Université Abou Bekr Belkaid جامعة أبي بكر بلقايد
تلمسان الجزائر

République Algérienne Démocratique et Populaire
Université Abou Bakr Belkaid– Tlemcen
Faculté des Sciences
Département d'Informatique

Mémoire de fin d'études

Pour l'obtention du diplôme de Master en Informatique

Option : *Réseau système distribué (R.S.D)*

Thème

Implémentation et évaluation des performances d'un algorithme de localisation basé sur la connectivité des nœuds dans les réseaux de capteurs

Réalisé par :

- **BRAHIMI Sabiha**
- **AKBI Leyla**

Présenté le 16 Juin 2016 devant le jury composé de MM.

- *BENAMMAR Abdelkrim* *(Président)*
- *DIDI Fedoua* *(Examinateur)*
- *BEKARA Chakib* *(Examinateur)*
- *LABRAOUI Nabila* *(Encadreur)*

Année universitaire : 2015-2016

Remerciements

Nous tenons à remercier en premier lieu « **Mme LABRAOUI** », notre encadreur pour sa disponibilité, ses idées, ses conseils, son suivi et ses encouragements qui nous ont permis de mener à bien ce projet de fin d'études.

Nous remercions particulièrement « **Melle Mesmoudi Asma** » qui nous a beaucoup aidé et soutenue pour réaliser ce travail, sa patience, ses conseils et ses encouragements prodigués tout au long de ce mémoire.

Nous remercions les membres du jury Mrs Benammar, Bekara et Mme Didi d'avoir examiné notre travail.

Nos sincères remerciements vont également à tous les enseignants de notre département.

Dédicaces

Je dédie ce travail, à tous ceux qui me sont chers, A mes parents pour leur patience infinie, leur soutien sans faille et leur amour sans limite. Ce mémoire est autant le fruit de leurs efforts que de mon travail.

A mon mari pour son soutien, sa compréhension, son attention, sa patience et ses encouragements.

A ma sœur, mon frère, ma belle-famille, mes grands-parents et mes amis

Leyla !!

Je dédie ce travail à mes chers parents qui m'ont toujours encouragée à aller au bout de mes idées, qui m'ont appris à travailler honnêtement, et m'ont réalisé tous les moyens afin d'apprendre. Je vous aime

A ma sœur, mes frères, mes belles sœurs, mes amis et tous ceux qui m'ont chers

Sab !!

Table des matières:

Résumé

Les réseaux de capteurs sans fil continuent de constituer sans doute un développement technologique majeur. Mais en l'absence d'information sur la position des nœuds 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. Le problème traité dans ce mémoire concerne l'auto-localisation de chacun de ces nœuds à partir de mesures de portée inter-capteurs telles que les valeurs RSSI, et l'emplacement de quelques capteurs dits ancres dont la position est connue. Alors que l'objectif de notre travail est l'implémentation de l'un des algorithmes de localisation range-free nommé REAL.

Mots clés : Réseaux de capteurs sans fil, auto-localisation, RSSI, range-free, ancre, REAL.

Abstract

Wireless sensor networks is still undoubtedly a major technological development, but in the Absence of Location Information. of nodes in a wireless sensor network, in the environment where they are deployed, the collected data may become valueless. In this memoir, we address the problem of self-localization of each of these nodes from inter-sensor measurements such as RSSI, and the positions of few sensors called anchors whose position is known.

So the objective of our work is the implementation of one of the range free localization algorithms called REAL.

Keywords : wireless sensors Networks, auto-localization, RSSI ,range-free, anchor, REAL.

الملخص

شبكات الاستشعار اللاسلكية لا تزال بلا شك في تطور التكنولوجي كبير, ولكن في غياب اي معلومة حول تمركز العقد في شبكات الاستشعار اللاسلكية ضمن البيئة التي تم نشرها فيها قد تصبح البيانات التي تم جمعها دون جدوى, المشكل المعالج في هذه المذكرة يخص التموضع الذاتي لكل عقدة من خلال مقاييس المدى بين أجهزة الاستشعار مثل ISSR وموقع بعض أجهزة الاستشعار ذات الموقع المحدد ولذلك فإن الهدف من عملنا هو تنفيذ واحد من مجموعة خوارزميات

المدى الحر تسمى LAER.

Liste des Figures

Liste des tableaux

Introduction générale

Les réseaux de capteurs sans fil sont considérés comme un type spécial de réseaux ad hoc. Ils apportent une perspective intéressante : celle de réseaux capables de s'auto-configurer et de s'auto-gérer sans qu'ils aient besoin d'interventions humaines. Les nœuds sont généralement déployés de manière aléatoire à travers une zone géographique, appelée zone d'intérêt.

Le déploiement rapide, le coût réduit et la tolérance aux pannes des réseaux de capteurs sont des caractéristiques qui les rendent un outil appréciable dans plusieurs domaines d'application, tels que les activités militaires et le domaine biomédical.

La localisation est un aspect primordial dans de nombreuses applications des réseaux de capteurs et c'est l'une parmi les principaux problèmes dans ce type de réseaux et nombreuses sont les solutions qui ont été proposées pour le résoudre.

Une première solution basée sur les satellites consisterait à équiper tous les capteurs de systèmes de positionnement tel que le GPS. Cependant, cette méthode serait trop coûteuse sur le plan énergétique en raison de la consommation excessive d'énergie nécessaire aux GPS.
Une solution alternative plus efficace se limiterait à n'équiper qu'un faible nombre de capteurs avec des GPS, qui seront appelés « ancres ». Les autres capteurs, appelés nœuds ou non-ancres, sont localisés en échangeant des informations avec les ancres.
Une deuxième solution basée sur l'auto-localisation représente une solution alternative au système basé sur GPS.

La problématique de la localisation dans les WSN a largement été étudiée, et plusieurs approches dites range-based et range-free ont été proposées. Dans ce mémoire nous nous sommes focalisé sur les algorithmes range-free et plus spécialement sur un nouveau système de localisation par zone qui estime la zone de résidence du nœud inconnu, en utilisant uniquement la position de certains nœuds de référence, appelés ancres.

Cet algorithme est fait pour surmonter les lacunes de certains algorithmes qui souffrent des mauvaises décisions prises au sujet de la présence à l'intérieur d'une zone donnée en introduisant un nouveau régime fondé sur la superficie pour les réseaux de capteurs sans fil (WSN), appelé REAL [19].

Afin d'aborder l'ensemble de ces étapes, nous avons organisé ce rapport en 3 chapitres :

Le chapitre 1 : nous présentons une description générale des réseaux de capteurs sans fil ainsi que leurs caractéristiques, contraintes et spécificités.

Le chapitre 2 : est consacré à la problématique de la localisation, les classifications des algorithmes de localisation et à la description de l'algorithme REAL.

Le chapitre 3: constitue le cœur de notre travail, dans ce chapitre nous présentons la plateforme TinyOS et langage NesC, l'implémentation et évaluation de l'algorithme REAL.

Chapitre I

Généralité sur les réseaux de capteurs Sans fil

II.3 architecture logicielle

TinyOS est un système d'exploitation open-source conçu pour des réseaux de capteurs sans fil. Il respecte une architecture basée sur une association de composants, réduisant la taille du code nécessaire à sa mise en place. Cela s'inscrit dans le respect des contraintes de mémoires qu'observent les réseaux de capteurs. Pour autant, la bibliothèque de composant de TinyOS est particulièrement complète puisqu'on y retrouve des protocoles réseaux, des pilotes de capteurs et des outils d'acquisition de données. L'ensemble de ces composants peut être utilisé tel quel, il peut aussi être adapté à une application précise. En s'appuyant sur un fonctionnement événementiel, TinyOS propose à l'utilisateur une gestion très précise de la consommation du capteur et permet de mieux s'adapter à la nature aléatoire de la communication sans fil entre interfaces physiques. [3]

II.4 Exemple de capteur sans fil

Il existe dans le monde plusieurs fabricants de capteurs. Nous citerons Crossbow, Cisco, Dalsa, EuroTherm, et Sens2B. Parmi ces capteurs, il existe quelques-uns qui sont capables de varier la puissance du signal émis afin d'élargir/réduire le rayon de communication et en conséquence la zone de communication. La Figure I.2 montre un capteur intelligent TELOSB fabriqué par Crossbow. [4]

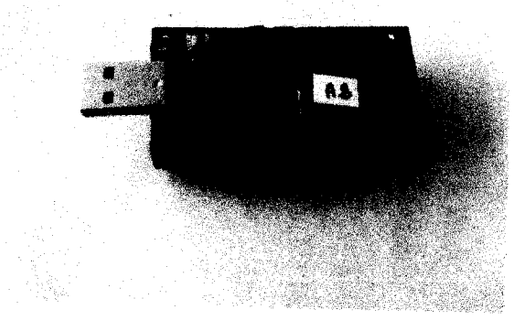


Figure I. 2: Capteur Telosb

II.2 Architecture d'un capteur

Le nœud de capteur est composé principalement de quatre unités: l'unité de captage, l'unité de calcul, l'unité de transmission et une source d'énergie (voir la Figure I.1). Il peut contenir également des unités supplémentaires tel que le système de localisation (GPS) pour connaitre l'emplacement précis du nœud,...

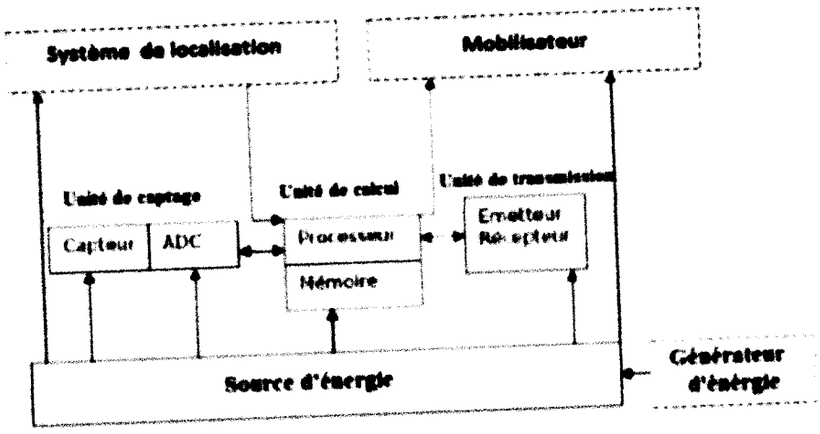


Figure I. 1: Architecture d'un capteur

• **Unité de captage:** elle contient deux sous unités, la première permet la collecte des phénomènes physiques observés telle que la température, et la deuxième convertit le signal en signal numérique (ADC) pour être envoyé à l'unité de traitement.

• **Unité de calcul:** Elle est composée d'une mémoire de stockage et d'un processeur. Elle possède en plus deux interfaces :

-La première, liée à l'unité de captage pour la réception des données collectées

- La seconde, liée à l'unité de transmission pour la transmission des données traitées.

• **Unité de transmission:** Elle est responsable de la transmission et la réception des données via un support de communication radio. Ce dernier peut être de type optique (comme dans les capteurs Smart Dust), où de type radio fréquence (MICA2). On note que la transmission consomme beaucoup d'énergie par rapport à l'unité de calcul.

