

11A 2004 / 2016 - 1

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE



UNIVERSITE DE saad dahlab blida
FACULTE DES MATHÉMATIQUES ET DE L'INFORMATIQUE

Département de l'informatique

MEMOIRE de fin d'étude

Présenté pour l'obtention du diplôme de MASTER

Domaine: Mathématiques et Informatique

Filière: Informatique

Option : GSI

SUJET

Localisation des nœuds dans un réseau de capteurs sans fil

Réalisé par :

BOUKOFTANE Djamila.

TERCHI kouather .

Dirigé par: **Mr BENYAHIA Mouhamed**

Organisme d'accueil : Algérie Télécom.

Encadreur : **Mr. ZEROUKI.**

Président de jury: **MBOUTOUMI.**

2016 /2017

MA-004-506-1

Dédicace :

A ma très chère mère,

BENSAADIA Meriam qui a œuvré pour ma réussite, de par son amour, son soutien, tous les sacrifices consentis et ses précieux conseils.

A mon père de la plus chère du monde

TERCHI Derradji de privations pour m'aider à avancer dans la vie. Puisse Dieu faire en sorte que ce travail porte son fruit ; Merci pour les valeurs nobles, l'éducation et le soutien permanent venu de toi.

Mes frères (fouad, yacine, mouhamed) et

mes sœurs (nassima, sabrina, souad, hadjer, amel, soumia) qui n'ont cessé d'être pour moi des exemples de persévérance, de courage et de générosité.

A Chéri MON Mari miloudi nourddine Qui m'a encouragé à poursuivre mes études

A mes chers Grandson es de ma famille tadj-elddine, insaf, abdelrrahman, mouhamed yacine, ayham fadi, choaibe, tinhinane

A Mes chers amis sabrina, kamal, maroua, bouchra, amel, imane, madina,

A ma collègue de travail ma chère amie mon binôme djamila (jojo)

Mes enseignions de GSI et de Msila

A tous les gens m'aiment.....

{Kawther}



Je dédie ce mémoire à

Au meilleur des pères dans ce monde

A ma Chère Mère la plus belle créature que Dieu a créée sur terre , À cet source de tendresse

A mon Père Dont le mérite, les sacrifices et les qualités humaines m'ont permis de vivre ce jour et Qu'ils trouvent en moi la source de leur fierté A qui je dois tout

. A mes frères (samir ,halime,billal, mouhamed)

Et ma sœur (naima)

*Les mots ne suffisent guère pour exprimer l'attachement, l'amour
Et l'affection que je porte pour vous*

A mes enseignions de MI généralement et de GSI spécialement Qui nous a donné tout leur temps pour notre éducation

A mes amis

Kamal, kawther, marwa,nihad, amina ,ilhem,adlen

À tous mes collègues À tous les étudiants de la promotion 2012/2013 filière informatique

A ma sœur et mon binôme kawther.

A tous ceux qui, par un mot, m'ont donné la force de continuer

{Djamila}

Remerciements

Pour nous 'avoir permis d'être ce que nous sommes devenu aujourd'hui, nous voudrions remercier le SEIGNEUR des mondes par qui tout est possible : DIEU

Nous 'adressons nos remerciements les plus sincères, au notre directeur de recherches monsieur Mouhamed BENYAHIA, qui a très volontiers accepté d'être le promoteur de ce projet. Pour sa patience, et surtout pour sa confiance, ses remarques et ses conseils, sa disponibilité et sa bienveillance. Qu'il trouve ici le témoignage de notre profonde gratitude. Ainsi que son expérience, ont joué un rôle important dans la conception de ce travail.

La sollicitude du monsieur l'encadreur Abdelkader ZARROUKI, Chef du département de corporal / de IGE au cours de cette année, nous 'a profondément touché. L'aboutissement de ce travail nous donne l'occasion de lui exprimer notre très sincère reconnaissance. Merci pour l'accueil dans la société Algérie télécom SPA de blida

Nous remercier également notre professeurs pour la qualité de l'enseignement qu'ils nous 'ont prodigué au cours des années passées à l'université de blida département de l'informatique

Nous voudrions également remercier les membres du jury pour avoir accepté d'évaluer ce travail et pour toutes leurs remarques et critiques.

Merci pour tous ceux qui nous ont aidés de près ou de loin, nous voulons remercier pour chaque mot ou service vous nous avez donné

Merci à tous

Résumé

Un RCSF est un ensemble autonome de nœuds capteurs dédiés pour capter des mesures d'un phénomène physique dans une zone d'intérêt, En l'absence d'information sur la position des nœuds d'un réseau de capteurs sans fil, au sein de l'environnement où ils sont déployés, les données récoltées peuvent s'avérer d'une utilité limitée. Le problème traité concerne l'auto-localisation de chacun de ces nœuds à partir de mesures de portée inter-capteurs telles que les RSSI, et de quelques capteurs dits ancres dont la position est connue.

Dans ce travail, nous allons se focaliser sur l'implémentation d'un protocole de localisation basé sur des algorithmes capables de localiser tous les nœuds dans un réseau de capteurs sans fil, et de faire l'analyse des performances de ce protocole par sa simulation établie avec le network simulateur 2 (NS2).

Mots-clés: Ancre, estimation de distance, estimation de position, RSSI, NS2, simulation, RCSFs, localisation des nœuds dans un réseau de capteurs, GPS, LTGA, LTGA mobile, processus de localisation, Range-based, Range-free.

Abstract

A WSN is a self-contained set of nodes sensors dedicated to capture measures of a physical phenomenon in an area of interest, in the absence of information on the position of the nodes of a wireless sensor network, within the environment where they are deployed, the data collected may be of limited usefulness. The problem treated concerned the self-localization of each of these nodes from measurements of scope inter-sensors such as the RSSI, and a few so-called sensors anchors whose position is known.

In this work, we will focus on the implementation of a protocol of localization based on algorithms that can locate all nodes in a network of wireless sensors, and to make the analysis of the performance of this Protocol by its simulation established with the network simulator 2 (NS2).

Keywords: Anchor, estimation of distance, location estimate, RSSI, NS2, simulation, RCSFs, location of nodes in a network of sensors, GPS, LTGA, LTGA mobile, localization process, Range-based and Range-free.

ملخص

شبكة الالتقاط اللاسلكي هي مجموعة من العقد من أجل التقاط ظواهر فزيائية في محيط مدرّوس، في غياب المعلومات عن احداثيات العقدة تكون البيانات الملتقطة غير مفيدة، المشكل المعالج هو كيفية تحديد احداثيات العقد عن طريق قياس نطاق ما بين العقد ك قياس شدة الإشارة، مع مجموعة من العقد التي تعرف احداثياتها تسمى بالعقد المرساة

في عملنا سنركز على انشاء بروتوكول يعتمد على خوارزميات قادرة على تحديد احداثيات كل العقد في شبكة الالتقاط اللاسلكي مع القيام بتحليل للأداء هذا البروتوكول بمحاكاة منجزة من طرف NS2.

الكلمات المفتاحية :

تحديد الالتقاط اللاسلكي محاكاة، احداثيات العقد في شبكة، NS2، RSSI، مرساة، تقريب البعد، تقريب المركز،

اجراء تحديد المركز:متحرك، المدى مجاني، المدى المعتمد، LTGA، LTGA، GPS، WSN، RCSF

Liste d'abréviation :

RCSF : réseaux de capture sans fils

GPS : Le Global Positioning System

ADC : le convertisseur analogique-numérique

WATS : Wide Area Tracking System

MAC : Média Access Control

ESRT: Event-to-Sink Reliable Transport protocol

PSFQ : Pump Slowly, Fetch quickly

MANET : Mobile Ad hoc NETWORKS

CH: Cluster Head

AoA : *Angle of Arrival*

RSSI : Received Signal Strength Indicator

DoA: Direction of Arrival

TDOA : Time Différence Of Arrival

RHC : *Radio Hop Count*

LTTA : Localisation Trilatération algorithme

LTGA : Triangulation algorithm nommées

NS-2 : Network Simulator 2

Table des matières :

• Dédicace.....	1-2
• Remerciement.....	3
• Résumé.....	4
• Liste des abréviations.....	5
• Table des matières.....	6-10
• Liste des figures.....	11-12
• Liste des tableaux	13
• Introduction général	14-15
Chapitre 01 : les réseaux de capteurs sans fil.....	16
1 Introduction	17
2 Un capteur sans fil	18
• 5.1 Définitions	18
• 5.2 Architecture d'un capteur	18
5.2.1 L'unité d'acquisition	18
5.2.2 L'unité de traitement	18
5.2.3 L'unité de communication.....	19
5.2.4 Unité d'énergie.....	19
3- Réseau de capteurs sans fil :.....	19
4- Les Domaines d'application des RCSFs	20
4.1 Découvertes de catastrophes naturelles.....	20
4.2 Commerciales	20
4.3 Agriculture	20
4.4 Le domaine militaire.....	21
4.5 Le bâtiment.....	21
4.6 Domaine médecine.....	21

Table de matières

4.7-Applications domestiques	22
5-Facteurs et contraintes des réseaux de capteurs sans fil.....	23
6- Architecture de communication	24
9.1-La couche physique	25
9.2 Couche liaison de données.....	25
9.3-couche réseau	25
9.4-Couche transport	25
9.5-couche application.....	26
7-Domains de recherche sur les réseaux de capteurs sans fil	26
7.1 La consommation énergétique	26
7.2 Routage et clustering	27
7.3 La Sécurité.....	27
7.4 Tolérance aux pannes.....	27
7.5 Agrégation des donnés	28
7.6 Localisation.....	28
8- conclusion.....	29

Chapitre 02 : Localisation dans les réseaux de capteurs sans fils30

1 Introduction	31
2 Définition de la localisation	32
3 Le processus de localisation dans les RCSFs.....	34
3.1 Définition d'un système de coordonnées.....	32
3.1.1 Les ancrés.....	33
3.1.2 Types des ancrés	33
3.2 Estimation des distances.....	34
3.2.1 Informations disponibles pour la localisation.....	35
- Puissance du signal radio	35
- Equation de Friis	35

Table de matières

- Propagation avec réflexion sur le sol ou Two Ray Ground reflection model...	35
- Modèle de Shadowing	36
3.2.2 Techniques de mesure d'estimation de la distance pour range-based.....	37
• Indicateur de puissance de signal reçu (RSSI).....	37
• Angle d'arrivée AoA	38
• Temps d'arrivée TOA	38
• Différence des temps d'arrivée TDOA	38
3.2.3 Techniques de mesure d'estimation de la distance pour range-free.....	39
- Le nombre de sauts – RHC (Radio Hop Count)	39
3.2.4 Analyse et discussion	41
3.3 Technique Dérivation des positions.	42
3.3.1 Trilatération.....	42
3.3.2 Multilatération	43
3.3.3 Le principe de Triangulation	44
4-Classification des approches de localisation.....	45
• 4.1 Les approches directes	45
• 4.2 Les approches indirectes.....	46
7-Algorithmes de localisation.....	46
• 7.1- Algorithmes centralisés.....	46
• 7.2-Algorithmes distribués.....	46
8-Critères des algorithmes de localisation	49
9- Conclusion	50
Chapitre 03 : Conception & implémentation.....	51
1 Introduction.....	52
2 Algorithme de localisation.....	53
2.1 Choix métrique d'estimation de distance	53
2.2 Choix métriques d'estimation de position et Implémentation les algorithmes de localisation.....	53

Table de matières

2.2.1 Trilatération	54
- Définition	54
- Algorithme de localisation basée sur LTTA	54
2.2.2 Triangulation.....	55
- Définition	55
- Algorithme de localisation basée sur LTGA.....	55
2.2.3 Comparaison entre les deux méthodes.....	56
2.3- méthode proposée	56
2.4 Utilisation des nœuds ancre mobile	57
3-Implémentation	60
3.1 NS2 network simulator 2.....	58
3.2 Les étapes d'implémentation.....	59
3.3 Implémentation un nouveau protocole de localisation pour RCSF.....	61
3.4 Implémentation la phase d'estimation de distance RSSI.....	61
3.5 L'implémentation de LTTA & de LTGA.....	62
3.6 Implémentation de notre proposition.....	63
3.7 Implémentation des mobiles ancre	63
4 Conclusion.....	67
Chapitre 04 : Interprétation des résultats.....	68
1- Introduction.....	69
2- Intérêt et nécessité de la simulation.....	70
2.1-Les Métriques de Performance.....	70
2.2- les paramètres utilisés dans les simulations	70
2.3- Paramètres de simulation de notre protocole de localisation	71
3-Simulation et discussion	71
3.1-Résultat d'algorithme LTTA.....	72
3.2-Résultat d'algorithme LTGA	72
3.3-Comparaison entre LLTA et LTGA	73

Table de matières

3.4-Distribution des nœuds Ancre	73
- Comparaison entre topologie fixée et topologie aléatoire	74
3.5-Résultat de proposition	73
- Comparaison entre LLTA et LTGA avec la proposition	73
1-Le taux d'erreur	75
3.6-Simulation des ancres mobiles	76
4 –Conclusion.....	77
• Conclusion générale	79
• Annexe	
• Bibliographie	

Liste des figures :

Figure 1.2 Architecture physique des nœuds capteurs sans fils.....	19
Figure 1.4 Application militaire.....	21
Figure 1.5 Application médecine	22
Figure 1.6 Application des RCSFs.....	23
Figure 1.7 Modèle en couches du réseau de capteurs sans fil.....	25
Chapitre 2	
Figure 2.1 : Exemple d'un nœud ancre.....	36
Figure 2.2 : Propagation avec une réflexion.....	38
Figure 2.3: comparaison de coûts des différentes techniques de localisation.....	42
Figure 2.4: comparaison de précision des différentes techniques de localisation.....	43
Figure 2.5: comparaison de consommation d'énergie des différentes techniques de localisation.....	43
Figure 2.6: Trilatération.....	44
Figure 2.7 : Angle d'incidence et position.....	46
Figure 2.8 : les approches de localisation.....	50
Chapitre 3	
Figure 3. 1 : Principe des méthodes de localisation par satellite (en 2D).....	52
Figure 3.2 : Pseudo-code pour la Algorithme de localisation LTTA.....	53
Figure 3.3 : Pseudo-code sur la Proposition d'Algorithme de localisation.....	60
Figure 3.4: Une seule ancre mobile pour aider les nœuds à s'auto-localiser.....	58
Figure 3.5 : NS2 schémas.....	59
Figure 3.6 : estimation de distance.....	62
Figure 3.7 : l'implémentation de LTTA avec C++.....	62
Figure 3.8 : l'implémentation de LTGA.....	63
Figure 3.9: l'implémentation de notre proposition	63
Figure 3.10 : la mobilité des nœuds ancrés par utilisation de setdest.....	64
Figure 3.11 : utilisation des timers.....	65

Liste des figures

Figure 3.12: diffusion de messages par nœud 0.....	65
Figure 3.13 : ajouter la bibliothèque Mobilenode.h.....	66
Figure 3.14 : ajouter la déclaration.....	66
Chapitre 4	
Figure 4.1: résultats de simulation de l'algorithme LTTA.....	72
Figure 4.2: résultats de simulation de l'algorithme LTGA.....	72
Figure 4.3 : comparaison de performance entre LLTA & LTGA.....	73
Figure 4.4: positionnement aléatoire des nœuds ancrés	74
Figure 4.5: positionnement organisées des nœuds ancrés.....	74
Figure 4.6 : résultats de simulation de la proposition.....	74
Figure 4.7 : comparaison en nombre des nœuds localisée.....	75
Figure 4.9 : les résultats de simulation par l'utilisation de nœud ancre.....	76

Liste des tableaux :

Chapitre 4 :

Tableau 4.1: Paramètres de simulation standards utilisés par NS2

..... .75-76

Tableau 4.2: les paramètres de simulation d'LOCP.....76

Introduction général

Introduction général :

L'apparition des nouvelles technologies ainsi que les progrès effectués dans le domaine des réseaux et du traitement de l'information ont entraîné l'apparition de nouveaux outils et objets tels que les réseaux de capteurs et leurs applications. Depuis quelques années, les réseaux de capteurs sans fil (RCSF) s'ouvrent à plusieurs domaines d'applications : militaire, sécurité civile (surveillance des risques d'incendie, des catastrophes naturelles, des centrales nucléaires, ...), transport (automobile, ferroviaire, aéronautique, spatial), industriel (contrôle de la qualité de production, surveillance des lieux, ...), environnement, etc. L'environnement intègre de plus en plus des capteurs pour le contrôle, la commande ou la surveillance des lieux ou des systèmes.

