UNIVERSITE SAAD DAHLAB DE BLIDA

Faculté des sciences
Département d'Informatique

MEMOIRE DE MAGISTER

Spécialité : Systèmes d'Informations et de Connaissance

POSITIONNEMENT COLLABORATIF DES NŒUDS D'UN WSN

Par

SILMI Souhila

Devant le jury composé de

H. ABED	Professeur, U. Blida	Présidente
S. OUKID	Maître de Conférences A, U. Blida	Examinatrice
D. BENNOUAR	Maître de Conférences A, U. Blida	Examinateur
S. MOUSSAOUI	Maître de Conférences A, U.S.T.H.B., Alger	Rapporteur
Y. ZAFFOUNE	Maître Conférences A, U.S.T.H.B., Alger	Invité

Juin 2012

RESUME

Le développement des technologies sans fil et des domaines des microsystèmes électromécaniques a déclenché le succès des réseaux de capteurs sans fils (WSNs). Ces capteurs sont déployés massivement dans une région dite région d'intérêt afin de collecter et de transmettre les données. Plusieurs applications comme l'observation des animaux, le suivi médical et la sécurité des frontières ont été proposés.

Si les perspectives d'utilisation des WSNs, sont claires et attrayantes, les problématiques qu'engendrent ces réseaux n'en sont pas moins nombreuses. La possibilité de déterminer la localisation physique d'un capteur est une fonctionnalité fondamentale pour un nombre important d'applications.

Dans ce mémoire, nous nous sommes intéressés à la problématique de la localisation à travers un état de l'art assez complet sur les techniques et les méthodes proposées dans la littérature. Un intérêt particulier a été accordé aux techniques collaboratives et distribuées qui permettent le positionnement d'un capteur.

L'étude comparative et la synthèse des travaux existants a permis l'élaboration d'un nouveau protocole basé sur une méthode distribuée collaborative, qui s'appuie sur une technique évolutive. L'évaluation du protocole a été réalisée et montre l'intérêt de la proposition dans ce domaine.

Mots clés : réseaux de capteurs sans fil, localisation, AT-Free, Ht-refine.

Abstract

The development of wireless technologies and the electromechanical micro systems initiate the success of the wireless sensors networks. These sensors are deployed massively in an area known as area of interest in order to collect and to disseminate the data. Several applications like the observation of animals, the medical monitoring and the safety of the borders were proposed.

The WSNs are very interesting and attractive but they generate a number of new problems. The possibility of determining the physical localization of a sensor is a fundamental function for a number of applications.

In this memory, we were interested by the problems of the localization through a complete state of the art on the techniques and the methods suggested in the literature. A private interest was granted to the collaborative and distributed techniques which allow the positioning of a sensor.

The comparative study and the synthesis of existing works allowed the development of a new protocol based on a collaborative distributed method, which is based on an evolutionary technique. The evaluation of the protocol was carried out and shows the interest of the proposal.

Keywords: wireless sensors network, Localization, AT-Free, Ht-refine.

REMERCIEMENTS

Avant tout, je tiens à remercier DIEU le tout puissant de m'avoir donné le courage et la force pour bien mener ce modeste travail. Merci au dieu, et après à ma mère, la personne la plus cherre sur la terre. Je te dois ce que je suis aujourd'hui grâce à ton amour, à ta patience et tes innombrables sacrifices. Que ce travail soit le témoignage sincère et affectueux de ma profonde reconnaissance pour tout ce que tu as fait pour moi. Que dieu, le tout puissant, te préserve et te procure santé et longue vie afin que je puisse à mon tour te combler.

Je dis merci à toute ma famille, mon grand frère NORDINE et mon frère ABD ELHALIM, mes chers sœurs SALIMA, FELLA, NORA, MALIKA, HABIBA et ma cherre HANANE. Je tiens à exprimer toute ma reconnaissance à mon beau frère, Dr DAHMOUNI MEFTAH, pour ses orientations et suivie.

Je tiens de remercier aussi MM MOUSSAOUI SAMIRA pour son encadrement de très grande qualité, ses nombreux conseils et le soutien constant tout au long de mon travail. Je remercie les examinateurs de ce mémoire Mm. OUKID, Mr. BENNOUAR et Mr. ZAFFOUNE pour le grand intérêt qu'ils ont accordé à mon travail, ainsi que pour leurs corrections. Merci également à MM H. ABED pour avoir présidé mon jury.

Je tiens à montrer toute ma gratitude à tous les collègues du laboratoire LSI à l'USTHB. Mes remerciements vont également à tous mes collègues en école doctorale, pour leurs collaborations le long de nos années d'études en magistère ensemble, soit au niveau de l'université de BLIDA, et même mes collègues de première année d'école doctorale, à l'école supérieur d'informatique ESI.

Enfin, je remercie toutes personnes ayant contribué de près ou de loin à la concrétisation de ce mémoire soyez –en tous remerciés du fond du cœur.

TABLE DES MATIERES

LISTE DES ILLUSTRATIONS, GRAPHIQUES ET TABLEAUX

INTRODUCTION

Les récentes avancées en micro-électronique et en communication numérique ont favorisé l'émergence des réseaux de capteurs sans fil. L'apparition de ce type de réseaux a provoqué un intérêt croissant depuis plusieurs années.

Concrètement, un réseau de capteurs sans fil est constitué de plusieurs entités de taille réduite, énergétiquement autonomes et dotés de capacités d'acquisition de données et de traitement. La tâche première de ces entités est de détecter un événement (changement de température, présence d'une substance chimique...). Elles sont donc capables de récolter des données relatives à leur environnement, de les traiter puis, si nécessaire, de les communiquer via un médium sans fil. Le déploiement de ce type d'appareils forme alors un réseau qui peut être utilisé dans différents domaines : militaire (surveillance d'une zone ennemie), civil (détection de feux de forêt), médical (suivi des patients).

Dû à leur faible coût et à leur petite taille, chaque appareil dispose d'une puissance de calcul, d'une capacité mémoire et d'une réserve énergétique limitées. De plus, leur capacité de détection et de communication est bornée par une portée maximale. Les applications pour lesquelles les réseaux de capteurs sans fil sont destinés doivent prendre en considération chacune de ces contraintes.

L'un des problèmes majeurs dans les réseaux de capteurs sans fil est celui de la localisation. En l'absence d'informations sur la position des nœuds, les données récoltées peuvent s'avérer n'être d'aucune utilité : il serait donc nécessaire de connaître l'emplacement de chaque capteur. Une première solution consiste à équiper tous les nœuds d'un système de positionnement global tel que le **GPS**. Cependant, cette méthode s'avère être trop couteuse sur le plan énergétique : les **GPS** sont des modules qui consomment beaucoup. La solution serait d'utiliser quelques points de références ayant des positions connues. On fera appel à des algorithmes qui, à partir des distances qui séparent ces références de nœuds ne connaissant pas leurs coordonnées, ou des angles qu'ils forment avec eux, peuvent localiser la totalité du réseau.

Dans ce mémoire, nous proposons d'étudier la problématique de la localisation dans les réseaux de capteurs sans fil (techniques, méthodes, et classification). Un certain nombre de travaux ont été proposés et d'autres ont été initiés. Ces travaux diffèrent par l'approche adoptée. Un intérêt particulier sera accordé aux techniques collaboratives et distribuées qui permettent le positionnement d'un capteur. Cette étude devra débuter avec un état de l'art sur ces méthodes et techniques existantes. Une synthèse et une comparaison sera réalisée avec la proposition d'un nouvel algorithme d'estimation des distances entre les nœuds.

Ce mémoire est organisé comme suit : dans le premier chapitre nous allons définir les réseaux de capteurs sans fil ainsi que leur architecture, leurs domaines d'application et leurs caractéristiques. Dans le deuxième chapitre nous allons introduire le problème de la localisation, les contraintes qui influencent la localisation et les techniques les plus utilisées par

les algorithmes de localisation dans les réseaux de capteur sans fil. Dans le chapitre 3 nous présenterons une synthèse sur les méthodes, les techniques, et la classification des algorithmes de localisations centralisés et des algorithmes distribuées. Dans le chapitre 4 nous continuant notre synthèse, par le fait de donner des comparaisons entre les différents méthodes et techniques de localisation existantes et présentés dans ce travail. Et c'est à partir de chapitre 5, qu'on abordant la conception d'un nouvel algorithme de localisation. Pour enfin arriver au chapitre 6, dans lequel, nous montrerons l'implémentation de ce nouvel algorithme et nous exposerons les différentes métriques qui servent à comparer ou à évaluer les algorithmes de localisation. Pour arriver à utiliser certaines des métriques présentées dans ce chapitre lors de l'évaluation de notre nouvel algorithme. Nous terminerons ce mémoire par une conclusion récapitulative des résultats et des points abordés à travers ce mémoire, avec des perspectives à ce travail.

CHAPITRE 1
LES RESEAUX DE CAPTEURS

1.1. Introduction

La miniaturisation croissante de l'électronique et les progrès réalisés dans le domaine de la télécommunication, qui n'ont cessé depuis l'émergence de l'informatique, permettent aujourd'hui de produire des composants à faible coût appelés 'capteurs', qui communiquent et consomment peu d'énergie. Ces petites entités électroniques, forment un réseau de capteurs et qui ont comme objectif, la récolte des grandeurs physiques de leur environnement proche (luminosité, mouvement, température, pression barométrique etc.).

Le thème des réseaux de capteurs connaît un intérêt croissant depuis plusieurs années. Les problématiques de localisation, en particulier, sont importantes dans la mesure où ces capteurs sont dans un environnement évolutif, et où ils peuvent être eux-mêmes amenés à modifier leur position, de manière autonome ou non.

Pour mieux cerner les enjeux du sujet, nous présenterons ce que sont les réseaux de capteurs, leur architecture, leurs usage, ainsi que la manière dont ils sont constitués. Ce chapitre commence par introduire les réseaux sans fil et donne une définition plus détaillée des réseaux de capteurs sans fil ainsi que leurs caractéristiques et leurs différents domaines d'application.

1.2. Les réseaux sans fil

2.1 Définitions

Durant ces dernières années, diverses technologies sans fil ont été proposées dans le but de substituer les transmissions filaires par des ondes radioélectriques. Ces technologies sont adaptées, de nos jours, à des contextes d'utilisation spécifiques et ont, notamment, donné naissance à deux types de réseaux : les réseaux personnels sans fil ou **WPAN** (**W**ireless **P**ersonal **A**rea **N**etwork) et les réseaux locaux sans fil ou **WLAN** (**W**ireless **L**ocal **A**rea **N**etwork) [KHE 04].

Un réseau **PAN** désigne un réseau restreint d'équipements informatiques habituellement destiné à une utilisation personnelle. La communication et l'échange d'informations se font entre ordinateurs, assistants personnels (**P**ersonal **D**igital **A**ssistant où **PDA**), imprimantes, téléphones mobiles et autres dispositifs dans un rayon limité (ne

dépassant généralement pas les quelques mètres). La norme **Bluetooth**, en raison de sa faible portée de transmission et des modèles de communication qu'elle définit, offre le meilleur exemple de technologies adaptées au **PAN**. En effet, un réseau **Bluetooth** basique (ou **piconet**) prend en charge au maximum huit terminaux, un de ces terminaux étant désigné comme maître et les autres comme esclaves. D'autre technologies sans fil tel que l'infra rouge et le **ZigBee** font aussi partie des technologies **PAN** les plus répondues [KHE 04].

Un réseau local sans fil (**WLAN**) est un système qui peut couvrir le périmètre d'un espace publique, d'une entreprise, d'un hôpital ou bien celui d'un campus universitaire. L'usage des technologies **IEEE** 802.11 (**IEEE : I**nstitue of **E**lectrical and **E**lectronics **E**ngineers) et **HiperLAN** (**H**igh **PER**formance radio **LAN**) est pertinent pour ce type de réseau puisque leur portée d'émission est de l'ordre de plusieurs centaines de mètres. Les réseaux locaux sans fil sont subdivisés en deux catégories : les réseaux avec infrastructures et les réseaux sans infrastructures.

Dans les réseaux à base d'infrastructures, seuls les terminaux peuvent se déplacer ; le cœur du réseau restant fixe. Ces réseaux sont composés de stations de bases (**SB**), aussi appelées stations de support mobile (Mobile support station), qui sont des sites fixes, rattachés parfois à un réseau filaire classique et de terminaux (unités mobiles). Une **SB** est munie d'une interface de communication sans fil pour communiquer avec les terminaux situés à l'intérieur de la zone couverte par cette **SB** (appelé cellule) [KHE 04].

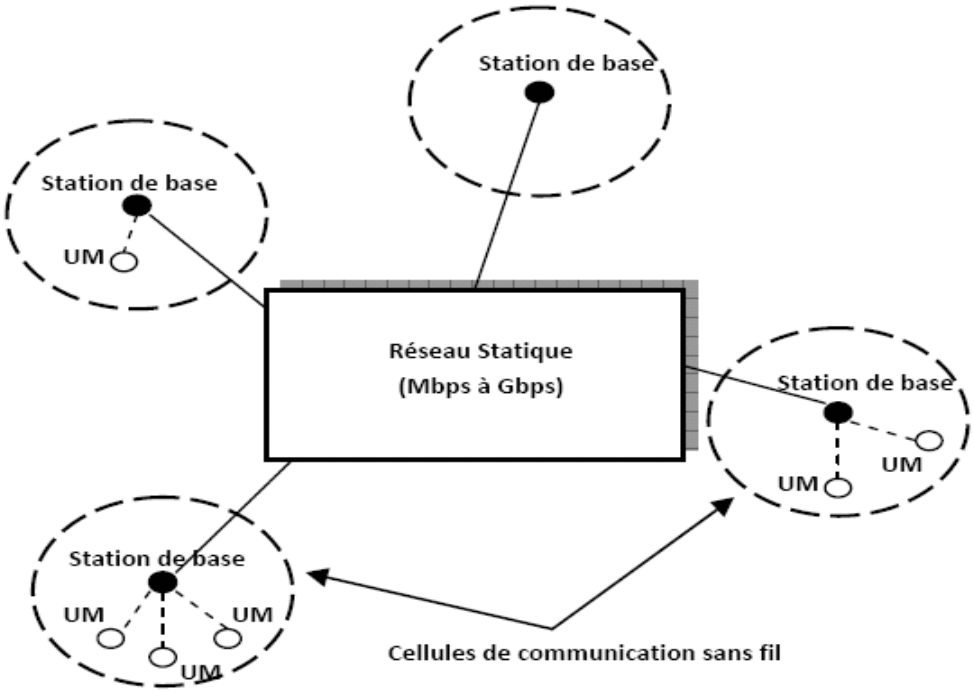


Figure 1. 1 : Réseau mobile avec infrastructure. [KHE 04]

La deuxième catégorie des réseaux locaux sans fil, est celle des réseaux sans infrastructures, dits réseaux ad hoc. Ce sont des réseaux qui ont l'avantage d'être rapide à déployer et étendent la notion de la mobilité à tous les composants de l'environnement mobile. Selon la Figure 1.2, contrairement aux réseaux à base d'infrastructures, aucune administration centralisée n'est nécessaire. Les terminaux sans fil (les unités mobiles -**UM**) communiquent entre eux d'une façon directe. Les réseaux ad hoc peuvent exister de façon autonome. Ils peuvent aussi être connectés à d'autres types de réseaux sans fil ou filaires, à base d'infrastructure ou non et forment, ainsi un réseau hybride.

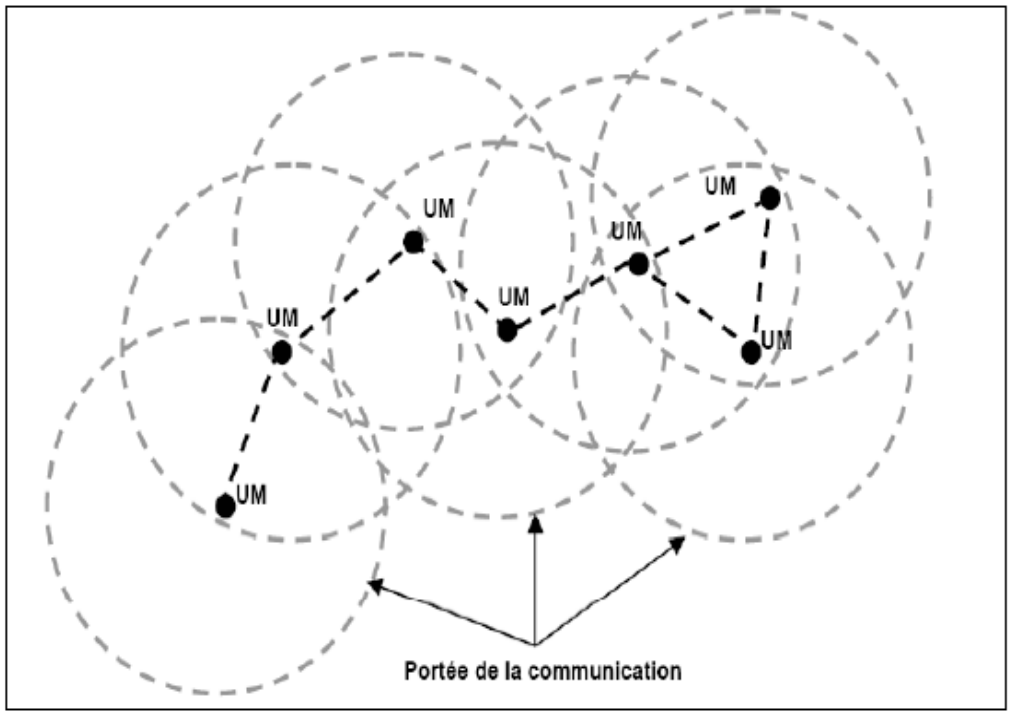


Figure 1. 2 : Réseau mobile sans infrastructure. [BEN 09]

Les réseaux de capteurs sans fil (**WSN**: **W**ireless **S**ensor **N**etwork) sont considérés comme un type spécial de réseaux ad hoc et ils sont souvent considérés comme étant les successeurs des réseaux ad hoc. En effet, les **WSN** partagent avec les **MANET** (**M**obile **A**d hoc **NET**works) plusieurs propriétés en commun, telles que l'absence d'infrastructure et les communications sans fil. Où les nœuds de ce type de réseaux consistent en un grand nombre de capteurs capables de recevoir et de transmettre des données environnementales d'une manière autonome. La position de ces nœuds est indéterminée. Car, ils sont dispersés aléatoirement à travers une zone géographique, appelée champ de captage, qui définit le terrain d'intérêt pour le phénomène capté. Mais l'une des différences clés entre les deux architectures est le domaine d'application. Contrairement aux réseaux **MANET**, qui n'ont

pas pu connaître un vrai succès, les **WSN** ont su attirer un nombre croissant d'industriels, vu leur réalisme et leur apport concret. Ces capteurs disposent d'une alimentation autonome, donc, leur durée de vie est limitée par la durée de vie de leur batterie. Cette forte contrainte a une influence majeure sur l'ensemble des techniques mises en place pour le déploiement de tels réseaux [SIL 11].

2.2 Les avantages des réseaux Ad Hoc

Á l'heure actuelle, les réseaux ad hoc sont exploités dans divers contextes et dans des environnements variés. Ces réseaux ont l'avantage d'être facile et moins coûteux à déployer. Ils représentent une alternative intéressante dans de nombreuses situations telles que les situations d'urgence (désastre naturel, tremblement de terre) impliquant la destruction totale ou partielle des infrastructures du réseau fixe. Ils sont aussi particulièrement utiles dans le cas des applications militaires ou de sauvetage pour lesquelles il n'est pas envisageable d'installer des infrastructures. D'autre part, leur utilisation est préconisée dans des contextes quotidiens lorsque les besoins ne justifient pas le déploiement d'infrastructures onéreuses.

Dans ces différents contextes d'application, un utilisateur a besoin d'accéder aux ressources numériques présentées dans le réseau ad hoc. Ces ressources peuvent, par exemple, correspondre à des documents mis à la disposition d'un groupe, lors d'une réunion de travail. Elles font, aussi, référence à des mesures prélevées par des capteurs réalisant des prélèvements pour la domotique ou la surveillance des désastres naturels [ZAF 08].

2.3 Les contraintes des réseaux Ad Hoc

Les technologies de communication qui utilisent les ondes radioélectriques sont connues pour générer beaucoup d'erreurs et pour avoir une faible bande passante. Ceci est dû à :

- Une plage de fréquence étroite, allouée par les autorités de régulation des télécommunications pour les réseaux radioélectriques.
- Un canal de transmission susceptible d'être victime d'interférences pouvant affecter le signal.

- Un vecteur non trivial d'atténuation et de distorsion du signal (selon la théorie du signal, l'atténuation est inversement proportionnelle à la distance qui sépare l'émetteur du récepteur).
- Une impossibilité de contrôler la propagation des ondes dans une direction particulière.

En outre, de nombreux défis doivent être relevés pour que les réseaux ad hoc puissent être, effectivement, exploités par les utilisateurs, pour le partage et pour l'accès aux ressources numériques offertes par les terminaux composant le réseau. Un premier défi est lié aux caractéristiques intrinsèques des réseaux sans fil : une communication dans un réseau ad hoc est immédiatement interrompue dès que les terminaux impliqués se trouvent hors de leur portée radio respective, ce qui peut être fréquent dans le cas de terminaux fortement mobiles. Cette dynamique du réseau soulève le problème de l'intégration dans différents réseaux. En particulier, un utilisateur nomade, du fait de sa mobilité, est souvent amené à rencontrer de nouveaux réseaux. Mais, il ne peut pas à priori accéder à leurs ressources numériques puisqu'il ne les connaît pas.

De manière générale, l'exploitation effective des réseaux ad hoc est rendue difficile par les spécificités de ces réseaux, qui sont :

- Les changements de topologie du réseau causés par des déconnexions volontaires ou involontaires des utilisateurs. Ainsi que, par des obstacles et des interférences dans les liaisons radio qui perturbent les communications.
- L'utilisation des batteries en tant que source d'énergie par les terminaux qui affecte la durée de vie du réseau.

Par ailleurs, les liens de communication dans les réseaux ad hoc sont asymétriques. Cela signifie que si un terminal reçoit un message d'un autre terminal, la réciproque n'est pas forcément vraie. Ensuite, l'accès au canal n'est pas contrôlé par une entité assurant la coordination des terminaux ou la synchronisation des communications. Par conséquent, tous les nœuds du réseau peuvent être victimes des interférences causées par l'émission de messages par les terminaux sans fil à portée de communication. Le taux d'erreurs est donc accru tandis que le débit de transmission est plus restreint.

Toutefois, malgré ces handicaps spécifiques, les réseaux ad hoc présentent des avantages indéniables, comme par exemple leur déploiement immédiat et leur faible coût d'un point de vue financier [ZAF 08].

1.3. Les réseaux de capteurs sans fil

3.1 Définition

Un réseau de capteurs sans fil, ou "**W**ireless **S**ensor **N**etwork" (**WSN**), est composé d'un ensemble d'unités de traitements embarquées, appelées communément "motes", communiquant via des liens sans fil. Chaque nœud mesure d'une manière continue des quantités physiques dans son environnement immédiat telles que la température et l'humidité. Les données récoltées traversent, par la suite, le réseau jusqu'elles arrivent à sa destination.

3.2 L'Architecture d'un capteur

L'un des problèmes auquel les développeurs des **WSN** font face est le manque d'énergie au sein de chaque nœud. Une mote fait usage de la technologie asynchrone dans son architecture, ce qui a comme conséquence une réduction de sa consommation.

On peut voir sur la Figure 1.3 les différents composants qui constituent un capteur. Pour être plus précis, chaque groupe de composants possède son propre rôle, à savoir, [HEN 08]:

Mote, **processeur**, **RAM** et **Flash** : On appelle généralement **Mote**, la carte physique utilisant le système d'exploitation pour fonctionner. Le cœur est composé par le bloc du processeur et des mémoires **RAM** et **Flash**. Cet ensemble est à la base du calcul binaire et du stockage temporaire pour les données et stockage définitif du système d'exploitation.

Radio et antenne : Un capteur est conçu pour mettre en place des réseaux sans fils. Les équipements étudiés sont généralement équipés d'une radio ainsi que d'une antenne afin de se connecter à la couche physique que constitue les émissions hertziennes.

LED, interface, capteur : Prévus pour mettre en place des réseaux de capteurs. On distingue des équipements bardés de différents types de détecteurs et autres entrées.

Batterie : Comme tout dispositif embarqué, ils disposent d'une alimentation autonome telle qu'une batterie, et parfois d'un panneau solaire pour permettre de recharger cette batterie, ce qui lui permet d'être disposé dans un endroit parfois inaccessible.

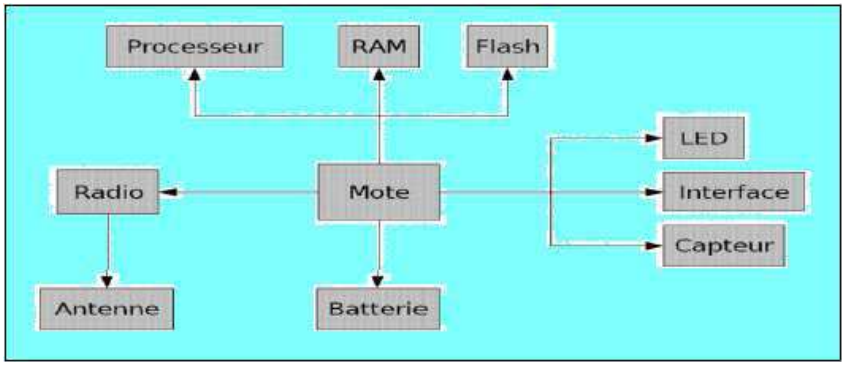


Figure 1. 3 : Architecture générale d'un capteur. [HEN 08]

On constate des spécificités liées à chaque type de batterie suivant le fabricant. Chacun d'eux développe son type de capteurs, ces types peuvent être **mica**, **mica2**, par exemple. Avec des contraintes hardware aussi strictes dues à la miniaturisation des capteurs, la partie software doit être la plus adaptée possible : d' où un lien très fort entre ces deux parties (hardware, software).

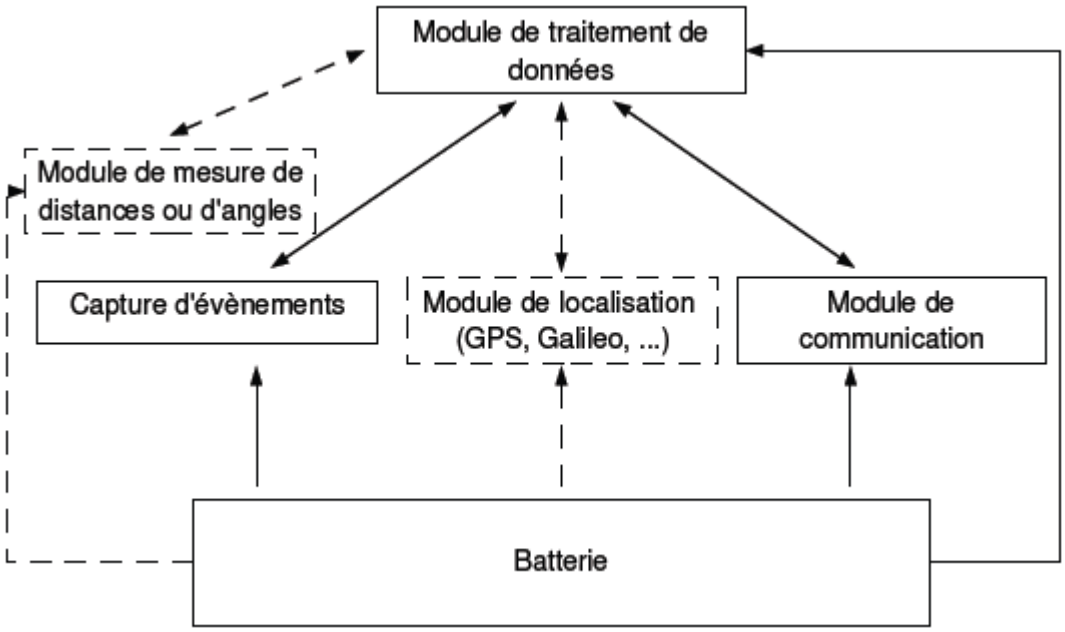


Figure 1. 4 : Architecture fonctionnelle d'un capteur.

En matière de fonctionnement, malgré la diversité des différents types de capteurs, leur fonctionnement reste le même. L'architecture fonctionnelle d'un capteur, représentée sur la Figure 1.4, se résume ainsi [SAA 08]:

- un module de capture : il détecte les événements ayant lieu dans son rayon de perception ;

- un module de traitement : doté d'un processeur et d'une mémoire, il traite les données détectées et éventuellement les communique ;

- un module de communication : il se charge de la transmission et de la réception des données via un médium sans fil ;

- un module de localisation (optionnel) : il peut s'agir d'un système **GPS**, **Galileo** (**CNES** et **ESA**) ou d'un autre système de localisation qui donne au capteur sa position exacte ou estimée;

- un module de mesure (optionnel) : il mesure la distance ou l'angle avec un capteur voisin;

Les interactions entre ces modules sont illustrées sur la Figure 1.4. Chaque module est alimenté par la batterie. La consommation d'énergie est essentiellement due aux modules de communication sans fil et de traitement des données. Le module optionnel de localisation est également une source de consommation d'énergie non négligeable. Pour prolonger la durée de vie des réseaux de capteurs, il est donc indispensable de minimiser les calculs et les communications. Ces contraintes majeures doivent être prises en compte par les méthodes proposées pour ce type de réseau.

Figure 1.5 : Exemple d'un capteur.

La Figure 1.5 (photo) représente un exemple de capteur alimenté par deux piles. Le déploiement de ce type d'appareils forme alors un réseau qui peut être utilisé dans différents domaines, et dans plusieurs exemples, les capteurs sont mobiles. On distingue deux types de réseaux [SAA 08]:

- Les réseaux statiques ;
- Les réseaux mobiles.

3.3 Le Modèle en couche dans les réseaux de capteurs sans fil

Dans les réseaux de capteurs, les nœuds sont déployés dans un environnement sans infrastructure, en n'ayant aucune information sur la topologie globale, même locale du

réseau construit. Pour cela, les nœuds capteurs doivent graduellement établir l'infrastructure de communication durant une phase d'initialisation. Cette infrastructure doit leur permettre de répondre aux requêtes venant des sites distants, d'interagir avec l'environnement physique, réagir aux données captées, et transmettre ces données via une communication multi-sauts [KHE 04].

Différemment des réseaux ad hoc traditionnels, les réseaux de capteurs exigent des nouvelles limitations pour la conception des protocoles de communication, tout en considérant d'autres facteurs, tels que les limitations matérielles des micro-nœuds, la mobilité et la consommation d'énergie. De ce fait, il ne serait pas judicieux de modifier la pile protocolaire existante dans les réseaux ad hoc traditionnels pour l'utiliser dans les réseaux de capteurs sans fil [KHE 04].

La pile protocolaire utilisée par les nœuds du réseau, est illustrée par la Figure 1.6. Cette pile prend en charge le problème de consommation d'énergie, intègre le traitement des données transmises dans les protocoles de routage, et facilite le travail coopératif entre les capteurs. Ce modèle comprend 5 couches qui ont les mêmes fonctions que celles du modèle **OSI**, la couche application, transport, réseau, liaison de données et physique, ainsi que 3 couches pour la gestion de la puissance d'énergie, la gestion de la mobilité ainsi que la gestion des tâches (interrogation du réseau de capteurs) [CHA 08].

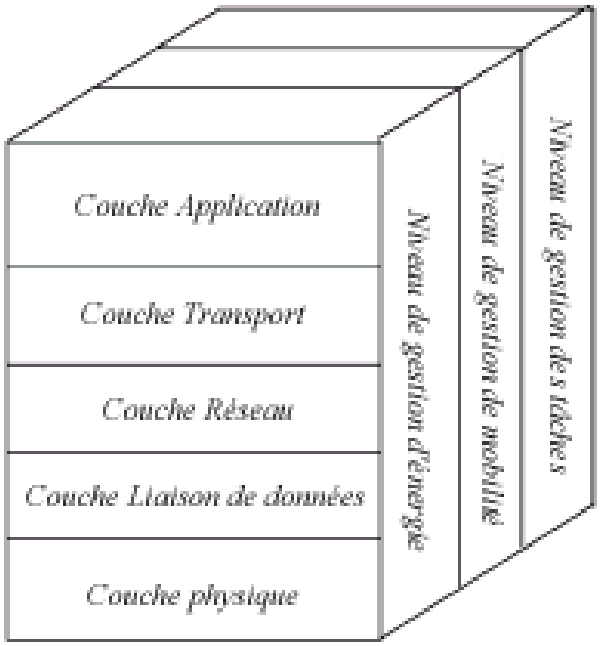


Figure 1.6 : Modèle en couche dans les réseaux de capteurs sans fil. [CHA 08]

Le rôle de ce modèle consiste à standardiser la communication entre les composants du réseau afin que différents constructeurs puissent mettre au point des produits (logiciels

ou matériels) compatibles. Le but d'un système en couches est de séparer le problème en différentes parties (les couches) selon leur niveau d'abstraction. Chaque couche du modèle communique avec une couche adjacente (celle du dessus ou celle du dessous). Chaque couche utilise ainsi les services des couches inférieures et en fournit à celle de niveau supérieur [CHA 08].

Suivant la fonctionnalité des capteurs, différentes applications peuvent être utilisées et bâties sur la couche application, qui est la plus haute couche du modèle, et assure l'interface avec les applications exécutées par le nœud. La couche transport sert à maintenir le flux de données en cas de nécessité dans les applications utilisées, particulièrement lors d'une connexion avec Internet, tandis que la couche réseau s'occupe du routage des données fournies par la couche transport [KHE 04].

Comme l'environnement des réseaux de capteurs est bruyant et les nœuds peuvent être mobiles, la couche **MAC** (**M**edia **A**ccess **C**ontrol) doit garantir une faible consommation d'énergie et un taux de collision minimum entre les données diffusées par les nœuds voisins. Enfin, la couche physique doit assurer des techniques d'émission, de réception et de modulation de données simples mais robustes [KHE 04].

Les niveaux de gestion d'énergie, de mobilité et de tâches sont responsables du contrôle de l'énergie consommée, des mouvements des nœuds et de la distribution des tâches à travers toute la pile protocolaire. Ces niveaux permettent aux capteurs de coordonner leurs tâches et minimiser la consommation d'énergie, [ALO 07]:

- **Le niveau de gestion d'énergie**: Un nœud capteur gère sa consommation d'énergie durant les opérations de captages, de traitements et de communications grâce au plan de gestion d'énergie. Par exemple, après la réception d'un message, le capteur peut éteindre son récepteur afin d'éviter la duplication du message déjà reçu. En outre, si le niveau d'énergie devient bas, le nœud diffuse à ses voisins une alerte les informant qu'il ne peut pas participer au processus de routage. L'énergie restante est réservée au captage.

- **Le niveau de gestion de mobilité** : ce niveau est un plan qui détecte et enregistre le mouvement du nœud capteur. Ainsi, un retour arrière vers l'utilisateur est toujours maintenu et le nœud peut garder la trace de ses voisins. En déterminant leurs voisins, les nœuds capteurs peuvent équilibrer l'utilisation de leur énergie [KHE 04].

- **Le niveau de gestion de tâche** : ce niveau est un plan qui balance et ordonnance les différentes tâches de captage de données dans une région spécifique. Notons qu'il n'est pas nécessaire que tous les nœuds de cette région effectuent la tâche de captage au même temps; certains nœuds exécutent cette tâche plus que d'autres selon leur niveau de batterie.

1.4. Réseaux ad hoc vs reseaux de capteurs sans fil

La différence principale entre les réseaux de capteurs sans fil et les réseaux ad hoc est que dans les réseaux Ad hoc traditionnels, les tâches qui traitent l'organisation, le routage, et la gestion de mobilité visent l'optimisation des différents paramètres de qualité de service (**QoS**), tel que l'efficacité de débit et des délais de transmission sous la contrainte de mobilité. La consommation d'énergie a une importance secondaire, puisque les batteries des unités mobiles utilisées peuvent être facilement remplacées.

Cependant, les réseaux de capteurs englobent un grand nombre de nœuds possédant des sources d'énergie irremplaçable à cause de leur utilisation distante non-assistée dans les environnements hostiles. Ces capteurs communiquent entre eux avec un taux de transmission très faible de l'ordre de 1 à 100kbps. Pour cela, et contrairement aux réseaux ad hoc classiques, le but principal des techniques utilisées est de prolonger la durée de vie des batteries afin de prévenir les dégradations de connectivité dans le réseau.

Enfin, les communications dans les réseaux de capteurs sont, dans la plus part des temps, unidirectionnels [KHE 04].

Ad hoc	**WSN**
1. Générique / communication	1. Objectif ciblé
2. Chaque nœud a son propre objectif	2. Nœuds collaborent pour remplir un objectif
3. Flot de données « Any to any »	3. Flot de données « Many to one »
4. Notion d'ID	4. Très grand nombre des nœuds n'ayant pas tous une ID
5. Débit est majeur	5. Energie est un facteur déterminant
6. Communication point à point.	6. Utilisation des broadcast
7. Portables, PDAs, …	7. Petits capteurs

8. Mobilité	8. Mobilité faible
9. Bande passante	9. Energie (Durée de vie du réseau)

Tableau 1.1 : Réseaux ad hoc vs réseaux de capteurs sans fil.

1.5. Systèmes d'exploitation pour les capteurs

5.1 Les Systèmes d'exploitation classiques

Les **OS** classiques sont généralement conçus pour un usage générique. Ils sont ainsi conçus en supposant une disponibilité sans limite des ressources. Leur objectif est la facilité d'usage, la rapidité et l'efficacité. Parmi leurs caractéristiques, on peut citer, [CHA 08]:

- Architecture Multi thread → Mémoire importante
- Modèle E/S
- Séparation entre espace noyau et utilisateur
- Pas de contraintes d'énergie
- Ressources disponibles

A partir de ces caractéristique, les **OS** classiques ne sont pas appropriés aux "motes" (nœuds capteurs), vus que ces derniers sont caractérisés par :

- Ressources énergétiques basses
- Mémoire limitée
- CPU lente
- Petite taille
- Parallélisme matériel limité
- Communication radio
- Bande-passante faible
- Portée radio courte

Il existe plusieurs systèmes d'exploitation destinés aux réseaux de capteurs parmi eux **TinyOS**, **MagnetOS**, **OSPM**, **EYES OS**, **SenOS**, **PicOS**. En effet, **TinyOS** est le plus répandu et le plus utilisé des OS pour les réseaux de capteurs sans-fil [CHA 08].

5.2 Le système d'exploitation TinyOS

TinyOS est un système d'exploitation open-source spécialement conçu pour les réseaux de capteurs sans fil, développé par l'université américaine de **Berkeley**. Sa conception a été entièrement réalisée en **NesC** [NES 03], langage orienté composant proche du **C** [TIN 11].

1.6. Caractéristiques des réseaux de capteurs sans fil

Comme tout autre système distribué, les réseaux de capteurs sans fil sont des réseaux constitués d'un ensemble de machines autonomes liés dans le but d'achever une tâche commune. La question qui se pose est : serait-il possible d'appliquer les algorithmes déjà existant dans les systèmes distribués filaires aux réseaux de capteurs ? Malheureusement, rare sont les résultats des travaux précédents qui peuvent leur être appliqués. La raison est que les recherches entreprises sur les systèmes distribués les considéraient comme étant filaires, possédant des ressources infinies en terme d'énergie et utilisant des interfaces telles que les écrans et les claviers, ainsi qu`elles donnaient plus d'importance aux machines elles-mêmes qu'à leurs positions [STA 06]. Contrairement à cela les réseaux de capteurs sans fil ont de faibles ressources en énergie, sont en temps réel, les nœuds utilisent les ondes radio pour communiquer entre eux, utilisent les capteurs comme interfaces et leurs positions sont d'une importance cruciale.

6.1 MAC

Un bon protocole **MAC** dans les réseaux de capteurs sans fil doit consommer un minimum d'énergie, éviter les collisions, être implémenté avec un code de taille réduite ne nécessitant pas une grande mémoire et être tolérant aux changements de fréquences [STA 06].

Les sources de consommation d´énergie sur un nœud capteur sont le module radio, le micro-processeur et le capteur. La communication radio est souvent la plus consommatrice parmi les trois. La consommation d'une unité de captage varie dans une très large plage selon son type. Un dispositif de captage consommant beaucoup d'énergie est souvent alimenté par sa propre source énergétique [CHA 09]. Dans quelques protocoles **MAC** sans fil, un nœud qui ne reçoit pas et qui n'envoie pas passe dans un état de sommeil afin d'économiser son énergie. Le tableau 1.2 montre l'énergie consommée par le module radio d'une carte **B2400ZB-tiny** pour chaque état.

État	Énergie consommée
Émission	26 mW
Réception ou écoute	29 mW
Veille	15 µW
Sommeil	3 µW

Tableau 1.2 : Energies consommées par le composant radio d'une carte.

Un module **MAC** économe en énergie essayera d'utiliser le moins souvent le module radio. L'utilisation inutile du module est due essentiellement à l'overhearing, des Idlelistening, des envois infructueux et des collisions.

L'Overhearing : est la réception par un nœud d'une trame qui ne lui est pas destinée. L'énergie consommée pour la réception et le traitement des données de cette trame est perdue et sans aucun intérêt.

L'Idlelistening : est le fait de demander à un nœud de rester éveillé sans qu'il ne reçoive de trame ou qu'il n'en transmette une. Le fait que son module radio soit activé et prêt pour recevoir consomme autant d'énergie que pour la réception.

L'envoi infructueux : cela arrive quant un nœud essaie de communiquer avec un autre nœud qui n'est plus accessible parce qu'il est en mode sommeil par exemple (ou hors de portée). Le nœud émetteur est en attente d'un acquittement et il retransmettra donc la même trame plusieurs fois ; Il consomme alors de l'énergie du fait qu'il soit resté en mode transmission et en mode réception pour l'éventuel acquittement.

Les protocoles **MAC** sont souvent classés en trois catégories: les protocoles basés sur un séquencement temporel, les protocoles basés sur un évitement de collision probabiliste et les protocoles hybrides. Dans les protocoles basés sur un séquencement temporel, le temps est découpé en intervalles, chaque intervalle est alloué à un seul nœud à la fois. De cette façon un nœud est garantit d'être le seul à utiliser le canal dans l'intervalle qui lui est consacré. La méthode **TRAMA** (**TR**affic-**A**daptive **M**edium **A**ccess control) est un exemple des protocoles basés sur un séquencement temporel, elle se base sur le trafic

annoncé par les nœuds pour désigner les émetteurs et les récepteurs pour chaque intervalle de temps. Pour ce faire, **TRAMA** applique trois mécanismes [STA 06]:

- un protocole appelé **NP** (**N**eighbor **P**rotocol) qui permet l'échange des tables de voisinage entre les nœuds. Ainsi, chaque nœud aura connaissance de ses voisins à deux sauts.
- un protocole d'échange de calendriers appelé **SEP** (**S**chedule **E**xchange **P**rotocol) où chaque nœud annonce ce qu'il a comme trafic à envoyer en précisant les récepteurs concernés.
- un protocole d'élection adaptative **AEA** (**A**daptative **E**lection **A**lgorithm) qui choisit les émetteurs et les récepteurs dans un intervalle de temps donné en fonction des informations collectées par **NP** et **SEP**.

La deuxième catégorie de protocoles est celle des protocoles basés sur la contention (Méthode d'occupation d'une liaison de communication accessible par plusieurs utilisateurs qui peuvent émettre à des moments aléatoires). Dans ce type de protocole, les nœuds peuvent envoyer au même moment. Afin d'éviter la collision, un algorithme de la famille **CSMA/CA** est utilisé. Le **CSMA/CA** comme le **CSMA/CD** impose à un émetteur de s'assurer que le canal est libre avant d'émettre. Un exemple de ces protocoles est le **S-MAC** (**S**ensor-**MAC**) qui est conçu pour assurer une méthode d'accès économe en énergie pour les réseaux de capteurs sans fil. Pour ce faire, les nœuds se mettent en mode sommeil pendant une certaine durée et se réveillent pour écouter le médium pendant une autre durée. Les nœuds échangent leur calendrier de périodes d'écoute en les diffusant à leurs voisins directs. Ainsi, chaque nœud connaît le calendrier de ses voisins et sait quand il faut se réveiller pour communiquer avec un nœud à sa portée. Les nœuds accèdent au médium en utilisant le **CSMA/CA**. Les nœuds non concernés par l'envoi ou la réception d'un message peuvent dormir pendant la durée de l'échange.

La dernière catégorie de protocoles **MAC** est celle des protocoles hybrides. Ce sont des protocoles qui font usage des principes des deux types de protocoles vus précédemment. Le protocole **Z-MAC**, par exemple, utilise un découpage temporel et gère les accès durant les intervalles avec la méthode **CSMA/CA**. Une fois le réseau déployé, **Z-MAC** commence par une phase de découverte à deux sauts pour chaque nœud, ceci est suivi par l'exécution d'un protocole nommé **DRAND** qui est un protocole qui assure qu'un intervalle de temps n'est pas assigné à deux nœuds situés à moins de deux sauts l'un de l'autre.

L’acquittement systématique est mal adapté à des réseaux denses tels que les **WSN**s. Il provoque une surcharge du réseau et donc des collisions avec les données utiles échangées. Les acquittements par piggy-backing sont donc les plus à privilégiés (i.e. L’acquittement d'un paquet allant de A vers B peut être véhiculé par un paquet utile allant de B vers A) [STA 06].

6.2 Routage

La communication entre les nœuds dans les **WSN** se fait par la méthode du multi-saut; les techniques de routage traditionnelles pour les réseaux Ad hoc, ne peuvent pas être appliquées dans les réseaux de capteur sans fil. Le routage internet, par exemple, assume une connexion câblée très fiable, les erreurs sur les données transportées par les paquets sont rares ; ceci n’est pas le cas dans les **WSN**s. De plus les réseaux de capteurs sans fil sont souvent considérés asymétrique (si le nœud A peut envoyer des messages au nœud B cela n’implique pas que le nœud B peut envoyer des messages à A).

En général les protocoles de routage dans les réseaux de capteurs sans fil sont classés en trois catégories :

Les protocoles Data-centric : ce sont des protocoles qui ne nécessitent pas de mécanismes d’adressages. Un nœud ayant extrait une information annoncera ses données par un paquet ADV ; les nœuds intéressés répondront par un ACK et c’est par la suite que les données seront envoyées. Dans la technique de l’inondation, un message est propagé dans tout le réseau jusqu'à atteindre la destination ou jusqu’à atteindre un nombre maximal de sauts. Dans la technique du commérage, chaque nœud choisit aléatoirement un voisin pour lui faire passer le message.

Les protocoles hiérarchiques : dans ce genre de protocole, on parle de clusters (groupes de nœuds). Chaque cluster a un chef qui se charge d’envoyer les données générées par son cluster aux autres chefs de clusters. Le chef de cluster est choisi soit à tour de rôle selon son niveau d’énergie ou soit selon le nombre de ses voisins ; dans ce cas le nœud avec le plus grand nombre de voisins sera choisi. Dans la méthode **TEEN** (**T**hreshold sensitive **E**nergy **E**fficient sensor **N**etwork protocol) (proposé par **Manjeshwar** et autres dans [MAN 01]), la transmission est faite uniquement quand la valeur détectée est supérieure à un certain seuil fixé par le cluster chef. Ensuite, il ne transmet pas la même donnée tant que la valeur n’a pas changé d’un seuil de variation donné.

Les protocoles basés sur la position : ce type de protocoles considère que les nœuds connaissent leur position et sont capables de connaitre la position des autres nœuds. Ainsi, les messages sont dirigés vers la région dans laquelle se trouve la destination. Apres le déploiement du réseau, chaque nœud commence tout d'abord par localiser sa propre position en utilisant des algorithmes dédiés. Il procède par la suite à l'étape d'initialisation de sa table de routage. Celle-ci doit inclure l'identifiant de chacun de ses voisins ainsi que sa position. D'autres informations sont utiles, telles que l'énergie restante et la fiabilité du lien peuvent y être ajoutées.

Le protocole **GEAR** (**G**eographical and **E**nergy **A**ware **R**outing) découpe le réseau en plusieurs régions, quand un nœud situé dans une région donnée veut envoyer un message à un autre nœud situé dans une autre région du réseau, le message est tout d'abord acheminé jusqu'à la région cible. L'émetteur envoie le paquet au nœud le plus proche de la région parmi ses voisins. Cette méthode nommée **GF** (**G**eographic **F**orwarding) a était largement approuvée pour être utilisée dans les réseaux de capteur sans fil. L'avantage de cette technique est que le nœud n'a plus besoin de maintenir une table excessivement large en taille, Il peut simplement transmettre les données en se basant sur la position géographique de ses voisins. La taille des informations stockées dans la table dépendrait de la densité du réseau plutôt que de sa taille [SUB 07].

Trois cas de Figure se présentent quand le message arrive à destination : soit le message contient l'identifiant d'un unique nœud couvrant cette région ceci, est appelé le unicast. La destination peut être aussi tout les nœuds présent dans la zone spécifiée, ceci est nommé multicast. Ou bien aucun nœud dans la zone ne doit recevoir le message, c'est le cas de l'anycast. Le Protocol **SPEED** (A Stateless Protocol for Real-time Communication) [TIA 03] supporte ces trois opérations. Ce protocole apporte une métrique supplémentaire par rapport à **GEAR** qui est le délai d'acheminement pour augmenter ainsi la qualité de service.

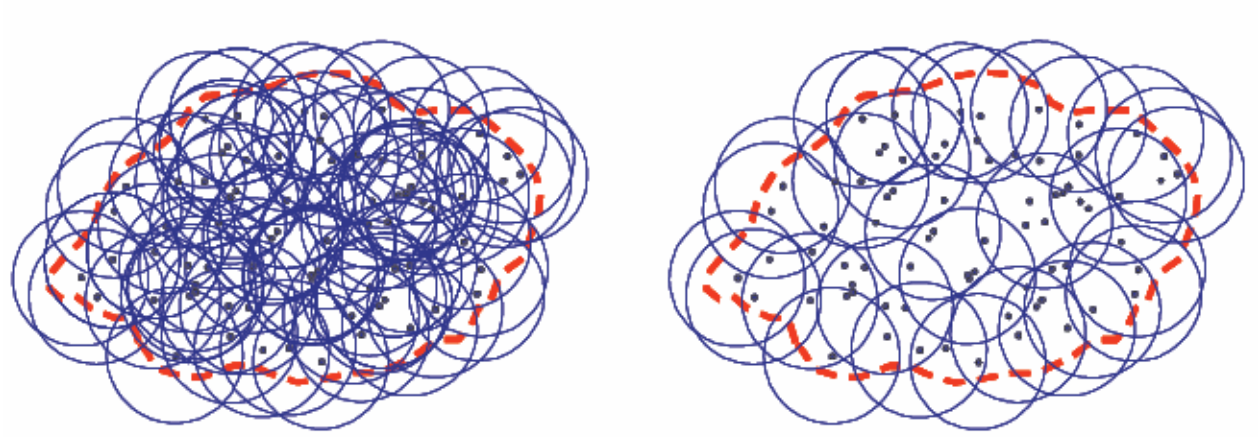
Un nœud désirant obtenir une certaine information exprime son besoin en diffusant une requête sur tout le réseau. Si un autre nœud reçoit la requête et qu`il est en possession de cette information, il répond avec une paire [attribut, valeur], cette paire est envoyée à l'initiateur de la requête avec la possibilité de prendre plusieurs chemins pour plus de fiabilité. De nombreuses recherches antérieures ont indiqué que plusieurs nœuds dans un réseau de capteurs sans fil ne répondent pas au principe de la symétrie, ce qui veut dire que

le chemin que prend le paquet de l'initiateur de la requête vers son receveur n'est pas forcément le même qu'il va prendre dans son chemin de retour.

6.3 Ordonnancement

Les réseaux de capteurs sont généralement denses et redondants, plusieurs capteurs peuvent observer une même portion de la zone de déploiement. La probabilité pour qu'un évènement soit capté et signalé par plusieurs nœuds est importante. Ceci peut être une source de désagrément au niveau de l'accès au medium. Plusieurs nœuds essayeront d'envoyer simultanément les mêmes données provoquant une très grande quantité de collisions.

Ordonnancer l'activité dans un réseau de capteurs consiste à alterner les charges de façon à épuiser l'énergie des nœuds équitablement et à minimiser la quantité de nœuds opérant sur une même région. Pendant qu'une partie participe à l'application, les autres sont dans un mode passif économisant ainsi leur énergie [GAL 07].



(a) Un réseau de capteurs déployé.

(b) Seule une partie des nœuds est nécessaire à la couverture de la zone.

Figure 1. 7 : Ordonnancement d'activité dans les WSNs. [GAL 07]

Cette approche (montrée par la Figure 1.7), bien qu'elle se soit révélée être primordiale pour l'économie d'énergie, a engendré une problématique relative à la couverture. Il y a en effet des risques pour qu'une fois un nœud passé en mode de conservation d'énergie, la zone qui était jusqu'ici couverte par ce nœud devienne non contrôlée. Aucun événement qui parviendra dans cette région ne sera intercepté.

La plupart des protocoles d'ordonnancement d'activités ayant pour critère la couverture de surface utilisent un outil d'évaluation de couverture; tout nœud devant décider de son statut d'activité doit au préalable évaluer la couverture fournie par ses

voisins de communication. Il ne peut être passif que si la surveillance de sa zone est assurée par ces derniers. Les nœuds actifs doivent surveiller une zone aussi large que celle couverte par l'ensemble des nœuds. Deux catégories de protocoles existent pour assurer le non dégradation de la couverture : les protocoles à couverture simple et les protocoles à couverture multiple.

Dans le contexte de couverture simple, tout point d'une zone doit être couvert par au moins un capteur pour conclure à la couverture totale de cette dernière. Dans le protocole **PEAS** (A Robust Energy Conserving protocol) par exemple, tous les nœuds sont initialement passifs. Périodiquement, chaque capteur sort de ce mode pour envoyer un message dit de sondage. Tous les nœuds actifs situés dans un rayon R (identique pour tous les nœuds) reçoivent ce message. Ils doivent ensuite évaluer la distance les séparant du nœud émetteur grâce à l'intensité du signal reçu ou par des mesures de délais de transmission. Si cette distance est inférieure à une certaine valeur, ils avertissent le nœud émetteur afin qu'il puisse demeurer passif. Dans le cas contraire, aucun message n'est envoyé et le nœud à l'origine du message de sonde décide de devenir actif. Dès lors qu'un nœud prend cette décision, il conserve ce statut jusqu'à épuisement de ses ressources énergétiques. Ce protocole est extrêmement tolérant aux fautes (pertes de messages, panne d'un capteur, etc.).