• **Source d'énergie**: elle est responsable de l'alimentation des différentes unités et elle réduit les dépenses, par exemple en mettant en veille les composants inactifs. [2]

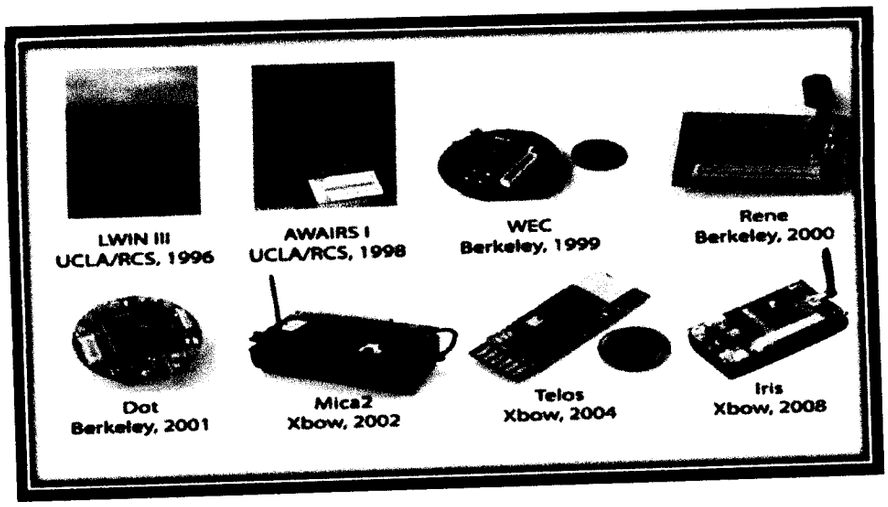


Figure I. 3 : Quelques exemples de capteurs [1]

II.5 Caractéristiques principales d'un capteur

Deux entités sont fondamentales dans le fonctionnement d'un capteur : l'unité d'acquisition qui est le cœur physique permettant la prise de mesure et l'unité de communication qui réalise la transmission de celle-ci vers d'autres dispositifs électroniques. Ainsi, fonctionnellement chaque capteur possède un rayon de communication (Rc) et un rayon de sensation (Rs). La Figure I.4 montre les zones définies par ces deux rayons pour le capteur A. La zone de communication est la zone où le capteur A peut communiquer avec les autres capteurs (le capteur B). D'autre part, la zone de sensation est la zone où le capteur A peut capter l'événement. En effet, pour qu'un capteur ait une portée de communication suffisamment grande, il est nécessaire d'utiliser un signal assez puissant. Cependant, l'énergie consommée serait importante. [4]

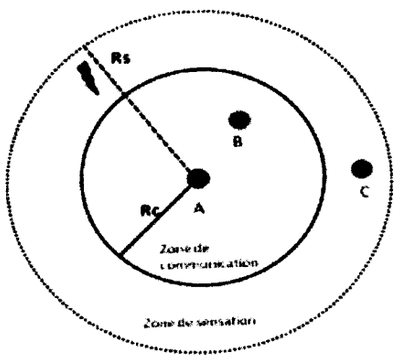


Figure I. 4 : Rayons de communication et de sensation d'un capteur [4]

II. Les réseaux de capteurs sans fil (RCSF)

II.1. Description des réseaux de capteurs

Les réseaux de capteurs sans fil (WSN) sont un type particulier de réseau Ad-hoc, dans lesquels les nœuds sont des « capteurs intelligents ». Ils se composent généralement d'un grand nombre de capteurs communicants entre eux via des liens radio pour le partage d'information et le traitement coopératif. Dans ce type de réseau, les capteurs échangent des informations par exemple sur l'environnement pour construire une vue globale de la région contrôlée, qui est rendue accessible à l'utilisateur externe par un ou plusieurs nœud(s). Les données collectées par ces capteurs sont acheminées directement ou via les autres capteurs de proche en proche à un « point de collecte », appelé station de base (ou SINK s'il s'agit d'un nœud). Cette dernière peut être connectée à une machine puissante via internet ou par satellite. En outre, l'utilisateur peut adresser ses requêtes aux capteurs en précisant l'information d'intérêt [5]

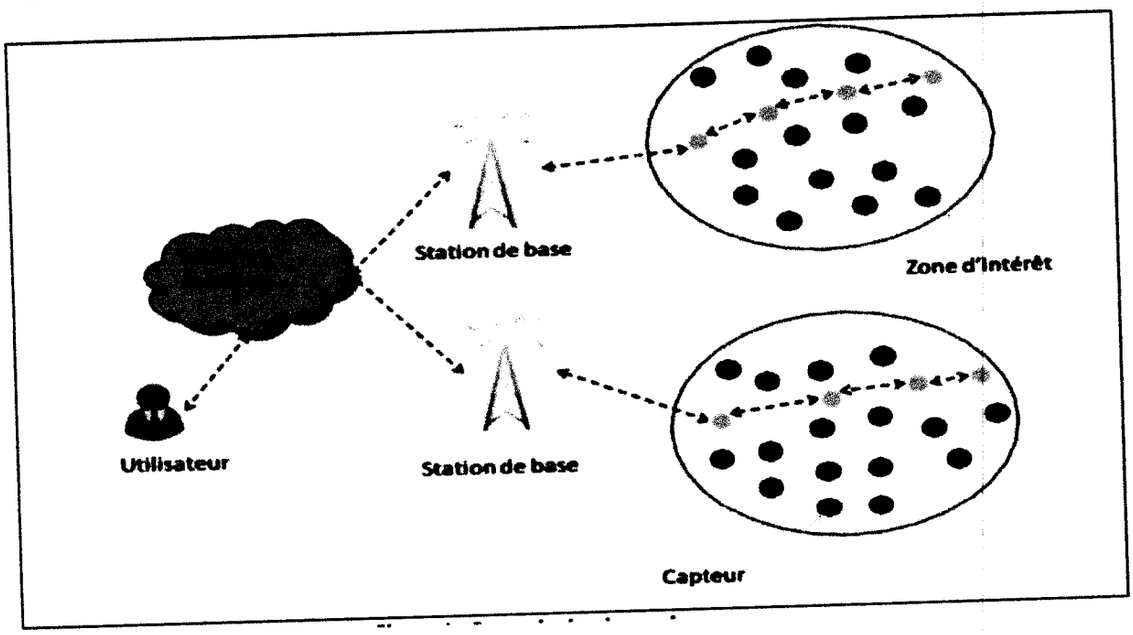


Figure I. 5: Exemple de réseau de capteur [5]

Un exemple de réseaux de capteurs est fourni dans la figure I.5 : les capteurs sont déployés d'une manière aléatoire dans une zone d'intérêt, et une station de base, située à l'extrémité de cette zone, est chargée de récupérer les données collectées par les capteurs. Lorsqu'un capteur détecte un événement pertinent, un message d'alerte est envoyé à la station de base par le

biais d'une communication entre les capteurs. Les données collectées sont traitées et analysées par des machines puissantes. Les réseaux de capteurs viennent en soutien de l'environnement et de l'industrie grâce aux récents développements réalisés dans le domaine des techniques sans fils. Depuis quelques décennies, le besoin d'observer et de contrôler des phénomènes physiques tels que la température, la pression ou encore la luminosité est essentiel pour de nombreuses applications industrielles et scientifiques.

II.2. Similarité et différence entre (RCSF) et réseaux Ad-hoc (MANETs)

Les réseaux de capteurs sont considérés comme un type particulier des réseaux ad-hoc et sont significativement différents des réseaux MANET (Mobile Ad-hoc NETwork)

Traditionnels. Ce tableau résume les similarités et les différences entre les réseaux de capteurs sans fil et les MANETs [6] :

Réseau de capteurs sans fils	**Réseau MANET(Ad-Hoc)**
1 .Utilisation d'un médium sans fil.	1. Utilisation d'un médium sans fil.
2. Réseau auto-configurable	2. Réseau auto-configurable
3. Topologie dynamique	3. Topologie dynamique
4.Mode de transmission: many to one	4.Mode de transmission: one to one (any to any)
5. Utilisation du broadcast	5. Communication point à point
6. La mobilité des nœuds est restreinte.	6. Mobilité des nœuds.
7. Grand nombre de nœuds (de l'ordre de mille) nœuds n'ayant pas tous un ID	7. Nombre de noeuds moyen (de l'ordre de cents) Notion d'ID
8. Des petits nœuds plus susceptibles aux pannes, avec moins de capacité de traitement et de stockage.	8. Des nœuds ayant plus de capacité de traitement et de stockage.
9. Noeuds collaborent pour remplir un objectif Commun	9. Chaque noeud a son propre objectif
10. Portée radio est de l'ordre de 40 m	10. Portée radio est dans l'environs de 250 m
11. Energie (facteur déterminant)	11. Débit majeur

Tableau I. 1: Similarités et les différences entre (RCSF) et réseaux Ad-hoc [6].

II.3. Les principales caractéristiques des réseaux de capteurs sans fil

- **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. Les hôtes mobiles sont responsables d'établir et de maintenir la connectivité du réseau d'une manière continue.
- **taille importante** : un réseau de capteurs peut contenir des milliers de nœuds.
- **interférences :** les liens radio ne sont pas isolés, deux transmissions simultanées sur une même fréquence, ou utilisant des fréquences proches, peuvent interférer.
- **topologie dynamique** : les capteurs peuvent être attachés à des objets mobiles qui se déplacent d'une façon libre et arbitraire rendant ainsi la topologie du réseau fréquemment changeante.
- **sécurité physique limitée** : les réseaux de capteurs sans fil sont plus touchés par le paramètre de sécurité que les réseaux filaires classiques. Cela se justifie par les contraintes et limitations physiques qui font que le contrôle des données transférées doit être minimisé.
- **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 nœud est limitée.
- **contrainte d'énergie, de stockage et de calcul** : la caractéristique la plus critique dans les réseaux de capteurs est la modestie de ses ressources énergétiques car chaque capteur du réseau possède de faibles ressources en termes d'énergie (batterie). Afin de prolonger la durée de vie du réseau, une minimisation des dépenses énergétiques est exigée chez chaque nœud. Ainsi, la capacité de stockage et la puissance de calcul sont limitées dans un capteur [6].

II.4. Domaines d'application des réseaux de capteurs:

Le concept de réseaux de capteurs sans fil est basé sur une simple équation :

« Capteurs + Processeur + Radio = Une centaine d'applications potentielles » [7]

La diminution de taille et de coût des micro-capteurs, l'élargissement de la gamme des types de capteurs disponibles (thermique, optique, vibrations...) et l'évolution des supports de

- **Applications médicales :** 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, etc.). Ils peuvent aussi faciliter le diagnostic de quelques maladies en effectuant des mesures physiologiques telles que : la tension artérielle, battements du cœur, 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. D'autre part, ces réseaux peuvent détecter des comportements anormaux (chute d'un lit, choc, cri, etc.) chez les personnes dépendantes (handicapées ou âgées) ;
- **Applications commerciales :** Il est possible d'intégrer des capteurs au processus de stockage et de livraison dans le domaine commercial. Le réseau ainsi formé pourra être utilisé pour connaître la position, l'état et la direction d'un paquet. Il devient alors possible pour un client qui attend la réception d'un paquet, d'avoir un avis de livraison en temps réel et de connaître la localisation actuelle du paquet. Pour les entreprises manufacturières, les réseaux de capteurs permettront de suivre le procédé de production à partir des matières premières jusqu'au produit final livré. Grâce aux réseaux de capteurs, les entreprises pourraient offrir une meilleure qualité de service tout en réduisant leurs coûts [8].