Donc en va donner d'abord une vue générale sur les réseaux de capteurs, leurs architectures de communication et leurs applications. Nous allons discuter également les principaux facteurs et contraintes qui influencent la conception des réseaux de capteurs sans fil.

Dans plusieurs domaines d'application des réseaux de capteurs on utilise le GPS pour déterminer l'information sur la position des capteurs, mais si on équipe chaque capteur avec dispositif GPS l'opération devient coûteuse et nécessiterait une consommation d'énergie à laquelle il faut prêter attention (car les capteurs ont une source d'énergie limitée), pour cela on essaie de minimiser le cout , et de trouver une solution pour la consommation d'énergie .Et notre souci primordial est de trouver un moyen pour localiser les capteurs sans utiliser le système GPS .

L'objectif de notre travail consiste à localiser les nœuds dans un réseau de capteurs sans fil sans utiliser le GPS parce que l'opération est coûteuse du point de vue financier comme du point de vue énergétique donc on essaie de localiser tous les nœuds avec une grande précision et ce à partir des positions connues de seulement quelques capteurs spécifiques (ancres) à l'aide des algorithmes de localisation des nœuds.

La suite de ce document est constituée de 4 chapitres :

Chapitre 1: décrit l'architecture d'un capteur et présente les principes et les caractéristiques des réseaux de capteurs aussi que ses domaines d'application, facteurs et contraintes de RCSF et citer les domaines de recherche sur les réseaux de capteurs sans fil.

Introduction général

Chapitre 2 : nous présentons quelques méthodes d'auto-localisation des nœuds dans un réseau de capteur sans fil (RCSFs)

Chapitre 3 : nous présentons nos algorithmes de localisation et les outils de développement et la phase d'implémentation de cet algorithme.

Chapitre 4 : nous avons présenté les résultats des algorithmes LTTA & LTGA et notre algorithme proposé avec l'étude du positionnement des nœuds ancre.

Chapitre 01 :

Les réseaux de capteurs sans fil (RCSFs).

1-Introduction :

Les réseaux de capteurs sans fil (RCSF) représentent un domaine en forte expansion, avec des applications très importantes dans le domaine de l'instrumentation, des économies d'énergie et de la gestion de l'environnement. D'ailleurs, en 2003 le magazine *Technology Review* du MIT¹, a considéré que le réseau de capteurs sans fil est l'une des dix nouvelles technologies qui bouleverseront le monde et notre manière de vivre et de travailler.

Leur importance provient aussi de leurs nombreux domaines d'application tels que la détection et la surveillance des désastres, le contrôle de l'environnement et la cartographie de la biodiversité, le bâtiment intelligent, l'agriculture de précision, la surveillance et la maintenance préventive des machines, la médecine et la santé, la logistique et les transports intelligents, etc..

Dans ce chapitre, nous décrirons les réseaux de capteurs, leurs caractéristiques ainsi que leurs domaines d'applications.

¹ *Technology Review* : est un magazine américain publié par Technology Review, une société de média appartenant au [Massachusetts Institute of Technology](http://www.massachusettsinstituteoftechnology.edu) . Le magazine d'une bimensuel sur ce site Internet : www.technologyreview.com .

2- Un capteur sans fil :

2.1-Définition :

Un capteur sans fil est un petit dispositif électronique capable de mesurer une valeur physique environnementale (température, lumière, pression, humidité, vibration, etc.), et de la communiquer à un centre de contrôle via une station de base. Chaque capteur assure les trois principales fonctions de base qui sont : l'acquisition de données, les traitements sur ces données et leurs communications aux stations de bases [1].

2.2-Architecture d'un capteur :

Suivant le type d'application, on distingue une multitude de capteurs: les capteurs de température, d'humidité, de pression, etc. Cependant, malgré cette diversité apparente, ils restent dotés d'une architecture matérielle similaire. Un capteur est composé principalement d'une unité de : acquisition, traitement, communication, et énergie tel que illustré sur la figure 1.1 Des composants additionnels peuvent être ajoutés selon le domaine d'application [2], comme par exemple un système de localisation tels qu'un GPS (Global Positioning System), un générateur d'énergie (exemple : cellules solaires) ou un mobilisateur lui permettant de se déplacer.

2.2.1 L'unité d'acquisition

Composée d'un ou plusieurs capteurs. Elle permet d'acquérir des mesures sur les paramètres environnementaux et les convertir en des signaux numériques par le biais d'un convertisseur ADC² puis les transmettre à l'unité de traitement.

2.2.2 L'unité de traitement

Composée d'un processeur et d'une mémoire intégrant un système d'exploitation spécifique comme TinyOS [3]. Cette unité possède deux interfaces, une interface avec l'unité d'acquisition et une interface avec l'unité de communication. Elle acquiert les informations en provenance de l'unité d'acquisition et les envoie à l'unité de communication. Cette unité est chargée aussi d'exécuter les protocoles de

² ADC : Analog-to-Digital Converter

communications qui permettent de faire collaborer un capteur avec d'autres capteurs. Elle peut aussi analyser les données captées et les traiter.

2.2.3 L'unité de communication

Unité responsable de toutes les émissions et réceptions de données via un support de communication radio. Elle peut être de type optique comme dans les capteurs Smart Dust [4], ou de type radiofréquence comme dans les capteurs MICA2 [5].

2.2.4 Unité d'énergie

Un capteur est muni d'une batterie ou de piles pour alimenter tous ses composants. Cependant, à cause de sa taille réduite, la batterie dont il dispose est limitée et généralement irremplaçable. Pour cela, l'énergie est la ressource la plus précieuse puisqu'elle influe directement sur la durée de vie des capteurs.

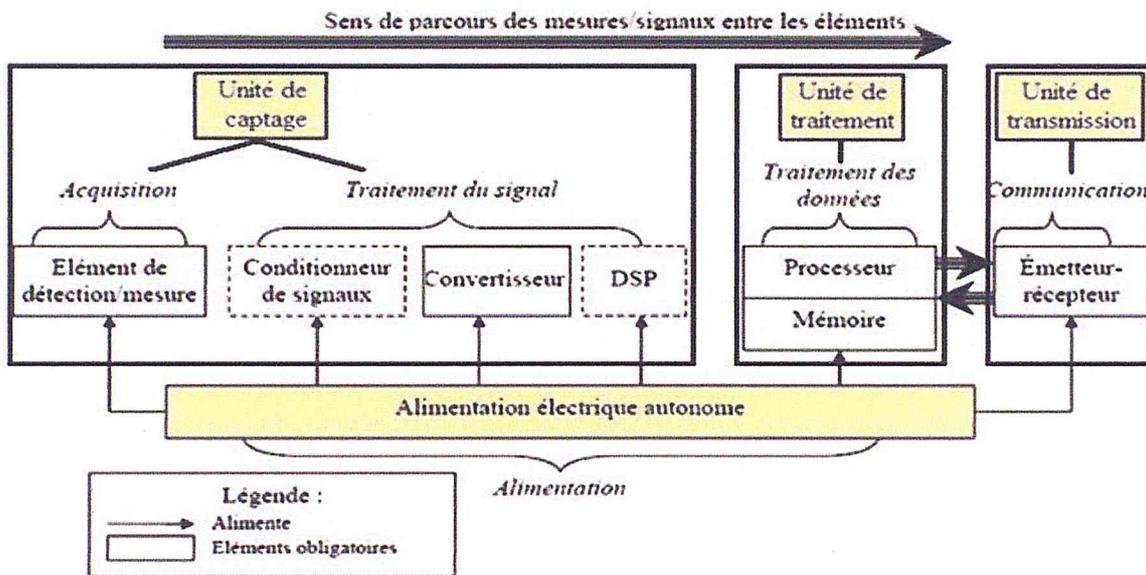


Figure 1.1 : Architecture physique des nœuds capteurs sans fils [6].

3- Réseau de capteurs sans fil :

Un **réseau de capteurs sans fil** est un réseau ad hoc utilisant un grand nombre de dispositifs très petits, nommés « nœuds » pour former un réseau sans infrastructure établie. Dans ce réseau chaque nœud est capable de détecter son environnement de traiter l'information au niveau local ou d'envoyer à un ou plusieurs points de collecte à l'aide d'une connexion sans fil [7]. La position de ces nœuds n'est pas obligatoirement prédéterminée. Ils peuvent être

aléatoirement dispersés dans une zone géographique, appelée « *champ de captage* » correspondant au terrain d'intérêt pour le phénomène capté.

4-Les Domaines d'application des RCSFs :

Les avantages qui caractérisent les réseaux de capteurs les permettent de s'adapter à plusieurs domaines d'applications. Parmi ces domaines où les réseaux de capteurs se révèlent très utiles et peuvent offrir de meilleures contributions, on peut noter :

4.1-Découvertes de catastrophes naturelles :

On peut créer un réseau autonome en dispersant les nœuds dans la nature. Des capteurs peuvent ainsi signaler des événements tels que feux de forêts, tempêtes ou inondations. Ceci permet une intervention beaucoup plus rapide et efficace des secours [8].

4.2 -commerciales :

Il est possible d'intégrer des capteurs au processus de stockage et de livraison dans le domaine commercial. Le réseau ainsi formé pourra être utilisé pour connaître la position, l'état et la direction d'un paquet. Il devient alors possible pour un client qui attend la réception d'un paquet, d'avoir un avis de livraison en temps réel et de connaître la localisation actuelle du paquet.

Pour les entreprises manufacturières, les réseaux de capteurs permettront de suivre le procédé de production à partir des matières premières jusqu'au produit final livré. Grâce aux réseaux de capteurs, les entreprises pourraient offrir une meilleure qualité de service tout en réduisant leurs coûts [9].

4.3- Agriculture :

Des nœuds peuvent être incorporés dans la terre. On peut ensuite questionner le réseau de capteurs sur l'état du champ (déterminer par exemple les secteurs les plus secs afin de les arroser en priorité). On peut aussi imaginer équiper des troupeaux de bétail de capteurs pour connaître en tout temps, leur position ce qui éviterait aux éleveurs d'avoir recours à des chiens de berger [10].

4.4- le domaine militaire :

Dans un domaine important comme le militaire, il faut utiliser des technologies de communication efficaces telle que les réseaux de capteurs qui sont caractérisés par la fiabilité, le déploiement rapide, l'auto-organisation et la tolérance aux pannes. Donc il est utilisé pour détection et collecte d'informations sur la position de l'ennemi, surveillance des zones hostiles (contaminées), détection d'agents chimiques, bactériologiques... .

Par exemple :

Le réseau WATS (Wide Area Tracking System) [11], développé par le laboratoire national de recherche « Lawrence Livermore ». L'objectif principal de ce réseau est d'effectuer la surveillance constante d'une zone d'intérêt, à l'aide des détecteurs des rayons gamma et des neutrons pour détecter et dépister les dispositifs nucléaires.



Figure 1. 2 : Application militaire.

4.5-le bâtiment :

Lors de tremblements de terre, pour aider les secouristes à retrouver les victimes (capteurs emprisonnés dans le béton à la construction qui détectent le niveau de bruit)

4.6-domaine de la médecine :

Cette technologie consiste en la surveillance la température, la fréquence cardiaque, l'oxygénation, et le pouls de patient [12].

Par exemple :

On peut placer des capteurs sur le corps d'un patient qui surveillent constamment sa température, la fréquence cardiaque, l'oxygénation, et le pouls de patient

Un émetteur-récepteur transmet les données à une station de base branchée à un ordinateur personnel ou poste soins infirmiers au même à un assistant numérique

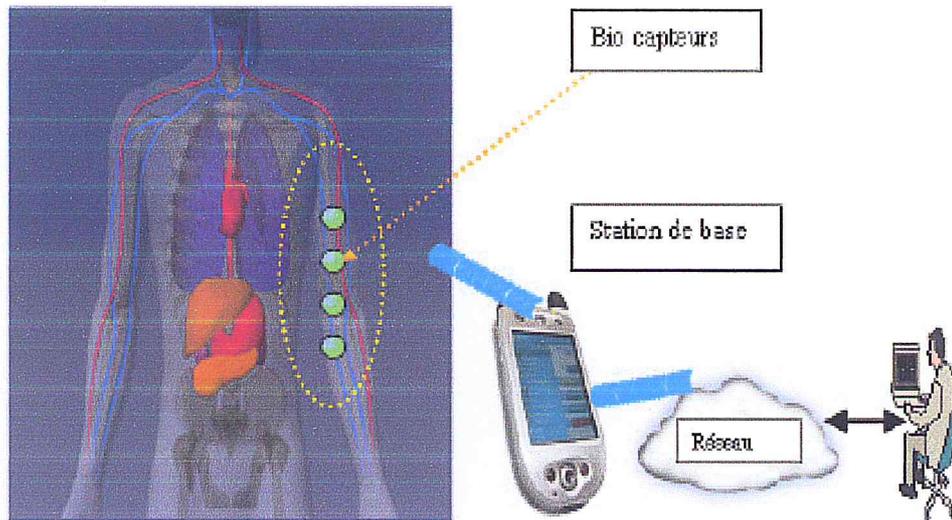


Figure 1.3 : Application médecine.