Dans le cas d'une couverture multiple, tout nœud ne peut décider d'être passif que si la zone couverte par ce nœud est totalement couverte par un ensemble connecté de voisins. Cet ensemble doit être d'une taille k données. Dans cet algorithme, les messages échangés par les nœuds contiennent leur position et leur rayon de communication théorique, permettant ainsi au nœud d'évaluer la connexité de l'ensemble des voisins. Trois approches d'ordonnancement existent:

La première approche est l'approche centralisée dans laquelle une entité centrale a pour but d'affilier à chaque nœud son rôle, l'entité centrale doit donc être informé de n'importe qu'elle changement de topologie dans le réseau. Cette approche nécessite malheureusement l'existence d'une infrastructure préalable au déploiement du réseau.

La deuxième approche est l'approche hiérarchique. Dans cette approche, le réseau est divisé en plusieurs Clusters. L'affiliation des rôles se fera au sein du Cluster chef. Dans un ordonnancement hiérarchique, les informations de changements de topologie ne sont propagées qu'au sein des sous-structures (Cluster). Comme tout autre protocole nécessitant la construction de cluster, le problème majeur est lié à la construction de ces sous-ensembles.

Concernant la troisième approche qui est l'approche localisée, chaque nœud décide de sa propre activité en ne se basant que sur l'observation de ses propres voisins. Les approches localisées ne reposent sur aucune infrastructure et n'ont pas vocation à hiérarchiser le réseau. Le comportement de chaque objet n'est influencé que par ceux de ses voisins directs. Les changements de topologie du réseau (dus à la mobilité, aux pannes ou à des changements de statut) ne sont par conséquent vécus par les nœuds que comme de simples modifications de leurs voisinages.

6.4 Tolérance aux pannes

Certains nœuds capteurs peuvent être bloqués ou tomber en panne à cause d'un manque d'énergie, d'un dégât matériel ou d'une interférence environnementale. La panne d'un nœud capteur ne doit pas affecter le fonctionnement global de son réseau. C'est le problème de fiabilité ou de tolérance aux pannes. La tolérance aux pannes est donc la capacité de maintenir les fonctionnalités du réseau sans interruption due à une panne d'un nœud capteur. Deux catégories d'algorithmes de tolérances aux pannes existent :

Algorithme préventif

Se sont des algorithmes qui permettent de retarder où d'éviter des types de pannes afin de garder le réseau fonctionnel le plus longtemps possible. Les techniques de conservation d'énergie à titre d'exemple, permettent de consommer moins d'énergie et évitent donc une extinction prématurée de la batterie. le mécanisme de mise en veille en est un parfait exemple.

Le routage multi-chemin utilise un algorithme préventif pour déterminer plusieurs chemins depuis chaque capteur vers un autre nœud. Ceci garantit la présence de plus d'un chemin fiable pour la transmission et offre une reprise rapide du transfert en cas de panne sur le premier chemin sélectionné (choisir un des chemins qui restent). Certains protocoles proposent comme solution tolérante aux pannes, la sélection d'un ensemble de nœuds mobiles chargés de se déplacer entre les capteurs et collecter les données captées. Ceci réduira l'énergie consommée au niveau de chaque capteur en éliminant sa tâche de transmission. Un nœud mobile est généralement doté d'une batterie plus importante que celle d'un nœud capteur.

L'agrégation est aussi considérée comme une approche préventive. L'opération d'agrégation effectue un traitement supplémentaire sur les données brutes captées depuis l'environnement. Un nœud agrégateur combine les données provenant de plusieurs nœuds

en une information significative, ce qui réduit considérablement la quantité de données transmises, demande moins d'énergie et augmente ainsi la durée de vie du réseau.

Algorithme curatif

Dans ce cas le mécanisme de tolérance aux pannes implémenté n'est exécuté qu'après la détection de la panne. Plusieurs algorithmes de recouvrement après les pannes sont proposés dans la littérature, par exemple : les techniques de recouvrement du chemin de routage qui sont des techniques qui permettent de créer un nouveau chemin pour retransmettre les données.

6.5 La 'scalability'

La surveillance d'un phénomène peut nécessiter le déploiement d'un nombre de nœuds qui est de l'ordre de plusieurs milliers de capteurs. Suivant l'application, ce nombre peut encore augmenter jusqu'à des millions de capteurs, les nouveaux schémas doivent pouvoir garantir un bon fonctionnement avec ce nombre élevé de capteurs et ils doivent aussi exploiter la nature fortement dense des réseaux de capteurs.

Cette densité peut varier entre quelques capteurs jusqu'à plusieurs centaines de capteurs dans une région de taille inférieure à 10 mètres de diamètre. La densité peut être calculée comme suit :

$$d(r) = (N \Pi r^2) / A$$

Où N est le nombre de nœuds capteurs éparpillés dans la région A, et r le domaine de transmission. *d(r)* donne alors le nombre de nœuds se trouvant dans le domaine de transmission r d'un nœud donné dans la région A.

La densité des nœuds dépend également de l'application pour laquelle le réseau de capteurs est employé. Une application de diagnostic de machines nécessite par exemple une densité proche de 300 nœuds par région de $25m^2$, tandis que la densité nécessaire pour le contrôle des véhicules ne peut pas dépasser 10 capteurs par une région de même taille.

6.6 Gestion de l'énergie

La possibilité d'intégrer les réseaux de capteur dans des domaines tels que l'environnement, l'automobile, le bâtiment et tant d'autres, accroit au fur et à mesure que le prix des nœuds décroit. Cependant, très peu de compagnies ont décidé d'intégrer les **WSN** dans leurs projets. Malgré une forte avancée technologique dans la puissance de calcul et dans la capacité en mémoire, le problème de l'énergie est toujours le même.

Comme les nœuds capteurs sont des composant micro-électroniques, ils ne peuvent être équipés que par des sources limitées d'énergie (<0.5 Ampère-heure, 1.2 V). De plus, dans certaines applications, ces nœuds ne peuvent pas être dotés de mécanismes de rechargement d'énergie, par conséquent, la durée de vie d'un nœud capteur dépend fortement de la durée de vie de la batterie associée [KHE 04]. Ils doivent donc fonctionner avec un bilan énergétique modéré. En outre, les capteurs doivent le plus souvent avoir une durée de vie de l'ordre de plusieurs mois, voir de quelques années [KAC 09]. Voici présentés ci-dessous les différents facteurs provoquant la dissipation de l'énergie d'un nœud [RAG 02] :

Le MCU : Généralement les microcontrôleurs possèdent plusieurs modes de fonctionnement ; actif, inoccupé (idle) et sommeil. La quantité d'énergie consommée par un **MCU** varie d'un mode à un autre.

La radio : la radio opère dans quatre modes de fonctionnement : émission, réception, inoccupation et sommeil. Une observation importante dans la plupart des radios est que le mode inoccupé induit une consommation d'énergie significative, presque égale à la consommation en mode réception [YAX 01]. Ainsi, il est plus judicieux d'éteindre complètement la radio plutôt que de passer en mode "inoccupé" quand l'on a ni à émettre ni à recevoir. Un autre facteur déterminant est que, le passage de la radio d'un mode à un autre engendre une dissipation d'énergie importante due à l'activité des circuits électroniques. Par exemple, quand la radio passe du mode sommeil au mode émission pour envoyer un paquet, une importante quantité d'énergie est consommée pour le démarrage de l'émetteur lui-même [RAG 02].

L'unité de captage : On peut citer parmi les opérations effectuées par cette unité: l'échantillonnage, la conversion des signaux physiques en signaux électriques, et la conversion analogique-numérique. Étant donné la diversité des capteurs, il n'y a pas de valeurs typiques de l'énergie consommée [CAP 09].

Un autre aspect non négligeable est le phénomène d'autodécharge de la batterie. Cette dernière perd de sa capacité et se vide de ces ressources au fil du temps [RAG 02]. Le coût d'une transmission d'un bit d'information est approximativement le même que le coût nécessaire au calcul d'un millier d'opérations [PRI 00]. La limitation des messages échangés dans les réseaux pour faire un gain d'énergie n'est possible qu'en améliorant les solutions de routage, de synchronisation ou de localisation. Le principe de

l'ordonnancement est souvent utilisé pour obtenir le gain d'énergie, est de mettre une partie des nœuds du réseau en état de repos, tandis que les autres capteurs restent éveillés dans le but d'accomplir la tâche qui leur est incombée [GOU 04].

6.7 Synchronisation

Plusieurs applications émergentes dans les réseaux de capteurs nécessitent que les nœuds du réseau soient cadencés à une même horloge. Une horloge globale peut aider à mieux analyser les données et prédire le comportement futur du system [QUN 04]. La synchronisation joue aussi un rôle important dans la conservation de l'énergie. Les capteurs peuvent être programmés pour qu'ils se mettent en état de repos ou d'éveil simultanément. Dans un réseau non synchronisé, un nœud gaspillerait une partie de son énergie à attendre que l'un de ses voisins se réveille.

Quelques approches traditionnelles déjà employées dans les systèmes distribués tel que le vecteur d'horloge, sont généralement non praticables dans les réseaux de capteur. La raison est due à la taille importante du réseau ; la gestion de ce vecteur se révèle être impossible. De plus le vecteur est de nature abstraite est n'indique pas la durée des événements en mesures physiques tels que les minutes et les secondes [STO 05].

Nelson et autres [NEL 02] ont proposé une méthode de synchronisation nommée **RBS** (**R**eference-**B**roadcast **S**ynchronization). Un nœud diffuse un message à tous ses voisins. Le voisin qui reçoit le paquet utilise la date à laquelle le paquet a été envoyé et celle a laquelle le paquet a été reçu comme référence pour fixer son horloge. Cette méthode peut être élargie à tout le réseau par le multi saut, malgré qu'à la base, elle est utilisée pour synchroniser un groupe de voisins.

6.8 Les coûts de production

Le coût de production d'un seul micro-capteur est très important pour l'évaluation du coût global du réseau, si ce dernier est supérieur à celui nécessaire pour le déploiement des capteurs classiques, l'utilisation de cette nouvelle technologie ne serait pas financièrement justifiée.

1.7. Principaux défis des reseaux de capteurs sans fil

7.1 L'Energie

→Assurer une consommation répartie de l'énergie au sein du réseau.

→ Gérer la consommation : 1 octet transmis = 11000 cycles.

7.2 Durée de vie

La durée de vie des capteurs est limitée par des batteries :

→ Maximiser la durée de vie du réseau de capteurs

→Voir l'instant où :

1. Le premier nœud tombe en panne
2. Une certaine fraction tombe en panne
3. Perte de la couverture /connectivité.

7.3 Densité

Il existe des études focalisées sur la densité des réseaux de capteurs, et donc le passage à l'échelle.

Les chercheurs ont conclu qu'il n'existe pas de meilleures solutions génériques pour ces différents défis (orientées application) [FLE 04].

1.8. La sécurité dans les réseaux de capteurs sans fil

Les propriétés des réseaux de capteurs sont à double tranchant. Certes ils permettent une grande facilité de production et de déploiement, mais rendent le système global de communication assez sensible à un certain nombre de défaillances. Afin d'assurer un déploiement à large échelle de cette technologie, il est nécessaire de pallier ces problèmes de sécurité aux différents niveaux d'une architecture **OSI**. En fonction de la couche réseau visée, plusieurs types d'attaques existent, est sont présentés sur [WOO 02].

1.9. Domaines d'application des réseaux de capteurs

Contrairement aux réseaux **MANET**, qui n'ont pas pu connaître un vrai succès, les **RCSF (R**éseaux de **C**apteur **S**ans **F**il)/**WSN (W**ireless **S**ensor **N**etwork) ont su attirer un nombre croissant d'industriels, vu leur réalisme et leur apport concret. En effet, le besoin d'un suivi continu d'un environnement donné est assez courant dans diverses activités de la société.

Et parmi les applications d'utilisations des capteurs (voir Figure 1.8), on trouve [ZAF 08]:

- Supervision de l'état de l'air, du sol,
- Environnements dangereux (centrales nucléaires, etc...),
- Surveillance d'habitat,
- Détection de séisme,
- Surveillance militaire,
- Inventaires,

- Espaces intelligents

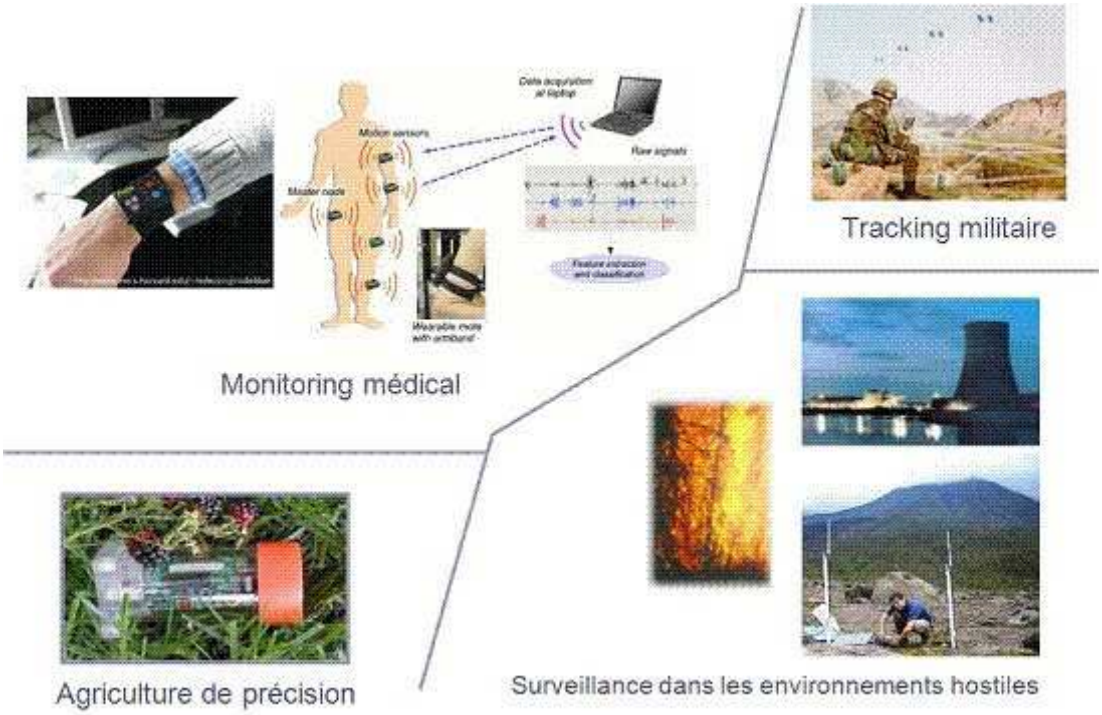


Figure 1.8 : Domaines d'application.

9.1 Domaine militaire

Que ce soit pour contrôler la communication, pour détecter la présence de menaces biologiques ou chimiques ou bien pour surveiller une zone alliée ou ennemie, on ne peut qu'imaginer l'importance des **WSN** dans un tel domaine.

Des nœuds peuvent être déployés à partir d'un avion afin d'étudier une zone et détecter l`éventuelle présence d'ennemis avant d'y envoyer des troupes. Les nœuds pourraient également être programmés pour avoir un comportement neutre jusqu'à ce qu'un évènement se produise, comme par exemple le passage d'un corps devant le capteur. De meilleures stratégies seraient envisagées et entreprises en équipant le personnel de capteurs sans fil ; des données en temps réel sur l'évolution de la situation pourront ainsi être récoltées et transmises à la base centrale. Si en plus le terrain est infesté de nœuds alliés, le soldat pourra à tout moment faire une requête sur l'emplacement des troupes ennemies par rapport à sa propre position.

En temps de guerre chimique ou biologique, un réseau de capteur couvrant une certaine zone alliée détectera la présence d'une quelconque toxine. Il serait possible

d’obtenir des informations utiles telles que la concentration et la nature de la toxine sans risque d’exposition humaine [CHA 08].

9.2 Domaine environnemental

Il est désormais plus facile grâce au **WSN** d’observer des phénomènes environnementaux qui étaient très difficiles ou même parfois impossibles à étudier dans le passé. Une quantité importante de motes peut être projetée sur une zone difficile d’accès. Il est aussi possible d’étudier le comportement de certaines espèces animales ou végétales sans que la présence de l’homme leur soit néfaste. L’étude menée sur **Great Duck Island** [MAI 02] en un parfait exemple : les chercheurs avaient comme objectif l’étude d’une espèce d’oiseaux maritimes très rare. Ils avaient pris le soin de positionner un réseau de capteurs comportant 32 nœuds sur l’ile avant l’arrivée de l’espèce menacée ce qui a permit d’étudier de près ces oiseaux durant leur période de reproduction.

Dans des zones forestières fréquemment victime d’incendies (voir Figure 1.9), des capteurs permettraient d’intercepter des feux de foret dés leur déclenchement. Ceci se révèlera être un atout majeur pour les pompiers, ils leur seraient ainsi plus facile de maitriser le feu avant qu’il n’atteigne des proportions désastreuses [CHA 08].

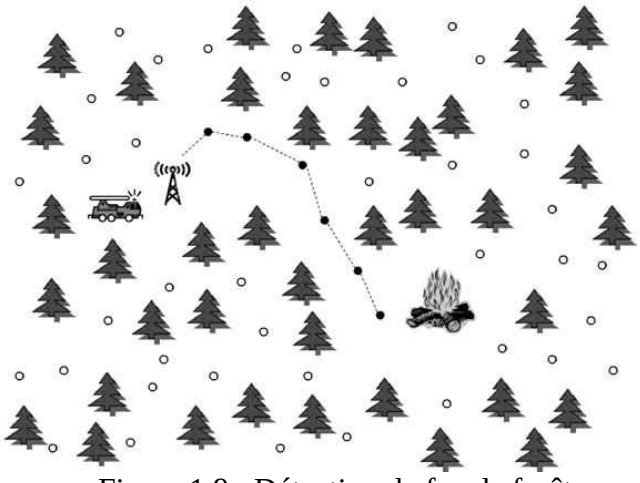


Figure 1.9 : Détection de feu de forêt.

9.3 Domaine médical

Les réseaux de capteurs ont le potentiel de révolutionner le domaine médical. Dans les hôpitaux ou dans les cliniques, équiper les patients avec des nœuds contrôlant leurs signes vitaux permettrait d’offrir aux médecins et aux infirmiers un suivi continu du statut de leurs patients. Les malades dans des cas particuliers pourraient bénéficier de plus de liberté dans leur déplacement tout en restant sous contrôle.

Dans l'université d'**Harvard** un projet sous le nom de **CodeBlue [SHN 05]** a fait son apparition avec des tests initiaux très concluant. L'objectif de ce projet est de créer des capteurs vitaux récoltant des informations telles que la saturation en oxygène et le rythme cardiaque du patient. Ces capteurs équipés d'antennes radio aident à localiser le malade en cas de problème. Le projet **Mercury** développé par la même université fait usage de ces capteurs vitaux : un réseau **Mercury** consiste en un nombre de capteurs sans fil et d'une station de base installée dans le domicile du patient (un ordinateur relié à internet peut servir comme une station de base). Les capteurs récoltent régulièrement les signes vitaux de la personne et transfèrent ces informations à la station de base qui est connectée à l'hôpital via internet pour offrir un contrôle continu. En cas de problème, une alerte est envoyée à l'hôpital le plus proche pour une intervention de l'équipe médicale en urgence [CHA 08].

9.4 Applications dans les maisons

Grâce aux avancées technologiques, des nœuds capteurs intelligents peuvent être intégrés dans les appareils électroménagers dans le cadre automatisation des maisons, tel que les aspirateurs, fours micro-ondes, réfrigérateurs, et magnétoscopes Ces nœuds capteurs peuvent interagir entre eux ainsi qu'avec les réseaux externes via Internet ou à travers les satellites pour permettre à l'utilisateur de contrôler plus aisément ces appareils d'une façon locale ou distante.

En plus de l'automatisation des maisons, les réseaux de capteurs sont utilisés lors de la conception des environnements intelligents, ou ils peuvent avoir deux perspectives différentes : la première consiste à adapter cet environnement aux besoins des utilisateurs en terme d'entrées /sorties, c'est l'approche centrée humains. La deuxième, qui est centrée technologie, vise à développer des nouvelles technologies matérielles, solutions réseaux, services middleware pour concevoir ce type d'environnements. Plusieurs exemples de projets de conception des environnements intelligents on été réalisé, nous pouvons citer, par exemple, le laboratoire ***Labscape*** de biologie cellulaire à l'université de **Washington**, où les nœuds capteurs peuvent être intégrés dans les meubles et les différentes appareils pour permettre à ces derniers de communiquer entre eux, ainsi qu'avec la chambre serveur qui peut elle-même communiquer avec d'autres chambres serveur pour consulter les services offerts tel que l'impression, le scanner et le fax. Ces capteurs seront intégrés avec les dispositifs existants pour former un système auto-organisé, auto-configuré et adaptatif basé sur les modèles de théorie de contrôle [SIL 11].

1.10. Conclusion

Dans ce premier chapitre nous avons procédé à l'étude des réseaux de capteurs sans fil qui sont un type de réseau ad hoc. On a étudié leur architecture, leurs caractéristiques, leurs différents domaines d'applications ainsi que les contraintes liées à leur utilisation. Dans le prochain chapitre nous allons aborder l'un des problèmes majeurs dans les réseaux de capteurs sans fil qui est celui de la localisation.

CHAPITRE 2
NOTIONS DE BASE SUR LA LOCALISATION DANS LES WSNS

2.1 Introduction

Dans un bon nombre d'applications pour les réseaux de capteurs, un événement détecté par un capteur n'est utile que si une information relative à sa localisation géographique est fournie. C'est le cas de la surveillance de feu de forêt, le suivi de véhicule, etc. Grâce à l'information de localisation, les protocoles de routage géographique ou orientés position vont pouvoir fonctionner sans le coûteux mécanisme de découverte de route proactive, et ainsi économiser de l'énergie et améliorer le taux d'acheminement. De plus, dans les protocoles de contrôle de topologie, où chaque capteur doit ajuster sa puissance de transmission pour minimiser sa consommation énergétique, les algorithmes ont le plus souvent besoin d'information sur la position des voisins [HEU 08].

Pour localiser chaque capteur, au lieu de se baser sur les balises globalement accessibles, ou les systèmes mondiaux de localisation (**GPS : Global Positioning System**), nous voudrions que le capteur ait une auto-organisation qui le permet d'organiser le système. Dans ce cadre, il y a des algorithmes qui ont besoin d'une information initiale pour leur démarrage, ou d'autres qui n'ont pas besoin de cette information. Certains parmi eux exigent d'avoir un matériel puissant en capacité. Alors que d'autre font des calculs en différé différemment aux algorithmes qui sont capables de les faire par eux mêmes.

Lors de ce chapitre, nous allons donner un état de l'art sur le problème de la localisation, les contraintes qui influencent la localisation et les techniques les plus utilisées par les algorithmes de localisation dans les réseaux de capteur sans fil.

2.2 Formalisation du Problème

Si les perspectives d'utilisation des réseaux de capteurs sont claires et attrayantes, les problématiques qu'engendrent ces réseaux n'en sont pas moins nombreuses. A priori, ils ne dépendent d'aucune infrastructure et les capteurs n'ont aucune information relative au réseau

auquel ils appartiennent. De plus, étant construits de façon ad hoc, ces réseaux doivent être auto-organisant. Parmi les problèmes cruciaux, on va citer celui de la localisation : il s'agit d'attribuer une position géographique (exacte ou estimée) aux capteurs [SAA 08].

Le problème de la localisation diffère selon les hypothèses faites concernant la mobilité et la capacité des capteurs. Afin d'organiser ces différentes configurations, la notation suivante est proposée :

< x, y, z >,

avec x, y Є {S,M} et z Є { Ø, dist, angle}.

Le premier (resp. le second) champ indique si les capteurs non localisés (resp. les ancres) sont statiques (noté S) ou mobiles (noté M). Il existe donc quatre configurations possibles :

1. les capteurs et les ancres sont statiques.
2. les capteurs et les ancres sont mobiles.
3. les capteurs sont statiques et les ancres sont mobiles.
4. les capteurs sont mobiles et les ancres sont statiques.

Le dernier champ indique la capacité de mesure des nœuds : ils peuvent n'avoir aucune capacité (ce qui est noté Ø) ou bien, ils peuvent mesurer soit les distances (noté dist), soit les angles (noté angle) avec leurs voisins.

Dans ce travail, nous nous intéressons, dans le cadre de nos propositions d'amélioration, au problème de la localisation dans la configuration <S, S, z > avec z Є {Ø, dist, angle}.

2.3 Les systèmes de localisation par satellites

La navigation par satellite a commencé à partir des années 1970. Trois systèmes satellitaires ont été explorés avant la mise en place du système **GPS**. Il y a eu le système U.S. Navy Navigation Satellite System aussi connu sous le nom de **TRANSIT**, puis le système U.S. Navy's **TIMATION** (**TIM**e navig**ATION** - 1964) et enfin le projet U.S. Air Force 621B. Le programme **TRANSIT** a été le premier à utiliser les émissions continues d'ondes à partir de satellites. Le département de la défense Américain (**D**epartment **O**f **D**efense - **DOD**) a décidé d'unir les avantages des deux parties pour donner naissance au système actuel connu

sous l'acronyme **GPS** (**G**lobal **P**ositioning **S**ystem) qui est mis en place depuis 1978 [EVE 07].

3.1 Le système de navigation GPS

Le principe de système de localisation par **GPS** est la trilatération ou bien la multilatération.

La trilatération : il est possible de connaître la position d'un point de l'espace, si on a toutes les distances séparant cette position avec d'autres positions connues de l'espace. La figure 2.1 illustre ce principe [SAA 08].

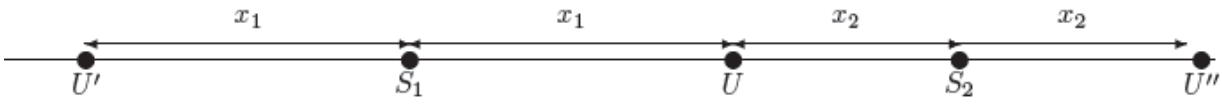


Figure 2. 1 : Principe du gps dans un espace 1-d. [EVE 07]

Si un des équipements U (satellite ou mobile, selon [EVE 07]) effectue une mesure et détermine que la distance le séparant du satellite S_1, dans ce cas, il existe deux positions possibles pour U, soit à la gauche de S_1, soit à sa droite. Afin de lever cette ambiguïté en position, il est nécessaire d'utiliser un second satellite. Ici c'est S_2 qui permet de lever cette ambiguïté. S_2 estime que la distance le séparant de l'équipement U est x_2, dans ce cas la seule position possible est celle indiquée par U sur la figure 2.1.

La multilatération : Tout comme pour la localisation dans un espace à une dimension, il est possible de procéder de la même manière pour déterminer la position d'un équipement dans un espace à deux ou trois dimensions. En deux dimensions, il s'agira de définir le point d'intersection des cercles dont les centres sont les positions des satellites et les rayons sont les distances entre le module et les satellites. En trois dimensions, les cercles sont remplacés par des sphères dont le point d'intersection correspond à la position du module.

Dans cette configuration, il est nécessaire de disposer de trois satellites à partir desquels on mesure trois distances. Le lieu géométrique correspondant au lieu se trouvant à une distance donnée d'un point fixe, est un cercle dans le cas bidimensionnel. La figure 2.2, montre que deux satellites conduisent à l'estimation de deux positions dans l'espace. Un troisième cercle est nécessaire pour déterminer la position unique [EVE 07].

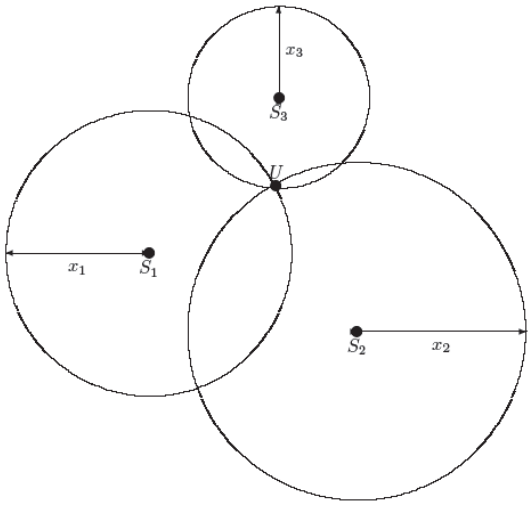


Figure 2. 2 : Principe de GPS dans un espace 2-d. [EVE 07]

3.2 Les améliorations du système GPS

Plusieurs éléments contribuent à détériorer les performances du système **GPS**, comme la traversée de l'atmosphère par les ondes, ou la traversée de certains matériaux entourant le récepteur **GPS** (comme des bâtiments par exemple). Certaines applications nécessitent que ces dégradations soient minimisées, comme dans l'aviation où la position exacte d'un avion par rapport à la piste est nécessaire, ou en agriculture lors de l'épandage de pesticides par les airs. Différentes approches sont proposées afin de minimiser l'effet des dégradations naturelles ou imputables à l'homme, [EVE 07] :

a) Le GPS assisté : Le système de **GPS** assisté (**A**ssisted **GPS** ou **A-GPS**) cherche à améliorer le rapport signal à bruit des signaux en réception.

b) Le système DGPS : Le système **DGPS** (**D**ifferential **GPS**) améliore les performances du système **GPS** en transmettant au terminal des informations sur l'écart entre les positions indiquées par les satellites et les positions réelles connues de stations de base terrestres. Le service de navigation **DGPS** complète le système mondial de localisation **GPS NAVSTAR** en fournissant des facteurs de correction pour les pseudo-distances et des informations auxiliaires. Ces informations sont diffusées à l'aide d'un réseau de stations terrestres implantées à des emplacements

stratégiques. Ce service permet d'offrir une précision horizontale de **10** mètres (ou mieux) 95% du temps dans toutes les zones couvertes.

c) **Le système GPS par mesure de différence de phase ou aussi Real Time Kinematic (RTK)** : Dans les techniques précédentes, c'est le déphasage de code qui est utilisé (techniques de corrélation de code). Une autre manière d'aborder le problème d'estimation de la distance satellites/récepteur, est de mesurer le déphasage des ondes électromagnétiques.

d) **Le système Wide Area Augmentation System (WAAS)** : Ce système a été mis au point aux États-Unis par l'Administration Fédérale de l'Aviation (**FAA F**ederal **A**viation **A**dministration) pour corriger les erreurs du système **GPS** en temps réel et arriver à une meilleure précision (**3** m). Un réseau d'une trentaine de stations de bases terrestres sert de référence.

3.3 Le système GLONASS

Le système de localisation **GPS** n'est pas le seul à être en activité actuellement. Les militaires Russes ont mis en place un système similaire pour rivaliser avec les États-Unis durant les années 1980. De plus, les militaires Américains ont dégradé volontairement les signaux **GPS** jusqu'en 2000 afin que les autres utilisateurs ne disposent pas de la précision la plus importante disponible avec ce système. Le système de navigation Russe est le système **GLONASS** (voir figure 2.3) [EVE 07].

FIGURE 2. 3 : le systeme GLONASS

3.4 Le système GALILEO

Actuellement, l'Union Européenne met en place **Galileo** qui permettra de concurrencer les deux autres systèmes, et qui sera la propriété intégrale de l'Union Européenne et de l'**ESA**.

Il est composé de **27** satellites et atteindra une précision inférieure au mètre pour les applications du domaine civil [EVE 07].

3.5 Le système IRNSS

De son côté, l'Inde met en œuvre son système **IRNSS** (**I**ndian **R**egional **N**avigational **S**atellite **S**ystem). Il offrira une précision au sol inférieur à 20 mètres. **IRNSS** comprendra sept satellites. 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. Un tel arrangement signifie que les satellites auront une visibilité continue avec les stations de contrôle indiennes.

2.4 Le concept d'ancre

Le but de la localisation est de définir les coordonnées des capteurs. Ces coordonnés peuvent être globales, lorsqu'elles sont alignées par un système extérieur comme par exemple le système **GPS**, comme elles peuvent être relatives, signifiant qu'elles sont « une transformation rigide » (rotation, réflexion, traduction) à partir du système de coordination global.

Les nœuds ancres sont une chose nécessaire pour localiser un réseau dans un système de coordination globale. Les nœuds ancres sont des nœuds de capteur ordinaires qui savent leurs coordonnées globales a priori. Cette connaissance a pu être connue par construction, ou acquise par un certain matériel additionnel comme un récepteur **GPS**.

Le placement des ancres peut avoir un impact significatif sur la localisation. Beaucoup de groupes de recherche ont constaté que l'exactitude de localisation s'améliore si des ancres sont placées dans une coque convexe autour du réseau. Localiser les ancres additionnelles au centre du réseau est également utile. En plus, il y a des preuves considérables que de vraies améliorations de la localisation peuvent être obtenues en prévoyant la disposition d'ancre dans le réseau.

L'avantage d'utiliser des ancres est évident : la présence de plusieurs nœuds pré-localisés peut considérablement simplifier la tâche d'assignation des coordonnées aux nœuds ordinaires. Cependant, les nœuds ancres ont les inconvénients inhérents. Tels que, la cherté des récepteurs GPS, ils ne peuvent pas typiquement être employés à l'intérieur, et peuvent également être confondus par des édifices hauts ou d'autres obstacles environnementaux. Les

récepteurs **GPS** consomment significativement l'alimentation par batterie, qui peut être un problème pour des nœuds importants du réseau. Alternativement au **GPS**, est la programmation des nœuds avec leurs endroits, qui peuvent être délicats (ex, en déployant 10.000 nœuds avec 500 ancres) ou même impossibles (ex, quand les nœuds sont jetés d'un avion). Finalement, les ancres sont nécessaires pour la localisation, mais leur utilisation ne vient pas sans coût.

2.5 Les Technologies de mesure

Plusieurs technologies permettent à un capteur de mesurer la distance qui le sépare d'un capteur voisin, et donc, le capteur peut déterminer sa position en se basant sur une des quatre techniques de base les plus utilisées, nommées Temps d'arrivée (**ToA)**, Différence des temps d'arrivées (**TDoA)**, Puissance du signal reçu (**RSSI**) ou bien de mesurer l'angle qu'il forme avec celui-ci par la technique angle d'arrivée (**AoA**).

5.1 Temps d'arrivée (ToA)

La technologie **ToA** (**T**ime of **A**rrival) suppose que les nœuds du réseau sont synchrones. La distance relative 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.

Principe :

Soit le capteur A qui cherche sa position. Cette technique implique trois nœuds, soit B_1, B_2 et B_3. Chaque nœud B_i communique sa position en envoyant un message à A, et le temps de propagation de signal est mesuré. Le temps multiplié par la vitesse de propagation des signaux (la vitesse de la lumière) rapporte une distance d_i. La distance d (A, B_i) définit un cercle autour du nœud. La position de A est sur la circonférence de ce cercle. Dans un modèle bidimensionnel, la position de A est clairement déterminée comme l'intersection de ces trois cercles [SAA 08].

Cette technologie est celle utilisée par le système **GPS**. 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, et il nécessite une horloge synchronisée entre les nœuds [SAA 08].

5.2 Différence des temps d'arrivée (TDoA)

La technologie **TDoA** (**T**ime **D**ifference of **A**rrival) 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 :

- **Cas $_1$** : un émetteur envoie des signaux de natures différentes (par exemple, l'ultrason, l'onde radio, ...) à un récepteur ;
- **Cas $_2$**: un récepteur reçoit des signaux d'une même nature d'au moins trois émetteurs ;
- **Cas $_3$** : un émetteur envoie un signal reçu par au moins trois récepteurs (dans ce dernier cas une vue globale des signaux sera connue) [SAA 08].

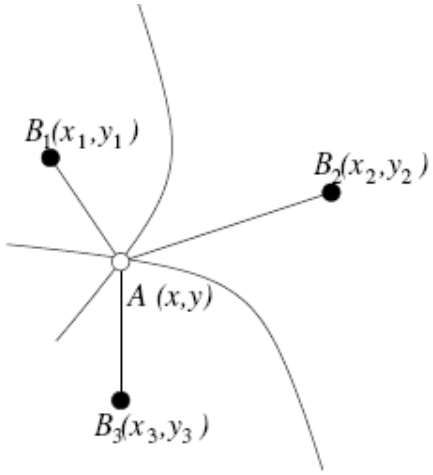


Figure 2. 4 : La technique TDoA.

Dans le cadre du deuxième cas (selon la figure 2.4), le nœud A reçoit des signaux d'une même nature des nœuds B_1, B_2 et B_3, qui connaissent leur position. Les nœuds B_1, B_2 envoient leur position simultanément. Les temps d'arrivée t_1 et t_2 des signaux de B_1 et B_2 sont mesurés par le nœud A. Le nœud A a la possibilité de récupérer l'identité de l'émetteur d'un signal et sa position. La différence de temps d''arrivée est calculée ($\delta_t = t_2 - t_1$). La différence de temps δ_t multipliée par la vitesse de la lumière, donne la différence de distance δ_d. La position (x_1, y_1) de B_1, la position (x_2, y_2) de B_2 et δ_d définissent une affectation avec l'équation :

$$\sqrt{(x-x_1)^2+(y-y_1)^2}-\sqrt{(x-x_2)^2+(y-y_2)^2}=\delta_d$$

La position de B_1 et de B_2 sont aux centres de l'affectation, et la position de A est la solution de cette équation. Deux hyperboles peuvent être définies en impliquant deux paires de nœuds (B_1, B_2) et (B_2, B_3), qui produisent deux différences de temps d'arrivée δ_1 et δ_2. Dans un modèle à deux dimensions, l'observateur de δ_1 et δ_2, le nœud A, sa position correspond à l'intersection entre les deux propositions. Il faut noter qu'il y a des cas ou les deux hyperboles s'intersectent en deux points. Dans ces cas, une troisième mesure indépendante est exigée pour résoudre l'ambiguïté.

TDoA est un mécanisme qui utilise un matériel spécifique, où chaque nœud est équipé d'un haut-parleur et d'un microphone. Quelques systèmes emploient l'ultrason tandis que d'autres emploient des fréquences audibles. Cependant, la technique mathématique générale est indépendante d'un matériel particulier.

5.3 Puissance du signal reçu (RSSI)

La puissance d'émission et de réception d'un signal peut être également exploitée pour obtenir la distance entre deux capteurs. La technologie **RSSI** (**R**eceived **S**ignal **S**trength **I**ndicator) [BAH 00] considère la perte de puissance d'un signal reçu 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 [SAA 08]. Théoriquement, l'énergie de signale diminue avec l'augmentation de la distance de la source. A partir de la, le nœud peut calculer sa distance par rapport à l'émetteur de signale, en faisant attention à la force du signal reçu.

5.4 Radio Hop Count

Quoique **RSSI** soit trop imprécis pour beaucoup d'applications, la radio peut encore être employée pour aider la localisation. L'observation principale est que si deux nœuds peuvent communiquer par la radio, leur distance entre eux est moins que R, avec une forte probabilité, où R est la gamme maximum de leurs radios, quelque soit leur lecture de la force du signal est. Ainsi, les données simples de connectivité peuvent être utiles pour la localisation [SAA 08].

L'information locale de connectivité fournie par la radio définit un graphique non pondéré, où les sommets sont des capteurs et les bords représentent des liaisons hertziennes directes entre les nœuds. h_{ij} le « hop count » (ou bien, le nombre de sauts) entre les nœuds capteurs S_i et le S_j, est alors défini comme la longueur du plus court chemin (Shortest-Path) dans le graphe entre le S_i et le S_j. Simplement, si le « hop count » entre le S_i et le S_j est h_{ij}, alors la distance entre le S_i et le S_j, le d_{ij}, est moins que le $R*h_{ij}$, où R est encore la portée radio maximum.

Il s'avère qu'une meilleure évaluation peut être faite si nous savons n_{local}, le nombre prévu de voisins par nœud. Puis, comme montré par **Kleinrock** et **Silvester**, il est possible de calculer une meilleure formule pour la distance couverte par un hop radio, selon cette formule :

$$d_{hop} = R\left(1 + e^{-n_{local}} - \int_{-1}^{1} e^{-(n_{local}/\pi)\arccos t - t\sqrt{1-t^2}}\, dt\right) \qquad (1)$$

5.5 Angle d'Arrivée (AoA)

Certain algorithmes dépendent de l'angle d'arrivée des données (**AoA**). Ces données sont typiquement recueillies par des rangées de radio ou de microphone, qui permettent à un nœud en écoute de déterminer la direction d'un nœud de transmission. Il est également possible de recueillir des données d'**AoA** par des méthodes de transmission optique.

Dans les méthodes, (**Radio Hop Count**, **TDoA**) plusieurs microphones séparés dans l'espace entendent un simple signal transmis. En analysant la différence de phase ou de temps entre l'arrivée de signal aux différents microphones, il est possible de découvrir l'**AoA** du signal.

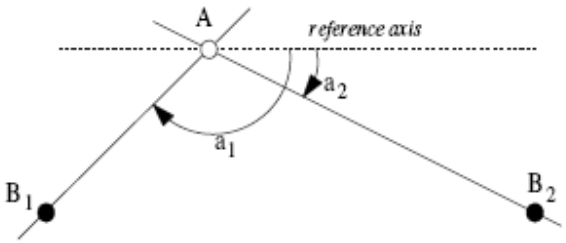


Figure 2. 5 : La technique AoA.

Principe :

Les nœuds B_1et B_2 connaissent leur position (sur la figure 2.5), contrairement au nœud A. Les nœuds B_1 et B_2 doivent pouvoir déterminer la direction d'où vient un signal. Ceci peut être réalisé avec une antenne réseau. Une ligne imaginaire est tracée de B_1 à A et une autre ligne imaginaire est tracée de B_2 à A. L'angle d'arrivée est défini comme l'angle que chacune de ces lignes fait avec une ligne orientée sur une référence commune. L'intersection de ces deux lignes détermine clairement la position de A. Cependant, si A, B_1, et B_2 se trouve tous sur la même ligne droite, une autre mesure indépendante est exigée pour résoudre l'ambiguïté.

2.6 Comparaison entre les technologies de mesures

Le tableau 2.1 résume une comparaison entre les technologies de mesures décrites précédemment :

Méthode	Précision	Distance Maximale	Matériel supplémentaire	Inconvénients
RSSI	mètres (2-4m)	Porté de communication	Aucun	Sensible aux interférences
ToA	Centimètres (2-3cm)	Porté de communication	Aucun	Synchronisation des nœuds
TDoA	Centimètres (2-3cm)	Porté de communication et celle de l'antenne sonore	Emetteur ultrasons	Ajout d'un matériel supplémentaire
AoA	Quelques degrés (5°)	Porté de communication	Boussole	Consommation énergétique plus importante.

Tableau 2.1 : Comparaison entre les technologies de mesure. [BEN 10]

Il est à mentionner que les résultats obtenus par ces méthodes varient selon des critères environnementaux : les obstacles, la vitesse du vent, l'humidité. Le choix d'une méthode doit, aussi, dépondre des besoins de l'application et les ressources disponibles [BEN 10].

2.7 Stratégies de localisation

L'utilisation du **GPS** permet d'éviter le problème de la localisation en extérieur. Cependant son coût, à la fois économique et énergétique, pose un problème. Pour répondre au besoin de la fonction de la localisation, et en maitrisant le rapport de coût, des algorithmes de localisation sont donnés dans la littérature. Parmi ces algorithmes, nous pouvons distinguer deux catégories principales, [HEU 08]:

7.1 Localisations fines

La localisation fine propose une approche basée sur les mesures. Pour laquelle, un matériel spécifique est nécessaire pour fournir par exemple la distance ou les mesures angulaires entre les nœuds voisins. Mais on trouve que :

- La mesure de la puissance du signal reçue (**RSSI**) : Cette mesure est irréaliste car le signal radio est perturbé par l'environnement et varie grandement au cours du temps,
- **ToA** (Time of Arrival) : ce mécanisme nécessite une horloge synchronisée entre les nœuds,
- **TDoA** (Time difference of Arrival) : deux signaux de différentes natures sont utilisés (ultra-sons et radio par exemple),
- **AoA** (Angle of Arrival) : Cette méthode permet de déterminer la direction de propagation d'une onde radio reçue par un ensemble d'antennes,
- Combinaison de **TDoA** et d'**AoA** pour améliorer la précision et étendre des environnements 2D à ceux 3D pour la méthode de **Capkun**.

Tous ces protocoles ne prennent pas en compte la consommation énergétique et le coût des nœuds avec comme hypothèses des capteurs suréquipés. En outre, il y a des techniques qui contournent le problème, on proposant d'avoir des nœuds dans le réseau qui connaissent préalablement leurs positions (les ancres), et donc les autres nœuds du réseau se basent sur ces derniers pour pouvoir calculer leurs propres positions. Par ailleurs, le système des ancres n'évite pas le problème de la localisation mais ne fait que le réduire à un sous-ensemble des nœuds du réseau. De plus il fait apparaître d'autres problèmes comme le placement optimal de ces ancres dans le réseau, pour permettre une meilleure localisation des nœuds.

7.2 Localisations approximatives

D'autres stratégies dites libres de mesures, quant à elles, sont des algorithmes qui peuvent fonctionner sans la présence d'une technologie de calcul des distances. La

connectivité d'un nœud est utilisée comme indicateur pour estimer des coordonnées approximatives du nœud. Selon l'application, une localisation approximative des capteurs peut être un bon compromis et dans ce cas plusieurs approches sont possibles :

- **Le système des badges actifs** : chaque nœud est marqué et transmet un paquet périodique toutes les 10 secondes. Le signal est reçu par des capteurs placés à des endroits précis et fixes à l'intérieur d'un bâtiment par exemple, et qui vont relayer l'information jusqu'au puits.
- **L'algorithme d'estimation de position** : calcule des probabilités de distribution des positions possibles des nœuds. En fonction de la position précédente des nœuds et des observations provenant d'ancres déployées dans le réseau, l'algorithme est capable de filtrer les positions impossibles.
- **Le système de coordonnées virtuelles** : chaque nœud détermine sa distance aux ancres en nombre de sauts et détermine ainsi ses propres coordonnées dans un système dont la dimension dépend du nombre d'ancres.

7.3 Conclusion

La précision des mesures réalisées par les différentes techniques de localisations à base de mesure varie selon plusieurs paramètres liés à l'environnement du réseau : obstacles, degré d'humidité, vitesse du vent, ... Etc. Mais si certaines méthodes de localisation interdisent d'utiliser ces technologies à cause de leurs erreurs de mesure, il est indispensable, pour celles qui les utilisent, de considérer ces erreurs lors du calcul des positions des capteurs [SAA 08]. Les systèmes de localisations approximatives montrées avant, sont peu adaptés aux réseaux de capteurs sans fil, car ils requièrent soit des ancres fixées à une architecture fixe soit un calcul centralisé.

2.8 Facteurs et contraintes pour la localisation dans les réseaux de capteurs

Il existe plusieurs paramètres qui influencent sur la conception et la réalisation des protocoles pour les réseaux de capteur. Parmi lesquels nous citons la tolérance aux pannes, la scalabilité, le coût de production, l'environnement d'exploitation, la topologie du réseau, les contraintes matérielles, le support de transmission et la consommation d'énergie. Ces facteurs importants servent comme directives pour le développent des algorithmes et protocoles utilisés dans les réseaux de capteurs, ils sont considérés également comme métriques de comparaison de performances entre les différents travaux dans le domaine [KHE 04].

En plus des contraintes énergétiques et calculatoires auxquelles la localisation dans les réseaux de capteur fait face, on doit aussi mentionner les contraintes liées aux paramètres réseau et celles liées aux paramètres du canal.

8.1 Les paramètres réseau

Lorsqu'on parle de paramètres réseau, on fait allusion à la taille (nombre de nœuds), à la topologie et à la connectivite.

La connectivité du réseau dépend principalement de la densité. Celle-ci est souvent définie comme étant le nombre de nœuds présents dans un mètre carré. Un réseau avec une grande densité manifeste de meilleures performances comparées à un réseau à faible densité. En outre, il y a de grandes chances, dans ce dernier cas, pour qu'un ensemble de nœuds se retrouve isolés du reste du réseau. Cependant, dans un réseau dense, l'erreur peut facilement s'accumuler et se propager d'un saut à l'autre; influant ainsi sur les performances de l'algorithme de localisation, et la propagation des erreurs, peut considérablement dégrader la précision de l'algorithme. Il est à noter que la topologie et les relations géométriques entre les nœuds ont aussi une répercussion directe sur la performance de n'importe quel algorithme de localisation.

8.2 Les paramètres du canal

La seconde catégorie de paramètres abordés dans cette section est celle des paramètres du canal. Une collaboration effective entre les nœuds dépend de la technologie sans fil utilisée et de son comportement dans l'environnement de déploiement. Par exemple, un réseau déployé dans un environnement extérieur sera confronté à des problèmes différents de ceux rencontrés dans le cas d'une localisation dans un environnement intérieur. Dans un environnement intérieur, les nœuds devront faire face à des effets de multi path beaucoup plus importants. Il est crucial d'examiner le comportement du canal sans fil pour pouvoir développer un bon algorithme collaboratif. Les algorithmes de localisation doivent être en mesure d'évaluer la qualité des estimations des mesures et d'intégrer cette information dans le processus de localisation, comme pour le sujet de la réduction des erreurs lorsque on est devant un réseau caractérisé par l'absence de visibilité directe (**NLOS**) [MAO 07].

2.9 Placement des nœuds dans le réseau

Une problématique liée à la localisation est celle du placement des nœuds dans un réseau de capteur (selon les composants du réseau, et la figure 2.6 donne un exemple des composants), c'est-à-dire comment positionner les nœuds les uns par rapport aux autres. Les types de placement existants sont :

– **Le placement uniforme :** cette configuration n'est pas réaliste, par exemple en cas de largage des capteurs, par avion,

– **Le placement dense :** pose des problèmes de coût et d'interférences entre les signaux,

– **Le placement incrémental :** il s'agit de ne pas faire de calcul global, seulement une estimation de la place optimale pour chaque nouveau capteur intégré au réseau. Ce mécanisme est bon pour l'adaptation après l'installation,

– **L'auto-déploiement** : son objectif est de maximiser l'aire couverte par le réseau, et assurer la prise en compte des obstacles. Le placement par auto-déploiement doit pouvoir être réalisé de manière autonome par les nœuds.

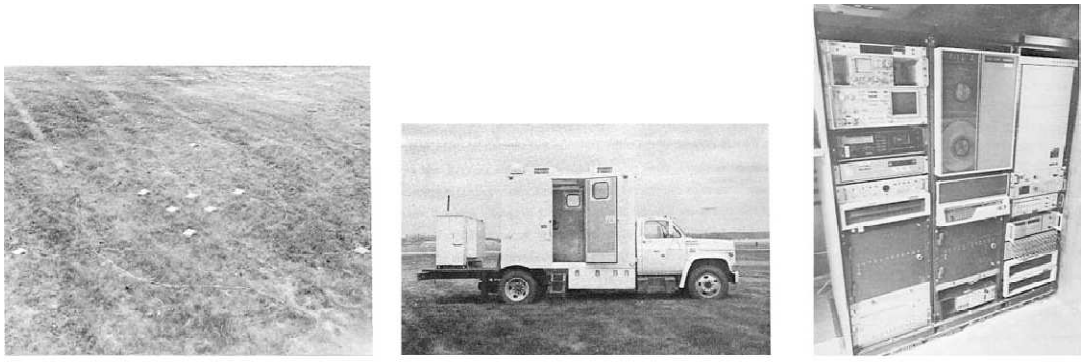


Figure 2. 6 : Les composants d'un WSN.

2.10 Gestion de la diffusion

La diffusion (ou broadcast) est l'opération de transmettre un message à l'ensemble des nœuds du réseau. Les problèmes de cette opération (broadcast) dans un réseau sans fil ont fait l'objet de nombreux travaux avec, comme premier but, la réduction du nombre de réémissions lors de la diffusion d'un message à l'intégralité du réseau. Des différentes méthodes sont utilisées, reposant sur une grande variété d'approches : probabilistes, travaux provenant de la

théorie des graphes, ou exploitant une caractéristique particulière (comme la qualité des liens par exemple) [CAR 03]. Une stratégie de diffusion cherche à maitriser et à améliorer les caractéristiques suivantes :

- **Fiabilité :**

Une diffusion est dite fiable lorsqu'elle permet de transmettre l'information à l'ensemble des nœuds joignables.

- **Déterminisme :**

Ce paramètre caractérise le comportement du nœud lors de sa décision de retransmettre à son voisinage le message de diffusion qu'il vient de recevoir. Ainsi, un algorithme de diffusion peut être déterministe ou probabiliste. Dans le premier cas, le nœud prend la décision en fonction de son état et des informations reçues dans le paquet. Le modèle probabiliste, quant à lui, décide d'une probabilité de retransmission (éventuellement à partir de certaines caractéristiques du réseau, et de l'état du nœud).

- **Information sur le réseau :**

Chaque nœud déduit, en général, son choix de retransmission à partir d'informations sur la topologie du réseau. Ces indications peuvent être contenues dans le message de diffusion et/ou mémorisées lors des communications précédentes. On peut classer en plusieurs catégories la quantité d'information nécessaire pour chaque algorithme de diffusion : global, partielle ou bien à N-sauts.

Permettre de joindre l'ensemble des nœuds du réseau, et minimiser le nombre d'émissions nécessaires pour une telle opération, sont les deux objectifs pour un algorithme de diffusion. Ces deux paramètres sont dépendants, et une bonne approche doit être capable de trouver le bon équilibre entre les deux, en fonction de l'environnement et de l'objectif à atteindre.

L'algorithme le plus trivial pour diffuser un message est le protocole d'inondation (blind flooding). Son fonctionnement est simple : chaque nœud qui reçoit le message d'inondation pour la première fois le réémet pour ses voisins, le mécanisme ne nécessitant qu'une table pour mémoriser les identifiants des messages d'inondation. Bien qu'il ne nécessite qu'une table pour mémoriser les identifiants des messages d'inondation, l'algorithme d'inondation est quand même un algorithme coûteux. Malgré sa simplicité, il impose une

charge énorme au réseau. Ce problème, dit de la tempête de messages de diffusion (Broadcast Storm Problem), a été traité en profondeur par **Ni** et autres en 1999 [CAR 03].

2.11 Conclusion

Dans cette partie de travail, nous avons présenté la problématique de ce sujet d'étude qui est la localisation, avec les concepts en relation. Dans ce cadre, nous avons vu quelques exemples de systèmes de localisation par satellites, où le **GPS** est le système le plus précis et le plus utilisé actuellement.

Les technologies de mesures représentent les différentes approches utilisées par les algorithmes de localisation afin d'estimer les positions des nœuds. Ces technologies permettent de déduire les distances ou les angles formés par les nœuds. Ils sont particulièrement utiles dans les approches de localisation géométriques. Par contre, un algorithme qui ne dépend pas d'une technologie de mesures va utiliser que les relations de voisinages pour déduire ses coordonnées.

