Global Positionning Système), une unité de mobilité pour assurer la mobilité du noeud capteur, ou une unité spécifique de capture comme une caméra pour l'acquisition vidéo.

1.2.2.2 Architecture logicielle

Des systèmes d'exploitation spécifiques ont été conçus pour les réseaux de capteur sans fil car un capteur n'a pas assez de mémoire pour supporter un système d'exploitation comme Linux ou Windows qui prennent beaucoup d'espace mémoire et la contrainte énergétique des noeuds capteurs exige l'utilisation de systèmes d'exploitation légers, tels que TinyOS [9] ou Contiki [10].

1.3 Réseaux de capteurs sans fil (RCSF)

1.3.1 Qu'est-ce qu'un réseau de capteurs sans fil ?

Un Réseau de Capteurs Sans Fil (RCSF) est composé d'un ensemble de dispositifs très petits, només noeuds capteurs, interconnectés entre eux, variant de quelques dizaines d'éléments à plusieurs milliers [11]. Dans ces réseaux, chaque noeud est capable de surveiller son environnement et de réagir en cas de besoin en envoyant l'information collectée à un ou plusieurs points de collecte, à l'aide d'une connexion sans fil.

1.3.2 Architecture des RCSF

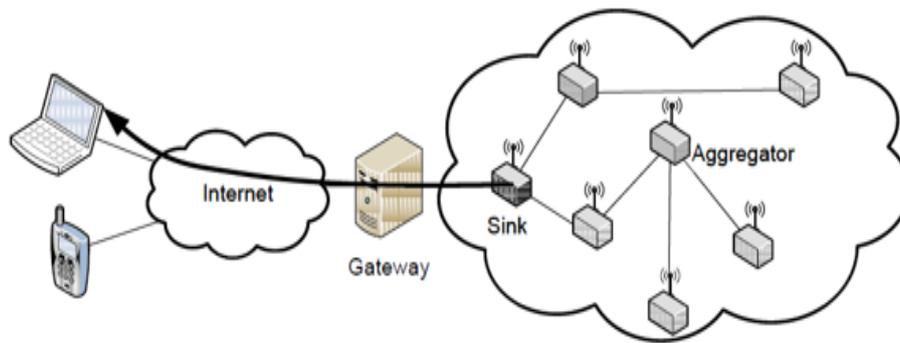


FIG. 1.3 – Architecture d'un réseau de capteurs sans fil.

La Figure 1.3 représente l'architecture habituelle des réseaux de capteurs sans fils. Ils sont construits autour des quatre principales entités suivantes [12] :

- **Le capteur (sensor)** : comme le dit bien son nom, il est en charge de mesurer une valeur relative à son environnement.
- **L'agrégateur (aggregator)** : il est en charge d'agréger les messages qu'il reçoit de plusieurs capteurs puis de les envoyer en un seul message au puits (sink). Cette opération a pour principal but de limiter le trafic sur le réseau et donc de prolonger la durée de vie globale du réseau de capteur.
- **Le puits (sink)** : le puits est le noeud final du réseau. C'est à lui qui fait envoi de l'ensemble des valeurs mesurées par le réseau. Il peut arriver qu'il y'ait plusieurs puits sur un même réseau de capteurs.
- **La passerelle (gateway)** : la passerelle est un dispositif qui a la particularité d'avoir deux interfaces réseau. Il permet de relier le réseau de capteurs sans fils à un réseau plus traditionnel, typiquement l'internet. En effet, habituellement le réseau de capteurs ne sert qu'à faire remonter les mesures, les applications traitant ces informations étant exécutées sur la machine de l'utilisateur final.

1.3.3 Différentes topologies dans les RCSF

La Topologie détermine l'organisation des noeuds capteurs dans le réseau. Il existe en général trois types de topologies [13] : la topologie en étoile, la topologie maillée et la topologie cluster-tree.

- **Topologie en étoile** : la topologie en étoile est composée d'un noeud central appelé coordinateur ou sink du réseau de capteurs et d'une pluralité des noeuds capteurs sans fil (voir Figure 1.4). Dans cette topologie, tous les noeuds capteurs transmettent leurs données directement vers le coordinateur [13]. Ainsi, l'architecture d'une telle topologie est caractérisée par une transmission à un saut. Dans cette topologie, le noeud central a la responsabilité de contrôler et de coordonner les noeuds capteurs qui communiquent exclusivement avec lui. L'intérêt de cette architecture est qu'elle limite au maximum le transport des données est donc très performante, en plus d'être relativement simple.

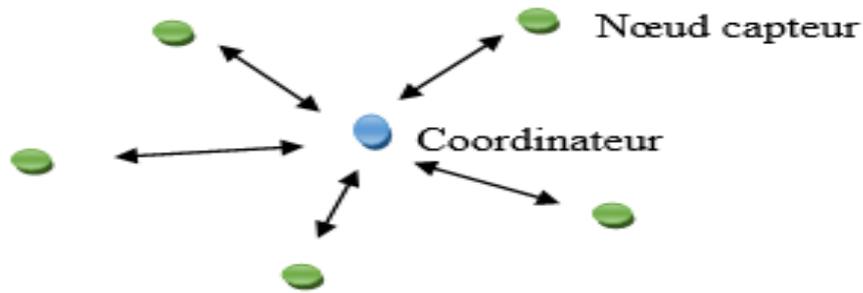


FIG. 1.4 – Topologie en étoile.

- **Topologie maillée** : dans les réseaux de capteurs mobiles classiques, la topologie maillée est la topologie standard. C'est un réseau distribué sans infrastructure, les nœuds peuvent communiquer entre eux sans contrôle centralisé ni point d'accès [13]. La topologie maillée (voir Figure 1.5) est constituée d'un ensemble de nœuds reliés les uns aux autres et agissant comme des routeurs. Chacun des nœuds transmet ses données à son voisin après avoir établi le meilleur chemin pour véhiculer les données à travers le réseau jusqu'au puits (coordinateur ou sink), cette caractéristique permet de maintenir le fonctionnement du réseau en cas de panne d'un nœud[13]. Cette topologie est alors plus robuste et flexible par rapport à la variation du nombre de nœuds. Quand il n'y a pas de transfert de données, les nœuds se mettent en mode veille pour économiser de l'énergie .

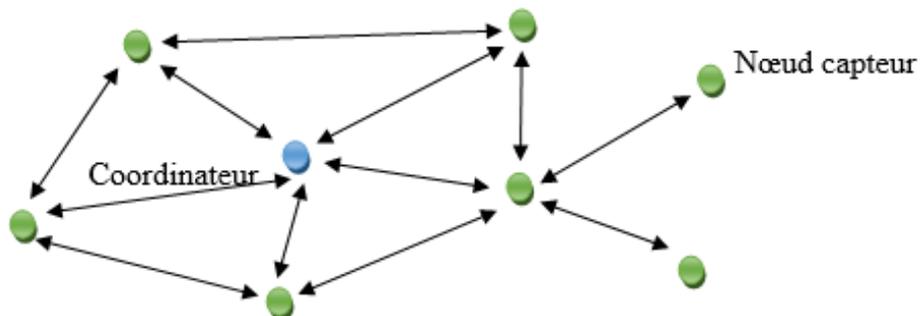


FIG. 1.5 – Topologie maillée.

- **Topologie arbre et cluster-tree** : la topologie cluster-tree (voir Figure 1.6) est partitionnée en groupes appelés "clusters". Un cluster est constitué d'un nœud particulier appelé "cluster-head" ou "tête de cluster" et d'autres nœuds qui communiquent qu'avec leur "tête de cluster", ensuite ce dernier est chargé de faire suivre les messages reçus vers le puits du réseau [13] . Cette topologie est alors hiérarchisée selon le rôle des éléments

du réseau (coordinateur, têtes de cluster, noeud). Il peut y avoir un changement de hiérarchie entre les noeuds et les têtes de cluster en fonction de l'énergie disponible ou la position géographique de chaque noeud.

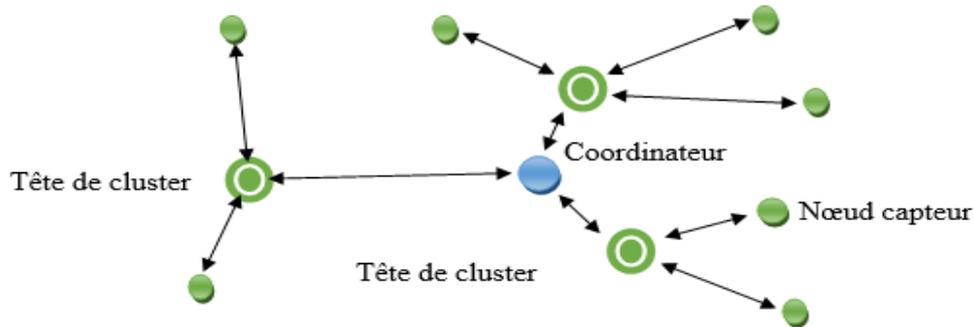


FIG. 1.6 – Topologie cluster-tree.

1.3.4 La durée de vie d'un RCSF

La durée de vie d'un réseau de capteurs est généralement définie par le temps durant lequel le réseau soient capable de maintenir assez de connectivité, couvrir la zone de captage ou garder son taux de perte de noeud inférieur à un certain niveau [14]. Il peut y avoir d'autres définitions de la durée de vie du réseau liées à d'autres paramètres du réseau où la durée de vie d'un réseau de capteurs est liée à la durée de vie des noeuds, et celle d'un noeud dépend principalement de la durée de vie de sa batterie. Cette dernière dépend des énergies consommées par les différents modules qui composent le noeud.

1.3.5 Caractéristiques des réseaux de capteurs sans fil

Un réseau de capteurs présente les caractéristiques suivantes [15] :

- Absence d'infrastructure : les réseaux Ad-hoc en général, et les réseaux de capteurs en particulier se distinguent des autres réseaux par la propriété d'absence d'infrastructure préexistante et de tout genre d'administration centralisée.
- Bande passante limitée : une des caractéristiques primordiales des réseaux basés sur la communication sans fil est l'utilisation d'un médium de communication partagé. Ce partage fait que la bande passante réservée à un noeud est limitée.
- Accès sans fil : les interférences sont inévitables (liens radios perturbés dans un hôpital).
- Gestion d'énergie (alimentation par batterie).
- les noeuds sont déployés en grand nombre.

- Les réseaux de capteurs sont non fiables : en tout moment, les capteurs peuvent être défaillants.
- La topologie des réseaux de capteurs change très fréquemment (topologie dynamique).
- Les réseaux de capteurs utilisent principalement le paradigme de communication broadcast.
- Les capteurs sont limités : en énergie (Piles AAA), capacités de calcul (4MHz), et en mémoire.

1.3.6 Systèmes d'exploitation pour les RCSF

Le système d'exploitation a pour mission de jouer le rôle de l'intermédiaire entre l'utilisateur et les périphériques matériels, et permet d'assurer les services basiques suivant : l'abstraction de la partie matérielle, la gestion des tâches, la gestion de la mémoire, la gestion de puissance et la gestion des périphériques. Parmi les systèmes d'exploitation développés pour les réseaux de capteurs sans fil, nous citons :

Contiki [16] : système d'exploitation open-source multi-tâche, s'appuie sur un fonctionnement événementiel, nécessite moins de mémoire RAM pour fonctionner, développé pour les systèmes embarqués avec contraintes de mémoire.

LiteOS [17] : système d'exploitation multi-thread, assure la gestion de mémoire dynamique et un interpréteur de commandes interactives.

TinyOS [18] : système principalement développé et soutenu par l'université américaine de Berkeley. C'est un système d'exploitation open-source conçu pour des réseaux de capteur sans fil. Son architecture est basée sur une association de composants s'appuyant sur le langage NesC (network embedded system C).

MantisOS [19] : système d'exploitation dédié aux réseaux de capteurs, développé par l'université du Colorado (USA) et écrit en langage C. Contrairement à TinyOS qui est basé sur un modèle de programmation événementielle, Mantis OS s'articule autour d'un modèle commandé par l'exécution de processus.

Le tableau 1.1 décrit un bref comparatif entre les quatre systèmes d'exploitation précédents :

TAB. 1.1 – Principales différences entre les quatre systèmes d’exploitation.

Propriétés \ systemes	TinyOS	LiteOS	Contiki	MantiOS
Année de publication	2000	2008	2004	2005
Approche	Even-driven	Multi-thread	Hybride	Even-driven
Disponibilité	Open-source	Open-source	Open-source	Open-source
Langage	NesC	LiteC++	C	C
Reconfiguration	Oui	Oui	Oui	Non

1.3.7 Domaines d’application

La taille de plus en plus réduite des micro-capteurs, leur coût de plus en plus faible, la large gamme des types de capteurs disponibles (thermique, optique, de vibrations, etc.) ainsi que le support de communication sans fil utilisé, permettent aux réseaux de capteur d’envahir rapidement plusieurs domaines d’application, comme exemples d’applications potentiels, nous citons :

1. **Applications militaires** : Comme exemple d’application dans le domaine militaire : un réseau de capteurs déployé sur un endroit stratégique où il est difficile d’y accéder, afin de surveiller toutes les activités des forces ennemies, ou d’analyser le terrain avant d’y envoyer des troupes (détection d’agents chimiques, biologiques ou de radiations). Des tests concluants ont déjà été réalisés dans ce domaine par l’armée américaine dans le désert de Californie. Par contre, il faut bien noter que les applications dans ce domaine nécessitent un maximum de sécurité. Or la sécurité dans les réseaux de capteurs sans fils n’est pas assurée. En fait, les algorithmes de routage sécurisés demandent plus de ressources en mémoire et traitement que ceux non sécurisés. Dans ce contexte un nouvel axe de recherche se lance : l’optimisation des algorithmes de routages sécurisés pour les réseaux de capteurs sans fils.
2. **Application environnementales** : Le contrôle des paramètres environnementaux par les réseaux de capteurs peut donner naissance à plusieurs applications. Par exemple, le déploiement des thermo-capteurs dans une forêt peut aider à détecter un éventuel début de feu et par suite faciliter la lutte contre les feux de forêt avant leur propagation. Le déploiement des capteurs chimiques dans les milieux urbains peut aider à détecter la pollution et analyser la qualité d’air. De même leur déploiement dans les sites industriels empêche les risques industriels tels que la fuite de produits toxiques (gaz, produits chimiques, éléments radioactifs, pétrole, etc.).

3. **Application domestique :** Avec le développement technologique, les capteurs peuvent être embarqués dans des appareils, tels que les aspirateurs, les fours à micro-ondes, les réfrigérateurs, etc. Ces capteurs embarqués peuvent interagir entre eux et avec un réseau externe via Internet pour permettre à un utilisateur de contrôler les appareils domestiques localement ou à distance. Le déploiement des capteurs de mouvement et de température dans les futures maisons dites intelligentes permet d'automatiser plusieurs opérations domestiques telles que : la lumière s'éteint et la musique s'arrête quand la chambre est vide, la climatisation et le chauffage s'ajustent selon les points multiples de mesure, l'alarme est déclenchée par le capteur anti-intrusion quand un étranger veut pénétrer dans la maison.
4. **Application médicale :** Dans le domaine de la médecine, les réseaux de capteurs peuvent être utilisés pour assurer une surveillance permanente des organes vitaux de l'être humain grâce à des micro-capteurs qui pourront être avalés ou implantés sous la peau (surveillance de la glycémie, détection de cancers à l'étape précoce, etc.). Ils peuvent aussi faciliter le diagnostic de quelques maladies en effectuant des mesures physiologiques telles que : la tension artérielle, battement du coeur, etc. à l'aide des capteurs ayant chacun une tâche bien particulière. Les données physiologiques collectées par les capteurs peuvent être stockées pendant une longue durée pour le suivi d'un patient [20].