4.7-Applications domestiques

Avec le développement technologique, les capteurs peuvent être embarqués dans des appareils, tels que les aspirateurs, les fours à micro-ondes, les réfrigérateurs, les magnétoscopes, etc [13]. Ces capteurs embarqués peuvent interagir entre eux et avec un réseau externe via Internet pour permettre à un utilisateur de contrôler les appareils domestiques localement ou à distance. Le déploiement des capteurs de mouvement et de température dans les futures maisons dites intelligentes permet d'automatiser plusieurs opérations domestiques telles que : la lumière s'éteint et la musique s'arrête quand la chambre est vide, la climatisation et le chauffage s'ajustent selon les points multiples de mesure,

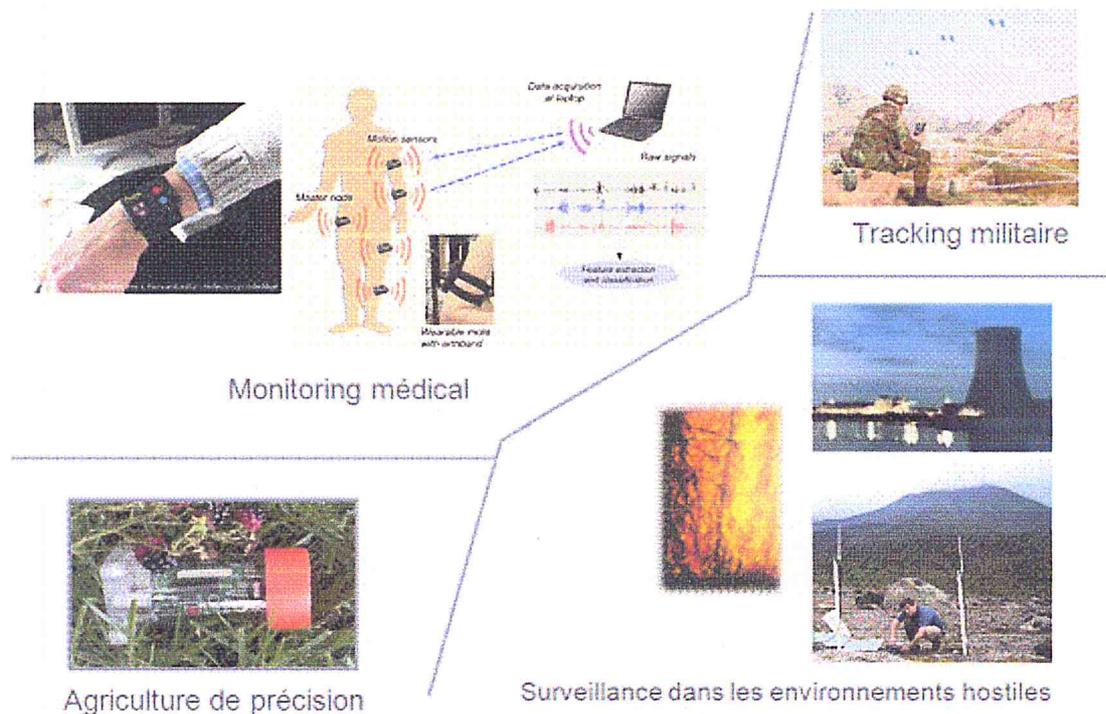


Figure 1. 4: application des RCSF.

5-Facteurs et contraintes des réseaux de capteurs sans fil :

Les principaux facteurs et contraintes influençant des réseaux de capteurs peuvent être résumés comme suit :

- Absence d'infrastructure : les réseaux Ad-hoc en général, et les réseaux de capteurs en particulier se distinguent des autres réseaux par la propriété d'absence d'infrastructure préexistante et de tout genre d'administration centralisée.
- Topologie dynamique : La topologie des réseaux de capteurs peut changer au cours du temps pour les raisons suivantes :
 - Les nœuds capteurs peuvent être déployés dans des environnements hostiles (champ de bataille par exemple), la défaillance d'un nœud capteur est, donc très probable.
 - Un nœud capteur peut devenir non opérationnel à cause de l'expiration de son énergie.
 - Dans certaines applications, les nœuds capteurs et les stations de base sont mobiles [14].

- Bande passante limitée : afin de minimiser l'énergie consommée lors de transfert de données entre les nœuds, les capteurs opèrent à bas débit. Typiquement, le débit utilisé est de quelques dizaines de Kb/s. Un débit de transmission réduit n'est pas handicapant pour un réseau de capteurs où les fréquences de transmission ne sont pas importantes.
- Les couts de production : puisque les réseaux de senseurs sont composée d'un très grand nombre des nœuds. Le prix de nœud est critique afin de pouvoir concurrence un réseau de surveillance traditionnel [15].
- Contrainte d'énergie, de stockage et de calcul : la caractéristique la plus critique dans les réseaux de capteurs est la modestie de ses ressources énergétiques car chaque capteur du réseau possède de faibles ressources en termes d'énergie, de calcul et de stockage. Afin de prolonger la durée de vie .
- Passage à l'échelle : le nombre de nœud déployés pour une application peut atteindre des milliers. Dans ce cas, le réseau doit fonctionner avec des densités de capteurs très grandes. Un nombre aussi important de nœuds engendre beaucoup de transmissions inter nodales et nécessite que la station de base soit équipée de mémoire suffisante pour stocker les informations reçues.

6- Architecture de communication :

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

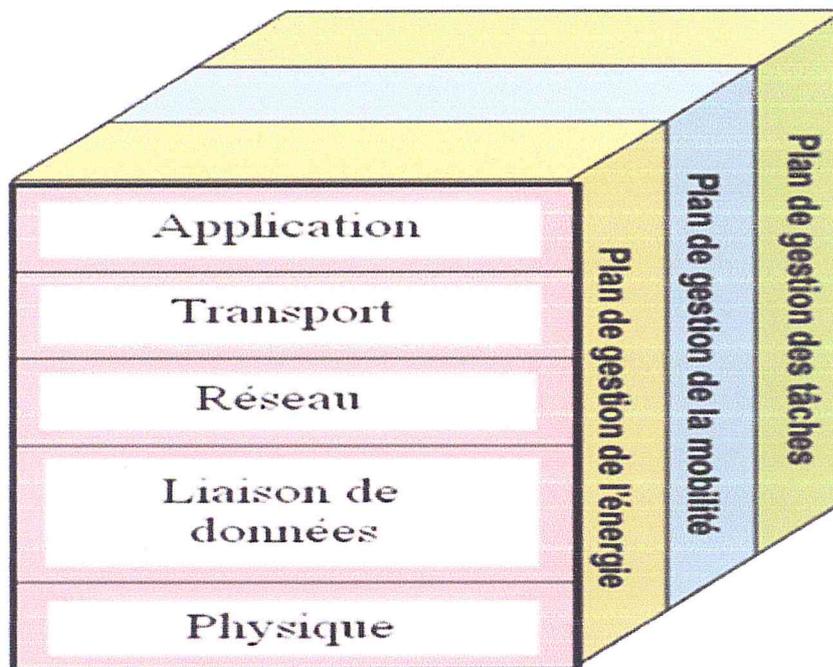


Figure 1. 5 : Modèle en couches du réseau de capteurs sans fil [16].

6.1-La couche physique : Elle est responsable de la sélection de fréquence, la génération de la fréquence porteuse, la détection du signal, la modulation/démodulation et le cryptage/décryptage des informations. La consommation d'énergie au niveau de la couche physique peut être affectée par l'environnement de l'application, le choix du type de la modulation ou la bande de fréquence utilisée. Il est avantageux en matière d'économie d'énergie que le concepteur de la couche physique choisisse une transmission à multi-sauts plutôt qu'une transmission directe qui nécessite une puissance de transmission très élevée.

6.2- Couche liaison de données : cette couche est responsable de multiplexage du flux de donnée, de la détection et le verrouillage des termes de donnée du contrôle d'accès au média (MAC : Média Access Control), de control erreur [17].

6.3-couche réseau : dans la couche réseau le but principal est de trouver une route et un transmission des données, captées, des nœuds capteurs vers le puits « sink » en optimisant l'utilisation de l'énergie des capteur.

6.4-Couche transport : Au niveau de la couche transport, quelques protocoles ont été proposés pour assurer une communication fiable de bout en bout dans les RCSF.

Event-to-Sink Reliable Transport protocol (ESRT) est un protocole qui assure un transport fiable des données des capteurs vers la station de base. Pump Slowly, Fetch quickly (PSFQ) est un protocole de transport basé sur une garantie locale d'arrivée des données, c'est-à-dire que chaque nœud intermédiaire doit conserver une copie du paquet transmis en cache jusqu'à s'assurer que ce dernier est bien arrivé au nœud suivant sur le chemin [18].

6.5-couche application : Elle assure l'interface avec les applications. Il s'agit donc de la couche la plus proche des utilisateurs.

- **Le plan de gestion d'énergie :** Il gère la manière dont le nœud utilise son énergie. Par exemple, si le nœud capteur est faible en énergie, il pourra informer ses nœuds voisins par multicast qu'il ne pourra pas participer dans le routage des paquets.
- **Le plan de gestion de la mobilité :** Il détecte les mouvements des nœuds et indique leurs placements.. De cette manière, chaque nœud peut connaître les nœuds qui lui sont voisins (il pourra alors balancer ses tâches vers un autre nœud au cas où il manque d'énergie). Il doit aussi maintenir à n'importe quel instant la route séparant le nœud mobile du nœud « Sink ».
- **Le plan de gestion des tâches :** Il assure un ordonnancement des tâches de capture dans une région bien déterminée tout en évitant la redondance des tâches de capture à un même instant, et ceci dans le but d'économiser de l'énergie sur le réseau [19].

7-Domains de recherche sur les réseaux de capteurs sans fil :

Dans la littérature, plusieurs travaux de recherche visent à proposer des solutions optimales et efficaces à un ou plusieurs problèmes des RCSF. Les principaux domaines de recherche abordés dans les RCSF sont les suivants : [14]

7.1- La consommation énergétique :

L'économie d'énergie est une des problématiques majeures dans les réseaux de capteurs. En effet, la recharge des sources d'énergie est souvent trop coûteuse et parfois impossible. Il faut donc que les capteurs économisent au maximum l'énergie afin de pouvoir fonctionner. En effet, un réseau de capteurs ne peut pas survivre si la perte de nœuds est très importante car

ceci engendre des pertes de communications dues à une très grande distance entre les nœuds restants. Les réseaux de capteurs fonctionnant selon un mode de routage par saut, chaque nœud du réseau joue un rôle important dans la transmission de données. Le mauvais fonctionnement d'un nœud implique un changement dans la topologie et impose une réorganisation du réseau

7.2-Clustering :

Le clustering consiste à découper le réseau en groupes de nœuds appelés Cluster. Pour chaque cluster un nœud maître appelé Cluster-Head est élu selon l'état courant du réseau afin d'accomplir des tâches spécifiques.

7.3- La Sécurité :

Les réseaux de capteur sans fil ont un intérêt particulier pour les applications militaires, environnementales, domotiques, médicales, et bien sûr les applications liées à la surveillance des infrastructures critiques. La conception de ces applications suppose que tous les nœuds engagés sont coopératifs et dignes de confiance. Cependant, ceci n'est pas le cas dans les déploiements du monde réel, où les nœuds sont exposés à différents types d'attaques qui peuvent carrément endommager le bon fonctionnement du réseau. Ces attaques exploitent essentiellement l'incertitude du canal de communication et le déploiement aléatoire des nœuds capteurs dans des zones difficiles à surveiller. Garantir la sécurité de ce type de réseau est une tâche difficile, surtout quand les nœuds sont constitués d'engins électroniques peu onéreux avec des capacités matérielles limitées. Le cas échéant, utiliser des protections physiques est, dans beaucoup de situations, quasiment impraticable. Capturer des nœuds est alors une possibilité intéressante pour les attaquants. Néanmoins, les WSN ne peuvent compter sur l'intervention humaine pour faire face aux tentatives d'un attaquant pour compromettre le réseau ou gêner ses propres opérations [17].

7.4 - Tolérance aux pannes :

Certains nœuds peuvent générer des erreurs ou ne plus fonctionner à cause d'un manque d'énergie, un problème physique ou une interférence. Ces anomalies ne doivent pas affecter le reste du réseau. Cette contrainte mesure la capacité de maintenir les fonctionnalités du réseau sans interruption dues à une panne intervenue sur un ou plusieurs capteurs [16].

7.5- Agrégation des données :

Dans les réseaux de capteurs, les données produites par les nœuds capteurs voisins sont très corrélées spatialement et temporellement. Ceci peut engendrer la réception par la station de base d'informations redondantes. Réduire la quantité d'informations redondantes transmises par les capteurs permet de réduire la consommation d'énergie dans le réseau et ainsi d'améliorer sa durée de vie. L'une des techniques utilisée pour réduire la transmission d'informations redondantes est l'agrégation des données. Avec cette technique, les nœuds intermédiaires agrègent l'information reçue de plusieurs sources. Cette technique est connue aussi sous le nom de fusion de données.

7.6 - Localisation :

La localisation est un processus très important dans un réseau de capteurs et peut être nécessaire dans plusieurs applications et contribue également au développement de protocoles de routage de l'information récoltée, pour la couverture de la zone d'intérêt, pour l'agrégation des données, etc. L'objectif de notre travail consiste de faire une étude sur les mécanismes de localisation des nœuds dans un réseau de capteurs [17].

8-Conclusion :

Les RCSF possèdent des caractéristiques particulières qui les différencient des autres types de réseaux sans fil. Dans ce premier chapitre, nous avons présenté les réseaux de capteurs sans fil, leurs architectures de communication, leurs contraintes ainsi que leurs domaines d'applications et leurs domaines de recherche. Parmi les domaines de recherche de réseau, nous intéressons dans notre étude à la localisation.

Chapitre 02 :

La localisation dans les réseaux de capteurs sans fil (RCSFs).

1-introduction :

Les réseaux de capteurs sans fil sont des réseaux composés d'un grand nombre des capteurs intelligents collaborant ensemble. L'une des problématiques principales des réseaux de capteurs sans fil est la localisation géographique des capteurs qui s'avère primordiale à la fois pour les protocoles de routage que pour la plupart des applications. Plusieurs approches ont été considérées pour la résolution du problème de localisation. La localisation consiste à déterminer la position physique pour chaque capteur .La première méthode la plus développée est l'utilisation du GPS (Global Positioning System) qui consiste à équiper chaque capteur d'un module GPS. Cette technique n'est pas applicable à l'ensemble du réseau de capteurs, car elle est bien trop coûteuse du point de vue financier comme du point de vue énergétique [18].

Dans ce chapitre on présente le système de localisation, Ce dernier est composé de trois sous-systèmes : un système de coordonnées, un système d'estimation des distances entre les nœuds et un système pour calculer la position relative des nœuds à la fin un algorithme de localisation qui définit la façon dont les informations de localisation sont manipulées dans le réseau afin que la grande partie des nœuds puisse se localiser.