Dans le prochain chapitre, nous allons traiter en détail les différents algorithmes de localisation.

CHAPITRE 3
TECHNIQUES ET METHODES DE LOCALISATION DANS LES WSNs

3.1 Introduction

Dans les réseaux de capteur, la localisation des capteurs est un des principaux problèmes dans ce type de réseaux et nombreuses sont les solutions qui ont été proposées pour le résoudre.

En focalisant l'intérêt de travail sur la localisation pour les réseaux de capteurs sans fil (**WSN**: **W**ireless **S**ensor **N**etwork), certains services de localisation sont disponibles, tel que le système **GPS** (**G**lobal **P**ositioning **S**ystem). Lorsque l'objet à localiser se trouve dans un environnement à ciel ouvert, le Système **GPS** est utilisé. Donc, le système **GPS** résout le problème de la localisation dans les environnements extérieurs. Cependant, pour les larges réseaux de capteurs où les nœuds doivent être de très petite taille, peu consommateurs et peu onéreux, équiper chaque unité avec une puce **GPS** est trop coûteux, alors que les capteurs peuvent être statiques et qu'ils ne vont se localiser qu'une seule fois.

Les deux modules optionnels, qui composent l'architecture d'un capteur présentée précédemment, s'avèrent très utiles pour résoudre le problème de la localisation. Car de plus en plus de services nécessitent des informations de localisation pour satisfaire les besoins des utilisateurs. En effet, ils apportent des informations pertinentes comme la position exacte de certains nœuds pour le module de localisation ou bien des données relatives aux distances ou aux angles pour le module de mesure. Mais chacune des méthodes de localisation fait appel à des diverses hypothèses sur les capacités des capteurs. Car, de nos jours, les techniques de localisation sont multiples.

Dans cette partie de travail, nous allons proposer une synthèse sur les différentes méthodes et techniques de localisation, selon une classification structurés. Donc, on commence par une illustration de la structure de notre classification, avant de donner une vues détaillé sur les méthodes et les techniques de localisation les plus utilisés selon la classification proposée.

3.2 Catégories des méthodes d'estimation de la position

Les méthodes d'estimation de la position permettent de calculer la position d'un nœud à partir des coordonnées de ses voisins. Elles sont au nombre de quatre [BOU 07]:

2.1 Méthode du barycentre simple

La méthode barycentrique est la méthode la plus utilisée dans les algorithmes de localisation. Il s'agit de calculer le centre de gravité, ou encore centre d'inertie ou centre de masse, d'un ensemble de points. Dans un algorithme par exemple, ces points sont les voisins.

$$(x, y, z) = \left(\frac{1}{n}\sum_{i=1}^{n} x_i, \frac{1}{n}\sum_{i=1}^{n} y_i, \frac{1}{n}\sum_{i=1}^{n} z_i \right)$$

2.2 Méthode du barycentre pondéré

Cette méthode repose sur les mêmes propriétés que la précédente à la différence prés qu'à chaque point est attribué un coefficient de pondération. Dans un algorithme par exemple, la pondération correspond au coefficient de précision des positions de chaque voisin.

$$(x, y, z) = \left(\frac{\sum_{i=1}^{n} \alpha_i \cdot x_i}{\sum_{i=1}^{n} \alpha_i}, \frac{\sum_{i=1}^{n} \alpha_i \cdot y_i}{\sum_{i=1}^{n} \alpha_i}, \frac{\sum_{i=1}^{n} \alpha_i \cdot z_i}{\sum_{i=1}^{n} \alpha_i} \right)$$

2.3 Méthode du centre d'Euler

En géométrie, le cercle d'Euler d'un triangle est le cercle passant par chacun des milieux des trois côtés du triangle, par chacun des pieds des trois hauteurs et par chacun des milieux des trois segments reliant l'orthocentre à un sommet du triangle. De nombreux points remarquables du triangle sont situés sur ce cercle et ce dernier est communément appelé cercle des neuf points ou cercle de Feuerbach. Le centre d'Euler est le centre du cercle d'Euler et il a la propriété d'être le centre du segment formé par le centre du cercle circonscrit au triangle et l'orthocentre. L'algorithme estime la position du nœud considéré en calculant les coordonnées du centre d'Euler avec trois voisins. Si le nombre de voisins est supérieur à trois, l'algorithme calcule le barycentre des centres d'Euler de tous les triangles possibles.

2.4 Méthode des moindres carrés

La méthode des moindres carrés est également beaucoup utilisée par les algorithmes de localisation. Généralement, elle sert à comparer des données expérimentales, donc entachées d'erreurs, avec un modèle et à trouver un modèle qui s'approche au mieux des données expérimentales. Dans un algorithme, il s'agit de calculer les coordonnées d'un point à partir de la position antérieure de ce point et des variations

des positions de ses voisins. Ainsi, la nouvelle position est en fonction des positions des voisins à l'itération (i-1) et de leurs positions à l'itération i.

3.3 Paramètres de classification des algorithmes de localisation

Les hypothèses faites par les méthodes de localisation sont nombreuses et variées. Tel que, il y a des solutions de natures différentes nécessitent une infrastructure préinstallée tandis que d'autres méthodes font intervenir des robots pour aider à la localisation. Pour pouvoir donner une étude sur les différentes techniques et protocoles de localisation dans les **WSNs**, nous avons proposé cette classification en prenant en compte les différents points qui intervient lors de paramétrage de réseau de capteur sans fil.

3.1 La mobilité de nœud

Le problème de la localisation diffère selon les hypothèses faites concernant la mobilité des capteurs. Selon le faite que les nœuds dans le réseau, ont la possibilité de déplacer ou bien sont statiques, donc on a deux grandes types de réseaux que nous allons étudie séparément par la suite.

A. Les réseaux de capteur sans fil statiques

Les réseaux de capteurs sans fil statiques, c'est les réseaux ou tous les composants construisant le réseau sont statiques et ne déplace pas, selon la configuration (< S,S, {Ø, dist, angle} >).

B. Les réseaux de capteur sans fil mobiles

La mobilité des capteurs peut prendre les trois formes suivantes :

– les capteurs et les ancres sont mobiles (< M,M, {Ø, dist, angle} >) ;

– les capteurs sont statiques et les ancres sont mobiles (< S,M, {Ø, dist, angle} >) ;

– les capteurs sont mobiles et les ancres sont statiques (< M, S, {Ø, dist, angle} >).

Dans la littérature, la configuration la plus étudiée est celle où les capteurs sont statiques et les ancres sont mobiles (< S,M, {Ø, dist, angle} >). Ces ancres sont souvent des robots ou des humains qui utilisent une stratégie de déplacement dans le réseau afin de contribuer à la localisation des autres nœuds. Les autres configurations liées à la mobilité, notamment celles considérant des capteurs indépendants entre eux (sans stratégie de déplacement des ancres pour aider à la localisation), n'ont jamais été étudiées.

La mobilité dans les réseaux de capteurs ajoute un problème supplémentaire : étant données les limites de calculs et de communications dues à la contrainte énergétique, les capteurs ne peuvent se permettre de se localiser de façon continue. Il s'agit donc de définir les dates auxquelles un capteur invoquera le calcul de sa position afin de maintenir un

niveau de précision des positions satisfaisant tout en économisant de l'énergie. Et pour ce faire il faut suivre une stratégie d'ordonnancement. Dans la littérature, on trouve des auteurs qui proposent trois stratégies d'ordonnancement des dates (**SFR - S**tatic **F**ixed **R**ate, **DVM - D**ynamic **V**elocity **M**onotonic et **MADRD - M**obility **A**ware **D**ead **R**eckoning **D**riven) et se focalisent uniquement sur cette question.

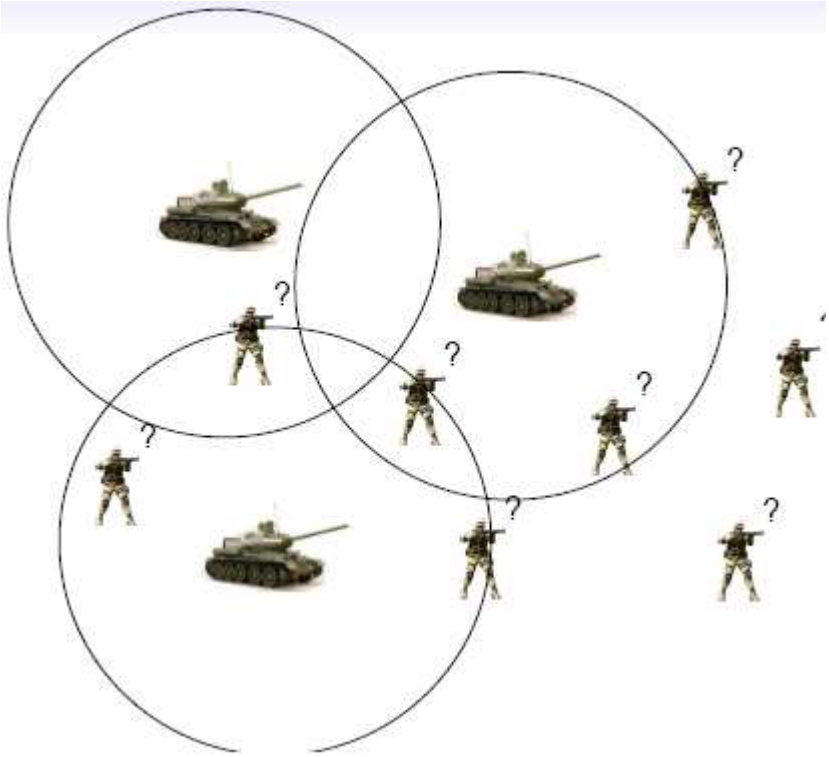


Figure 3.1 : Exemple d'un réseau de capteur mobile dans le domaine militaire.

3.2 Architecture de déploiement de réseau de capteurs

En plus de paramètre de mobilité des nœuds, les algorithmes de localisation se diffèrent selon l'architecture de travail, soit l'algorithme est implémenté sur une architecture centralisé ou bien distribué.

A. Approche centralisée

Les algorithmes qui suivent une approche centralisés, utilisent une simple unité de traitement centrale pour rassembler toutes les différentes données nécessaires au calcul réalisé par l'algorithme de localisation. Donc, un schéma centralisé pour le problème de localisation est commandée. Le raison principal derrière l'intérêt des schémas centralisés de localisation est la probabilité de fournir des évaluations plus précises de position que ceux fournis par des algorithmes distribués [MAO 07].

B. Approche distribuée

Les algorithmes distribués se fondent sur l'auto-localisation de chaque nœud dans le réseau en utilisant les mesures réalisés par les nœuds et les informations locales qu'il collecte de ses voisins [MAO 07].

3.3 Technologies de mesure

Dans ce rapport, nous avons étudie les différents technologies de mesure, qui rentrent dans le cadre de la fonctionnalité de localisation. Mais on trouve des algorithmes de localisation qui ne se basent pas sur ces techniques de mesures, et à partir de la nous distinguons deux catégories de méthodes:

A. Les méthodes libres mesure

Dans la famille de méthode libre mesure, les capteurs cherchant à déterminer leurs positions s'appuient uniquement sur les informations de voisinage, et les positions des ancres s'ils existent à travers le réseau. Donc, aucune mesure de distance ou d'angle n'est utilisée. Par conséquent, ces méthodes ne peuvent fournir que des positions estimées aux capteurs. Il existe deux techniques courantes dans ce type de méthodes [HU 04]:

a. Technique local :

La première consiste à définir des zones contenant les capteurs dont les centres de gravité correspondent à leurs positions estimées. En présence d'un pourcentage faible d'ancres dans le réseau, cette solution est intéressante. En revanche, si l'augmentation de ce pourcentage améliore la précision des positions, elle engendre dans le même temps des calculs importants pour les capteurs. Cette catégorie des algorithmes est appelée aussi algorithme de localisation basé connectivité, car ils exploitent l'information de connectivité, C.-à-d, 'qui est dans la marge de communications de laquelle' pour estimer la position de capteur.

b. Techniques basé saut :

Dans la seconde technique, chaque capteur estime les distances qui le séparent des ancres et applique la multilatération pour calculer sa position estimée. Ce schéma est également adopté par des méthodes basées mesures.

B. Les méthodes basés mesures

Les méthodes basées mesures utilisent les technologies de mesure afin de mesurer les distances ou les angles entre deux capteurs voisins. Grâce à cette capacité de mesure, un capteur pourra, sous certaines conditions, obtenir sa position exacte. Autrement, une position estimée lui sera attribuée. Les méthodes basées mesures sont les plus répandues.

3.4 Algorithmes de localisation dans les reseaux de capteurs statiques

4.1 Les algorithmes centralisés libre mesure

A. L'algorithme de Doherty

Dans l'algorithme centralisé de **Doherty** [DOH 01], le problème de localisation basé sur la connectivité est formulé comme problème d'optimisation convexe et résolu en utilisant des algorithmes existants pour résoudre des programmes linéaires et des algorithmes de programmation semi-définis. Les programmes semi-définis sont une généralisation des programmes linéaires et font réduire au minimum la forme.

4.2 Les algorithmes centralisés basés mesures

Dans la littérature, il existe trois approches principales pour concevoir des algorithmes centralisés de localisation basés distance :

1. graduation multidimensionnelle (**MDS**) :
2. programmation linéaire,
3. approches d'optimisation stochastique.

A. Auto-localisation par régression de matrices de Gram (approche centralisé)

On exploite une technique de régression matricielle reposant sur le formalisme des espaces de **Hilbert** à noyau reproduisant et impliquant deux matrices de **Gram**, l'une partiellement connue correspondant aux positions relatives des capteurs, l'autre regroupant les mesures de portée inter-capteurs. Dans cette approche, ils sont proposés deux méthodes de résolution du problème, l'une en mode centralisée et l'autre distribuée [HON 09].

Principe Centralisé

On note x_i le vecteur colonne des coordonnées du $i^{ème}$ capteur, avec i ∈ {1, 2, . . . , m} lorsqu'il s'agit d'une ancre. Soit P la matrice de Gram $[x_1\ x_2 \cdots x_N]^T\ [x_1\ x_2 \cdots x_N]$ constituée à partir des coordonnées des capteurs. Celle-ci peut être décomposée en quatre sous-matrices P_{aa},P_{ac},P_{ca} et P_{cc} selon qu'elles impliquent les ancres (a) et/ou capteurs (c). Étant donnée la sous-matrice P_{aa}, connue, on souhaite déterminer les autres sous-matrices. On dispose pour cela des mesures de RSSI pour chaque couple de capteurs, définissant ainsi la matrice K. On décompose de même cette dernière en quatre sous-matrices K_{aa},K_{ac},K_{ca} et K_{cc}. A partir des deux sous-matrices P_{aa} et K_{aa} associées aux couples ancre – ancre, on cherche à déterminer une application associant les deux matrices P et K.

$$\left[\begin{array}{c|c} K_{aa} & K_{ac} \\ \hline K_{ca} & K_{cc} \end{array}\right] \longrightarrow \left[\begin{array}{c|c} P_{aa} & P_{ac} \\ \hline P_{ca} & P_{cc} \end{array}\right]$$

Le caractère non-paramétrique de l'approche proposée, lui confère une flexibilité particulièrement appréciable dans le cadre des réseaux de capteurs, comme illustré par des expérimentations [HON 09].

B. Multi-Dimensional Scaling- Mobile Application Part (MDS-MAP)

La méthode **M**ulti-**D**imensional **S**caling **(MDS)** tient ses origines de la psychométrie. Elle a été proposée dans le but de comprendre comment réagit une personne face aux similarités des membres d'un groupe d'entités. C'est une technique d'analyse de données qui est maintenant utilisée dans différents domaines. Elle a été appliquée avec succès dans le marketing, la sociologie, la physique et la biologie. **MDS** est utilisée pour produire une image plus facile à comprendre qu'une matrice des similarités.

Shang et autres ont développé l'algorithme centralisé **M**ulti-**D**imensional **S**caling - **M**obile **A**pplication **P**art (**MDS-MAP)**, en employant la graduation multidimensionnelle (**MDS)**. Plusieurs travaux existants proposent des algorithmes centralisés, comme sur [SHA 03], en utilisant la graduation multidimensionnelle (**MDS**) [DAT 06]. **MDS** est un ensemble de techniques d'analyse de données qui montre la structure des données de distance comme image géométrique.

Principe de MDS :

Les chemins les plus courts (c.-à-d., le nombre de sauts) entre toutes les paires de nœuds sont d'abord calculés, puis sont employés pour construire une matrice de distance de **MDS**. Par la suite, la **MDS** est appliqué à la matrice de distance et une valeur approximative des coordonnées relatives de chaque nœud est obtenue. Finalement, les coordonnées relatives sont transformées en coordonnées absolues en alignant les coordonnées relatives prévues des ancres avec leurs coordonnées absolues. En effet, la méthode **MDS-MAP** a trois étapes [SHA 03]:

Commençant par l'information de connectivité fournie dans le réseau, nous employons d'abord un algorithme de court-chemins de toutes paires pour estimer rudement la distance entre chaque paire possible de nœuds.

Ensuite, nous employons la graduation multidimensionnelle **MDS**, une technique de la psychologie mathématique, pour dériver les positions des nœuds qui ont adapté ces distances prévues.

Finalement, nous normalisons les coordonnées en résultant pour tenir compte de tous les nœuds dont les positions sont connues.

Raffinement : Les évaluations de positions obtenues à travers les autres étapes, peuvent être raffinées en utilisant une minimisation des moindres carrés.

Shang a amélioré leur algorithme en divisant le réseau de capteurs entier en définissant des régions locales. La procédure de **MDS** est appliquée individuellement dans différentes régions. Alors ces cartes locales sont raccordées ensemble pour former une carte globale en employant des nœuds communs partagés entre les régions limitrophes. L'algorithme amélioré peut réaliser une meilleure exécution sur les réseaux de forme irrégulière en évitant l'utilisation d'information de distance entre les nœuds lointains. L'algorithme amélioré peut également être mis en application dans un mode distribuée [MAO 07].

Via les simulations a été démontré que cette technique simple, dépasse souvent des méthodes existantes, et elle exige seulement des informations de connectivité pour produire un résultat significatif. Si les distances entre les nœuds voisins peuvent être estimées, cette information peut facilement être incorporée par paires au calcul de Shortest-Path pendant la première étape de l'algorithme.

4.3 Les algorithmes distribués libre mesure

A. Algorithme de localisation qualitative (QLoP : Qualitative Location Protocol)

Le but de cet algorithme est de permettre à chaque nœud de déterminer grossièrement la position de ses voisins en utilisant uniquement des informations locales.

L'algorithme déployé sur un nœud obtient ces informations par l'échange de la table de voisinage. Ces tables vont lui servir pour calculer un indice de proximité pour chaque voisin à un saut. La position qualitative d'un voisin peut être *très proche*, *proche* ou *loin***.** Ainsi, cette position imprécise peut être utilisée pour construire un protocole de routage unicast efficace dans un environnement sans fil avec un haut niveau

d'interférences : choisir les voisins considérés comme *très proches* permet de choisir les nœuds avec un rapport signal sur bruit important. Des applications au contrôle de topologie et aux coordonnées virtuelles pour le routage sont également possibles [HEU 08].

Principe :

Le nœud *A* calcule l'indice de proximité d'un voisin *B* de la façon suivante :

$$PIA(B) = (|V(a) \cap V(b)|) - (max(|V(a)|, |V(b)|)) / 2 \qquad (3)$$

avec $V(a)$ le voisinage de A.

L'idée générale du protocole de localisation **QLoP** (**Q**ualitative **L**ocation **P**rotocol [HEU 08]) est d'attribuer un indice fort aux voisins ayant à la fois un voisinage commun important et un voisinage distinct faible. L'indice de proximité va ainsi évaluer la similarité des voisinages. Ainsi, nous prenons en compte dans l'algorithme le rapport entre le nombre de voisins communs et le nombre de voisins disjoints. Ainsi l'indice de proximité va représenter la distance qualitative d'un voisin. Les auteurs de ce protocole ont constaté que cette proximité qualitative va de paire avec la proximité géographique dans le cas des réseaux de capteurs denses. Ce mécanisme permet d'établir trois classes distinctes parmi les voisins :

Soit $PI(x)$ l'indice de proximité du voisin x :

$$inter = | max(PI(xi)) - min(PI(xi))| \, 3 \qquad (4)$$

$$classe_x = \begin{cases} 1 \; si \; PI(x) \geq max(PI(xi)) - inter \\ 2 \; si \; max(PI(xi)) - inter > PI(x) \geq max(PI(xi)) - 2.inter \\ 3 \; si \; PI(x) < max(PI(xi)) - 2.inter \end{cases}$$

QLoP n'utilise aucune spécificité matérielle particulière ou mesures de **RSSI** mais se base uniquement sur les informations topologiques de son voisinage à 2 sauts. Ainsi cet algorithme permet de classer les voisins d'un capteur dans 3 classes de proximité : le 1-voisinage logique, le 2-voisinage logique et le 3-voisinage logique. Les expérimentations ont mis en évidence l'amélioration des performances de routage sur la topologie logique, donnée par ce protocole, par rapport au routage à plat : le taux de livraison et la distance moyenne atteignable dans ces conditions extrêmes sont améliorés de façon significative. La consommation énergétique complète ces résultats [HEV 09].

B. L'algorithme Centroid

Bulusu et autre dans [BUL 00] proposent la méthode **Centroid**, une technique basé connectivité, tel qu'elle utilise une grille des nœuds ancres (nœuds utilisés comme des points de référence) avec des positions connus, chaque nœud *unknown*[1] place sa position au centre de surface de localisation des ancres à qui il est relié [SAV 02]. Tel qu'il a défini une connectivité métrique, ce qui est le rapport du nombre de signaux reçus avec succès par rapport au nombre de signaux de cet émetteur, pour mesurer la qualité de communication pour ce paire de transmetteur-récepteur. Un récepteur qui ne connait pas sa position, emploie le centre de surface de ses points de référence en tant que son évaluation de position, là où un point de référence est un émetteur avec une position connu dont la connectivité métrique dépasse un certain seuil (90% dans [BUL 00]).

C. Le système de coordonnées virtuelles

Pour se passer de l'utilisation des ancres ou, plus précisément, pour ne dépendre d'aucun système de coordonnées globales, certaines solutions construisent un système de coordonnées « virtuelles » (différent du système terrestre). Elles ne nécessitent pas de module de localisation mais elles attribuent aux capteurs des positions (avec des coordonnées virtuelles) dans ce système. Car dans un système qui utilise les ancres, chaque nœud détermine sa distance par rapport aux ancres en nombre de sauts et détermine ainsi ses propres coordonnés dans un système dont la dimension dépend du nombre d'ancres. Ce type des méthodes n'est pas utilisé pour la localisation dans le sens géographique du terme, mais plutôt pour l'identification ou l'adressage des capteurs. Ces positions sont alors utilisées dans des domaines tels que le routage [HEU 08].

D. La méthode HTRefine

Au commencement de la méthode **HTRefine**, toutes les ancres diffusent leurs positions. Lorsqu'un capteur reçoit la position d'une ancre, il estime la distance qui le sépare d'elle. Pour ce faire, **HTRefine** utilise la technique d'estimation des distances **DV-Hop** (une approche développer par **Niculescu** et **Nath** dans [NIC 01]).

[1] Capteur *unknown* : est un capteur qui ne connaît pas sa position préalablement.

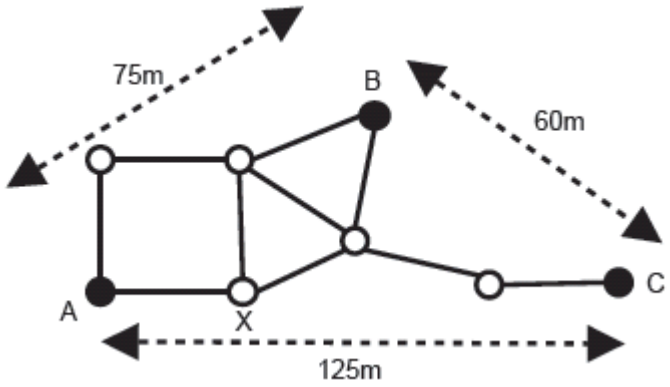


Figure 3.2. : La technique DV-HOP.

L'algorithme DV-Hop [CHE 08]

Lors de la première étape, chaque nœud ancre diffuse un message d'information à travers tout le réseau contenant la position de l'ancre avec la valeur de nombre de sauts initialisée à 1. Chaque nœud qui reçoit ce message, maintient la valeur de nombre des sauts par rapport à l'ancre depuis tous les messages qu'il a reçu. Un message qui contient un nombre de sauts plus grand par rapport à une ancre, est considéré comme information compromis et il est ignoré. Alors que les messages d'informations non éventées sont diffusés dans le réseau après incrémentation de la valeur de nombre de sauts. Par ce mécanisme, tous les nœuds dans le réseau obtiennent le nombre minimal de sauts par rapport à chaque nœud ancre.

Lors de la deuxième étape, une fois que l'ancre à reçu la valeur de nombre de sauts des autres ancres, elle estime une moyenne de la taille d'un saut, qui est alors diffusée au réseau entier. Après la réception de la taille d'un saut, les nœuds multiplient la taille d'un saut par la valeur de nombre de sauts pour estimer la distance physique par rapport à l'ancre.

La moyenne de la taille d'un saut est estimée par les ancres par la formule suivante :

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

Tel que (x_i, y_i) et (x_j, y_j) représente les coordonnées des ancres i et j, et h_{ij} est le nombre de sauts minimum qui séparent les deux ancres i et j.

Chaque ancre diffuse son HopSize calculé, via le réseau. Les nœuds *unknowns* reçoivent l'information de HopSize et sauvegardent la première information reçue, ensuite ils font transmettre cette information à leurs voisins. Cette manière de procéder assure que

le plus grand nombre de nœuds reçoivent ce message d'information et ce même pour les nœuds qui ont le plus petit nombre de saut entre eux. A la fin de cette étape, les nœuds *unknown*s calculent la distance qui les sépare des autres nœuds en se basant sur le nombre de sauts et la taille d'un saut.

Soit (x, y) les coordonnées non connues de nœud D, et (x_i, y_i) les coordonnées connues de la $i^{ème}$ ancre. On va dire que d_i est la distance entre le nœud *unknown* D et la $i^{ème}$ ancre dans le réseau. A base de ces informations, nous avons les formules suivantes :

$$\begin{cases} (x-x_1)^2 + (y-y_1)^2 = d_1^2 \\ (x-x_2)^2 + (y-y_2)^2 = d_2^2 \\ \ldots. \\ (x-x_i)^2 + (y-y_i)^2 = d_i^2 \end{cases}$$

Les coordonnées du nœud D sont calculées par les formules suivantes :

$$B = \begin{bmatrix} d_1^2 - d_n^2 - x_1^2 + x_n^2 - y_1^2 + y_n^2 \\ d_2^2 - d_n^2 - x_2^2 + x_n^2 - y_2^2 + y_n^2 \\ \vdots \\ d_{n-1}^2 - d_n^2 - x_{n-1}^2 + x_n^2 - y_{n-1}^2 + y_n^2 \end{bmatrix}$$

$$A = -2 * \begin{bmatrix} x1 - xn & y1 - yn \\ x2 - xn & y2 - yn \\ \ldots & \ldots \\ x(n-1) - xn & y(n-1) - yn \end{bmatrix}$$

$$P = \begin{bmatrix} x \\ y \end{bmatrix}$$

Tel que :

$$P = (A^T A)^{-1} A^T B.$$

E. Algorithme Amorphe de localisation

Algorithme Amorphe de localisation de **Nagpal**, utilise une approche similaire à celui de la technique **DV-Hop**, les coordonnées des ancres sont diffusés à travers le réseau, donc, chaque nœud peut garder le nombre de saut vers cette ancre. Les nœuds peuvent calculés leurs positions en basant sur les positions d'ancres reçus et le nombre de saut correspondant [HU 04].

Nagpal et autres dans [NAG 03], démontrent par algorithme qu'encore de meilleures évaluations de distance en nombre de sauts peuvent être calculées en faisant la moyenne des distances avec des voisins. Cet avantage ne commence pas à apparaître

jusqu'au n_{local} >15 ; cependant, il peut ramener l'erreur de nombre de sauts vers le bas jusqu'à moins de 0.2R. Un exemple sur l'utilisation de nombre de sauts montre un diagramme où on a une obstruction dans la topologie. C'est l'une des manières dont ce principe peut éprouver l'erreur dramatique.

F. La méthode AT-Free

AT-Family (**F**amily of **A**pproximation **T**echniques) à partir de [SAA 08], est une famille de trois techniques d'approximation de localisation dans les réseaux statiques, s'appuyant sur les capacités des capteurs. Les trois méthodes **AT-Free**, **AT-Dist** et **AT-Angle** considèrent respectivement les cas où les capteurs n'ont aucune capacité de mesure ou bien ils ont la capacité de mesurer soit les distances, soit les angles avec leurs voisins. Leur principe général reste le même mais leur fonctionnement diffère en fonction des capacités de mesure des capteurs. Ces méthodes assurent deux propriétés :

1. La première propriété assure à chaque capteur une indication sur la précision de sa position (en lui fournissant sa borne d'erreur de position) ;
2. La seconde propriété est de permettre d'éliminer de fausses informations de localisation.

Ces propriétés ont un fort impact sur la précision des positions attribuées.

AT-Free fait partie de l'ensemble des méthodes dites libres de mesure, qui attribue aux capteurs des positions estimées à l'aide des techniques d'approximation. Dans **AT-Free**, les capteurs connaissent uniquement les positions des ancres (il s'agit de la configuration < S, S, Θ >). En fonction de ces positions, chaque capteur définit une zone géographique le contenant et prend pour position le centre de gravité de celle-ci. Chaque nœud peut savoir si sa position estimée est proche de sa position réelle, et dans le cas échéant, devient une ancre estimée afin d'aider à son tour les autres nœuds à se localiser. Cette section montre également comment **AT-Free** peut être utilisée en étape préliminaire par d'autres techniques de localisation afin d'améliorer la précision de leurs résultats.

Principe :

Au départ, chaque ancre diffuse sa position. Un nœud peut donc en déduire le nombre de sauts qui le sépare de chacune des ancres (seul le plus petit nombre de sauts est retenu). A la réception de la position d'une ancre, un nœud considère les cas suivants :

a) s'il reçoit la position directement de l'ancre, il en déduit qu'ils sont voisins et donc qu'il appartient au disque centré en l'ancre de rayon r.
b) s'il reçoit la position par un nœud intermédiaire, il en déduit qu'il n'est pas voisin de l'ancre et donc qu'il n'appartient pas au disque de rayon r centré en l'ancre. En

revanche, il appartient au disque centré en l'ancre de rayon (r × h), où h représente le nombre de sauts entre le nœud et l'ancre.

La conjonction de toutes ces informations pour chacune des ancres définit une zone contenant le nœud dont la position estimée est calculée comme étant le centre de gravité de cette zone.

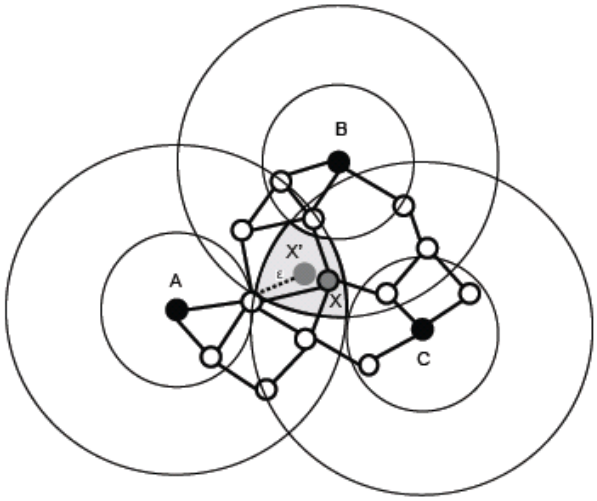


Figure 3. 3 : Approximation de la position dans AT-Free.

Lorsqu'un capteur estime sa position en fonction de sa zone (selon la figure 3.3.), il est capable de calculer une borne de son erreur de position et savoir ainsi si sa position estimée est proche de sa position réelle.

Pour augmenter la précision des positions attribuées par **AT-Free**, il faut fournir un maximum d'informations de localisation. L'idéal serait qu'un capteur diffuse sa première position estimée (avec sa borne d'erreur) et dès qu'une nouvelle position est calculée, il la rediffuse avec son nouvel **e** et ainsi de suite. Au final, chaque nœud obtiendrait la meilleure position possible avec **AT-Free**. De par la contrainte énergétique, cette solution n'est pas réaliste. Il est donc nécessaire de mettre en place une stratégie de diffusion afin de trouver le meilleur compromis entre le nombre autorisé de diffusions pour un capteur et la précision des positions. En considérant la stratégie de diffusion, on peut constater que la précision des positions attribuées par **AT-Free** dépend des choix des paramètres G, r, k et l. Par conséquent, il est intéressant de souligner qu'**AT-Free** est totalement paramétrable. Donc, le choix des valeurs de ces constantes joue un rôle sur les performances d'**AT-Free**.

Cette méthode est intéressante car elle est paramétrable et peut être utilisée comme étape préliminaire d'une autre technique. En effet, **AT-Free** permet au nœud de déterminer sa position estimée et de calculer la précision de sa position par rapport à sa position réelle avec une borne d'erreur, si celle-ci est jugée petite le nœud devient une ancre et diffuse sa

position, cette aspect est très important car la précision dépend du nombre d'ancres dans le réseau.

4.4 Les algorithmes distribués à base des mesures

A. Algorithme de localisation collaborative pour les réseaux de capteurs sans fil

C'est un algorithme de localisation distribué collaborative qui n'exige aucun appui externe d'infrastructure sous forme de nœuds d'ancre ou de balise. Cet algorithme utilise des distances mesurées inter les nœuds (en utilisant la technologie **TDoA,** entre un signal radio et un signal acoustique), et il donne la localisation pour le réseau entier dans un système de coordination local jusqu'à une traduction à une rotation et à une réflexion globales. Ce système de coordination local constitué par l'algorithme peut être aisément utilisé avec un certain nombre d'applications comme le cheminement, l'agrégation et le stockage de données distribué.

Cet algorithme se compose de trois phases distinctes :

1. Choix de référence : Pendant la première phase, tous les nœuds dans le réseau collaborent pour choisir un ensemble de nœuds qui sont référés comme nœuds de référence.
2. localisation de référence : les coordonnées de ces nœuds de référence sont estimées.
3. et localisation de nœud : les nœuds non référence déterminent leurs coordonnées.

Les coordonnées locales établies peuvent être transformées en n'importe quel autre système de coordination pour l'usage avec les applications qui exigent des coordonnées absolues.

La contribution majeure de cet approche est le faite qu'elle propose une résolution de problème de localisation pour la totalité des capteurs de réseaux, en choisissant une liste des capteurs les biens placés sur la zone de captage, en suite une estimation des coordonnées de ces nœuds est faite pour utiliser ces nœuds lors de la localisation du reste du réseau [SAR 07].

B. Auto-localisation par régression de matrices de Gram (version distribué)

Afin de répondre aux contraintes imposées par les réseaux de capteurs sans fil, une version distribuée de l'approche d'auto localisation par régression de matrices est mise en œuvre. Cette technique reposant sur le formalisme des espaces de **Hilbert** à noyau

reproduisant et impliquant deux matrices de **Gram**, l'une partiellement connue correspondant aux positions relatives des capteurs, l'autre regroupant les mesures de portée inter-capteurs .Il convient de noter que l'approche distribuée ne présente pas de difficulté majeure au regard des formes quadratiques manipulées. Dans un cadre distribué, chaque capteur communique avec ses voisins pour déterminer ses propres coordonnés, donc la solution est souvent calculée de manière efficace puisque le nombre total de voisins ancres est raisonnable dans un réseau de capteurs **[HON 09]**.

C. La méthode SumDistMinMax

Dans cette méthode, chaque ancre commence par diffuser sa position. Lorsqu'un capteur reçoit cette position, il estime la distance qui le sépare de cette ancre en appliquant la technique **SumDist** qui est la technique la plus simple pour l'estimation de la distance entre un capteur et une ancre.

Le principe de SumDist

Elle consiste à sommer les distances mesurées entre chaque paire de capteurs voisins séparant l'ancre et le capteur qui cherche à estimer sa position. Lorsqu'une ancre envoie sa position, elle joint au message la distance qui la sépare d'elle même, c'est à dire 0. A la réception de ce message, chaque capteur voisin mesure la distance avec l'émetteur, l'enregistre, ajoute cette distance à celle contenue dans le message et fait suivre la position de l'ancre avec la distance mise à jour. Le même processus est répété pour tous les capteurs. Au final, chaque capteur obtient la position de chacune des ancres et calcule une estimation de la distance qui le sépare de celle-ci [SAA 08].

Exemple :

La figure 3.4 illustre le fonctionnement de **SumDist**. Sur cette figure, la distance estimée entre les nœuds S et D est égale à $d_{SY} + d_{YD}$ sachant que de par l'inégalité triangulaire $d_{SD} <= d_{SY} + d_{YD}$.

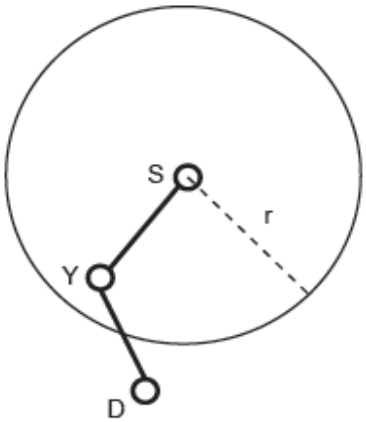


Figure 3. 4 : La technique SumDist. [SAA 08]

Soit x_1, x_2, ..., x_q, a étant un chemin d'un capteur $\mathbf{x_1}$ vers une ancre **a**, l'estimation de la distance entre x_1 et a (notée $d\hat{}_{x1a}$) est définie de façon récursive comme suit :

$$d\hat{}x_1a = dx_1x_2 + d\hat{}x_2a$$

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

Le principe de MinMax

La méthode **MinMax** détermine pour chaque capteur, une « boîte » le contenant dont le centre de gravité correspond à sa position estimée.

Exemple :

Dans la figure 3.5., le capteur X associe une « boîte » à chacune des ancres A, B, C. Chaque « boîte » est centrée en la position de l'ancre ((x_A, y_A)) et dépend de la distance estimée ($d\hat{}_{XA}$) avec le capteur X. Par exemple, la « boîte » centrée en A est construite comme suit :

$$[x_A - d\hat{}_{XA}, y_A - d\hat{}_{XA}] * [x_A + d\hat{}_{XA}, y_A + d\hat{}_{XA}]$$

L'intersection de ces « boîtes » forme une nouvelle « boîte » définie par :

$$[\max (x_i - d\hat{}_{Xi}),\max (y_i - d\hat{}_{Xi})] * [\min (x_i + d\hat{}_{Xi}),\min (y_i + d\hat{}_{Xi})] \text{ , } i \in \{A, B, C\}$$

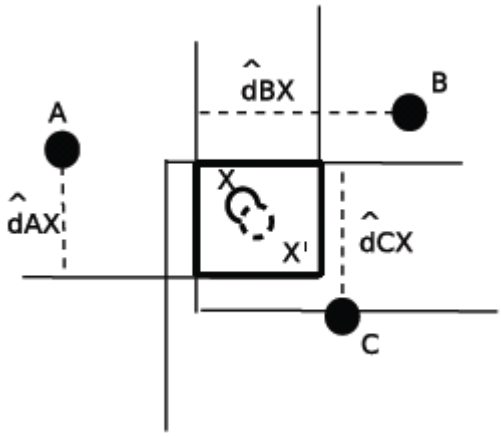


Figure 3.5 : La technique MinMax. [SAA 08]

Le capteur X estime sa position (notée X_0) comme étant le centre de gravité de cette «boîte».

Raffinement

Finalement, chaque capteur exécute un processus de raffinement où les relations de voisinage et les mesures de distances entre voisins sont prises en compte afin de faire tendre sa position estimée vers sa position réelle.

L'inconvénient

L'accumulation des erreurs de mesure notamment dues à la technique d'estimation des distances **SumDist** et au processus de raffinement.

D. La méthode Ad hoc Positioning System (APS)

La méthode **A**d Hoc **P**ositioning **S**ystem (**APS**) [NIC 01], est une méthode de localisation parmi les plus connues. Après une étape où les ancres inondent le réseau, chaque capteur déduit la distance qui le sépare des ancres. Sur l'article original de la méthode **APS**, La technique d'estimation des distances utilisée dans **APS** est nommée **Euclidean**. Et on trouve des descriptions de la méthode **APS** qui utilise la technique **DV-Hop**. Tel que cela peut être fait seulement en estimant la longueur moyenne d'un saut (**DV-Hop**) ou en utilisant des mesures de distance (**DV-Distance**).

Le principe de la technique Euclidean

La figure 3.6 illustre le principe de cette technique, où A, B, C sont des capteurs et D une ancre. A cherche à calculer sa distance avec D. B et C sont voisins de A et voisins entre eux. A et B ont calculé leurs distances avec D (d_{BD}, d_{CD}). A connaît alors les distances (d_{AB}, d_{AC}, d_{BC}, d_{BD}, d_{CD}). Par conséquent, il connaît tous les côtés et une diagonale du quadrilatère ABCD. La seconde diagonale correspond à la distance d_{AD}. Mais deux distances sont possibles (d_1 et d_2). Un processus de vote des voisins et un autre dit du voisin commun permet à A de déduire la distance qui le sépare de D. En fait, ces processus font intervenir un troisième capteur ayant lui aussi estimé sa distance avec D. Selon le cas où ce capteur est voisin de B ou C, ou bien voisin des deux, ces processus, par de simples relations géométriques, déduisent la distance d_{AD}. (La déduction de distance dans **APS** se base sur un processus de vote des voisins et un autre dit du voisin commun)

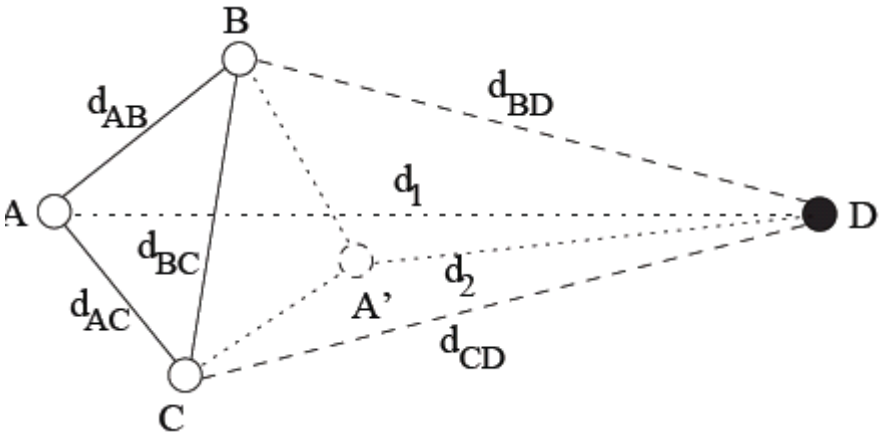


Figure 3. 6 : La technique Euclidean. [SAA 08]

Une fois les distances avec les ancres obtenues, les capteurs appliquent alors la multilatération décrite ci-avant.

Raffinement : Dans **APS**, aucune phase de raffinement n'est proposée puisque les contraintes imposées par le voisinage et par les distances sont prises en compte lors de l'estimation des distances.

La méthode **APS** calcule, sous certaines conditions, une distance exacte lorsqu'un capteur connaît deux voisins ayant estimé leurs distances avec une ancre. Les auteurs de cette méthode envisagent les topologies ad hoc qui ne changent pas souvent, comme les infrastructures provisoires de réseaux de capteur, d'intérieur ou extérieures [NIC 02]. Donc la méthode **APS** n'est pas adaptée pour les réseaux de capteurs mobiles. Cet algorithme est décentralisé, il n'a pas besoin d'infrastructure spéciale, et fournit le positionnement absolu [NIC 02].

E. La méthode Ad hoc Positioning System à base AoA (APS_{AoA})

La méthode $\mathbf{APS_{AoA}}$ est une extension de la méthode précédente qui considère des capteurs ayant, non pas la capacité de mesurer les distances avec leurs voisins, mais celle de mesurer les angles [NIC 03].

Au départ, les ancres diffusent leurs positions et les capteurs voisins cherchent à déduire les angles qu'ils forment avec chacune d'elles par rapport à leurs propres axes de référence. Ces capteurs font alors suivre leurs angles à leurs voisins pour qu'ils puissent déduire leurs angles à leurs tours et ainsi de suite.

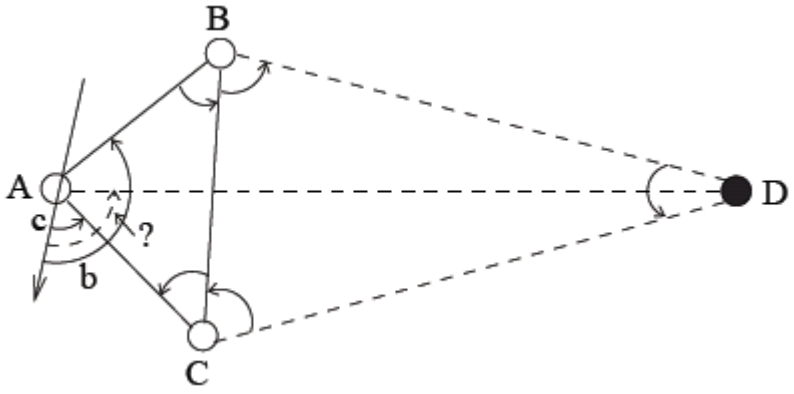


Figure 3.7 : APS AVEC AoA. [NIC 03]

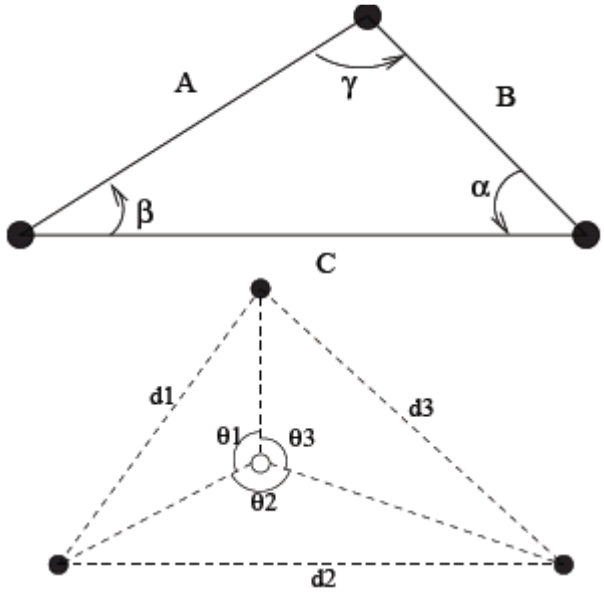
Le principe :

Le capteur A cherche à déduire son angle avec l'ancre D par rapport à son axe de référence. Les capteurs B et C, voisins de A et voisins entre eux, connaissent les angles respectifs qu'ils forment avec D. Ainsi les angles des triangles ABC et BCD sont connus et

l'angle DAC peut être calculé. L'angle par rapport à l'axe de A sera donc égal à c + l'angle DAC (voir figure 3.7).

Dès lors qu'un capteur a obtenu les angles qu'il forme avec au moins trois ancres, comme illustré par la figure 3.8., il obtient sa position grâce à la triangulation qui donne le système suivant :

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



a) la triangulation, b) calcul de la position.

Figure 3. 8 : Les principes de calcul dans APS avec AoA. [NIC 03]

F. L'algorithme Bounding Box

L'algorithme **Bounding-Box** [KWO 04] est un algorithme de localisation qui, en utilisant les distances qui séparent un nœud *unknown* de plusieurs ancres, déduit avec un simple calcul mathématique la position du nœud *unknown*. Une boite (Bounding-Box) peut être imaginée comme un rectangle dans lequel une ancre est située en son centre et dont la largeur est égale à deux fois la distance qui sépare l'ancre du nœud qui veut se localiser. L'intersection des différentes boites, construites à partir des informations en provenance de plusieurs ancres, donnera une zone dans laquelle le nœud *unknown* est sûr d'y appartenir. Le nœud considérera le centre de gravité de cette zone comme estimation de sa position.

La boite centrée en une ancre A est construite comme suit :

$$[X_A - D_A, Y_A - D_A] \; [X_A + D_A, Y_A + D_A]$$

Où X_A et Y_A sont les coordonnées de A, et D_A la distance qui sépare A du nœud *unknown*.

L'intersection des Bounding-Box est obtenue par la formule suivante :

$$[min(Xi - Di), min(Yi - Di)] \times [min(Xi + Di), min(Yi + Di)] \quad i = 1 \cdots n$$

Où Xi et Yi sont les coordonnées de l'ancre i, Di la distance qui sépare cette ancre du nœud cible et n le nombre d'ancres audibles.

Cette méthode, de par sa simplicité, est idéale dans le cas ou les capteurs sont très limités en puissance de calcul ; l'utilisation d'algorithmes plus complexes se révélerait tout simplement impossible. Une approche qui donne des résultats plus précis devrait, cependant, être envisagée si les ressources du capteur le permettent.

G. La méthode Gradient

Cette méthode utilise la multilatération pour déterminer la position des nœuds. Pour obtenir les distances qui séparent chaque nœud de chaque ancre, un gradient est propagé à partir de ces ancres dans le but de trouver les plus courts chemins (en termes de distance) menant à partir de n'importe quel nœud A à n'importe quelle ancre B audible. Le gradient se propage suivant l'algorithme présenté ci-dessous [STO 05]:

1- Pour chaque nœud J et pour chaque ancre K faire :
   - Si J = K alors D_{JK} (distance qui sépare J de K) = 0
   - Sinon $D_{JK} = \infty$
2- Pour chaque nœud J, répéter :
   - Pour chaque ancre K et voisin I, extraire la distance D_{IK}
   - Pour chaque ancre K et voisin I, appliquer la formule suivante :

$$D_{JK} = \min (D_{JK} + D_{IJ}, D_{JK})$$

Après un certain temps la valeur D_{JK} sera égale à la taille du plus court chemin qui sépare J de K. Il ne reste plus qu'à appliquer la multilatération en utilisant les données obtenues. L'utilisation des gradients afin de calculer les distances nécessaires à la multilatération a été proposée par de nombreux chercheurs dans le domaine. Ces algorithmes assument tous, qu'il y a au moins trois ancres audibles quelque part dans le réseau. L'accumulation des erreurs de mesure, notamment dues à la technique d'estimation des distances, influera en mal sur la précision.

H. Algorithme Point In Triangle Test

L'algorithme **APIT** (**A**lgorithme **P**oint **I**n **T**riangle Test) proposé par [STO 00], dite qu'il est libre de mesure et donne une estimation des positions en subdivisant la zone de déploiement en régions triangulaires formées à partir des ancres. Les auteurs de [DAT 06]

décomposent le réseau en utilisant non pas des triangles mais des polygones. Nous exposons ici le principe de la première approche. Et malgré que l'algorithme **APIT** ne base pas sur un matériel de mesure, mais le fait qu'il utilise l'information de force de signal radio, les auteurs de [DAT 06] ont considéré cet algorithme comme un algorithme à base de mesures.

Principe :

Soit n le nombre d'ancres dans le réseau. Suivant le principe montré par la figure 3.9., chaque ancre commence par diffuser sa position. Les nœuds *unknown*s effectuent un **PIT** (**P**oint **I**n **T**riangle Test) avec chaque combinaison d'ancres audibles (les ancres dont les positions leurs ont été parvenues). La technique du **PIT** permet de tester si un nœud est à l'intérieur d'un triangle formé par trois ancres. **APIT** répète le procédé pour toutes les combinaisons des ancres audibles pour former C^n_3 triangles possibles. Après avoir déterminé son appartenance ou son non appartenance à chacun des triangles, le nœud déduit une zone d'appartenance et prend comme position le centre de cette zone. L'algorithme est décrit par le pseudo code suivant [HE 05]:

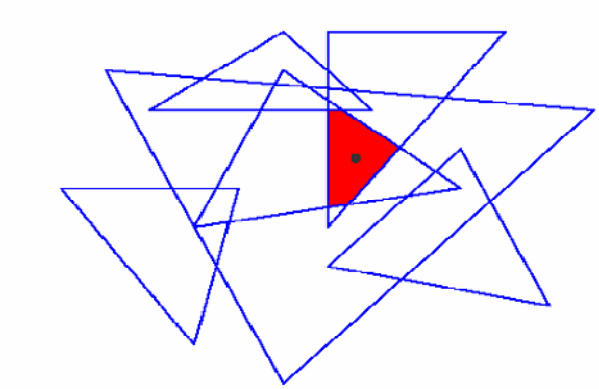


Figure 3.9 : Localisation par l'algorithme APIT. [DAT 06]

I. Multi-latération collaborative

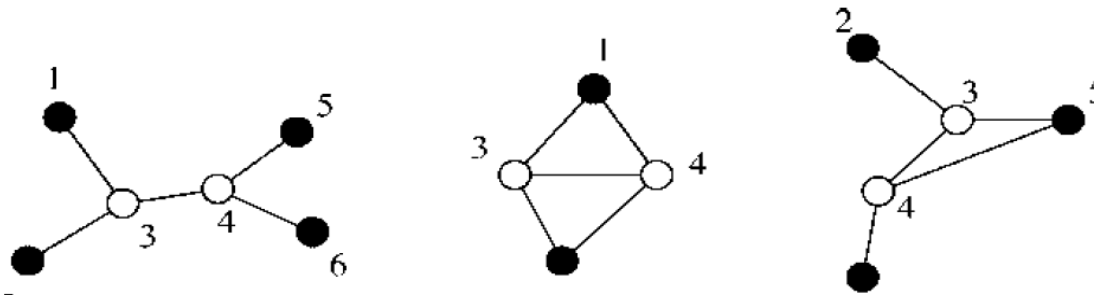
Les auteurs de [SAV 03] proposent un algorithme de localisation multi-sauts basé sur la notion de groupe collaboratif. L'algorithme se déroule en trois phases :

Dans la première phase, les nœuds vont s'auto organiser en groupe. Un nœud va se localiser en utilisant comme références les ancres qui appartiennent au même groupe. Les groupes sont créés de telle sorte que les nœuds dont les positions sont inconnues peuvent avoir une solution unique. Les nœuds qui ne satisfont pas les contraintes imposées pour la création de groupe ne feront partie d'aucun groupe.

Durant la seconde phase, les nœuds utilisent les distances qui les séparent des ancres pour obtenir une estimation initiale de leurs coordonnées.