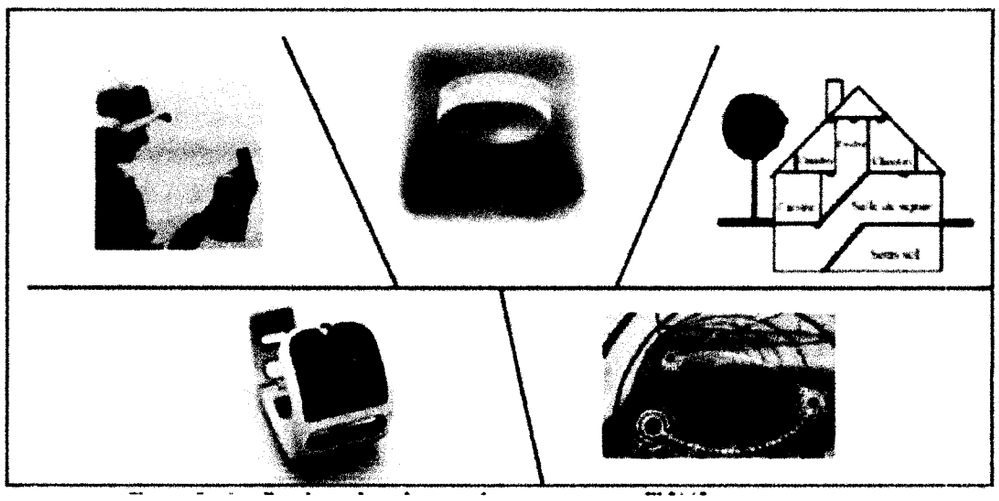


Figure I. 6: Application des réseaux de capteurs sans fil [8]

II.5. Communication dans les réseaux de capteurs

II.5.1. Architecture de communication basée sur le modèle OSI

Le modèle de communication comprend cinq couches qui ont les mêmes fonctions que celles du modèle OSI ainsi que trois couches pour la gestion d'énergie, la gestion de la mobilité et la gestion des tâches[9].

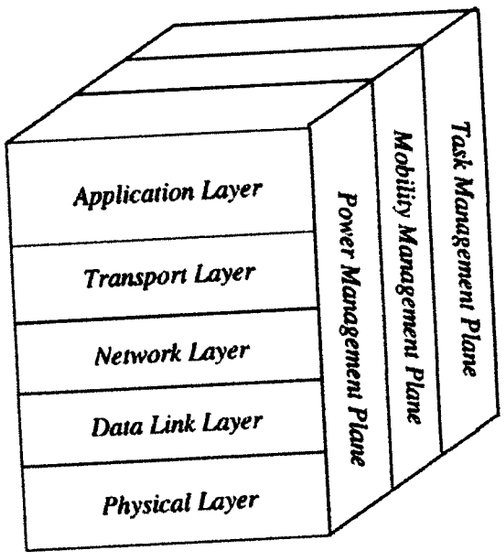


Figure I. 7: communication dans les réseaux de capteurs [9]

- Couche physique : Matériels pour envoyer et recevoir les données
- Couche liaison : Gestion des liaisons entre les nœuds et les stations de base, contrôle d'erreurs
- Couche réseau : Routage et transmission des données
- Couche transport : Transport des données, contrôle de flux
- Couche application : Interface pour les applications au haut niveau
- Plan de gestion d'énergie : Contrôle l'utilisation d'énergie
- Plan de gestion de mobilité : Gestion des mouvements des nœuds
- Plan de gestion de tâche : Balance les tâches entre les nœuds afin d'économiser de l'énergie.

II.5.2. Technologies de la communication dans les réseaux de capteurs

- **Bluetooth / IEEE 802.15.1**

Bluetooth est une spécification de l'industrie des télécommunications. Elle utilise une technique radio courte distance destinée à simplifier les connexions entre les appareils électroniques. Malheureusement, un grand défaut de cette technologie est sa trop grande consommation d'énergie [10].

- **ZigBee / IEEE 802.15.4**

ZigBee est une norme de transmission de données sans fil permettant la communication de machine à machine. Zigbee offre des débits de données moindres, mais sa très faible consommation électrique et ses coûts de production très bas en font une candidate idéale pour la domotique ou les matériels de type capteur, télécommande ou équipement de contrôle dans le secteur industriel.

- **Dash 7 / ISO/IEC 18000-7**

Dash7 est une nouvelle technologie de réseaux de capteurs sans fil en utilisant la norme ISO/IEC 18000-7. Sa consommation électrique est très faible, la durée de vie de batterie peut arriver à plusieurs ans.

Sa distance de communication est 2km. Elle fournit une faible latence pour le suivi des objets en mouvement, un protocole petit pile, des supports de capteurs et de sécurité et un débit de transmission allant jusqu'à 200kbits/s.

III. Conclusion:

Nous avons présenté dans ce chapitre les capteurs ainsi que leurs architectures matériels et logiciels. Nous avons défini par la suite les réseaux de capteurs et leurs domaines d'application. Dans le chapitre suivant, nous présenterons la localisation conçus pour les RCSF et la classification des algorithmes de localisation.

Chapitre II

La localisation dans les réseaux de capteurs

- **Le système Galileo**

Galileo désigne le système européen de navigation par satellite, initiative lancée par l'Union européenne et l'Agence spatiale européenne (ESA). Ce système mondial assurera une complémentarité avec le système actuel GPS. Satellite Galileo repose sur une constellation de trente satellites et des stations terrestres permettant de fournir des informations concernant leur positionnement à des usagers de nombreux secteurs.

Avec Galileo, l'Europe possédera son propre système mondial de navigation par satellite et s'affranchira des services offerts par le GPS américain qui comportent de nombreuses restrictions. Galileo fournira des services de localisation précis, sécurisés et certifiés à l'échelle du globe. Il sera placé sous le contrôle d'autorités civiles au contraire de son homologue américain, militaire lui [12].

- **Le système Glonass**

Signifie en russe GLObal'naya NAvigatsionnaya Sputnikovaya Sistema soit système mondial de navigation par satellite a débuté en 1976 avec pour but une couverture mondiale en 1991. C'est un système de positionnement développé par l'actuelle Union Soviétique et contrôlé pour le gouvernement russe par l'agence spatiale russe. Il est une alternative et complémentaire du GPS américain et du futur Galileo européen [13].

- **Le système Beidou**

Le système de navigation et de positionnement par satellite chinois Beidou (BDS), signifiant « le Grande Ourse », consiste en deux constellations de satellites et a pour objectif de permettre l'indépendance de la Chine vis-à-vis des systèmes de navigations étrangers (GPS américain, Galileo européen et GLONASS russe).L'évolution future qui rendra le système mondiale.il sera totalement opérationnel en 2020 [14].

- **Le system IRNSS (Indian Regional Navigation Satellite System)**

Est une proposition de système de positionnement par satellites qui serait construit et contrôlé par le gouvernement Indien. Un but de contrôle complet de la part du gouvernement Indien a été cité, ce qui implique que toutes les parties du projet soit construites en Inde. Le système proposé consisterait en 7 satellites et une partie au sol (grounds segment). Trois des satellites seront placés en orbite géostationnaire et les quatre autres en orbite géosynchrone inclinée de 29 degrés par rapport au plan de l'équateur [13].

Parmi ces systèmes satellitaires, le GPS est le plus utilisé dans le monde. Dans les réseaux de capteurs sans fil, .cette solution consiste à accorder chaque nœud par un GPS pour spécifier la localisation des nœuds. Mais il est disponible seulement en extérieur « outdoor », et encore si aucun obstacle ne vient obstruer le champ de vue des récepteurs : le fonctionnement sous un feuillage dense, ou dans des villes aux rues étroites, n'est pas possible, où seulement dans de très mauvaises conditions. De plus il est particulièrement coûteux, tant en ce qui concerne le matériel qui est dupliqué en nombreux exemplaires dans un réseau à forte densité de capteur.

II.2. les systèmes basés sur l'auto-localisation

Le système d'auto-localisation représente une solution alternative au système basé sur GPS. Dans ce système les nœuds blind doivent eux-mêmes, déterminer leurs positions, à travers des techniques de localisation.

Dans ce système le terme nœuds ancres désigne les capteurs qui peuvent obtenir leurs localisations par un placement manuel ou par GPS. Et le terme nœuds blind désigne les nœuds qui ne connaissent pas leurs informations de localisation.

III. L'auto-localisation dans les RCSF

Le système d'auto-localisation comporte trois composants (voir Figure II.1):

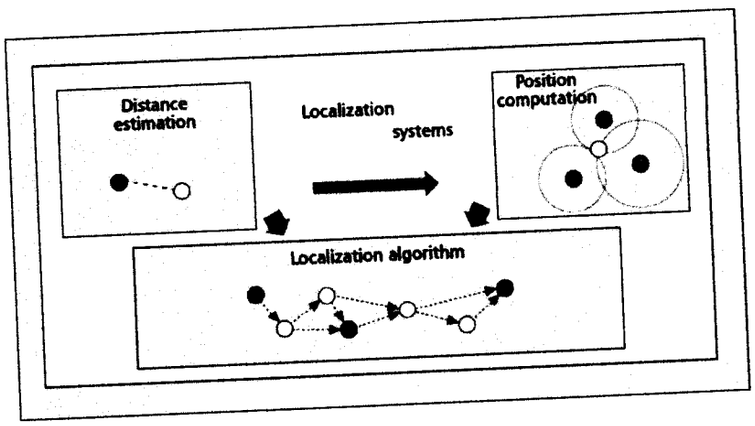


Figure II. 1 : Division des systèmes de localisation en trois composantes distinctes [28]

III.1. Estimation de la distance/angle

✓ **Temps d'arrivée**

La technologie ToA (Time of Arrival) suppose que les nœuds du réseau sont synchrones. 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

I. Introduction

La localisation dans les réseaux de capteurs déployés de manière aléatoire consiste à déterminer les coordonnées géographiques des différents capteurs. La localisation des nœuds est nécessaire, non seulement pour localiser les différents évènements survenus dans la zone surveillée, mais aussi pour le développement de protocoles de routage de l'information récoltée, pour la couverture de la zone d'intérêt, pour l'agrégation des données, etc. Elle est la première tâche exécutée par les nœuds après leur déploiement.

La connaissance des positions des capteurs dans l'environnement surveillé est souvent indispensable pour une grande majorité des applications (militaires, suivis des animaux, ...), afin de pouvoir déterminer l'origine des événements détectés. « Où ? » est la question qui suit immédiatement la détection d'un événement (par exemple, où est le feu ?). En outre, la localisation peut être utilisée dans les protocoles de routage géographique dans les réseaux à grande échelle, en transmettant les données seulement dans la direction de la destination.

Dans ce chapitre, nous présenterons des méthodes de localisations et les étapes pour construire un système de localisation conçu essentiellement pour les réseaux de capteurs sans fil afin de permettre aux capteurs de calculer leurs positions. Ensuite nous présentons les algorithmes de localisation range-based et range-free. Et on finira par décrire un nouvel algorithme de type range-free qui se base sur les mêmes données que le plan APIT, nommé REAL.

II. Les principaux systèmes de localisation

II.1. Les systèmes basés sur les satellites

- **Le GPS (Global Positioning System)**

Sigle signifiant *Global Positioning System*. Système de géolocalisation par satellite. Le réseau de 24 satellites (plus 4 satellites en réserve) actuellement en fonctionnement, développé par l'armée américaine, est mis à disposition des civils. Il permet de déterminer les coordonnées géographiques de n'importe quel point situé à la surface du globe. Sa précision peut atteindre 1 mètre. Le GPS s'utilise en association avec une carte pour se repérer et se positionner : randonnées, voile, trek... [11].

utilisée par le système GPS (Global Positioning System). Lorsque les nœuds ne sont pas synchrones, 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. Cela suppose que les nœuds du réseau ont un temps de traitement du signal identique [16].