2-Définition la localisation :

Le terme *localisation* est utilisé pour faire référence à un système permettant de déterminer l'emplacement d'un objet. Dans un premier temps, l'utilisation des systèmes de navigation par satellite (GPS), permettant une localisation relativement précise en environnements extérieurs dégagés (avec une précision de l'ordre de quelques mètres), mais elle est très coûteuse financièrement et sur le plan de la consommation d'énergie. La plupart des procédés employés pour déterminer une position sont basés sur des calculs géométriques comme la triangulation (en mesurant des angles par rapport à des points fixes ou des nœuds connaissant leur position) et la trilatération (en mesurant la distance entre les nœuds). Pour connaître la distance entre deux nœuds, plusieurs techniques peuvent être utilisées, comme la synchronisation, la puissance de signal reçu ainsi que les caractéristiques physiques de l'onde porteuse. D'autres approches, comme les caractéristiques du signal radio reçu et l'angle de l'arrivée peuvent être également appliquées pour le calcul de position.

3-Le processus de localisation dans les RCSFs :

De la problématique de la localisation découle trois problèmes sous-jacents. Les deux premiers sont directement liés au matériel utilisé (définition d'un système de coordonnées et estimation des distances), tandis que le troisième concerne les techniques logicielles utilisées.

3.1-Définition d'un système de coordonnées :

(un repère) : en connaissant les positions nœuds du réseau (appelés les ancres ou « beacons ») dans un certain système de coordonnées et les position relatives des autres nœuds par rapport à ces ancres, il est possible de retrouver les positions absolues des nœuds dans le même système .Toute la question demeure de bien « sélectionner »les points repères (les ancres) [19].Dans un système de coordonnées il existe deux types des coordonnées :

-Les coordonnées relatives sont définies par rapport à une référence commune, mais cette référence n'est pas la même pour tous les nœuds ; elle n'est donc pas globale. Par exemple, en France, les positions des villes sont définies par rapport à Paris et aux Etats-Unis elles sont définies par rapport à Washington.

-Pour passer des coordonnées relatives aux coordonnées absolues, un ensemble d'ancres est nécessaires (au moins trois en 2D). Ces amers doivent connaître leurs coordonnées absolues dans un repère global. Un ensemble de transformations géométriques (rotation, translation et éventuellement homothétie) sont nécessaires pour obtenir les coordonnées absolues de tous les nœuds [20].

3.1.1- les ancres :

Les ancres (souvent appelées beacons) sont au préalable nécessaires pour localiser les nœuds d'un réseau de capteurs dans le système de coordonnées globale. Les ancres sont simplement des nœuds capteurs ordinaires qui connaissent leurs coordonnées *à priori*. Cette connaissance pourrait être difficilement codée, ou bien facilement acquise par un certain matériel supplémentaire comme un récepteur GPS. Au minimum, trois ancres non – colinéaires sont nécessaires pour définir un système de coordonnées en deux dimensions. Les coordonnées peuvent être globales, c'est à dire qu'elles sont alignées avec un système extérieur comme le système GPS par exemple, ou bien relatives. .

3.1.2 Les types des ancres :

On distingue deux types des ancres :

▪ Ancre statique :

Les ancres sont simplement des nœuds capteurs ordinaires qui connaissent leurs coordonnées *à priori*. Les ancres statiques sont les capteurs figés dans leur position initiale [21]

▪ Ancre mobile :

Les ancres mobiles sont des nœuds qui connaissent leurs coordonnées et aident les autres nœuds non localisés par un déplacement en suivant une trajectoire spécifique. La figure suivante illustre comment un ancre mobile se déplace à travers une ou plusieurs trajectoires [22].

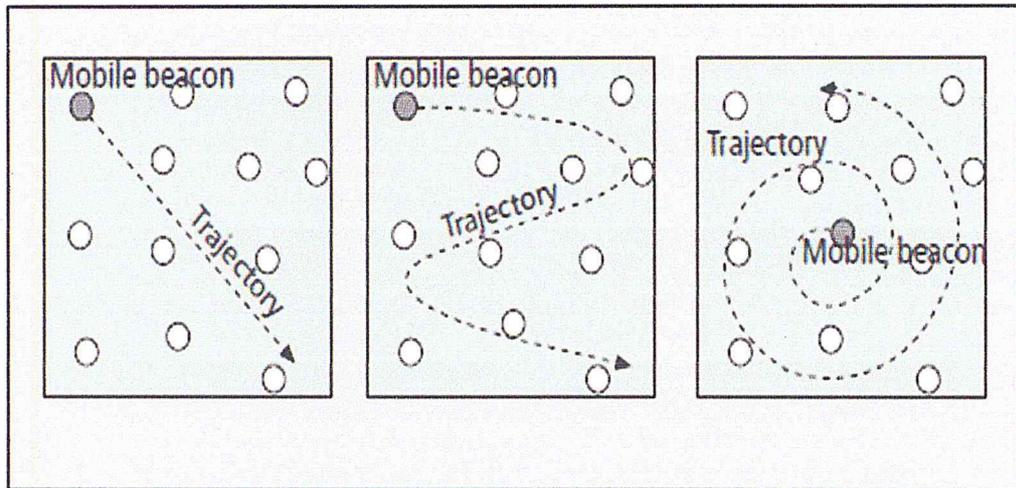


Figure 2.1 : Exemple d'un nœud ancre [22].

3.2-Estimation des distances :

Plusieurs techniques sont développées pour les estimations des distances entre les nœuds voisins. Parmi lesquelles nous trouvons celles qui ont fondées sur l'utilisation d'autres matériels comme la technique de la différence entre les temps d'arrivée de deux signaux « *TDoA* » et celle qui estime l'angle d'arrivée du signal « *Angle of Arrival (AoA)* » et celles qui sont basées sur les dispositifs radio, Dans cette techniques les nœuds communiquent entre eux et collectent différents indicateurs de qualité des communications radios. Le matériel radio peut rapporter diverses informations sur le signal radio entre deux nœuds. En effet le simple fait qu'ils communiquent entre eux nous indique qu'ils sont à portée radio l'un de l'autre. De plus le matériel radio de nos nœuds peut nous rapporter diverses caractéristiques à propos du signal radio entre les deux nœuds, à partir desquelles les distances séparant les nœuds peuvent être estimées [23].

3.2.1 Informations disponibles pour la localisation :

Cette section décrit les différentes informations pouvant être exploitées par les systèmes de localisation.

- **Puissance du signal radio :**

La puissance d'émission et de réception d'un signal peut être également exploitée pour obtenir la distance entre deux capteurs .La technologie RSSI (Received Signal Strength Indicator) considère la perte de puissance d'un signal entre son émission et sa réception. Cette perte varie en fonction de la distance entre les deux capteurs : plus les capteurs sont éloignés (resp.proches), plus la perte est importante (resp. faible). Cette perte sera alors traduite en une distance.

- **Equation de Friis :**

$$\frac{P_r}{P_t} = G_t \times G_r \times \frac{1}{L} \times \left(\frac{\lambda}{4\pi R}\right)^2 \quad (1)$$

Les bilans de liaison utilisent généralement le modèle de l'équation (1) [24] pour prédire le niveau de puissance reçue P_r à une certaine distance R (de l'émetteur au récepteur) connaissant la puissance émise P_t . Le milieu traversé est caractérisé par les pertes L . Ces pertes sont liées à la fréquence du signal dont la longueur d'onde λ intervient dans la formule. Finalement, les gains des antennes d'émission et de réception utilisées, G_t et G_r , entrent en ligne de compte. Ce modèle fait donc l'hypothèse d'une répartition uniforme de l'énergie sur des sphères concentriques.

- **modele de propagation avec réflexion sur le sol (Two Ray Ground reflection model) :**

[25] considère que le signal suit deux chemins principaux (figure 2.2) pour arriver au récepteur. Le premier est direct et le second réfléchi par le sol.

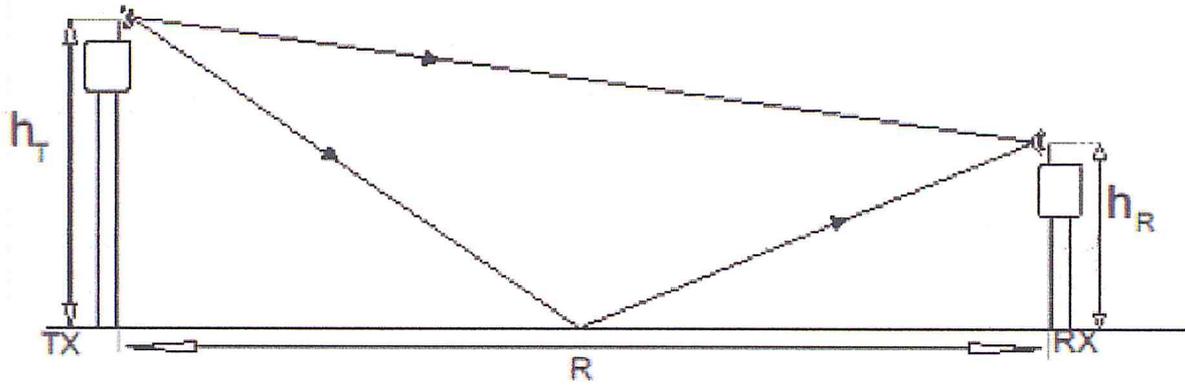


Figure 2.2 : Propagation avec une réflexion [25]

Le passage du modèle *Free Space* au modèle *Two Ray ground reflection* dépend de l'espacement R entre les antennes : entre 0 et la distance critique D_c (équation 3), le modèle de propagation en espace libre s'applique. Au-delà, le modèle avec réflexion au sol s'utilise. Ce dernier tient également compte de la hauteur des antennes d'émission h_T et de réception h_R (équation 2).

$$\frac{P_R}{P_T} = \frac{G_R \times G_T}{L} \left(\frac{h_T \times h_R}{R^2} \right)^2 \quad (2)$$

$$D_c = \frac{4\pi h_R h_T}{\lambda} \quad (3)$$

- **Modèle de Shadowing :**

Ce modèle comporte deux parties :

- Le modèle d'atténuation : une distance de référence d_0 est définie et un niveau de puissance reçue lui est associé. La puissance reçue à une distance d quelconque est liée à ces informations par la relation (équation 4).

$$\frac{P_r(d_0)}{P_r(d)} = \left(\frac{d}{d_0} \right)^\beta \quad (4)$$

- La variation de la puissance à une distance donnée : en un point donné, la puissance reçue varie au cours du temps. Une loi *log-normale* X_{dB} a été retenue pour représenter cette variation. Le modèle complet s'écrit alors suivant l'équation(5) où β est un coefficient d'atténuation empirique.

$$\left[\frac{P_r(d_0)}{P_r(d)} \right]_{dB} = -10\beta \log\left(\frac{d}{d_0}\right) + X_{dB} \quad (5)$$

Si l'écriture du *Shadowing model* est plus complexe que celle des modèles précités, ses résultats sont plus proches de la réalité car la zone de portée n'est pas assimilée à un cercle mais à une surface dont les bornes varient au cours du temps, même sans déplacement de la source [25].

3.2.2 Techniques de mesure d'estimation de la distance pour range-based :

Selon la caractéristique du signal à récupérer, le processus de collecte change. Nous présentons dans ce qui suit les techniques de la mesure de distance pour *range-based*.

➤ Indicateur de puissance de signal reçu (RSSI) :

La technique de localisation basée sur le RSSI utilise la puissance du signal reçu pour estimer la distance entre un nœud émetteur et un nœud récepteur. La distance est estimée en utilisant un modèle de propagation des ondes radio. La plupart des appareils sans fil ont la capacité de mesurer l'intensité du signal reçu. L'intensité du signal sans fil reçu par un capteur d'un autre capteur est une fonction monotone décroissante de leur distance. Cette relation entre la puissance du signal reçu et la distance est modélisée par le modèle log normal suivant:

$$P_r(d)[dBm] = P_0(d_0)[dBm] - 10n_p \log_{10}\left(\frac{d}{d_0}\right) + X_\alpha \quad (8)$$

Avec $P_0(d_0)[dBm]$: est une puissance de référence en milliwatts dB à une distance de référence d_0 de l'émetteur, n_p est l'exposant de perte de trajet qui mesure la vitesse à laquelle l'intensité du signal reçu diminue avec la distance, X_α est une variable aléatoire de moyenne de distribution gaussienne nulle avec l'écart-type σ et il représente l'effet aléatoire causé par l'ombrage. n_p et σ sont dépendants de l'environnement. L'exposant de perte de trajet n_p est généralement supposé être une constante. Le grand avantage de la méthode est son faible coût car tous les nœuds sont déjà équipés de dispositif de communication sans fil ; de plus elle ne rajoute aucun surcoût de communication [26].

➤ **Angle d'arrivée AoA :**

A la place des distances, il est possible de localiser un nœud en utilisant des angles. La méthode AoA (Angle of Arrival, DoA Direction of Arrival) permet de mesurer l'angle d'arrivée d'un signal. Pour cela le nœud qui mesure l'angle doit être équipé soit par un ensemble de petites d'antennes ou plusieurs récepteurs à ultrason [27]. Lorsque un nœud reçoit un signal, il utilise la phase ou le temps d'arrivée du signal au niveau de chaque antenne (ou récepteur ultrason) pour déduire l'angle d'arrivée du signal. La nécessité d'avoir du matériel supplémentaire ainsi que des distances minimales entre les différents récepteurs (à ultrason) représentent les inconvénients de la méthode en termes de coût et d'encombrement (taille).

➤ **Temps d'arrivée TOA :**

La technologie ToA (Time of Arrival) suppose que les nœuds du réseau sont synchrones. La distance qui sépare deux capteurs se déduit de la vitesse de propagation du signal et de la différence entre les dates d'émission et de réception du message. Cette technologie est celle utilisée par le système GPS (Global Positioning System). Cette technique nécessite la synchronisation parfaite entre l'émetteur et le récepteur [28]. Si le récepteur connaît l'heure d'émission du signal, il peut alors calculer la distance le séparant de l'émetteur en déterminant le temps pris par le signal pour arriver au niveau du récepteur.

➤ **Différence des temps d'arrivée TDOA :**

Les mesures TDOA (Time Difference Of Arrival) se basent sur la différence entre le temps d'arrivée d'un signal d'émetteur à deux récepteurs, on note les coordonnées des deux récepteurs par X_i et X_j , et les coordonnées de l'émetteur par X_t . La mesure TDOA Δt_{ij} est liée aux positions des deux récepteurs par

$$\Delta t_{ij} = t_i - t_j = \frac{1}{c} (\|X_t - X_i\| - \|X_t - X_j\|) \quad (9)$$

Avec t_i et t_j sont respectivement les temps d'arrivée du signal de l'émetteur au récepteurs i et j , et c est la vitesse de propagation du signal de l'émetteur. En supposant que l'emplacement des récepteurs est connu et que les deux récepteurs sont parfaitement synchronisés, l'équation (9) définit une branche d'hyperbole à laquelle l'émetteur doit se trouver. Les foyers de l'hyperbole sont les emplacements des récepteurs i et j . Dans un système des récepteurs N , il y a $N-1$ mesures TDOA linéairement indépendants, donc $N-1$ équations linéairement indépendants comme (9). En 2, les mesures TDOA de trois récepteurs au minimum sont requises pour déterminer l'emplacement de l'émetteur.