Finalement une phase de raffinement est exécutée pour obtenir les coordonnées finales.

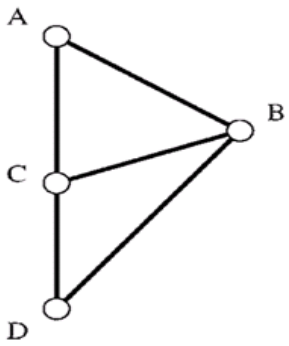
Pour qu'un nœud puisse avoir une et une seule solution dans un environnement 2D, il faut qu'il soit connecté avec au moins trois références non alignées sur la même droite. Ces références peuvent être des ancres ou bien des nœuds *unknown*s ayant estimé leurs positions ; les positions estimées de ces derniers doivent être uniques. Dans la fig. 3.10 (a), les nœuds 3 et 4 sont des nœuds *unknown*s tous deux connectés à trois nœuds. Notons que du point de vue du nœud 3, un de ces liens se termine par un nœud *unknown*, le nœud 4. Si on assume que le nœud 3 à une solution unique alors le nœud 4 aura aussi une solution unique. La condition présentées ici est nécessaire mais pas suffisante pour garantir qu'il y ait une et une seule solution pour un nœud donné.



(a) (b) (c)

Figure 3.10 : Les principes de calcul pour le protocole du multi lateration collaborative. [SAV 03]

Si les références qu'utilise un nœud sont alignées sur la même droite, le nœud *unknown* aura deux solutions possibles. Un nœud peut tester si trois autres nœuds sont alignés sur la même droite en utilisant la trigonométrie. Dans la figure 3.11, en assumant que les nœuds A, C et D ont une solution unique, le nœud B va essayer de déduire si sa position, en utilisant ces trois points de référence, est unique aussi. Le nœud B va d'abord calculer les angles ABC, CBD et ABD ; en utilisant l'angle ABD, le nœud B peut calculer la distance AD. Si la distance AD est égale à la somme des deux distances AC et CD alors les trois références sont colinéaires.



$$\angle CBD = \cos^{-1}\left(\frac{|CD|^2 - |BC|^2 - |BD|^2}{2 \times |BC| \times |BD|}\right)$$

$$\angle ABC = \cos^{-1}\left(\frac{|AC|^2 - |AB|^2 - |BC|^2}{2 \times |AB| \times |AC|}\right)$$

Figure 3.11 : Détection d'une configuration colinéaire pour le protocole du multi lateration collaborative. [SAV 03]

Un autre arrangement qui peut causer des problèmes de ce genre est illustré dans la figure 3.10(b). Les nœuds 3 et 4 ont tous les deux trois liens ; les positions des deux nœuds peuvent être échangées sans violer les contraintes imposées par les mesures de distances. Pour éviter ce genre de situation une autre contrainte est établie :

> Dans chaque paire de nœuds, dont le lien qui les connecte est utilisé comme contrainte par ces deux nœuds, il est nécessaire d'avoir au moins un lien qui connecte un de ces deux nœuds à une référence non utilisée par l'autre nœud.

La figure 3.10 (c) satisfait cette propriété. Les nœuds 3 et 4 ont tous les deux au moins une référence indépendante. Le nœud 4 est lié à l'ancre 1 et le nœud 3 est lié à l'ancre 2.

Les conditions précédentes sont suffisantes pour garantir qu'un nœud *unknown* à une solution unique.

La figure.3.12 montre comment les distances qui séparent un nœud C de deux ancres A et B sont utilisées pour obtenir les coordonnées de C. Si la distance entre C et A est da alors la valeur de la coordonnée xc est incluse dans l'ensemble [xa – da, xa+ da]. Dans le cas de l'ancre B, qui est à deux sauts plus loin de C, la longueur du plus court chemin est utilisée pour borner la valeur de xc; dans ce cas la valeur de xc est bornée par [xb-(b+c), xb+ (b+c)]. Le nœud obtient de cette manière une boite le contenant ; il prendra le centre de cette boite comme position.

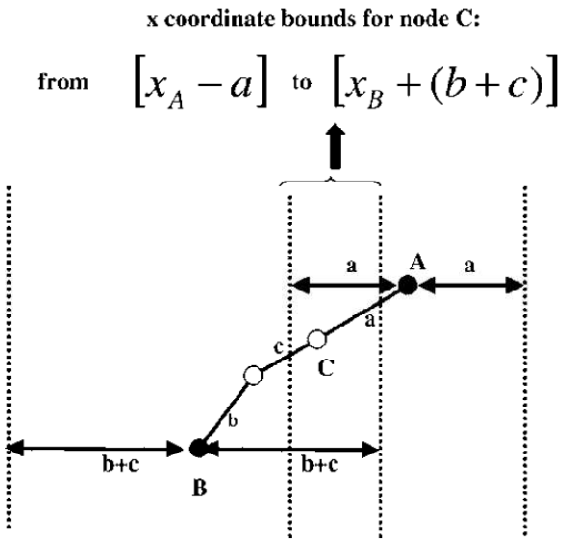


Figure 3.12 : Borner les coordonnées d'un nœud pour le protocole du multi lateration collaborative. [SAV 03]

Raffinement : Une phase de raffinement en diffusant les positions estimée sera par la suite utilisée pour obtenir les résultats finaux.

J. Multi-Dimensional Scaling (MDS) distribué

Une version distribuée de **MDS-MAP** est proposée dans [SHA 04]. Dans cet algorithme plusieurs cartes sont générées en parallèles. L'idée est de construire une carte locale pour chaque nœud en utilisant le principe de la **MDS** classique. Les cartes seront fusionnées pour former une carte globale. Dans des environnements irréguliers, cette technique montre de bien meilleures performances que celles de la version classique. Il est donc préférable de l'utiliser dans des réseaux dans lesquels la taille du plus court chemin qui sépare deux nœuds ne correspond pas à la distance euclidienne entre ces deux nœuds. Les tests effectués par les auteurs de cet algorithme montrent que les positions obtenues en utilisant les technologies de mesure des distances sont beaucoup plus précises que dans le cas contraire. En comparant les algorithmes basés **MDS** et les algorithmes basés vecteurs de distances (**AT-Dist**), les auteurs ont conclu que les algorithmes basés **MDS** donnent le plus souvent de meilleurs résultats.

Dans [YIS 04], une autre variation de la **MDS-MAP** est proposée. Un chemin de communication entre deux nœuds situés loin l'un de l'autre est établi. Ce chemin peut être découvert par des algorithmes de routage. Seuls les nœuds qui se trouvent sur ce chemin vont calculer leurs cartes. Ces cartes seront alors alignées et fusionnées. Les tests ont montré que la précision de l'algorithme dépend de la connectivité du réseau, des erreurs de mesures de distance, de la longueur du chemin choisi et du nombre des voisins en commun

entre deux nœuds adjacents tout au long du chemin. Avec une connectivité supérieure à 10 ; l'algorithme donne de très bons résultats.

Dans la variante proposée par [XIA 04]. L'algorithme commence par définir une carte locale à partir d'un ensemble de voisins non colinéaires. Une fois cette première carte définie les positions des nœuds sont diffusée et ça sera au tour de leurs voisins de construire leurs propres cartes. La carte va donc grandir en taille jusqu'à couvrir pratiquement l'ensemble du réseau. La simulation a montré que cette technique est deux fois plus précise qu'**APS** dans des topologies en forme de C.

K. La méthode AT-Dist

La connaissance des distances entre nœuds voisins fournit des informations considérables pour la localisation. La méthode proposée dans cette section suppose que les capteurs ont la capacité de mesurer les distances avec leurs voisins (< S, S, dist >). Grâce à cette capacité, **AT-Dist** améliore la technique d'approximation des positions d'**AT-Free** et définit des règles permettant de localiser exactement certains nœuds [SAA 08].

Principe :

Le principe général reste le même que celui d'**AT-Free.** Le changement concerne l'estimation de la distance séparant un nœud et une ancre distante. Précédemment, cette distance était estimée comme étant le produit du rayon d'émission r et du nombre de sauts entre le nœud et l'ancre. **AT-Dist** utilise la technique d'estimation des distances **SumDist** faisant la somme des distances entre les nœuds séparant un capteur d'une ancre. Les distances ainsi estimées sont notées $\hat{d}$.

Soit X étant un capteur recevant la position d'une ancre A (selon la figure 3.13). S'ils sont voisins ($A \in N_{\Delta}(X)$), X en déduit la distance exacte notée d_{AX}. Par conséquent, X appartient au cercle centré en A de rayon d_{AX}. Par contre, si X et A ne sont pas voisins ($A \notin N_{\Delta}(X)$) alors X en déduit qu'il n'appartient pas au disque centré en A de rayon r mais appartient à celui de centre A et de rayon $\hat{d}_{AX}$. De par l'inégalité triangulaire, la distance euclidienne entre X et A est inférieure ou égale à $\hat{d}_{AX}$. X applique cette technique pour chaque position qu'il reçoit et la conjonction de ces données de localisation définit une zone géographique Z_X. Le centre de gravité de cette zone représente la position estimée de X.

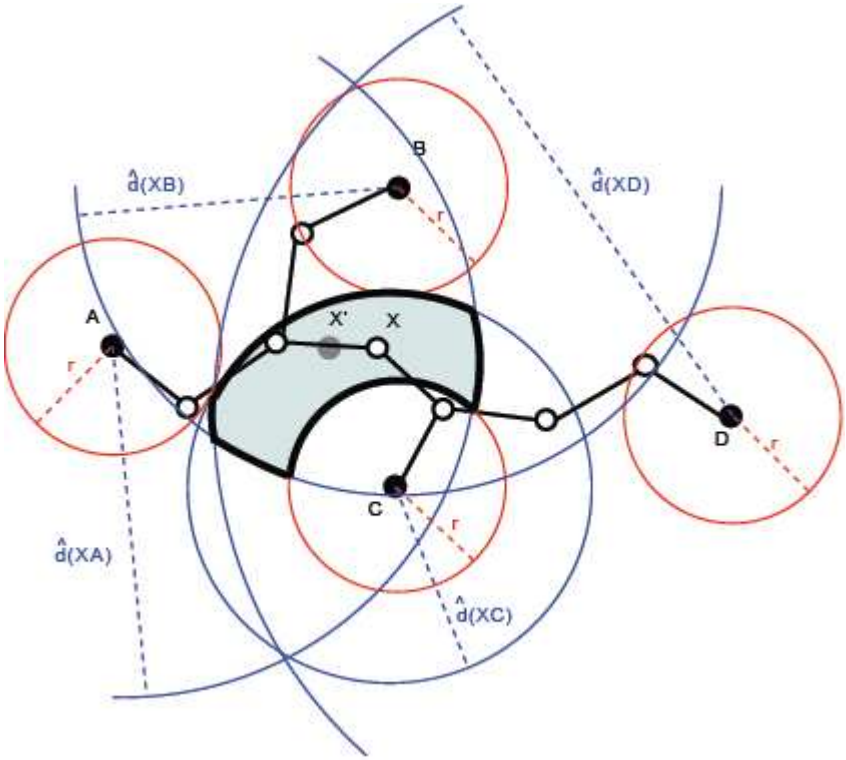


Figure 3.13 : Approximation de la position dans AT-Dist. [SAA 08]

Comme dans **AT-Free**, le capteur déduit, lors de l'estimation de sa position, sa borne d'erreur de position e. Si cette e est inférieure au seuil G, le capteur devient une ancre estimée et diffuse sa position ainsi que son e. En utilisant la connaissance des distances, des informations de localisation sont ajoutées. Cela permet d'être plus sévère sur la gestion des diffusions d'une ancre en n'en autorisant qu'une seule (G = r). Comme dans **AT-Free** toujours, la technique d'approximation est adaptée afin de prendre en compte les bornes d'erreurs de position des ancres estimées.

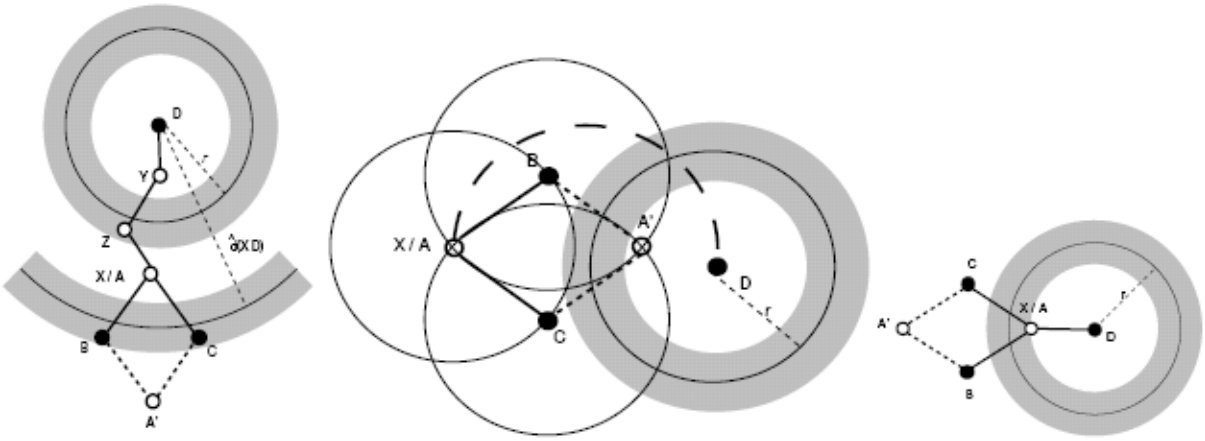
La méthode **AT-Dist** utilise trois règles de localisation qui lui permettent de localiser exactement certains capteurs. Ces règles résolvent l'ambiguïté lorsqu'un capteur peut se trouver en plusieurs positions. Ces règles constituent à elles seules une méthode de localisation puisque, selon la configuration et l'environnement du réseau, elles peuvent localiser tous les nœuds. Cette méthode, appelée **MuR (Method using Rules)**, a la particularité de pouvoir être exécutée en étape préliminaire par toutes les techniques de localisation s'appuyant sur les distances afin d'augmenter le nombre d'ancres dans le réseau.

Règles de localisation (MuR - Method using Rules)

MuR [SAA 06] a été proposé afin de résoudre l'ambigüité lorsqu'un nœud peut être localisé à plusieurs endroits: si un nœud a deux ancres voisines, les deux points d'intersection des deux cercles centrés aux ancres et de rayons les rangs qui les séparent,

représentent deux positions possibles. Nous adaptons les règles de **MuR** afin de prendre en compte les ancres estimées.

Dans la figure 3.14, le nœud X cherche sa position, B, C sont des ancres voisines. On suppose que la position réelle de X est en A. Dans les deux premières règles, l'ancre D permettant de résoudre l'ambigüité n'est pas une ancre voisine de X.



a) règle 1 **b) règle 2** **c) règle 3**

Figure 3.14 : Règles de localisation mur. [SAA 06]

Règle 1 : Cette règle est illustrée en figure 3.14 (a). Dans un premier temps D est une ancre. Si A se trouve entre les cercles centrés en D de rayons respectifs r et $\hat{d}_{XD}$ et A' se trouve a` l'extérieur alors X conclut qu'il est en A. Cependant, si les deux nœuds sont entre ces deux cercles X, on ne peut pas conclure. Maintenant si D est une ancre estimée, il faut prendre en compte e, donnant ainsi la zone en gris sur la figure définie par les rayons ± e. A devra être à l'intérieur des deux zones en gris et A' à l'extérieur.

Règle 2 : Cette règle est illustrée en figure 3.14 (b). Dans un premier temps D est une ancre. Si A' se trouve à l'intérieur du cercle centré en D et de rayon r alors A' ne peut être la position de X puisque X et D ne sont pas voisins. Si A est à l'extérieur de ce cercle alors X conclu qu'il est en A. Cependant, si A et A' sont à l'extérieur alors X ne peut pas conclure. Considérons D comme une ancre estimée, A doit être à l'extérieur de la zone grise et A' à l'intérieur.

Règle 3 : Dans cette règle X et D sont voisins et elle est de ce fait exactement le contraire de la règle 2. Elle est illustrée en figure 3.14 (c).

L. La méthode AT-Angle

La méthode précédente **AT-Dist** a montré l'utilité pour les capteurs d'avoir des informations relatives aux mesures de distances. Cette section propose une technique de

localisation nommée **AT-Angle** qui s'appuie non plus sur les distances mais sur les angles entre deux nœuds voisins (< S, S, angle >) [SAA 08].

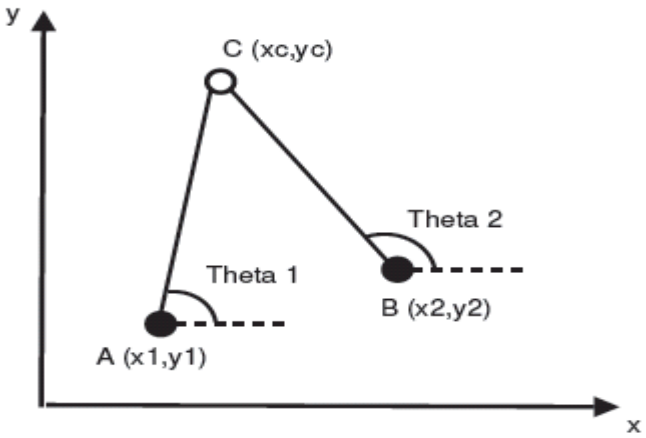


Figure 3.15 : Calcul d'un angle. [SAA 08]

Principe :

Grâce à la connaissance des angles, donnés sur la figure 3.15, un nœud peut se localiser exactement. En effet, si un capteur possède deux ancres dans son voisinage, il est capable de déduire sa position exacte. Le nœud C cherche à calculer sa position (x_c, y_c). A et B sont deux ancres voisines de C de coordonnées respectives (x_1, y_1), (x_2, y_2). C connaît les angles \A, C = Θ_1, \B, C = Θ_2. Les cordonnées (x_c, y_c) peuvent être calculées grâce au système suivant :

$$\frac{yc-yi}{xc-xi} = \tan(\Theta) \text{ Avec i € } \{1,2\}$$

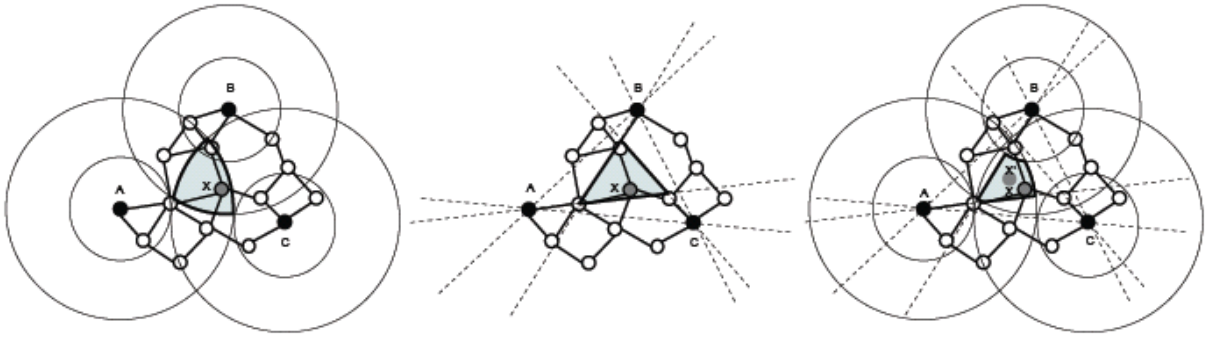
Le nœud C devient alors une ancre et diffuse sa position.

Une part du fonctionnement de cette technique est identique à la technique d'approximation d'**AT-Free** : au départ les ancres diffusent leurs positions et les nœuds récepteurs de ces messages estiment leurs distances avec ces ancres en multipliant leur rayon d'émission r par les nombres de sauts minimaux qui les séparent des ancres. Lorsqu'un nœud reçoit la position d'une ancre. L'intersection de ces disques forme une zone contenant le nœud et le centre de gravité de cette zone est considéré comme étant une estimation de sa position.

Les mesures d'angle étant très sensibles aux perturbations dues à l'environnement du réseau, la technique d'approximation doit utiliser ces données comme des indicateurs d'orientation et non comme des mesures exactes. Dans la technique d'approximation **AT-Angle**, une ancre ne diffuse pas seulement sa position mais aussi les angles qu'elle forme avec ses voisins directs (ancres ou non) ainsi qu'avec les autres ancres distantes (les angles sont alors calculés en fonction des positions des ancres). Chaque mesure d'angle est

reportée sur un des axes de référence (nord, sud, est, ouest). Par la suite, les angles seront reportés sur l'axe est. Il est fait comme hypothèse que l'ensemble des capteurs respecte cette convention.

Lorsqu'un capteur considère la position d'une ancre et, par exemple, un angle qu'elle forme avec un de ses voisins, il peut tracer une droite passant par la position de cette ancre dont l'inclinaison correspond à l'angle considéré. Ensuite, le nœud cherche à déduire son appartenance à cette droite ou au côté de la droite auquel il appartient et, le cas échéant, fait suivre : la position de l'ancre, l'angle et sa position par rapport à cette droite. Si le nœud ne déduit aucune information alors il ne diffuse rien [SAA 08].



a) Cercle **b) droites** **c) les deux**

Figure 3.16 : Les principes d'approximation. [SAA 08]

La figure 3.16 est une illustration de la technique d'approximation. X est le nœud cherchant à estimer sa position et A, B, C sont des ancres. Les figures 3.16a, 3.16b, 3.16c représentent les zones (en gris) du noeud X en utilisant respectivement uniquement les cercles, les droites et ces deux informations. Finalement, X estime sa position comme étant le centre de gravité de la zone.

M. L'algorithme ADNL-Dist

Les auteurs **Champ** et **Boudet**, de cet algorithme, intéressent particulièrement au problème de localisation multi sauts en présence d'ancres et de capacités de mesure de distances. Certains nœuds connaissent donc leur localisation géographique au moyen d'un **GPS** par exemple, ce sont les ancres, et les capteurs ont la possibilité de connaître la distance qui les sépare de leurs voisins de manière plus ou moins précise.

Nous présentons l'algorithme **ADNL** (**A**ccurate **D**istributed **N**ode **L**ocalization algorithm) dont le principe consiste à utiliser un algorithme de "forces" pour localiser les capteurs, et de propager les nouvelles informations de localisation intelligemment afin de permettre aux autres nœuds de se localiser à leur tour. Les résultats de simulation montrent l'efficacité de la méthode proposée [CHA 10A] [CHA 10B]. Nous avons une version de

cette algorithme qui se base sur des mesures d'angles d'arrivé à la place des mesures de distances, cette version est présenté sur l'article [CHA 10C].

3.5 Propriétés des familles de méthodes 'AT-Family'

Chacune des méthodes d'**AT-Family** respecte deux propriétés très importantes qui ont un fort impact sur leurs performances, [SAA 08] :

Premièrement, un nœud est capable de connaître la précision de sa position estimée et peut ainsi devenir une ancre. Dans chacune des méthodes, lorsqu'un nœud estime sa position, il déduit une borne de son erreur de position notée e. Si cette borne est jugée petite alors le nœud considère être bien localisé, devient à son tour une ancre et diffuse sa position ainsi que son e.

Ainsi, les capteurs n'utilisent pas seulement les ancres exactes (équipées **GPS** par exemple) mais également les nœuds dont les positions estimées sont proches de leurs positions exactes. Cette propriété est très importante puisque la précision des positions dépend fortement du nombre d'ancres dans le réseau.

Deuxièmement, un capteur est capable d'éliminer de fausses informations relatives à sa localisation. Grâce aux zones définies dans les méthodes d'**AT-Family**, les nœuds rejettent des informations si elles sont incohérentes. Par exemple, lorsqu'un nœud reçoit une information le localisant en dehors de sa zone géographique, il ne la prendra pas en considération. Les sources d'erreurs de localisation peuvent être dues à plusieurs phénomènes : les erreurs de mesure (de distances ou d'angles), les erreurs de position, les comportements byzantins, ... Avant de pouvoir éliminer certaines fausses informations, les nœuds doivent déduire leurs zones à partir des données de localisation (les positions des ancres et les mesures). Ces données pouvant être perturbées par des erreurs, les capteurs doivent donc s'assurer de la validité de leurs zones avant de commencer à éliminer de fausses informations. Il faudra attendre d'avoir reçu un certains nombre de données de localisation provenant des ancres avant qu'un capteur considère sa zone comme valide. Ce nombre doit garantir que la majorité des informations reçues n'a pas été perturbée par des erreurs et une fois atteint, le capteur pourra commencer à éliminer des informations. Il correspond au même seuil, dépendant de l'environnement du réseau, présenté dans le processus de vote et noté confidence. Par conséquent, lorsque le réseau est fortement (resp. faiblement) perturbé par des erreurs, sa valeur est élevée (resp. faible).

Ces deux propriétés ont un impact sur la précision des positions attribuées par les méthodes d'**AT-Family**. Elles ont une influence sur deux points essentiels qui garantissent

de bons résultats : le nombre d’ancres pour la première propriété et l’élimination de certaines fausses informations pour la deuxième.

3.6 Conclusion

En étudiant les algorithmes de localisation, le dispositif le plus attrayant des algorithmes de localisation basés sur la connectivité et donc libres mesures, est leur simplicité. Toutefois ils peuvent seulement fournir une évaluation à grain grossier de position de chaque nœud, ainsi elle signifie qu'elles sont seulement appropriées aux applications exigeant une évaluation approximative de position. Également l'erreur de localisation dépend fortement de la densité de nœud dans le réseau, du nombre d'ancres et de la topologie de réseau. Car, l'erreur de position sera plus grande dans un réseau avec une plus petite densité de nœud, peu d'ancres, ou une topologie de réseau irrégulière [MAO 07].

Par contre, Les approches basés sur les mesures représentent plus de performances par rapport à l'exactitude, mais peuvent exiger de matériel supplémentaire. D’autres méthodes basées sur les mesures s’appuient sur une modélisation des pertes en précision lors du calcul des angles ou des distances. Ce modèle est ensuite utilisé pour déterminer les positions des capteurs. Or, cette perte en précision est liée à l’environnement de la zone de déploiement du réseau et aucune modélisation n’est capable de représenter n’importe quel environnement.

En effet, trouver une méthode qui garantisse des positions précises quel que soit l’environnement du réseau semble difficile voir impossible sans un minimum de connaissances relatives à cet environnement. Chaque méthode proposée dans ce chapitre doit donc être considérée comme une méthode à part entière si l’environnement du réseau le permet, sinon elle doit être considérée comme un socle sur lequel seront ajoutées les spécificités liées à l’environnement. Donc, une bonne maîtrise et connaissance de ces diverses méthodes est nécessaire afin de dimensionner judicieusement sa propre solution de localisation. Cette solution ne doit pas être surdimensionnée, sinon elle entraîne un surcoût soit au niveau de l’infrastructure soit au niveau du terminal de localisation**.**

CHAPITRE 4
ETUDE COMPARATIVE D'ALGORITHMES DE LOCALISATION

4.1 Introduction

Les réseaux de capteurs sans fil sont une technologie significative attirant l'intérêt d'une marge considérable des chercheurs. Parmi les axes de recherches on trouve le problème de localisation. Ce dernier, a reçu une attention importante depuis le passé, car, plusieurs applications ont besoin de l'information de localisation des capteurs et donc plusieurs services de localisation ont été crées. Récemment, un nombre des systèmes de localisation ont été proposé pour les réseaux de capteurs. Donc, pour évaluer leurs performances, on est appelé a faire des comparaisons et des évaluations.

4.2 Les critères de comparaison

La méthode la plus évidente qui vient à notre esprit pour mesurer les performances d'un algorithme de localisation, est de comparer les positions estimées des nœuds avec leurs positions réelles. Les algorithmes de localisation dans les **WSNs** doivent cependant prendre en considération les contraintes auxquels les réseaux de capteurs sans fil font habituellement face (limitations en calcul et en énergie …). Un ensemble plus large de critères d'évaluation est construit de ce fait. On peut citer parmi ces critères celui du coût, de la couverture et de l'extensibilité (scalabilité) ; les utilisateurs des algorithmes de localisation peuvent ainsi choisir parmi un ensemble d'algorithmes qui répondent le mieux à leurs besoins [SIC 04]. Un algorithme peut être évalué selon deux types de critères :

- les critères binaires;
- les critères quantitatifs.

Où, un critère binaire est un algorithme qui possède ou non une certaine propriété (il peut être soit configurable ou non, ou bien nécessiter la présence d'ancres capteurs ou non). Les classifications et les critères binaires sont utilisés par les chercheurs pour réduire l'ensemble des algorithmes à évaluer ou à utiliser. Par exemple, on peut seulement prendre en considération une localisation régit par les caractéristiques suivantes : environnement distribué, aucune ancre disponible, et présence d'une technologie de calcul des distances

(**RSSI**, **TDoA**, **AoA**) ; ceci va immédiatement limiter le nombre d'algorithmes à comparer. Certains critères d'évaluations ont cependant besoin d'une quantification pour avoir un sens d'où le nom de critères quantitatifs ; ces mêmes critères sont décrits en détail ci-dessous :

1. **Extensibilité**

L'algorithme de localisation peut fonctionner aussi bien dans un réseau à faible échelle (moins d'une dizaine de nœuds) que dans un réseau qui comprend des centaines ou même des milliers de nœuds.

L'augmentation du nombre de nœuds dans un algorithme centralisé peut avoir des effets dramatiques; le nombre exorbitant de messages communiqués peut étouffer le réseau. Donc, les algorithmes distribués sont mieux adaptés au critère d'extensibilité.

2. **Précision**

Quel est le degré de précision d'un algorithme de localisation (de combien les positions estimées se rapprochent t'elles des positions réelles) ?

On peut penser que la précision de l'emplacement de chaque capteur est l'objectif primordial de n'importe quel algorithme de localisation, mais cela dépends principalement des besoins de l'application. Certaines applications auront besoin d'une très bonne estimation de ces emplacements tandis que d'autres peuvent se contenter d'une estimation plus ou moins bonne. Dans un système de pistage (tracking), les positions estimées des capteurs auront une incidence directe sur l'application.

Pour définir la granularité du système, la précision peut être normalisée par rapport à la valeur de l'espacement inter-nœuds. Par exemple, si l'espacement moyen est de 100m une erreur de 1m peut être acceptable (1%). Le même degré de précision ne peut être considéré si l'espacement inter-nœuds est très petit. Dans quelques travaux, la précision est normalisée par rapport au rayon de transmission au lieu de l'espacement inter-nœuds [SAA 08].

3. **La couverture**

Quelques algorithmes peuvent ne pas être capables de localiser des nœuds n'ayant pas suffisamment de voisins. Ce critère dépend principalement de la densité : plus la densité est grande et plus les chances d'avoir suffisamment de voisins l'est aussi. En outre, il serait utile d'examiner la possibilité d'ajouter des nœuds au réseau après que l'algorithme de localisation initiale soit terminé.

4. Le coût

Quelle est la quantité d'énergie requise pour qu'un nœud puisse compléter le processus de localisation ? Quel est le temps nécessaire pour que la solution converge ? Quel est le nombre de messages envoyés ? Le coût du matériel et du logiciel doivent également être pris en considération.

Un algorithme qui permet de minimiser plusieurs de ces contraintes mais n'offrant pas forcément une bonne précision, est le plus susceptible d'être choisi si la durée de vie du réseau est un but primordial. On peut par exemple minimiser la communication et la complexité de calcul au détriment de la précision.

5. Discussion

Si on devait décrire le parfait algorithme de localisation ce serait un algorithme offrant des résultats précis, par un moyen décentralisé, faisant appel à un minimum de communication et de calcul, permettant l'ajout de nœuds supplémentaires au réseau et nécessitant la présence d'un minimum d'ancres. La perfection n'existant pas, on doit toujours faire un compromis entre ces différents critères en privilégiant quelques uns au détriment des autres.

4.3 Comparaison entre des algorithmes à base des mesures

L'intérêt donné à la localisation pour les réseaux de capteur sans fil, est prévu de se développer avec la multiplication des applications de ces réseaux. Typiquement, les algorithmes de localisation basés sur l'**AoA** et les mesures de temps de propagation peuvent réaliser une meilleure exactitude que des algorithmes de localisation basés sur des mesures de **RSS**. Cependant, cette exactitude est réalisée aux dépens d'un coût d'équipement plus élevé. **Patwari** et autres, a donné par **Cramer-Rao** (**CRB**) (une technique statistique qui fournit une limite inférieure sur le désaccord de n'importe quel estimateur impartial, pour le problème de lateration [BRO 05]) les limites inférieures pour l'évaluation de position en utilisant des mesures de **ToA**, de **RSS** et d'**AoA,** respectivement. Cependant la limite inférieure de **Cramèr-Rao** peut être trop optimiste quand l'erreur de mesure dévie de gaussien. D'ailleurs la limite de **Cramèr-Rao** suppose que l'estimateur fondamental est un estimateur impartial. Cette prétention ne peut être satisfaite par beaucoup de techniques de localisation.

3.1 Algorithmes de localisation basé sur l'AoA

Pour les algorithmes de localisation basé sur la technologie de mesure **AoA**, en l'absence du bruit et de l'interférence, l'information de lignes de roulement de deux récepteurs ou plus, intersectés pour déterminer une position unique, en faisant une évaluation de position de l'émetteur. A la présence de bruit, plus de deux lignes de roulement non intersectés à un seul point, qui font appelle à des mesures triangulaire ou bien des méthodes bien précises, ont besoins des fois pour estimer la position de transmetteur [MAO 07], comme sur la figure 4.1.

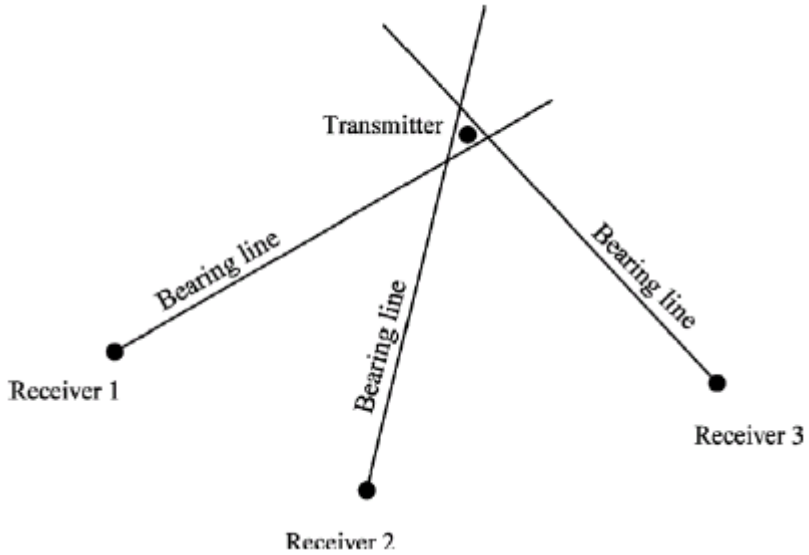


Figure 4. 1 : Le principe d'intersection des lignes de roulement. [MAO 07]

L'évaluation de position par des mesures de roulement, est un bonne sujet de recherche, et le travail pilote dans le secteur est celui de **Stanfield**. Cette approche a été généralisée et mise en application dans beaucoup de systèmes pratiques. Une autre approche bien connue est l'estimateur de maximum de vraisemblance **ML**. Noter que la solution de forme close dans l'approche de **Stanfield** dépend de deux prétentions : d'abord, l'erreur de mesure est petite, tel que : $e_i \approx \sin e_i$, $1 <= i <= N$. en second lieu, R est connu. On peut choisir d'accepter la première prétention mais de rejeter la deuxième prétention, dans ce cas un procédé itératif peut être employé pour obtenir la solution au problème de minimisation, qui n'a aucun avantage par rapport à la technique de **ML** [MAO 07].

3.2 Algorithmes de localisation basé sur le TDoA

Etant donné la mesure Δt_{ij} de **TDoA** et les coordonnées des récepteurs i et j, l'équation de **TDoA (Eq-TDoA)** définit une branche d'une hyperbole dont les intérêts sont donnés aux positions des récepteurs i et j et sur ce que l'émetteur r_t doit se trouvé. Dans un

environnement à deux dimensions (R^2), des mesures de trois (3) récepteurs au minimum sont exigées pour déterminer uniquement l'endroit de l'émetteur (comme illustrer sur la figure 4.2).

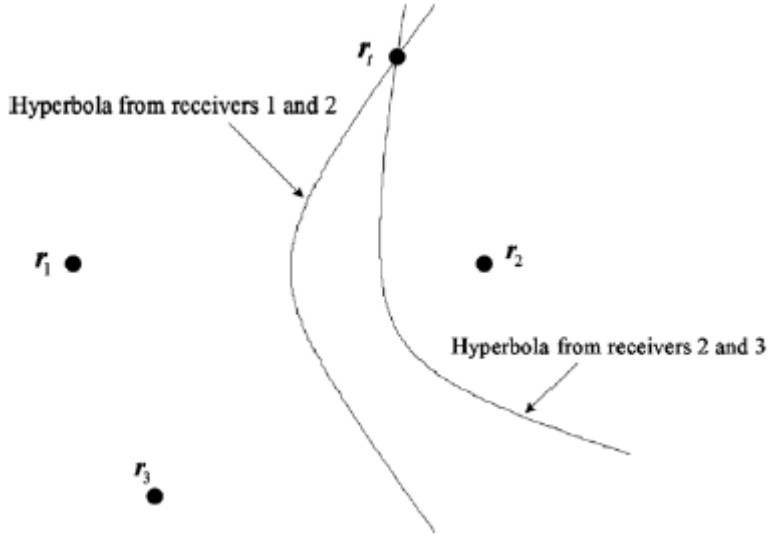


Figure 4. 2 : L'intersection de trois récepteurs. [MAO 07]

Dans un système à N récepteurs, on écrit N-1 équations de **TDoA** qui doit être résolu. Dans la pratique, Δt_{ij} n'est pas disponible ; et au lieu de cela nous avons la mesure bruyante $\Delta t_{ij}^{'}$ de **TDoA** donné par :

$$\mathbf{\Delta t_{ij}^{'} = \Delta tij + n_{ij}.}$$

Où le n_{ij} dénote un bruit additif, et habituellement, il est indépendant de zéro moyen de variable aléatoire distribuée gaussienne. Le système d'équation généré pour un système à N récepteurs est non linéaire, donc il est difficile à le résolu. Particulièrement, quand les récepteurs sont arrangés d'une mode arbitraire. D'ailleurs, en présence du bruit, il peut ne pas avoir une solution.

Une solution récursive à l'estimateur de **ML** peut alors être utilisée. Cette méthode se fonde sur une bonne supposition initiale de position de l'émetteur. D'ailleurs, en quelques situations cette méthode peut avoir comme conséquence des erreurs significatives d'évaluation de position dues à la dilution géométrique de la précision (**GDOP** : **G**eometric **D**ilution **O**f **P**recision).

GDOP décrit une situation en laquelle une erreur de rangement relativement petite peut avoir comme conséquence une grande erreur d'évaluation d'endroit parce que l'émetteur est situé sur une partie de l'hyperbole lointaine des deux récepteurs. **Fang** dans leur travaux, a donné une solution exacte aux équations hyperboliques quand le nombre de

mesures de **TDoA** sont égal au nombre de coordonnées inconnues d'émetteur. Cependant sa solution ne peut pas se servir des mesures supplémentaires. Il y a d'autres techniques qui peuvent traiter des situations plus générale avec des mesures supplémentaires incluent la méthode sphérique d'interpolation (the spherical interpolation method), ce qui est dérivé de la minimisation des moindres carrés d'équation d'erreur, et de la méthode du clivage et conquérir (the divide and conquer method). L'estimation de cette méthode est constituée en combinant les évaluations de maximum de vraisemblance **ML** en utilisant les recouvrements possibles des sous-sections du vecteur des données des mesures. En plus de ces méthodes, il existe d'autres et qui sont adapté selon le niveau de bruit de l'environnement [MAO 07].

3.3 Algorithmes de localisation basé sur la distance

Les algorithmes de localisation basée sur la distance, les plus connus, sont basés sur l'utilisation du **GPS**. Dans le meilleur des cas, les mesures de distance à trois satellites de **GPS** permettent au récepteur de **GPS** de déterminer uniquement sa position. En réalité, quatre satellites, plutôt que trois, sont exigés en raison de l'erreur de synchronisation dans l'horloge de récepteur. La quatrième mesure de distance fournit les informations dont l'erreur de synchronisation du récepteur peut être corrigée et l'horloge de récepteur peut être synchronisée à une exactitude mieux que 100 ns.

Généralement dans un **WSN**, pour un capteur non localisé x_t avec des mesures bruit-souillées de distance d_1', d_2', …d_N' à la présence de N ancres localisés à x_1, …, x_N, le problème d'évaluation d'endroit peut être formulé en utilisant une approche de maximum de vraisemblance **ML** comme suit :

$$\hat{\mathbf{x}}_t = \arg\min[\mathbf{d}(\hat{\mathbf{x}}_t) - \tilde{\mathbf{d}}]^{\mathrm{T}}\mathbf{S}^{-1}[\mathbf{d}(\hat{\mathbf{x}}_t) - \tilde{\mathbf{d}}],$$

Où d' est le vecteur des mesures de distances à N×1, d(x'$_t$) est aussi un vecteur de N×1, et S est la matrice de covariance des erreurs de mesure de distance.

Des méthodes numériques peuvent être exploitées, comme celle de l'algorithme de descente de gradient, pour rechercher la solution et qui donne des évaluations de position de niveau supérieur [MAO 07].

3.4 Algorithmes de localisation basée sur RSS profilant

Ce type des techniques ont été principalement employés pour l'évaluation de position dans **WLAN**s. Où, chaque capteur exploite les mesures de puissance de signal

qu'il se rassemble, provenir des nœuds d'ancre dans sa région de détection ; crée ainsi sa propre empreinte digitale de **RSS**, ce qui est transmis, alors, à la station centrale [MAO 07].

3.5 Localisation basée sur des mesures hybrides

Il y a un certain nombre d'autres algorithmes de localisation basés sur la fusion de données des mesures hybrides, [MAO 07] :

- **McGuire** et autres, a exploré la fusion de données des mesures de **RSS** et de **ToA** pour la localisation terminale mobile dans un réseau cellulaire de **CDMA** (**C**ode **D**ivision **M**ultiple **A**ccess).
- **Li** et **Zhuang** ont considéré la localisation mobile d'utilisateur en utilisant des mesures hybrides de **TDoA**/**AoA** dans un système à large bande du macrocell **CDMA** avec le duplex de division de fréquence.
- **Gu** et **Gunawan** ont considéré la localisation mobile d'utilisateur dans un réseau cellulaire de **CDMA** en utilisant des mesures hybrides d'**AoA**/**ToA**.

Fondamentalement, la localisation basée sur des mesures hybrides peut réaliser une amélioration d'exécution au-dessus de cela basé sur un type simple de mesure parce que le bruit de mesure pour différents types de mesures vient de différentes sources. Par conséquent, les erreurs dans l'évaluation d'endroit pour chaque type de mesure sont largement indépendantes. Cette indépendance entre différents types de mesures peut être exploitée par des techniques de fusion de données pour créer des estimateurs qui ont une meilleure exactitude que des estimateurs basés sur les types simples de mesure. Parmi ces techniques hybrides, la fusion des mesures de **RSS** et de **ToA** semble être la plus attrayante pour un **WSN,** en raison de son besoin en matériel relativement simple.

4.4 Etude comparative des algorithmes AT-Free et HT-Refine

4.1 Fonctionnement global de AT-Free

Un nœud *unknown* ne fait que calculer sa position et rediffuser les paquets reçus. Une ancre estimée effectue les mêmes tâches qu'un nœud non localisé à l'exception, qu'elle diffuse sa position au fur et à mesure que sa borne d'erreur diminue. Pour une ancre parfaite, elle ne diffuse qu'une seule fois sa position dans le réseau et effectuera les mêmes traitements, lors de la réception d'un paquet, que les ancres initiales.

Après le déploiement du réseau, la première phase est la diffusion des coordonnées des ancres initiales, un capteur qui reçoit ces positions va les enregistrer en mémoire et

attendre un certain moment pour recevoir une éventuelle amélioration. Juste après, le nœud déclenche le calcul de ses coordonnées et de sa borne d'erreur. Si cette dernière est inférieure au seuil Γ alors le capteur *unknown* deviendra une ancre estimée et diffusera à son tour sa position et sa borne d'erreur. Comme décrit dans le chapitre précédent, **AT-Free** gère la diffusion en limitant le nombre de diffusion d'une ancre estimée. Ainsi le capteur ne diffuse que si sa borne d'erreur diminue. Lorsque cette dernière devient inférieure à ρ, le nœud est considéré comme étant très bien localisé (ancre parfaite) et ne diffusera qu'une seule fois pour ne pas surcharger le réseau.

La figure 4.3 montre le diagramme d'activités des différents capteurs dans le réseau.

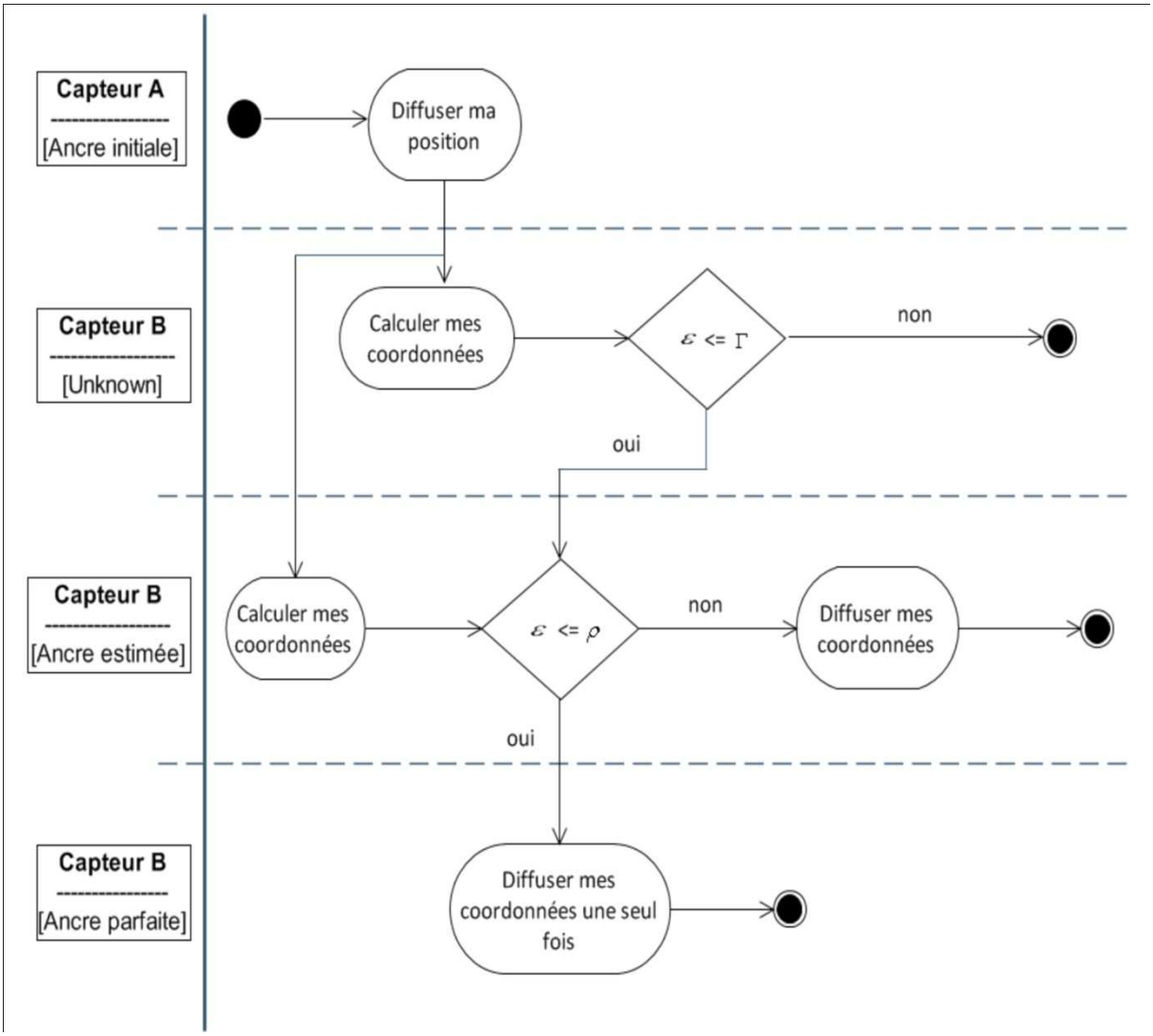


Figure 4. 3 : Diagramme d'activités de la méthode AT-Free.

4.2 Implémentation de la méthode AT-Free

Dans la méthode **AT-Free**, , un capteur *unknown* peut devenir une ancre selon sa borne d'erreur ε, il peut transiter, soit en une ancre estimée, dans ce cas de figure, il diffuse sa position au fur et à mesure que sa borne diminue pour améliorer la précision de calcul dans le réseau, soit en une ancre parfaite, dans ce cas, le capteur est considéré comme très bien localisé et il diffusera une seule fois sa position. Sachant que Γ et ρ sont respectivement le seuil pour devenir une ancre estimée et celui pour devenir une ancre parfaite, le diagramme, de la figure 4.4, représente les transitions d'états qu'un capteur *unknown* peut effectuer, [ARO 10] :

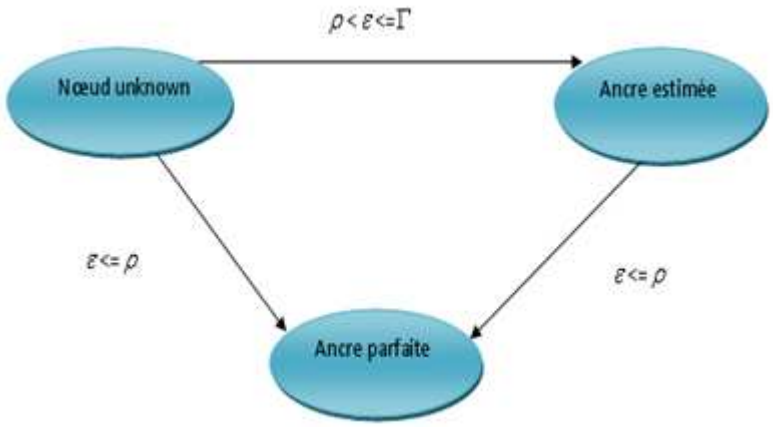


Figure 4. 4 : Diagramme de transition d'état d'un capteur unknown dans AT-Free. [ARO 10]

Un capteur *unknown* ou bien ancres non initiale, sauvegarde les coordonnées, l'identifiant, le nombre de sauts minimums et la borne d'erreur de l'ancre émettrice. Lorsque le nœud *unknown* reçoit les coordonnées d'une ancre pour la première fois, il les sauvegarde dans sa mémoire et va attendre un certain temps t dans le but d'améliorer le nombre de sauts minimums qui le sépare de lui. Après épuisement du t, il déclenche le calcul de ses coordonnées et de sa borne d'erreur ε par rapport à cette ancre. Selon la valeur de sa borne, le nœud pourra alors devenir une ancre estimée ou parfaite et ainsi aider les autres capteurs à optimiser leurs localisations. Apres avoir déclencher le calcul, les informations de l'ancre concernée seront supprimé de la mémoire. Le t sera modélisée par un compteur Δt qui sera décrémenté périodiquement.

Par contre, pour les ancres, chaque ancre initiale contient une structure pour sauvegarder l'identifiant et le nombre de sauts minimums qui les séparent des autres

ancres. Puisque les ancres initiales connaissent leurs coordonnées, cette structure servira à optimiser l'inondation du réseau et ainsi éliminer les paquets qui ne sont plus nécessaires. En effet, lors de la diffusion des paquets, une ancre à besoin de savoir si le paquet reçu est déjà passé par elle ou non. Si tel n'est pas le cas, alors, elle incrémente le nombre de sauts qu'a traversé le paquet et le diffuse ; sinon, elle compare le nombre de sauts parcourus par le paquet et le nombre de sauts sauvegardés dans sa mémoire et décidera si elle doit le supprimer ou le diffuser.

Pour l'implémentation de ces structures de données, nous avons choisi d'utiliser deux listes chaînées, gérées par la politique **FIFO**, pour représenter les deux structures. Ainsi, elles permettent l'ajout et la suppression des éléments d'une façon dynamique. De plus, nous avons utilisé un tableau booléen qui a pour taille le nombre maximal de diffusions autorisées. Ce dernier est initialisé à '**vrai**'. Chaque case du tableau représente un intervalle de la borne d'erreurs. Aussi, lorsqu'un capteur calcule sa borne, il teste dans quel intervalle il se situe. Si la case concernée est égale '**faux**', cela signifie qu'il l'a déjà diffusée.

$$\Gamma \geq \lambda\Gamma \geq \lambda^2\Gamma \geq \ldots \geq \lambda^i\Gamma \geq \ldots \geq \rho, \qquad i \in IN$$

Chaque $[\lambda^{i+1}\Gamma, \lambda^i\Gamma]$ représente un intervalle de diffusion.

Avec λ une constante multiplicatif : $0 < \lambda < 1$. Donc pour k diffusions autorisées, on aura :

Si ($\lambda^{i+1}\Gamma < \varepsilon \leq \lambda^i\Gamma$) et (Tab[i] = 'vrai') alors

- Diffuser.
- Tab[i] = faux.

Sachant que Tab est un tableau booléen qui a pour dimension le nombre de diffusions autorisées k.