La Figure 1.7 présente quelques domaines d'applications des RCSF [20] :

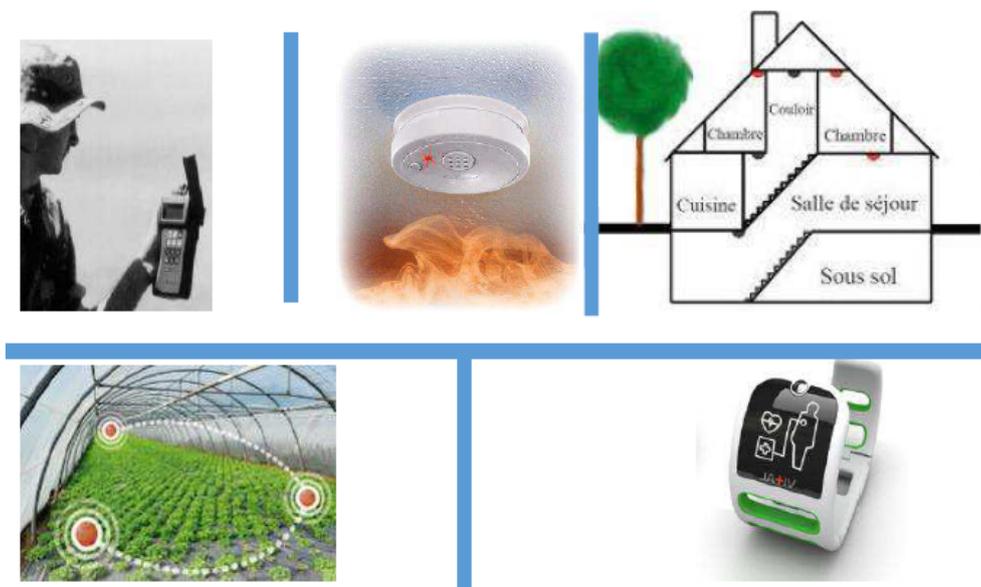


FIG. 1.7 – Quelques domaines d'applications des RCSF.

1.4 Conclusion

Dans ce chapitre, nous avons présenté les concepts généraux liés aux réseaux de capteurs sans fil, à savoir la définition et l'architecture d'un noeud capteur et d'un réseau de capteurs, leurs caractéristiques et leurs domaines d'applications.

Dans le chapitre suivant nous allons traiter la problématique de la localisation dans les réseaux de capteurs. Nous présentons les différentes méthodes existantes pour trouver les positions des noeuds, et nous décrivons les différents algorithmes de localisation.

CHAPITRE 2

Localisation dans les RCSF

2.1 Introduction

Ces dernières années, les réseaux de capteurs sans fil attirent les intérêts des chercheurs, en raison de leurs vastes applications comme les maisons intelligentes, et la surveillance de l'environnement (par exemple la détection des feux de forêt) et le suivi de déplacement des troupes ennemies dans un contexte militaire. Pour ces applications, la localisation des équipements mobiles communicants est une problématique importante car elle conditionne souvent leur bon fonctionnement. Dans le cas d'absence d'information sur la position des noeud d'un réseau de capteurs sans fil, on exploite plusieurs techniques pour récupérer leurs positions, le noeud capteur peut être équipé d'un système de localisation spécifique (par exemple : GPS), dans ce cas il peut se localiser de manière autonome mais tout simplement cette solution économiquement est non viable, donc, le noeud doit lui-même déterminer sa position à travers des méthodes de localisation qui permettent de mesurer la distance qui le sépare d'un capteur voisin ou bien de mesurer l'angle qu'il forme avec celui-ci.

Ce chapitre présente la problématique de la localisation dans les RCSF, les technologies de mesure et les méthodes de calcul de position utilisée pour la localisation, et passe ensuite à la description des algorithmes de localisation et se termine par les formes d'implémentation.

2.2 Problématique de la localisation dans les RCSF

Les perspectives et les défis des réseaux de capteurs sans fil sont très motivants, mais plusieurs problématiques doivent être prises en compte, parmi celles qui sont cruciales, nous pouvons citer : l'économie d'énergie, le routage, la qualité de service, la sécurité, la mobilité et la localisation. La problématique de la localisation dans les RCSF est particulièrement impor-

tante, dans la mesure où les noeuds capteurs sont dans un environnement évolutif et où ils peuvent éventuellement changer de position, par exemple embarqué dans un véhicule ou sur le corps d'un être humain ou d'un animal. Les applications de localisation en espace confiné (bâtiments publics, hôpitaux, appartements présentiels, etc.) disponibles actuellement reposent sur des solutions imprécises ou coûteuses en énergie ou en matériel (GPS). Cependant, ce dernier n'est pas applicable lorsqu'il s'agit de localiser des entités mobiles dans des milieux confinés ou d'intérieur.

2.3 Localisation dans les RCSF

Le but final d'un RCSF est de surveiller ou contrôler une zone d'intérêt, la localisation permet de construire une cartographie complète du réseau à partir de la connaissance de la position de quelques noeuds du réseau, ces noeuds particuliers portent le nom "ancres". La cartographie complète d'un réseau de capteurs est nécessaire car une mesure représente l'état d'un point particulier. Un système de localisation peut être décomposé en trois parties distinctes qui sont [21] :

- **Technologies de mesure** : cette partie permet d'estimer la distance entre deux noeuds, cette information est utilisée par la suite par les deux autres parties.
- **Calcul de la position** : permet d'estimer la position d'un noeud en se basant sur les mesures de distances disponibles et sur les positions des noeuds de références (ancres) en utilisant méthode de calcul de positions (la triangulation, trilatération, etc.).
- **Algorithme de localisation** : C'est la partie la plus importante du système de localisation, elle définit la manière avec laquelle les informations disponibles (distances, angles, position des noeuds déjà localisés) sont manipulées afin que la grande partie ou la totalité des noeuds puissent estimer leurs positions.

Les performances d'un système de localisation sont directement liées aux performances de chacune de ces parties ayant leur propre objectif et méthodes de résolution qui sont décrites dans le reste de ce chapitre.

2.3.1 Technologies de mesure

plusieurs technologies permettent à un capteur de mesurer la distance qui le sépare d'un capteur voisin ou bien de mesurer l'angle qu'il forme avec celui-ci. Cette mesure est très importante car elle sera utilisée pour le calcul de la position et par l'algorithme de localisation.

1. Indication de puissance du signal (RSSI)

La technologie RSSI (Received Signal Strength Indicator)[22] considère la perte de puissance d'un signal entre son émission et sa réception, cette perte varie en fonction de la

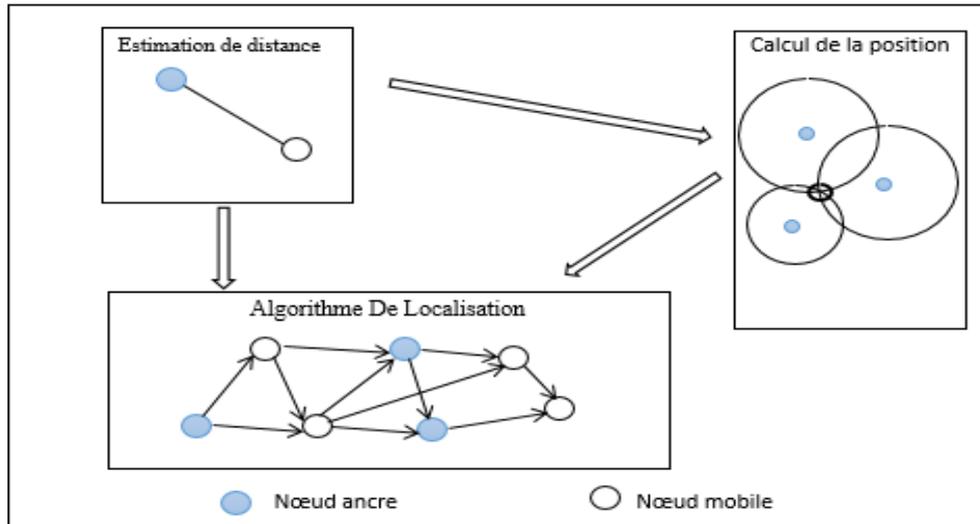


FIG. 2.1 – Composition d’un système de localisation.

distance entre les deux capteurs : plus les capteurs sont éloignés (resp. proches), plus la perte est importante (resp. faible), cette perte sera alors traduite en une distance.

La technique RSSI se base sur la puissance du signal reçu pour estimer la distance entre les deux points d’émission et de réception, la formule générale pour calculer la puissance du signal reçu dans un espace libre est [23] :

$$P_r = P_r(d_0) - 20 \log_{10}(d/d_0) \tag{2.1}$$

où

d : est la distance entre les stations de bases.

p_0 : la puissance du signal reçu à distance d_0 de l’antenne mobile.

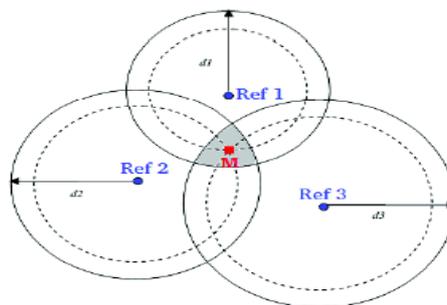


FIG. 2.2 – Principe de l’utilisation des RSSI.

Avantages [24] :

- Pas besoin d'équipement radio spécifiques ;
- Faible coût car tous les noeud sont déjà équipés de dispositifs de communication sans fil, de plus elle ne rajoute aucun surcoût de communication ;
- Algorithme de positionnement simple.

Inconvénients [24] :

- Plus les capteurs sont éloignés, plus la perte du signal est importante ;
- Sensibilité aux bruits (Ce bruit est dû à la propagation d'onde radio qui n'est pas uniforme en environnement réel) ;
- Obstacles (comme les murs, les meubles et autres reflètent et absorbent les ondes radio) et types d'antenne.

2. Temps de parcours (ToA)

La technique ToA (Time of Arrival) exploite la relation entre la distance parcourue par un signal et le temps de parcours. En effet, la distance entre l'émetteur et le récepteur est la multiplication du temps pris par le signal avec la vitesse des ondes radio [1].

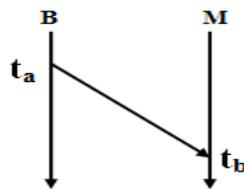


FIG. 2.3 – Principe de ToA.

$$\text{Distance} = \text{Temps} * \text{Vitesse} \quad (2.2)$$

où : $\text{vitesse} = 3 * 10^8$ m/s et $\text{Temps} = t_b - t_a$

t_b : la date de réception.

t_a : la date d'envoi.

Pour calculer la distance nous distinguons deux cas :

- **Lorsque les noeuds sont synchrones (les horloges des noeuds battent aux même rythme) :** la distance qui sépare deux capteurs se déduit de la vitesse de propagation du signal et de la différence entre les dates d'émission et de réception du message. Cette technologie est celle utilisée par le système GPS.
- **Lorsque les noeuds ne sont pas synchrones (l'horloge du noeud émetteur indépendante de celle du récepteur) :** l'envoi d'un message aller-retour est nécessaire. En fonction de son horloge, de la vitesse de propagation du signal et du temps

de traitement du signal reçu, un capteur récepteur obtient la distance qui le sépare du capteur émetteur en calculant la différence entre les dates d'émission et de réception, en y soustrayant le temps de traitement du signal, puis en divisant le résultat par deux.

Avantages [24] :

- Algorithmes de positionnement simple ;
- Précision plus élevée en milieu confiné.

Inconvénients [24] :

- Nécessite la synchronisation parfaite entre l'émetteur et le récepteur ;
- Nécessité d'avoir le trajet direct ;
- Nécessité d'une résolution temporelle élevée au récepteur.

3. Différence du temps d'arrivée (TDoA)

La technologie TDoA (Time Difference of Arrival) se base sur la différence des dates d'arrivée d'un ou plusieurs signaux et suppose également que la vitesse de propagation des signaux est connue. Cette technologie s'applique dans les cas suivants [1] :

- un émetteur envoie des signaux de natures différentes (par exemple, l'ultrason, l'onde radio, ...) à un récepteur ;
- un récepteur reçoit des signaux de même nature d'au moins trois émetteurs ;
- un émetteur envoie un signal reçu par au moins trois récepteurs (dans ce dernier cas une vue globale des signaux sera connue).

Dans chacun des cas, les récepteurs mettent en corrélation leurs informations et en déduisent les distances qui les séparent des émetteurs. Il s'agit d'une simple résolution d'un système d'équations dont les distances sont les inconnues.

Avantages [25] :

- Algorithme de positionnement simple ;
- Précision plus élevée en milieu confiné ;
- Pas besoin de synchronisation d'horloges entre le mobile et les stations de bases.

Inconvénients [25] :

- Nécessité d'avoir le trajet directe ;
- Synchronisation d'horloges nécessaire entre les paires de stations de base ;
- Nécessité d'une résolution temporelle élevée au récepteur.

4. L'angle d'arrivée (AoA)

La technologie AoA (Angle of Arrival) calcule l'angle formé entre deux capteurs. Chaque capteur est doté d'antennes orientées de sorte à déduire l'angle qu'il forme avec un voisin lorsque ce dernier lui envoie un signal. Cet angle est reporté par rapport à un axe propre au capteur. Toutefois, un capteur peut être équipé d'une boussole, dans ce cas, l'angle sera reporté sur un des axes nord, sud, est ou ouest [2].

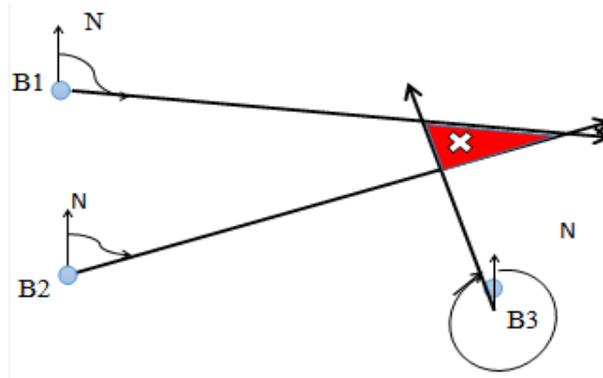


FIG. 2.4 – Principe de AOA.

Avantages [25] :

- Moins de stations de base fixes nécessaires ;
- Algorithme de positionnement simple.

Inconvénients [25] :

- Antenne directionnelles : couteuse et complexes à mettre en oeuvre ;
- Nécessité d'avoir trajectoire directe ;
- Mauvaises performances dans un canal ayant un profil de propagation par trajet multiple sévère.

TAB. 2.1 – liste des caractéristiques essentielles des méthodes décrites dans les paragraphes précédents.

Méthode	Précision	Distance Maximale	Matériel supplémentaire	Défis (challenges)
RSSI	2-4 m	La portée	Rien	Variations du RSSI, Interférences
ToA	2-3 cm	La portée	Rien	Synchronisation des noeuds
TDoA	2-3 cm	Quelques mètres (2-10 m)	Capteur à ultrason	la distance maximale
AoA	Quelques degrés (5°)	La portée	Un ensemble de récepteurs	Travailler avec des noeuds de petite taille

Le tableau 2.1 présente les différentes caractéristiques essentielles des méthodes décrites (RSSI, ToA, etc.). Le choix de la méthode d'estimation de distance/angle entre les noeuds est un facteur important qui influe sur les performances finales du système de localisation. Géné-

ralement, pour calculer une position, un noeud a besoin de trois distances (angles) estimés et chacune d'elle avec une erreur associée.