La précision des mesures TDOA est affectée par l'erreur de synchronisation entre les récepteurs et les chemins multiples. La précision et les capacités de résolution temporelle de mesures permettront d'améliorer TDOA lorsque la séparation entre les récepteurs augmente parce que cette différence augmente entre les temps d'arrivée.

3.2.3-Techniques de mesure d'estimation de la distance pour range-free :

Ces méthodes ne calculent jamais de distances entre voisins. Elles utilisent d'autres informations telles que la connectivité pour identifier la position des nœuds. Dans un objectif de simplicité et de réduction du coût, ces méthodes supposent que le déploiement des nœuds respecte certaines contraintes et proposent des calculs plus ou moins complexes pour évaluer la position. Elles semblent donner de bons résultats dans les réseaux denses et réguliers. En conclusion Algorithmes range-free estiment positions relatives des nœuds plutôt que de calculer les positions absolues des nœuds [29].

❖ Le nombre de sauts – RHC (*Radio Hop Count*) :

La méthode RHC est basée sur le constat suivant : si deux nœuds peuvent se communiquer, alors la distance entre eux est inférieure à la portée maximale du dispositif de communication avec une grande probabilité [30]. Pour la localisation, seulement l'information de connectivité est utilisée, l'indicateur de puissance du signal reçu n'est pas pris en compte.

C'est le schéma le plus basique, il utilise un échange de vecteur de distance afin que tous les nœuds du réseau parviennent à calculer la distance les séparant du ancre. Chaque ancre maintient une table $\{X_i, Y_i, h_i\}$ où (X_i, Y_i) sont les coordonnées des autres ancres du réseau et h_i est le nombre de sauts séparant ce dernier du nœud en question.

Chaque ancre calcule la distance le séparant des autres ancres dans le réseau, en utilisant les informations de localisation obtenues à partir d'un système de positionnement, il en déduit une approximation de la distance par saut. C'est la distance par saut qui va constituer

l'information de correction pour tout le réseau.

Chaque nœud ancre calcule :

$$hopSize = \frac{\sum \sqrt{(X_i - X_j)^2 - (Y_i - Y_j)^2}}{H_{ij}} \quad i \neq j$$

Avec (X_i, Y_i) représentent les coordonnées d'un nœud, (X_j, Y_j) les coordonnées d'un ancre et H_{ij} le nombre de saut.

3.2.4-analyse et discussion :

Le choix de la méthode d'estimation de distance/angle entre les nœuds est un facteur important qui influe sur les performances finales du système de localisation. Généralement, pour calculer une position en besoin de trois distances (angles) estimées et chacune d'elle avec une erreur associée. a de nombreuses contraintes telles que la taille des nœuds, l'énergie et Coût. Il est en effet nécessaire de considérer ces contraintes avant de concevoir tout mécanisme de localisation. Transmission de données nœuds de communication prend beaucoup de puissance et de consommer plus d'énergie. Alors les figures suivent donnée la différence entre ces méthodes par rapport au coût, l'énergie et la précision [18].

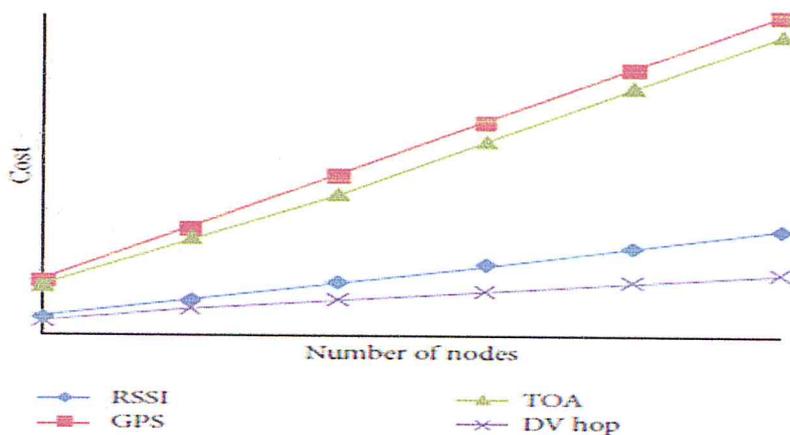


Figure 2.3: comparaison de coûts des différentes techniques de localisation.

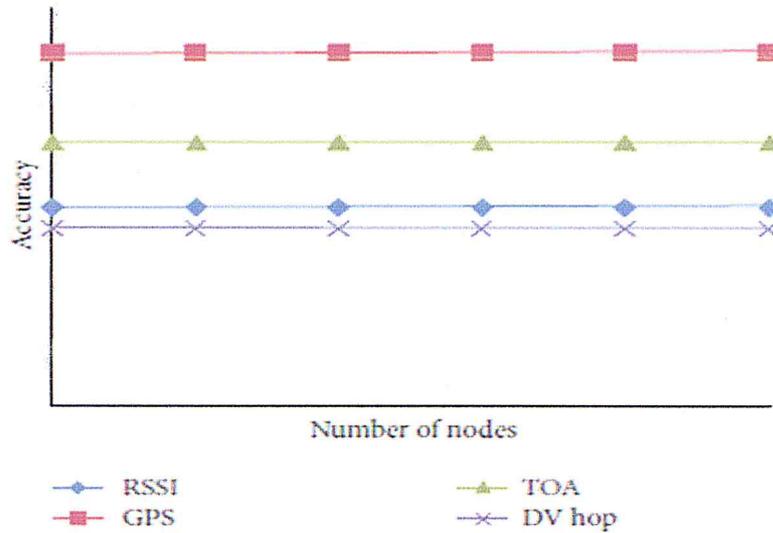


Figure 2.4: comparaison de précision des différentes techniques de localisation.

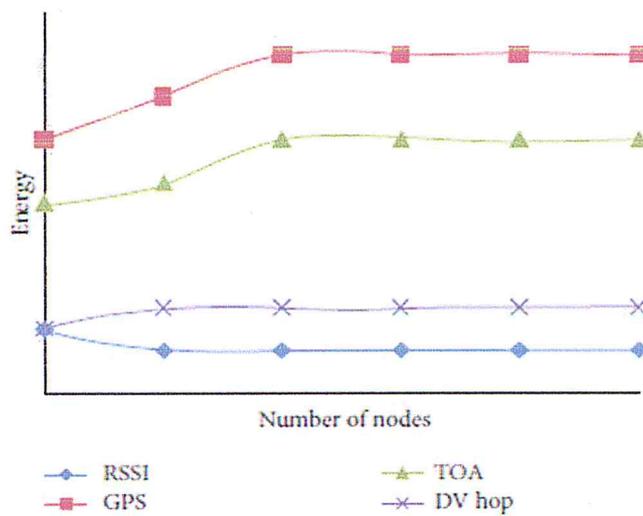


Figure 2.5: comparaison de consommation d'énergie des différentes techniques de localisation.

Les figures précédentes montrent que plus la précision est élevée, plus le coût est élevé, nous remarquons que le GPS est la méthode utilisée pour la localisation précise et absolue mais la conséquence soit la consommation d'énergie ou le coût d'équipements, mais on remarque qu'il y a des méthodes pour la localisation comme DV HOP ne pas pratiquement précise mais consomme d'énergie plus que RSSI [18].

3.3 Technique Dérivation des positions :

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

3.3.1 Trilatération :

Trilatération c'est un algorithme range-based c'est ta-dire utiliser estimation de distance pour calculé position de 2D, Cette algorithme permet de positionner un nœud à l'aide de trois nœuds de référence. Il existe deux types de mesure de distances entre les nœuds de référence et le nœud que l'on cherche à positionner, une méthode à base de temps d'arrivée et une à base de mesure de puissance du signal . Cet algorithme consiste donc à mesurer les distances qui séparent le nœud $M(x,y)$ à positionner aux nœuds de référence.

Soient les trois nœuds B_1, B_2 et B_3 , M le nœud mobile que l'on désire localiser. L'exécution les méthodes de mesure de distance a produit le triplet (d_1, d_2, d_3) et les échanges de données ont permis au mobile de connaître les positions des nœuds [32].

$$\begin{aligned} (x-x_1)^2+(y-y_1)^2 &= d_1^2 \\ (x-x_2)^2+(y-y_2)^2 &= d_2^2 \\ (x-x_3)^2+(y-y_3)^2 &= d_3^2 \end{aligned}$$

La position recherchée se trouve donc à l'intersection des cercles $C_1(B_1,d_1)$ et $C_2(B_2,d_2)$. Dans le cas général, C_1 et C_2 se rencontrent en deux points distincts M et M' . Grâce aux données de l'ancre B_3 (figure 2.6).

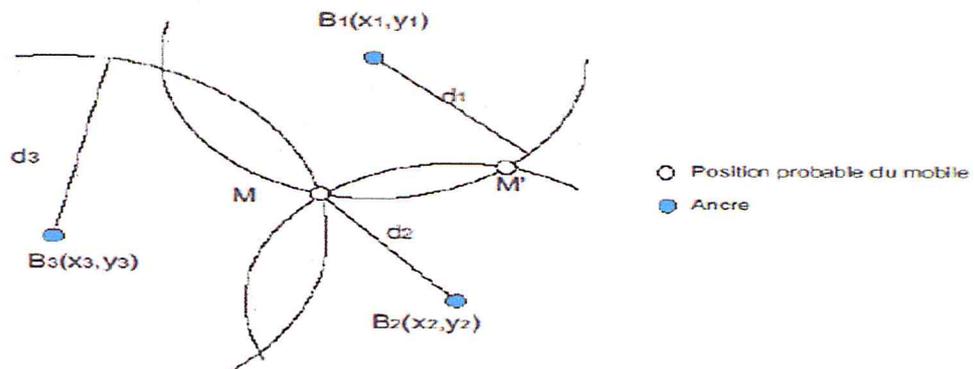


Figure 2.6: Trilatération

Pour calculer l'intersection de ces trois cercles, on peut résoudre le système de cette façon : soustraire la troisième équation à la première et soustraire la troisième équation à la deuxième, ce qui donne :

$$\begin{aligned}(x-x_1)^2-(x-x_3)^2+(y-y_1)^2-(y-y_3)^2 &= d_1^2-d_3^2 \\ (x-x_2)^2-(x-x_3)^2+(y-y_2)^2-(y-y_3)^2 &= d_2^2-d_3^2\end{aligned}$$

On calcule alors les $()^2$ et on met x et y en évidence, on trouve alors :

$$\begin{aligned}2(x_3-x_1)x+2(y_3-y_1)y &= d_1^2-d_3^2+x_3^2-x_1^2+y_3^2-y_1^2 \\ 2(x_3-x_2)x+2(y_3-y_2)y &= d_2^2-d_3^2+x_3^2-x_2^2+y_3^2-y_2^2\end{aligned}$$

Qui est donc un système linéaire. On a donc un système de deux équations à deux inconnues

$Ax = b$ dont la solution est $x = A^{-1}b$

$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 2(x_3-x_1) & 2(y_3-y_1) \\ 2(x_3-x_2) & 2(y_3-y_2) \end{pmatrix}^{-1} \begin{pmatrix} d_1^2-d_3^2+x_3^2-x_1^2+y_3^2-y_1^2 \\ d_2^2-d_3^2+x_3^2-x_2^2+y_3^2-y_2^2 \end{pmatrix}$$

Alors il faut calculer l'inverse de matrice A^{-1} est :

$$\det(A) = |A| = \begin{vmatrix} \alpha_{11} & \alpha_{12} \\ \alpha_{21} & \alpha_{22} \end{vmatrix} = \alpha_{11}\alpha_{22} - \alpha_{21}\alpha_{12}$$

Remarque :

si $\det(|A| \neq 0)$, mathématiquement il est impossible de calculer la position et A^{-1} , pour éviter cette situation il faut rechercher une solution qui permet de pallier à ce problème.

3.3.2 La multilatération :

La multilatération nous permet de retrouver la position d'un nœud à partir des positions d'un certain nombre d'ancres et des distances entre ces ancres et le nœud à localiser. A le même principe que la trilatération, en utilisant plus que trois points de référence (ancres). La position

d'un nœud est calculée en résolvant l'intersection de plusieurs hyperboles basées sur la différence des temps d'arrivée TDoA. Soit une cible i , connaissant les positions (x_a, y_a) de m ancrs ($1 \leq a \leq m$) ainsi que les distances d_{ia} , où d_{ia} représente la distance euclidienne entre i et l'ancre a . Ayant ces informations et pour calculer la position (x_i, y_i) de la cible i nous formons le système suivant :

$$\begin{aligned} (x_1 + x_i)^2 + (y_1 - y_i)^2 &= d_{i1}^2 \\ (x_2 + x_i)^2 + (y_2 - y_i)^2 &= d_{i2}^2 \\ (x_3 + x_i)^2 + (y_3 - y_i)^2 &= d_{i3}^2 \\ &\vdots \\ (x_m + x_i)^2 + (y_m - y_i)^2 &= d_{im}^2 \end{aligned}$$

Ce système peut être linéarisé en soustrayant la dernière équation des $m-1$ équations précédentes. En réordonnant les termes, nous obtenons un système d'équations linéaires.

Ayant des erreurs dans les estimations de distances, il paraît qu'une solution exacte pour un tel système d'équations est presque impossible. La solution la plus proche de la solution exacte c'est au sens des moindres carrés. Cette solution est plus détaillée dans [33].

3.3.3 Le principe de Triangulation :

L'angle d'incidence du signal émis par le mobile peut être utilisé pour retrouver sa position . Considérons deux balises $B_1(x_1, y_1)$ et $B_2(x_2, y_2)$ à portée du mobile $M(x, y)$. Chacune dispose des moyens matériels de mesurer l'angle sous lequel elle reçoit le signal de M . Soient α et β ces deux angles.

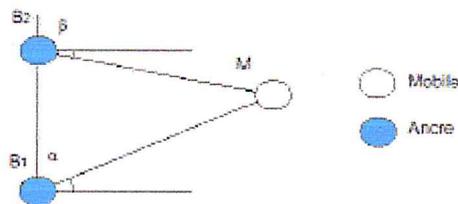


Figure 2.7 : Angle d'incidence et position

Le point M se retrouve à l'intersection des droites passant par les couples (B₁,M) et (B₂,M) (figure 2.7). En définissant un nouveau repère dont l'origine est B₁ et où (B₁,B₂) se confond avec l'axe des ordonnées, le rapprochement des équations des droites conduit à l'expression suivant de la position de M[34].

$$M\left(\frac{y_2}{\tan \alpha + \tan \beta}; \frac{y_2 \tan \alpha}{\tan \alpha + \tan \beta}\right)$$

4-Classification des approches de localisation :

L'algorithme de localisation est la partie principale du système de localisation. Donc Il existe de nombreuses approches pour résoudre le problème de la localisation sont proposées pour permettre aux nœuds d'estimer leur position. Nous pouvons distinguer deux types de stratégie de localisation : les stratégies directes (localisation absolue) et les stratégies indirectes (localisation relative) [28].