On a choisi d'implémenter deux timers : Timer_send et Timer_dec, le premier est utilisé par les ancres initiales. Il a pour rôle d'autoriser l'ancre à diffuser sa position après un temps T, choisi aléatoirement dans un intervalle défini ; ceci pour éviter le cas où toutes les ancres diffusent au même moment (trop de collisions). Le deuxième timer déclenchera la décrémentation des Δt . Timer_dec est un timer période, à chaque fois qu'il décrémente les Δt de la file, il se réinitialise de façon automatique. Le fonctionnement des timers est modélisé par le diagramme de séquence donné par la figure 4.5 :

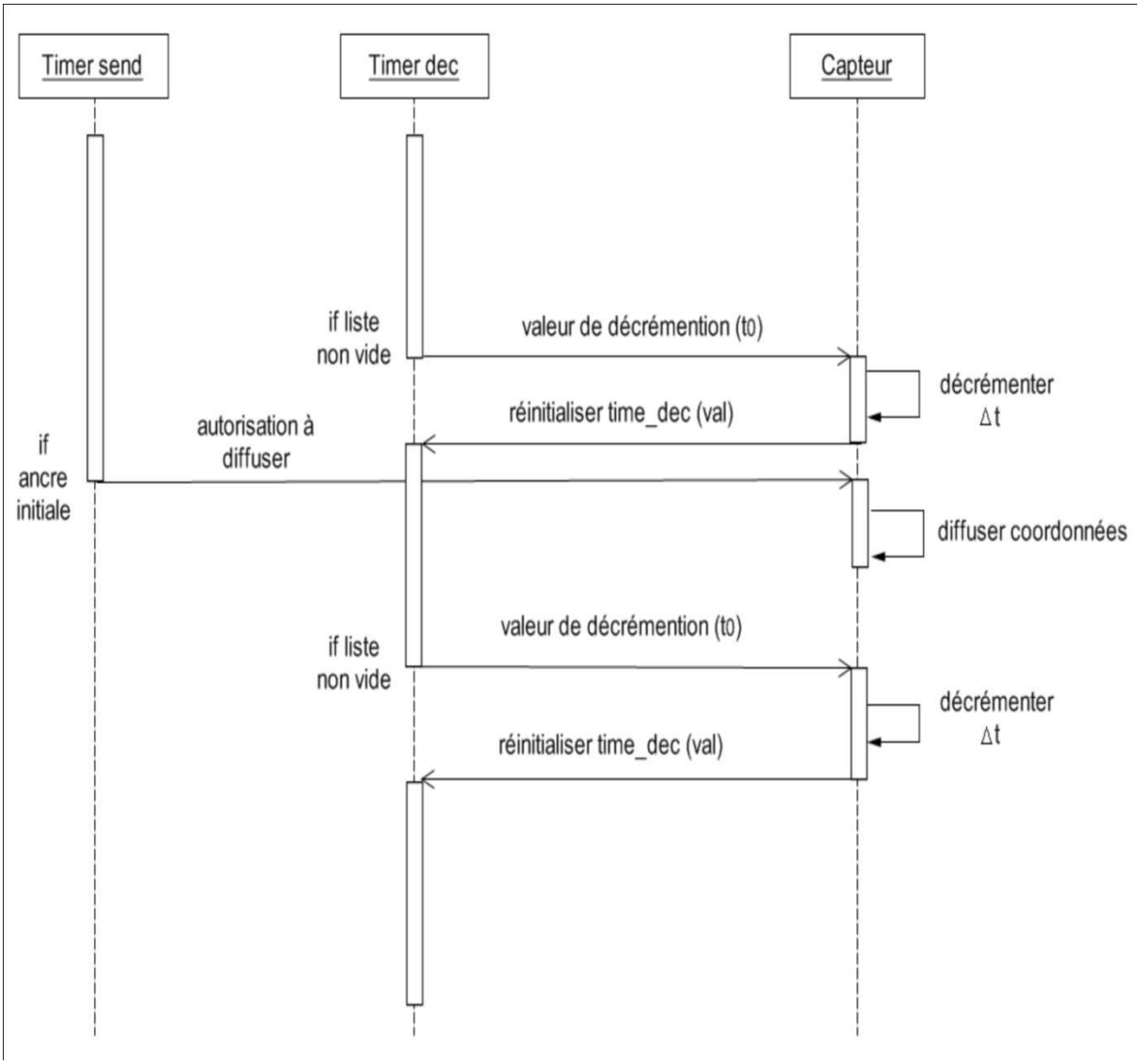


Figure 4. 5 : Diagramme de séquence des timers de la méthode AT-Free.

Lors de réception d'un paquet, les mêmes contrôles sont effectués par tous les types de nœuds, la différence se résume dans le type de données à sauvegarder :

Ancre **GPS** / parfaite : le nœud enregistre seulement l'identifiant et le nombre de sauts parcourus.

Capteur *unknown* / ancre estimée: Le capteur enregistre l'identifiant, les coordonnées, la borne d'erreur de l'ancre émettrice ainsi que le nombre de sauts parcouru par le paquet.

Les traitements que les nœuds du réseau effectuent sont comme suit :

Pseudo code :

- Si je suis l'émetteur du paquet alors supprimer (paquet)
- Sinon
  - Si le paquet a déjà été reçu :
    - comparer le nombre de sauts parcouru par le paquet a celui stocké en mémoire :
      - S'il y a une amélioration alors : mettre à jour les informations adéquates et rediffuser le paquet.
    - Sinon : supprimer (paquet).
  - Sinon : sauvegarder les informations nécessaires et effectuer un forward.

La figure 4.6 représente l'organigramme des tâches effectuées à la réception d'un paquet.

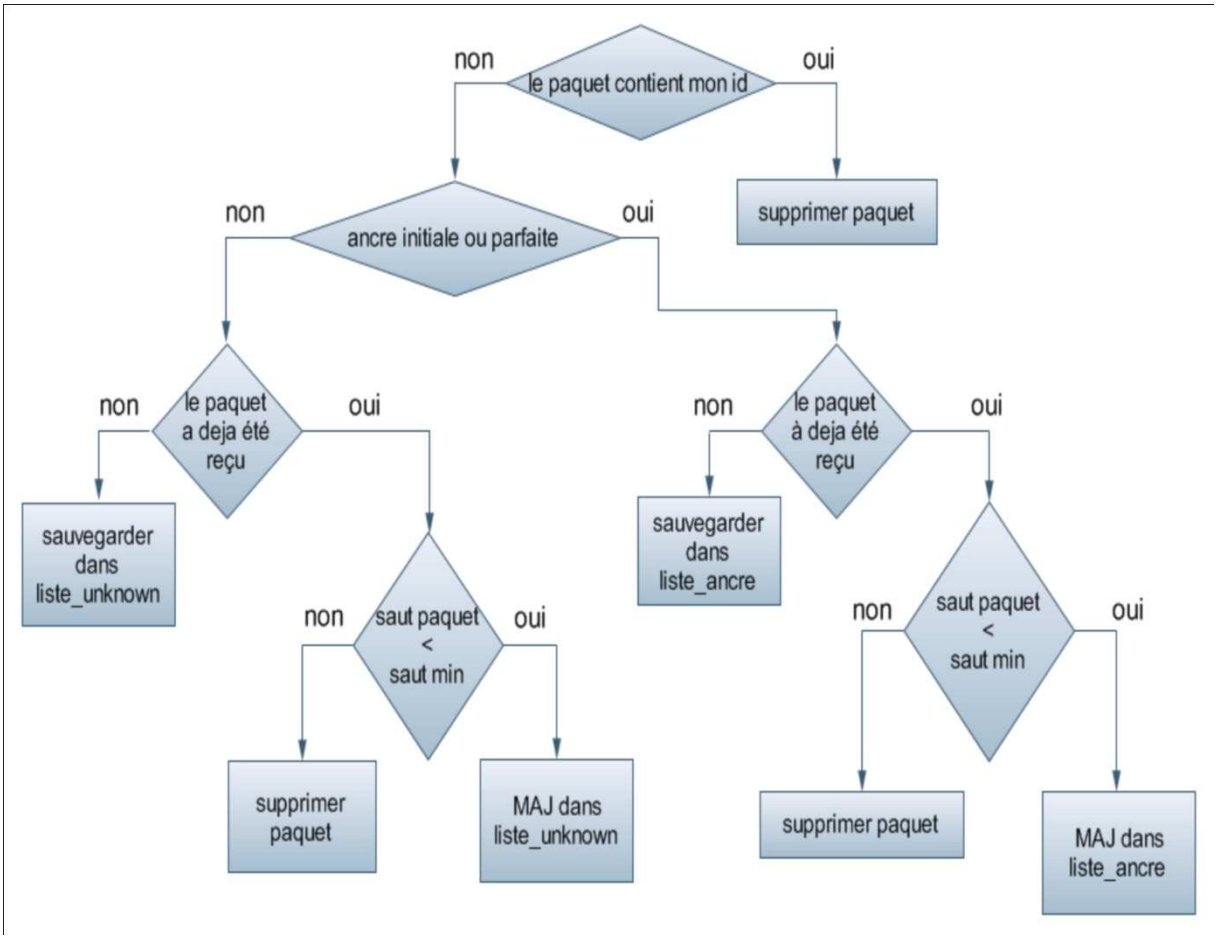


Figure 4. 6 : Organigramme de la réception dans la méthode AT-Free.

Liste_ancre et Liste_*unknown* représentent respectivement la structure de données d'un nœud ancre et d'un nœud *unknown*. Saut_min représente le nombre de sauts sauvegardé en mémoire.

Dans l'implémentation d'**AT-Free**, au sein de chaque nœud le réseau est représenté par une grille (matrice). Une case de cette grille est égale à 0.01 r (avec r le rayon de communication). Lorsqu'un nœud reçoit la position d'une ancre, il incrémente les cases dans la grille qui pourraient correspondre à sa position :

Si le nœud et l'ancre sont voisins, les cases considérées appartiennent au disque centré en l'ancre et de rayon r.

• Si le nœud et l'ancre ne sont pas voisins, les cases incrémentées se situent entre les deux cercles centré en l'ancre et de rayons respectifs d1 et d2 tels que :

$$d_1 = r \text{ et } d_2 = r \times h$$

h est le nombre de sauts minimum entre l'ancre et le nœud.

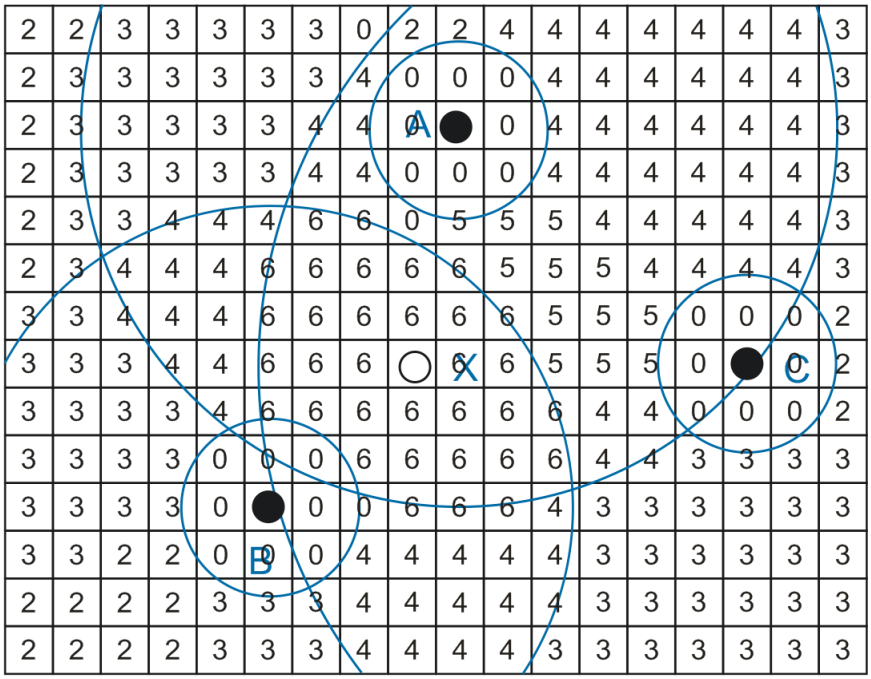


Figure 4. 7 : Représentation du réseau par une grille.

La figure 4.7 est un exemple de grille : lorsque le nœud X reçoit la position de l'ancre A (resp B, C), il incrémente toutes les cases entre les deux cercles centrées en A (resp B, C) et de rayons respectifs r et r x h_A (resp r x h_B, r x h_C) avec h_A (resp. h_B, h_C) le nombre de sauts minimums entre X et l'ancre A (resp. B, C). La zone contenant le nœud X est représentée par les cases de valeurs maximales. Dans la figure 4.6, cette zone est constituée des cases égales à 6. X calcule le centre de gravité de cette zone et obtient une estimation de sa position [ARO 10].

4.3 Fonctionnement global de HT-Refine

Pour résumer le fonctionnement de cette méthode ainsi que les différentes activités qui interagissent entre les ancres, les nœuds *unknown*s et les timers, nous avons modélisé **HT-Refine** avec le schéma **UML** (**U**nified **M**odeling **L**anguage) sur la figure 4.8 :

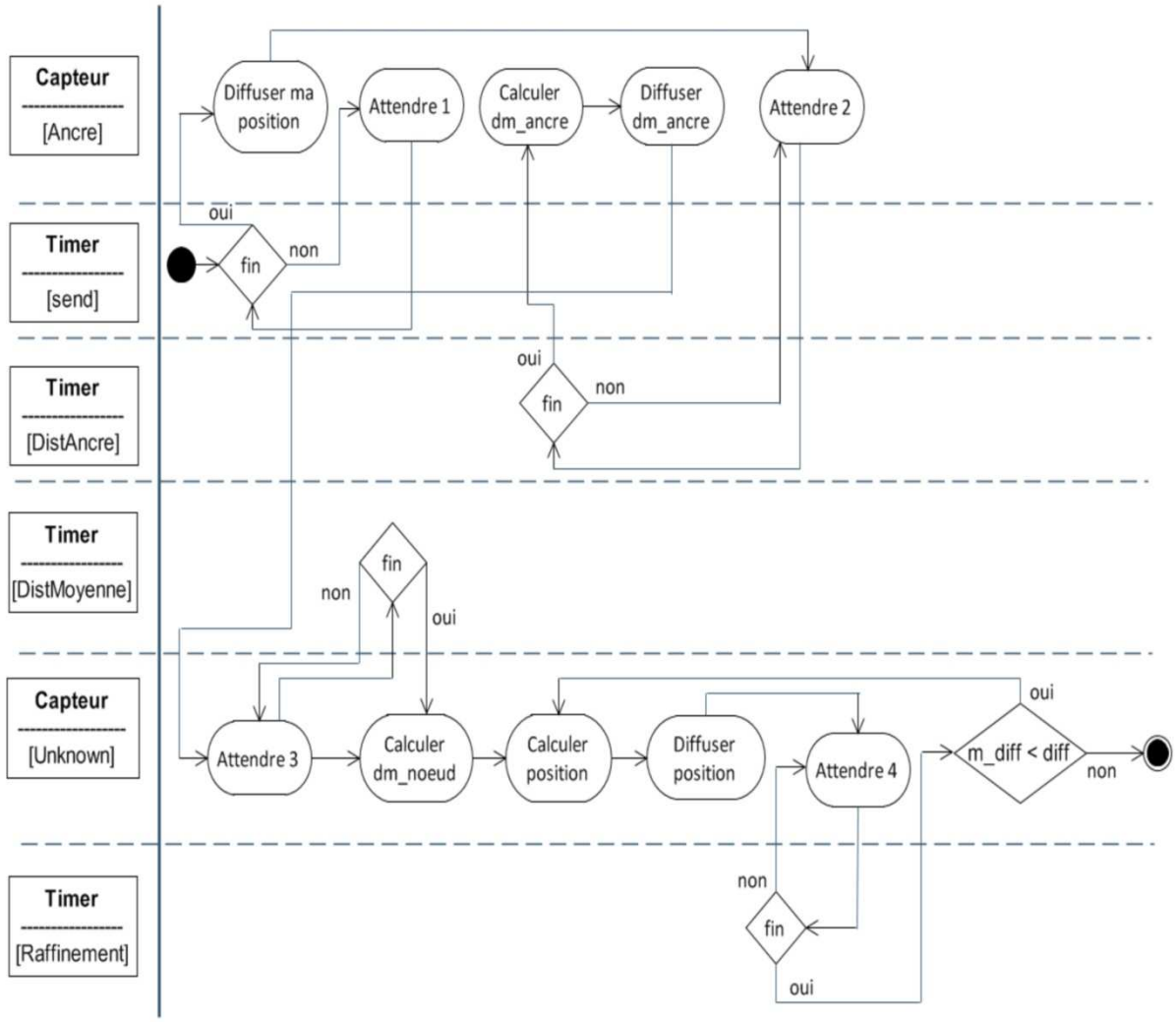


Figure 4. 8 : Diagramme d'activités de la méthode HT-Refine.

Avec m_diff le nombre de diffusions effectuées par le nœud et diff le nombre de diffusions autorisées. Les activités Attendre 1 et Attendre 2 sont utilisées par l'objet ancre afin d'attendre respectivement la fin des timers **TimerSend** et **TimerDistanceAncre**. Les activités Attendre 3 et Attendre 4 correspondent à l'objet capteur *unknown* pour attendre respectivement la fin des deux timers **TimerDistanceMoyenne** et **TimerRaffinement**.

4.4 Implémentation de la méthode HT-Refine

Cette méthode s'exécute en deux phases [SAV 02] :

- **DV-Hop**

- Raffinement des positions.

La phase **DV-Hop** permet aux nœuds du réseau d'avoir une première estimation de leurs positions. **DV-Hop** est constituée de deux vagues de diffusions effectuées par les ancres ; la première sert à diffuser leurs positions et la deuxième à diffuser les distances moyennes calculées. La phase de raffinement servira à améliorer les estimations des nœuds, le nombre de vagues de diffusions dans cette phase dépend du nombre d'itérations définies.

Afin de gérer ces différentes étapes d'un point de vu conceptuel, nous avons conçu des timers et des structures de données en adéquations avec les caractéristiques de **HT-Refine**.

La méthode **HT-Refine** utilise les deux types classiques de capteurs utilisés dans la localisation, les ancres et les nœuds *unknown*. Contrairement à **AT-Free**, il n'y a pas de transition d'état entre les capteurs.

Pour déterminer dans quelle phase est le réseau, lors de la réception d'un paquet, le nœud teste un champ indice sur le paquet transmis, qui lui indiquera le traitement nécessaire pour la phase adéquate, [ARO 10] :

- Si indic = 0 : phase DV-hop première vague de diffusion.
- Sinon
  - Si indic = 1 : phase DV-hop deuxième vague de diffusion.
  - Sinon : phase raffinement.

La première vague de diffusion est la phase où les ancres diffusent leurs positions, la seconde vague est celle où les ancres diffusent les distances moyennes qu'elles ont calculées. Le format du paquet utilisé dans la phase de raffinement est le même que celui de la première vague de diffusion de la phase **DV-Hop**.

On a choisi d'implémenter quatre timers; chacun d'entre eux définit le début d'une certaine opération spécifique.

Pour la phase **DV-Hop** :

- **TimerSend** : c'est le même timer utilisé lors de la méthode **AT-Free**, il sert à débuter la phase de diffusion des positions des ancres.
- **TimerDistanceAncre** : Ce timer ne concerne que les ancres, après qu'une ancre ait diffusée ses coordonnées, elle attend le déclanchement de ce timer pour calculer

les distances qui la séparent des autres ancres. Ensuite, elle calcule sa distance moyenne et la diffuse dans le réseau.

- **TimerDistanceMoyenne** : la finalité de ce timer déclenche le calcul de la distance moyenne par les nœuds *unknown*s en utilisant les informations reçues de la part des ancres. Une fois la distance moyenne obtenue, le nœud estime sa position.

Et pour la phase de raffinement :

- **TimerRaffinement** : Après avoir diffusé ses positions estimées une première fois, le nœud va initialiser ce timer. La fin de ce dernier déclenchera les opérations suivantes : le recalcule des coordonnées et l'envoi de ces nouvelles positions. Le timer sera réinitialisé dans chaque étape itérative.

Le diagramme de séquence, du la figure 4.9, illustre les interactions entre le nœud et les timers.

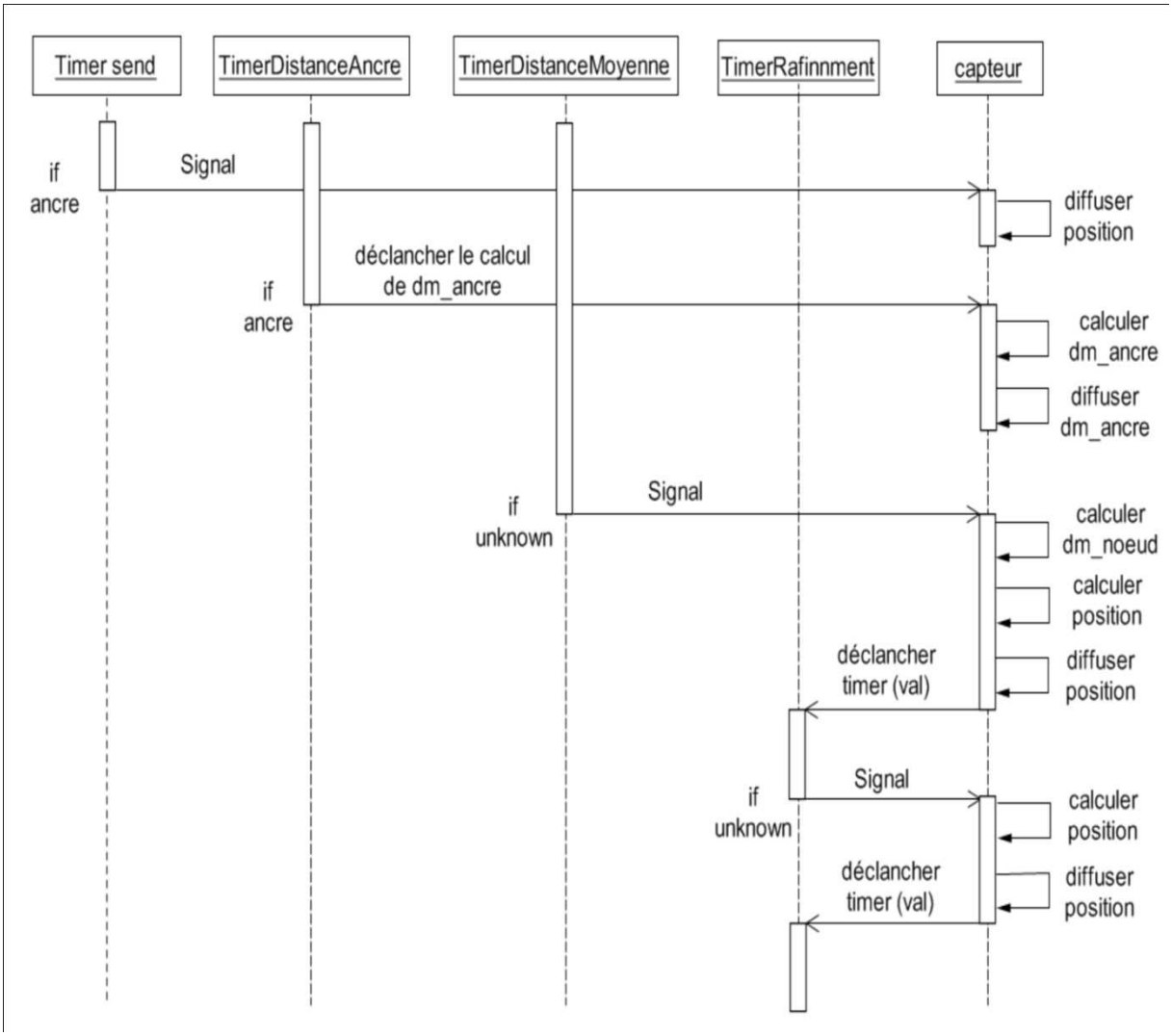


Figure 4. 9 : Diagramme de sequence de la méthode HT-Refine.

Avec dm_ancre, la distance moyenne calculée par l'ancre et dm_noeud la distance

moyenne calculée par le nœud *unknown* par rapport aux dm_ancres qu'il a reçues.

Lorsqu'un nœud reçoit un paquet, il détermine le traitement adéquat selon la phase **DV-Hop** ou **HT-Refine**.

La figure 4.10 montre les différents traitements que peut effectuer un capteur lorsqu'il reçoit un paquet.

- ***Organigramme***

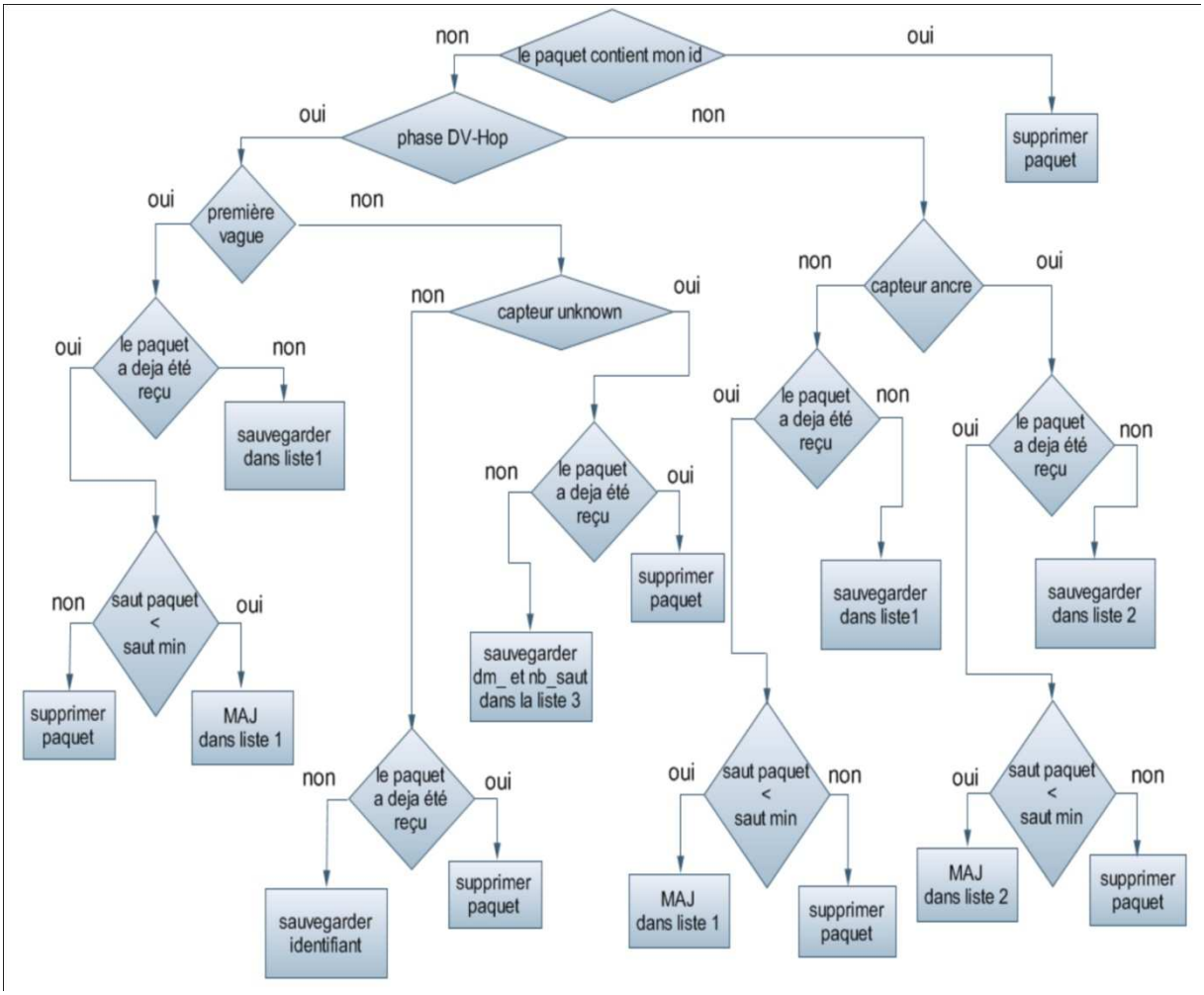


Figure 4. 10 : Organigramme de la fonction reception pour ht-refine.

Avec :

Liste 1 : représente la structure de données qui contiendra les coordonnées reçues des autres nœuds et le nombre de sauts parcourus par le paquet.

Liste 2 : représente la deuxième structure de données des ancres qui contiendra l'identifiant et le nombre de sauts des paquets reçus correspondant à un nœud. Rappelons qu'elle ne sert qu'à optimiser la diffusion de la phase de raffinement en supprimant les paquets que le nœud a déjà reçus.

Liste 3 : correspond à la structure de données des nœuds *unknown*s qui contiendra les

distances moyennes reçues.

Pour la phase de raffinement, dès lors que le capteur *unknown* estime sa position, il va la diffuser afin de raffiner les estimations des autres nœuds ; cela va se faire de façon itérative. L'algorithme suivant illustre le traitement effectué par le timer « **TimerRaffinement** » pour gérer cette phase :

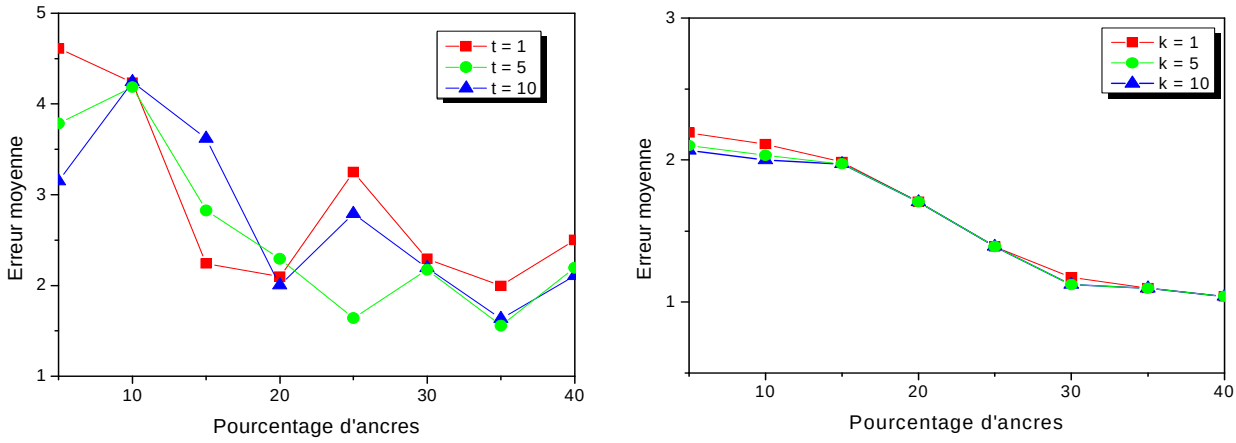
```
Si (mon_nb_iteration < nb_iteration_autorisé) alors
{
    Calculer ma position. ;
    Enlever les éléments correspondant aux nœuds unknowns de la liste ;
    Diffuser ma position ;
    Réinitialiser le timer ;
    mon_nb_iteration ++ ;
}
```

4.5 Résultats d'évaluation des performances de HT-Refine et de AT-Free

Pour évaluer le degré de précision, on fera appel à la métrique de l'erreur moyenne absolue (**MAE**) ; On ne prend pas en considération les ancres (nœuds qui sont localisés avec une erreur de 0% dés le départ). Nous étudierons aussi l'énergie totale consommée lors du processus de localisation et le nombre de paquets transmis.

A. Impact du nombre d'ancres sur l'erreur moyenne

Le graphe 4.1 illustre l'impact du pourcentage d'ancres sur l'erreur moyenne.



(a) HT-Refine ***(b) AT-Free***

Graphe 4.1 : Impact du nombre d'ancres sur l'erreur moyenne. [ARO 10]

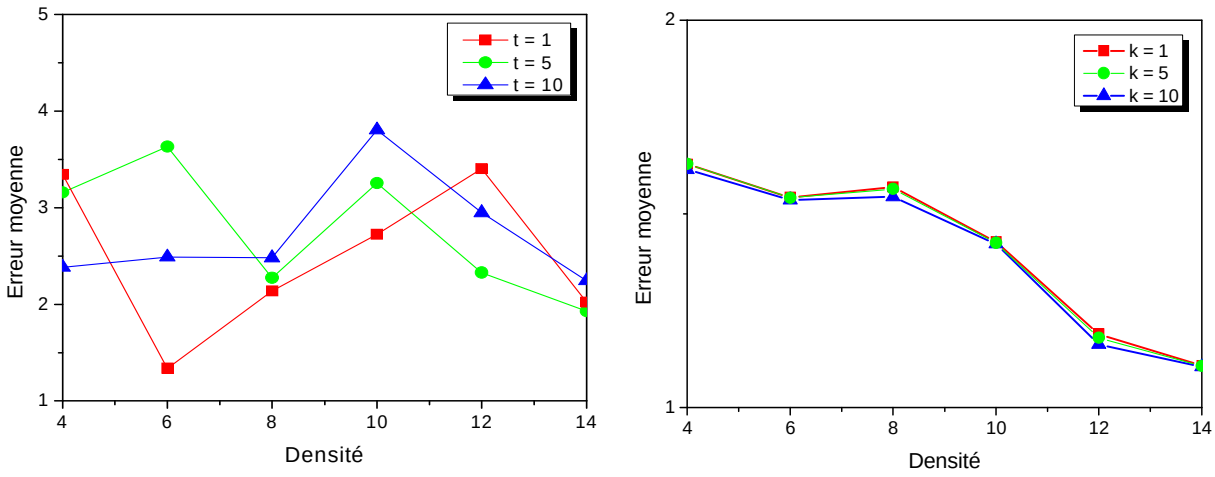
Nous remarquons dans le graphe 4.1 (a) que globalement l'erreur moyenne diminue avec l'augmentation du nombre d'ancres dans le réseau. Il est claire que plus le pourcentage d'ancres est élevé plus l'erreur moyenne diminue, cela confirme le principe de l'utilisation des ancres dans la localisation. Ce qui est intéressant dans ces résultats c'est que l'erreur moyenne diminue d'une façon plus ou moins instable : on remarque quelques augmentations de l'erreur moyenne ; Les ancres ajoutées ont eu un impact négatif sur le calcul de la distance moyenne entre les nœuds.

On constate que l'erreur moyenne ne diminue pas avec l'augmentation du nombre de raffinements, on peut même constater que dans certains cas l'erreur moyenne a augmenté. Rappelons que le calcul des positions dans **HT-Refine** est basé sur la distance moyenne d'un saut. Quand la topologie utilisée a une variance importante des distances entre voisins cela provoquera des erreurs lors de l'estimation des distances entre les ancres et les nœuds *unknown*s. Ces erreurs vont se propager lors de la phase de raffinement. En effet lorsqu'un capteur X calcule sa position dans la phase **DV-Hop**, il va la diffuser afin de raffiner les résultats des autres nœuds. Prenons le cas où le nœud X s'est bien localisé, lors de la nième phase de raffinement, le capteur X reçoit les coordonnées des autres nœuds *unknown*s qui ont estimé leurs positions avec une grande marge d'erreur, X va recalculer sa position en se basant sur de fausses informations ce qui induit une dégradation de la précision alors qu'au départ, le nœud concerné avait une bonne estimation de sa position.

Le graphe 4.1 (b) montre que plus il y'a des ancres dans le réseau plus la précision de calcul augmente, mais cette fois-ci, pour **AT-Free** et contrairement à **HT-Refine**, l'erreur moyenne diminue de façon beaucoup plus stable. Les capteurs participent à l'amélioration des calculs en devenant des ancres estimés. Un nœud peut savoir si sa position calculée est loin ou pas de sa position réelle et va juger s'il peut diffuser ou non. On remarque aussi qu'à partir d'un certain pourcentage d'ancres, le nombre de diffusions autorisées n'a plus d'influence.

B. Impact de la densité sur l'erreur moyenne

Dans cette section, nous avons varié la densité du réseau en augmentant le nombre de nœuds. Les nœuds ont été rajoutés avec des coordonnées aléatoires. Le but est d'évaluer l'influence de la densité sur la précision du calcul de la position. Nous avons réalisé ces scénarios avec un pourcentage d'ancre fixé à 20 %.



(a) HT-Refine ***(b) AT-Free***

Graphe 4.2 : Impact de la densité sur l'erreur moyenne. [ARO 10]

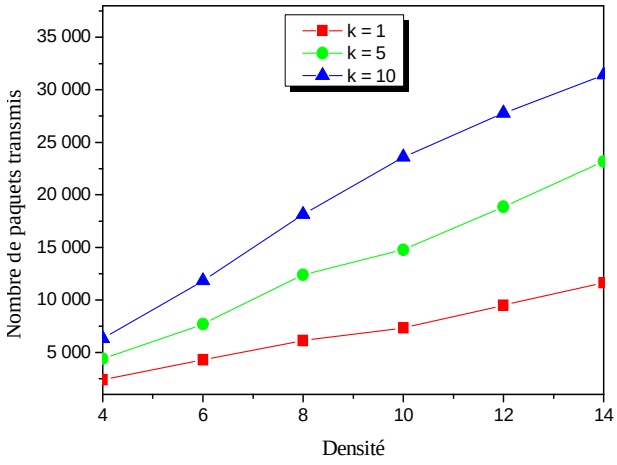
Selon le graphe 4.2, nous remarquons dans **HT-Refine** que l'augmentation de la densité et le nombre de raffinement n'améliore pas la précision du calcul. Se sont les mêmes raisons évoquées dans la section « impact du pourcentage d'ancres sur l'erreur » qui ont provoqué cette instabilité.

Pour ce qui est de la méthode **AT-Free**, on remarque que l'erreur moyenne diminue en augmentant la densité. L'accroissement de la densité ajoute des contraintes de voisinages ; un nœud recevra donc plus d'informations. Pour le nombre de diffusion autorisé, on constate qu'il n'a pas d'influence sur l'amélioration des résultats. Nous avons vu dans la section précédente que le nombre de diffusions n'a plus d'impact sur le résultat si le pourcentage d'ancres est supérieur à un certain seuil.

C. Impact de la densité sur le nombre de paquets transmis et sur l'énergie

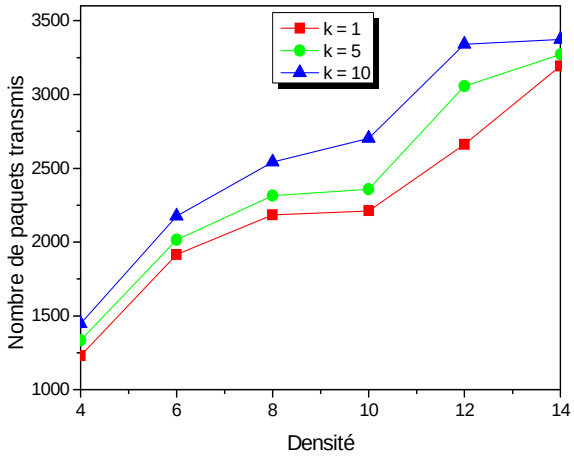
Le graphe 4.3 montre l'impact de la densité sur le trafic dans les deux méthodes. Plus la densité augmente plus le nombre de paquets augmente aussi; ce qui est tout à fait logique. On définit le trafic comme étant le nombre de paquets transmis. Dans le cas où le nombre d'itérations est fixé à un, on remarque que **HT-Refine** a un trafic supérieur à celui d'**AT-Free**. En effet, après que les ancres aient diffusé leurs coordonnées, chaque nœud va essayer de trouver les distances minimales, en nombre de sauts, qui le séparent des ancres dont les coordonnées lui ont été parvenues. Cette phase est présente dans les deux algorithmes. Cela dit, dans le cas de **HT-Refine**, le nombre de paquets transmis durant la phase de calcul de la distance inter-nœuds moyenne va s'ajouter au nombre de paquets

transmis durant la phase précédente ; ce qui explique pourquoi **HT-Refine** a un degré de communication plus élevé que celui d'**AT-Free**.



HT-Refine

(a)



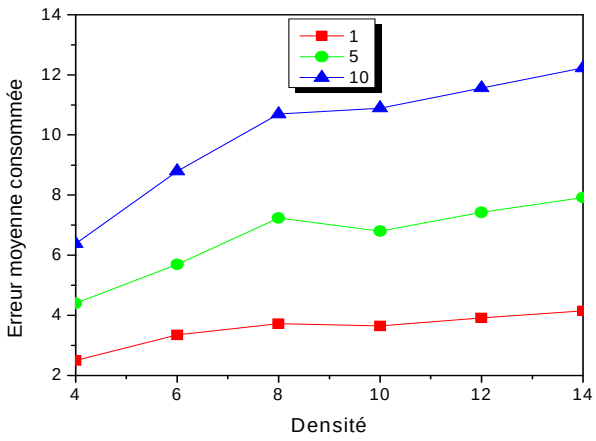
AT-Free

(b)

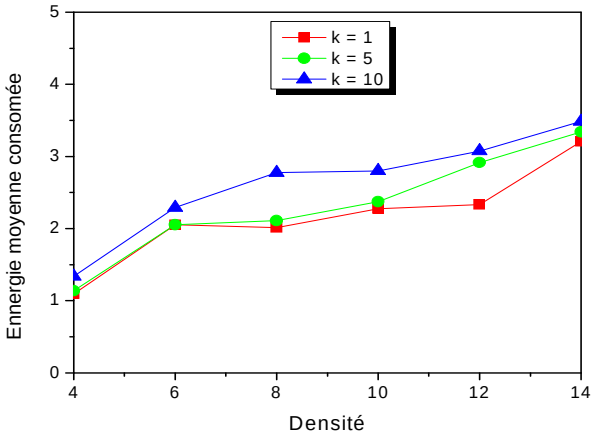
Graphe 4.3 : Impact de la densité sur le nombre de paquets transmis. [ARO 10]

La gestion des diffusions, quant à elle, joue un rôle important dans le cas où le nombre d'itération est supérieur à un. Dans **AT-Free**, pour qu'un nœud, dont la position vient d'être estimée, puisse diffuser ses coordonnées il faudrait que l'erreur entre sa position estimée et sa position réelle soit inférieure à un certain seuil. Pour que le nœud puisse diffuser une deuxième fois, il faudrait que cette erreur baisse au dessous d'un deuxième seuil et ainsi de suit. Si l'erreur baisse au dessous d'un seuil ρ, le nœud sera considéré comme parfaitement localisé et il ne cherchera plus à améliorer ou à diffuser ses coordonnées. Les contraintes imposées par l'algorithme d'**AT-Free** ont pour effets de réduire le nombre de diffusion durant le processus de localisation. **HT-Refine**, quant à lui, ne définit aucune politique de gestion des diffusions ; le graphe 4.3 (a) montre comment le degré de communication évolue d'une façon dramatique en fonction du nombre d'itérations dans le cas de **HT-Refine**.

Le graphe 4.4 représente la consommation moyenne d'énergie par rapport à la densité. Sans grande surprise, le schéma que suivent ces graphes ressemble à celui que suivent les graphes du nombre de paquets transmis par rapport à la densité. Il est à rappeler que l'énergie nécessaire pour transmettre un bit de données équivaut à exécuter mille instructions.



HT-Refine



AT-Free

Graphe 4.4 : Impact de la densité sur l'énergie. [ARO 10]

4.6 Evaluation comparative

Le calcul des positions dans **HT-Refine** est basé sur la distance moyenne d'un saut. Quand la topologie utilisée a une variance importante sur les distances entre voisins, cela provoquera des erreurs lors de l'estimation des distances entre les ancres et les nœuds *unknown*s. Ces erreurs vont se propager lors de la phase de raffinement. En effet lorsqu'un capteur X calcule sa position dans la phase **DV-Hop**, il va la diffuser afin de raffiner les résultats des autres nœuds.

Contrairement à **HT-Refine**, dans **AT-Free**, les capteurs participent à l'amélioration des calculs en devenant des ancres estimés. Un nœud peut savoir si sa position calculée est loin ou pas de sa position réelle et va juger s'il peut diffuser ou non. A partir d'un certain pourcentage d'ancres, le nombre de diffusions autorisées n'a plus d'influence, car avec une forte densité d'ancres, le nombre d'informations de localisation diffusées est déjà conséquent dès le départ, donc l'augmentation de la valeur de paramètre k n'a pas d'intérêt. A l'inverse, dans une configuration à faible densité d'ancres, l'augmentation de la valeur de k ajoutera aux nœuds des informations de localisation supplémentaires.

HT-Refine	AT-Free
L'erreur moyenne diminue de façon instable par rapport à l'augmentation du nombre d'ancres.	L'erreur moyenne diminue de façon très stable par rapport à l'augmentation du nombre d'ancres.
L'augmentation du nombre de diffusions	L'augmentation du nombre de diffusions

n'a aucun impact sur l'amélioration des résultats.	améliore les résultats jusqu'à attendre un certain pourcentage d'ancres

Tableau 4.1 : Evaluation comparative entre ht-refine et at-free.

4.7 Conclusion

Nous avons donné une description détaillée des protocoles de localisations **AT-Free** et **HT-Refine**. La conception des deux protocoles a été présentée à l'aide de différents diagrammes **UML**, afin de simplifier la compréhension.

A travers notre comparaison, nous avons constaté que l'absence d'une politique de gestion des diffusions dans l'algorithme **HT-Refine** à pour conséquence un trafic très important et une propagation des erreurs d'estimation ; **HT-Refine** est un algorithme mal adapté aux topologies aléatoires dans lesquelles la distance entre les voisins varie énormément. **AT-Free** de son côté offre une politique de gestion des diffusions qui minimise le nombre de messages transmis ; de plus, dans **AT-Free** un nœud n'est autorisé à diffuser sa position que si son erreur de mesure passe au dessous d'un certain seuil. Le nœud qui reçoit les positions d'un nœud estimé est sûr de la fiabilité des informations reçues.

4.5 Comparaison quantitative entre quelques algorithmes de localisation distribuée

Parmi les algorithmes de localisation, on a les trois algorithmes suivants :

-**S**ystème de **P**ositionnement **A**d Hoc (**APS**) [NIC 01]

- N-hop multilateration [SAV 02]

- Robust positioning [SAV 01]

Ces trois algorithmes abordent la question de la localisation dans les réseaux de capteurs Ad hoc. Nous voulons déterminer la position de différents capteurs sans basé sur une infrastructure externe (station de base, satellites, …), et ces algorithmes partagent une structure commune et triphasée :

1. déterminent les distances entre les capteurs et les ancres,
2. calcul des positions des nœuds
3. optionnellement, raffinement de positions calculées selon un procédé itératif.

Nous somme intéressés aux algorithmes vraiment distribués, car ils peuvent être utilisé pour des réseaux de capteur à grande échelle (plus de 100 capteurs). Ces algorithmes doivent être :

- auto-organisés (le nœud ne dépend pas d'une infrastructure existante).
- Robustes (être tolérant aux échecs de nœud et aux erreurs).
- Gère l'énergie efficacement (exiger peu de calcul et, particulièrement, lors de communication).

Ces conditions éliminent immédiatement certains des algorithmes de localisation proposés pour les réseaux de capteur. Donc les trois algorithmes choisis sont entièrement distribués et utilisent la diffusion locale pour la communication avec les voisins directs. Ce dernier dispositif leur permet d'être exécutés avant que n'importe quelle routine multi-sauts soit mise en place. Pour la simplicité et la facilité de la présentation nous limitons l'environnement à 2 dimensions, mais tous les algorithmes sont capables du fonctionnement à 3 dimensions.

Il est important de se rendre compte que les trois paramètres principaux de contexte (connectivité, fraction d'ancre et erreurs de gamme) sont dépendants. Des pauvres mesures de gamme peuvent être compensées en utilisant beaucoup d'ancres et/ou une connectivité élevée.

Ces trois méthodes donc utilisent lors de la phase 01, les techniques : **Sum-dist**, **DV-hop**, **Euclidean**, et pour la phase 2, on a les techniques : **Lateration** et **Min-Max**, en plus une phase optionnel pour le raffinement, selon la classification donnée par le tableau 4.2.

Phase	**Ad-hoc positioning**	**Robust positioning**	**N-hop multilateration**
1. distance	Euclidean	DV-hop	Sum-dist
2. position	Lateration	Lateration	Min-max
3. raffinement	non	oui	oui

Tableau 4.2 : Classification des algorithmes.

Lors d'étude de la phase 01, **DV-Hop** s'est avéré stable et prévisible, et **Sum-dist** et **Euclidean** ont montré à des tendances à l'évaluation de dessous les distances entre les ancres et les inconnus. La technique **Euclidean** a avéré avoir des difficultés en propagation d'information de distance dans des conditions non-idéales, menant à la basse assurance dans la majorité de cas. Les résultats de la phase 02 prouvent que la **latération** est capable d'obtenir des positions très précis, mais également qu'il est très sensible à l'exactitude et à la précision des évaluations de distance. **Min-max** est plus robuste, mais est sensible au placement des ancres particulièrement aux bords du réseau [LAN 03].

4.6 Comparaison entre les algorithmes distribués et les algorithmes centralisés

En termes de précision, les algorithmes centralisés offrent le plus souvent de meilleurs résultats que les algorithmes distribués. Dans un algorithme centralisé, une vue sur tout le système est disponible. Cependant, ce genre d'algorithmes souffre de problèmes d'évolutivité; leur implémentation n'est généralement pas faisable dans de larges réseaux. En effet, le nombre fulgurant de messages transmis dans ce type d'algorithmes risque d'étouffer le réseau.

D'un autre côté, les algorithmes distribués sont des algorithmes assez difficiles à mettre en œuvre. La propagation des erreurs est un problème très répondu dans de telles approches. De plus, pour arriver à un état stable, un algorithme distribué va souvent nécessiter plusieurs itérations (plusieurs envois de messages).

D'un point de vue consommation d'énergie, chaque nœud dans un algorithme centralisé doit acheminer les données le concernant à la station de base par multi-sauts. Dans le cas des algorithmes distribués, les informations ne sont échangées qu'entre voisins. Plusieurs échanges seront, cependant, requis dans le cas d'une approche distribuée pour obtenir de bons résultats. Dans [RAB 04], les auteurs ont conclu que si dans un algorithme distribué donné, le nombre de sauts moyen qui séparent un nœud de sa base centrale dépasse le nombre d'itérations. Alors cet algorithme sera plus économe en énergie qu'un modèle centralisé.

N'importe quel algorithme distribué peut être adapté à une solution centralisée. Une version distribuée d'un algorithme centralisé peut aussi être envisagée pour certaines applications. **Oh-Heum** et autres, dans [OHH 08], explique les étapes qui permettent de passer d'une solution centralisée à une solution distribuée. Dans un premier temps, tout le réseau sera divisé en régions qui se chevauchent. L'algorithme centralisé sera implémenté en chaque région, ces dernière seront par la suite recousues en utilisant les nœuds qu'ils leurs sont en communs pour reformer ainsi la carte.

4.7 Conclusion

Dans ce chapitre, nous avons présenté en une première partie les critères de comparaison les plus utilisées dans le domaine de la localisation dans les réseaux de capteurs sans fil. Car, la nouvelle technologie donne de nouvelles occasions, mais elle introduit aussi des nouveaux problèmes. Cela est particulièrement vrai pour les réseaux de capteurs où les capacités individuelles d'un capteur sont limitées. Par conséquent, la

collaboration entre les capteurs est exigée, mais les économies d'énergie sont un souci important, qui implique que la communication devrait être réduite au minimum. Ces objectifs contradictoires exigent des solutions sophistiqués pour beaucoup de situations. Dans ce cadre, des comparaisons, selon les caractéristiques des algorithmes de localisation, sont proposés dans ce chapitre.

Comme résultat de travail d'étude effectué sur les différents approches et protocoles de localisation, nous avons donné une classification sur les différentes méthodes lors de chapitre précédent. Et comme suite de cette étude, une nouvelle approche sous la forme d'une nouvelle famille de protocole, est donnée lors de chapitre prochaine.

CHAPITRE 5
LE PROTOCOLE DE LOCALISATION SL-FREE

5.1 Introduction

Les réseaux de capteur sans fil sont une technologie significative attirant l'intérêt d'une marge considérable des recherches. L'intérêt donné à la localisation pour les réseaux de capteur sans fil, est prévu de se développer avec la multiplication des applications de ces réseaux. Bien que beaucoup de systèmes de localisation de réseau aient été récemment proposés et évalués, il reste le terrain de proposition des nouveaux protocoles et démarches de localisation ouvert pour les chercheurs.

L'examen des techniques de localisation dans les réseaux de capteurs sans fil peut être trouvé dans le chapitre 3 de ce mémoire. Le centre des références pour les différentes méthodes, est sur des techniques de localisation dans le réseau cellulaire, les environnements de réseau local (**WLAN**) sans fil, et sur l'aspect de traitement des signaux des techniques de localisation. Les réseaux de capteur varient de manière significative des réseaux cellulaires traditionnels et **WLAN**, et ont des nœuds de capteur qui sont petits, peu coûteux, coopératifs et déployé avec de grande quantité. Ces dispositifs des défis présentent des opportunités actuelles uniques pour la localisation dans les **WSN**.

Le service qui assure la fonctionnalité de localisation, le plus connu actuellement, est celui de **GPS**, et l'approche retenu par le **GPS**, ne conviens pas avec les réseaux de capteurs car le **GPS** exige qu'une infrastructure soit présente (c'est les satellites). Généralement, des solutions se sont développées dans le secteur de robotique, et de calcul ubiquiste, qui ne sont pas applicable dans le contexte des réseaux de capteurs car ils exigent trop de capacité de traitement et d'énergie.

D'après qu'un nombre significatif d'approches s'est développé pour la localisation dans les **WSNs**, ils restent beaucoup de problèmes non résolus dans le secteur. Les défis à adresser sont tous deux dans la caractérisation analytique des réseaux de capteur (de l'aspect de la localisation) et du développement des algorithmes (efficaces) de localisation pour différentes classes des réseaux de capteur et selon une série de conditions. Presque toutes les méthodes produites jusqu'ici conviennent à un réseau dont les participants sont toujours statiques ou mobiles (par exemple **DV-Hop** est destiner aux réseaux statiques,

MCL ne marche pas lorsque les nœuds sont statiques). Nous voudrions concevoir un algorithme qui se comporte bien, indépendamment si un réseau soit toujours statique, toujours mobile, ou peut être statique ou mobile. Nous exigeons de notre algorithme d'être efficaces et implantable sur un réseau homogène dans lequel tous les nœuds ont la même portée radio.

Dans cette partie de travail, nous allons donner une conception d'une nouvelle famille des protocoles qui assure la fonctionnalité de localisation pour les **WSNs**. Nous somme intéressés aux algorithmes vraiment distribués, car ils peuvent être utilisés pour des réseaux de capteur à grande échelle (plus de 100 capteurs). Ces algorithmes doivent être auto-organisés, robustes et gèrent l'énergie efficacement.

5.2 Modèle d'environnement et hypothèses

Avant de parler de notre proposition en détail, nous décrions d'abord le contexte dans lequel ces algorithmes doivent fonctionner.

Une première considération, dans le but de respecter les contraintes des réseaux de capteur, implique qu'il n'y a aucun contrôle fin sur le placement des nœuds de capteur lors de l'installation de réseau (par exemple, quand les nœuds sont jetés d'un avion). En conséquence, nous assurons que les nœuds sont aléatoirement distribués à travers l'environnement. Pour pallier à cette contrainte, les algorithmes de localisation proposés basent sur une approche distribuée.

La connectivité des nœuds dans le réseau (c.-à-d. le nombre moyen de voisins) est un paramètre important qui a un impact fort sur l'exactitude des algorithmes de localisation. Il peut être placé à l'initialisation en choisissant une densité spécifique de nœud, et dans certains cas il peut être placé dynamiquement en ajustant la puissance de transmission radio pour chaque nœud.