2.3.2 Méthode de calcul de positions

Quand un noeud a suffisamment d'information de distances/angles et de position (des ancres), il peut calculer sa position. Pour cela, quatre types de métriques sont généralement retenus dans les systèmes de localisation : Trilateration, triangulation, Centroid, MinMax.

1. Trilateration

Cette méthode repose sur la connaissance des distances séparant la cible de différents points d'ancres et des coordonnées spatiales de ces ancres. Soient trois ancres B_1, B_2 et B_3 , M le noeud mobile que l'on désire localiser. L'exécution du protocole de mesure de distance a produit le triplet (D_1, D_2, D_3) et les échanges de données ont permis au mobile de connaître les positions des ancres. Les équations (2.3) et (2.4) décrivent la relation entre B_1, B_2 et M .

$$(x - x_1)^2 + (y - y_1)^2 = D_1^2 \quad (2.3)$$

$$(x - x_2)^2 + (y - y_2)^2 = D_2^2 \quad (2.4)$$

La position recherchée se trouve donc à l'intersection des cercles $C_1(B_1, D_1)$ et $C_2(B_2, D_2)$. Dans le cas général, C_1 et C_2 se rencontrent en deux points distincts M et M' . Grâce aux données de l'ancre B_3 , l'un de ces deux candidats peut être retenu comme position du mobile (Figure 2.5).

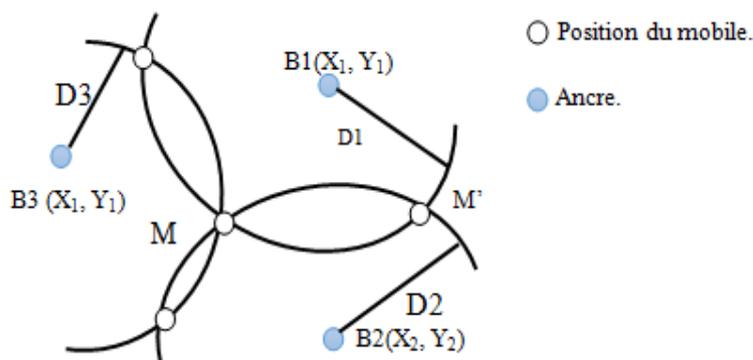


FIG. 2.5 – Principe de Trilateration.

2. Triangulation

L'angle d'incidence du signal émis par le mobile peut être utilisé pour retrouver sa position[5]. Considérons deux ancres $B_1(X_1, Y_1)$ et $B_2(X_2, Y_2)$ à portée du mobile $M(X, Y)$. Chacune dispose des moyens matériels de mesurer l'angle sous lequel elle reçoit le signal de M . Soient α et β ces deux angles.

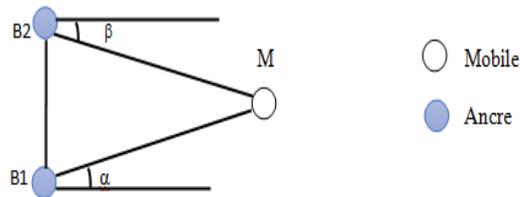


FIG. 2.6 – Angle d'incidence et position.

Le point M se retrouve à l'intersection des droites passant par les couples (B_1, M) et (B_2, M) (Figure 2.6). En définissant un nouveau repère dont l'origine est B_1 et où (B_1, B_2) se confond avec l'axe des ordonnées, on obtient la position du mobile donnée par la formule N° (2.5).

$$M\left(\frac{Y_2}{\tan \alpha + \tan \beta}; \frac{Y_2 \tan \alpha}{\tan \alpha + \tan \beta}\right) \tag{2.5}$$

3. 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. Sur la figure 2.7, le capteur M associe un " cadre " à chacune des ancres A, B, C . Chaque " cadre " est centré en la position de l'ancre (X_A, Y_A) et dépend de la distance estimée (d_{XA}) avec le capteur M . Par exemple, le " cadre " centré en A est construit comme suit :

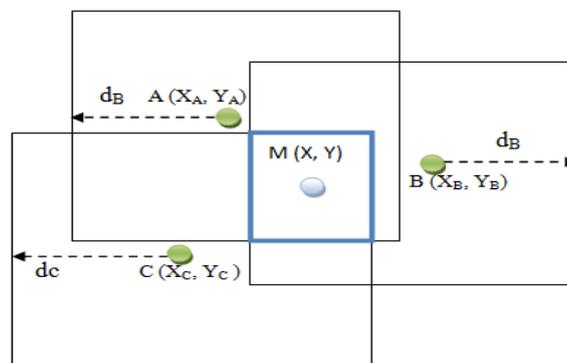


FIG. 2.7 – Principe de MinMax

$$(X_B - d_B, Y_B - d_B) \text{ et } (X_B + d_B, Y_B + d_B) \quad (2.6)$$

L'intersection de ces " cadres " forme un nouveau " cadre " définie par :

$$(\max(X_i - d_i), \max(Y_i - d_i)) \text{ et } (\min(X_i + d_i), \min(Y_i + d_i)); i \in \{A, B, C\}. \quad (2.7)$$

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

$$(X, Y) = \left(\frac{\max(X_i - d_i) + \min(X_i + d_i)}{2}, \frac{\max(Y_i - d_i) + \min(Y_i + d_i)}{2} \right) \quad (2.8)$$

4. 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 la valeur de la moyenne et de la coordonnée Y de la valeur de tous les sommets du triangle.

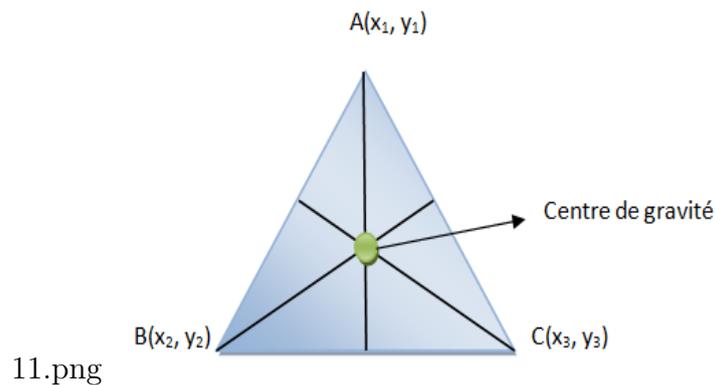


FIG. 2.8 – Principe de la méthode Centroïde

Formule centroïde : Dans la figure précédente, les trois sommets du triangle sont $A(X_1, Y_1)$, $B(X_2, Y_2)$ et $C(X_3, Y_3)$.

Méthode centroïde : Le Centroïde de points, A, B et C est :

$$(\hat{x}, \hat{y}) = \left(\frac{X_1 + X_2 + X_3}{3}, \frac{Y_1 + Y_2 + Y_3}{3} \right) \quad (2.9)$$

Voici quelque propriétés de la méthode Centroid

- Centroïde est un terme qui décrit le centre d'un objet ;
- Il est aussi appelé centre de gravité, le géocentre et le barycentre ;
- Le centre de gravité est toujours à l'intérieur du triangle.

TAB. 2.2 – les principales caractéristiques des méthodes de calcul de position.

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.

Le tableau 2.2 présente les différentes caractéristiques essentielles des méthodes décrites (Triangulation, MinMax, etc.). Un ensemble de méthodes qui peuvent être utilisées pour le calcul de la position a été exposé. Le choix de la méthode de calcul de la position influe sur les performances finales du système de localisation et dépend des informations collectées et des ressources du processeur.

2.3.3 Algorithmes de localisations

Dans cette section nous allons présenter les différentes catégories des algorithmes de localisation.

(a) Algorithmes à ancres statiques et noeuds statiques :

Dans ce cas tous les noeuds ont des positions fixes. Ces algorithmes peuvent être classifiés en deux sous classes :

- Algorithme Range-based : exigent de mesurer la distance entre les noeuds à localiser et les noeuds ancres et utilisent différentes techniques de mesure de distance (RSSI, AOA etc.) pour estimer la position d'un noeud.
- Algorithme Range-Free : les algorithmes de cette catégorie n'exigent pas la mesure de la distance ou de l'angle entre les noeuds, mais estiment la distance entre deux noeuds suivant l'information de connectivité ou celle de l'énergie consommée lors de la transmission d'un signal.

(b) Algorithmes à ancres statiques et noeuds mobiles :

l'utilisation du noeud mobile et étroitement liée à notre vie quotidienne. Ceux-ci peuvent être divisés en deux catégories :

- Algorithmes prédictifs : le principe de ces algorithmes consiste en la prédiction des coordonnées des noeuds mobiles en se basant sur l'historique de leurs positions précédentes.
- Algorithmes à base de clusters : ces algorithmes sont plus appropriés pour les

RCSF. Ils divisent le réseau en plusieurs clusters, et chaque noeud ancre localise respectivement les noeuds simple se trouvant dans son propre cluster.

(c) **Algorithmes à ancres mobiles et noeuds statiques :**

A présent, quelques algorithmes de localisation qui utilisent des noeuds ancres mobiles afin de localiser des noeuds statiques. On distingue deux catégories :

- Algorithmes de localisations géométriques : ces algorithmes changent le problème de localisation en un problème géométrique et calculent ainsi les coordonnées des noeuds inconnus en se basant sur les relations géométriques entre les noeuds mobiles et les noeuds fixes.
- Algorithmes pour planification de la trajectoire : un ancre mobile diffuse des paquets contenant des informations sur sa position à un moment donné tout en se déplaçant selon une trajectoire spécifique et les noeuds reçoivent ces paquets afin de pouvoir estimer leurs positions. Cette approche réduit le coût d'un RCSF, et offre plus de précision.

(d) **Les algorithmes à ancres mobiles et noeuds mobiles :**

Le processus de localisation de tels algorithmes peut être compliqué si nous tenons compte de l'environnement ou les noeuds peuvent être déployés, ce qui peut représenter une contrainte pour la mobilité des noeuds, par exemple : algorithmes à distribution probabiliste.

On remarque que tous les algorithmes de localisation ont une caractéristique commune est qu'ils requièrent tous des noeuds ancres (ou noeud de référence) afin de localiser les noeuds mobiles (dans la position est inconnue).

2.3.4 Forme d'implémentation

Nous distinguons deux façons d'implémenter un algorithme de localisation selon leur organisation de calcul [2] :

2.3.4.1 Les méthodes centralisées

Les algorithmes centralisés sont conçus pour fonctionner sur une machine centrale très puissante au niveau ressources. Les noeuds capteurs recueillent des informations (signal, voisins, distances, etc) de leur environnement et les transmettent à une station de base qui à son tour les analyse, calcule les positions et les transmet aux noeuds. Les algorithmes centralisés contournent le problème des ressources limitées des noeuds en acceptant des coûts de communications très élevés pour envoyer les informations à la machine centrale. Ces algorithmes deviennent de plus en plus coûteux quand la taille du réseau augmente,

car ça épuise les noeuds qui sont trop proches de la station de base qui subissent un très grand nombre de communications.

En outre, les algorithmes centralisés exigent qu'une station de base puissante soit déployée parmi les noeuds, ce qui n'est pas toujours possible. Dans le cas où c'est possible, le problème de la mise à l'échelle peut être résolu en déployant plusieurs stations de bases. Cependant, la centralisation permet à un algorithme d'être plus complexe, car les calculs se font sur la machine centrale et non pas par les noeuds eux-mêmes.

2.3.4.2 Les méthodes distribuées

Dans le cas d'un algorithme distribué, tous les noeuds communiquent avec leurs voisins pour estimer les distances et échanger les informations de voisinage, afin de dériver leur position. Par conséquent, à la fin du processus de localisation, chaque noeud doit connaître sa position ainsi que celles de ses voisins sans l'aide d'aucune unité centrale. Les algorithmes distribués, extrapolent généralement les positions des noeuds à partir de celles des ancrés. Ainsi, ils localisent les noeuds directement dans le système de coordonnées global de ces ancrés.

Comme le calcul des positions se fait par les noeuds eux-mêmes, les algorithmes distribués ne sont pas complexes. Pour les réseaux à grande échelle, on considère qu'une méthode distribuée est nécessaire car les méthodes centralisées demanderaient trop de communication pour l'acheminement des informations vers l'unité centrale et consommeraient donc trop d'énergie.

2.4 conclusion

Dans ce chapitre nous avons abordé la problématique de la localisation dans les réseaux de capteurs sans fil et les paramètres de la localisation, telles que les AOA, RSSI, etc. qui doivent d'abord être mesurés avec précision avant d'être utilisés par un algorithme adéquat pour calculer une estimation de la position d'un Mobile, ces technologies sont fortement déconseillées dans des milieux fermés. Dans le chapitre suivant nous allons utiliser la technique RSSI pour la localisation d'un capteur mobile, et nous implémentons les deux méthodes Centroïde et MinMax pour le positionnement du mobile.

CHAPITRE 3

Implémentation et Simulation

3.1 Introduction

En l'absence d'information sur la position des éléments d'un réseau de capteurs sans fil, au sein de l'environnement où ils sont déployés, les données récoltées peuvent s'avérer d'une utilité limitée. Une étape préalable à tout traitement consiste donc à estimer la position de ces capteurs à chaque instant, à partir de mesures de portée inter-capteurs telles que les RSSI (Received Signal Strength Indication), et de la position supposée connue d'une fraction de capteurs appelés ancres. Ce problème de localisation à fait et fait encore l'objet de nombreux travaux de recherche dans divers domaines.

En effet, dans le domaine des environnements ambiants intelligents, le service requis et/ou demandé est rendu à l'utilisateur selon le contexte par un réseau de capteurs enfouis dans des objets électroménagers. Souvent, le service à fournir nécessite la connaissance des coordonnées de l'utilisateur à un instant bien précis. Au-delà de cet instant, le service adéquat n'est plus le même puisqu'il se trouve que l'utilisateur a changé déjà de position. En conséquence, les défis ici sont de pouvoir trouver la position qui reflète au mieux la vraie position de l'utilisateur et cela dans un intervalle de temps bien déterminé. Divers méthodes de localisation, telles que trilération, minimax ou centroïde, ont été proposées pour tenter de répondre au premier défi. Généralement, la précision de ces méthodes dépend du nombre d'ancres déployées dans l'environnement. Cependant, vu les caractéristiques de cet environnement et la manière dont le signal s'y propage ainsi que les collisions, l'information provenant des ancres du réseau arrivent au-delà du délai demandé ou n'arrivent jamais au noeud destinataire. Donc, le deuxième défi est cette contrainte du temps qui se pose au niveau de la transmission des messages dans les réseaux de capteurs et non pas dans les méthodes de localisation elles-mêmes. En effet, les

mécanismes de transmission et de retransmission de paquets dans les réseaux de capteurs tel que CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance)[27] ne dépendent pas du temps.

Dans le travail [27] une amélioration de CSMA/CA dans Contiki 2.6 a été proposée pour prendre en compte les contraintes du temps réel telles que la disponibilité des informations dans un intervalle de temps prédéfini, l'élimination de l'information qui devient obsolète au-delà de cet intervalle et du coup libération du canal au profit des autres noeuds capteurs. Connaissant les limitations des réseaux de capteurs, l'utilisation de temps au lieu de nombre de retransmission pour la régulation du trafic permet de gagner aussi en termes d'énergie qui est une ressource critique pour ce genre de réseaux. La proposition faite dans [27] a porté sur le nombre de message reçus dans un délai pré-fixé sans aucune application pratique dans le domaine des réseaux de capteurs. Nous allons mettre en oeuvre cette proposition dans le cadre de la localisation de mobile. Nous avons opté pour deux méthodes de localisation qui sont MinMax et Centroïde connues pour leur simplicité d'implémentation mais aussi pour leur précision de localisation. La première méthode est basée mesure donc nécessite d'estimer la distance qui sépare le noeud mobile et le noeud ancre afin de calculer la position de la cible mobile. Par contre la deuxième méthode est libre de mesure, c'est-à-dire, les positions des ancres sont suffisantes pour calculer la position de la cible mobile. Nous allons implémenter ensuite simuler la localisation d'un noeud dans un réseau de capteur sous la contrainte du temps sous le simulateur Cooja fourni dans le système d'exploitation Contiki 2.6.