✓ **Différence des temps d'arrivée**

La technologie TDoA (Time Difference of Arrival) requiert d'autres matériels et elle n'est pas basée sur le signal radio seulement. En effet, chaque nœud devrait être équipé d'un haut-parleur et d'un microphone. TDoA 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 :

– un émetteur envoie des signaux de natures différentes (par exemple, l'ultrason, l'onde radio,...) à un récepteur ;

– un récepteur reçoit des signaux d'une 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) [16].

✓ **Puissance du signal**

Dans les réseaux de capteurs sans fil, chaque capteur est équipé d'une radio. La question est de savoir comment utiliser la radio pour aider à localiser le réseau ? La technologie RSSI (Received Signal Strength Indicator) propose une solution élégante pour l'estimation des distances dans les réseaux de capteurs. Elle considère la perte de puissance d'un signal entre son émission et sa réception. Cette perte varie en fonction de la 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 [16].

En pratique, les mesures par RSSI contiennent des erreurs de l'ordre de quelques mètres. Ce bruit se produit parce que la propagation des ondes radio tend à être fortement non-uniforme dans des environnements réels. Par exemple, la radio se propage différemment sur l'asphalte que sur l'herbe. Des obstacles physiques tels que les murs, meubles, etc, reflètent et

absorbent les ondes radio. Par conséquent, la précision sur la distance en utilisant la force du signal n'est pas bien démontrée par rapport à d'autres techniques comme la « TDoA » [16].

✓ **Angle d'arrivée**

La technologie AoA (Angle of Arrival) consiste à calculer l'angle formé entre deux capteurs. La direction (l'angle) est généralement recueillie par la radio et un ensemble de microphones, qui permettent à un nœud écouteur de déterminer sa direction par rapport à l'émetteur. Il est également possible de la recueillir par le moyen d'une communication optique.

Dans cette technique, on a besoin de plusieurs (3-4) microphones spatialement séparés qui entendent un seul signal transmis. En analysant la phase ou la différence entre les temps d'arrivée du signal aux différents microphones, il est possible de découvrir l'angle d'arrivée du signal. Ces méthodes peuvent obtenir une précision de l'ordre de quelques degrés (Priyantha, et al., 2001). Malheureusement, elles exigent plus de matériels (un haut-parleur et plusieurs microphones) que la technique « TDoA », donc elle est plus coûteuse et les nœuds tendent à être plus volumineux. Il existe peu d'algorithmes de localisation qui sont basés sur la technique « AoA», même si plusieurs d'entre eux sont capables de l'utiliser quand elle est présente.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 [16].

III.2. Dérivation des positions

La dérivation des positions consiste à calculer les positions finales de chaque nœud capteur en utilisant un des algorithmes de localisation. Chaque algorithme utilise une méthode de calcul qui dépend de la technique d'estimation de distance utilisée. Nous classifions ces méthodes en trois catégories :[16]

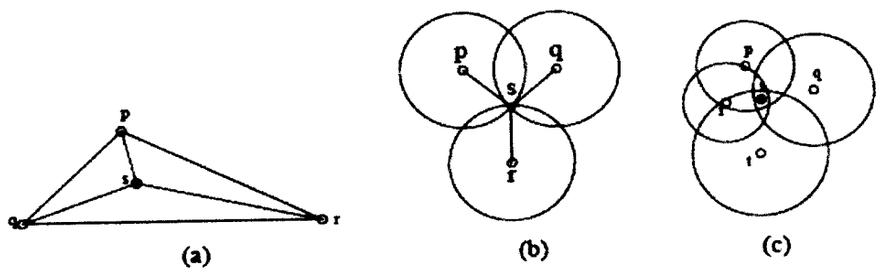


Figure II. 2: (a) triangulation, (b) trilatération, (c) multilatération

- **La trilatération** (Langendoen, et al., 2003) est la méthode la plus simple pour déterminer les positions des capteurs en utilisant la géométrie des triangles. Elle est fondée sur le même principe qu'un système GPS. Elle consiste à s'appuyer sur trois points de référence, c'est à dire des nœuds dont on connaît la position (les ancres), et sur les distances qui les séparent du nœud dont on cherche à estimer la position. Cette dernière correspond alors au point d'intersection des trois cercles [16].
- **La triangulation** est une technique permettant de déterminer la position d'un point en mesurant les angles entre ce point et d'autres points de référence dont la position est connue (les ancres), et ceci plutôt que de mesurer directement la distance entre les points. Ce point peut être considéré comme étant le troisième sommet d'un triangle dont on connaît deux angles et la longueur d'un côté [16].
- **La multilatération** (Langendoen, et al., 2003) a le même principe que la trilatération en utilisant plus que trois points de référence (ancres). La position d'un nœud est calculée en résolvant l'intersection de plusieurs hyperboles basées sur la différence des temps d'arrivée TDoA. Soit une cible i, connaissant les positions (xa, ya) de m ancres ($1 \leq a \leq m$) ainsi que les distances dia, où dia représente la distance euclidienne entre i et l'ancre a. Ayant ces informations et pour calculer la position (xi, yi) de la cible i nous formons le système suivant :

$$(x_1 - x_i)^2 + (y_1 - y_i)^2 = {d_{i1}}^2$$
$$(x_2 - x_i)^2 + (y_2 - y_i)^2 = {d_{i2}}^2$$
$$(x_3 - x_i)^2 + (y_3 - y_i)^2 = {d_{i3}}^2$$

$$\vdots$$

$$(x_m - x_i)^2 + (y_m - y_i)^2 = {d_{im}}^2$$

Ce système peut être linéarisé en soustrayant la dernière équation des m−1 équations précédentes. En réordonnant les termes, nous obtenons un système d'équations linéaires. Ayant des erreurs dans les estimations de distances, il paraît qu'une solution exacte pour un tel système d'équations est presque impossible. La solution la plus proche de la solution exacte c'est au sens des moindres carrés [16].

III.3. Algorithmes de localisation

Les algorithmes de localisation sont classés selon plusieurs critères. Tel que : l'organisation de calcule, avec ou sans infrastructure et la dépendance des techniques de mesure. Ces derniers est le critère le plus important dans notre travail, nous classifions cette dépendance dans ce qui suit.

IV. Classification des algorithmes de localisation

Nous distinguons deux façons d'implémenter un algorithme de localisation selon le critère de la dépendance des techniques de mesures : Les algorithmes range-based et les algorithmes range–free.

IV.1. Les algorithmes range-based

Les méthodes range-based utilisent les technologies ToA , RSSI, AoA afin de mesurer les distances ou les angles entre deux capteurs voisins. Grâce à cette capacité de mesure, ces méthodes sont plus précises. Mais parmi ces inconvénients que certaine technologies comme (ToA ,DoA ,AoA) sont très couteux ou sensible à l'environnement comme RSSI .Parmi les algorithmes de localisation range-based, nous citons :

Dynamic fine-grained, APS using AoA et **MDS** [17].

IV.2. Les algorithmes range-free

Ces méthodes ne convertissent jamais de mesure prise en distance ou angle. Elles utilisent d'autres informations telles que la connectivité et le nombre de sauts. Elles proposent des calculs plus ou moins complexes pour évaluer la position. Dans cette famille des méthodes, la position n'est pas trop précise mais elle est adapté pour certaine application, nous citons : la détection des feux de forêt.

V. Exemple d'un algorithme de localisation de type range-free

Les méthodes Range-based sont précises. Cependant, elles sont soit couteuses ou sensibles aux bruits et aux obstacles. Inversement, les méthodes range-free sont certes moins précises mais elle permet de calculer les positions partir de simple déduction. Donc nous nous somme focaliser sur les méthodes range free.

V.1. Motivation

Les algorithmes de localisation de type range-free les plus connu sont DV-Hop [23], Amorphous [24], Centroid [25] et APIT [26]. D'ailleurs, les expérimentions présentées dans la littérature ont montré le protocole APIT présente de meilleures performances lorsqu'un déploiement aléatoire est considéré et un faible coût de communication est désiré. Cependant, l'inconvénient majeur d'APIT est que son test d'appartenance peut échouer dans certain cas. En effet, il est sensible au placement des capteurs voisins.

En fait, un nouveau protocole de localisation de type range-free, appelé REAL « REliable Area based Localisation algorithme », a été récemment proposé par la publication REAL [19]. Son objective est de surpasser la qualité de localisation du protocole APIT, tout en gardant les même donné utilisé tel que : combinaisons de trois ancres voisine et les valeurs des puissances des signaux reçus. Les résultats présenté dans REAL **[19],** ont montré que le protocole REAL garantis de très bonne performance et surpasse de loin les protocoles APIT [26]. Pour ces multiples raisons, nous nous somme intéressées à l'implémentation de ce protocole.

V.2. Description de l'algorithme REAL

Dans ce chapitre, nous présentons l'algorithme REAL [19]. Dans cet algorithme les capteurs déterminent leurs emplacements en fonction des informations liées aux ancres. Dans un premier temps, un capteur choisit trois ancres de toutes les ancres voisines et teste si elle réside à l'intérieur de la forme géométrique spécifique formé par ces trois ancres ou non. Cette approche répète ce test avec toutes les combinaisons possibles des trois ancres voisines.
À ce stade, REAL calcule le centre de gravité (COG ; Center Of Gravity) de l'intersection de toutes les formes obtenu, à l'intérieur duquel un nœud doit résider afin d'estimer sa position [19].
L'algorithme REAL comprend cinq étapes, ces étapes sont décrites en détail dans ce qui suit :

V.2.1. Etape 1 : La détection des ancres voisines

Chaque ancre commence à diffuser un message beacon y compris ses informations de localisation. Les capteurs et les ancres peuvent recevoir ce message beacon quand ils seront dans la portée de communication les uns des autres. Chaque ancre construit son tableau de voisinage désigné par AN. Ce dernier comprend : les identificateurs et les positions des ancres

ainsi que les valeurs RSSI entre les ancres voisines, que nous désignons par $RSSI_{A_i,A_j}$. Ensuite, chaque ancre diffuse ses informations recueillies aux capteurs voisins.

V.2.2. Etape 2 : La Collecte des informations de localisation

Le capteur s collecte les informations nécessaires de toutes les ancres voisines, et il peut construire son tableau d'ancrage, notée AHs. Ce dernier comprend les identificateurs et les positions des ancres voisines, les valeurs RSSI entre le nœud blind et ses ancres voisines, désignée par $RSSI_{s,A_i}$et les valeurs RSSI des ancres entre eux notée $RSSI_{A_i,A_j}$qui est obtenu à partir du tableau NAs. En outre, le capteur s construit une autre liste contenant toutes les combinaisons possibles des trois ancres voisines. Elle est notée par ACs [19].

V.2.3. Etape 3 : La création de la zone de présence

Après avoir recueilli les informations des ancres voisines, le nœud inconnu peut déterminer sa zone de résidence. Pour les trois ancres voisines données, i.e.A_i (x_{A_i},y_{A_i}), $A_j(x_{A_j},y_{A_j})$ et $A_k(x_{A_k},y_{A_k})$, le capteur 's' sélectionne son nœud ancre le plus proche, qui a la plus grande valeur RSSI entre $RSSI_{s,A_i}$, $RSSI_{s,A_j}$et $RSSI_{s,A_k}$. nous supposons que A_i est l'ancre la plus proche, si et seulement si $RSSI_{s,A_i} > RSSI_{s,A_j}$ et $RSSI_{s,A_i} > RSSI_{s,A_k}$.