4.1- Les approches directes :

Connues également sous le nom de localisation absolue. L'approche directe elle-même peut être classifiée en deux types : configuration manuelle et localisation basée sur le GPS. La méthode de configuration manuelle est très encombrante et coûteuse. Elle n'est ni pratique ni adaptée pour les réseaux de capteurs à grande échelle et en particulier, ne s'adapte pas bien pour les réseaux de capteurs mobiles. D'un autre côté, la méthode de localisation basée sur le GPS, permet de résoudre en théorie le problème de localisation de chaque nœud du réseau, et s'adapte bien pour les nœuds mobiles. Toutefois, équiper chacun des capteurs d'un récepteur GPS constitue souvent une solution irréalisable en pratique, à cause du coût prohibitif d'un tel équipement pour un réseau constitué de millier de capteurs, de la réserve énergétique limitée des capteurs et du mauvais fonctionnement de cette technologie en intérieur [28].

Le système de positionnement GPS : est le système de localisation basé sur les réseaux satellitaires le plus utilisé. Le système GPS est initialement conçu par le département de la défense des États-Unis. Il repose sur un réseau de 24 satellites à 20000Km d'altitude de la terre et permet de calculer la position et l'altitude d'un mobile à chaque seconde partout dans le monde avec une précision de l'ordre de 10 mètres. Développé pour améliorer la précision du positionnement jusqu'à 1 mètre.

4.2- Les approches indirectes :

L'approche indirecte est également connue sous le nom de localisation relative, dans laquelle les positions des nœuds sont dérivées par rapport aux positions d'autres nœuds dans leur proximité. Les approches indirectes de localisation ont été introduites pour surpasser les inconvénients des techniques de localisation basées sur le GPS. Dans ces techniques, quelques nœuds capteurs nommés Beacon ou ancres sont équipés de récepteurs GPS et servent de repères pour les autres nœuds ordinaires qui vont calculer leur position selon des méthodes appropriées. Ces approches s'avèrent moins coûteuses que les approches directes et dans les approches indirectes, le processus de localisation est classé en deux catégories: les méthodes *range-based* et les méthodes *range-free*.

7- Algorithmes de localisation :

Nous distinguons deux façons d'implémenter un algorithme de localisation selon leur organisation de calcul : les algorithmes centralisés et les algorithmes distribués.

7.1- Algorithmes centralisés :

Les algorithmes centralisés sont conçus pour fonctionner sur une machine centrale très puissante au niveau ressources. Les nœuds capteurs recueillent des informations (signal, voisins, distances, etc.) de leur environnement et les transmettent à une station de base qui à son tour les analyse, calcule les positions et les transmet aux nœuds. Les algorithmes centralisés contournent le problème des ressources limitées des nœuds en acceptant des coûts de communications très élevés pour envoyer les informations à la machine centrale. Ces algorithmes deviennent de plus en plus coûteux quand la taille du réseau augmente, car ça épuise les nœuds qui sont trop proches de la station de base qui subissent un très grand nombre de communications. En outre, les algorithmes centralisés exigent qu'une station de base puissante soit déployée parmi les nœuds, ce qui n'est pas toujours possible. Dans le cas où c'est possible, le problème de la mise à l'échelle peut être résolu en déployant plusieurs stations de bases. Cependant, La centralisation permet à un algorithme d'être plus complexe, car les calculs se font sur la machine centrale et non pas par les nœuds eux-mêmes [18].

7.2-Algorithmes distribués :

Dans le cas d'un algorithme distribué tous les nœuds communiquent avec leurs voisins pour estimer les distances et échanger les informations de voisinage, afin de dériver leur position. Par conséquent, à la fin du processus de localisation, chaque nœud doit connaître sa position

ainsi que celles de ses voisins sans l'aide d'aucune unité centrale. Les algorithmes distribués, extrapolent généralement les positions des nœuds à partir de celles des ancres. Ainsi, ils localisent les nœuds directement dans le système de coordonnées global de ces ancres.

Comme le calcul des positions se fait par les nœuds eux-mêmes, les algorithmes distribués ne sont pas complexes. Pour les réseaux à grande échelle, on considère qu'une méthode distribuée est nécessaire car les méthodes centralisées demanderaient trop de communication pour l'acheminement des informations vers l'unité centrale et consommeraient donc trop d'énergie [34].

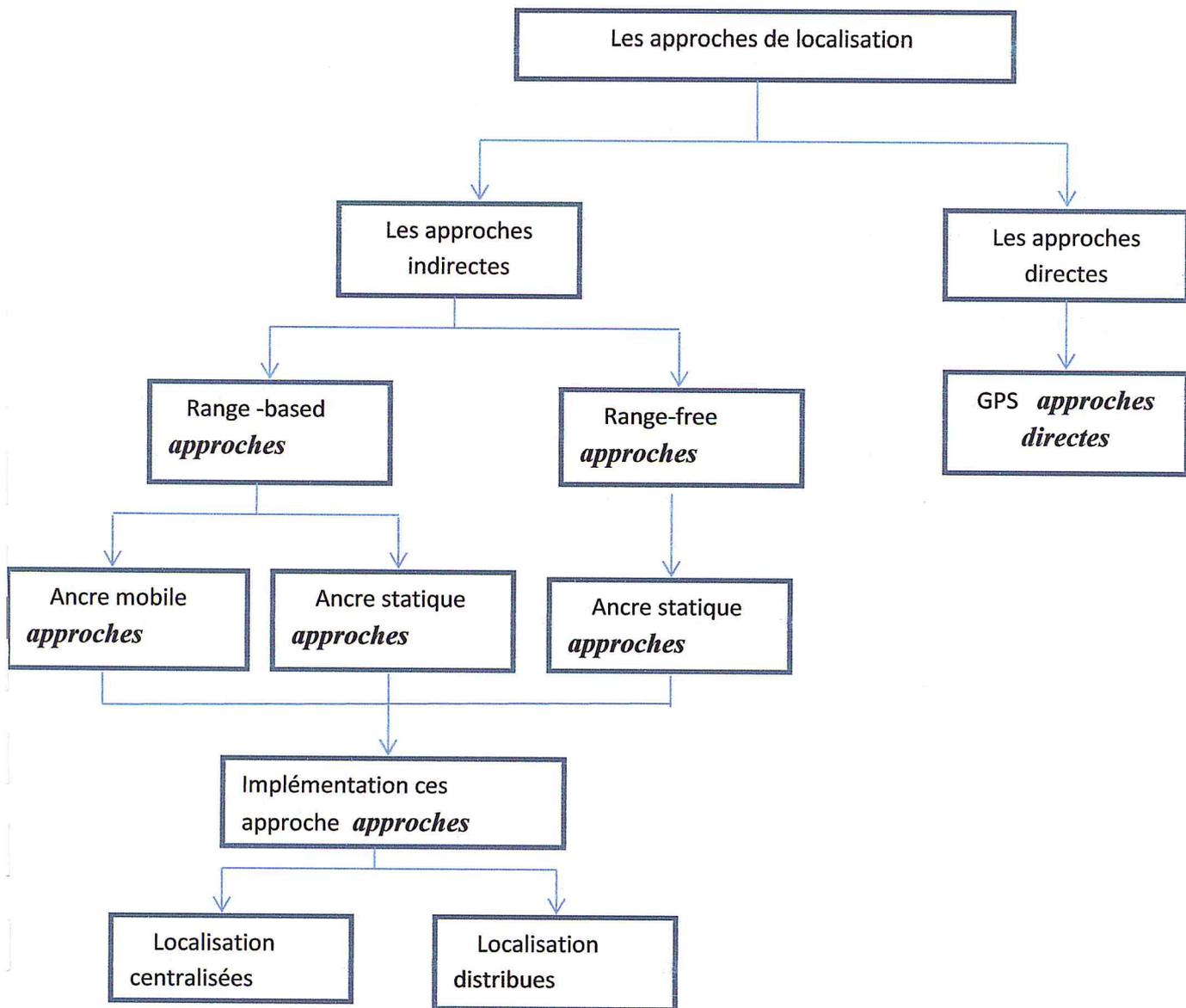


Figure 2.8 : les approches de localisation.

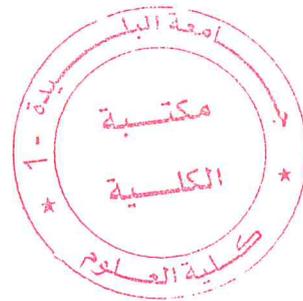
1. Critères des algorithmes de localisation

Un algorithme de localisation est évalué selon une liste de critères dont nous citons :

- Précision de la localisation : Erreur entre la position réelle et la position estimée.
- Cout énergétique de la localisation : Dans les WSN, une gestion de l'énergie très économique est nécessaire et comme le facteur dominant de la consommation d'énergie est la communication radio, il faut trouver un algorithme qui communique le moins possible via la radio.
- Robustesse au bruit : Il faut analyser comment un algorithme se comporte face au bruit rencontre dans les mesures de distances avec les voisins.
- Passage à l'échelle : L'algorithme est-il optimiser pour un réseau à grande échelle (algorithme implémentable de façon distribuée ou non).
- Tolérance à la basse connectivite : Dans le cas d'un réseau ou chaque nœud ne sait communiquer qu'avec un petit nombre de ses voisins, les performances de l'algorithme.
- Réactivité du système : rapidité du processus de localisation des nœuds. Ce paramètre est important si on veut gérer la mobilité des nœuds dans le réseau [35].

9- Conclusion :

Dans ce chapitre nous avons abordé la problématique de la localisation plus particulièrement dans les réseaux de capteurs sans fil ont détaillé le processus de localisation et les étapes pour concevoir un système de localisation adapté. Nous avons également présenté un certain nombre de techniques de mesure de position comme TDOA, AoA et TOA. Suite à cela, nous avons arrêté notre choix sur les techniques de RSSI (expliquée dans la section 3.2.2). pour faire l'objet de nos simulations, dans le but de déterminer la position des capteurs et utilisé dans la phase de estimation de position la triangulation et Trilatération Enfin, nous avons présenté quelques algorithmes de localisation et ces contraintes.



Chapitre 03 :

Conception & implémentation

1-introduction :

La localisation des nœuds est l'un des services les plus importants. Elle consiste à associer à chaque nœud les coordonnées de sa position. Plusieurs méthodes sont utilisées pour réaliser cette fonctionnalité (comme par exemple les dispositifs GPS mais cette technique n'est pas accessible). Un ensemble de systèmes (technologies) ont été développés pour assurer ce service, alors la localisation dans les réseaux de capteurs dépend de plusieurs dispositifs matériels. Cette dépendance se relève à l'utilisation des ancres (beacons), et à l'estimation des distances entre les nœuds. Dans cette section, nous avons présenté notre choix de chaque phase dans le system de localisation (estimation de distance, calcule de position) avec la justification de notre choix, ensuite nous allons présenter les outils de développement utiliser dans notre travail et la phase de l'implémentation de l'algorithme.

- ✓ Trilatération
- Ou
- ✓ Triangulation

2.2.1 Trilatération :

➤ **Définition :**

Cette méthode a le même principe que GPS, qui peut localiser n'importe quel objet par l'entrer en contact avec ces satellites qui lui communiquent leurs coordonnées pour cette raison nous avons opté pour la Trilatération. Au moins trois satellites sont nécessaires pour une localisation à deux dimensions, alors qu'au moins quatre le sont pour la troisième dimension (voir figure 3.1).

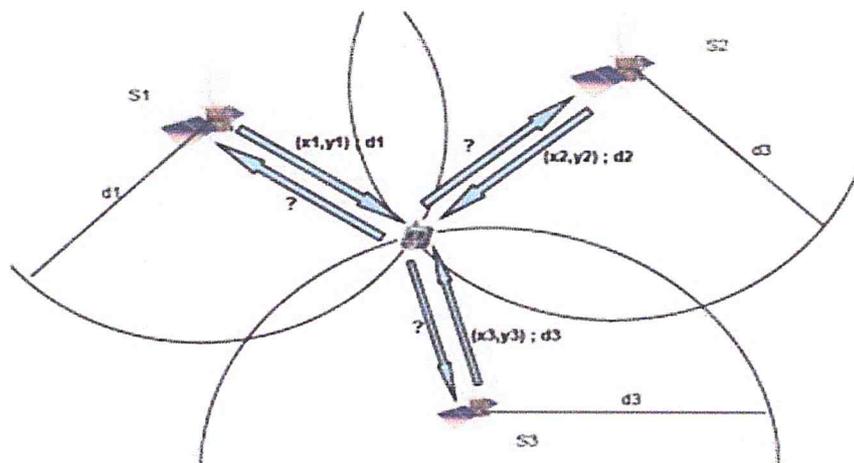


Figure 3. 1 : Principe des méthodes de localisation par satellite (en 2D).

➤ **Algorithme de localisation basée sur LTTA :**

Localisation Trilatération algorithme nommées par LTTA présente une alternative distribuée au traitement du problème de localisation dans un réseau de capteurs. Définissons, dans un premier temps, les notations qui seront utilisées dans la suite de notre travail pour les approches distribuées : soit un réseau de $N = m + n$ capteurs, évoluant dans un espace de dimension $p = 2$ (avec $n \gg m > p$). Soit $x_i \in R^p$ les coordonnées du capteur i , l'ensemble $\{x_i\}_m$ se réfère aux ancres dont les positions sont connues, et $\{x_i\}_{m+1}$ correspond aux positions inconnues des capteurs restants. Considérons que la portée maximale de détection de chacun des capteurs est égale à r , le capteur i considère alors le capteur j comme voisin si et seulement si la distance $Dist(i,j)$, mesurée par la technique RSSI, est inférieure à r . Si un nœud estime sa position, il a diffusé sa coordonnée pour aider ses voisins pour calculer leurs positions.

2-algorithme de localisation :

Comme présenté dans le chapitre 2, le processus de localisation est composé de trois phases :

- Définition d'un système de coordonnées.
- Estimation de distance.
- Dérivation des positions.

Notre algorithme est un algorithme distribué donc tous les nœuds communiquent avec leurs voisins ancrés pour estimer les distances (utilisée les méthodes ranges-based pour l'estimation de distance), avec la connaissance de positions de ces nœuds ancrés. Ils dérivent ensuite de façon distribuée la position de tous les nœuds dans le réseau, à la fin du processus de localisation, chaque nœud doit connaître sa position sans l'aide d'un ordinateur central qui effectuerait les calculs ou GPS.

Les méthodes centralisées demanderaient trop de communication pour l'acheminement des informations vers station de base et consommeraient trop d'énergie donc on considère qu'une méthode distribuée est nécessaire pour les grands réseaux.

Dans cette partie, nous nous focalisons sur les choix des techniques et méthodes dans les phases de localisation :

2.1 Choix métrique d'estimation de distance :

Ce choix est pour déterminer les distances entre les nœuds capteurs. Nous avons choisi dans notre sujet d'étude la méthode de localisation par RSSI par ce que celle ne consomme pas beaucoup d'énergie avec un degré de précision élevé en plus le coût d'équipement moins cher, de plus notre environnement de développement peut simuler cette valeur, cette méthode est basée sur la puissance reçue associée à un modèle de propagation dans l'environnement (voir les graphes dans la section 3.2.4 dans le chapitre 2). Le modèle utilisé dans notre travail est le modèle de Friis.