Dans certains scénarios d'application de ces réseaux, les nœuds peuvent être mobiles. Dans cette partie de travail, cependant, nous nous concentrons sur les réseaux statiques, où les nœuds ne se déplacent pas. Puisque c'est déjà une condition provocante pour la localisation distribuée. Nous supposons que quelques nœuds appelés ancre ont la connaissance a priori de leur propre position en ce qui concerne un certain système de coordination global. Noter que les nœuds ancres ont les mêmes possibilités (traitement, communication, consommation d'énergie, etc.) comme tous les autres capteurs *unknown*s. Nous nous intéressons dans un premier temps au problème de la localisation dans les

configurations <S, S, z > avec z Є {Ø, dist, angle}. Donc, nous avons un réseau de capteurs statiques (S), et même pour les ancres, sont statiques (S). Les nœuds ont la capacité d'avoir des mesures selon le cas. Si on travail avec un algorithme qui ne base pas sur les mesures (c'est le cas pour le protocole **SL-FREE**), donc le z = Ø, sinon le nœud doit avoir la capacité de calculer la distance (z = dist), ou bien l'angle d'arrivé des messages (z = angle).

Dans le meilleur des cas, la fraction des nœuds ancres devrait être aussi basse comme possible pour réduire au minimum les coûts d'installation. Il est important de se rendre compte que les trois paramètres principaux de contexte (connectivité, fraction d'ancre, et erreurs de gamme) sont dépendants. Des pauvres mesures de gamme peuvent être compensées en utilisant beaucoup d'ancres et/ou une connectivité élevée.

5.3 La famille de méthodes de localisation 'SL-Family' (Self-Localisation Family)

Nous proposons une famille d'algorithme d'auto-localisation dans les réseaux de capteurs statiques basés sur des estimations des distances relatives, appelées **SL-Family** (**S**elf **L**ocalisation **F**amily), et qui assure un passage vers une localisation fine.

L'idée générale du ce type de protocoles de localisation relative est d'attribuer un indice fort aux voisins ayant à la fois un voisinage commun important et un voisinage distinct faible. L'indice de proximité va ainsi évaluer la similarité des voisinages. Ainsi, nous prenons en compte dans l'algorithme le rapport entre le nombre de voisins communs et le nombre de voisins disjoints. En effet, en générale, deux nœuds qui sont proches ont en commun un grand nombre de voisins et peu de voisins non communs. L'intérêt de cette constatation est qu'elle peut nous servir à mieux estimer les distances entre les nœuds. L'indice de proximité va illustrer la distance relative d'un voisin. Pour avoir plus de précision dans les résultats fournis par ces algorithmes, la dernière phase de l'algorithme consiste à assurer le passage vers une localisation fine.

Ces algorithmes doivent être auto-organisés, tel que le nœud ne dépend pas d'une infrastructure existante. La robustesse aussi est demandée pour être tolérant aux échecs de nœud et aux erreurs. De plus ils exigent peu de calcul et, particulièrement, lors de communication.

5.4 Le protocole SL-Free

Dans cette partie de travail, nous allons décrire le premier algorithme de la nouvelle famille de protocoles de localisation **SL-Family**, qui s'appelle **SL-FREE**, car il fait parti de cette famille et il est libre de mesure.

L'algorithme déployé sur un nœud obtient les informations par l'échange de la table de voisinage. Ces tables vont lui servir pour calculer un indice de proximité pour chaque voisin à un saut. Nous supposons que le réseau est de densité non faible et que la portée radio R est identique pour tous les capteurs. Notre réseau est construit de n nœuds parmi eux existe m ancres (donc, $m \leq n$, toujours). Dans le cadre de ce protocole, nous avons fixé le choix de la configuration, en premier lieu, sur un réseau des nœuds statiques et qui ne font pas de calcul de mesures (<S, S, Ø >).

Notre algorithme **SL-FREE** est un algorithme libre-mesure ; son principe est basé sur les deux hypothèses suivantes:

1- En comparant les tables de voisinages des nœuds A et B, on peut conclure sur le degré de rapprochement entre A et B ; plus les nœuds sont proches l'un de l'autre et plus leur tables de voisinages seront similaires.

2- On est sûre que si la distance maximale pouvant séparé A et B est inférieure ou égale à R, A et B étant voisins; donc peuvent communiquer entre eux.

4.1 Phase 1 : Localisation approximative

Lors de la phase 1 de cet algorithme, chaque nœud du réseau estime la distance relative qui le sépare de chacun de ses voisins, pour pouvoir se localiser approximativement dans sa région. Donc, le capteur calcule son indice de proximité par rapport à chacun de ses voisins en fonction des informations reçues de son voisinage à un (1) saut. Avec l'indice de proximité, le capteur calcul la distance relative qui le sépare de ce voisin, en suivant ces étapes :

1. Chaque nœud définit son ensemble de voisinage grâce aux informations locales récoltées, suites à la réception du message '**HELLO**' à travers les liens de communication sans fil.
2. Après écoulement d'un certain temps, chaque nœud diffuse, sur un saut, un message qui contient sa liste de voisinage, construite via les messages reçus avant.

3. Le nœud calcul les distances relatives qui le sépare de chacun de ses voisins, tel que si on a B voisin de A donc la distance relative qui sépare A et B est régie par la formule suivante :

$$d_{AB} = h \times R \qquad h \leq 1 \qquad (1)$$

Où R est le rayon de communication et h un coefficient à déterminer.

La valeur de h dépend de l'indice de rapprochement. Plus l'indice de rapprochement est bas et plus la valeur de h sera proche du zéro. La précision de l'estimation va donc dépendre de la technique utilisée pour trouver h.

Soit la formule suivante, [ARO 10]:

$$\begin{cases} F = \dfrac{|V(A) \cap V(B)|}{\min(|V(A)|, |V(B)|)} \\ h = (1 - F) \end{cases} \qquad (2)$$

V(A) et V(B) sont des ensembles finis. F appartiennent à [0,1].

4. Dans le cas où F=0 ; donc $V(A) \cap V(B) = \phi$, A se situe à une distance très éloignée de B, mais A peut communiquer avec B donc $d_{AB} = R$.
5. Dans le cas où F=1 ; Les positions de A et B sont confondu (A et B se trouve très proches l'un de l'autre).

La valeur de h est le rapport entre le nombre de voisins communs (par le fait de prendre l'intersection entre les ensembles de voisinage) et le nombre de voisins disjoints (par le fait de prendre le minimum entre les cardinalités des ensembles de voisinage) entre les deux capteurs A et B.

Le coefficient h, vas nous permettre d'avoir un indice sur le rapprochement entre les deux nœuds voisin, et qui est représenté par le variable F.

D'après (1) et (2) on obtient:

$$d_{AB} = (1 - \frac{|V(A) \cap V(B)|}{\min(|V(A)|, |V(B)|)}) \times R \qquad (1')$$

Donc, la distance relative, est calculée par une simple multiplication de la valeur du variable h, et le rayon de communication R.

4.2 Phase 2 : Calcul de la taille de l'indice de proximité et des distances réels

A. Nœud ancre

Lors de cette phase, les ancres vont calculer la taille de l'indice de proximité (IndSize).

Le nœud ancre, en plus des distances relatives qu'il a calculé lors de la première phase, a les distances réelles qui le séparent de ses ancres voisines (via un système de coordination globale, comme un **GPS** par exemple). Donc, il peut appliquer la relation triple pour déduire le rapport entre les distances réelles et les distances relatives qu'il a calculé avant. De cette manière, l'ancre fait le rapport entre la distance relative d'un voisin ancre et sa distance réelle pour ce même voisin, pour arriver d'avoir la correspondance entre ces deux informations. Dans le but d'aider les nœuds *unknown*s de bénéficier de cette correspondance, cette information représente la taille de l'indice de proximité (IndSize). Par la suite, il calcule la moyenne de toutes les valeurs d'IndSize, et il la diffuse, ($IndSize_i$), à travers le réseau.

Principe :

Soit A et B deux ancres voisins, si d^r_{AB} est la distance réelle entre ces deux ancres et d_{AB} la distance relative entre A et B calculée lors de la première phase de cet algorithme, la taille de l'indice de proximité est donnée par la relation suivante :

$$\textbf{IndSize} = d^r_{AB} \ / \ d_{AB} \qquad (03)$$

B. Nœud unknown

Un nœud *unknown*, est un nœud qui ne connue pas sa position. Lors de la phase 02, de cet algorithme, il doit faire les opérations suivantes :

1. Lors de réception d'un message de la part d'une ancre sur la taille d'indice de proximité, le nœud *unknown* dois sauvegarder cette information dans une table.
2. Au niveau de chaque capteur, le réseau est implémenté sur une grille, qui est divisée en cases. La valeur de la case, en mémoire, est mise à jour suite à la réception d'un message sur la taille d'indice size, d'une ancre.
3. Si le nœud et l'ancre (l'ancre émettrice de message sur la taille d'indice de proximité) sont voisins, les cases considérées appartiennent à la boite. Une boite (comme utilisé par la méthode **Bounding-Box** [KWO 04]) peut être imaginée comme un rectangle dans lequel une ancre est située au centre, et dont la largeur est égale à deux fois la distance (la distance relative calculée lors de phase 01 du

protocole **SL-FREE**) qui sépare l'ancre du nœud *unknown* en cours de se localiser.

4. Pour chaque message sur la taille d'indice de proximité reçu, une boite est construite, et l'information sur la taille d'indice de proximité est sauvegardée dans une table dédiée.
5. Mais si l'ancre émettrice, du message, n'est pas l'un des voisins de nœud *unknown* en cours, le nœud considère seulement les cases qui sont en dors de la zone d'appartenance d'ancre émettrice du message, et ne sauvegarde pas cette information, sur la taille d'indice de proximité, dans sa table.
6. Le nœud *unknown* calcule la moyenne des informations reçu sur la taille d'indice size, **MoyIndSize** via la formule suivante:

$$\textbf{MoyIndSize} = \frac{\sum_{i \epsilon m} IndSizei}{m} \qquad (04)$$

(m est la taille de la table du sauvegarde)

7. Le nœud *unknown* D calcul son estimation de la distance réelle par rapport à chaque voisin (par exemple le nœud A), via la formule suivante :

$$\mathbf{d^r_{DA} = MoyIndSize * d_{DA}} \qquad (05)$$

d^r_{DA} est la distance réelle entre le nœud *unknown* D et son voisin le nœud A, d_{DA} est la distance relative entre le nœud D et le nœud A calculé lors de la phase01 du protocole **SL-FREE**.

A la fin de la phase 02 de protocole **SL-FREE**, le nœud *unknown*s a une estimation sur les distances réelles qui le sépare de chacun de ses voisins.

4.3 Phase 3 : Localisation fine

Lors de la phase 3, un modèle général d'estimation de la localisation à deux dimensions (2D), en utilisant m ancres, est développé.

Lors de la phase précédente du protocole **SL-FREE**, des zones d'appartenances sont construites au niveau de chaque capteur (sous format des boites). Pour définir les coordonnées réelles de capteurs, une évaluation des intersections entre les différentes zones définies, est faite.

L'intersection des différentes boites, construites à partir des informations en provenance de plusieurs ancres, donnera une zone dans laquelle le nœud *unknown* est sûr d'y appartenir. Le nœud considérera le centre de gravité de cette zone comme estimation de sa position réelle (Voir figure 5.1).

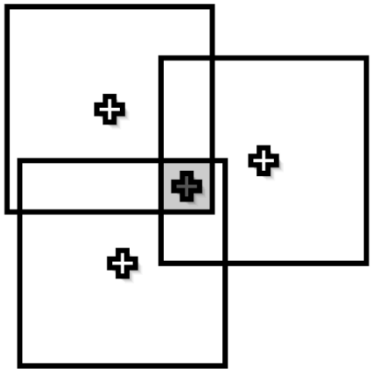


Figure 5. 1 : Principe du positionnemnt par boite.

Donc la position réelle de capteur est le centre de gravité de la zone d'intersection entre les différentes boites construites. Soit (x_i, y_i) les coordonnées réelles de capteur définies par le protocole **SL-FREE**, tel que ils représentent les coordonnées de centre de gravité de la zone d'intersection déduite.

5.5 Points forts du protocole SLFree

5.1 Gestion d'énergie

Le principe de base pour la famille des protocoles **SL-Family**, est peu consommateur en énergie car il ne provoque pas plus de communication à travers le réseau lors de son lancement, par contre il utilise seulement des informations nécessaires à beaucoup d'autres protocoles déployés dans les réseaux de capteurs : auto-organisation et protocoles de routages pro-actifs par exemple. De plus, si le réseau est peu dynamique (faible mobilité, peu de naissance ou de mort de nœuds dans le réseau) ou bien il est statique (c'est le cas pour le protocole **SL-FREE**), cet échange de paquets peut être réduit voire limité simplement à la phase de déploiement du réseau [HEU 08].

5.2 Rapidité et efficacité

Le protocole **SL-FREE** assure une information de localisation de façon rapide et efficace. Tel que, il exploite des informations de connectivités existantes généralement et qui sont disponibles au sein de chaque nœud. Parmi les protocoles qui exigent la

disponibilité de ces données au niveau de chaque capteur, on a le protocole **TRAMA** qui est un protocole de la couche **MAC** et qui travail par l'échange des tables de voisinages entre les capteurs au sein du réseau.

Dans la famille d'algorithmes **SL-Family**, le nœud à partir de la première phase, a une information sur sa localisation qui représente sa position relative. Donc, il peut utiliser cette information pour des services connus dans les **WSNs**, comme le routage géographique. Notons que s'il ne reçoit pas assez d'informations depuis les ancres, il peut continuer de travailler avec cette information de localisation approximative. Ainsi, cette position imprécise peut être utilisée pour construire un protocole de routage unicast efficace dans un environnement sans fil avec un haut niveau d'interférences.

5.3 L'évolutivité

Afin de garantir l'évolutivité de notre algorithme, il est intéressant que les nœuds *unknown*s calculent indépendamment leur propre position en utilisant seulement l'information locale. Dans un autre mot, l'algorithme de localisation devrait être distribué plutôt que centralisé. A partir de la, que le protocole **SL-FREE** est proposé dans une version distribué. Donc pour bénéficies des données de connectivités à travers le réseau, le protocole **SL-FREE** est un protocole collaborative. Car les nœuds à travers le réseau collaborent pour fournir une estimation des positions relatives, et même lors de la phase de localisation fine.

5.4 L'impact de la densité

La démarche de calcul de la distance relative, du protocole **SL-FREE**, assure la prise en compte de la densité dans la région d'existence de nœud à travers le réseau, car le capteur, lors de l'estimation de sa position relative, se base sur les informations de connectivité à un saut seulement. Même lors du passage vers une localisation fine, le capteur bénéficie des informations sur la taille d'indice size venus des ancres dans sa zone. Contrairement aux autres démarches comme pour :

SumDist : dans la technique d'estimation de distance **SumDist**, chaque capteur avant de réémettre le message, ajoute à l'information de distance sur le message, la mesure de distance qui le concerne, et donc le nœud destinataire va diviser cette somme par le nombre de sauts que le massage à parcouru. Dans ce type des mesures les obstacles dans le réseau influencent les calculs, et même une densité élevée pour un capteur donné, va faire

croitre le nombre de sauts malgré que la distance réelle ne soit pas grande, donc les résultats vont avoir une marge d'erreur importante.

DV-Hop [SAA 08]: la distance moyenne calculée par une ancre est influencée par la densité de la région de l'ancre qui a réalisé le calcul. Car lors de la deuxième étape de cette méthode, une fois que l'ancre à reçu le nombre de sauts des autres ancres, elle estime une moyenne de la taille d'un saut, qui est alors diffusée au réseau entier. Après la réception de la taille d'un saut, les nœuds multiplient la taille d'un saut par le nombre de sauts pour estimer la distance physique par rapport à l'ancre. Par ce principe, un nœud éloigné de cette ancre qui a diffusé ce message, et qui fait parti d'une région différente en densité, va appliquer la distance moyenne calculée par cette ancre et donc ses mesures ne seront pas suffisamment fiables selon sa zone d'appartenance.

5.5 La précision des données

Dans un travail de **Nagpal** et autres dans [NAG 03], ils sont démontrés par algorithme qu'encore de meilleures évaluations de distance de « hop count » peuvent être calculées en faisant la moyenne pour la définition des distances avec des voisins. A partir de la, l'algorithme **SL-FREE**, lors de la phase 02, a préféré prendre la moyen des tailles indices reçus des ancres et n'on pas la première information reçu.

5.6 L'Overhearing

L'overhearing est la réception par un nœud d'une trame qui ne lui est pas destinée. L'énergie consommée pour la réception et le traitement des données de cette trame est perdue et sans aucun intérêt. A partir de la, le protocole **SL-FREE** assure le non traitement de ce type de message. Via un mécanisme d'argumentation des messages. Tel que chaque message comporte l'identificateur de nœud émetteur, et lorsqu'il s'agit d'une diffusion, le message comporte aussi un champ sur la liste des nœuds destinataires. Lors de réception d'un message, le protocole **SL-FREE** ne lance pas de traitement, avant de vérifier que le nœud récepteur était destinataire, ou bien je viens d'avoir une nouvelle donnée que je n'ai pas encore reçu de ce voisin.

5.7 Gestion des diffusions

Le protocole **SL-FREE** fait les diffusions suivantes :

- La diffusion du message '**Hello**' dans la portée de la communication de chaque capteur,
- La diffusion de la table de voisinage dans la portée de la communication de chaque capteur,
- La diffusion de la taille d'indice par chaque ancre à travers le réseau, mais avec une information porté sur chaque message et qui concerne la liste des destinataires pour que la diffusion soit gérée.

Dans le but de gérer la diffusion, nous avons utilisé des messages qui comportent la liste des nœuds destinataires, pour assurer la réduction du nombre de réémissions lors de la diffusion de message. Donc nous avons utilisé un algorithme déterministe pour gérer la diffusion.

5.6 Conclusion

La nouvelle technologie donne de nouvelles occasions, mais elle introduit aussi des nouveaux problèmes. Cela est particulièrement vrai pour les réseaux de capteurs sans fil où les capacités individuelles d'un capteur sont limitées. Par conséquent, la collaboration entre les capteurs est exigée, mais les économies d'énergie sont un souci important, qui implique que la communication devrait être réduite au minimum. Ces objectifs contradictoires exigent des solutions sophistiqués pour beaucoup de situations.

La famille de protocole de localisation dans les réseaux de capteurs sans fil, **SL-Familly** favorise une démarche distribué et collaborative. Donc, afin de garantir l'économie d'énergie et l'évolutivité de notre algorithme, nous avons choisi une démarche distribuée, où les nœuds *unknown*s calculent indépendamment leurs propres positions en utilisant seulement l'information locale. Aussi, nous avons utilisé une approche de diffusion concerne les deux phases : la première diffusion est une inondation faites par les nœuds, lors de la première phase du protocole, et la deuxième diffusion est fait par chaque ancre. L'ancre diffuse la taille de l'indice de proximité pour informer les nœuds *unknown*s. Le premier protocole de localisation de cette famille des protocoles est le protocole **SL-FREE**. Il assure une évaluation approximative à la disposition des capteurs, en première phase. De cette manière, une information approximative est disponible au niveau des nœuds. Tel que, les expérimentations ont montré que l'estimation approximative va de paire avec l'estimation approximative géographique dans le cas des réseaux de capteurs dense. Après avoir reçu assez de données à travers les liens de communications dans le réseau, le capteur passe à une localisation fine.

Mettre à la disposition des nœuds une information de localisation de façon rapide et efficace, est une opportunité intéressante que le protocole **SL-FREE** assure par son principe de base. De plus, garantir un passage vers une localisation fine, c'est une deuxième opportunité où le nœud assure d'être bien localisé pour bien bénéficies des données qu'il fourni pour les utilisateurs finaux de réseau de capteur.

CHAPITRE 6
EVALUATION DES PERFORMANCES DU PROTOCOLE SLFREE

6.1 Introduction

Avant la phase de fabrication et de déploiement, un réseau de capteurs nécessite une phase de simulation afin de s'assurer du bon fonctionnement de tous les dispositifs. Quand il s'agit d'un réseau à grande échelle où le nombre de capteurs peut atteindre plusieurs milliers, cela engendre un coût financier relativement important. Il faut donc réduire au maximum les erreurs possibles de conception en procédant par une phase de validation. Cette phase doit tenir compte des contraintes sévères en mémoire, consommation d'énergie et en capacité de traitement des capteurs.

Nous avons effectué une simulation d'intégration de notre protocole sur l'environnement de simulation **NS** (**N**etwork **S**imulator), cet environnement de simulation est largement utilisé dans le domaine académique. Riche en protocoles et en fonctionnalités, il permet de simuler une large gamme de réseaux filaires et sans fil. Étant un logiciel open source et extensible, il offre la possibilité d'implantation de nouveaux protocoles.

Dans ce chapitre, nous décrirons les différents détails liés à l'implémentation, en mettant en relief les structures de données, les algorithmes développés et les différentes phases de méthode. Afin de faciliter la compréhension de notre conception, nous avons choisi de modéliser certaines phases par des organigrammes et des diagrammes **UML**. Nous présenterons par la suite, les différentes métriques de simulations les plus utilisées pour évaluer les performances d'un algorithme de localisation; et nous utiliserons quelques unes de ces métriques pour évaluer notre nouveau protocole de localisation proposé. Nous terminerons par l'analyse des résultats obtenus à l'issue de cette simulation.

6.2 Environnement de Simulation

2.1 La simulation comme méthode de validation

L'évolution des systèmes de télécommunications actuels, vers l'intégration de plusieurs services, l'amélioration de la qualité de service, et le déploiement à grande échelle, a rendu leur étude et leur modélisation très complexes. D'autres parts, ces systèmes doivent être de plus en plus performants pour satisfaire à la fois les avancées technologiques et les demandes grandissantes des opérateurs. Or la complexification des réseaux a un impact inévitable sur les performances. De plus, faire des études analytiques ou expérimentales à des échelles aussi grande est très difficile, voire impossible à réaliser en pratique à cause du coût élevé et de la difficulté de la mise en œuvre. C'est pourquoi les simulations sont considérées dans ce cas comme la méthodologie la plus adéquate pour évaluer les performances et étudier le comportement des infrastructures à grande échelle.

Dans le domaine des réseaux sans fils, la simulation comme méthode de validation possède des atouts incontournables. Elle nous offre la possibilité d'isoler ce qu'on veut et de bien maîtriser les variables de l'environnement. Mais lorsque l'on vise des modèles de plus en plus réaliste et de plus en plus grand, garantir la cohérence des résultats n'est plus très évident pour les simulateurs. En effet, la

complexité des phénomènes physiques qui constituent le canal radio impose de faire un compromis entre la « justesse » du modèle et le coût de la simulation **[BEN 08]**. Ceci est problématique dans la mesure où dans les simulateurs déployés actuellement **NS2**, **GloMoSim** (**G**lobal **M**obile **S**imulator), **JiST/SWANS** (**J**ava **i**n **S**imulation **T**ime / **S**calable **W**ireless **A**d hoc **N**etwork **S**imulator), **GTSNetS** (**G**eorgia **T**ech **Net**work **S**imulator), etc., qui offrent tous des environnements de simulation très développés, mais des choix d'implémentation très différents pour la couche physique.

2.2 Définition

NS2 (**N**etwork **S**imulator **2**) est un environnement de simulation pour les réseaux sans fil et filaires. Il a été conçu à l'université **UC Berkely** (L'université de **Californie** à **Berkeley**). Il a été considérablement utilisé et amélioré au cours des dernières années par de nombreux chercheurs. Le **NS** est un environnement de simulation à base d'évènements discrets, offre la possibilité pour tester, analyser, évaluer et critiquer avant d'envisager l'implémentation concrète dans un réseau. La simulation nous permet donc d'effectuer ces tests à moindres coûts et de prendre des décisions primordiales. Le simulateur **NS** offre un support substantiel pour la simulation des protocoles de contrôle des transmissions (**TCP T**ransmission **C**ontrol **P**rotocol), des protocoles de routage, des protocoles de localisation, et des protocoles de multi casting à travers des réseaux filaires et sans fil (réseaux locaux et satellitaires).

2.3 Description de l'environnement NS

Le simulateur de réseaux **NS** est un simulateur orienté objet, développé sur la base du langage **C++** et du langage de script **OTcl (Object Tcl),** où **Tcl (T**ool **C**ommand **L**anguage**)**, est un script interprété. La combinaison de ces deux langages permet, d'une part, la définition et la création des objets manipulés, ce qui est réalisé efficacement (et rapidement) par le langage **C++,** et d'autres part, la configuration des objets et la gestion des évènements à travers le langage de script **OTcl**. L'interpréteur **OTcl**, dérivé du langage **Tcl**, permet également une interaction avec les utilisateurs en leurs permettant d'écrire des scripts pour la définition du modèle de réseau (le nombre de nœuds, les liens, …etc.), la détermination du trafic dans le réseau (sources, destinations, type de trafic, …etc.), et la définition des protocoles utilisés, et cela, sur la base du langage **Tcl**.

Le simulateur de réseaux **NS2** sous **Linux** est disponible en version « AllInOne » (tous en un) en incluant, ainsi, plusieurs composants dont : **Tcl**, **Otcl**, **Tclcl**, **Xgraph**, **Ns**, **Nam**.

TCL	*Langage de script en open source, utilisé pour programmer NS, il faut l'installer en premier.*
TK	*C'est une interface utilisateur graphique pour TCL, elle s'installe donc juste après.*
OTCL	*Extension de Tcl/Tk pour la programmation Orientée Objet.*
Tclcl	*Interface Tcl/C++, il permet de faire le lien entre la hiérarchie existante en C++ et celle existante en Tcl dans Ns, grâce à ce composant on peut utiliser les 2 langages dans une même simulation de cette façon on obtient un simulateur flexible et puissant.*
NS	*Programme final, Ns a besoin des 4 composants précédents pour fonctionner.*

Nam	*Permet de visualiser graphiquement une simulation.*
X-Graph	*Afficher des graphiques pouvant contenir les instants d'émission des paquets (par exemple).*

Tableau 6.1 Les composants de simulateur NS2.

Le simulateur **NS** contient une double arborescence de classes définie (voir la figure 6.1) :

- en **OTcl** dite arborescence interprétée utilisée par l'interpréteur.
- en **C++** dite compilée, elle forme l'arborescence utilisée par le moteur de simulation.

Les deux arborescences sont très proches l'une de l'autre. Du point de vue de l'utilisateur, il y a une correspondance univoque entre une classe d'une arborescence et une classe de l'autre arborescence [ANE 99].

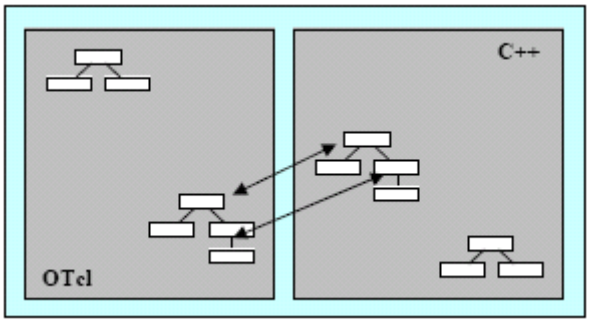


Figure 6. 1 : Dualite de C++ et Otcl dans NS. [ANE 99]

Pour ce qui est de notre simulation, nous avons pu manipuler les deux langages, en effet l'ajout d'un nouveau module dans le cœur de ce logiciel, qui est une tache assez complexe qui nécessite une maîtrise du langage de programmation **C++** ainsi qu'une bonne compréhension des interactions entre les différents composants et classes du simulateur. Quant au langage **Tcl**, nous nous sommes servis du langage **Tcl** aussi bien pour l'écriture des différentes parties du protocole de localisation que pour la définition des paramètres de l'environnement de simulation sur une plate forme **Linux**, en utilisant, pour cette fin, les opportunités offertes par le langage **Tcl**, en l'occurrence, la facilité de manipulation des variables, de définition des procédures, l'utilisation des boucles, …etc.

6.3 Composants de la topologie

Pour définir l'architecture et la topologie du modèle à étudier nous devons tout d'abord présenter les classes de bases utilisables.

3.1 Nœud

La classe Node est une classe **OTcl** : elle n'a donc pas d'existence en tant que telle dans le simulateur. Cette classe et ses méthodes sont définies dans le fichier tcl/lib/ns-node.tcl. Un nœud est une collection de classifiers et d'agents. Le classifier démultiplexe les paquets. L'agent est habituellement l'entité d'un protocole. L'assemblage des composants est représenté par la figure 6.2, [ANE 99] :

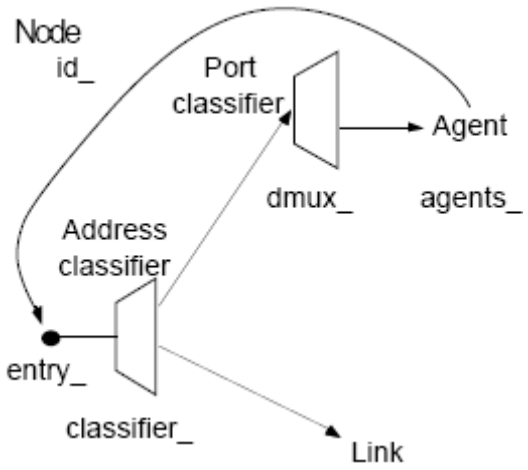


Figure 6. 2 : Composants d'un nœud.

3.2 Lien

Le lien sert à relier les nœuds. Il modélise le système de transmission. Le lien est principalement caractérisé par un délai de propagation et une bande passante. C'est une classe **OTcl** qui regroupe un ensemble de composants dérivés de la classe Connector. Cette classe et ses méthodes sont définies dans le fichier tcl/lib/ns-link.tcl. Quelque soit le type du lien, il comporte 5 références sur des objets qui le composent, selon la figure 6.3, [ANE 99].

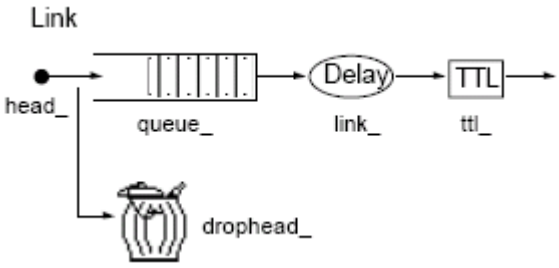


Figure 6. 3 : Composants d'un lien.

3.3 Agent

L'agent est un autre composant d'un nœud. Il modélise les constructeurs et les consommateurs de paquets IP. La classe agent fournit des méthodes utiles au développement de la couche transport de modèle **OSI** (**O**pen **S**ystems **I**nterconnection) et à d'autres protocoles de gestion. Cette classe est à la fois dans l'interpréteur et dans le simulateur. C'est la classe de base pour définir des nouveaux protocoles dans NS. Elle fournit l'adresse locale et de destination, les fonctions pour générer les paquets et l'interface à la classe Application. Actuellement, **NS** comporte de nombreux agents citons: **UDP**, protocoles de routage, différentes versions de **TCP**, **RTP** (**R**eal-time **T**ransfert **P**rotocole), etc. Ils constituent les points terminaux du réseau qui reçoivent ou délivrent les paquets des applications. A chaque agent est attribué un port, tel que l'adresse d'un agent se compose du numéro de son nœud et de son port [ROS 04].

3.4 Les applications

Au dessus de la couche transport se trouve l'application qui va générer le trafic. Il y aura au plus une application par agent.

6.4 Paramétrage de l'Environnement de simulation

4.1 Paramétrage général

Afin de faciliter la compréhension des résultats de simulations, les erreurs de positionnement sont normalisées par rapport au rayon de transmission r. Par exemple, si un nœud est localisé avec une erreur de 50%, cela signifie que la distance entre sa position exacte et sa position estimée est égale à $\frac{r}{2}$.

Les capteurs sont répartis aléatoirement dans une aire $A = 100 \times 100$. Par défaut, le nombre de capteurs n est fixé à 150 et le rayon de transmission r à 14 m. Ce qui représente une densité en capteurs égale à 9.24. La densité d d'un réseau de capteurs représente le nombre moyen de voisins pour un nœud. Elle est obtenue par l'équation suivante [SAA 08] :

$$d = \frac{n\pi r^2}{A}$$

Par la suite, les simulations mettront en évidence l'influence de la densité et du pourcentage d'ancres sur les performances des méthodes. Chaque scénario est exécuté 5 fois avec différents placements des ancres. C'est en général la moyenne des résultats obtenue qui sera prise en compte. Dans notre implémentation la borne d'erreur est de 10%.
Les paramètres utilisés dans notre simulation sont représentés dans le tableau 6.2 :

Modèle de propagation	TowRayGround
Couche MAC	802.11
Antennes	Omnidirectionnelles
Energie initiale	70 joules
Energie de transmission	1,4 Watts
Energie de réception	1,00 Watts
Energie dans l'état Idle	0,7 Watts
Porté de l'antenne radio	14 Mètre

Tableau 6.2 : Les paramètres de simulation.

4.2 Le calcul des coordonnées et de la borne d'erreur

Dans notre implémentation des protocoles, au sein de chaque nœud, le réseau est représenté par une grille (matrice). Une case de cette grille est égale à $(0.01 \times r)$ (avec r le rayon de communication). Lorsqu'un nœud reçoit la position d'une ancre, il incrémente les cases dans la grille qui pourraient correspondre à sa position :

• Si le nœud et l'ancre sont voisins, les cases considérées appartiennent au disque centré en l'ancre et de rayon r.

• Si le nœud et l'ancre ne sont pas voisins, les cases incrémentées se situent entre les deux cercles centré en l'ancre et de rayons respectifs d_1 et d_2 tels que :

$$d_1 = r \quad \text{et} \quad d_2 = r \times h$$

h est le coefficient entre l'ancre et le nœud selon le protocole implémenté, tel que pour **AT-Free** est le nombre de sauts minimums entre le nœud et l'ancre, et pour le protocole **SL-Free** est l'indice de proximité.

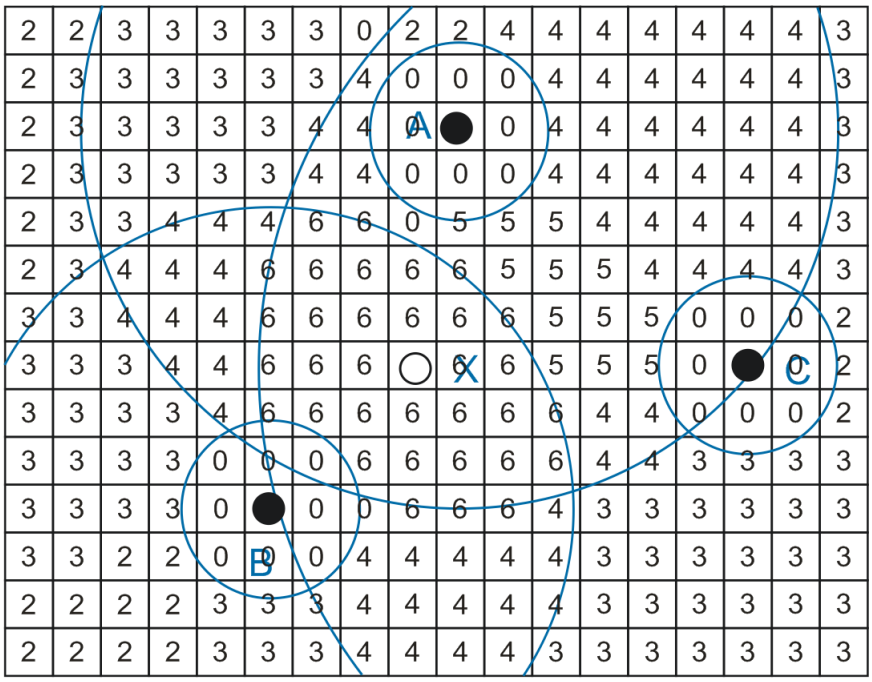


Figure 6. 4 : Representation du reseau par une grille.

Lors de l'implémentation du protocole, nous avons utilisé la grille d'implémentation pour localiser la zone d'appartenance de chaque nœud. La figure 6.4 est un exemple de grille : lorsque le nœud X reçoit l'information de l'ancre A (resp B, C), il incrémente toutes les cases entre les deux cercles centrées en A (resp B, C) et de rayons respectifs r et (r x h_A) (resp (r x h_B), (r x h_C)) avec h_A (resp. h_B, h_C) est le coefficient de protocole implémenté. La zone contenant le nœud X est représentée par les cases de valeurs maximales. Dans la figure, cette zone est constituée des cases égales à 6. X calcule le centre de gravité de cette zone et obtient une estimation de sa position.

4.3 Intégration

Nous avons implémenté notre protocole '**SL-Free'** sous forme de classe « **SLFree** ». En plus de l'implémentation de deux autres protocoles utilisés pour faire des comparaisons, celles de protocole '**AT-Free'** et '**HT-Refine'**, qui vont être implémenté avec les classes « **ATFree** » et « **HTRefine** ». Ces protocoles sont dérivés de la classe « Agent ». Les figures 6.5 et 6.6 montrent l'hiérarchie des classes dans **NS** et indiquent l'emplacement des protocoles implémentés [ANE 99].

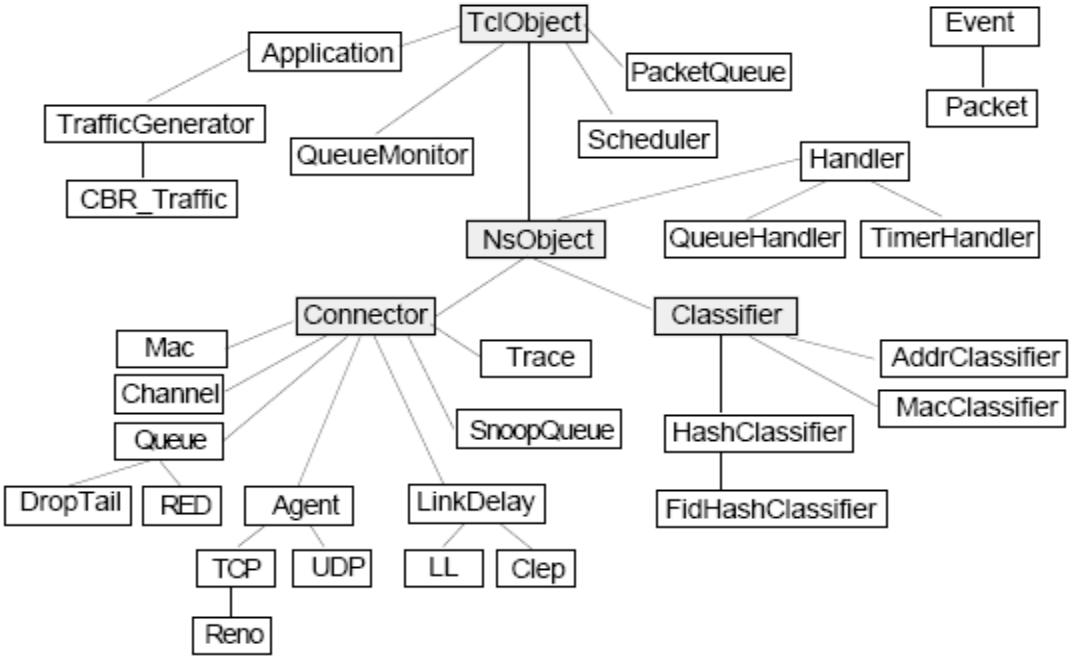


Figure 6. 5 : Arborescence de derivation des classes c++.

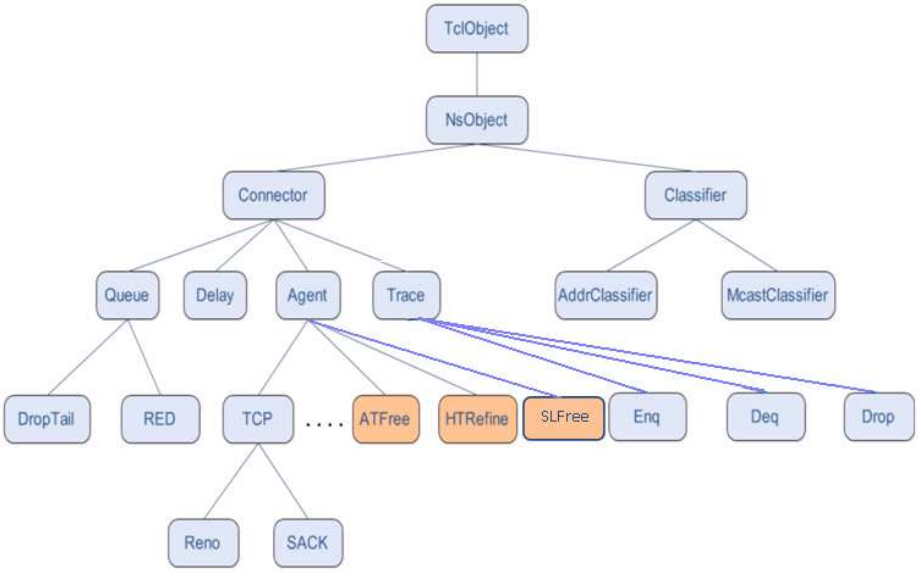


Figure 6. 6 : L'emplacement des protocoles dans ns.

6.5 Implémentation du Protocole SL-Free DANS NS

L'environnement de simulation **NS** est constitué de deux hiérarchies : compilée et interprétée. Ce simulateur est riche en outils et en protocoles ; ceci dit le protocole **SLFREE** n'est pas présents dans **NS**. Il est donc nécessaire de l'incorporer.

Lors d'introduction d'un nouveau protocole, à l'outil **NS**, nous devons d'abord programmer le protocole par le biais du langage **C++**. Une fois **NS** compilé, les simulations peuvent être effectuées en utilisant des scripts **OTcl** [ROS 04].

SLFREE est un protocole qui assure la fonctionnalité de localisation d'un nœud dans un réseau de capteurs sans fils. Nous présentons dans ce qui suit un nouveau module pour **NS** capable de simuler le comportement de protocole **SLFREE**.

La vérification de la performance du protocole s'est réalisée par une application « .tcl », tandis que les **API** (**A**pplication **P**rogramming **I**nterface) suivants sont nécessaires pour introduire le nouveau protocole **SLFree** en **NS** :

- SLFree.h : le fichier header **C++** qui définit les différentes classes et structures de données.
- SLFree.cc : le fichier source **C++** qui contient les différentes fonctions membres nécessaires pour traduire le protocole proposé.
- ns-SLFree.tcl : la librairie **OTcl** nécessaire pour définir les différentes fonctions de l'objet **tcl**.

Afin de poursuivre la définition du nouveau protocole en **NS** et de déterminer les valeurs par défaut de ses différents paramètres, quelques fichiers doivent être modifiés comme : ns-lib.tcl, ns-packet.tcl, ns-default.tcl, packet.h, makefile.in [BEN 05]. Ces modifications sont données en annexe. Après toutes ces modifications sur les fichiers de répertoire du simulateur, il faut lancer la recompilation de **NS** pour qu'il prenne en considération les changements effectués.

6.6 Evaluation du protocole SL-Free

6.1 Métriques d'évaluation des algorithmes de localisation

La simulation est une solution plus aisée pour valider les performances d'un algorithme de localisation; elle permet aux chercheurs de tester les performances d'un algorithme en se basant sur des critères tels que les erreurs de mesure, le nombre de messages transmis, l'énergie consommée...etc. A partir de là, l'évaluation et la comparaison des algorithmes de localisation peuvent être faites en utilisant diverses métriques. Afin dévaluer les performances des algorithmes de localisation sur la base des critères quantitatifs décrits dans les sections précédentes, nous abordons ici les métriques les plus communément utilisées.

6.2 Métriques de précision

L'objectif de ces métriques est de présenter le degré de rapprochement entre les positions estimées et les positions réelles. Les métriques de précision sont séparées en deux groupes : celles qui utilisent les coordonnées réelles des nœuds et celles qui ne le font pas.

- ***Métriques avec connaissance des coordonnées réelles***

Cette catégorie est destinée à des fins de simulation car elle présuppose la connaissance de l'emplacement exacte de chaque nœud.

Erreur Absolue Moyenne (MAE) :

Le moyen le plus simple pour évaluer les performances d'un algorithme est de déterminer l'erreur résiduelle entre les coordonnées estimées et les coordonnées réelles pour chaque nœud et de déduire la moyenne de ces résultats. Les auteurs de [BRO 06] définissent la **MAE** par le biais de la formule suivante :

$$MAE = \frac{\sum_{i=1}^{n} \sqrt{\left(x_i - \hat{x}_i\right)^2 + \left(y_i - \hat{y}_i\right)^2 + \left(z_i - \hat{z}_i\right)^2}}{n}$$

$(\hat{x}, \hat{y}, \hat{z})$ Sont les coordonnées estimées et (x, y, z) sont coordonnées réelles. La métrique résultante est la moyenne entre tous les résidus.

D'après les auteurs de [SLI 02**],** il serait aussi bénéfique de connaître l'erreur maximale entre tous les résidus:

$$\text{MAX\_ERROR} = \max_{i=1..n} \sqrt{\left(x_i - \hat{x}_i\right)^2 + \left(y_i - \hat{y}_i\right)^2 + \left(z_i - \hat{z}_i\right)^2}$$

FROB :

Les auteurs de **[EFR 06]** font usage de la métrique **FROB** (Frobenius). On suppose que les distances qui séparent les nœuds sont connues et celles estimées ont déjà été calculées. La métrique est présentée ci-dessous :

$$\text{FROB} = \sqrt{\frac{1}{n^2} \sum_{i=1}^{n} \sum_{j=1}^{n} \left(\hat{d}_{ij} - d_{ij}\right)^2}$$

$\hat{d}_{ij}$ et d_{ij} sont respectivement la distance calculée en utilisant les coordonnées estimées et la distance calculée en utilisant les coordonnées réelles ; n est le nombre de nœuds dans le réseau.

GER et GDE :

Proposée par **Priyantha** et autres dans [PRI 03], la **GER** (**G**lobal **E**nergy **R**atio) est régit par la formule suivante:

$$\text{GER} = \frac{1}{n(n-1)/2} \sqrt{\sum_{i=1}^{n} \sum_{j=i+1}^{n} \left(\frac{\hat{d}_{ij} - d_{ij}}{d_{ij}}\right)^2}$$

L'erreur entre la distance réelle et la distance estimée ($\hat{d}_{ij} - d_{ij}$) est normalisée par la distance réelle (d_{ij}). On peut remarquer la similitude entre **GER** et **FROB** (**FROB** ne normalise pas les distances).

Une variante de la **GER** a été proposée par les auteurs de [AHM 05]. Elle répond au nom de l'Erreur des Distances Globale (**GDE**):

$$\text{GDE} = \frac{1}{R} \sqrt{\frac{\sum_{i=1}^{n} \sum_{j=i+1}^{n} \left(\frac{\hat{d}_{ij} - d_{ij}}{d_{ij}}\right)^2}{n(n-1)/2}}$$

Où R représente la portée de la radio. Les résultats de la localisation seront donc normalisés par le rayon de communication.

ARD :

Gotsman et **Koren** [GOT 05] définissent une métrique qu'ils appellent: moyenne des écarts-type relatifs **ARD** (**A**verage **R**elative **D**eviation).

$$ARD = \frac{2}{n(n-1)} \sum i<j \frac{\left|\hat{d}_{ij} - d_{ij}\right|}{\min\left(\hat{d}_{ij}, d_{ij}\right)}$$

La différence entre la distance estimée et la distance réelle est normalisée dans ce cas par la plus courte des deux.

- ***Métriques sans connaissance des coordonnées réelles***

Les métriques de précision vues précédemment supposent une connaissance des positions réelles des capteurs. Cette information n'est malheureusement pas connue dans le cas d'un déploiement réel ; l'erreur doit être déterminée à partir des informations disponibles seulement. Par exemple, dans un algorithme basé mesure, un nœud A va obtenir la distance qui le sépare d'un nœud B de deux façons différentes : la première en utilisant les technologies de calcul des distances (**RSSI, ToA**, **TDoA**, **AoA**) et la deuxième en calculant la distance euclidienne à partir des positions estimées des nœuds **A** et **B** ; l'erreur sera obtenue en comparant les deux distances. **Girod [GIR 05]** définie cette métrique par:

$$Erreur = \frac{2}{n(n-1)} \sum_{i,j,i<j} R_{ij} - \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2}$$

R_{ij} est la distance obtenue par les technologies de mesure, (x_i, y_i, z_i) sont les coordonnées estimées du nœud i.

Bassaran [BAS 07] définissent une métrique proche de **FROB** nommée **SPFROB** (shorts path **FROB**) ; cette métrique utilise le plus court chemin entre les nœuds. **SPFROB** est potentiellement utile dans des algorithmes multi sauts tels que **HT-Refine** qui utilise le plus court chemin pour déduire la position qui sépare un nœud d'une ancre.

6.3 Métrique du coût

Comme discuté précédemment, le coût peut être privilégié au détriment de la précision. Les différentes métriques de coût sont présentées ci-dessous :

- ***Pourcentage des ancres***

Il est désirable de minimiser le nombre d'ancres. Leur utilisation peut être très contraignante. Rappelons qu'une ancre peut déduire sa position dans le cas où elle n'est pas placée manuellement en faisant appel à un système de localisation par satellites. Ces systèmes sont connus pour être chers et gourmands en termes d'énergie (minimisent la durée de vie des ancres).

Le pourcentage d'ancres est représenté par le nombre d'ancres divisé par le nombre total de nœuds dans le réseau (ancres + *unkowns*). On essayera d'évaluer la précision d'un algorithme par rapport au pourcentage d'ancres utilisées. Dans les algorithmes de localisation basés ancres, on doit aussi prendre en considération leur emplacement dans le réseau.

Malgré un pourcentage d'ancres réduit, des résultats précis sont obtenus dans des réseaux à forte densité (une moyenne de 15 voisins par nœud) [NAG 03]. Il a aussi été vérifié que la précision d'un algorithme dépend de la méthode de mesure des distances utilisée.

- ***Communication***

Comme discuté précédemment l'envoi et la réception de messages est une procédure couteuse. On doit minimiser le nombre de messages transmis pour maximiser la durée de vie du réseau. Cette métrique sera évaluée de deux manières: en calculant l'énergie consommée ou en comptant le nombre de paquets transmis. Dans le cas où la taille d'un paquet est connue, l'énergie consommée sera déduite du nombre de paquets transmis [LAN 03].

- ***Temps de convergence***

Le temps pris par les nœuds pour rassembler les mesures initiales et celui pris pour que l'algorithme converge, fournissent des métriques de comparaison importantes. Ces métriques vont être évaluées par rapport à la taille du réseau. Un réseau qui prend trop de temps pour se localiser n'est d'aucun intérêt si l'application exige un déploiement rapide et un traitement immédiat des données liées à la position.

6.4 Métriques Hybrides

L'idée derrière ce type de métriques est de combiner plusieurs métriques en une seule. La méthode avec laquelle ces métriques sont combinées varie d'une métrique hybride à une autre. On peut mentionner comme exemple la métrique du coût de performance [AHM 05].

Métrique du coût de performance (PCM) :

C'est une simple métrique hybride où le coût de performance C et l'erreur de localisation **GDE** sont pondérés par un paramètre α, comme illustré dans la formule suivante :

$$PCM = \alpha(GDE) + (1-\alpha)C$$

Le paramètre α dépend de l'évaluation qu'on veut effectuer. On peut choisir quelle sera la variable à privilégier. Ici, **GDE** est une variante de la métrique de précision **GER** comme décrite dans la section précédente. La valeur C quant à elle est la moyenne de l'énergie requise pour qu'un nœud puisse compléter le processus de localisation.

6.5 Métriques de couverture

Quelques algorithmes de localisation peuvent ne pas être capables de localiser tous les nœuds. La couverture définit simplement le pourcentage de nœuds qui peuvent être localisés par l'algorithme sans se soucier de la précision (métrique qui a déjà été décrite). La densité ainsi que le placement des ancres peuvent avoir des effets sur la couverture.

- ***Densité***

Dans certains algorithmes, afin qu'un nœud puisse se localiser, il doit avoir obtenu suffisamment de mesures de la part de ses voisins ; ce genre de conditions aura un impact sur la couverture**.** Si la densité de déploiement est faible, il peut être impossible de localiser certains nœuds.

La figure 6.7 montre la relation qu'il y a entre la densité, le nombre d'ancres et l'erreur de positionnement dans un algorithme multi-sauts [BAS 08]. Lorsque la densité augmente, le nombre de

voisins moyen augmente aussi. Un nœud recevra de ce fait plus de données, chose qui peut potentiellement améliorer les performances de localisation. Cela dit, l'augmentation de la densité aura aussi pour effet une augmentation du trafic, il y aura donc plus de collisions.

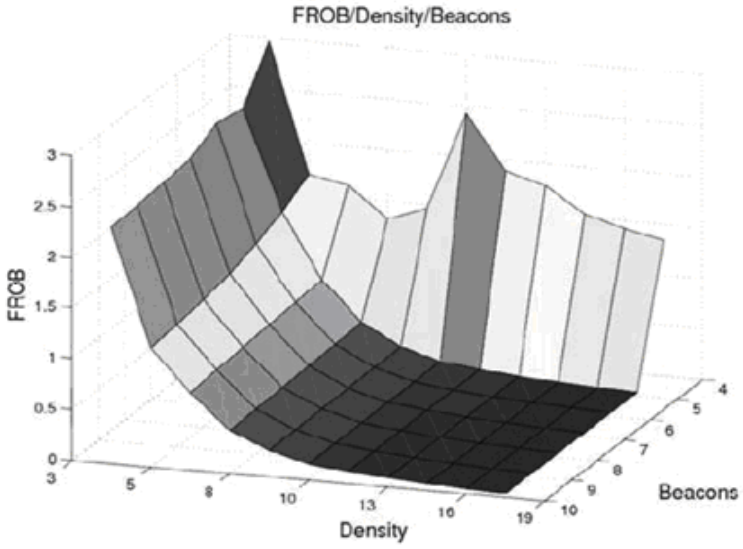


Figure 6.7 : Metrique FROB en fonction de la densité et du pourcentage de nombres d'ancres. [BAS 08]

- ***Emplacement des ancres***

La position des ancres dans le réseau peut avoir un impact considérable sur le processus de localisation, particulièrement si l'algorithme considère que les ancres ont des coordonnées fixes. Les hypothèses sur le placement des ancres ne prennent pas en considération le facteur environnemental, le terrain, et les conditions de propagation du signal. La disposition des ancres par rapport à un nœud unknown affecte la précision des résultats.

Il a été observé dans les systèmes **GPS** que la précision décroit quant les satellites **GPS** se trouvent proche l'un de l'autre. La métrique **GDOP** (***G**eometric **D**ilution **O**f **P**recision*) est utilisée dans les systèmes de localisation par satellites. Selon la forme géométrique formée par les satellites avec la cible, la **GDOP** indique si la qualité du résultat est bonne ou mauvaise. Les auteurs de [SAV 05] ont utilisé le principe de la **GDOP** dans les réseaux de capteurs sans fil. Ils en ont conclu que le placement des ancres dans une enveloppe convexe est la meilleure configuration pour obtenir de bons résultats (Figure 6.8).