De plus , si $RSSI_{s,A_j} < RSSI_{A_i,A_j}$et $RSSI_{s,A_k} < RSSI_{A_i,A_k}$, alors le capteur s doit résider en dehors de la forme géométrique définie par l'union des deux cercles C_{A_j} et C_{A_k}, à savoir CMU (A_j,A_k), où C_{A_j} est centré à A_j et avec un rayon égal à la distance $d_{i,j}$, entre A_j et A_i , et le cercle C_{A_k}, est centré au A_k avec un rayon égal à la distance $d_{i,k}$, entre A_k et A_i. Cette zone est appelé le complément de Combine Union Mode (Toute la surface du réseau, sauf la zone définit par l'union de deux cercles C_{A_j} et C_{A_k}) ; elle est désignée par $\overline{CMU}(A_j,A_k)$.

Cette zone de résidence initiale peut être réduite en traçant un troisième cercle C_{A_i} centré à l'ancre le plus proche A_i avec un rayon égal à la distance la plus longue, qui peut être soit la distance $d_{i,j}$ entre les deux ancres A_j et A_i, ou la distance $d_{i,k}$ entre les deux ancres A_k et A_i. Par conséquent, la zone de résidence finale est l'intersection de $\overline{CMU}(A_j,A_k)$ et C_{A_i} . Dans ce qui suit, l'algorithme 1 résume l'étape 3.

Algorithme 1 : La zone de présence du nœud **s**

1: Let **AHs** be the set of neighbor anchors for a sensor **s**.

2: Let **ACs** be the set of all possible three anchors' combinations.

3: **Z** be the area of the network

4: $R_{i,j,k}$ be the residence area of the sensor

5: **for** each **sub-set** $(A_i, A_j, A_k) \in$ ACs do

6: Let A_i be the nearest anchor **then**

7: **if** $RSSI_{s,A_j} < RSSI_{A_i,A_j}$ *and* $RSSI_{s,A_k} < RSSI_{A_i,A_k}$

8: $R_{i,j,k} = Z \setminus CMU(A_j, A_k)$

9: $R_{i,j,k} = R_{i,j,k} \cap C_{A_i}$

10: **end if**

11: **end for.**

V.2.4. Etape 4 : Le raffinement de la zone de présence

Le nœud inconnu peut estimer sa position comme étant le centre de gravité de la zone de présence finale. Cependant, ceci serait informatiquement très coûteux en calcul pour chaque capteur dans le réseau. Par conséquent, un algorithme «*grid-scan*» est suggéré. Le même algorithme a été utilisé par APIT. L'algorithme Grid Scan se compose de deux étapes :

1. Le capteur blind divise l'ensemble de la zone de réseau en un ensemble de cellules de grille de points équidistant.
2. Ensuite, il scanne les grilles et marque ceux qui appartiennent à la zone de résidence du capteur grilles valides. Ainsi, la zone de chevauchement est définie par les cellules de la grille qui ont le plus haut score. Ce processus est répété pour toutes les combinaisons possibles de trois ancres voisines La zone de présence finale du capteur est représentée par les points de grille de valeur maximale dans le tableau de pointage de la grille. Cette zone correspond à la zone d'intersection de toutes les zones de présences possible du capteur.

V.2.5. Etape 5 : L'estimation de la position

Le nœud blind détermine son emplacement comme étant le centre de gravité des points de grille représentant la zone de résidence finale. Alors les coordonnées du nœud est calculée par :

$$(x_{est}, y_{est}) = \left(\frac{1}{T} \sum_{q=1}^{T} x_q , \frac{1}{T} \sum_{q=1}^{T} y_q \right)$$

Où T est le nombre de points de grille de la zone de présence, (x_q , y_q) Sont les coordonnées de centre de cellule de grille (i.e. les points de grille).

V.3. Exemple illustratif de l'algorithme REAL

Dans cette section un exemple est présenté pour expliquer l'algorithme proposé REAL :

- Les ancres voisines A_1, A_2 et A_3 commencent par diffuser le message beacon. Chaque ancre reçoit le beacon de ses ancres voisines, construit un tableau contenant l'ID de l'ancre, l'emplacement de l'ancre et RSSI de chaque ancre, comme présenté dans la Tableau II.2. Alors, chaque ancre échange son tableau avec ses voisins [19].

TABLE I
NEIGHBORING ANCHORS OF ANCHOR A_1

Anchor ID	Anchor location	RSSI values for anchor
A_2	(x_2, y_2)	$RSSI_{A_2,A_1}$ $RSSI_{A_2,A_3}$
A_3	(x_3, y_3)	$RSSI_{A_3,A_1}$ $RSSI_{A_3,A_2}$

Tableau II. 1 : Les ancres voisines pour A_1 [19]

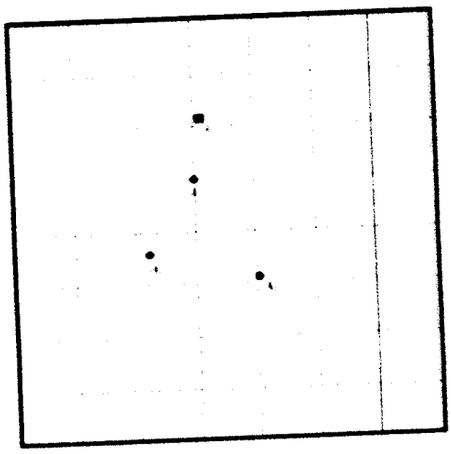
- Ensuite le nœud inconnu s reçoit les messages provenant des ancres A_1, A_2 et A_3. Ensuite il construit un tableau contenant l'ID de l'ancre, l'emplacement de l'ancre et la valeur RSSI de chaque ancre comme illustré dans le tableau II [19].

TABLE II
NEIGHBORING ANCHORS OF THE SENSOR s

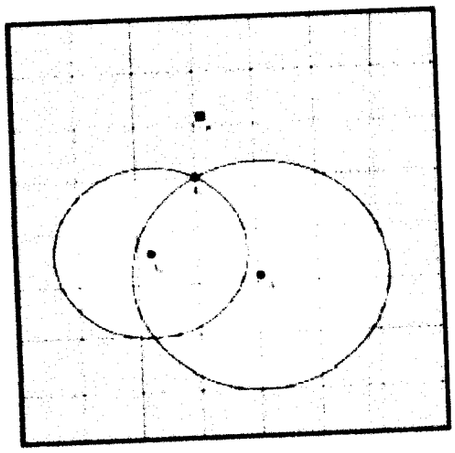
Anchor ID	Anchor location	RSSI values between anchors	RSSI value between anchor and sensor
A_1	(x_1, y_1)	$RSSI_{A_1,A_3}$ $RSSI_{A_1,A_2}$	$RSSI_{s,A_1}$
A_2	(x_2, y_2)	$RSSI_{A_1,A_2}$ $RSSI_{A_2,A_3}$	$RSSI_{s,A_2}$
A_3	(x_3, y_3)	$RSSI_{A_1,A_3}$ $RSSI_{A_2,A_3}$	$RSSI_{s,A_3}$

Tableau II. 2: Les ancres voisines pour le capteur s [19]

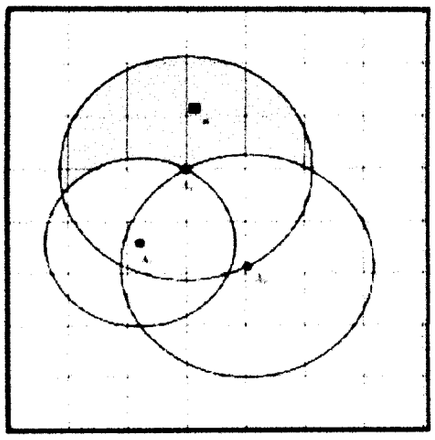
REAL utilise les données dans la Tableau II.3 pour construire la zone de résidence pour le capteur. Initialement, le capteur Choisit son ancre la plus proche qui a la plus grande valeur de puissance de signal des trois ancres (en comparant $RSSI_{s,A_1}$,$RSSI_{s,A_2}$et $RSSI_{s,A_3}$) . On suppose que cette ancre est A_1 Dans ce cas (voir figure II.1 (a)). Alors, capteur 's' vérifie s'il est localisé à l'intérieur $\overline{CMU}(A_2,A_3)$ en exécutant algorithme 1 (ligne 7). Si c'est le cas, donc le capteur 's' réside à l'intérieur de la zone verte, comme montre la figure II.1 (b). La zone de résidence initiale pour le capteur 's' (La zone verte) est réduite en exécutant l'algorithm1 (la ligne 9), comme illustre la figure II.1 (c).



(a)



(b)



(c)

Figure II. 3: Construction de zone de résidence [19]

- Le capteur S répète l'étape 3 pour des combinaisons diverses des trois ancres voisines en exécutant l'algorithme 1 (la ligne 6 à 10) pour déterminer toutes les zones de résidence possibles pour le capteur. Noter que cet exemple montre seulement une combinaison de trois ancres voisines.
- L'algorithme Grid Scan décrit précédemment est utilisé pour déterminer la zone de résidence finale de capteur, C'est la zone avec le maximum de chevauchement.
- Finalement, le capteur calcule son emplacement comme le centre de gravité de cette zone [19].

V.4. Performance de l'algorithme REAL

L'algorithme APIT est le plus proche à l'algorithme REAL proposé parmi tous les algorithmes à base de zone de présence des capteurs. La raison en est que les deux régimes partagent un ensemble commun de paramètres du système, à savoir ils ont le même nombre des combinaisons de trois ancres voisines et les mêmes valeurs de RSSI. Par conséquent, la performance de REAL system est comparable à celle de l'origine APIT et deux autres versions améliorées de cet original notée APIT – I [26] et APIT – II [26].

Les expérimentations présentées dans la publication REAL ont montré que REAL est plus performent en termes de calcul, de mémoire et d'énergie. En effet, les algorithmes APIT, APIT-I et APIT-II ont le même coût de communication parce qu'ils sont tous basés sur le test

d'APIT qui exige des communications entre des nœuds capteurs voisins pour maintenir le statut de voisinage, où chaque ancre et capteur émettent un message broadcast. Alors que REAL ne nécessite aucune interaction entre les capteurs voisins et seulement les ancres diffusent deux messages beacon. d'ailleurs, REAL n'est pas sensible à la densité de capteurs voisins qui a un impact significatif sur l'exactitude des tests de présence de capteurs, et par conséquent sur l'erreur d'estimation comme il est le cas de APIT, APIT - I et APIT – II.

De plus, REAL surpasse de loin les trois protocoles APIT, APIT - I et APIT – II. En effet, la somme des surfaces des trois zones de résidence des capteurs obtenues pour une combinaison de trois ancres voisines, est plus grande est importante. Cela conduit plusieurs nœuds à estimer leurs positions et par conséquent l'erreur de localisation diminue.

VI. Conclusion

Dans ce chapitre, nous avons défini la localisation conçus pour les RCSF et un certain nombre de techniques de mesure de position ainsi que les algorithmes utilisés range-based et range-free dans la localisation des capteurs. Ensuite nous avons présenté un nouveau protocole de localisation de type range-free, appelé REAL dont nous avons donné un exemple illustratif pour mieux le comprendre.