2.2 Choix métriques d'estimation de position et Implémentation les algorithmes de localisation :

Puisque notre algorithme est distribué et après l'estimation de distance par les nœuds qui sont dans le protège de communication des nœuds ancrés par l'utilisation de méthode RSSI, chaque nœud peut calculer sa position par utilisation des méthodes d'estimation de position, dans cette phase nous avons opté pour l'utilisation de deux méthodes d'estimation de distance à savoir:

- Ce pseudo code exécutée pour chaque nœud :

```

Entrées : List des Nœuds Ancres AL= {1, 2, 3, ..., m}, counter=0 //incrémenter après chaque réception
Pour C=1 dans AL
  Pour S=m+1 vers N dans
    Envoi_Messages( $x_i(C)$ ) //broadcast des coordonnées aux voisins du nœud C
    Si  $Dist(C,S) < r$  //si S est dans la zone de captage de C
      Reception_Messages(Packet p) //récupérer les coordonnées par les nœuds voisins
      Estimation_Distance//estimation de distance par Rssi
      Sauvgarder_Données//enregistre les données concernant les coordonnées & la distance
      //pour chaque message réceptionné par les voisins ancres
      Counter++
    Si counter ==3
      Trilatération() //si le nœud reçu 3 messages, il calculé sa position
  Envoi_Messages( $x_i(S)$ ) //broadcast des coordonnées aux voisins du nœud S
  
```

Figure 3.2: Pseudo-code pour la Algorithme de localisation LTTA

Si un nœud estime sa position, il diffuse alors ses coordonnées pour aider ses voisins à calculer leurs positions.

2.2.2 Triangulation :

Le principe de Triangulation est que chaque nœud capable d'estimer sa position à l'aide de seulement deux nœuds ancres (présenté précédemment dans chapitre 2), ce qui facilité le processus de localisation surtout dans les réseaux qui ont un nombre réduit de nœuds ancres, c'est-à-dire que la Triangulation exige que les nœuds qui ne connaissent pas leurs positions d'être dans le champ de captage de seulement deux nœuds.

➤ Algorithme de localisation basée sur LTGA :

Localisation Triangulation algorithm nommées par LTGA c'est un algorithme distribué.

- Ce pseudo code exécutée pour chaque nœud :

```

Entrées : List des Nœuds Ancres AL= {1, 2, 3, ..., m}, counter=0 //incrémenter après chaque réception
Pour C=1 dans AL
  Pour S=m+1 vers N dans
    Envoi_Messages( $x_i(C)$ ) //broadcast des coordonnées aux voisins du nœud C
    Si  $Dist(C,S) < r$  //si S est dans la zone de captage de C
      Reception_Messages(Packet p) //récupérer les coordonnées par les nœuds voisins
      Estimation_Distance//estimation de distance par Rssi
      Sauvgarder_Données//enregistre les données concernant les coordonnées & la distance
      //pour chaque message réceptionné par les voisins ancres
      Counter++
    Si counter ==2
      Triangulation () //si le nœud reçu 2 messages, il calculé sa position
    Envoi_Messages( $x_i(S)$ ) //broadcast des coordonnées aux voisins du nœud S
  
```

Figure 3.3: Pseudo-code pour la Algorithme de localisation LTGA

2.2.3 Comparaison entre les deux méthodes :

Comme nous avons présenté précédemment que le processus de localisation basé sur l'algorithme LTGA exige que chaque nœud recevoir trois messages envoyés par trois nœuds ancres, et pour l'algorithme LTGA deux messages à partir de deux ancres.

Pour le premier cas (LTGA), le nombre des nœuds ancres doit être plus grand que l'algorithme LTGA, et pour meilleur localisation, estimation de distance par utilisation de trois ancres doit être plus précise que deux ancres.

2.3 - méthode proposée :

Notre proposition consiste à combiner les deux méthodes la localisation basé sur l'algorithme LTGA et la localisation basé sur l'algorithme LTGA (Trilatération et Triangulation) c'est-à dire si un nœud reçoit deux messages à partir de deux voisins ancres il lance l'algorithme LTGA, et après l'estimation de position si le même nœud reçoit le troisième message, il lance LTGA et mis à jour sa position.

- Ce pseudo code exécutée pour chaque nœud :

```

Entrées : List des Nœuds Ancres AL= {1, 2, 3, ..... m}, counter=0//incrémenter après chaque réception
Pour C=1 dans AL
  Pour S=m+1 vers N
    Envoi_Messages(xi(C)) //broadcast des coordonnées aux voisins du nœud C
    Si  $Dist(C,S) < r$  //si S est dans la zone de captage de C
      Reception_Messages(Packet P)//récupérer les coordonnées par les nœuds voisins
      Estimation_Distance//estimation de distance par Rssi
      Sauvegarder_Données//enregistre les données concernant les coordonnées & la distance
      //pour chaque messages réceptionné par les voisins ancres
      Counter++
      Si Counter==2
        Triangulation()
      Si Counter==3
        Trilatération()
    Envoi_Messages(xi(S)) //broadcast des coordonnées aux voisins du nœud S
  
```

Figure 3.4 : Pseudo-code sur la Proposition d'Algorithme de localisation.

2.4 Utilisation des nœuds ancres mobile

Il y'a deux types d'ancres :

- **Ancre statique**
- **Ancre mobile**

Le grand nombre des ancres nécessaires, leur coût, leur faible précision et leur forte consommation d'énergie, motivent l'utilisation d'un seule nœud ancre mobile. Au lieu d'avoir plusieurs ancres statiques, un seul mobile est déployé avec les nœuds, puis il traverse la zone de surveillance en communiquant avec les autres nœuds afin de les aider à s'auto-localiser comme montre la figure 3.5. L'ancre mobile diffuse des informations tout au long de sa trajectoire. Il peut être un opérateur humain, un robot déployé avec le réseau de capteurs, ou dans le cas d'un déploiement d'un avion, l'avion lui-même

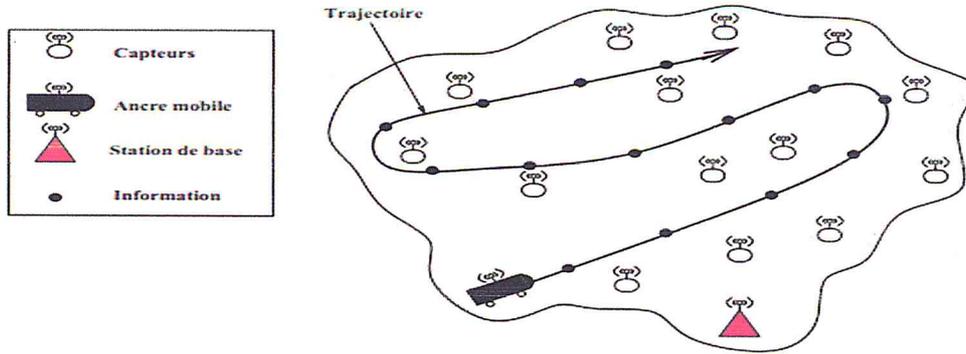


Figure 3.5: Une seule ancre mobile pour aider les nœuds à s’auto-localiser

Notre idée est d'utiliser les algorithmes que nous avons déjà présenté avec la mobilité des nœuds ancrés pour gagner le coût d'équipements des nœuds ancrés (GPS), on suppose qu'un nœud ancre X équipé par GPS ayant une trajectoire spécifique, à chaque moment spécifique il diffuse des paquets vers ces voisins, si un nœud reçoit trois ou deux messages du nœud ancre mobile dans des différents moments, il calculera sa position ensuite il exécute l'algorithme de localisation selon le nombre de messages reçus (si 2 messages alors Triangulation , si 3 messages alors Trilatération).

3-Implémentation :

Avec l'évolution des réseaux sans fil et l'élaboration de plusieurs normes pour ces réseaux, et avec le besoin des simulations dans le contexte de l'évaluation des performances, de nombreux simulateurs des réseaux ont été développés. Les simulateurs les plus connus sont : NS-2 (Network Simulator 2), OPNET et GloMoSim/Qualnet, Tossim etc. [20]

Alors on va présenter l'environnement de développement (NS2) et leur étape de l'implémentation de notre protocole réseau sur cet outil pour localiser tous les nœuds dans un réseau de capteur ainsi leur objet et composants.

3.1- NS2 network simulator 2 :

Dans notre projet nous allons choisir le NS-2 (open source) pour la simulation de notre protocole de localisation des capteurs dans réseau donc NS est un outil logiciel de simulation de réseaux informatiques. Il est essentiellement élaboré avec les idées de la conception par objets, de la réutilisation du code et de modularité. Il est aujourd'hui un standard de référence en ce domaine, plusieurs laboratoires de recherche recommandent son utilisation pour tester les nouveaux protocoles. Le simulateur NS actuel est particulièrement bien adapté aux

réseaux à commutation de paquets et à la réalisation de simulations de grande taille (le test du passage à l'échelle). Il contient les fonctionnalités nécessaires à l'étude des algorithmes de routage unicast ou multicast, des protocoles de transport, de session, de réservation, des services intégrés, des protocoles de localisation

NS-2 offre plusieurs avantages qui sont :

- NS2 est open source.
- Il est extensible, donc n'importe qui peut ajouter votre propre protocole, ou faire une modification pour tester son algorithme.
- NS2 est orienté objet basé sur C++.
- NS2 offre un mécanisme pour récupérer la valeur de Rssi pour l'estimation de distance (nous allons en discuter plus tard)

À la simulation, NS utilise OTcl est un langage de commande comme le shell UNIX mais qui sert à contrôler les applications. Son nom signifie Tool Command Language. Tcl offre des structures de programmation telles que les boucles, les procédures ou les notions de variables. Il y a deux principales façons de se servir de Tcl: comme un langage autonome interprété ou comme une interface applicative d'un programme classique écrit en C ou C++. En pratique, l'interpréteur Tcl se présente sous la forme d'une bibliothèque de procédures C qui peut être facilement incorporée dans une application. Cette application peut alors utiliser les fonctions standards du langage Tcl mais également ajouter des commandes à l'interpréteur.

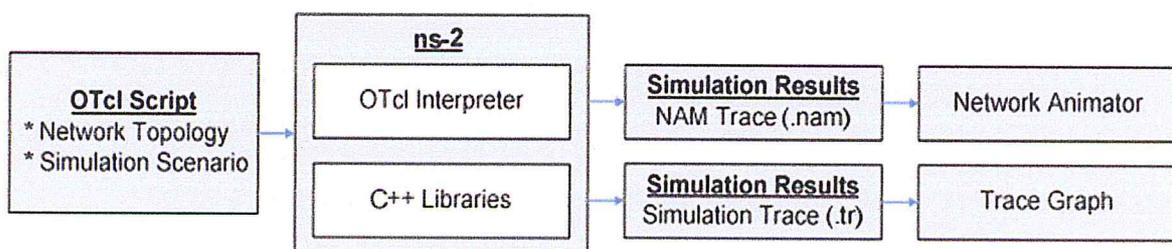


Figure 3.6 : NS2 schémas

Le résultat d'une simulation est un fichier trace d'extension «.tr », contenant tous les événements de la simulation. Un traitement ultérieur de ce fichier permet d'en soustraire l'information souhaitée. Par ailleurs, le simulateur permet la création d'un fichier d'animation

(d'extension .nam), est un outil de visualisation qui présente deux intérêts principaux : représenter la topologie d'un réseau décrit avec NS-2, et afficher temporellement les résultats d'une trace d'exécution NS-2.

➤ Les composants du réseau

Les composants du réseau sont : nœud, lien, la file d'attente, etc. Certains de ces derniers sont des composants simples, ils sont créés à partir de classes C ++, les autres sont des éléments composés. Ces composants sont

- Node : un nœud peut être une machine hôte, un switch, un routeur, une passerelle, etc
- Agent : La classe agent fournit des méthodes utiles au développement, C'est la classe de base pour définir des nouveaux protocoles dans NS-2.
- Application : Classe mère de toutes les applications (ftp, telnet, web).
- Queue : la classe mère de tous les buffers (DropTail, RED)
- LinkDelay : cette classe simule le délai de propagation et le temps de transmission sur les liens du réseau.
- Packet : la classe de tous les paquets circulant sur le réseau.
- TimerHundler : la classe mère de tous les timers (temporisateurs) utilisés par les protocoles du réseau.

3.2-Les étapes d'implémentation :

- Installation de NS2.34 sous Ubuntu 10.10 (voir annexe A)
- Intégration des packages de **LOCP** dans NS2 (voir annexe B)
- Les fichiers principaux de protocole de localisation **LOCP**
- L'implémentation phase d'estimation de distance **RSSI**
- L'implémentation de **LTTA & LTGA**
- Implémentation de notre proposition.
- Utilisation des ancres mobiles.

3.3 Implémentation un nouveau protocole de localisation pour RCSF

L'objectif de notre étude consiste à localiser les nœuds capteurs. Nous avons donc utilisé les composants mentionnés précédemment pour construire un nouveau protocole de localisation qui s'appelle **Locp (Localization protocol)** qui utilise les informations (échange des paquets) fournies par la couche réseau.

Notre protocole de localisation **Locp est implémenté sur 3 fichiers principaux :**

1. Le fichier **Locp_Packet.h** : ici sont déclarés les types de paquets qu'utilise **Locp** dans les échanges entre les nœuds du réseau.
2. Le fichier **Locp.h** : c'est le fichier principal (header) dans lequel sont définis tous les timers (temporisateurs) & les signatures méthodes nécessaires, et l'agent qui exécute les fonctionnalités du protocole.
3. Le fichier **Locp.cc** : dans lequel tous les timers, les Tclhooks (liens entre les composants c++ et ceux d'OTcl) sont implémentés, ainsi que l'agent et ces fonctionnalités.

Nous allons commencer par le fichier **Locp_packet** :

- **Locp_Packet.h** : ce fichier contient les types de paquets pour notre protocole **LOCP** (Le format des paquets **Locp_beacon**), c'est un paquet qui est diffusé par les nœuds ancrés pour informer les autres nœuds non ancrés de ces positions

Le fichier **locp.h** contient des définitions & signatures des méthodes utilisées dans notre protocole voir

Finalement le dernier fichier (classe) **locp.cc** qui représente le fichier le plus important dans notre implémentation 2 points principaux qui sont : méthode **send_beacon**, méthode **recv**.

3.4 Implémentation la phase d'estimation de distance RSSI

Plusieurs techniques trouvées pour l'estimation de distances entre les nœuds dans un réseau de capteurs sans fil. Parmi ces méthodes on trouve celle basée sur le RSSI. La figure suivante montre l'implémentation de cette technique :

```
double
LOCP::Distance(double Pr, double Pt, double Gt, double Gr, double L, double lambda)
{
    return sqrt((Pt * Gt * Gr * lambda * lambda) / (L * Pr)) / (4 * PI); //find it in propagation.cc
}
```

Figure 3.6 : estimation de distance

Selon la figure 3.6 et selon l'étude du chapitre 2 (section 3.2.1) la valeur de Pt (power_transmitted) et Pr (power_received) sont très importants dans le calcul de l'équation de Friis, Pr est un paramètre qui varie selon la distance entre les nœuds capteurs, chaque fois que la distance entre les nœuds augmente, la valeur de Pr est décrétementé.