3.2 Problématique

Souvent, la précision de la localisation donnée par les divers systèmes de localisation basés sur la mesure RSSI est loin d'être satisfaisante comme nous l'avons déjà souligné dans le deuxième chapitre. Prenons comme exemple la fourniture de services dans les environnements intelligents pour mieux expliquer les raisons et les conséquences de cette imprécision :

- Premièrement, une localisation moins précise ou complètement erronée implique la fourniture de services inadéquats. Cette imprécision dépend souvent de la méthode utilisée mais aussi du nombre de capteurs exploités par cette méthode.
- Deuxièmement, les informations qui arrivent en retard, et qui sont prises en compte lors du calcul de la localisation, causent aussi des imprécisions sur la vraie position de la cible à localiser. En effet, un service fourni au-delà d'un certain délai devient inapproprié. Ce retard est généralement dû aux collisions subies par les messages transmis par les capteurs dans le réseau.
- Troisièmement, l'erreur de la localisation peut être due au manque d'information sur

le contexte telle le délai requis pour la localisation est trop par rapport au nombre de capteurs déployés pour la couverture complète de l'environnement. La problématique abordée ici est d'exploiter au maximum le réseau de capteurs déployé d'une manière rationnelle en prenant en compte sa densité et les délais de localisation imposés par le mobile pour améliorer la précision de la localisation.

3.3 Objectifs de notre travail

L'objectif de notre travail consiste à proposer un système de localisation basé sur RSSI pour localiser un noeud mobile dans un endroit précis à base d'un réseau de capteurs sans fil. Contrairement aux travaux qui implémentent les méthodes de localisation sans prendre en compte les caractéristiques des réseaux de capteurs telle que le routage, les mécanisme et transmission, nous allons utiliser le système d'exploitation Contiki pour implémenter et simuler notre système de localisation. Le choix de Contiki n'est pas fortuit, la plupart des primitives nécessaires à l'émulation d'un réseau de capteurs sont données par ce système. Notre système de localisation utilise la méthode MinMax (détaillée dans le chapitre 2) pour calculer la position à partir des informations provenant du réseau de capteurs sans fil. Dans le but de comparer MinMax avec d'autres méthodes, notre système de localisation utilisera aussi la méthode Centroïde au lieu de MinMax. Nous allons évaluer la précision de la localisation en fonction du :

- nombre de noeuds dans le réseau (la taille du réseau),
- nombre de message reçu à temps,
- délai accordé pour la localisation.

Notre système de localisation s'appuiera ensuite sur le mécanisme CSMA/CA proposé dans [27] pour réduire l'erreur de la localisation. Enfin, une comparaison des résultats des deux systèmes est donnée pour conclure.

3.4 Système de localisation proposé

Notre système est composé de 3 éléments, comme tout autre système de localisation, qui sont :

Technologie : nous utilisons les réseaux de capteurs qui communiquent avec la technologie Zigbee (IEEE 802.15.4). Nous supposons que ces capteurs représentent des objets de la vie quotidienne dotés de dispositifs qui permettent d'interagir avec l'environnement.

Métrique : nous utiliserons la puissance du signal reçu (RSSI) [27] pour l'estimation de la distance entre le noeud émetteur et le noeud récepteur du signal.

Méthodes de localisation : pour le calcul de la position à partir des distances, nous utiliserons la méthode Min-max [27] qui est réputée pour être précise et simple à implémenter.

3.5 Outils d'implémentation et de simulation

Les réseaux de capteurs sont considérés parmi les systèmes à ressources limitées. De ce fait, les outils logiciels conçus aux ordinateurs ne sont plus adaptables à ce type de systèmes. Il y a des outils logiciels légers qui sont dédiés spécialement aux réseaux de capteurs que ce soient des systèmes d'exploitation ou des langages de programmation. Dans cette partie, on se base sur la présentation du système d'exploitation Contiki qui est considéré comme un système d'exploitation complet et réputé. Puis, on présente le simulateur Cooja qui fonctionne à base de Contiki.

3.5.1 Le choix de Contiki

Contiki est un système open source, léger, flexible, multitâches et générique qui s'appuie sur un modèle de fonctionnement hybride [27]. 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 2002. Destiné à être embarqué dans des capteurs miniatures ne disposant généralement que de ressources limitées, Contiki a présenté l'idée d'utiliser la communication IP dans des réseaux de capteurs basse consommation.

Nous avons choisi Contiki car il présente les avantages suivantes :

- Simple ;
- Open source ;
- Programmé en langage c ;
- Disponible sur le web ;
- Réduction de la consommation d'énergie.

3.5.2 Le simulateur Cooja

Cooja est un simulateur fourni avec Contiki. Il permet de simuler un réseau de capteurs. Grâce à ce simulateur, nous pouvons tester rapidement un code écrit en langage C, sans avoir besoin de flasher de vrais capteurs. Nous pouvons répartir un nombre quelconque de noeuds sur une zone donnée. Nous visualisons alors en temps réel (ou accéléré) la topologie du réseau. Dans une simulation nous avons plusieurs fenêtres :

1. **La fenêtre Network :** En haut à gauche de l'écran, dans cette fenêtre s'affiche la représentation graphique du réseau, nous montre tous les noeuds dans le réseau simulé.

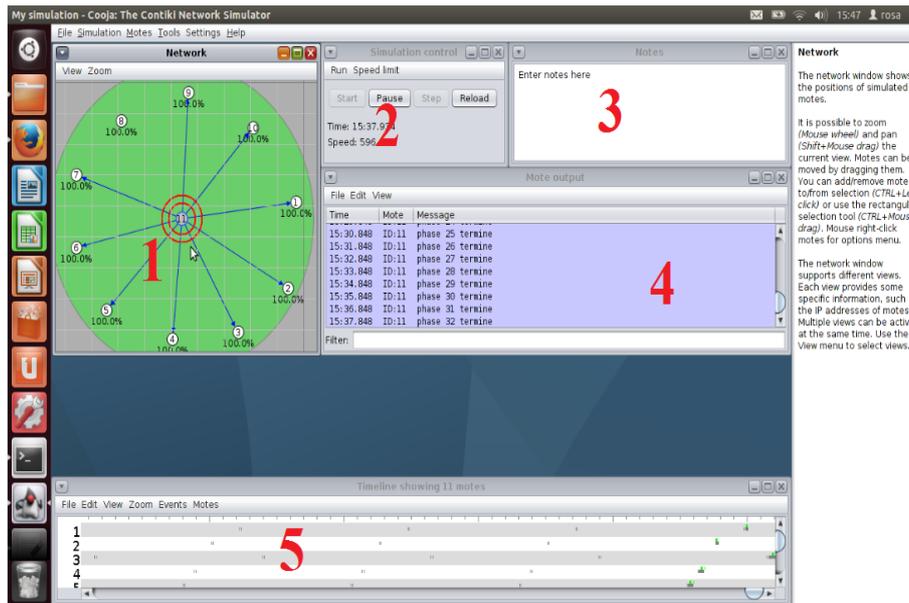


FIG. 3.1 – Interface graphique du le simulateur Cooja.

2. **La fenêtre Simulation control** : C'est la fenêtre responsable du contrôle de la simulation où nous pouvons lancer, mettre en pause et recharger de notre simulation (démarrage / mise en pause / arrêt).
3. **La fenêtre Notes** : En haut à droite est l'endroit où nous pouvons mettre des notes pour notre simulation.
4. **La fenêtre Mote Output** : sur le côté droit de l'écran, Ce module est très important car c'est ici que sont imprimées les sorties des capteurs. Un champ de texte permet d'entrer un filtre pour cibler un capteur ou un type de message en particulier.
5. **La fenêtre Timeline** : En bas de l'écran, nous affiche tous les événements de communication dans la simulation dans le temps, très pratique pour comprendre ce qui se passe dans le réseau.

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

La première ligne : Nous montre l'adresse Rime du noeud.

La deuxième ligne : Nous donne l'adresse MAC du noeud ainsi que son identifiant.

La troisième ligne : Nous montre le mécanisme par défaut ContikiMac, 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.

La quatrième ligne : Désigne le nom du processus qu'a démarré.

3.5.3 Contraintes rencontrées dans cooja

Problèmes des fonctions mathématiques qui portent, généralement, sur des réels et qui sont prédéfinis dans " math.h ". C'est au programmeur de les programmer et de faire attention aux opérations qui se font en double précision. Il faut les convertir en entier et gérer ensuite la réservation mémoire. Cette contrainte n'est pas évidemment un défaut dans Contiki mais c'est une exigence pour les capteurs afin de réduire le temps de calcul et l'occupation mémoire. La récupération des coordonnées des capteurs déployés dans une zone géographique représentée par une interface graphique n'est pas aussi programmée. Après une recherche sur les forums dédiés au système Contiki, nous avons récupéré des informations qui permettent à un noeud de connaître sa position sur l'interface graphique. Malheureusement, cette position est donnée sous forme de valeur entière en mètre, donc l'erreur de la localisation ne peut être que de l'ordre du mètre. Cette fois-ci, il fallait modifier dans le code source de Contiki pour apporter une solution avec transtypage.

3.6 Simulation

3.6.1 Scénario de simulation

Pour réaliser notre proposition nous avons déployé un ensemble de capteur dans un environnement où se déplace un mobile qui désire connaître sa position à chaque instant. Dans notre cas nous avons vingt capteurs fixes (ancres) dont les positions sont connues au paravent (X_i, Y_i) tel que $i=1..N$, et un noeud mobile qui cherche à estimer sa position à l'aide de ces ancres. Notre travail se résume dans les étapes suivantes :

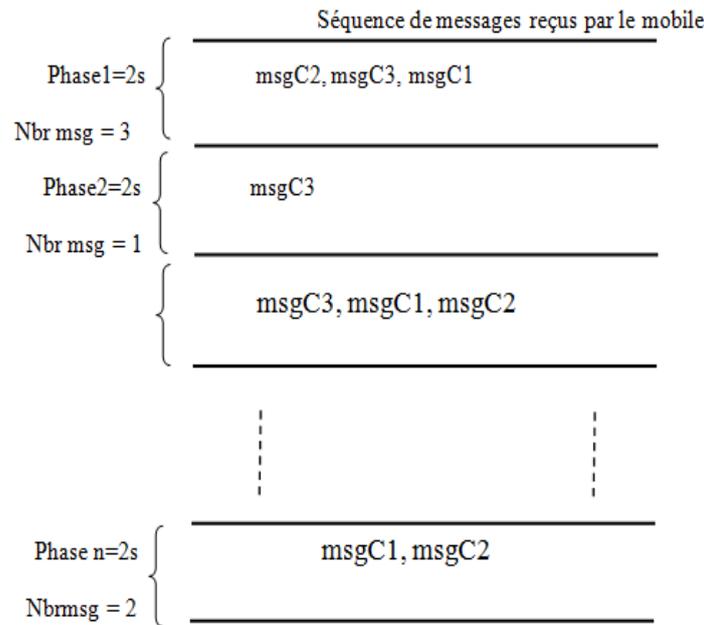
- Le noeud mobile diffuse un message ('Hello', x) vers tous les ancres du réseau de capteurs où x est le temps nécessaire à un ancre pour répondre au noeud ;
- Les ancres répondent au noeud par un message unicast indiquant leurs positions (X_i, Y_i) ;
- Le noeud mobile utilise la méthode MinMax qui utilise la position des ancres et la puissance du signal reçu (RSSI) pour calculer sa position ou bien la méthode Centroid qui a besoin de connaître que les positions des ancres.

3.6.2 Résultats de simulation

Les tests ont été effectués sur quatre réseaux, ils contiennent respectivement 3, 5, 10, 20 capteurs ancres, ces quatre tests ont été effectués en variant le temps de régénération (1s, 2s, 3s, 4s respectivement). Nous faisons durer la simulation environ deux minutes et les résultats qui nous intéressent sont les suivantes :

Nombre de messages : Nombre de message reçus par le mobile dans un intervalle connu (phase) de temps où le mobile cherche à calculer sa position.

Exemple : Prenons le cas de trois capteurs ancrés et le mobile, les capteurs ancrés envoient à chaque intervalle de temps (phase) un message unicast. Un exemple de suite de message reçus par le mobile est représenté par la figure(3.4) :



MsgCi: le message envoyé par l'ancre i vers le mobile. $i = \{1, 2, 3\}$.

Nbr msg: c'est le nombre de messages reçus par le mobile pendant une phase.

FIG. 3.2 – La suite des messages recus par le mobile

D'après la figure, durant chaque deux secondes (séparer par des lignes dans la figure), le mobile reçoit un certain nombre de messages (soit 0, 1, 2 ou bien 3). Dans la phase une le mobile à reçus trois messages venus de trois ancrés par contre dans la phase deux il n'a reçu qu'un seul message.

Taux de réception du nombre de message par le mobile : la moyenne du nombre de fois où le mobile a reçus un nombre précis de message dans une phase (soit 0, 1, 2 ou bien 3). Pour calculer le taux reprenons l'exemple précédent

$$\text{Taux de réception du nombre de message par le mobile} = (\text{nbf} * 100) / 70$$

Où :

- **nbf** : Les nombres de phases dans les quelle le mobile a reçus le même nombre de messages.
- **70** : Est le nombre totale de phase pendant la période de simulation.

La moyenne dRE : C'est la moyenne de distances qui sépare la position réelle du mobile avec la position estimée, dans le cas où il a reçu le même nombre de messages durant la simulation avec des distances différentes dRE.

$$\text{La moyenne dRE} = \sum_{i=1}^3 (\text{dRE})_i * (\text{nbf})_i / \text{Nbf} \quad (3.1)$$

Où :

dRE : Erreur est définie comme étant la distance euclidienne entre les coordonnées réelles (attribuées aléatoirement) du noeud et celles estimées par les algorithmes de localisation. L'erreur est calculée comme suit :

$$\text{dRE} = \sqrt{(x_r - x_t)^2 + (y_r - y_t)^2} \quad (3.2)$$

où :

X_r, Y_r : Les positions réelles d'un noeud.

X_t, Y_t : Les positions du noeud trouvées par la méthode

Nbf : C'est le nombre de fois que le mobile a trouvé la même valeur dRE.