Des ancres mobiles peuvent aussi être utilisées pour offrir un supplément de couverture ou pour réduire le nombre d'ancres nécessaires.

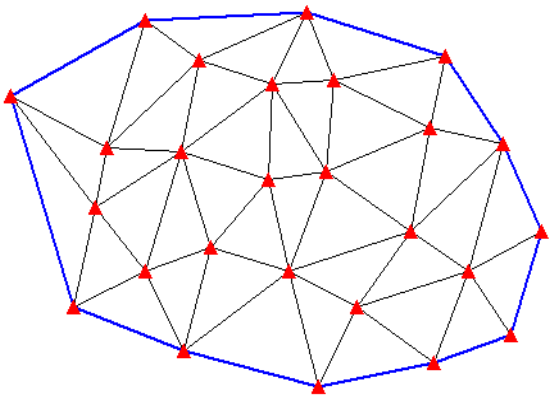


Figure 6.8 : L'enveloppe convexe est représentée en bleu.

- ***Évaluation par rapport à la couverture***

Les chercheurs doivent être préparés à essayer pleins de scénarios dans lesquelles les ancres et les nœuds seront placés de différentes manières, ainsi qu'à essayer différentes densités. Les auteurs de [BUL 02] ont noté que l'augmentation du nombre d'ancres ne garantit pas forcement une meilleure précision ou une meilleure couverture. Ils ont défini un point qu'ils ont appelé « point de saturation» : aucun gain supplémentaire sur la précision ne se ferra après avoir atteint ce point. Par conséquent, on doit non seulement trouver le nombre minimal d'ancres pour que l'algorithme donne de bons résultats mais aussi le point à partir duquel l'ajout d'autres ancres n'améliorera pas la précision.

6.6 Discussion

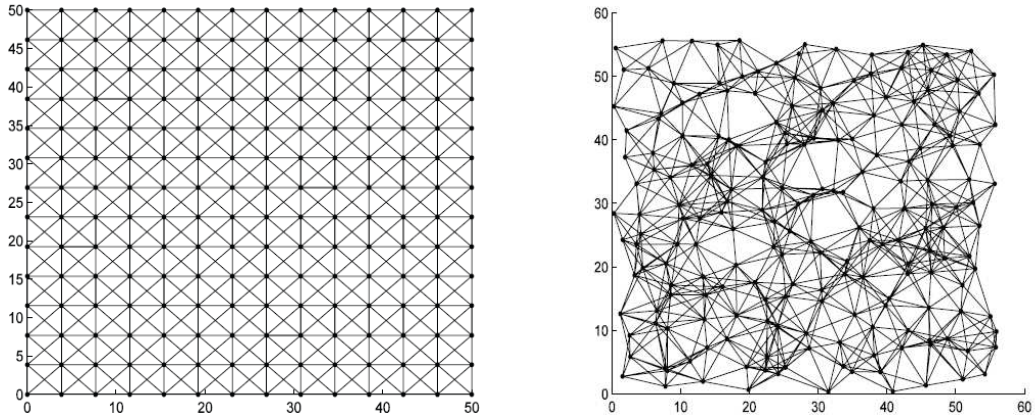
La précision, le coût et la couverture sont les trois critères qui permettent d'évaluer un algorithme de localisation. Une bonne compréhension de ces critères est importante. Le déploiement d'un réseau avec un grand nombre de nœuds est couteux et requiert un placement soigneux pour garantir une bonne couverture. En essayant de minimiser ou de supprimer les ancres, un algorithme de localisation peut compromettre sa précision ; les algorithmes non basés ancres sont généralement centralisés. Le coût des algorithmes peut prendre plusieurs formes et dépend des besoins de l'application. Les métriques hybrides peuvent être utiles si plusieurs métriques doivent être analysées simultanément.

La première étape pour évaluer un algorithme de localisation est de choisir un ensemble parmi les métriques présentées précédemment et d'appliquer cet ensemble à la simulation.

6.7 Topologies

Les topologies dans les réseaux de capteurs sans fil peuvent être classées en deux catégories: *ordonnée* et *aléatoire*. La distribution des nœuds dans une topologie ordonnée est faite uniformément alors que dans une topologie aléatoire cette distribution ne suit pas une loi bien définie : les nœuds sont dispersés aléatoirement. La figure 6.9 montre un exemple de chaque catégorie. Les topologies uniformes ont l'avantage d'être simples à visualiser. Les topologies aléatoires, quant à elles, reflètent mieux la réalité : il est souvent présumé que les nœuds seront déployés à partir d'un véhicule en marche. Un placement uniforme dans ce genre de situation ne peut pas toujours être garanti. Les

topologies aléatoires sont donc généralement préférables pour effectuer des tests sur un algorithme de localisation. Les catégories citées auparavant peuvent encore être subdivisées en deux sous catégories (*régulière* et *irrégulière*) selon la forme de la zone de déploiement.



(A) ***(B)***

Figure 6.9 : Exemple de topologies ordonnée et aléatoire.

A. Topologies régulières

Dans les topologies régulières (comme celles montrées sur la Figure 6.9), les nœuds sont distribués sur une zone qui a la forme d'une grille. Les algorithmes de localisation multi-saut basés vecteur de distance offrent de bons résultats quand ils sont évalués dans ce genre de topologies.

B. Topologies irrégulières

Dans ce genre de topologies, la distance dérivée à partir du plus court chemin qui sépare deux nœuds peut largement différer de la distance réelle entre ces nœuds. Comme résultat, on aura une moyenne d'erreur beaucoup plus importante que dans le cas d'une topologie régulière. Les topologies en forme de C, en forme de L et en forme d'anneau sont des exemples de topologies irrégulières. Des exemples de topologies en forme de C sont représentés sur la figure 6.10.

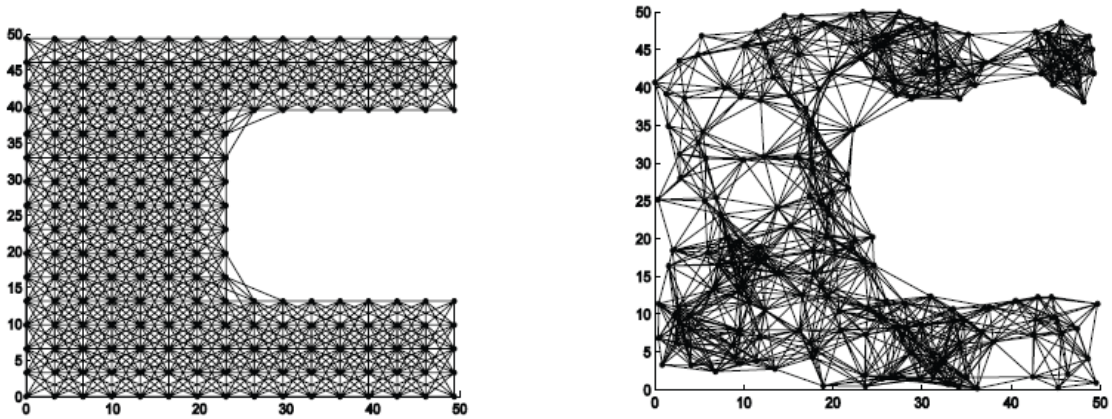


Figure 6.10 : Exemple de topologies en forme de C.

6.8 Calcul de la précision de l'estimation

La précision de l'estimation d'une position est exprimée sous la forme d'un coefficient compris entre 0 et 1. Les nœuds possédant un **GPS** ont une position absolue et ont donc un coefficient égal à 1. Au contraire, les nœuds qui n'ont pas de position ont une précision nulle. Pour les autres nœuds, deux méthodes permettent de calculer la précision de leur position estimée [SIL 11].

6.9 Calcul de la précision avec la distance euclidienne

Cette méthode consiste à comparer la position réelle d'un nœud avec sa position calculée en utilisant la distance euclidienne.

$$\Delta = \sqrt{(x_{calc} - x_{reelle})^2 + (y_{calc} - y_{reelle})^2 + (z_{calc} - z_{reelle})^2}$$

$$C_{acc} = 1 - \frac{\min(R_{\max}, \Delta)}{R_{\max}}$$

Le coefficient de précision est alors :

Où R_{max} est le rayon de couverture maximale.

Si la distance entre la position estimée et la position réelle est supérieure au rayon de couverture maximale, l'estimation est incohérente et la précision est nulle (si $C_{acc} > R_{max}$ alors précision = 0).

6.10 Calcul de la précision avec l'aire de recouvrement

Cette méthode repose sur l'aire de recouvrement des couvertures radio. Cette dernière est formée par le recoupement de toutes les couvertures radios des voisins du nœud considéré. Ainsi, la précision de l'estimation (C_{acc}) va dépendre étroitement de la taille de l'aire de recouvrement, plus celle-ci sera petite plus la précision sera bonne et inversement.

$$C_{acc} = 1 - \frac{A_{Recouvrement}}{A_{Radio}}$$

Où $A_{Recouvrement}$ est l'aire de recouvrement des couvertures radio des voisins et A_{Radio} l'aire de recouvrement radio d'un nœud.

6.7 Résultats et interprétation

Le protocole **SL-FREE** passe par trois phases pour arriver à assurer une localisation fine des capteurs à travers le réseau. Les informations de connectivité locale, disponibles au niveau de chaque capteur, lui permettent d'avoir une localisation approximative, lors de la première phase du protocole. Après un certain temps de récolte des données, les capteurs auront plus de données de précision, ils font des calculs pour pouvoir passer à une localisation fine, lors de la dernière phase de protocole.

Lors de l'implémentation de notre protocole, nous avons écris une procédure qui respecte la conception donnée. Pour évaluer les résultats de calcul des positions des capteurs en basant sur le

protocole **SL-Free**, nous avons fait sortir des résultats intermédiaires d'implémentation de chaque phase.

7.1 La localisation approximative

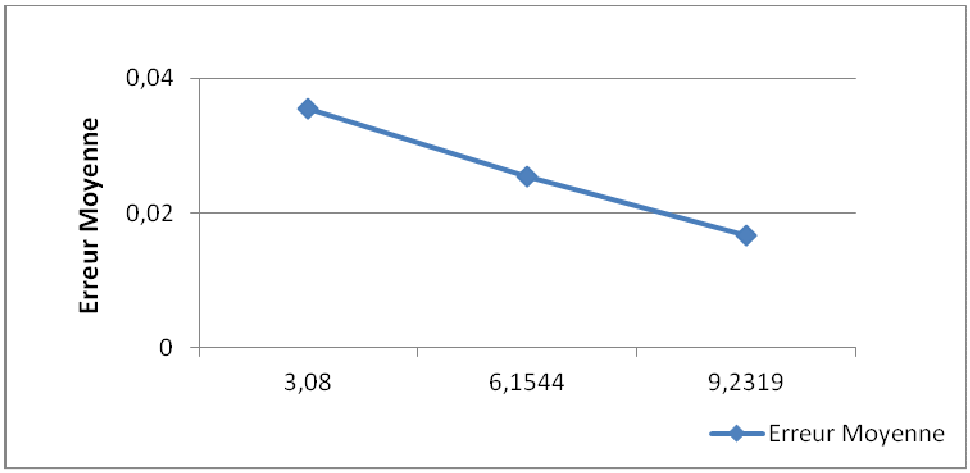
Les expérimentations faites sur la localisation approximative nous ont permis d'évaluer le protocole **SL-FREE** sur les localisations approximatives données par la première phase.

7.2 Impact de pourcentage d'ancres sur l'erreur moyenne

La densité en ancre, n'a pas d'influence sur les résultats de la localisation approximative, car le type de capteur ne demande pas des traitements différents lors la première phase de protocole **SL-Free**. Nous avons réalisé ces expérimentations avec un pourcentage d'ancre fixé à 20 %.

7.3 Impact de la densité dans le réseau sur l'erreur moyenne

La phase de calcul de la localisation approximative est liée à la densité dans le réseau, car elle se base sur la récolte de l'information de connectivité de nœud.



Graphe 6.1 : Impact de la densité sur l'erreur moyenne

Le graphe 6.1 montre l'impact de la densité sur l'erreur moyenne. Le calcul de la densité se base sur trois paramètres du réseau, qui sont la surface, la portée radio de nœud et le nombre de nœuds dans le réseau. Dans cette étape d'expérimentations, nous avons fixé la surface de réseau à 100 m^2 et la porté radio à 14 m, et nous avons varié le nombre de nœuds pour illustrer l'influence de ce dernière paramètre sur la précision des calculs. Tel que, nous avons commencé par un réseau à faible densité (n = 50). Les nœuds ont été rajoutés avec des coordonnées aléatoires, jusqu'à arrivé à un réseau avec une densité importante (n = 150).

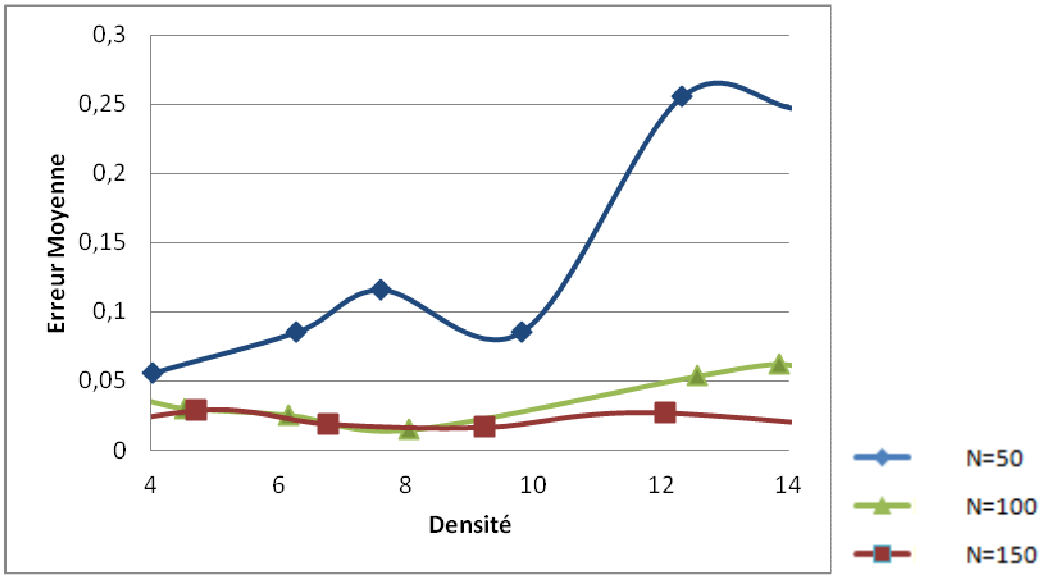
Le graphe 6.1 montre que l'erreur moyenne diminue de façon stable, en fonction de l'augmentation de la densité en nombre de nœuds dans le réseau. La variation de la densité est en relation avec le nombre de capteurs dans le réseau. L'accroissement de la densité ajoute des contraintes de voisinages; un nœud recevra donc plus d'informations de connectivité, et son calcul devient plus précis.

La précision des résultats données par le protocole **SL-Free** sont comparables à celui données par les protocoles **HT-Refine** et **AT-Free**. Tel que, la précision pour un réseau avec une densité de

150 sont très proches des résultats de protocole **AT-Free**, et ils dépassent largement les résultats de **HT-Refine**.

7.4 Impact de la porté radio sur l'erreur moyenne

Dans le but d'évaluer l'influence de la portée radio sur la précision du calcul de la position lors de la localisation approximative, nous avons fait des expérimentations en variant la densité de nœud selon son porté radio**.** Pour évaluer cette métrique, on doit fixer un choix sur le nombre de nœuds dans le réseau (50, 100 ou bien 150), et même sur la surface, qui est fixée à 100 M^2. Donc, nous avons trois graphes mais chaque graphe montre l'évolution de l'erreur moyenne en augmentant la portée radio de nœud (selon le graphe 6.2) pour un nombre de nœuds fixe.



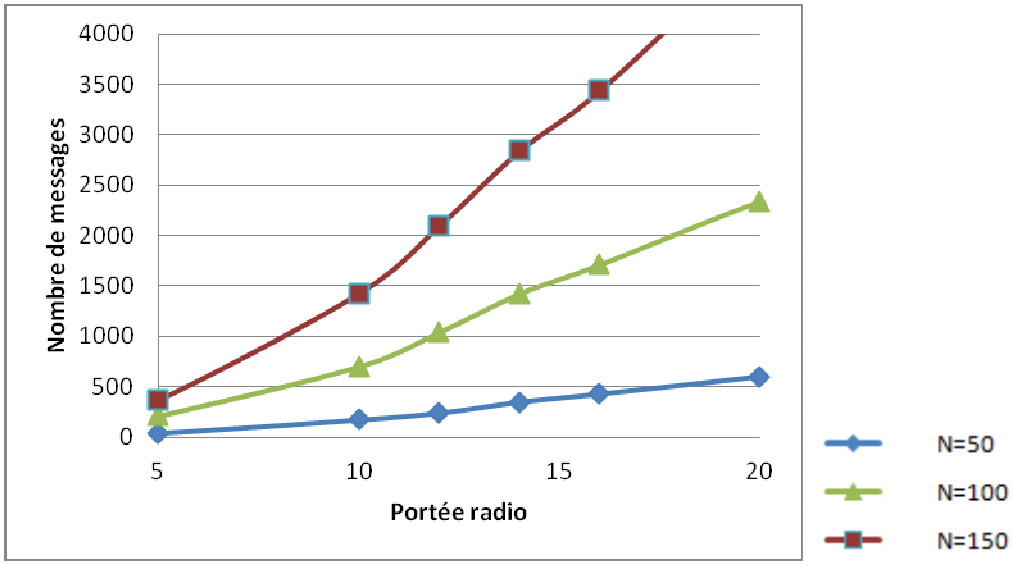
Graphe 6.2 : Impact de la porte radio sur l'erreur moyenne

Le graphe 6.2 est composé de trois graphes, qui représentent respectivement le cas d'un réseau à 50 nœuds, 100 nœuds ou bien 150 nœuds. Le principe de travail de protocole **SL-Free** se base sur l'information de connectivité seulement, donc plus on a d'information disponible plus le calcul est précis (comme montre la parti précédente d'évaluation), avec un nombre réduit des nœuds on constate une différence sur le degré d'instabilité comme démontre les trois graphes. A fin de focaliser notre intérêt sur l'impact du porté radio, nous allons fixer nos remarques sur le troisième cas (pour n = 150), car il est plus stable par rapport à la densité en nombre de nœuds dans le réseau.

Ce graphe montre qu'il y'a une certain diminution sur l'erreur moyenne en augmentant en densité selon la porté radio de nœud. Ces résultats montrent encore les remarques de la section précédente, et qui preuve l'influence de la densité sur l'erreur moyenne des calculs. Donc, augmenter en porté radio assure une zone plus large de voisinage pour chaque noud, et plus d'information sur le voisinage, et qui va raffiner les calculs des positions.

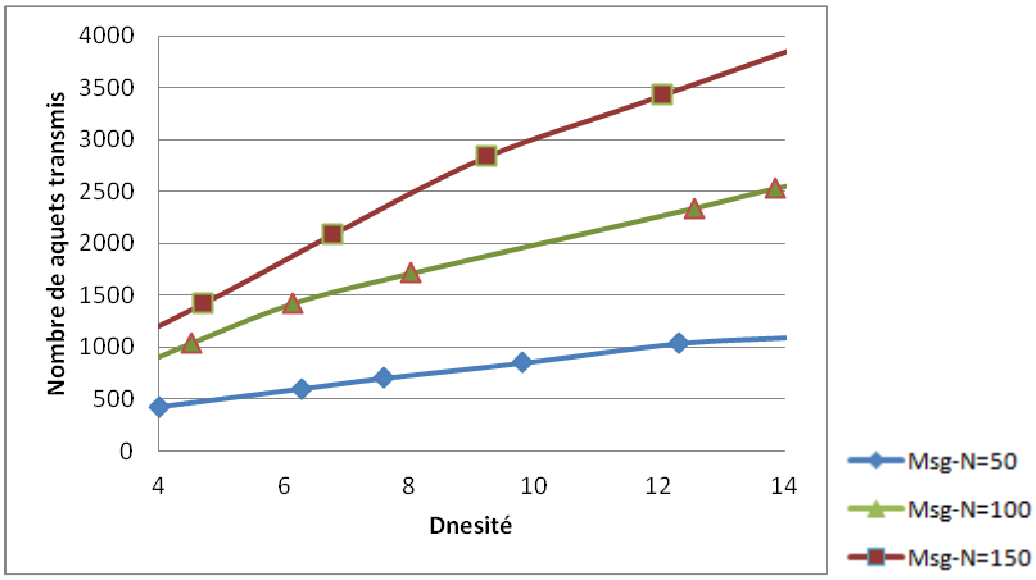
7.5 Impact de la densité sur le nombre de paquets transmis

Le graphe 6.3 montre l'impact du porté radio sur le trafic lors de la localisation approximative du protocole **SL-FREE**. Plus la portée radio augmente plus le nombre de paquets augmente aussi; ce qui est tout à fait logique, car on définit le trafic comme étant le nombre de paquets transmis.



Graphe 6.3 : Impact de la portee radio sur le trafic.

Le graphe 6.4 montre l'impact de la densité sur le trafic lors de la localisation approximative de protocole **SL-FREE**. Le graphe 6.3 ressemble au graphe 6.4, car il représente un cas particulier de ce dernier, puisque la densité est calculée en fonction du porté radio et donc elle suit le même changement.



Graphe 6.4 : Impact de la densite sur le trafic.

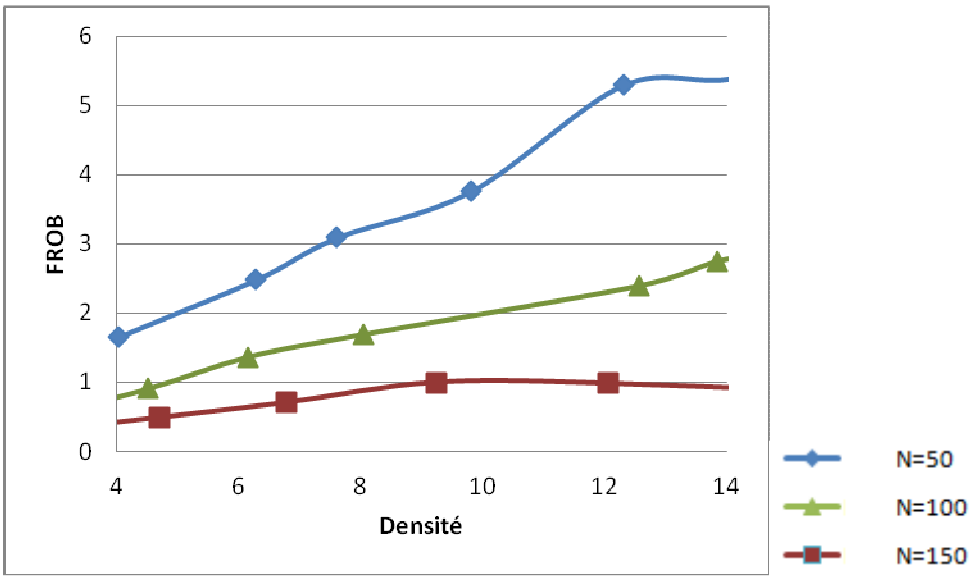
Plus la densité augmente plus le trafic généré dans le réseau est important, car avec une densité plus grande, le capteur va avoir plus de voisins et il va procéder à des échanges de données de connectivité avec ces derniers.

Si on compare ces résultats avec celles données par le protocole **HT-Refine**, nous allons constater que le protocole **SL-FREE** est beaucoup plus intéressant que le protocole **HT-Refine** sur ce point par exemple, car le trafic engendré est beaucoup moins important. Par contre ces résultats sont proches de ceux donnés par le protocole **AT-FREE**.

Les graphes du nombre de paquets transmis par rapport à la densité montré avant, donnent une idée sur le schéma de consommation en énergie provoqué par le protocole **SL-Free**. Il est à rappeler que l'énergie nécessaire pour transmettre un bit de données équivaut à exécuter mille instructions.

7.6 Impact de la densité sur la métrique FROB

La métrique **Frobenius** nous montre la marge de différence entre les distances réelles connus par le simulateur et les distance données par les calculs du protocole implémenté.



Graphe 6.5 : Impact de la densité sur la métrique FROB.

Le graphe 6.5 montre la marge d'erreur sur les distances selon la densité de réseau, soit en remarquant l'impact en nombre de nœuds dans le réseau (varie entre 50, 100 et 150), ou bien l'impact de la variation de la porté radio de nœud.

Pour le troisième graphe (avec n = 150), il ya une diminution sur la métrique de FROB, et qui montre que les résultats sont plus précis. Donc, en augmentant en porté radio, la densité augment, les informations de connectivité sont plus riches, et donc les résultats sont plus précis.

6.8 Conclusion

Dans une première partie de ce chapitre, nous avons présenté les éléments nécessaires constituant une topologie réseau sur le simulateur **NS**. Nous avons essayé de mettre l'accent sur les étapes suivies pour réussir l'implémentation d'un nouveau protocole ainsi que les différentes méthodes et classes utilisées. En effet, les difficultés rencontrées lors de l'ajout d'un module pour chaque protocole, sont surtout au niveau programmation du protocole en **C++** et la liaison entre ce dernier **(**code **C++)** et le code **OTcl**.

Notre étude sur les différentes méthodes et techniques de localisation, nous a montré que n'importe quelle démarche, cherche, en principe, à définir une variante qu'il faut utiliser pour déterminer ou bien approcher de la distance réelle entre capteurs. Dans ce cadre, le protocole de localisation **SL-FREE** a proposé de définir l'indice de proximité.

Dans la seconde partie de ce chapitre nous avons, en premier lieu, modéliser le problème avant de tester le fonctionnement du protocole **SL-FREE** sur une architecture réseau qui se rapproche des scénarios rencontrés dans le monde réel. Nous avons ensuite analysé et interprété les résultats obtenus. Nos simulations étaient sur une topologie aléatoire, car les topologies aléatoires sont généralement préférables pour effectuer des tests sur un algorithme de localisation. Lors de nos simulations, nous avons fait des évaluations dans le cadre d'étudier les performances de notre proposition. Les résultats obtenus montrent que notre proposition apporte une contribution intéressante à la problématique abordée. Car le protocole **SL-FREE** assure une information de localisation approximative en première phase. Ce protocole donne des résultats comparables et même plus performants que les deux protocoles étudiés avant, c'est-à-dire **HT-Refine** et **AT-Free**. En deuxième et troisième phase, le protocole **SL-FREE**, donne des localisations fines sur lesquels d'autres travaux de simulations sont encours.

CONCLUSION

Les réseaux de capteurs sans fil ont connu au cours de ces dernières années un formidable essor aussi bien dans l'industrie que dans le milieu universitaire. Cela est principalement attribuable à l'ampleur sans précédent des possibilités qu'offre cette technologie. Flexibilité, hautes capacités de captage, coût réduit, installation rapide sont les caractéristiques qui ont permis aux réseaux de capteurs d'avoir des nouveaux domaines d'applications multiples et excitants. Ce large étendu d'application fera de cette technologie émergeante une partie intégrale de nos vies futures.

Cependant, la réalisation des réseaux de capteurs nécessite la satisfaction de certaines contraintes qui découlent d'un nombre de facteurs guidant la phase de conception, tel que la tolérance aux pannes, la scalabilité, le coût, le matériel et la topologie. Comme ces contraintes sont très exigeantes et spécifiques aux réseaux de capteurs, des nouvelles techniques et protocoles pour ce type de réseaux, sont nécessaires. Plusieurs chercheurs sont actuellement engagés dans le développement des technologies requises au niveau des différentes couches de la pile protocolaire des réseaux de capteurs.

Parmi les problèmes très connus pour ce type des réseaux, il y a le problème de localisation. Les premières approches de résolution de ce problème, ont été proposées dans un contexte centralisé : une unité centrale devait effectuer tous les traitements et affilier à chaque nœud ses coordonnées. Toutefois, il est vite devenu évident que cette stratégie était mal adaptée à ce type de réseau puisqu'elle ne respectait pas la principale contrainte d'un réseau de capteurs sans fil, qui est la contrainte énergétique. De plus, la robustesse de l'approche centralisée est faible; une panne de l'unité centrale aura une répercussion sur tout le système. Une implémentation distribuée à travers tout le réseau constitue donc une réponse adéquate au problème de localisation.

En étudiant les algorithmes de localisation, le dispositif le plus attrayant des algorithmes de localisation basés sur la connectivité (donc libres mesures) est leur simplicité. Toutefois, ils peuvent seulement fournir une évaluation à grain grossier de position de chaque nœud, ainsi elle signifie qu'elles sont appropriées aux applications qui peuvent travailler avec une évaluation approximative de position. Par contre, Les approches basés sur les mesures représentent plus de performances par rapport à l'exactitude, mais peuvent exiger un matériel supplémentaire. D'autres méthodes basées sur les mesures s'appuient sur une modélisation des pertes en précision lors du calcul des angles ou des distances. Ce modèle est ensuite utilisé pour déterminer les positions des capteurs. Or, cette perte en précision est liée à l'environnement de la zone de déploiement du réseau et aucune modélisation n'est capable de représenter n'importe quel environnement.

Trouver une méthode qui garantisse des positions précises quel que soit l'environnement du réseau semble difficile voir impossible sans un minimum de connaissances relatives à cet environnement. Chaque méthode proposée, dans cette étude, doit donc être considérée comme une méthode à part entière si l'environnement du réseau le permet, sinon elle doit être considérée comme un socle sur lequel seront ajoutées les spécificités liées à l'environnement. Donc, une bonne maîtrise et connaissance de ces diverses méthodes est nécessaire afin de dimensionner judicieusement sa

propre solution de localisation. Cette solution ne doit pas être surdimensionnée, sinon elle entraîne un surcoût soit au niveau de l'infrastructure soit au niveau du terminal de localisation.

La simulation est une solution plus aisée pour valider les performances d'un algorithme de localisation; elle permet aux chercheurs de tester les performances d'un algorithme en se basant sur des critères. A partir de là, l'évaluation et la comparaison des algorithmes de localisation peuvent être faites en utilisant diverses métriques. Il faut donc réduire au maximum les erreurs possibles de conception en procédant par une phase de validation via la simulation, pour assurer le bon fonctionnement de tous les dispositifs.

Notre étude sur les différents méthodes et techniques de localisation, nous a montré que n'importe quelle démarche, cherche, en principe, à définir une variante qu'il faut utiliser pour déterminer ou bien approcher de la distance réelle entre capteurs. Dans ce cadre, nous avons donné une proposition d'un nouveau protocole de localisation, que nous l'avons appelé **SL-FREE**. Ce protocole de localisation a proposé de définir l'indice de proximité entre les capteurs comme étant la base de calcul des positions. La nouvelle famille de protocole de localisation dans les réseaux de capteurs sans fil, **SL-Familly** favorise une démarche distribué et collaborative. Donc, afin de garantir l'économie d'énergie et l'évolutivité de notre algorithme, nous avons choisi une démarche distribuée, ou les nœuds unknowns calculent indépendamment leur propre position en utilisant seulement l'information locale. Le premier protocole de cette famille des protocoles est le protocole **SL-FREE**. Il assure, en première phase, une évaluation approximative à la disposition des capteurs. De cette manière une information approximative est disponible au niveau des nœuds. Tel que, les expérimentations ont montré que l'estimation approximative va de paire avec l'estimation approximative géographique dans le cas des réseaux de capteurs dense.

Nous avons décris les différents détails liés à l'implémentation du protocole **SL-FREE**, en mettant en relief les structures de données, les algorithmes développés et les différentes phases de méthode. Afin de faciliter la compréhension de notre conception, nous avons choisi de modéliser certaines phases par des organigrammes et des diagrammes **UML**, avant de tester le fonctionnement du protocole **SL-FREE** sur une architecture réseau qui se rapproche des scénarios rencontrés dans le monde réel. Nous avons ensuite analysé et interprété les résultats obtenus. Nos simulations étaient sur une topologie aléatoire, car les topologies aléatoires sont généralement préférables pour effectuer des tests sur un algorithme de localisation. Et lors de nos simulations, nous avons fait des évaluations dans le cadre de l'étude des performances de notre proposition.

Les résultats obtenus montrent que notre proposition apporte une contribution intéressante à la problématique abordée. Car le protocole **SL-FREE** assure une information de localisation approximative en première phase, et qui montre des résultats comparables et même, dans certains scénarios, plus performantes que les deux protocoles étudiés avant, celle de **HTRefine** et **ATFree**. Et en deuxième et troisième phase, le protocole **SL-FREE** réalise un raffinement des estimations de positionnement.

Nous avons décrit un algorithme collaboratif distribué pour la localisation dans les réseaux de capteurs sans fil indépendamment du type de mobilité des nœuds, qui soient statiques ou mobiles. La méthode présentée, **SL-FREE** présente des résultats et des idées très prometteuses, et elles résolvent un grand nombre de problèmes. Dans le cadre de cette nouvelle méthode, nous cherchons toujours à créer un cadre plus général de localisation qui nous libère de certaines des restrictions et des

prétentions faites dans notre travail jusqu'ici. La vision initiale de cette proposition, cherche de mettre en évidence toute une famille des protocoles et qui se base sur le principe des informations de connectivités. Ce premier travail présente le protocole **SL-FREE**, où le nœud n'a pas besoin d'avoir une capacité particulière de mesure (équipement particulier de mesure d'angles où…). Comme perspective, nous comptons proposer pour qu'un nœud bénéficie des informations de mesures disponibles, des mesures d'angle, ….

Nous comptons continuer nos travaux dans les directions suivantes: l'ajout d'une phase de raffinement pour la déduction de la taille d'indice size, par les ancres lors de la phase 2 du protocole. Pour assurer des améliorations dans le cadre de la marge d'erreur des résultats. Nous comptons aussi prendre en considération la non homogénéité de la portée de communication. Aussi, nous envisageons de traiter la présence des obstacles dans le domaine.

APPENDICE A

LISTE DES SYMBOLES ET DES ABREVIATIONS

AEA	Adaptative Election Algorithm
ADNL	Accurate Distributed Node Localization algorithm
AoA	Angle d'Arrivée
API	Application Programming Interface
APIT	Approximate Point In Triangle Test
APS	Ad Hoc Positioning System
ARD	Average Relative Deviation
AT-Family	Family of Approximation Techniques
ATM-Family	Family off Approximation Techniques in Mobile context
CDMA	Code Division Multiple Access
CONSER	Cooperative Serials Program
COTS	Commercial OffThe- Shelf
CRB	Cramer-Rao
CSMA/CA	Carrier Sense Multiple Access with Collision Avoidance
CSMA/CD	Carrier Sense Multiple Access / Collision Detection
DARPA	Defense Advanced Research Projects Agency
DGPS	Differential GPS
DOD	Department Of Defense
DVM	Dynamic Velocity Monotonic
ESA	Agence spatiale européenne
FAA	Federal Aviation Administration
FROB	FROBenius
HiperLAN	High PERformance radio LAN
IEEE	Institue of Electrical and Electronics Engineers
IP	Internet Protocol
IRNSS	Indian Regional Navigational Satellite System
JiST/SWANS	Java in Simulation Time / Scalable Wireless Ad hoc Network Simulator
Galileo	European Satellite Navigation Competition
GDE	Global Distance Error
GDOP	Geometric Dilution of Precision
GEAR	Geographical and Energy Aware Routing
GER	Global Energy Ratio
GF	Geographic Forwarding
GloMoSim	Global Mobile Simulator
GLONASS	GLObal NAvigation Satellite System
GPS	Global Positioning System
GTSNetS	Georgia Tech Network Simulator
MANET	Mobile Ad hoc NETworks

MAC	Media Access Control
MADRD	Mobility Aware Dead Reckoning Driven
MAE	Erreur Absolue Moyenne
MCL	Monte Carlo localization
MCU	Microcontroller
MDS	Multi Dimensional Sealing
MDS–MAP	Multi Dimensional Sealing – Mobile Application Part
ML	Maximum Likelihood
MuR	Method using Rules
NS	Network Simulator
NSF	National Science Foundation
NP	Neighbor Protocol
OS	Operator System
OSI	Open Systems Interconnection
PAN	Personal Area Network
PEAS	A Robust Energy Conserving protocol
PDA	Personal Digital Assistant
PIT	Point In Triangle Test
QoS	Qualité de Service
QLoP	Qualitative Location Protocol
OTcl	Object Tool Command Language
RSSI	Received Signal Strength Indicator
RTP	Real-time Transfert Protocole
SB	Base Station
SEP	Schedule Exchange Protocol
SFR	Static Fixed Rate
SL-Family	Self Localisation Family
SPEED	A Stateless Protocol for Real-time Communication
Tcl	Tool Command Language
TCP	Transmission Control Protocol
TDoA	Time Difference of Arrival
TEEN	Threshold sensitive Energy Efficient sensor Network protocol
TIMATION	TIMe navigATION
ToA	Time of Arrival
TRAMA	TRaffic-Adaptive Medium Access control
UDP	User Datagram Protocol
UML	Unified Modeling Language
VINT	Virtual InterNet Testbed
WAAS	Wide Area Augmentation System
WLAN	Wireless Local Area Network
WSN	Wireless Sensor Network
WPAN	Wireless Personal Area Network
Z-MAC	Zebra MAC

APPENDICE B
LA SIMULATION DU PROTOCOLE SLFREE SUR NS2

B.1Définition de Simulateur NS

NS2 (**N**etwork **S**imulator **2**) est un environnement de simulation pour les réseaux sans fil et filaires. Il a été conçu à l'université **UC Berkely** (L'université de **C**a**lifornie** à **Berkeley**). Il a été considérablement utilisé et amélioré au cours des dernières années par de nombreux chercheurs. Le **NS** est un environnement de simulation à base d'évènements discrets, offre la possibilité pour tester, analyser, évaluer et critiquer avant d'envisager l'implémentation concrète dans un réseau. La simulation nous permet donc d'effectuer ces tests à moindres coûts et de prendre des décisions primordiales. Le simulateur **NS** offre un support substantiel pour la simulation des protocoles de contrôle des transmissions (**TCP T**ransmission **C**ontrol **P**rotocol), des protocoles de routage, des protocoles de localisation, et des protocoles de multi casting à travers des réseaux filaires et sans fil (réseaux locaux et satellitaires).

B.2Développement de Simulateur NS

Au départ, le simulateur **NS** était une variante de « The **REAL N**etwork **S**imulator » en 1989, puis, a évolué principalement au cours de ces dernières années grâce à des efforts administrés par l'université de **Berkeley**. En 1995, le développement de **NS** a été soutenu par le **DARPA** (**D**efense **A**dvanced **R**esearch **P**rojects **A**gency), et cela, à travers le projet **VINT** (**V**irtual **I**nter**N**et **T**estbed) aux laboratoires **LBL**, **Xerox PARC**, **UCB**, et **USC/ISI**. Actuellement, **NS** est soutenu par le **DARPA** avec et par **NSF** (**N**ational **S**cience **F**oundation) avec **CONSER** (**C**ooperative **Ser**ials Program) tout en collaborant avec d'autres organismes de recherches tels que le **ACIRI**.

En fait, **NS** a toujours inclus des contributions consistantes et substantielles à d'autres chercheurs tels que les codes correspondants aux environnements sans fils de **UCB** Daedelus (1997) et des projets de **CMU** Monarch (1999) et **S**un **M**icrosystems.

B.3Description de l'environnement NS

Le simulateur de réseaux **NS** est un simulateur orienté objet, développé sur la base du langage **C++** et du langage de script **OTcl (Object Tcl),** où **Tcl (T**ool **C**ommand **L**anguage**)**, est un script interprété. La combinaison de ces deux langages permet, d'une part, la définition et la création des objets manipulés, ce qui est réalisé efficacement (et rapidement) par le langage **C++,** et d'autres part, la configuration des objets et la gestion des évènements à travers le langage de script **OTcl**. L'interpréteur **OTcl**, dérivé du langage **Tcl**, permet également une interaction avec les utilisateurs en leurs permettant d'écrire des scripts pour la définition du modèle de réseau (le nombre de nœuds, les liens, …etc.), la détermination du trafic dans le réseau (sources, destinations, type de trafic, …etc.), et la définition des protocoles utilisés, et cela, sur la base du langage **Tcl**.

NS est un logiciel très évolutif. En effet, de nombreux composants y sont intégrés chaque année. Ainsi, il existe de nombreuses versions pour les différents protocoles implémentés. De plus, il y a un grand nombre de classes prédéfinies qui permettent de mettre en œuvre plusieurs

implémentations intégrant des protocoles de transmission **TCP, UDP/IP (U**ser **D**atagram **P**rotocol **/I**nternet **P**rotocol**)**, des diverses files d'attentes (files de paquets), un routage fixe et dynamique.

Le simulateur de réseaux **NS2** sous **Linux** est disponible en version « AllInOne » (tous en un) en incluant, ainsi, plusieurs composants dont : **Tcl**, **Otcl**, **Tclcl**, **Xgraph**, **Ns**, **Nam**. Le regroupage de ces différents composants rend l'utilisation de l'environnement de simulation plus facile puisque les liens sont automatiquement établis (entre les composants), ce qui n'est pas le cas pour les systèmes **Windows** où ces composants sont installés indépendamment les uns des autres en imposant ainsi quelques manipulations pour la mise au point du simulateur (définition des liens entre les différents composants installés suivant un ordre chronologique respecté).

TCL	*Langage de script en open source, utilisé pour programmer NS, il faut l'installer en premier.*
TK	*C'est une interface utilisateur graphique pour TCL, elle s'installe donc juste après.*
OTCL	*Extension de Tcl/Tk pour la programmation Orientée Objet.*
Tclcl	*Interface Tcl/C++, il permet de faire le lien entre la hiérarchie existante en C++ et celle existante en Tcl dans Ns, grâce à ce composant on peut utiliser les 2 langages dans une même simulation de cette façon on obtient un simulateur flexible et puissant.*
NS	*Programme final, Ns a besoin des 4 composants précédents pour fonctionner.*
Nam	*Permet de visualiser graphiquement une simulation.*
X-Graph	*Afficher des graphiques pouvant contenir les instants d'émission des paquets (par exemple).*

Tableau B.1 : Les composants de simulateur NS2.

Le simulateur **NS** contient une double arborescence de classes définie (voir la figure A.1 [ANE 99]) :

- en **OTcl** dite arborescence interprétée utilisée par l'interpréteur.
- en **C++** dite compilée, elle forme l'arborescence utilisée par le moteur de simulation.

Les deux arborescences sont très proches l'une de l'autre. Du point de vue de l'utilisateur, il y a une correspondance univoque entre une classe d'une arborescence et une classe de l'autre arborescence [ANE 99].

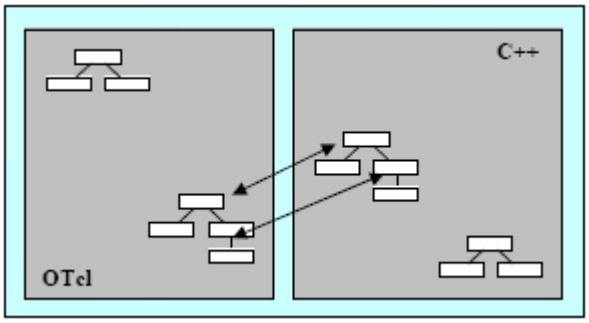


Figure B. 1 : Dualité de C++ et OTCL dans NS.

Pour ce qui est de notre simulation, nous avons pu manipuler les deux langages, en effet l'ajout d'un nouveau module dans le cœur de ce logiciel, qui est une tache assez complexe qui nécessite une maîtrise du langage de programmation **C++** ainsi qu'une bonne compréhension des interactions entre les différents composants et classes du simulateur. Quant au langage **Tcl**, nous nous sommes servis du langage **Tcl** aussi bien pour l'écriture des différentes parties du protocole de localisation que pour la définition des paramètres de l'environnement de simulation sur une plate forme **Linux**, en utilisant, pour cette fin, les opportunités offertes par le langage **Tcl**, en l'occurrence, la facilité de manipulation des variables, de définition des procédures, l'utilisation des boucles, …etc.

B.4Éléments de la simulation

1. Simulateur

La simulation est configurée, contrôlée et exploitée via l'interface fournie par la classe **OTCL** Simulator. Elle n'existe que dans l'interpréteur. Un script de simulation commence toujours par créer une instance de cette classe par la commande:

```
set ns_ [new Simulator]
```

2. Ordonnanceur (scheduler)

L'ordonnanceur est défini dans le fichier scheduler.{h,cc}. L'Ordonnanceur a en charge de choisir l'événement le plus proche en termes de temps, d'exécuter les traitements, de faire progresser le temps de simulation et d'avancer à l'événement suivant etc. Un seul événement est traité à la fois. Si plusieurs événements doivent être traités au même instant. Ils sont exécutés en série mais au même instant en termes de temps simulé.

3. Consommation de temps

Aucun objet dans la simulation ne peut faire avancer le temps. Pour consommer du temps, il faut obligatoirement passer par l'ordonnanceur.

4. Le modèle d'énergie

Attribut de la classe nœud ; il sert à représenter le niveau d'énergie d'un nœud. Il est caractérisé par une valeur initiale (initialEnergy_), une valeur de décrémentation (txPower_) lors de chaque transmission et une valeur de décrémentation (rxPower_) suite à la réception d'un paquet. Le nœud peut aussi consommer de l'énergie lorsqu'il ne reçoit pas et n'envoie pas.

5. Traitement séquentiel en temps simulé

Le temps de simulation est découplé du temps réel. Si aucun objet ne fait de consommation de temps, vis à vis du temps simulé tous les traitements se font en même temps (mais par rapport au temps réel ils sont exécutés en série). Un simulateur est naturellement une machine pseudo-parallèle [ANE 99].

B.5 Composants de la topologie

Pour définir l'architecture et la topologie du modèle à étudier nous devons tout d'abord présenter les classes de bases utilisables.

1. Nœud

La classe Node est une classe **OTcl** : elle n'a donc pas d'existence en tant que telle dans le simulateur. Cette classe et ses méthodes sont définies dans le fichier tcl/lib/ns-node.tcl. Un nœud est une collection de classifiers et d'agents. Le classifier démultiplexe les paquets. L'agent est habituellement l'entité d'un protocole. L'assemblage des composants est représenté par la figure A.2, [ANE 99] :

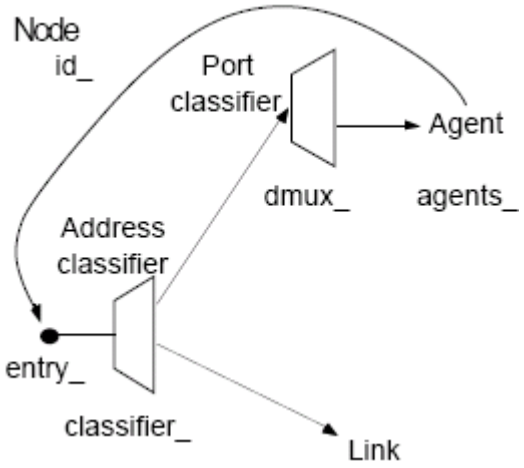


Figure B. 2 : Composants d'un nœud.

2. Lien

Le lien sert à relier les nœuds. Il modélise le système de transmission. Le lien est principalement caractérisé par un délai de propagation et une bande passante. C'est une classe **OTcl** qui regroupe un ensemble de composants dérivés de la classe Connector. Cette classe et ses méthodes sont définies dans le fichier tcl/lib/ns-link.tcl. Quelque soit le type du lien, il comporte 5 références sur des objets qui le composent, selon la figure A.3, [ANE 99].

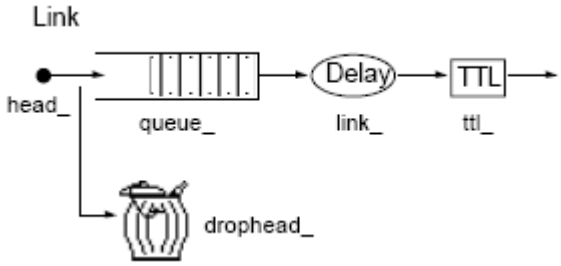


Figure B. 3 : Composants d'un lien.

3. Agent

L'agent est un autre composant d'un nœud. Il modélise les constructeurs et les consommateurs de paquets **IP**. La classe agent fournit des méthodes utiles au développement de la couche transport de modèle **OSI** (**O**pen **S**ystems **I**nterconnection) et à d'autres protocoles de gestion. Cette classe est à la fois dans l'interpréteur et dans le simulateur. C'est la classe de base pour définir des nouveaux protocoles dans NS. Elle fournit l'adresse locale et de destination, les fonctions pour générer les paquets, l'interface à la classe Application. Actuellement NS comporte de nombreux agents citons: **UDP**, protocoles de routage, différentes versions de **TCP**, **RTP** (**R**eal-time **T**ransfert **P**rotocole), etc. Ils constituent les points terminaux du réseau qui reçoivent ou délivrent les paquets des applications. A chaque agent est attribué un port, tel que l'adresse d'un agent se compose du numéro de son nœud et de son port [ROS 04].

4. Les applications

Au dessus de la couche transport se trouve l'application qui va générer le trafic. Il y aura au plus une application par agent.

B.6Implémentation du Protocole SLFree sous ns2

L’environnement de simulation **NS** est constitué de deux hiérarchies : compilée et interprétée. Ce simulateur est riche en outils et en protocoles ; ceci dit le protocole **SL-FREE** n’est pas présents dans **NS**. Il est donc nécessaire de l’incorporer.

Lors d’introduction d’un nouveau protocole, à l’outil **NS**, nous devons d’abord programmer le protocole par le biais du langage **C++**. Une fois **NS** compilé, les simulations peuvent être effectuées en utilisant des scripts **OTcl** [ROS 04].

SL-FREE est un protocole qui assure la fonctionnalité de localisation d’un nœud dans un réseau de capteurs sans fils. Nous présentons dans ce qui suit un nouveau module pour **NS** capable de simuler le comportement de protocole **SL-FREE**.

La vérification de la performance du protocole s’est réalisée par une application « .tcl », tandis que les **API** (**A**pplication **P**rogramming **I**nterface) suivants sont nécessaires pour introduire le nouveau protocole **SL-FREE** en **NS** :

- SLFree.h : le fichier header **C++** qui définit les différentes classes et structures de données.
- SLFree.cc : le fichier source **C++** qui contient les différentes fonctions membres nécessaires pour traduire le protocole proposé.
- ns-SLFree.tcl : la librairie **OTcl** nécessaire pour définir les différentes fonctions de l’objet tcl.

1. Les étapes de l'ajout d’un nouveau protocole dans NS

Pour ajouter un nouveau protocole dans **NS,** nous allons suivre les étapes suivantes [ANE 99]:

1. déclaration de l'en-tête du paquet du protocole (ou unité de données du protocole).
2. Déclaration de la classe du protocole (méthodes et attributs). Le nom des attributs est suffixé par le caractère "_".

3. définition de la liaison entre le code **C++** et le code **OTcl** par la déclaration des variables statiques dérivées de la TclClass et de PacketHeaderClass respectivement pour l'agent et pour le paquet du protocole.
4. ajout du protocole ID dans *~/ns-2.34/commun/packet.h* par un #define. L'indicateur de fin de liste des protocoles ID est PT_NTYPE. Le protocole ID sert à identifier le type de paquet.
5. ajout du nom du protocole dans PT_NAMES (*~/ns-2.34/commun/packet.h*). L'ajout doit se faire à la position indiquée par la valeur du protocole ID. Pour les traces, la liste sert à remplacer l'ID du protocole par une chaîne de caractères.
6. ajout du paquet dans la liste gérée par le packet manager. Ceci est rendu nécessaire pour le calcul de l'offset de l'en-tête du paquet.
7. enfin modification du *~/ns-2.34/makefile.in* par l'ajout du fichier objet du protocole à la liste des fichiers objet de ns.
8. recompilation de ns par la commande *make*.

2. Paquet et en-tête

La modélisation de notre protocole consiste à définir des nouvelles structures de données dans un fichier header et à implémenter les fonctions nécessaires dans un fichier source **C++**. Tout d'abord, on a déclaré, dans le fichier .h, une structure de données pour l'entête du nouveau protocole qui va supporter les données de contrôle nécessaires :

- L'identificateur de nœud.
- Les coordonnées (x, y),
- La liste de voisinage.

En plus d'une structure de données à géré et qui contient les listes de voisinage des voisins de nœud.

3. L'Agent SL-FREE

Pour créer un nouvel agent nous devons définir son héritage, créer sa classe et définir les fonctions virtuelles de l'**API** Agent. Ensuite nous allons définir les fonctions qui feront office de liaison avec l'interpréteur. Le travail se termine par l'écriture de code **OTcl** pour l'accès à l'agent [ANE 99].

Dans notre cas, l'agent de protocole **SL-Free** opère au dessus de la couche Agent, sa classe sera par conséquent une hérité de la classe Agent.
Pour définir la liaison entre l'agent SLFreeAgent et l'interpréteur, nous devons faire le « mapping » entre le nom de la classe Agent et l'objet crée. Ceci est élaboré comme suit :

Une instance *Agent/SLFree* est crée à partir de l'interpréteur par la commande: *new Agent/SLFree*. Cette commande appelle via *create_shadow()* la méthode *SLFreeAgentClass::create()* pour la création d'un *SLFreeAgent* dans le simulateur. Deux objets *Agent/SLFree* seront donc crées, un pour l'interpréteur et son clone *SLFreeAgent* pour le simulateur.

4. Les structures de données

Chaque nœud possède les structures de données suivantes, pour assurer les traitements demandés lors d'exécution de protocole :

- ✓ une structure de données, qui contient les informations suivantes:
  - o identificateur id ;

- les coordonnées (x, y) ;
- une liste sur les nœuds voisins.

- Si **indic** = 0 : c'est un message 'hello'.
- Sinon
  - Si **indic** = 1 : message pour donner la table de voisinage.
  - Sinon : message envoyé par les ancres.

✓ Une structure de données sur les indices de proximité calculés pour chaque voisin et la distance relative (ajouter sur le fichier .h).
✓ Une structure de données pour sauvegarder les informations reçues des ancres sur la taille d'indice size.

Apres avoir déclenché le calcul, les informations sur le nœud concerné, seront supprimé de la mémoire.

5. Format des messages

Message 'HELLO'

Sur le message '**HELLO**' envoyé initialement par chaque nœud, contient :

indic	**id_**	**Tab_**	**x_**	**y_**

indic : indicateur pour le type de paquet. (ajouter sur le fichier .h).

id_ : L'identifient de nœud émetteur.
x_ et **y_ :** coordonnées de nœud émetteur de message, (si l'émetteur est un unknown, x_ = y_ = -1).
Tab_ : La table de voisinage de nœud.