REAL est un algorithme très efficace et plus performant que les autres algorithmes. Etant donné le manque d'expérimentation dans la publication REAL [19], nous avons implémenté celui-ci dans le chapitre qui suit tout en présentant le système d'exploitation TinyOS et le langage NesC avec lequel nous avons travaillé.

Chapitre III

Evaluation et implémentation de l'algorithme REAL

I. Introduction

L'algorithme REAL [19] est l'un des algorithmes de calcul de position proposé comme solution de la localisation au niveau des réseaux de capteurs sans fil.

Nous avons développé une application réelle sur des capteurs pour illustrer l'apport de la localisation des nœuds selon l'algorithme REAL. Pour concrétiser cet apport, nous avons évalué un réseau de capteurs avec trois ancres. Dans ce qui suit, nous détaillons les éléments matériels et logiciels qui ont une relation avec notre application.

II. Environnement de travail

L'implémentation de l'algorithme nécessite l'utilisation des capteurs telosb comme outil matériel, NesC qui est un langage orienté composant comme outil logiciel, TinyOs qui est un système d'exploitation dédié aux équipements à ressources limitées et Java pour une meilleure présentation des résultats recueillis par les capteurs.

II.1. Outils matériels

La figure suivante représente les capteurs qu'on a utilisés pour l'implémentation de notre algorithme :

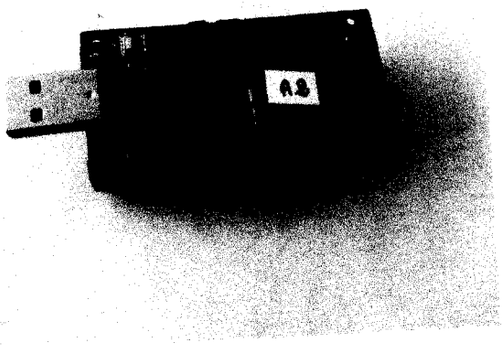


Figure III. 1 : Capteur Telosb

Les capteurs Telosb utilisés sont composés de :

- **Microcontrôleur Msp430 :** fabriqué par Texas Instrument, cadencé à 8MHz. Il dispose de 10 KOctets de RAM et de 48 KOctets de mémoire flash. Ce microcontrôleur est optimisé pour répondre aux contraintes d'économie d'énergie des

réseaux de capteurs, d'où son cadencement faible, et de ce fait, il a une capacité de calcul assez limitée [20].

- **Port USB** permettant de programmer (flasher) les capteurs, ou de faire remonter des informations et de les récupérer ensuite à l'aide d'un terminal à partir d'un ordinateur [20].
- **LED** outil visuel indispensable de vérification du fonctionnement du code intégré. Chaque capteur dispos de trois LED (rouge, vert et bleu), elles permettent de suivre le changement d'état de fonctionnement du capteur (réception, envoi, etc.) avec une programmation adéquate [20].

Ces capteurs offrent une faible consommation d'énergie, elle est compatible avec la distribution open-source de TinyOs. Alimenté par deux piles AA (1.5 V).

Nous avons eu besoin aussi d'un ordinateur, on a utilisé un PC DELL, processeur i3.

II.2. Outils logiciels

Dans cette section nous présentons les outils logiciels utilisés dans le développement de notre application. Pour réaliser notre application nous avons utilisé Ubuntu 9.10, le langage NesC dans le système d'exploitation TinyOS2.x, et le langage java sous l'environnement NetBeans IDE 6.7.1.

➢ Présentation de TinyOS2.x

TinyOS est un système d'exploitation open-source, intégré, modulaire, destiné aux réseaux de capteurs. Il respecte une architecture basée sur une association de composants, réduisant la taille du code nécessaire à sa mise en place afin de respecter les contraintes de mémoires qu'observent les réseaux de capteurs.

En effet, TinyOS est constitué de plusieurs modules disponibles pour les applications et offrant des fonctions de capture de mesures ou de communication. Il n'existe pas d'exécutable pour le noyau du système, il est construit au moment de la compilation de l'application en fonction des composants qu'elle utilise. Le langage de programmation associé, le nesC, qui est une extension du langage C, permet de déclarer les composants ainsi que les liens qui les unissent et de faire l'association code/composants.

TinyOS s'appuie sur un fonctionnement évènementiel, c'est à dire qu'il ne devient actif qu'à l'apparition de certains évènements, par exemple l'arrivée d'un message radio.

Le reste du temps, le capteur se trouve en état de veille, garantissant une durée de vie maximale connaissant les faibles ressources énergétiques des capteurs. Ce type de fonctionnement permet une meilleure adaptation à la nature aléatoire de la communication sans fil entre capteurs [21].

TinyOS utilise comme langage de programmation NesC.

➢ **Le langage NesC**

Le langage NeC (Network Embedded System C) est un langage utilisé par tinyOS, syntaxiquement proche du C qui a une architecture basé sur des composants. Cela permet de réduire la taille mémoire du système et de ses applications, chaque composant correspondant à un élément matériel du type LED ou timer ou carte de capture par exemple. Les composants peuvent être réutilisés dans n'importe quelle application. L'implémentation des composants, dans le but d'élaborer des applications, s'effectue en déclarant des tâches, des commandes ou des évènements (la fin d'un timer, réception d'un message,...).

Dans la pratique, NesC permet de déclarer deux types de composants : les modules et les configurations :

- Les modules : constituent les briques élémentaires de code et implémente une ou plusieurs interfaces; l'interface définie d'une manière abstraite les interactions entre deux composants.
- Les configurations : pour regrouper les fonctionnalités des modules.

L'extension du langage **NesC** est **.nc**, compilé vers C.

Dans l'implémentation de notre application nous avons adopté le langage de programmation NesC cat c'est un **langage orienté composant** pour programmer les dispositifs à ressources limitées tels que les capteurs,

➢ **IDE NetBeans (Integrated Development Environment):**

Nous avons utilisé NetBeans qui est un IDE (Integrated Development Environment) modulable et open-source, et qui offre de nombreuses fonctionnalités :

- Il contient un éditeur avec coloration syntaxique suivant le langage choisi.
- Il permet la navigation et la gestion des différents projets et de leurs classes.
- Il contient un debugger et un compilateur.

- Il renomme automatiquement les fonctions/classes/variables dans l'ensemble du code.
- Il permet la complétion lors de la saisie du code.
- Il est compatible avec Windows, Unix et Mac [22].

Notre choix s'est porté sur cette plateforme car elle est gratuite (open source) et simple à utiliser.

➢ **Le langage java**

JAVA a été développé par Sun au début des années 90 dans une filiation avec le langage C et surtout le langage objet C++ mais dans une optique de plus grande portabilité d'une machine à une autre et d'une plus grande fiabilité. Les programmes JAVA sont compilés en "bytecode", un langage intermédiaire indépendant de la plateforme [22].

Dans l'implémentation de notre application nous avons utilisé le langage de programmation java parce qu'il est **orienté objet** utilisé pour interpréter et bien exploiter ce qui se passe au centre de contrôle et pour communiquer les données envoyées par les capteurs. D'autre part, pour réaliser l'interface graphique grâce à sa librairie Swing.

III. L'architecture réseau

L'architecture de notre réseau est un ensemble de trois ancres, un Blind et une station de base. Les trois nœuds ancres émettent entre eux leur position (x,y) et par la suite chaque ancre envoie sa position et un tableau RSSI au Blind, suivant les valeurs reçus, le Blind calcule sa position estimée selon l'algorithme REAL et l'envoie à la station de base.

La figure suivante illustre l'architecture de notre application :

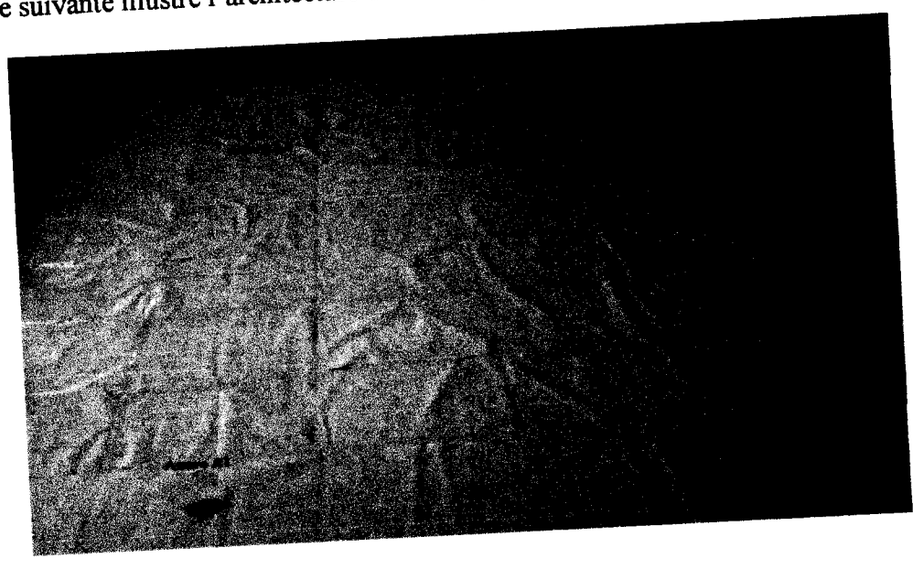


Figure III. 2 : Architecture du réseau

IV. Installation matériel

Pour l'installation matériel nous avons utilisé un capteur relié à l'ordinateur via un port USB qui joue le rôle d'une station de base, trois capteurs ancres et un autre qui représente le Blind comme l'a montré la figure précédente. Chacun des TelosB communiquent entre eux via une liaison sans fil et la station de base communique avec l'ordinateur via le port USB.

1. Les trois capteurs ancres sont flashés avec le programme «AncreC». La Figure III.3 illustre la documentation de ce programme.

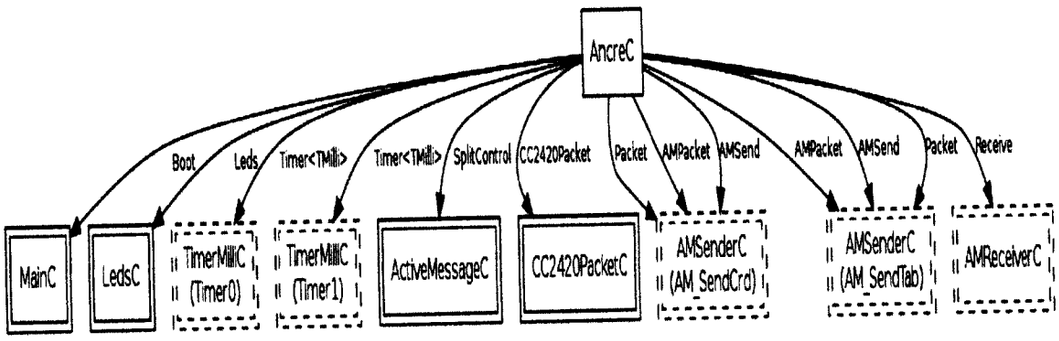


Figure III. 3: Représentation graphique du programme « AncreC »

Ce programme utilise les interfaces suivantes :

- **Boot :** permet d'initialiser tous les composants au démarrage, elle est fournie par la configuration MainC qui est le cœur de l'application

- **Leds :** utilisée pour la manipulation des leds, fournie par LedsC

- **Timer<TMilli> :** c'est une interface de synchronisation qui permet de gérer le timer d'émission, de round et d'allumage des leds

- **SplitControl :** utilisée pour l'émission radio fournie par la configuration

- **CC2420Packet :** c'est une interface pour calculer la puissance du signal entre deux nœuds