Afin de pouvoir comparer les approches nous avons retenu la même topologie présentée pour tous les tests, avec un nombre précis de noeuds (3, 5, 10, 20 ancres plus un mobile) et le temps de régénération (0,5 ; 1 ; ...) et en considérant les mêmes positions des noeuds. Les résultats des tests sont enregistrés dans les tableaux suivants :

– **Teste avec méthode Centroïde**

TAB. 3.1 – Les résultats de simulation avec la méthode centroïde

Nombre de capteurs	Le temps de régénération(s)	Nombres de messages reçus par phase	Taux de nombre de message reçus(%)	Moyenne dRE(m)
1+3	0,5	0	44,09	–
		1	34,64	14,09
		2	15,74	27,63
		3	5,51	6,78
	1	0	1,36	65,57
		1	21,91	27,30
		2	28,76	12,36
		3	47,94	6,78

1+5	1,5	0	2,08	—
		1	3,12	29,71
		2	18,75	14,82
		3	37,50	12,03
		4	21,87	7,24
		5	19,79	3,58
	2	3	9,30	7,99
		4	27,90	5,56
		5	62,79	3,58
1+10	2,5	5	1,85	3,77
		6	3,70	5,73
		7	9,25	7,02
		8	12,96	4,35
		9	37,03	3,99
		10	35,18	3,71
	3	8	33,33	3,74
		9	55,50	4,66
		10	11,10	3,71
1+20	3,5	11	2,77	6,04
		12	8,33	6,28
		13	8,33	1,21
		14	5,55	4,72
		15	11,11	3,81
		16	27,77	1,94
		17	16,66	1,50
		18	8,33	1,88
		19	8,33	1,53
		20	2,77	0,39
		4	15	3,44
	16		10,34	1,83
	17		6,89	0,91
	18		17,24	1,45
	19		34,48	1,39
	20		27,58	0,39

– Teste avec méthode MinMax

TAB. 3.2 – Les résultats de simulation avec la méthode MinMax

Nombre de capteurs	Le temps de ré-génération(s)	Nombres de messages reçus par phase	Taux de nombre de message reçus(%)	Moyenne dRE(m)
1+3	0,5	2	78,72	17,98
		3	21,27	17,34
	1	2	60,78	18,88
		3	39,22	17,71
1+5	1,5	2	16,39	18,50
		3	29,86	17,13
		4	34,42	8,98
		5	19,67	1,27
	2	3	9,30	18,08
		4	37,20	10,85
5		53,48	1,27	
1+10	2,5	6	2,38	3,02
		7	14,28	2,97
		8	26,19	1,85
		9	28,57	1,38
		10	28,57	0,36
	3	6	2,94	3,77
		7	2,94	2,98
		8	14,70	1,42
		9	23,52	0,77
		10	55,88	0,36
1+20	3,5	15	6,66	0,93
		16	13,33	0,78
		17	20,00	0,35
		18	26,66	0,48
		19	26,66	0,41
		20	6,66	0,24
	4	16	15,38	0,70
		17	11,53	0,65
		18	26,92	0,62

		19	34,61	0,41
		20	11,53	0,24

3.6.3 Analyse des résultats

Les résultats des tests pour les deux méthodes Centroid et MinMax se rapprochent beaucoup car un simulateur ne permet pas de reproduire les vraies conditions de propagation des signaux hertziens. Dans notre simulation, nous nous intéresserons essentiellement au nombre de message reçu par phase et la métrique précision de localisation représente la précision de localisation de l'algorithme lors d'estimation des distances entre les noeuds et des positions. Nous supposons que nous connaissons également les vraies positions des noeuds. Après la simulation qu'a été effectuée avec deux méthodes différentes qu'on a déjà citée (centroïde, MinMax) les résultats obtenus montrent qu'il y a une différence entre les taux de nombre de message reçus par les mobiles.

- **Pour la topologie de 1+3 ancres** : Pour la méthode MinMax nous remarquons que le taux d'erreur varie entre 17m et 18m et cela dans un intervalle de temps d'une seconde, par contre dans la méthode centroïde le taux d'erreur varie entre 27 m et 6,78 m. Lorsque le nombre de message reçu par phases augmente le taux moyen d'erreur diminue. Dans ce cas nous déduisons que la méthode centroïde est la plus précise.

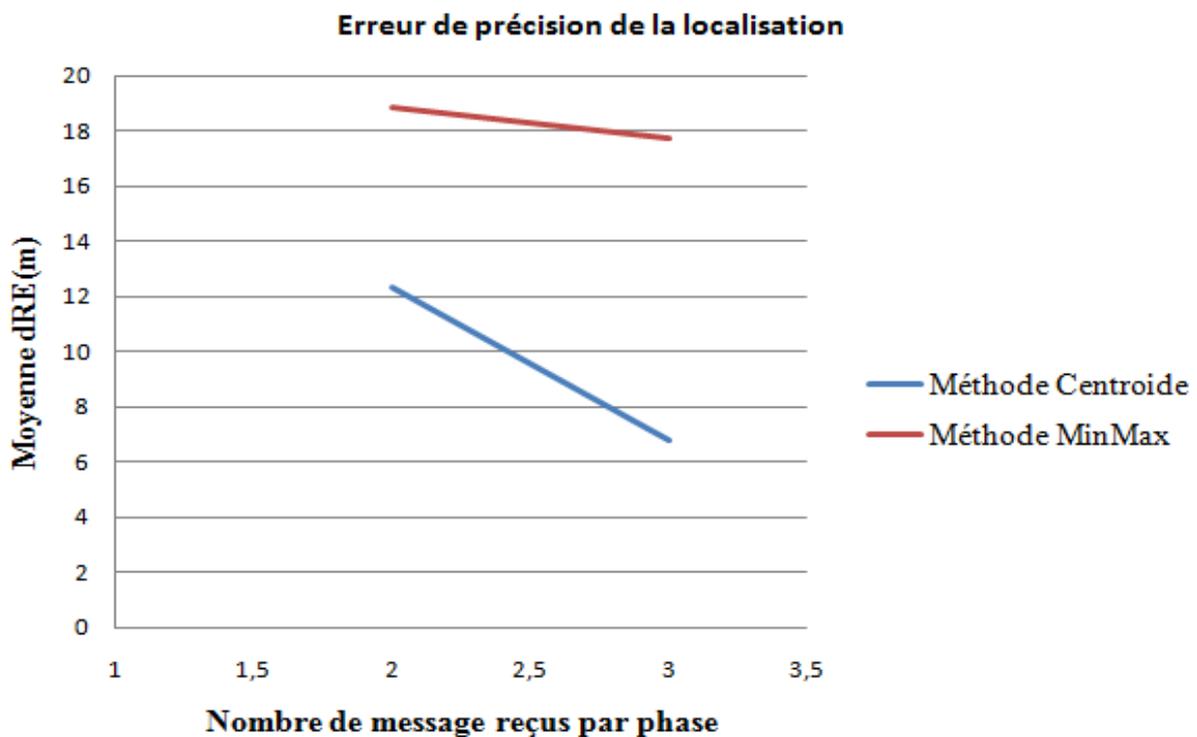


FIG. 3.3 – Comparaison d'erreur de précision entre Centroïde et MinMax pour trois capteurs.

- **Pour la topologie de 1+5 ancrés** : Avec une topologie de cinq ancrés et un temps de régénération de deux seconde le nombre de message reçu par phase augmente et le taux d’erreur diminue pour les deux méthodes, cependant la méthode MinMax donne des résultats plus exacts par apport à la méthode Centroid.

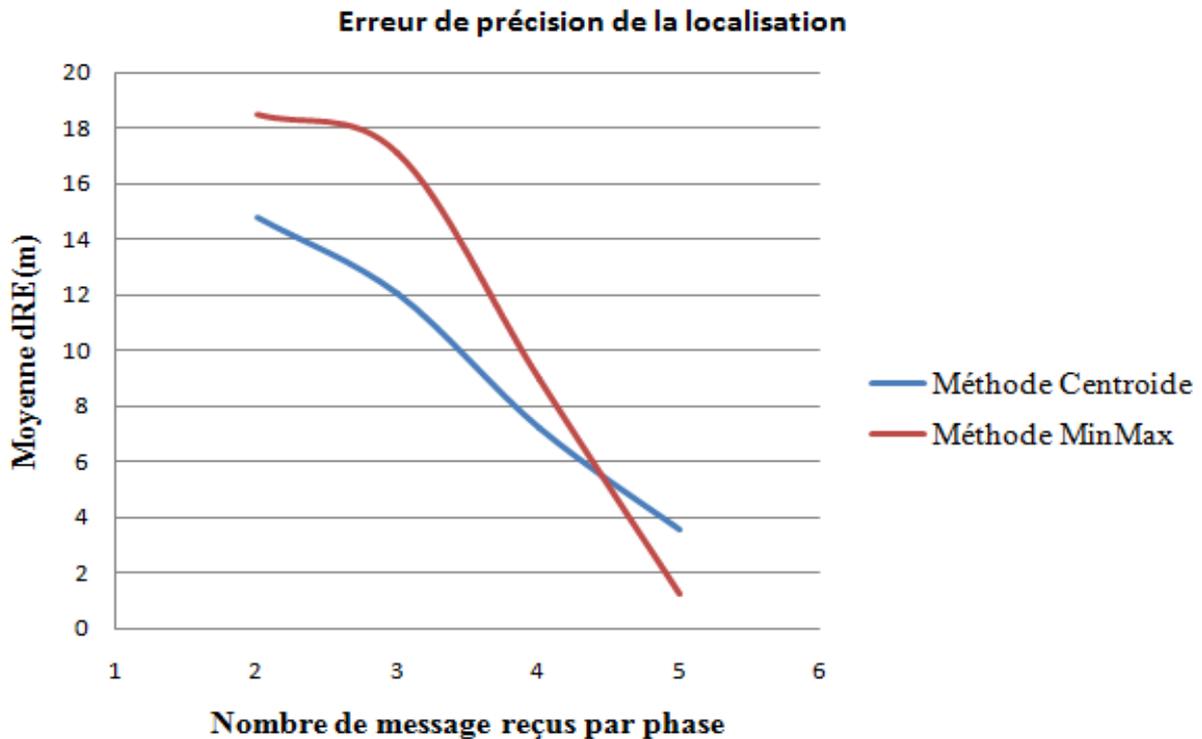


FIG. 3.4 – Comparaison d’erreur de précision entre Controide et MinMax pour cinq capteurs.

- **Pour la topologie de 1+10 ancrés** : Dans cette partie le mobile à reçus un nombre important de messages (7, 8, 9, 10 messages) qu’à permet de donner plus de précision dans la position on comparant par rapport à les deux topologies précédentes. Nous remarquont une différence de précision importante entre les résultats des deux méthodes où la méthode MinMax à donner estimation de position plus proche par rapport a la position réelle du mobile.

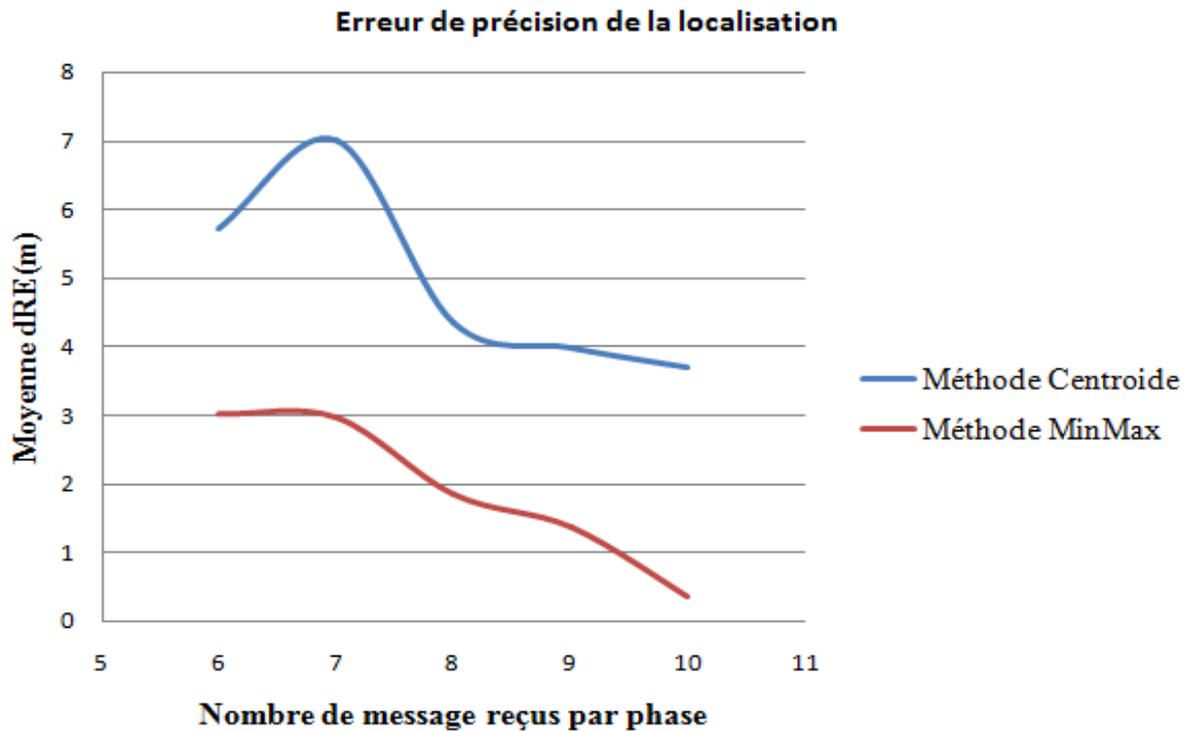


FIG. 3.5 – Comparaison d’erreur de précision entre Controïde et MinMax pour dix capteurs.

- **Pour la topologie de 1+20 ancrés** : Avec un grand nombre d’ancres le mobile reçoit un nombre important de messages (varié entre 15 jusqu’à 20 messages pour MinMax et de 11 jusqu’à 20 messages pour la méthode centroid) dans des intervalles de temps différent (3.5s, 4s). Puisque le nombre de messages reçus par le mobile est important donc les résultats de la simulation donne des positions plus précise et une distance qui sépare les deux positions du mobile celle estimé et la position réelle moins importante pour les méthodes, mais celle de MinMax est la plus proche.

Avec un grand nombre d’ancres le mobile reçoit un nombre important de messages (varié entre 15 jusqu’à 20 messages pour MinMax et de 11 jusqu’à 20 messages pour la méthode centroid) dans des intervalles de temps différent (3.5s, 4s). Puisque le nombre de messages reçus par le mobile est important donc les résultats de la simulation donne des positions plus précise et une distance qui sépare les deux positions du mobile celle estimé et la position réelle moins importante pour les méthodes, mais celle de MinMax est la plus proche.

Le graphique de les figures 3.5, 3.6, 3.7, 3.8 illustre l’erreur de distance moyenne en fonction de nombre de message reçus pour les deux méthodes. Evaluation de l’erreur de positionnement moyenne (nous parlons de l’erreur qu’il y a entre les vraies positions des noeuds et les positions calculées par la localisation) par rapport en fonction de nombre de message reçus. Le taux d’erreur a été diminuée progressivement en fonction de nombre de message reçus.

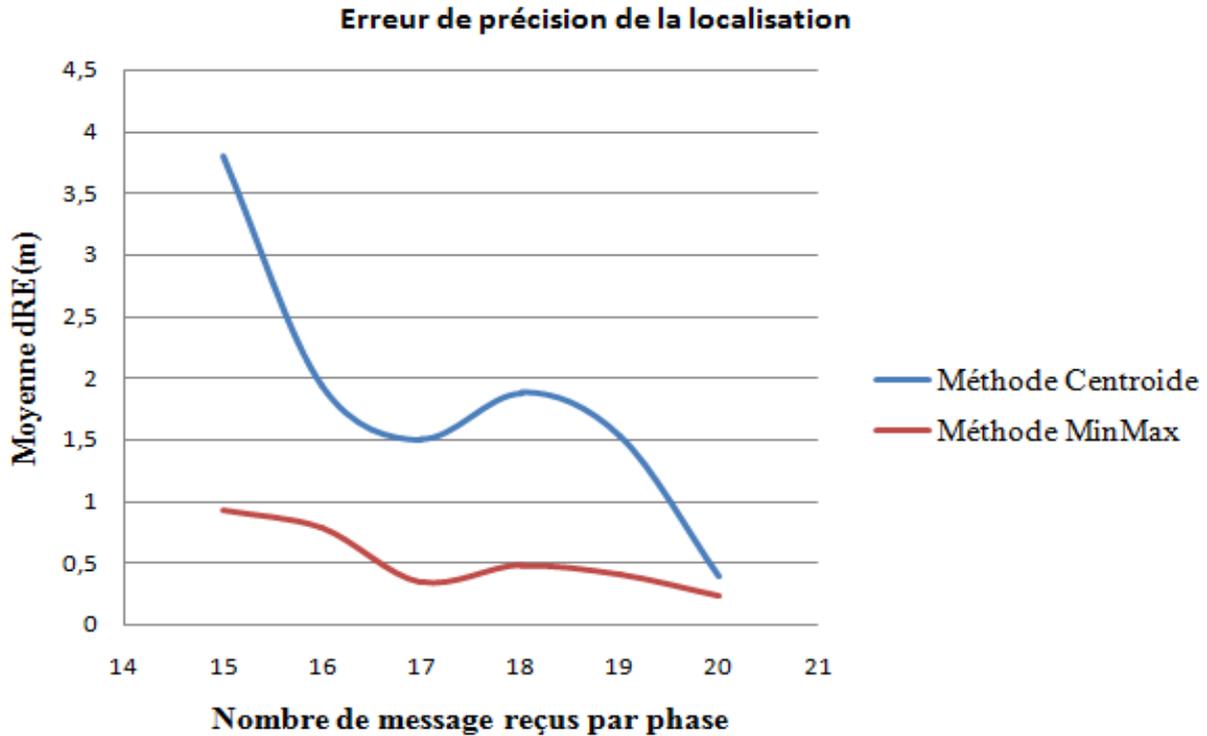


FIG. 3.6 – Comparaison d’erreur de précision entre Centroïde et MinMax pour vingt capteurs.