Pour déterminer le type de message reçu, lors de la réception d'un paquet, le nœud teste le champ « indic » qui lui indiquera le traitement nécessaire pour le message adéquat.

Message sur la table de voisinage

Après écoulement d'une timer (cette timer est initialisé pour donner du temps aux nœuds de recevoir le maximum d'informations via les messages d'initialisation de réseau 'hello') chaque nœud diffuse sa table de voisinage, à travers le réseau pour leurs voisins à 1 saut.

Message envoyé par les ancres

Après écoulement d'une timer, l'ancre procède au calcul de la taille d'indice de proximité. L'ancre diffuse cette information à travers le réseau pour permettre aux nœuds unknowns de faire évaluer leurs distances relatives vers des distances réelles.
Ce message envoyé par les ancres, a la structure suivante :

indic	**id_**	**IndSize_**

indic : indicateur pour le type de paquet. (ajouter sur le fichier .h).

id_ : L'identifient d'ancre émetteur.
IndSize_: la taille de l'indice de proximité calculé par cette ancre.

Donc, selon le champ indic, on a 3 types des messages. Et selon le type de paquet les traitements sont montrés les figures B.4, B.5, B.6 et B.7.

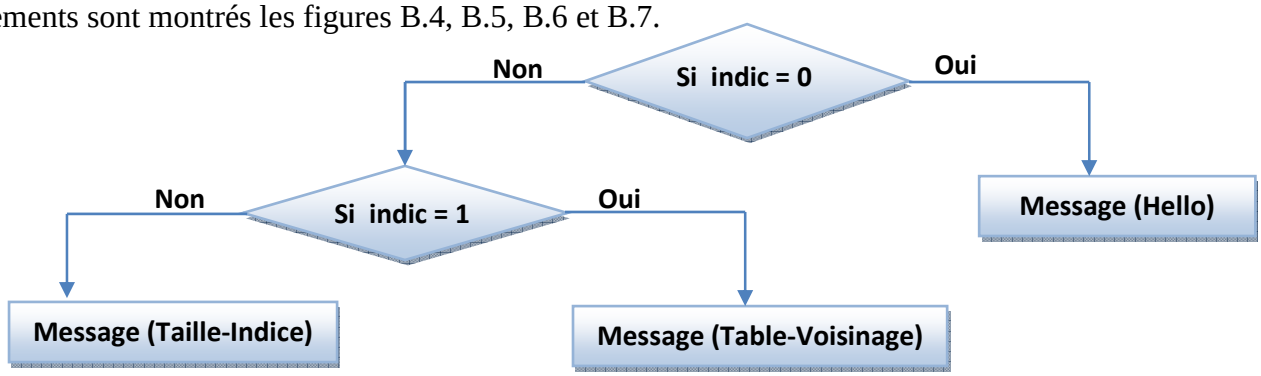


Figure B. 4 : Organigramme sur le type de message reçu dans SL-FREE.

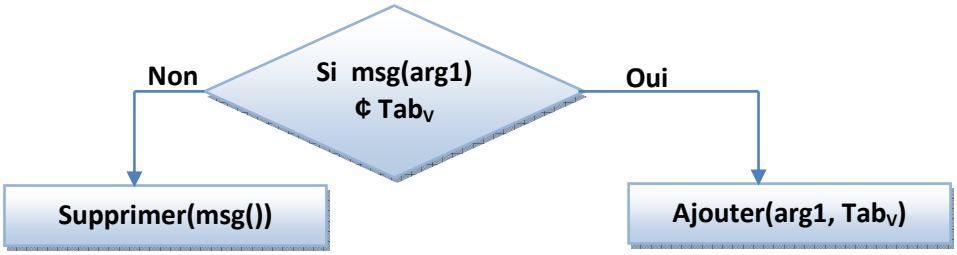


Figure B. 5 : Traitement d'un paquet 'hello' dans SL-FREE.

Le principe de notre algorithme est que le message 'hello' et 'liste d voisinage' ne soit pas transmettre à travers le réseau que sur un saut, donc si un nœud reçoit un message de ce type c'est que l'émetteur est un voisin à un saut. Et donc, automatiquement on considère l'émetteur comme voisin et on prend sa liste de voisinage, si le message la contient.

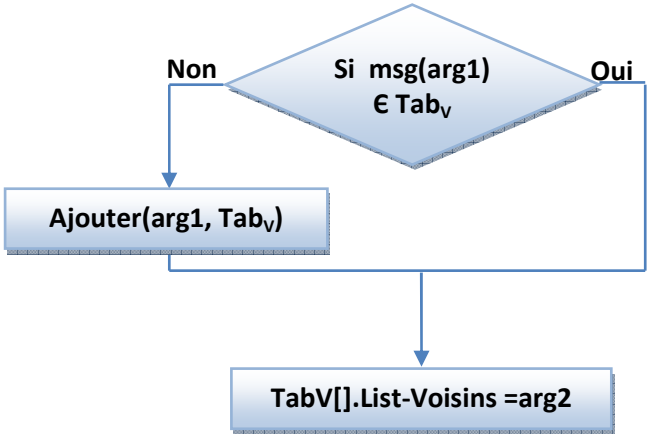


Figure B. 6 : Traitement d'un paquet sur la 'table-voisinage' dans SL-FREE.

Suite à l'écoulement de timer 3, l'ancre calcul la taille de l'indice de proximité. En suite il diffuse cette information.

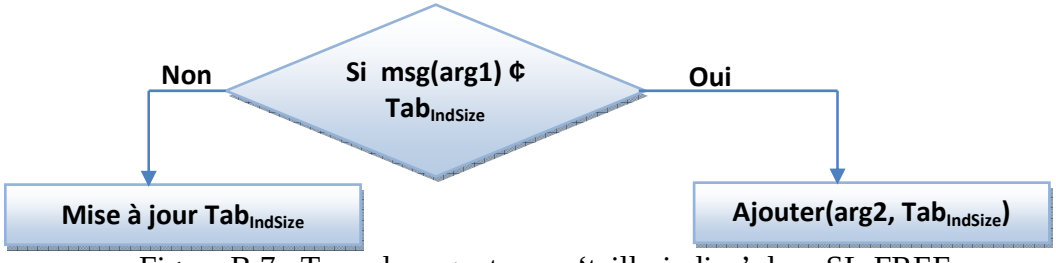


Figure B.7 : Type de paquet reçu, 'taille-indice' dans SL-FREE.

Donc, un nœud a besoin de la structure suivante :

Id_ : l'identificateur de nœud.

Tab_V : Une table qui imbrique une autre structure de type table aussi et qui contient l'identificateur d'un voisin, leur coordonné (si c'est une ancre sinon ces champs sont à vide) et la table de voisinage pour ce voisin.

$Tab_{IndSize}$: la table qui contient l'identificateur et la taille de l'indice de proximité reçu de chaque ancre.

6. Les Timers

La première timer t_1 (TimerSend): le nœud ne diffuse pas son message 'hello' à travers le réseau, avant l'écoulement de ce temps. La valeur de **t_1** est choisie aléatoirement dans un intervalle défini ; ceci pour éviter le cas où tous les nœuds diffusent au même moment (trop de collisions).

La deuxième timer t_2 (TimerDifTab): le nœud envoie sa table de voisinage aprés l'écoulement de ce timer.

Le troisième timer t_3 (TimerIndSiz): l'ancre doit attendre ce temps avant de procéder au calcul de la taille d'indice de proximité, en deuxième phase de l'algorithme **SL-FREE**.

B.7 Modification des fichiers existants de NS

Afin de poursuivre la définition du nouveau protocole en **NS** et de déterminer les valeurs par défaut de ses différents paramètres, quelques fichiers doivent être modifiés comme : ns-lib.tcl, ns-packet.tcl, ns-default.tcl, packet.h, makefile.in [BNE 05].

Packet .h

Parmi les classes qui assurent la gestion des paquets dans **NS**, nous citons la classe packet_info. Cette classe contient toutes les valeurs littérales de tous les paquets. Elle est utilisée comme un « Intermidaire » entre les valeurs numériques du type du paquet et leurs noms symboliques.

Quand un nouveau type du paquet est défini, son code numérique devrait être ajouté au packet_t de l'énumération et son nom symbolique devrait être ajouté au constructeur de p_info.[BNE 05].

Donc, pour définir le paquet de nouveau protocole, il faut modifier le fichier ~/ns-allinone-2.34/ns-2.34/common/packet.h. Nous changeons la line 63 de variable PT_NTYPE, pour mettre la définition de nouveau paquet PT_SLFREE = 62. Mais il faut s'assurer que le variable PT_NTYPE est à la fin de cette liste [ELM 10].

Nous ajoutons à la ligne 254, les lignes suivantes de code, pour définir le type et la priorité de notre protocole, et donc, il faut savoir positionner ces lignes selon le type de la priorité de protocole:

```
type == PT_SLFREE
Et sur la ligne 390:
// SLFREE patch
name_[PT_SLFREE] = "SLFREE";
```

Makefile.in

Pour la compilation du nouveau protocole **SL-FREE**, nous avons ajoutés « agent_slfree.o » et « slfree_agent.o » à la variable OBJ_CC dans le fichier ~/ns-2.34/makefile.in. Ensuite nous avons recompilé **NS** en tapant "make" sous la direction de ~/ns-2.34/ [BNE 05].

ns-packet.tcl

Maintenant nous modifierons des fichiers **TCL** pour créer l'agent de cheminement. D'abord nous définissons le nom de protocole pour l'utiliser dans le fichier tcl, on ajoutant la ligne suivante à la ligne 172 de fichier ~/ns-allinone-2.34/ns-2.34/tcl/lib/ns-packet.tcl [ELM 10]:

```
# SLFREEE patch
SLFREE  #Localisation in WSNs
```

ns- default.tcl

Dans ~/tcl/lib/ns-default.tcl une valeur par défaut doit être donnée :

```
# Defaults defined for Slfree
 Agent/Slfree set accessible_var_ true
```

ns-lib.tcl

Dans ce fichier, nous allons placer l'agent de cheminement, à la ligne 633 :

```
SLFREE { Set    ragent    [$self create-slfree-agent   $node]  }
```

Sur la ligne 860 de ce fichier, les lignes de code suivantes doivent être ajoutées aussi :

```
Simulator instproc create-slfree-agent  { node }  {
#Create SLFREE routing agent
Set ragent [new Agent/SLFREE [$node  node-addr]]
$self  at 0.0 "$ragent  start"
$node set ragent_ $ragent
Return $ragent}
```

ns-agent.tcl

Dans ce fichier, nous placerons les numéros d'accès de l'agent de cheminement, tel que le sport est le port de source, et dport est le port de destination. A la ligne 202 de ~/ns-allinone-2.34/ns-2.34/tcl/lib/ns-agent.tcl [ELM 10]:

```
Agent/SLFREE instproc init args {
$self     next   $args}
Agent/SLFREE set sprot_ 0
Agent/SLFREE set dprot_ 0
```

De plus il ya des tutoriaux qui demande l'ajout de code suivant au fichier ~/ns-allinone-2.34/ns-2.34/tcl/lib/ns-mobilenode.tcl :

```
#Special processing for SLFREE
Set slrponly [string first "SLFREE" [$agent info class]]
        if {$slrponly != -1 } {
          $agent if-queue [$self set ifq_(0)]   ;# ifq between LL and MAC  }
```

cmu-trace.h

Sur le fichier trace/cmu-trace.h ajuter la ligne : void format_atfree(Packet *p, int offset);

cmu-trace.cc

```
void CMUTrace::format_atfree (Packet *p, int offset) {
 /*cette procedure est utilize pour definir les information du fichier trace*/
}
```

Et :

```
 void CMUTrace::format(Packet* p, const char *why){
case PT_ATFREE:    format_atfree(p, offset);
                   break;
```

Lors de la simulation, **NS** introduit tous les entêtes des paquets définie sur son environnement, par défaut. Et dans le but de bien contrôler le trafic généré par le protocole en étude, lors de la simulation, il est préférable d'inclure seulement les en-têtes de paquet qui sont d'intérêt à notre simulation spécifique.

COMPILATION

Après toutes ces modifications sur les fichiers de répertoire de notre simulateur, il faut lancer la recompilation de **NS** pour qu'il prenne en considération les changements effectués. La recompilation est lancée par les commandes suivantes:

```
make clean
make
```

APPENDICE C
ALGORITHMES DE LOCALISATION DANS LES RESEAUX DE CAPTEUR MOBILES

C.1 Introduction

Selon les auteurs, tous les algorithmes destinés au réseau statique peuvent être arrangé pour les réseaux mobiles, via l'application d'une stratégie d'ordonnancement sur l'algorithme. Mais ces conversions sont faites avec aucune considération pour la façon dont la mobilité peut être exploitée pour réaliser la localisation. Contrairement à l'algorithme de **Bergamo** et **Mazzini** dans [BER 02] qui considère la localisation avec des nœuds mobiles [HU 04].

D'une autre part, le problème de localisation dans un domaine mobile, a été intensivement étudié en robotique. Le travail de localisation en robotique assume habituellement une carte antérieure ou précédemment installée, et essaye de déterminer la position du robot en basant sur son mouvement et les données de capteur.

Ladd et autres, ont utilisé une approche de localisation de robotique pour réaliser la localisation précise pour les réseaux sans fil, en exploitant la variation instruite dans des forces de signal de RF reçues. Leur approche se comporte bien pour la localisation pour des environnements fixes et à l'intérieur, mais assume des positions fixes d'ancres et a besoin d'une phase d'apprentissage, donc, cette démarche n'est pas bien adaptée aux applications mobiles de réseau de capteur.

C.2 Stratégie d'ordonnancement

Dans un contexte de mobilité, un capteur ne peut invoquer le calcul de sa position de façon continue à cause de la contrainte énergétique. Pour répondre à ce problème, il est nécessaire de définir une stratégie d'ordonnancement des dates auxquelles les capteurs calculeront leurs positions tout en maintenant un niveau de précision raisonnable [SAA 08].
Cette section décrit les trois stratégies d'ordonnancement proposées dans [TIL 05], où les auteurs supposent qu'un capteur obtient sa localisation exacte lorsqu'il invoque le calcul de sa position.

1. La stratégie Static Fixed Rate (SFR)

La stratégie **SFR** (**S**tatic **F**ixed **R**ate) est basée sur un raisonnement simpliste : dans cette méthode, chaque capteur invoque le calcul de sa position après l'écoulement d'une période fixe t_{sfr}. Ainsi, lorsqu'un capteur calcule sa position au temps t, il obtient une position (x_t, y_t) et effectuera le prochain calcul de sa position au temps $t+ t_{sfr}$. Enfin, entre les temps t et $t + t_{sfr}$, le capteur se considère localisé en (x_t, y_t).

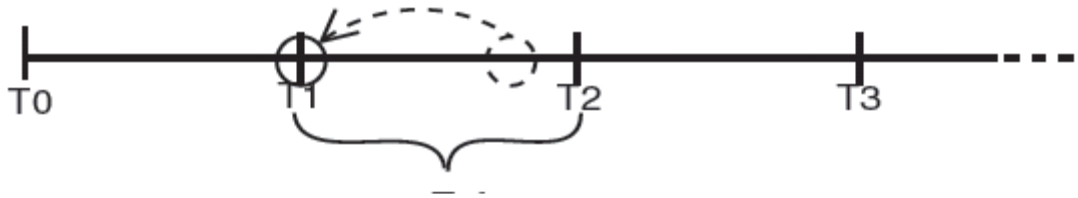


Figure C.1 : La stratégie static fixed rate SFR.

Exemple :

Sur la figure C.1, les temps t_1, t_2, t_3 ... respectent la période fixée t_{sfr}. Lorsque le capteur se déplacera entre les temps t_1 et t_2 par exemple, sa position estimée sera celle calculée en t_1.

Cette stratégie est simple mais présente un défaut majeur puisqu'elle ne prend pas en compte la mobilité du capteur lors du calcul de la période $t_{sfr.}$ En effet, si un capteur est à l'arrêt, il est inutile qu'il continue à calculer sa position. Pourtant, à chaque période t_{sfr}, il invoquera le calcul de sa position entraînant ainsi une consommation d'énergie supplémentaire. A l'inverse, si la vitesse de déplacement du capteur est élevée ou bien si la trajectoire de son déplacement n'est pas en ligne droite, le capteur aura tout intérêt à invoquer fréquemment le calcul de sa position afin de maintenir un bon niveau de précision, ce qui n'est pas le cas dans cette stratégie d'ordonnancement.

2. La stratégie Dynamic Velocity Monotonic (DVM)

A l'inverse de **SFR**, la stratégie **DVM** (**D**ynamic **V**elocity **M**onotonic) base son ordonnancement sur la mobilité des capteurs : plus la vitesse du capteur augmente (resp. diminue), plus la période t_{dvm} diminue (resp. augmente), ce qui maintient un niveau d'erreur de position raisonnable.

Principe :

Un capteur calcule une estimation de sa vitesse en fonction des deux dernières positions qu'il a obtenues. En divisant la distance entre ces deux positions par la période t_{dvm} séparant les temps de calcul de celles-ci, il obtient une estimation de sa vitesse actuelle et peut ordonnancer son prochain temps de calcul de sorte à maintenir le niveau de précision souhaité. Cela suppose que le capteur ne changera pas de vitesse jusqu'au prochain temps. Ensuite, comme dans **SFR**, si un capteur obtient sa position à un temps t, il considérera cette position comme étant la sienne jusqu'au prochain temps t + t_{dvm}.

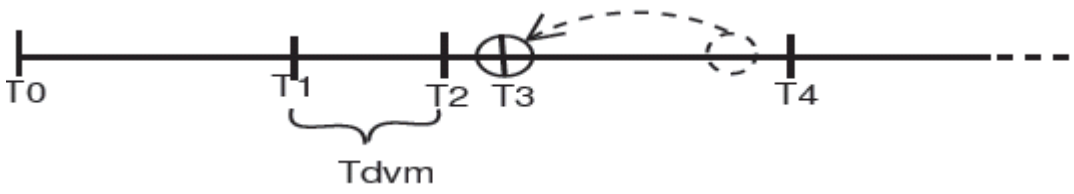


Figure C.2 : La stratégie DVM**.**

Exemple :

La figure C.2, présente un exemple d'exécution de **DVM**. Les différents temps t_1, t_2, t_3 ... sont calculés en fonction de la vitesse du capteur. Par exemple, pour déterminer le temps t_3, le capteur utilise les positions obtenues aux temps t_1 et t_2. La vitesse du capteur ayant augmenté entre ces deux

temps, la période t_{dvm} est raccourcie pour obtenir le temps t_3. Par contre, puisque la vitesse diminue entre t_2 et t_3, t_{dvm} est allongée pour obtenir le temps t_4 et ainsi de suite,

DVM trouve ses limites en ne prenant pas en compte les trajectoires des capteurs dans sa stratégie d'ordonnancement des dates.

3. La stratégie Mobility Aware Dead Reckoning Driven (MADRD)

La stratégie **MADRD** (**M**obility **A**ware **D**ead **R**eckoning **D**riven) est un protocole prédictif qui base son ordonnancement des temps de calcul sur la vitesse du capteur mais aussi sur sa trajectoire ce qui lui permet de prédire sa localisation au cours du temps.

Principe :

Comme dans **DVM**, l'estimation de la vitesse s'effectue en fonction des positions calculées lors des deux derniers temps. Mais, à partir de ces données, une estimation de la trajectoire du capteur est également déduite. Ainsi, durant la période t_{madrd}, le capteur peut calculer une estimation de sa position en se basant sur la prédiction de sa mobilité. Lorsqu'il calcule à nouveau sa position, il vérifie la validité de sa prédiction en constatant l'écart entre sa nouvelle position et celle prédite. La valeur de cet écart est comparée à un seuil qui traduit la qualité de la prédiction. L'ordonnancement du prochain temps de calcul de la position du capteur dépendra de la qualité de la prédiction : si elle est jugée bonne (l'écart est inférieur au seuil), le capteur continue à utiliser la prédiction pour estimer sa position et redéfinir la valeur de la période t_{madrd}; si elle est mauvaise, le prochain temps de calcul sera proche afin de refaire une prédiction.

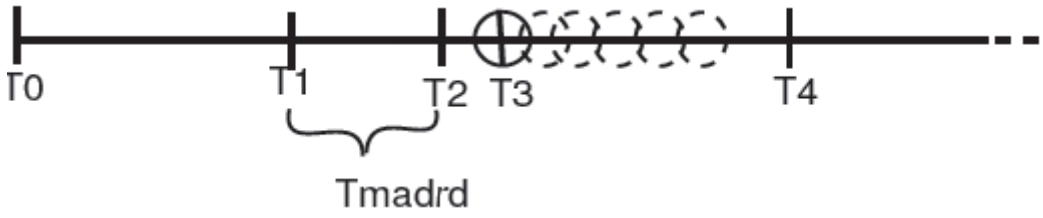


Figure C.3 : La stratégie MADRD

Exemple :

La figure C.3 est une illustration de cette stratégie d'ordonnancement : après les temps t_1 et t_2, le capteur vérifie la validité de la trajectoire prédite entre ces deux temps et ordonne t_3. Même chose pour le temps t_4 avec les temps t_2 et t_3. Au cours de son déplacement, le capteur estimera sa position en fonction de la prédiction qu'il a calculée.

Cette méthode est très intéressante puisqu'en présence de prédictions parfaites, la fréquence de calcul de la position d'un nœud sera très faible. Mais même lorsqu'elles ne le sont pas, cette stratégie maintient un bon niveau de précision des positions tout en prenant en compte la contrainte énergétique.

C.3 Adaptation des stratégies d'ordonnancement

Les stratégies d'ordonnancement décrites précédemment considèrent des capteurs capables de calculer leurs positions exactes dans un contexte où tous les capteurs sont mobiles. Les auteurs basent leurs travaux sur des réseaux dont les capteurs ont la capacité de calculer leurs positions réelles (tous les capteurs sont des ancres). De par cette hypothèse, ils souhaitent se focaliser uniquement sur l'étude de la stratégie d'ordonnancement des dates de calcul des positions des capteurs et concluent que la meilleure stratégie d'ordonnancement est **MADRD** suivie de **DVM** et enfin **SFR**. Cette hypothèse forte incite à se poser la question suivante : cette conclusion se généralise-t-elle dès lors que des erreurs de position sont introduites ? D'autre part, il faut rappeler que cette étude se restreint au modèle de mobilité où les capteurs et les ancres sont mobiles.

Les méthodes de localisation présentées ci-dessus sont applicables quelle que soit la mobilité des capteurs ou des ancres et fournissent des informations sur la précision des positions calculées. Afin d'intégrer ces données, les stratégies **SFR**, **DVM** et **MADRD** doivent être adaptées.

Parmi les différentes formes de mobilité, celle où les ancres sont mobiles mais les capteurs non (< S,M, {Ø, dist, angle}>) est particulière. En effet, dans cette configuration, les stratégies **DVM** et **MADRD**, qui ordonnancent les dates de calcul des positions en prenant en compte la mobilité des noeuds, n'ont aucun sens. La stratégie la plus adaptée à cette configuration est **SFR**. Dans cette stratégie d'ordonnancement, la période séparant deux calculs de position est fixe. Ainsi, un capteur invoquera le calcul de sa position à chaque date définie jusqu'à obtenir une position jugée précise (lorsque la valeur de e est petite).

Dans les deux autres formes de mobilité (< M,M, {Ø, dist, angle} > et < M, S, {Ø, dist, angle} >), les capteurs sont mobiles. Par conséquent, les trois stratégies peuvent être appliquées mais doivent être adaptées afin de prendre en compte la précision des positions. La stratégie **SFR** est la seule n'ayant pas besoin d'être modifiée. Elle ne considère pas la mobilité des capteurs puisque la période est fixe et peut donc être utilisée en l'état. En revanche, il n'en est pas de même pour **DVM** et **MADRD** puisque dans ces deux stratégies, l'ordonnancement des dates se fait en fonction de la mobilité des capteurs.

Or, les informations relatives à la mobilité (vitesse et trajectoire) dépendent de la précision des positions (leurs e). Ainsi, si ces stratégies d'ordonnancement s'appuient sur des positions exactes (e = 0), alors **DMV** et **MADRD** n'ont besoin d'aucune modification. Par contre, dès qu'un capteur calcule sa position avec une borne d'erreur élevée, la période séparant le prochain calcul de sa position doit être raccourcie. En conclusion, la période définie doit prendre en compte la mobilité mais aussi la précision de la position du capteur : plus un capteur a une position précise (resp. grossière), plus la période entre deux calculs de sa position est élevée (resp. faible).

C.4 Les algorithmes distribués libre mesure

1. La méthode ATM-Free

ATM-Family (**F**amily off **A**pproximation **T**echniques in Mobile context) est une extension au contexte de mobilité, des méthodes de localisation **AT-Family** décrites précédemment.

Il s'agit ici d'expliquer l'extension de la méthode **AT-Free** dans un contexte de mobilité où les capteurs ne connaissent ni distances, ni angles avec leurs voisins.

Principe :

Lorsqu'un capteur X souhaite calculer sa position, il lui faut les positions des ancres se trouvant dans son 2-voisinage. Pour cela, il diffuse à son 1-voisinage un message <$Node_{ask}$ >. Les nœuds recevant ce message diffusent à leurs tours un message <$Anchor_{ask}$ > dans le but de connaître les positions des ancres dans leurs 1-voisinages respectifs, c'est à dire dans le 2-voisinage de X. Les ancres recevant ce dernier message communiquent à leurs nœuds expéditeurs leurs positions. Enfin, les voisins de X envoient chacun un message communiquant les positions des ancres dans leurs voisinages ainsi que leurs positions s'ils sont des ancres. Au final, le capteur X obtient deux ensembles d'ancres : le premier, noté $N^1_\Delta(X)$, contient les ancres à 1 saut et le deuxième ensemble, noté $N^2_\Delta(X)$, contient celles distantes de 2 sauts. Ces deux ensembles sont définis tels que $N^1_\Delta(X) \setminus N^2_\Delta(X) = \emptyset$. Dès la formation de ces deux ensembles, le capteur X en déduit les éléments suivants :

- Pour toute ancre a appartenant à $N^1_\Delta(X)$, X appartient au disque centré en a de rayon r.
- Pour toute ancre a appartenant à $N^2_\Delta(X)$, X n'appartient pas au disque centré en a de rayon r.

De la conjonction de ces informations, le capteur définit une zone le contenant et estimera sa position en son centre de gravité. Evidemment, si X ne contient aucune ancre dans son 2-voisinage, il ne peut estimer sa position.

Exemple :

Sur l'exemple de la figure C.4, le capteur X possède deux ancres à un saut de lui (A, B $\in N^1_\Delta(X)$) et une ancre à distance deux (C $\in N^2_\Delta(X)$). Le capteur X appartient aux disques centrés respectivement en A et B de rayon r, mais n'appartient pas à celui centré en C de rayon r. X estime alors sa position en X'.

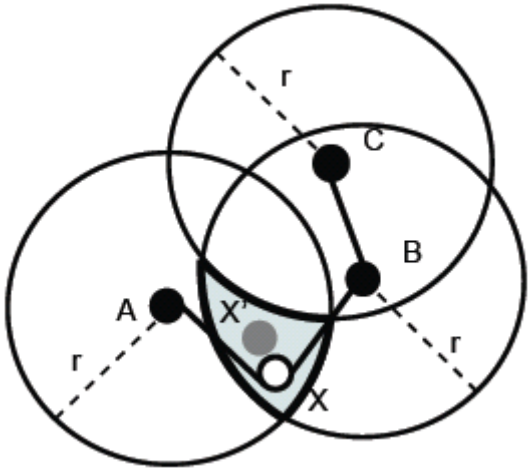


Figure C.4 : Estimation de la position de x dans ATM-FREE.

C.5 Les algorithmes distribués à base des mesures

1. L’algorithme de Bergamo et Mazzini

L’algorithme de **Bergamo** et **Mazzini** dans [BER 02] considère la localisation avec des nœuds mobiles, tel que, les auteurs ont considéré la mobilité des nœuds seulement et ont assumé un réseau avec deux ancres placés à des endroits fixes et qui peuvent transmettre à travers le réseau entier, et aux nœuds qui peuvent mesurer la force de signal exactement.

Au lieu d'utiliser la mobilité pour améliorer la localisation, ils ont étudié comment la mobilité rend la localisation plus difficile et ils ont constaté que les erreurs augmentent avec l'augmentation de la vitesse de nœud [HU 04].

2. La localisation basée sur un grille de Markov

Cet algorithme exploite la même démarche utilisé pour la localisation en robotique, qui assume habituellement une carte antérieure ou précédemment installée, et essaye de déterminer la position du l’objet mobile (au lieu d’utiliser un robot, dans les **WSN** utilisent un avion ou bien n’import quel type d’une station mobile) en basant sur son mouvement et les données de capteur .Cet algorithme a été proposé de traiter des densités multimodales et non gaussiennes. Cependant, la représentation de grille peut imposer une mémoire significative et une charge de calcul, particulièrement si on est intéressé par une haute résolution [HU 04].

3. L’algorithme Monte Carlo Localization

La localisation de **Monte Carlo** est un filtre particulier combiné avec les modèles probabilistes de la perception et du mouvement de robot.

Cet algorithme dépasse d'autres algorithmes de localisation proposés, dans l'exactitude et l'efficacité des calculs. L'idée principale de **MCL** est de représenter la distribution postérieure des positions possibles en utilisant un ensemble d'échantillons pesés. Chaque étape est divisée en phase de prévision et phase de mise à jour. Dans la phase de prévision, le robot fait un mouvement et l'incertitude de sa position augmente. Dans la phase de mise à jour, de nouvelles mesures (telles que des observations de nouvelles bornes limites) sont incorporées pour filtrer et mettre à jour des données. Le processus répète ces étapes, et le robot met à jour continuellement sa position prévue [HU 04].

Hu et **Evant** dans [HU 04], proposent d’adapter les idées de la localisation par robot mobile appelée **M**onte **C**arlo **L**ocalization (**MCL**). **MCL** ne marche pas lorsque les nœuds sont statiques [DAT 06].

4. L’algorithme d’estimation de position

L’algorithme calcule des probabilités de distribution des positions possibles des nœuds. En fonction de la position précédente des nœuds et des observations provenant d’ancres déployées dans le réseau, l’algorithme est capable de filtrer les positions impossibles. Cet algorithme adapte l’algorithme de **Monte Carlo**, pour l'usage dans des applications de réseau de capteur mobiles.

Cependant, il y a des différences substantielles entre la localisation de robot et la localisation de nœud pour des réseaux de capteur [HU 04] :

1. Tandis que la localisation de robot localise un robot dans une carte prédéfinie, la localisation dans des réseaux de capteur fonctionne dans un espace libre ou un terrain sans carte prédéfinie.
2. De plus, un robot a relativement un bon contrôle et des connaissances probabilistes sur son mouvement dans une carte prédéfinie. Par contre, Un nœud capteur a typiquement peu ou pas de contrôle sur sa mobilité, et est ignorant de sa vitesse et direction.
3. Un robot peut obtenir l'information de rangement précise des bornes limites, mais un nœud capteur peut seulement apprendre qu'il est dans la portée radio.
4. Finalement, dans la localisation de robot, les différentes mesures sont intégrées multiplicativement, l'indépendance conditionnelle arrogante entre eux, et les poids d'échantillons doivent être normalisés après mise à jour, Dans cet algorithme, dû aux contraintes de puissance de calcul et de mémoire, nous adoptons une approche de filtrage dans laquelle chaque mesure peut être considérée indépendamment, et le poids de chaque échantillon est 0 ou 1.

5. La méthode ATM-Dist

Cette méthode représente une extension d'**AT-Dist** au contexte de mobilité en considérant des capteurs capables de mesurer les distances qui les séparent de leurs voisins. La première étape de cette technique d'approximation est identique à celle d'**ATM-Free** : lorsqu'un capteur X invoque le calcul de sa position, il définit les deux ensembles $N^1_\Delta(X)$ et $N^2_\Delta(X)$. X reçoit les positions des ancres à 1 et 2 sauts de lui, mais aussi les mesures de distance entre les voisins à 1 saut et les ancres à 2 sauts. Avec les ancres appartenant à son 1-voisinage, il mesure les distances exactes et, avec celles appartenant à son 2-voisinage, il estime les distances avec **SumDist**. Lorsque $|N^1_\Delta(X)| >= 2$, X cherche à obtenir une position exacte :

- si $|N^1_\Delta(X)| >=3$ et sans erreur de mesure, X peut utiliser la multilatération. Autrement, il est préférable d'utiliser la règle 3 de la méthode **Mur [SAA 06]** (donner lors de chapitre 3);
- si $|N^1_\Delta(X)| = 2$ et $|N^2_\Delta(X)| >= 1$ alors X cherche à appliquer les règles 1 ou 2, de la méthode **Mur [SAA 06]**.

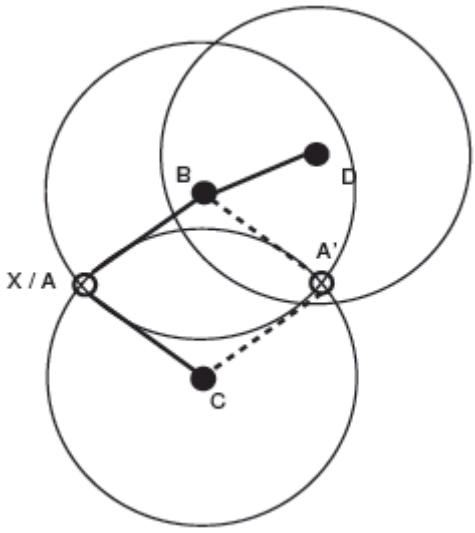


Figure C.5 : Règle 2 dans ATM-DIST.

La figure C.5 est un exemple de l'application de la règle 2 où X possède une ancre dans son 2-voisinage. L'ambiguïté de la position de X est due aux positions des ancres B, C Є $N^1_\Delta(X)$ et des distances entre XB et XC. Elle est résolue grâce à l'ancre C appartenant au voisinage de B.

Une position estimée sera calculée si X possède seulement une ancre dans son voisinage ($|N^1_\Delta(X)| = 1$) ou bien lorsque X n'obtient pas sa position exacte avec les règles 1, 2 et 3. Cette

position estimée est obtenue de la façon suivante : premièrement, X appartient aux cercles centrés en les ancres appartenant à $N^1_\Delta(X)$ de rayons les distances entre le nœud et les ancres, mais n'appartient à aucun disque centré en les ancres de $N^2_\Delta(X)$ de rayons r. X calcule le centre de gravité de la zone le contenant et en déduit une estimation de sa position.

Enfin, si aucune ancre n'appartient au voisinage de X ($N^1_\Delta(X) = Ø$), alors aucune position n'est calculée.

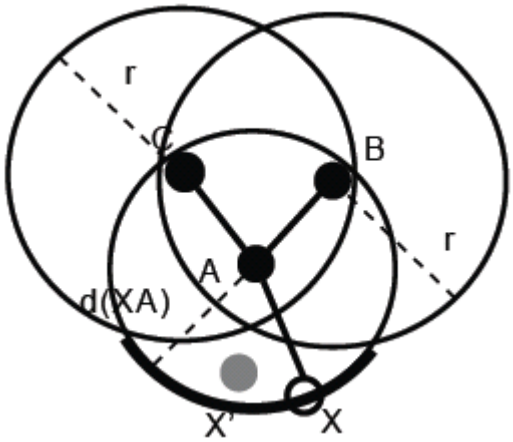


Figure C.6 : Estimation de la position de x dans ATM-DIST.

Exemple:

La figure C.6 est un exemple de la technique d'approximation énoncée : le capteur X appartient au cercle centré en A Є $N^1_\Delta(X)$ de rayon la distance d_{XA} mais n'appartient pas aux disques centrés respectivement en B et C de rayons r. Ainsi, X en déduit qu'il appartient à l'arc de cercle représenté en trait épais sur la figure. Le centre de gravité de cet arc représente une estimation de sa position.

6. La méthode ATM-Angle

Cette section décrit l'extension d'**AT-Angle** au contexte de mobilité où sont considérés des capteurs capables de mesurer les angles avec leurs nœuds voisins. La première étape de la technique d'approximation d'une position d'un nœud X reste identique à celle utilisée dans **ATM-Free**. Le capteur X définit ses deux ensembles $N^1_\Delta(X)$ et $N^2_\Delta(X)$. Les informations relatives aux angles formés entre le capteur et ses voisins à 1 saut ainsi que celles concernant les angles formés entre les voisins à 1 saut et les ancres à deux 2 sauts sont prises en compte.

Lorsque $|N^1_\Delta(X)| >= 2$, X est capable de déduire sa position exacte (en utilisant la technique définie dans la section présentant **AT-Angle**) : en connaissant les positions et les angles formés avec au moins deux ancres voisines, il est possible de calculer une position exacte.

Quand $|N^1_\Delta(X)| = 1$, X calcule une estimation de sa position. Il appartient au disque centré en l'ancre appartenant à $N^1_\Delta(X)$ de rayon r. En revanche, il n'appartient à aucun disque centré en une ancre de $N^2_\Delta(X)$ de rayon r. Ensuite, pour chacun des angles, il trace les droites correspondantes et, pour chacune d'elles, détermine son côté. En recoupant l'ensemble de ces informations, le capteur X estime sa position.

Enfin, s'il n'y a aucune ancre dans le voisinage de X ($N^1_\Delta(X) = Ø$), aucune position ne sera calculée.

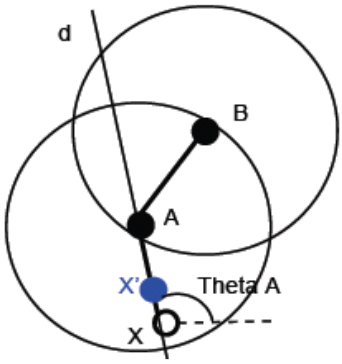


Figure C.7 : Estimation de la position de x dans ATM-ANGLE.

Exemple :

Dans l'exemple de la figure C.7, X utilise les informations des ancres A Є $N^1_\Delta(X)$ et B Є $N^2_\Delta(X)$. X appartient au disque de centre A de rayon r mais n'appartient pas à celui centré en B de rayon r. X connaît l'angle θ_A qu'il forme avec A et déduit qu'il appartient à la droite passant par A d'inclinaison θ_A. De ces informations, on obtient deux points d'intersection : le premier est défini entre la droite et le cercle centré en A et le de deuxième entre la droite et le cercle de centre B. Ces deux points d'intersection définissent un segment contenant X et son milieu X' correspond à l'estimation de la position de X.

REFRENCES

[AHM 05] Ahmed A, and al., "A New Approach to Relative Localization in Wireless Sensor Networks", 25th IEEE, International Conference on Distributed Computing Systems Workshops (2005).

[ARO 10] C. AROUS, D. HANNOU, S. MOUSSAOUI, "Étude comparative de méthodes de localisation dans les réseaux de capteurs sans fil", USTHB (Juillet 2010).

[ANE 99] P. Aneli, E. Horlait, "NS-2 : principes de conception et d'utilisation", Version 1.3, (15 Septembre 1999).

[BAC 05] J. Bachrach, C. Taylor, "Localization in Sensor Networks", (2005).

[BAS 07] C. Başaran, "A Hybrid Localization Algorithm for Wireless Sensor Networks", Master's Thesis, Yeditepe University, Turkey (2007).

[BAS 08] C. Başaran, & al., "A Hybrid Localization Algorithm for Wireless Sensor Networks", Institute of Electronics, Information and Communication Engineers-IEICE (2008).

[BEN 08] E. Ben Hamida, Guillaume Chelius, Jean-Marie Gorce, "On the Complexity of an Accurate and Precise Performance Evaluation of Wireless Networks Using Simulations", MSWiM'08, Vancouver, BC, Canada (October 27–31, 2008).

[BEN 09] BENKOUIDER Sarah, "ÉTUDE DU PROBLÈME DE LOCALISATION DANS LES RÉSEAUX VANET", Thèse du magistère N° d'ordre : 04/2009-M/INF à USTHB, (Soutenu le 22/12/2009).

[BEN 10] Benkhelifa I., "Étude de la localisation dans les réseaux de capteur", École doctorale Département Informatique U.S.T.H.B, (2010).

[BER 02] P. Bergamo, G. Mazzini, "Localization in sensor networks with fading and mobility", in: The 13th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, vol. 2, pp. 750–754, (2002).

[BEN 05] Abdelaziz BEN MANSOUR, "Etude et simulation du protocole SIP entre Softswitch et un serveur d'application dans un concept IMS".

[BOU 07] Mathieu Bouet, Erwan Ermel, Guy Pujolle, "Performances d'une méthode de localization dans les réseaux sans fil mobiles", Journées Doctorales en Informatique et Réseaux (JDIR 2007), Marne la Vallée, France, pp. 69-77.

[BUL 00] N. Bulusu, J. Heidemann, D. Estrin, "GPS-less low-cost outdoor localization for very small devices", IEEE Personal Communications 7 (5) 28–34 (2000).

[CAR 03] Julien CARTIGNY, "Contributions à la diffusion dans les réseaux Ad Hoc", Université des Sciences et Technologies de Lille, LIFL - UMR 8022 – Cité Scientifiqueque, Bat. M3 - 59655 Villeneuve d'Ascq Cedex, (19 Décembre 2003).

[CAP 09] Capteur, http://fr.wikipedia.org/wiki/Capteur, (2009).

[CHA 09] Chalhoub Gérard, "Réseaux de capteurs sans fil", (2009).

[CHA 08] YACINE CHALLAL, "Réseaux de capteurs sans fil", (17/11/2008).

[CHA 10A] Julien Champ, Vincent Boudet, "ADNL : Protocole de Localisation Distribué pour Réseaux de Capteurs Sans-fils", Journées Doctorales en Informatique et Réseaux (JDIR'10), Sophia Antipolis, France (Mars 2010).

[CHA 10B] Julien Champ, Vincent Boudet, "ADNL: accurate distributed node localization algorithm in wireless sensor networks", European Wireless, Lucca, Italy, (April 2010).

[CHA 10C] Julien Champ, Vincent Boudet, "ADNL-Angle: Accurate Distributed Node

Localization for Wireless Sensor Networks with Angle of Arrival Information". AD-HOC NOW 2010, Edmonton, Canada (August 2010).

[CHE 08] Hongyang Chen, Kaoru Sezaki, Ping Deng, Hing Cheung So, "An Improved DV-Hop Localization Algorithm for Wireless Sensor Networks", IEEE (2008).

[DAT 06] S. Datta, C. Klinowski, M. Rudafshani, S. Khaleque, "Distributed localization in static and mobile sensor networks". 2nd IEEE International conference on wireless and mobile computing, networking and communications (WiMob), (2006).

[DOH 01] L. Doherty, K. Pister, L. El Ghaoui, "Convex position estimation in wireless sensor networks", in: IEEE INFOCOM, vol. 3, pp. 1655–1663 (2001).

[EFR 06] Efrat A. and al., "Force-Directed Approaches to Sensor Localization". Proceedings of the 8th SIAM Workshop on Algorithm Engineering and Experiments (ALENEX) (2006),

[ELM 10] Elmurod A.Talipov, http://elmurod.net/?p=157.

[EVE 07] Evennou Frédéric, "Techniques et technologies de localisation avancées pour terminaux mobiles dans les environnements indoor", TECH/IDEA/iROC, (22 Janvier 2007).

[FLE 04] E. Fleury, David simplot-ryl, "Réseaux de capteurs", CITI/INSA de Lyon- ARES /INRIA, (16 décembre 2004).

[GAL 07] Gallais A. et autres, "La k-couverture de surface dans les réseaux de capteurs". IRCICA/LIFL, Univ. Lille 1, INRIA, (2007).

[GIR 05] Girod L., "A Self-Calibrating System of Distributed Acoustic Arrays". Ph.D. Thesis, UCLA, USA (2005).

[GOU 04] Gouda M. G. et autres, 'Sentries and Sleepers in Sensor Networks".

[HE 05] TIAN HE, CHENGDU HUANG, BRIAN M. BLUM, JOHN A. STANKOVIC, and TAREK F. ABDELZAHER, "Range-Free Localization and Its Impact on Large Scale Sensor Networks", ACM Transactions on Embedded Computing Systems, Vol. 4, No. 4, (November 2005).

[HEU 08] Karel Heurtefeux, Fabrice Valois, "Localisation collaborative pour réseaux de capteurs", Colloque Francophone sur l'Ingénierie des Protocoles (CFIP), Les Arcs : France, (2008).

[HEU 09A] Karel Heurtefeux, "Protocoles localisés pour réseaux de capteurs". INRIA SWING / CITI, INSA-Lyon, (26 Novembre 2009).

[HEU 09B] Karel Heurtefeux, Fabrice Valois, "Localisation distribuée pour routage en environnement bruité dans les réseaux de capteurs". 'CFIP'2009 (2009)'.

[HON 09] Paul HONEINE, Cédric RICHARD, Hichem SNOUSSI, Mehdi ESSOLOH, 'Auto-localisation dans les réseaux de capteurs sans fil par régression de matrices de Gram ". Laboratoire LM2S, Institut Charles Delaunay (FRE CNRS 2848). (2009).

[HEN 08] Héni JMEL. Maxime CAUDRON. Aurélien BRISSET et Pierre Marc GUITARD. 'PROJET AVANCE SE : RESEAU DE CAPTEURS SANS FILS", ENSEIRB (2008).

[HU 04] Lingxuan Hu, David Evans, "Localization for Mobile Sensor Networks". In Tenth Annual International Conference on Mobile Computing and Networking (MobiCom 2004), Philadelphia, (26 September - 1 October 2004).

[KAC 09] Kacimi R, “Techniques de conservation d'énergie pour les réseaux de capteurs sans fil”, Thèse de doctorat, à l'INPT, (soutenue le 28-09-2009).

[KHE 04] Lyes KHELLADI & Nadjib BADACHE, “Rapport de recherche : Les réseaux de capteurs : état de l’art”. N° LSI-TR0304, (Février 2004).

[KWO 04] C. Kwok, D. Fox, M. Meila, “Real-time particle filters”, Proceedings of the IEEE 92 (3) 469–484 (2004).

[LAN 03] Koen Langendoen, Niels Reijers, “Distributed localization in wireless sensor networks: a quantitative comparison”. Faculty of Information Technology and Systems, Delft University of Technology, 2828 CD Delft, Netherlands. Computer Networks 43 499–518 (2003).

[MAI 02] Mainwaring A. and al., “Wireless Sensor Networks for Habitat Monitoring”. WSNA '02 Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications, (2002).

[MAN 01] Manjeshwar, A. ; Agrawal, D.P. “TEEN: a routing protocol for enhanced efficiency in wireless sensor networks”, International proceedings of 15th parallel and distributed processing syposium, page 2009-2015, (2001).

[MAO 07] G. Mao, B. Fidan, and B. D. O. Anderson, “Wireless sensor network localization techniques”, to appear in Computer Networks, (3 January 2007).

[NAG 03] R. Nagpal, H. E. Shrobe, and J. Bachrach, “Organizing a global coordinate system from local information on an ad hoc sensor network”, in 2nd International Symposium on Information Processing in Sensor Networks (IPSN), Palo Alto, CA, USA, (April 2003).

[NEL 02] Nelson J. and al., “Fine-Grained network time synchronization using reference broadcast”, Proceedings of the 5th Symposium on Operating Systems Design and Implementation, (2002).

[NIC 01] Dragos. Niculescu, B. Nath, “Ad hoc positioning system (APS)’, in: IEEE GLOBECOM, vol. 5, pp. 2926–2931 (2001).

[NIC 02] Dragos Niculescu and B. Nath, “DV based positioning in ad hoc networks,” Telecommunication Systems, Division of Computer and Information Sciences, 27th (August 2002).

[NIC 03] Dragos Niculescu, B. Nath, “Ad Hoc Positioning System (APS) Using AOA”. 0-7803-7753-2/03/, IEEE (2003).

[OHH 08] Oh-Heum K., and Ha-Joo S., “Localization through map stitching in wireless sensor networks’, IEEE Transactions on Parallel and Distributed Systems. 2008.

[POT 00] Gregory J. Pottie, ‘Wireless integrated network sensors”. Electrical Engineering Department. University of California, Los Angeles, (2000),

[PRI 03] Priyantha N.B. and al., “Anchor-Free Distributed Localization”, MIT Laboratory for Computer Science. http://nms.lcs.mit.edu/cricket/, (2003).

[QUN 04] Qun Li and Daniela Rus, “Global Clock Synchronization in Sensor Networks”. Department of Computer Science, Dartmouth College Hanover, NH 03755 IEEE (2004).

[RAB 00] Rabiner W., and al., “Energy efficient communication protocol for wirelessmicrosensor network”, Proceedings of the 33rd Hawaii International Conference on System Sciences (2000).

[RAB 04] M. Rabbat, R. Nowak, “Distributed optimization in sensor networks”, in: Third

International Symposium on Information Processing in Sensor Networks, pp. 20–27, (2004).

[RAG 02] Raghunathan V. and al., “Energy-aware wireless microsensor networks”. IEEE SIGNAL PROCESSING MAGAZINE, (2002).

[RAY 92] Roy Want. Andy Hopper. Veronica Falcão. Jonathan Gibbons. ’ The Active Badge Location System”, Olivetti Research Ltd. (ORL). Cambridge, England 1992.

[ROS 04] Francisco J. Ros. Pedro M. Ruiz, “Implementing a New Manet Unicast Routing Protocol in NS2”, University of Murcia, (December 2004).

[SAA 06] Saad Clément, “Une nouvelle méthode de positionnement dans les réseaux de capteurs”, Algotel 2006, 8eme rencontres francophones sur les aspects algorithmiques de télécommunication, Trégastel, France, (Mai 2006).

[SAA 08] Saad Clément, “Quelques contributions dans les réseaux de capteurs sans fil : Localisation et Routage”, (Soutenue publiquement le 10 juillet 2008).

[SAV 01] C. Savarese, J. Rabaey, and J. Beutel, “Locationing in distributed ad-hoc wireless sensor networks”, In IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP), pages 2037–2040, Salt Lake City, UT, (May 2001).

[SAV 02] A. Savvides, H. Park, and M. Srivastava, “The bits and flops of the n-hop multilateration primitive for node localization problems”, First ACM International Workshop on Wireless Sensor Networks and Application (WSNA), Atlanta, GA, pp. 112-121, (2002).

[SAV 03] A. Savvides, W. Garber, S. Adlakha, R. Moses, and M. B. Srivastava, “On the Error Characteristics of Multihop Node Localization in Ad-Hoc Sensor Networks”, in Proceedings of the 2nd International Workshop on Information Processing in Sensor Networks (IPSN’03), (2003).

[SAVI 03] ANDREAS SAVVIDES, HEEMIN PARK, MANI B. SRIVASTAVA, “The n-Hop Multilateration Primitive for Node Localization Problems”, Networked and Embedded Systems Lab, Electrical Engineering Department, University of California, Los Angeles, 6731-H Boelter Hall, Box 951594, Los Angeles, CA 90095-1594, USA (2003).

[SAV 05] A. Savvides, W.L. Garber, R.L. Moses, M.B. Srivastava, “An analysis of error inducing parameters in multihop sensor node localization”, IEEE Transactions on Mobile Computing 4 (6) 567–577 (2005).

[SHA 03] Y. Shang, W. Ruml, Y. Zhang, and M. Fromherz, “Localization from mere connectivity”, in Proc. 4th ACM Intl. Symp. on Mobile Ad Hoc Networking and Computing (MobiHoc), Annapolis, MD, (2003).

[SHA 04] Y. Shang, W. Ruml, Y. Zhang, M. Fromherz, “Localization from connectivity in sensor networks”, IEEE Transactions on Parallel and Distributed Systems 15 (11) 961–974 (2004).

[SHN 05] Victor Shnayder, Borrong Chen, Konrad Lorincz, Thaddeus R, FulfordJones et Matt Welsh, “Sensor Networks for Medical Care”, Computer Science Group, Harvard University, Cambridge, Masachusetts, (2005).

[SIC 04] Sichitiu M.L. and Ramadurai V., “Localization of wireless sensor networks with a mobile beacon”, Center for Advanced Computing and Communications (CACC), Raleigh, NC, Tech. Rep. TR-03/06, (2004).

[SIL 11] SILMI S., MOUSSAOUI S., " Localisation dans les réseaux de capteurs sans fil", Rapport Interne, Laboratoire LSI-Usthb, LSI-TR-01-11, (jan.2011).

[SLI 02] Slijepcevic S. and al., "Location errors in wireless embedded sensor networks: sources, models, and effects on applications",, Mobile Computing and Communications Review 6 67–78. 2002.

[STA 06] Stankovic J. A., "Wireless Sensor Networks", ACM SenSys, (Nov 2006).

[STO 00] Stojmenovic I. and al., "Depth first search and location based localized routing and QoS routing in wireless networks", In Proceedings of the IEEE International Conference on Parallel Processing, (2000).

[STO 05] Stojmenovic I., "Handbook of sensor networks: algorithms and architectures", University of Ottawa, (2005).

[SUB 07] Subramanian S. & al, "On Optimal Geographic Routing in Wireless Networks with Holes and Non-Uniform Traffic", (2007).

[TIA 03] Tian He & al., "SPEED: A Stateless Protocol for Real-Time Communication in Sensor Networks", (2003).

[TIL 05] S. Tilak, V. Kolar, N. B. Abu-Ghazaleh, and K. D. Kang, "Dynamic localization control for mobile sensor networks", in IEEE IWSEEASN, (2005).

[TIN 11] TinyOs, http://fr.wikipedia.org/wiki/TinyOS. 2011.

[TSU 05] James Bao-Yen Tsui, vFundamentals of Global Positioning System receivers - a software approach", Wiley, (2005).

[WOO 02] Wood A. D. and Stankovic J. A, "Denial of service in sensor networks", IEEE (2002).

[XIA 04] Xiaoli Li and al., "A map-growing loca'lization algorithm for ad-hoc wireless sensor networks", ICPADS '04 Proceedings of the Parallel and Distributed Systems, Tenth International Conference, (2004).

[XIA 07] Bin Xiao. Hekang Chen. Shuigeng Zhou, "A Walking Beacon-Assisted Localization in Wireless Sensor Networks", (ICC 2007 proceedings).

[YAX 01] Ya Xu et autres, "Geography-informed energy conservation or ad hoc routing", USC/Information Sciences Institute, (2001).

[YIS 04] Yi Sang and al., "A new algortihm for relative localization in wireless", International Parallel (and Distributed) Processing Symposium (IP(D)PS) (2004).

[ZAF 08] Youcef ZAFOUNE, Aicha MOKHTARI, Souhila SILMI, "Towards a Distributed Form of Centralized Approach for Mobile Agents Localization in Ad hoc Networks", The 2nd International Workshop on Cooperative Wireless Communications and Networking (CONET 2008),London, UK, (8-11 September 2008).