- **AMSend :** pour l'envoi du paquet

- **Packet :** pour accéder aux données du message

2. Le capteur Blind est flashé avec le programme « BlindC » la Figure III.4 illustre la documentation de ce programme.

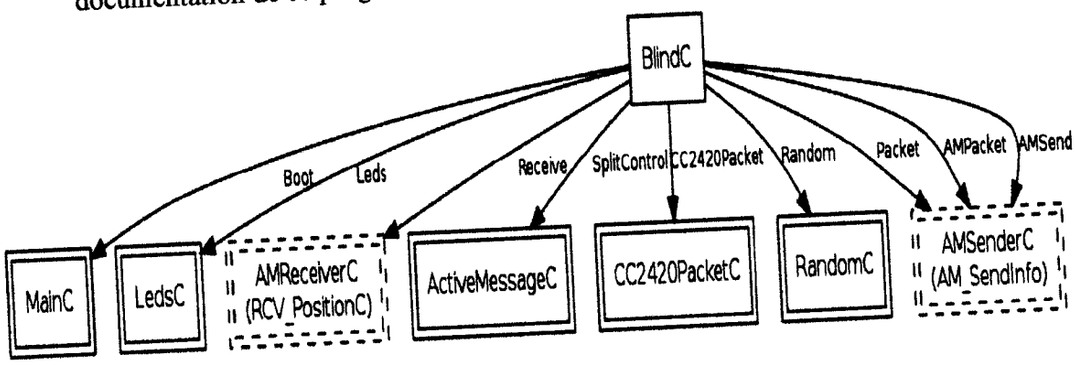


Figure III. 4 : Représentation graphique du programme « BlindC »

Ce programme utilise les mêmes interfaces que le programme « AncreC »

3. La station de base est flashé avec le programme « ReceiverC » qui a une communication radio avec le Blind et une communication serial avec le PC.

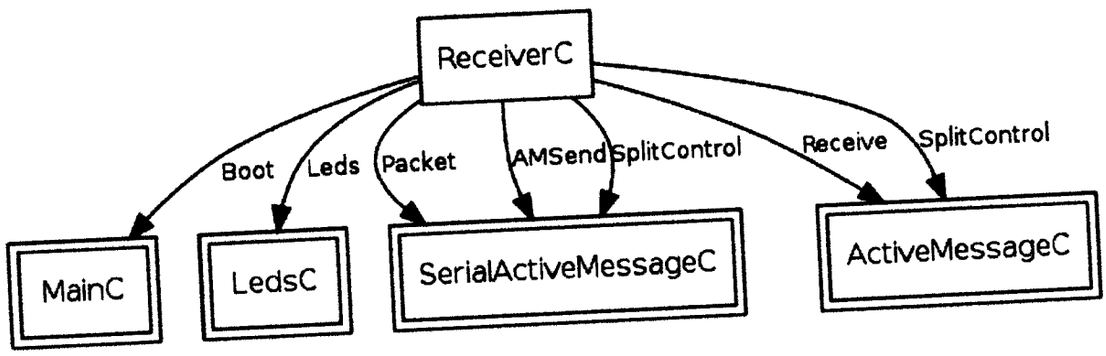


Figure III. 5 : Représentation graphique du programme « ReceiverC »

- **ActivemessageC :** permet l'accès à la liaison sans fil et l'encapsulation des messages qui pourront être ensuite envoyés via la liaison sans fil.

-**SerialActiveMessageC :** permet l'accès à la liaison filaire et l'encapsulation des messages qui pourront être ensuite envoyés via la liaison série.

V. Les éléments de l'application

Dans cette section, nous détaillons les principales fonctionnalités de notre application qui est composée de trois parties : une partie ancre (AncreC), une deuxième partie Blind (BlindC) et une troisième station de base (ReceiverC) reliée à un PC. La figure suivante représente tous les éléments de notre application :

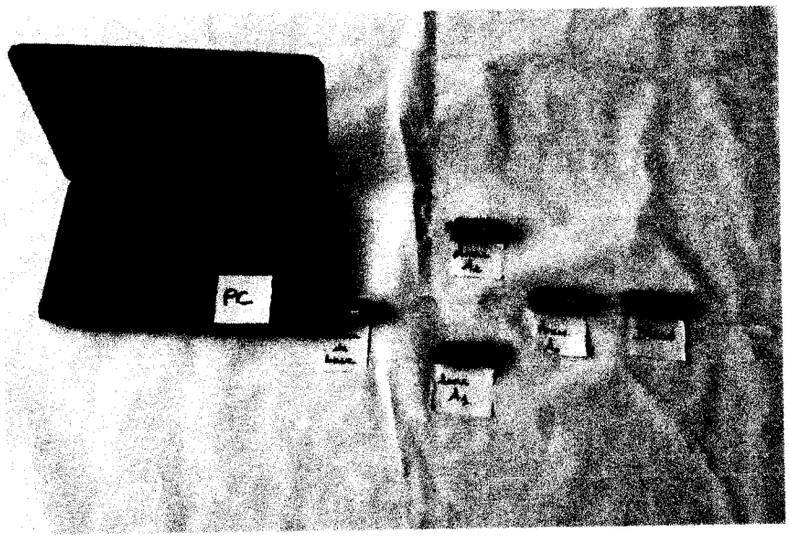


Figure III. 6 : Eléments matériel

V.1. Partie ancre (AncreC)

Chaque ancre commence par envoyer un broadcast contenant ses informations de localisation (sa position (x,y) qui est prédéfinie dans le code) et son identifiant, chaque ancre construit sa liste d'ancres voisines ; chaque liste comprend l'ID de l'ancre et la valeur RSSI (la puissance du signal entre lui-même et l'ancre voisine).

Après que chaque ancre collecte toutes les informations reçus, il envoie un message au Blind qui contient son ID, sa position et un tableau RSSI contenant les valeurs RSSI entre l'ancre et tous les voisins.

La commande pour calculer la valeur RSSI entre les ancres :

```
Call CC2420Packet.getRssi(msg)-45
```

V.2. Partie Blind (BlindC)

Après que le Blind collecte les informations de toutes les ancres, il calcule sa position estimée selon l'algorithme REAL ensuite il envoie sa position réelle et sa position estimée ainsi que les positions des ancres à la station de base.

V.3. Partie station de base

La station de base reçoit les informations envoyées par le Blind via la liaison radio ensuite elle envoie ses informations collectées au PC via la liaison série.

Dans cette étape nous exécutons une application développée en Java au niveau du centre de contrôle. Nous avons un programme Java qui reçoit et affiche les paquets reçus, mais pour que l'application puisse interpréter les informations reçues, il faut qu'il sache la structure du contenu dans le fichier "**Message.h**".

TinyOs fournit l'outil MIG (Message Interface Generator), qui est capable d'analyser un **"fichier.h"** et de convertir toutes les structures en classes Java disposant des méthodes nécessaires qui nous permettent d'accéder aux valeurs des attributs. Nous faisons appel à cet outil depuis le fichier Makefile, dans notre application coté ReceiverC. Son appel prend la syntaxe suivante :

```
mig java –target=null $ (CFLAGS) –java-classname=PositionMsg Message.h
Position_Msg-o $@
```

Maintenant pour afficher les valeurs reçues, nous lançons le programme Java en exécutant la commande suivante en mode console :

```
Java MsgReader
```

VI. Exemple d'exécution

Dans cette section, nous présentons les implémentations réalisées dans le cadre de notre projet. Nous étudions les performances de l'algorithme de localisation REAL en termes d'erreur de localisation. Nous présentons les résultats obtenus en illustrant la position des ancres et la position estimée et réelle par le Blind.

- **Erreur de localisation**

Un des facteurs d'évaluation le plus important dans la technologie de localisation est l'exactitude de localisation, qui se réfère au degré de précision de l'information calculée par un nœud capteur ordinaire obtenue par l'algorithme de localisation ou le système. Dans les WSN, l'erreur de localisation est généralement utilisée comme une description quantitative de l'exactitude de la localisation.

Cette erreur de localisation est calculée comme la moyenne de la différence absolue entre la position réelle et la position estimée par chaque nœud du réseau.

Supposons que le réseau est constitué de N nœuds, que la position réelle de chaque nœud est$\{(x,y)_a | a \in N\}$, et que la position estimée par chaque nœud est [26] :

$$LocError = \frac{1}{N}\sum_{\partial \in N}\sqrt{(x,\hat{x})^2 + (x,\hat{x})^2}$$

Dans notre expérience, nous utilisons 3 ancres, un Blind de position inconnue et une station de base. Les nœuds sont déployés dans une zone de 50 mètre sur 50 mètres.

La figure suivante représente notre interface java qui est composé d'un repère et un curseur qui détermine les coordonnées de chaque point en haut de l'axe Y.

Avant que la station de base reçoive les données envoyées par le Blind, le repère est vide.

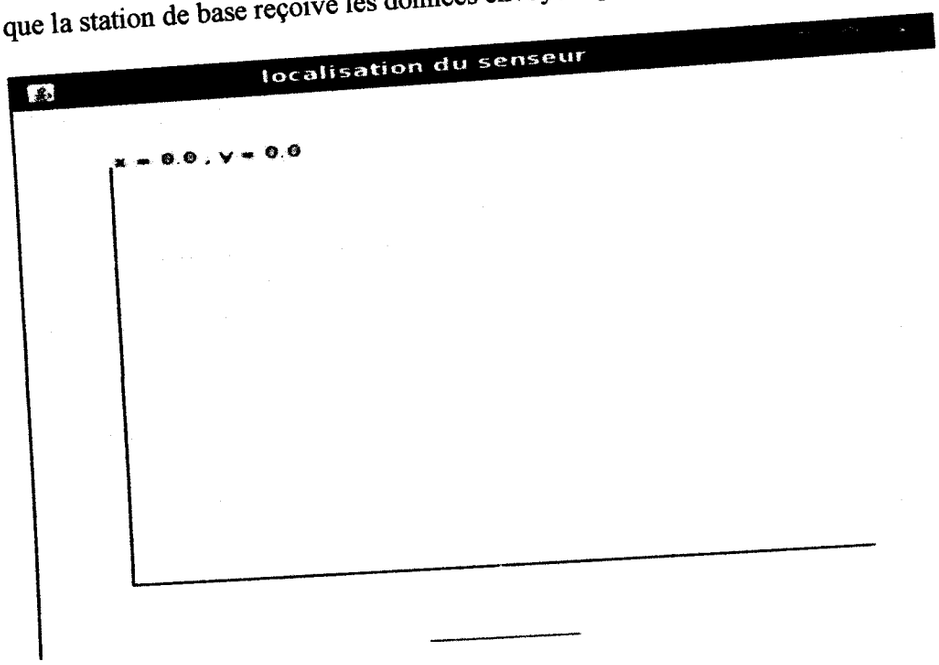


Figure III. 7 : Interface Java

Une fois que les ancres collectent les informations nécessaires, elles les transmettent au Blind, ce dernier calcule la position estimée et l'envoi à la station de base.