Notons bien que tous les résultats présentés ci-dessous comporte deux parties principales : La première nous analysons le comportement de notre approche en variant deux paramètres le nombre de noeud, et le temps de régénération de communication ; Dans la deuxième partie, nous faisons une étude comparative, où nous avons comparé les résultats des deux méthodes une par rapport à l’autre. La principale métrique évaluée dans nos simulations est l’erreur obtenue sur la localisation en fonction de nombre de message reçus.

La principale métrique évaluée dans nos simulations est l’erreur obtenue sur la localisation. Elle est définie comme étant la distance euclidienne entre les coordonnées réelles (attribuées aléatoirement) du noeud et celles estimées par les méthodes de localisation.

3.6.4 Interprétation et évaluation des résultats

Dans cette section nous présentons les différents résultats obtenus lors de l’évaluation des deux méthodes. Nous nous intéresserons essentiellement au nombre de messages reçus puisqu’il constitue le paramètre le plus critique dans la détermination de l’importance de la position estimer. Dans ce qui suit, les résultats concernent deux métriques : le nombre de messages reçus par phase et la moyenne de distances qui sépare la position réelle du mobile avec la position estimé.

Les erreurs de localisation de noeud mobile obtenues en exécutant les quatre différentes topolo-

gies (pour 1+3 capteur, 1+5, 1+10 et 1+20) sont reportées dans les tableaux 3.2, 3.3. D'après ces tableaux nous remarquons que l'erreur de localisation varie d'une topologie à une autre en utilisant la même méthode. Par exemple, pour la méthode MinMax nous remarquons que le noeud mobile s'est localisé avec une erreur de 17,34 tandis que l'erreur de localisation du même noeud est de 17,11 seulement (pour le temps de régénération 0,5 et 1s respectivement). Ce grand écart vient du fait que le noeud mobile a reçu plus de paquets dans l'intervalle de temps 0,5 que dans 1s. Par conséquent, l'erreur de localisation ne vient pas seulement de l'erreur de l'estimation de la distance mais également du nombre des paquets reçus. Le plus simple critère est de regarder la moyenne de l'erreur de localisation par rapport au nombre de messages reçus. Dans les deux tableaux 3.2, 3.3 nous remarquons que le mobile ne reçoit pas la totalité des messages, par exemple pour la méthode MinMax dans la topologie 1+3 (temps de régénération :1s) la moyenne de nombre de message reçu par phase pour 2 et 3 respectivement est 60,78% et 39,22%, ce qui influence sur les résultats et diminue la précision de la localisation.

3.7 Amélioration du système de localisation proposé

3.7.1 Solution proposée

Dans cette section nous proposons une amélioration de notre système de localisation afin de prendre en compte les différentes anomalies signalées précédemment dans la problématique. Nous allons exploiter l'idée proposée dans [27] qui permet une amélioration dans l'exploitation du réseau de capteurs déployé dans l'environnement pour prendre en considération la contrainte temps réel dans la localisation. En effet, l'amélioration apportée au protocole CSMA/CA-adapté permet de réduire le nombre de collisions et d'éliminer les informations inutiles une fois les délais répartis sont dépassés. Comme auparavant, nous allons aussi utiliser les méthodes de localisation MinMax et Centroïde, dans l'objectif de montrer que notre système de localisation reste performant avec le changement de ses composantes.

3.7.2 Rappel sur CSMA/CA-adapté

Habituellement, dans le protocole CSMA/CA la retransmission de paquet dans le cas d'échec de ce dernier dépend du nombre de retransmission. En effet un paquet peut être mis dans liste d'attente pour une durée indéterminée avant qu'il soit supprimé si le nombre de transmission fixé est atteint. La solution proposée dans [27] permet d'allouer une durée du temps pour chaque paquet afin qu'il soit transmis ou retiré de la liste d'attente. Un capteur X désirent recevoir des informations dans un délai d_t envoie un message contenant la durée d_t au capteur destinataire (Y par exemple). Ce dernier doit répondre dans le délai, si le délai est dépassé et le message

n'a pas été transmis pour des raisons de collision ou autre, le message sera supprimé.

Pour comprendre comment la période d_t est exploitée dans CSMA/CA, nous supposons que le capteur Y reçoit le message à l'instant t_1 et dispose d'un temps $d_t = t_3 - t_1$ pour transmettre un paquet comme le montre la figure 3.9. La durée de transmission globale t_1 est divisée en deux parties. La période $(t_2 - t_1)$ réservée pour se réveiller aléatoirement pour la transmission et la période $(t_3 - t_2)$ réservée uniquement pour la retransmission. Cette manière de diviser garantit d'avoir pour chaque capteur un temps pour la retransmission qui est $t_3 - t$. La procédure est la suivante :

- Le capteur va faire un tirage aléatoire pour avoir l'instant t pour transmettre, t est inclut dans l'intervalle $t_2 - t_1$.
- Si la transmission est réussie c'est-à-dire réception d'un acquittement, le capteur va attendre jusqu'à ce que l'intervalle $t_3 - t$ s'écoule pour transmettre un nouveau paquet.
- Sinon, il va retransmettre durant l'intervalle $d_t = (t_3 - t)$ jusqu'à ce qu'il reçoit un acquittement.
- Si l'intervalle est écoulé et il n'a pas reçus d'acquieement, il libère le canal et élimine le paquet.

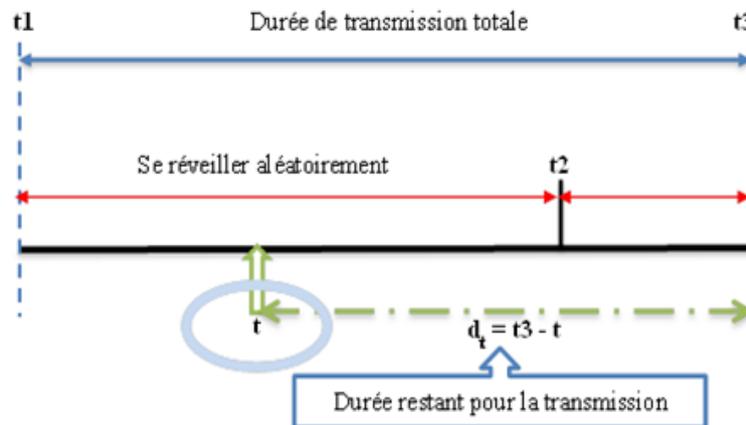


FIG. 3.7 – La durée d_t
pour envoyer un message

3.7.3 Implémentation des méthodes

Parmi les méthodes de localisation dans les réseaux de capteurs nous distinguons entre deux catégories, les méthodes qui ne sont pas basées sur la distance inter-noeuds et d'autres qui y sont. Les premières sont celles qui ne calculent pas de distances entre voisins. Elles utilisent d'autres informations telles que la connectivité pour estimer la position des noeuds. Les deuxièmes sont des méthodes qui estiment les distances entre les noeuds pour calculer les positions. Plusieurs

techniques sont développées pour les estimations des distances entre les noeuds voisins, parmi lesquelles nous trouvons celles qui sont basées sur les dispositifs radio, comme la méthode de la force du signal reçu RSSI.

Dans ce mémoire, nous avons implémenté et testé deux méthodes de dérivation des positions MinMax et Centroid, les méthodes choisis selon certains critères tel que : la méthode doit être simple et non complexe et son développement n'exige pas des grands calculs ni de grande capacité de stockage de mémoire. De plus, nous ajoutons la rapidité de l'algorithme.

L'implémentation d'une méthode de localisation a nécessité l'utilisation des outils logiciels bien spécifiques aux réseaux de capteurs sans fil, tels que le langage C, ContikiOS qui est un système d'exploitation dédié aux équipements à ressources limitées, le simulateur Cooja qui est fourni par ContikiOS.

3.7.4 Méthode Centroïde

Centroid est une méthode relativement simple et intuitive. Son but est de permettre au capteur mobile de trouver sa position à l'aide des positions connues de seulement quelques capteurs spécifiques appelés (ancres). Cette méthode ne calcule jamais de distances entre voisins. Elles utilisent d'autres informations telles que la connectivité pour identifier la position des noeuds, elle semble donner de bons résultats dans les réseaux denses et réguliers.

L'algorithme 1 présente le pseudo-code de programme Centroïde à mettre dans le noeud mobile.

```

Algorithme Centroïde (nbf, x,y) ;
entier msg, xx=0,yy=0 ;
si (nbf+1=num) alors
xx=xx+x ;
yy=yy+y ;
msg++ ;
finsi
xx=xx/msg ;
yy=yy/msg ;

```

3.7.5 Méthode MinMax

MinMax est une méthode de localisation. Son but est de permettre aux capteurs de trouver leur position à l'aide des positions connues des capteurs spécifiques appelés (ancres) et des distances entre ces ancres et le noeud à localiser. Afin de calculer la distance entre le noeud

mobile et le noeud ancre, il faut utiliser la mesure RSSI reçue par le noeud inconnu. Par défaut dans le simulateur Cooja, un modèle linéaire est utilisé pour donner l'atténuation de la puissance du signal en fonction de la distance (Unit Disk graph Medium (UDGM) : Distance Loss). La formule utilisée dans ce modèle n'est pas donnée dans la documentation de Cooja. Alors, pour trouver cette formule et pouvoir convertir la mesure RSSI en distance, nous avons déroulé des simulations où nous avons collecté des mesures RSSI sur différentes distances (voir tableau 3.1 et fait une interpolation sur ces données comme le montre la figure 3.3. Sur cette dernière, nous remarquons la distance est une fonction linéaire de RSSI, ce qui donné par la formule (3.1) :

$$d = -0.59 * rssi - 5.7 \quad (3.3)$$

où : rssi la valeur de RSSI.

TAB. 3.3 – Tableaux représentent les mesures RSSI sur différentes distances

Distance(m)	5	10	20	30	40	50
RSSI (dBm)	-18	-26	-44	-61	-77	-94

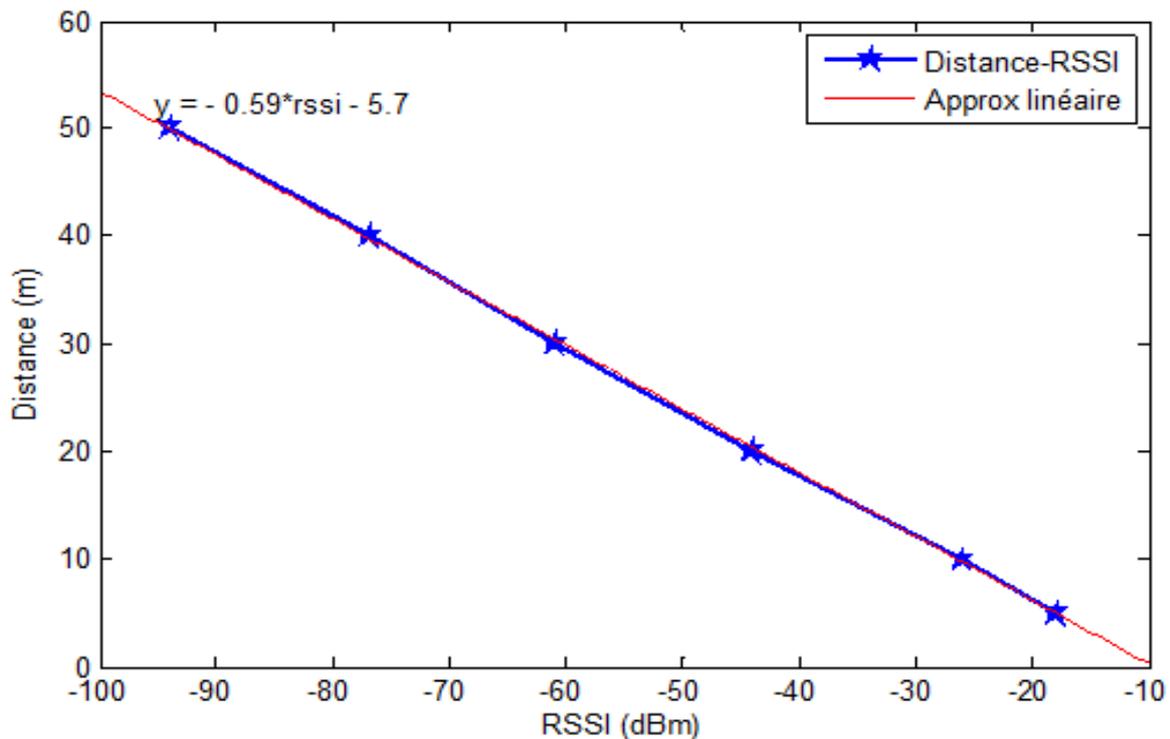


FIG. 3.8 – Les mesures RSSI sur différentes distances

Nous signalons que la puissance de transmission utilisée dans ce modèle est de 31dBm. Le principe de MinMax est présenté dans le chapitre 2. L'algorithme 2. donne le pseudo-code de

la méthode MinMax. L'algorithme 2 présente le pseudo-code du programme MinMax à mettre dans le code de noeud mobile.

```

Algorithme Min-Max(cord, d)
entier min, max, min1,max1, dd;
entier a, b, c, d, msg, i, j;
Tableau d'entier d, dmax, dmin;
Tableau de position cord;
si (msg<2)faire
afficher (le nombre de message < 2);
sinon i=0; j=0;
Tant que (i<msg*2) faire
dmax[j]=cord[i] - d[j]; dmin[j]=cord[i] + d[j];
i=i+2; j++;
fin de tant que max=dmax[0]; min=dmin[0];
pour i allant de 1 jusqu'a msg faire
si max<dmax[i] alors
max=dmax[i];
fin de si
si min > dmin[i] alors
min=dmin[i];
fin de si finpour
a=max; b=min;
i=0; j=0;
Tant que (i<msg*2) faire
dmax[j]=cord[i+1]-d[j];
dmin[j]=cord[i+1]+d[j];
i=i+2; j++;
fin tantque
max=dmax[0]; min=dmin[0];
pour i allant de 1 jusqu'a msg faire
si max<dmax[i] alors
max=dmax[i];
fin de si
si min > dmin[i] alors
min=dmin[i];
fin de si

```

```

finpour
c=max ; d=min ;
i=0 ; j=0 ;
xx=(a + b)/2 ;
yy=(c + d)/2 ;

```

3.7.6 Algorithmes des programmes à mettre dans les noeuds

Pour valider notre proposition, nous utilisons le système d'exploitation Contiki. Nous simulons un réseau de capteurs sous le simulateur Cooja, où sont déployés des noeuds ancrés et un noeud mobile qui désire connaître sa position. Le mobile envoie un message broadcast "hello+ d_t ". Les noeuds ancrés qui reçoivent ce message tentent de répondre par unicast pendant la durée d_t . Le fonctionnement de notre système de localisation est donné par les algorithmes 1 et numero 2.