La figure suivante montre un exemple d'exécution en fixant le Blind au point (1,4) :

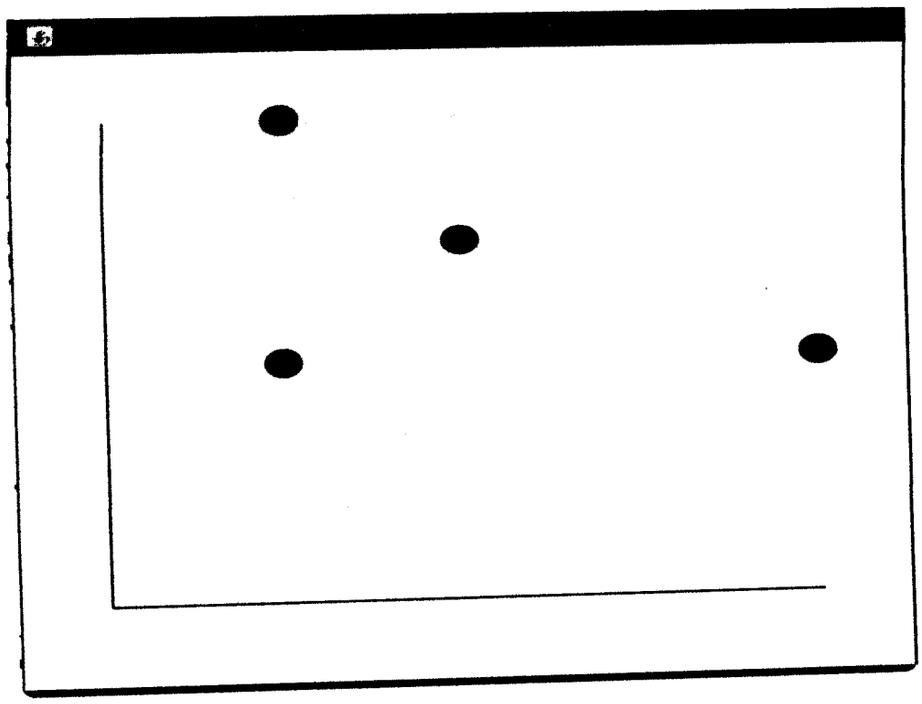


Figure III. 8 : Exemple d'exécution 1

Les cercles noirs, rouge et jaune, représentent respectivement les ancres dans un repère orthonormé de dimension 50métres sur 50métres, la position réelle du Blind et la position estimée, d'où leurs coordonnées sont fixées dans le code : A0(1,2), A1(4,2), A2(2,3) et le Blind (1,4).

La figure suivante montre le calcul de l'erreur de localisation, les valeurs RSSI ainsi que le neoud le plus proche au blind dans le terminal de TinyOs :

```
id de noeud: 0 avec coordonnes x=1 et y=2
   puisance de signale de paquet reÃ§u [-41]
id de noeud: 1 avec coordonnes x=4 et y=2
   puisance de signale de paquet reÃ§u [-45]
id de noeud: 2 avec coordonnes x=2 et y=3
   puisance de signale de paquet reÃ§u [-30]
l'ancre le plus proche est le noeud 2
notre senseur se trouve out a la point x=238 et y= 428 erreur de localisation: 1
41
```

Figure III. 9 : Erreur de localisation près de A2

Nous avons ensuite déplacé le Blind du côté de l'ancre A_1 avec les coordonnées (4,1) tout en gardant les mêmes positions des ancres. De ce fait, nous assistons à l'affichage d'une position résultante avec un cercle jaune comme le montre la figure suivante :

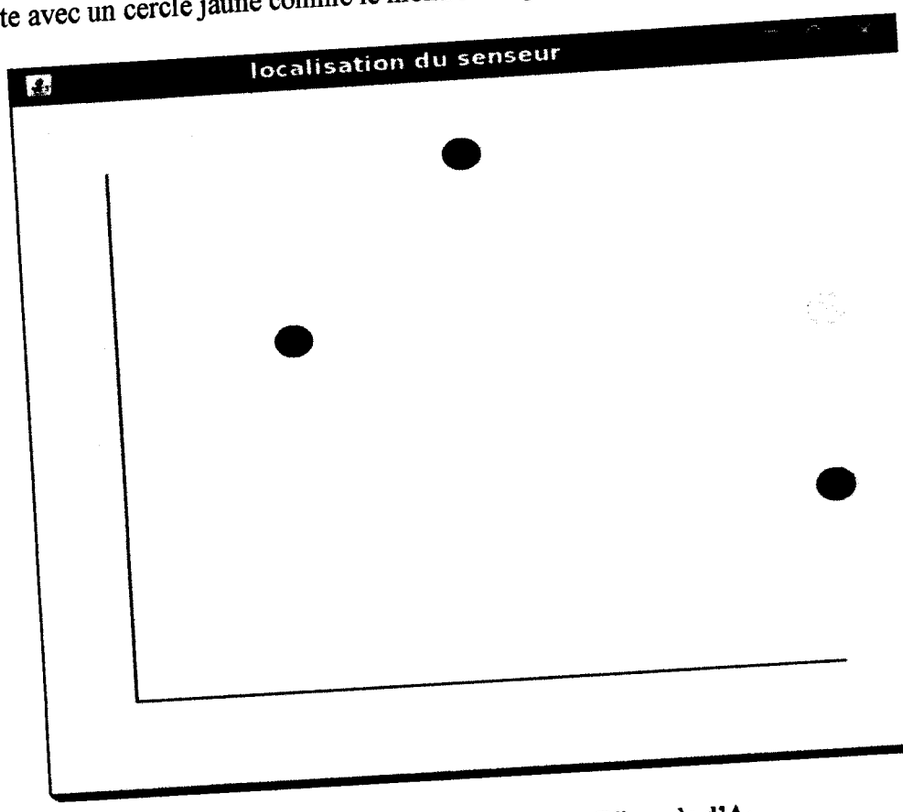


Figure III. 10 : Déploiement du Blin près d'A_1

Ici la position estimée se trouve sur l'ancre A_1. La figure qui suit illustre les identifiants des ancres, les valeurs RSSI et l'erreur de localisation :

```
id de noeud: 0 avec coordonnes x=1 et y=2
   puisance de signale de paquet reÃ§u [-53]
id de noeud: 1 avec coordonnes x=4 et y=2
   puisance de signale de paquet reÃ§u [-29]
id de noeud: 2 avec coordonnes x=2 et y=3
   puisance de signale de paquet reÃ§u [-43]
l'ancre le plus proche est le noeud 1
notre senseur se trouve out a la point x=455 et y= 244 erreur de localisation: 1
55
```

Figure III. 11 : L'erreur de localisation près de A_1

Remarque : les coordonnées du senseur ainsi que l'erreur de localisation sont des valeurs multipliées par 100 c'est-à-dire que les coordonnées réelles son x=4.55 (4.55*100 qui donne 455), y=244 (2.44*100=244) et 1.55 l'erreur d'estimation.

- **Le diagramme d'erreur**

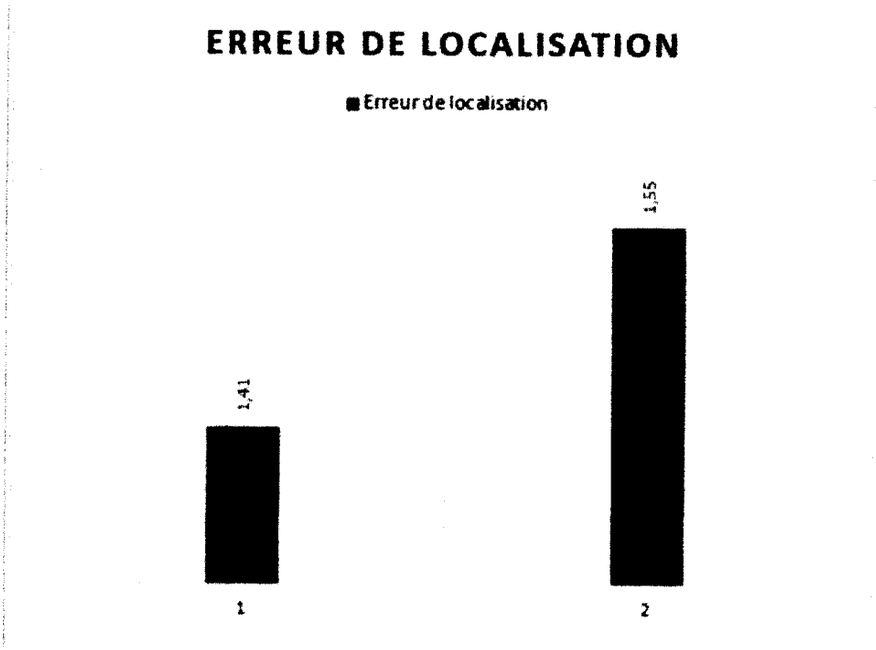


Figure III. 12 : Diagramme d'erreur de localisation

Observation : L'histogramme 1et 2 représente respectivement l'erreur de localisation selon le Blind à la position (1,4) et au point (4,1).

On remarque dans le diagramme de la Figure III.12, que l'erreur de localisation change lorsqu'on déplace le Blind.

VII.Conclusion

Dans ce chapitre, nous avons présenté le système d'exploitation TinyOS, langage NesC avec lequel nous avons travaillé ainsi que l' IDE NetBeans et langage java . Ensuite, nous avons implémenté l'algorithme REAL [19] en donnant quelques exemples d'exécutions et enfin nous avons conclu qu'en déplaçant le blind l'erreur de localisation change.

Bibliographie

[1] M. David, ''Sécurité dans les réseaux de capteurs sans fil Sténographie et réseaux de confiance'', Thèse de doctorat, Université de Franche-Comté, 2010.

[2] H. Sedjelmaci, ''Mise en œuvre de mécanismes de sécurité basés sur les IDS pour les réseaux de capteurs sans fil'', Thèse de doctorat, Université de Tlemcen, 2012.

[4] N. Hadj Taleb, ''Conception et développement d'une application de surveillance de l'environnement à base d'un réseau de capteurs sans fil'' Diplôme ingéniorat, L'Ecole Nationale d'Ingénieurs de Sfax, 2011.

[5] K. BEYDOUN, ''Conception d'un protocole de routage hiérarchique pour les réseaux de capteurs'', Thèse de doctorat, UNIVERSITE DE FRANCHE-COMTE, 2009.

[7] A. Makhoul, ''Réseaux de capteurs : localisation, couverture et fusion de données'', Thèse de doctorat, UNIVERSITE DE FRANCHE-COMTE, 2008.

[9] M. Achir. ''Technologies basse consommation pour les réseaux Ad Hoc'',Thèse pour obtenir le grade de Docteur de l'INPG. Institut National Polytechnique de Grenoble. juillet 2005.

[10] A. Ouadjaout, '' La Sécurité et la Fiabilité du Routage dans les Réseaux de Capteurs Sans Fils'', Mémoire de magistère, USTHB, Algérie, 2006.

[13] O. Arondel, T. Ponpardin,''Systeme de positionnement Galileo/Glonass'', thèse doctorat, Ecole supérieur d'ingénieurs2009.

[16] N. Labraoui,''La sécurité dans les réseaux sans fil Ah-doc'', Thèse de doctorat, Université Tlemcen, 2012.

[17] D. S. Yannick, ''Localisation des nœuds dans les Réseaux de capteurs sans fil'', Mméoire de Master 2 RTM, Université d'Avignon, 2007/2008.

[18] R. Dalce, ''Méthodes de localisation par le signal de communication dans les réseaux de capteurs sans fil en intérieur'', Thèse de doctorat, Université de Toulouse, 2013.

[19] A. Mesmoudi, M. Feham, N. Labraoui, C. Bekara, ''An Efficient Area-Based Localization Algorithm for Wireless Sensor Networks'', Article scientifique, International Review on Computers and Software,2015.