Nous définissons les types suivants : Le type position est un tableau de réels à 2 dimensions pour les coordonnées (x,y). Le type adresse est l'identifiant d'un noeud capteur. Le type timer est un compteur de temps qui décroît jusqu'à 0.

L'algorithme 1 présente le pseudo-code du programme à mettre dans le noeud inconnu.

```

Algorithme noeud_inconnu
(" hello"+ $d_t$ ) ;
Tableau d'entier d
Tableau de position cord
timertemp=armer_timer( $d_t$ ) ;
Tantque 1 faire
Tantque attendre(temp)>0 faire
Si num_message=num alors
d[i]=distance_rssi(RSSI[i]) ;
cord[i]=cordonnee(ancre[i]) ;
Finsi
Fin Tantque
temp=armer_timer( $d_t$ ) ;
p=Min-Max(cord, d) ;
Fin Tantque

```

L'algorithme 2 présente le pseudo-code du programme à mettre dans le noeud ancre.

Algorithme noeud _ ancre

```
entier tr,tt ;
Booleen Transmettre=faux ;
adressedest ;
positionposition _ noeud
Constante a ;
timertemp
Si noeud _ ancre recoit " hello " + dt alors
dt=duree() ;
tr=dt-a ;
dest=ID _ noeud() ; récupérer l'adresse du noeud
Transmettre=vrai ;
Tant que transmettre faire
tt=tirage _ aleatoire(tr) ;
temp=armer _ timer(tt) ;
attendre(temp) ;
unicast(dest, position _ noeud) ; message unicast
temp=armer _ timer(dt-tt) ;
attendre(temp) ;
Fin Tantque
Finsi
```

3.8 Simulation du système de localisation après l'amélioration

pour ces tests nous avons simulé quatre réseaux qui contiennent 3, 5, 10, 20 capteurs ancrés et un mobile avec un temps de régénération respectivement 0,5 ; 1,5 ; 2,5 ; 3,5. Les résultats sont présentés dans les deux tableaux 1.3, 1.4.

– Méthode centroïde

TAB. 3.4 – Les résultats de simulation avec la méthode centroid après l'amélioration

Nombre de capteurs	Le temps de ré-génération(s)	Nombres de messages reçus par phase	Taux de nombre de message reçus(%)	Moyenne dRE(m)
1+3	0,5	0	4,22	–
		1	2,81	39,17
		2	29,57	9,87
		3	63,38	6,11
1+5	1,5	3	3,50	5,49
		4	3,50	4,19
		5	92,98	3,59
1+10	2,5	8	2,32	1,84
		9	23,25	0,62
		10	74,41	0,35
1+20	3,5	16	7,69	1,86
		17	11,53	1,92
		18	23,07	0,72
		19	19,23	0,62
		20	38,46	0,29

– Méthode MinMax

TAB. 3.5 – Les résultats de simulation avec la méthode MinMax après la simulation

Nombre de capteurs	Le temps de régénération(s)	Nombres de messages reçus par phase	Taux de nombre de message reçus(%)	Moyenne dRE(m)
1+3	0,5	2	50	20,69
		3	50	17,83
1+5	1.5	3	1,53	7,20
		4	38,92	3,5
		5	61,53	1,13
1+10	2,5	8	18,60	1,30
		9	4,65	0,97
		10	76,74	0,38
1+20	3,5	16	20	1,96
		17	8,57	0,81
		18	20	0,31
		19	25,71	0,25
		20	25,71	0,20

Analyse des résultats

D'après les résultats des deux tableaux 3.4 et 3.5 nous constatons que le nombre de message reçu est augmenté, en comparant avec les résultats obtenus auparavant, nous distinguons que le nombre de messages a augmenté par exemple pour le cas de la topologie 1+10 (pour méthode MinMax et le temps de régénération égale à 2.5s) est le nombre de message reçu par phase est égale 9, avec une moyenne respectivement 23, 25 avec celle après la proposition : le nombre moyenne de message reçus par phase est 9, avec une moyenne respectivement de 4, 65. Pour comparer les résultats de la simulation avant la proposition et les résultats des simulation après l'amélioration nous avons calculer le taux d'erreur moyenne durant 2 minutes de simulation pour chaque une des méthode implémentée après et avant la proposition. Nous constatons que :

- Le taux de messages à temps augmente dans les deux méthodes.
- L'erreur de calcul diminue dans les deux méthodes.
- MinMax est meilleure par rapport à Centroïde.

3.9 Comparaison des résultats

le tableau suivant représente la moyenne des taux d'erreur dans les différentes simulations avec différents temps de régénération :

Méthode		Temps de régénération			
		0,5	1,5	2,5	3,5
Avant l'amélioration	Méthode Centroïde	9,60	10,50	4,28	2,55
	Méthode MinMax	17,48	11,48	1,47	0,48
Après L'amélioration	Méthode Centroïde	7,89	3,67	0,44	0,76
	Méthode MinMax	19,26	2,35	0,57	0,25

FIG. 3.9 – Tableau comparative de taux d'erreu des deux méthode avant et après l'amélioration.

Pour les deux méthodes, nous constatons une augmentation de nombre de messages reçu par phase, en ce qui concerne le taux d'erreur est aussi très petit. le fait d'avoir des nombres de messages par phase est nettement amélioré la précision de la position.

3.9.1 Evaluation et interprétation

en comparant avec les résultats obtenus avant est après la proposition pour la méthode Cenroid (figure 3.12), nous distinguons que le taux d'erreur a diminué au fonction de nombre de message récus par phase. Les Figures 3.11 et 3.12 comparent le taux d'erreur par au fonction de nombre de message par phase avant et après la proposition (respictivement pour Centroid et MinMax).

L'augmentation de nombre de message reçu par le mobile pendant une phase possède un avantage majeur. l'avantage et que le mobile reçoit plus d'information (plus de position d'ancres et de distance), qui va lui permet de calculer sa position la plus exact. donc nous arrivons a exploiter la totalité ou bien la majorité des capteurs même si on diminue le temps de régénération.

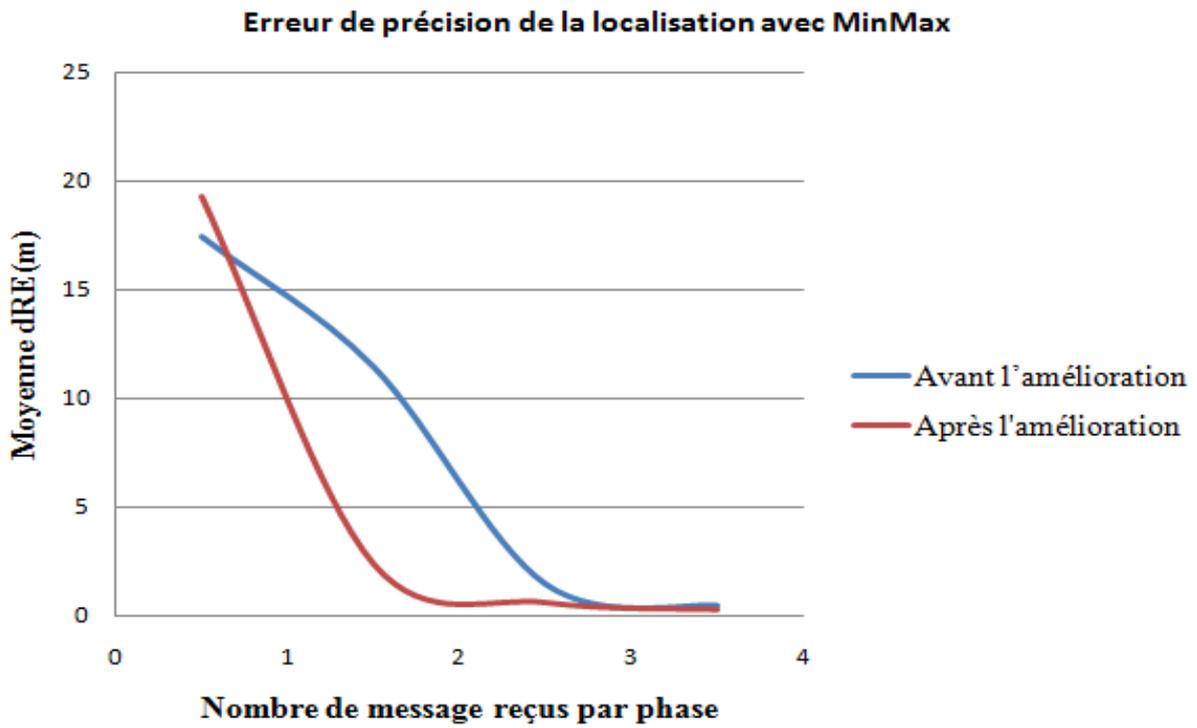


FIG. 3.10 – Différence des taux d'erreur avant et après l'amélioration avec MinMax

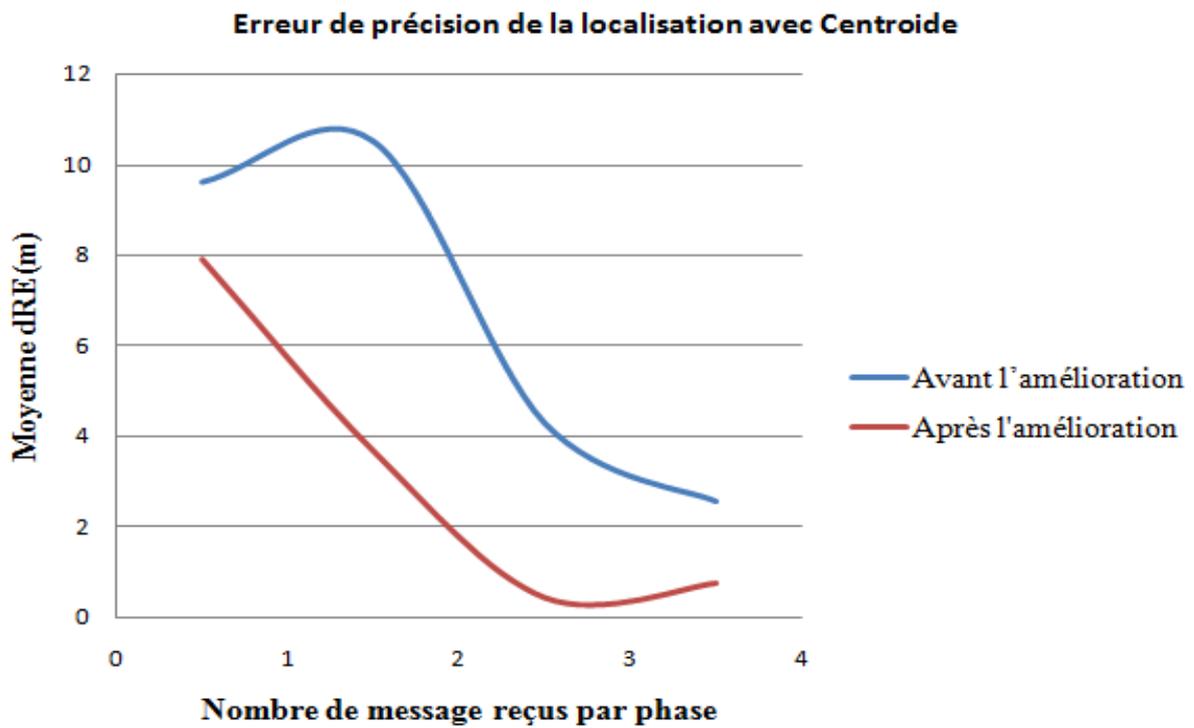


FIG. 3.11 – Différence des taux d'erreur avant et après l'amélioration avec Centroide

3.10 Conclusion

Dans ce chapitre nous avons présenté l'environnement de simulation avec lequel nous avons travaillé : simulateur Cooja et sa caractéristique. Nous avons également défini les paramètres de simulation ainsi que la métrique d'évaluation prise en compte dans notre étude. Les simulations réalisées ont mené à étudier des différents critères pour avoir plus de précision lors de la localisation dans les réseaux de capteur sans fil. Cependant, des cas non prévisibles peuvent se présenter : le mobile n'a pas pu recevoir tous les messages envoyés par les ancres. Pour le résoudre, nous avons proposé d'adapter les deux méthodes déjà implémentées.

Nous avons proposé un système de localisation composé de trois éléments : la technologie Zigbee (IEEE 802.15.4), métrique RSSI pour l'estimation de la distance entre les nœuds et deux méthodes de localisation MinMax et Centroïde, puis nous avons introduit une approche pour améliorer l'implémentation, puis nous avons proposé un système de localisation indoor et prendre en compte la contrainte temps réel, nous avons exploité l'idée proposée dans [26] qui permet une amélioration dans l'exploitation du réseau de capteurs déployés dans l'environnement. L'idée est de maximiser le nombre de messages reçus. Contrairement aux deux méthodes implémentées avant l'adaptation, notre adaptation résiste mieux aux délais de transmission. La localisation du mobile est devenue plus exacte. Donc nous arrivons à exploiter la totalité ou bien la majorité des capteurs.

Conclusion Générale

Une grande majorité des applications dans les réseaux de capteurs utilise un déploiement aléatoire d'un grand nombre de capteurs, en raison soit de l'hostilité de la zone à surveiller, soit de son immensité. La phase de localisation est donc nécessaire non seulement au fonctionnement du réseau (routage géographique par exemple), mais également à l'exploitation des données récoltées. Il est donc nécessaire de les localiser, avec la meilleure précision possible. Cette problématique, malgré les nombreux travaux de recherche qui s'y étaient attachés, reste une problématique ouverte. En effet, c'est dans le cadre de cette thématique que s'insère notre projet de fin cycle.

Certaines applications de localisation imposent des fortes contraintes, comme la précision de positionnement et le délai. CSMA-Contiki est adapté pour envoyer un maximum d'informations par capteurs et peu importe le temps occupé, mais le mobile ne reçoit pas la totalité ou la majorité des messages envoyés par les ancres. Notre travail fait l'objet d'assurer une meilleure exploitation des capteurs en terme de quantité des informations (plus de précision), pour cela nous avons procédé en trois phases :

- La première phase : proposition d'un système de localisation qui est composé de la technologie de RCF, la métrique RSSI, le mécanisme CSMA et les méthodes.
- La deuxième phase : nous avons implémenté deux méthodes de localisation MinMax et Centroïde, MinMax a pour but de trouver les positions des nœuds à l'aide des positions connues des capteurs spécifiques appelés (ancres) et des distances entre ces ancres, Centroïde permet au capteur mobile de trouver sa position à l'aide des positions connues de seulement quelques capteurs spécifiques appelés (ancres) l'implémentation de ces méthodes a été faite sous le système d'exploitation Contiki. Après l'évaluation et l'interprétation des

- résultats obtenus dans les divers tests effectués, nous avons remarqué que le noeud mobile.
- La troisième phase : nous avons pensé à augmenter le nombre de message reçu pour une meilleure exploitation des capteurs en terme de quantité des informations. et pour cela nous avons proposé un système de localisation indoor. Notre système est composé de 3 éléments : La technologie Zigbee (IEEE 802.15.4), la métrique puissance du signal reçu (RSSI) [23] pour l'estimation de la distance entre le noeud émetteur et le noeud récepteur du signal et la méthodes de localisation MinMax. pour prendre en compte la contrainte temps réel dans la localisation indoor, nous avons exploiter l'idée proposée, nous avons aussi utilisé une autre méthode de localisation Centroide dans l'objectif de montrer que notre système de localisation reste performant avec le changement de ses composantes. En effet, l'amélioration apportée à CSMA/CA permet de réduire le nombre de collisions et d'éliminer les informations inutiles une fois les délais répartis sont dépassés, nous avons éliminé le phénomène des retardataires ce qui diminue le nombre de collisions. l'avantage et que le mobile reçoit plus d'information (plus de position d'ancres et de distance) cequi augmente le nombre de message reçu par le mobile pendant une phase, qui va lui permet de calculer sa position la plus exact.

Une amélioration de la précision de la localisation dans un millieux de réseau de capteurs sans fil en fonction de délais alloué.

Ce travail nous a permis de découvrir la programmation sur des systèmes embarqués, qui est beaucoup plus limitée en termes de ressources matérielles et de calculs. Les réseaux de capteurs sont le monde de demain, avec un très large choix d'utilisation tel que la domotique, pour commander l'allumage de chauffage en fonction de la température, la localisation de personnes telles que des pompiers dans des bâtiments lors d'interventions.

Comme perspective, il serait très intéressant de joindre les avantages des algorithmes basé mesure en termes de précision aux algorithmes libre de mesure. Une bonne hybridation permettra d'avoir un algorithme de localisation ayant comme avantage une meilleure précision, à moindre coût.

BIBLIOGRAPHIE

- [1] A. Barros., C. Bérenguer., F. Brissaud & D. Charpentier, "Capteurs Intelligents : Nouvelles technologies et Nouvelles problématiques pour la sureté De Fonctionnement", Institut National de l'Environnement Industriel et des Risques, 2008.
- [2] J. Shi, "Implémentation de mécanismes de sécurité efficaces pour les réseaux de capteurs", Mémoire de Master, 2010.
- [3] H. Bouziane, "Un système automatique pour la génération des emplois du temps", Mémoire master en Informatique, 2012.
- [4] D. Martins, "Sécurité dans les réseaux de capteurs sans fil Stéganographie et réseaux de confiance", L'U.F.R. des Sciences et Techniques de l'université de Franche-Comté, 2010.
- [5] M.S. E. William, "Conception d'une architecture hiérarchique de réseau de capteurs pour le stockage et la compression de données", Thèse de doctorat en Informatique, 2010.
- [6] K.Beydoun, "Conception d'un protocole de routage hiérarchique pour les réseaux de capteurs", Thèse de doctorat en Informatique, 2009.
- [7] Vernon S. Somerset, "Intelligent and Biosensors", Edited by Vernon S. Somerset, Intech, 2010.
- [8] Y. Younes, "Minimisation d'énergie dans un réseau de capteurs", Mémoire de Master, Département d'Informatique, 2012.
- [9] Tinyos. [http :/www.tinyos.net/](http://www.tinyos.net/), consulté le 04/2016.
- [10] A. Dunkels, B. Grönvall, & T. Voigt. "Contiki : a Lightweight and Flexible Operating System for Tiny Networked Sensors. In Proceedings of the First IEEE Workshop on Embedded Networked Sensors", Article, 2004.
- [11] M. Badet & W. Bonneau. "Mise en place d'une plateforme de test et d'expérimentation", Projet tutoré, 2006.

- [12] R. David "collecte des données d'un réseau de capteurs sans fils en utilisant une surcouche réseau pair à pair ", mémoire de maîtrise en science appliquées, 2010.
- [13] B. Cécile & M. Thierry VAL, "Modélisation de la consommation d'énergie en vue de la conception conjointe (matériel/logiciel) des application embarquées. Application aux réseaux de capteurs sans fil(WSN)", Thèse doctorat, 2015.
- [14] N. Fourty., V. Den bossche & T. VAL. "An advanced study of energy consumption in an IEEE 802.15.4 based network : everything but the truth on 802.15.4 node lifetime". Article, 2012.
- [15] Contiki. [http :/www.Contiki.net/](http://www.Contiki.net/), consulté le 04/2016.
- [16] LiteOS. [http :/www.LiteOS.net/](http://www.LiteOS.net/), consulté le 04/2016.
- [17] Q. CAO., T. Abdelzaher., J. Stankovic, AND T. HE, "The LiteOS Operating System : Towards Unix Like Abstraction for Wireless Sensor Networks", Article, 2008.
- [18] S. Bhatti., J. Carlson., H. Dai., J. Deng., J. Rose., A. Sheth., B. Shucker., C. Gruenwald., A.Torgerson, and R. Han, "Mantis OS : an embedded multithreaded operating system for wireless micro sensor platforms", Article, 2005.
- [19] P.Johnson and D.C,"Remote continuous monitoring in the home", Article, 1996.
- [20] X. Yong., G. Andres., A. Andres & B. Mickaël, "Agrégation de données dans les réseaux de capteurs", Rapport de Projet, 2010.
- [21] A. Boukerche, "Algorithms And Protocols For Wireless Sensor Networks", Article, 2008.
- [22] K. Messaoud, "Réseau de capteurs sans fil : Etude en vue de la réalisation d'un récepteur GPS différentiel à faible coût", thèse, 2009.
- [23] CS 515, "Mobile and Wireless Networking Homework 1 Solutions", Article, 2002.
- [24] Y. P. Zhang, "Recent advances in integration of antennas on silicon chip and on ceramic package", article, 2005.
- [25] E. M BARBOSA NOGUEIRA "Conception d'un système d'antennes pour la localisation en temps réel avec réseau de capteurs sans fil", thèse de doctorat, 2013.
- [26] A. Dunkels., B. Grönwall., T. Voigt " Contiki - a Lightweight and Flexible Operating System for TinyNetworkedSensors ", article, november 2004.
- [27] B. youcef ,M. mohamedcherif, "Simulation d'un réseau de capteurs sous Cooja : adaptation du protocole CSMA/CA au temps réel."Mémoire de master professionnel, université de Béjaia, 2014/2015.

ANNEXE A

Annexe

Contiki est un périphérique OS capteur sans fil est de construire de petits noeuds réseau de capteurs sans fil. Il est très facile de port sur tous les périphériques de ressources limitées embarqués. Si vous travaillez sur Internet des choses comme l'application, ce système d'exploitation est très approprié pour construire une application de noeud de capteur pour installer OS contiki et son simulateur dans Ubuntu.

Pour préparer un environnement contiki sous linux. Cette annex montre comment obtenir l'environnement en question étape par étape sous ubuntu 12.04 LTS.

– Etape 1 : Télécharger le code source de Contiki

Cette méthode nécessite une installation existante d'un système d'exploitation basé sur Linux, dans notre cas nous avons installé contiki 2.6 sous ubuntu 12.04 LTS, on doit d'abord le télécharger, plusieurs versions de contiki existent sur le net, il est préférable de télécharger et d'utiliser la dernière version celle-ci est la plus performante.

la version que nous avons utilisée est contiki 2.6 téléchargée depuis le officiel de contiki :

```
http://downloads.sourceforge.net/project/contiki/Contiki/Contiki-2.6/contiki-2.6.zip
```

nous décompressez le fichier téléchargé dans le répertoire personnel.

ou bien on tapant :

```
user@machine :~$ contiki-2.6.zip unzip
```

– Etape2 :Installez tous les paquets nécessaires pour la compilation et l'exécution de Contiki

Pour les capteurs on trouve principalement deux grandes familles de microcontrôleurs :

msp430 de Texas Instrerements Inc et avr d'Atmel.

Dans le cas au on veut utilise une plate-forme MSP430 (tels que Tmote) nous avant seulement besoin des paquets suivants : binutils-MSP430 , gcc-MSP430 , MSP430-libc , msp430mcu et mspdebug . Si on a l'intention de coder pour plusieurs plates-formes, nous devant également installer les chaînes d'outils suivants pour d'autres micro-contrôleur : binutils-RVA , gcc-RVA , gdb-RVA , avr-libc et avrdude .Dans notre cas nous allons installer les boites à outils de la plate forme MSP430 et leur version du compilateur gcc. installer les paquets suivants via la commande apt :

```
user@machine :~$ sudo apt-get install binutils-msp430 gcc-msp430 msp430-libc
```

```
user@machine :~$ sudo apt-get binutils-avr gcc_ avr gdb-avr avr_ libc
```

– **Etape 3 : installer les outils JAVA**

Le simulateur Cooja qui est écrit en java a besion de deux destributions de ubuntu OpenJDK (l'implémentation de Java Développement Kit) et la OpenJRE (la machine virtuelle). l'installation de ces deux derniers est simple avec la commande suivante :

```
user@machine :~$ sudo apt-get install open-7-jdk openjdk-7-jre
```

Après l'installation de java nous devons éditer la variable d'environnement path :

```
user@machine :~$ export JAVA_HOME=/usr/lib/jvm/java-7-openjdk-i386
```

```
user@machine :~$ export PATH=PATH:JAVA_HOME/bin
```

– **Etape 4 : installer le programme ant** ant est un programme sera utilisé pour compiler le simulateur Cooja en ligne de commande. L'installation de ant se fait avec la commande suivante :

```
user@machine :~$ sudo apt-get install ant
```

– **Etape 5 : tester le fonctionnement de contiki**

A cette étape, nous avons téléchargé et décompressé contiki et nous avons installé suffisamment d'outils pour exécuter Contiki, nous nous rendons au sous-dossier hello-world dans l'arborescence avec la commande cd puis nous compilons Contiki avec make comme suit :

```
user@machine :~$ cd./contiki-2.6/examples/hello-world
```

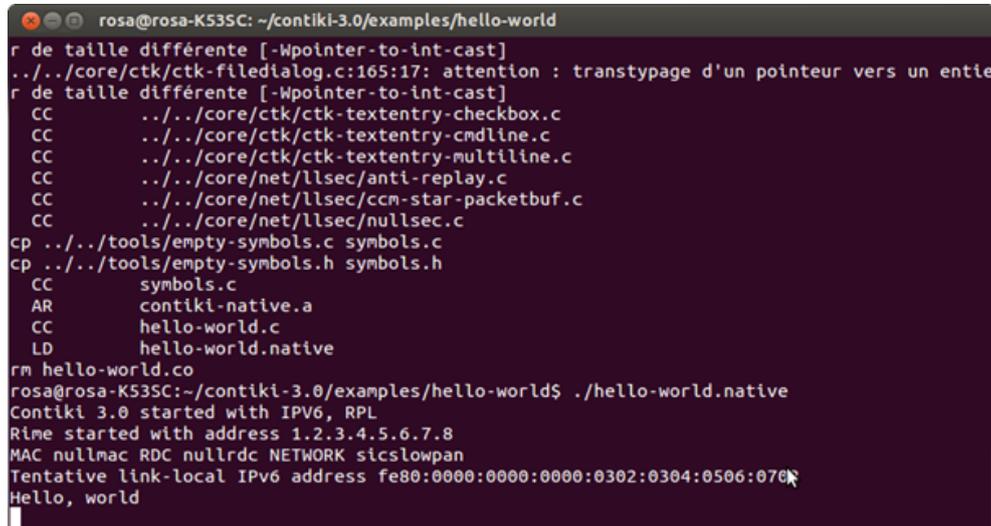
```
user@machine :~$ cd./contiki-2.6/examples/hello-world$make TARGET=native hello-world
```

Une fois la compilation est terminée, et si la compilation s'est déroulé sans erreurs ; un fichier exécutable portant le nom hello-word.native est crée, nous exécutons ce fichier qui

contiens le programme hello world à partir du terminal :

```
user@machine :~/contiki-2.6/examples/hello-world.native
```

Vous devriez voir ce qui suit sur le terminal : Le code semble se bloquer, cependant,



```

rosa@rosa-K53SC: ~/contiki-3.0/examples/hello-world
r de taille différente [-Wpointer-to-int-cast]
../core/ctk/ctk-filedialog.c:165:17: attention : transtypage d'un pointeur vers un entie
r de taille différente [-Wpointer-to-int-cast]
CC      ../core/ctk/ctk-textentry-checkbox.c
CC      ../core/ctk/ctk-textentry-cmdline.c
CC      ../core/ctk/ctk-textentry-multiline.c
CC      ../core/net/llsec/anti-replay.c
CC      ../core/net/llsec/ccm-star-packetbuf.c
CC      ../core/net/llsec/nullsec.c
cp ../tools/empty-symbols.c symbols.c
cp ../tools/empty-symbols.h symbols.h
CC      symbols.c
AR      contiki-native.a
CC      hello-world.c
LD      hello-world.native
rm hello-world.co
rosa@rosa-K53SC:~/contiki-3.0/examples/hello-world$ ./hello-world.native
Contiki 3.0 started with IPV6, RPL
Rime started with address 1.2.3.4.5.6.7.8
MAC nullmac RDC nullrdc NETWORK sicslowpan
Tentative link-local IPv6 address fe80:0000:0000:0302:0304:0506:0708
Hello, world

```

FIG. A.1 – l'exécution du programme hello-world sous Contiki.

il est encore en cours d'exécution sur Contiki, mais ne produisant pas de sortie que le programme Bonjour tout le monde est terminé. Appuyez sur Ctrl-C pour quitter.

- **Etape 6 : lancer le simulateur Cooja** L'accès au simulateur se fait à travers le terminal avec la commande suivante :

```
user@machine :~$ cd contiki-2.6/tools/cooja
```

Pour exécuter Cooja il faut se déplacer dans le sous répertoire cooja dans l'arborescence de Contiki et exécuter la commande ant run :

```
user@machine :~$ cd contiki-2.6/tools/cooja ant run
```

Enfin, la compilation s'est déroulé sans erreurs et le simulateur s'exécute correctement, là nous sommes prêts à simuler des réseaux de capteurs qui exécutent Contiki et bénéficier de tous les outils offerts par Cooja.

Résumé

En l'absence d'information sur la position d'un noeud d'un réseau de capteurs sans fil (RCSFs), au sein de l'environnement où ils sont déployés, les données récoltées peuvent s'avérer d'une utilité limitée, donc le problème traité concerne la localisation d'un mobile. Dans ce mémoire nous proposons et implémentons deux méthodes pour résoudre le problème, l'une c'est la méthode Centroïde qui utilise les positions des ancres où leurs positions est connue et MinMax qui utilise la mesure RSSI de portée entre le mobile et les ancres pour calculer sa position. Nous avons implémenté ces deux méthodes sous contiki 2.6, dans l'objectif d'améliorer l'estimation de la position du mobile dans un intervalle de temps. Cette contrainte du temp est prise en compte dans notre système de localisation en utilisant une version améliorée du mécanisme de transmission CSMA/CA dans les réseaux de capteurs.

Mots clés : Réseaux de capteurs sans fil, métrique RSSI, MinMax, Centroïde, CSMA/CA-adapté, Contiki, Cooja.

Abstract

In the absence of information on the position of a node of a wireless sensor network (WSN) , in the environment where they are deployed , the collected data may be of limited use , so the treaty issue is the location of a mobile . In this paper we propose and are implementing two methods to solve the problem , one that is the centroid method using the positions of anchors where their positions known and MinMax that uses RSSI measurement range between the mobile and the anchors calculate its position . We have implemented these methods in contiki 2.6, with the aim of improving the position estimation of the mobile in a time interval . This constraint temp is reflected in our localization system using an enhanced version of the transmission mechanism CSMA/CA in sensor networks .

Keywords : wireless sensors Networks, RSSI metric , MinMax , centroïde , CSMA / CA-adapted , Contiki , Cooja.