3.5 L'implémentation de LTTA & de LTGA

✚ L'implémentation de LTTA :

La figure ci-dessous représente l'implémentation de l'algorithme LTTA écrit avec langage c++, la méthode Trilateration_model se déclenche après chaque réception de paquets

```
void
LOCP::Trilateration_model(Packet *p){
    struct hdr_cmn *ch = HDR_CMN(p);
    struct hdr_ip *ih = HDR_IP(p);
    if(ch->ptype() != PT_LOCP) return;
    if(ch->ptype() == PT_LOCP) {
        struct hdr_locp_beacon *bcn = HDR_LOCP_BEACON(p);
        if(Ancr_==1 || counter==3){
            Packet::free(p);
            return;
        }
        double ReceivedX=bcn->X_pos;
        double X = min + (max - min) * r;
        Pr=Pr*X;
        double Dist=Distance(Pr,Pt,Gt,Gr,L,lambda);

        AddReceivedData(Matric,bcn->X_pos,bcn->Y_pos,Dist,counter);
        counter++;
        if(counter==3){ Trilateration();}
    }
}
```

Figure 3.7 : l'implémentation de LTTA avec C++

✚ L'implémentation de LTGA

La figure sous dessous représente l'implémentation de l'algorithme LTGA écrit avec langage c++ :

```

void
LOCP::Triangulation_model(Packet *p){
    struct hdr_cmn *ch = HDR_CMN(p);
    struct hdr_ip *ih = HDR_IP(p);
    if(ch->ptype() != PT_LOCP) return;
    if(ch->ptype() == PT_LOCP) {
        struct hdr_locp_beacon *bcn = HDR_LOCP_BEACON(p);
        if(Ancr_==1 || counter==2){
            Packet::free(p);
            return; }
        double Pr=bcn->beacon_received_power;
        double min=0.9;
        double max=0.98;
        double r = Random::uniform();
        double X = min + (max - min) * r;
        Pr=Pr*X;
        double Dist=Distance(Pr,Pt,Gt,Gr,L,lambda);

        AddReceivedData(Matric,bcn->X_pos,bcn->Y_pos,Dist,counter);
        counter++;
        if(counter==2){ Triangulation();}    ]}

```

Figure 3.8 : l'implémentation de LTGA

3.6 Implémentation de notre proposition

La figure ci-dessous représente l'implémentation de l'algorithme LTGA écrit avec langage c++, à chaque réception des paquets les nœuds déclenchent cette procédure :

```

void
LOCP::Melange_TrianguTrelateration_model(Packet *p){
    struct hdr_cmn *ch = HDR_CMN(p);
    struct hdr_ip *ih = HDR_IP(p);
    if(ch->ptype() != PT_LOCP) return;
    if(ch->ptype() == PT_LOCP)
    {
        struct hdr_locp_beacon *bcn = HDR_LOCP_BEACON(p);
        if(Ancr_==1 || counter==3)
        {
            Packet::free(p);
            return;
        }
        double Pr=bcn->beacon_received_power;
        double min=0.9;
        double max=0.98;
        double r = Random::uniform();
        double X = min + (max - min) * r;
        Pr=Pr*X;
        double Dist=Distance(Pr,Pt,Gt,Gr,L,lambda);
    }
}

```

Figure 3.9: l'implémentation de notre proposition

3.7 implémentation des ancres mobiles :

Comme nous l'avons vu dans le chapitre précédent il existe deux types d'ancres, ancre statique & ancre mobile, stratégies de mobilité dans lequel le nœud ancre peut se déplacer.

L'idée principale dans le cadre de notre stratégie est de déplacer les nœuds ancrés sur les frontières. Le nœud ancre est situé initialement à la position (0, 0) et il va se déplacer en passant sur les positions {1, 2, 3, 4, 5, 6, 7, 8, 9 jusqu'à 24} correspondants aux coordonnées {(0, 30), (0, 60), (0, 90), (0, 120), (30, 0), (30, 30), (30, 60), (30, 90), (30, 120), (60, 0), (60, 30), (60, 60), (60, 90), (60, 120), (90, 0), (90, 30), (90, 60), (90, 90), (90, 120), (120, 0), (120, 30), (120, 60), (120, 90), (120, 120)} respectivement.

Par l'utilisation du setdest, qui peut être utilisé pour créer le cheminement des mouvements pour les nœuds mobiles, la figure suivante illustre l'utilisation de cette méthode pour l'objectif de la mobilité des nœuds ancrés :

```

$ns at 1.0 "$node_(0) setdest 0.0 30.0 30.0"
$ns at 2.0 "$node_(0) setdest 0.0 60.0 30.0"
$ns at 3.0 "$node_(0) setdest 0.0 90.0 30.0"
$ns at 4.0 "$node_(0) setdest 0.0 120.0 30.0"

$ns at 5.0 "$node_(0) setdest 30.0 0.0 30.0"
$ns at 6.0 "$node_(0) setdest 30.0 30.0 30.0"
$ns at 7.0 "$node_(0) setdest 30.0 60.0 30.0"
$ns at 8.0 "$node_(0) setdest 30.0 90.0 30.0"
$ns at 9.0 "$node_(0) setdest 30.0 120.0 30.0"

$ns at 10.0 "$node_(0) setdest 60.0 0.0 30.0"
$ns at 11.0 "$node_(0) setdest 60.0 30.0 30.0"
$ns at 12.0 "$node_(0) setdest 60.0 60.0 30.0"
$ns at 13.0 "$node_(0) setdest 60.0 90.0 30.0"
$ns at 14.0 "$node_(0) setdest 60.0 120.0 30.0"

$ns at 15.0 "$node_(0) setdest 90.0 0.0 30.0"
$ns at 16.0 "$node_(0) setdest 90.0 30.0 30.0"
$ns at 17.0 "$node_(0) setdest 90.0 60.0 30.0"
$ns at 18.0 "$node_(0) setdest 90.0 90.0 30.0"
$ns at 19.0 "$node_(0) setdest 90.0 120.0 30.0"

$ns at 20.0 "$node_(0) setdest 120.0 0.0 30.0"
$ns at 21.0 "$node_(0) setdest 120.0 30.0 30.0"
$ns at 22.0 "$node_(0) setdest 120.0 60.0 30.0"
$ns at 23.0 "$node_(0) setdest 120.0 90.0 30.0"
$ns at 24.0 "$node_(0) setdest 120.0 120.0 30.0"
    
```

Figure 3.10 : la mobilité des nœuds ancrés par utilisation de setdest

Selon la figure ci-dessus, la première ligne en jaune, le noeud_(0) au moment 1.0s commence à se déplacement vers la destination (0,30) à une vitesse de 30 m/s. Cette ligne de commande peut être utilisée pour modifier la direction et la vitesse de déplacement des nœuds mobiles. Dans chaque mouvement de la noeud_(0) diffuser sa coordonnée sur ces voisins.

Pour connaitre quelle est le moment ou le nœud_(0) diffuser sa coordonnée sur ces voisins, selon le théorème de Galilée [21], la vitesse=distance/temps, la distance Dist ((nouveau_x - ancien_x)²+ (nouveau_y - ancien_y)²) est toujours égale a 30m alors que la vitesse égale à 30m/s on aura donc le temps d'envoi est toujours égale a 1s alors on utilise chronomètre

(Timers), à chaque seconde, le nœud change sa position vers une autre position, et diffuse l'information concernant sa nouvelle position à ces nouveaux voisins, la figure suivante illustre l'utilisation des timers pour déclencher cette procédure.

```
//=====
void
wfrpBeaconTimer::handle(Event*) {
    if(agent->Ancr_){
        agent->send_beacon(agent->iNode->X(),agent->iNode->Y());
    }
    Scheduler::instance().schedule(this, &intr,1);
}

//=====
```

Figure 3.11 : utilisation des timers

La figure suivante représente la diffusion de nœud_(0) à des moments mentionnée précédemment :

```
S (0.000000): send beacon by 0 for its neighbors
channel.cc:sendUp - Calc highestAntennaZ_ and distCST_
highestAntennaZ_ = 1.5, distCST_ = 79.7
SORTING LISTS ...DONE!
S (1.000000): send beacon by 0 for its neighbors
S (2.000000): send beacon by 0 for its neighbors
S (3.000000): send beacon by 0 for its neighbors
S (4.000000): send beacon by 0 for its neighbors
S (5.000000): send beacon by 0 for its neighbors
S (6.000000): send beacon by 0 for its neighbors
S (7.000000): send beacon by 0 for its neighbors
S (8.000000): send beacon by 0 for its neighbors
S (9.000000): send beacon by 0 for its neighbors
S (10.000000): send beacon by 0 for its neighbors
S (11.000000): send beacon by 0 for its neighbors
S (12.000000): send beacon by 0 for its neighbors
S (13.000000): send beacon by 0 for its neighbors
S (14.000000): send beacon by 0 for its neighbors
S (15.000000): send beacon by 0 for its neighbors
S (16.000000): send beacon by 0 for its neighbors
S (17.000000): send beacon by 0 for its neighbors
S (18.000000): send beacon by 0 for its neighbors
S (19.000000): send beacon by 0 for its neighbors
S (20.000000): send beacon by 0 for its neighbors
S (21.000000): send beacon by 0 for its neighbors
S (22.000000): send beacon by 0 for its neighbors
S (23.000000): send beacon by 0 for its neighbors
S (24.000000): send beacon by 0 for its neighbors
end simulation
```

Figure 3.12 : diffusion de messages par nœud 0

✚ Savoir la position réelle sous NS2

Pour apprendre la position réelle de chaque nœud capteur, suivez les étapes suivantes :

On ajoute la bibliothèque *Mobilenode.h* dans le fichier haeder *.h* :

```
#ifndef __wfrp_h__
#define __wfrp_h__

#include <cmu-trace.h>
#include <mobilenode.h>
#include <vector>
#include <prqueue.h>
#include <classifier/classifier-port.h>
```

Figure 3.13 : ajouter la bibliothèque *Mobilenode.h*

Et aussi ajouter cette ligne (dans la ligne 168):

```
MobileNode *iNode;
```

Figure 3.14 : ajouter la déclaration

On ajoute ce code dans le constructeur de l'agent (le fichier *.cc* dans la ligne 206):

```
iNode = (MobileNode *) (Node::get_node_by_address(index));
```

Il est facile de savoir la position de tous les nœuds dans NS par l'instruction suivant

```
double x=iNode->X();
double y=iNode->Y();
```

4-Conclusion

Nous avons présenté dans ce chapitre la conception de notre algorithme de localisation par le choix de chaque phase (estimation de distance et de position). Les deux algorithmes LTGA et LTGA (Trilatération, Triangulation), ont été bien détaillés. Nous avons, ensuite, expliqué le fonctionnement de notre algorithme qu'on a proposé, puis nous avons passés vers l'utilisation des ancras mobile dans le processus de localisation, avec l'étude de l'objectif de protocole LOCP. Enfin, l'implémentation de notre proposition sous NS2 et ses caractéristiques ont été détaillée.

Dans le chapitre suivant nous allons détailler et exposer nos résultats de notre implémentation.

Chapitre 04 :

Interprétation des résultats

1-Introduction

L'objectif du protocole LOCP, comme on a pu voir précédemment, est de localiser tous les nœuds de capteurs sans l'utilisation des équipements GPS. Dans ce chapitre, nous allons exposer les résultats de notre implémentation c'est-à-dire nous allons présenter une série de simulations réalisées à l'aide du simulateur NS-2 .Nous avons utilisé la version 2.34 installée sous Ubuntu (voir Annexes A). Le choix de ce dernier est motivé par ses caractéristiques. Les objectifs de ces simulations sont d'abord évaluer les performances du protocole LOCP selon certains paramètres de mesure, ensuite comparer et conclure l'impact des quelques paramètres sur le fonctionnement des deux algorithmes LTTA et LTGA, puis comparer entre ces deux algorithmes avec notre proposition sur des nœuds ancrés fixes et des ancrés mobiles.

2-Intérêt et nécessité de la simulation

L'évaluation des performances du protocole de localisation peut se faire en utilisant trois techniques, à savoir : la méthode analytique, les mesures (la distance entre la réel position et la position estimer), et enfin la simulation. Le recours à la simulation présente de nombreux avantages. En effet, elle est plus rapide, moins coûteuse en ressources, répétable et permettant d'isoler des paramètres qui peuvent parfois affecter les performances. De plus, il existe des scénarios très difficiles à étudier dans la réalité [39].

2.1-Les Métriques de Performance

Les simulations sont faites par rapport à deux métriques, dans le but de tester notre protocole sur différents aspects. Les principales métriques d'évaluation des performances de notre protocole de localisation LOCP sont :

- ✓ La moyenne du taux d'erreur des nœuds localisée.
- ✓ La précision de nos algorithmes de localisation.

Calculer le taux d'erreur

Pour évaluer nos algorithmes de localisation et mesure la performance de notre protocole, ns2 offre un mécanisme pour savoir la réelle position du nœud [40].

Le Taux d'erreur c'est la distance entre la position estimée et la position réelle de nœud capteur :

$$\text{Le Taux d'erreur} = \text{Distance (La position estimé, La position réelle)}$$

2.2- les paramètres utilisés dans les simulations :

Tout d'abord il faut commencer par initialiser les valeurs des paramètres à utiliser pendant les simulations. Le simulateur NS-2 implémente les différentes couches nécessaires (application, transport, routage, MAC et physique) pour la simulation d'un réseau de capteur. Le tableau 4.1 résume les différents paramètres utilisés dans les simulations.

Paramètres	Valeur
Protocole de localisation	LOCP
Modèle de propagation	Two-Ray-Ground
Couche MAC	Mac/802_11
Portée de transmission	30 mètres
Temps de simulation	10 seconds
Surface de simulation	120*120 m²

Tableau 4.1: Paramètres de simulation standards utilisés par NS2

2.3- Paramètres de simulation de notre protocole de localisation :

Paramètres	Valeur
Taille de réseau (en nœuds)	{25, 50, 75, 100}
Nombre des ancrs mobiles	16 capteurs
Algorithme de localisation	Triangulation, Trilatération, Proposition

Tableau 4.2: les paramètres de simulation d'LOCP.

Notre étude consistée à crée du réseau du taille différent (25, 50, 75,100) en utilisant 16 ancre et à chaque fois on changer la méthode pour localiser les capteurs (**Triangulation, Trilatération, Proposition**)

3-Simulation et discussion :

Dans cette partie, nous présentons les résultats de simulation et nous discuterons ensuite les résultats obtenus.

3.1-Résultat d’algorithme LTTA

Afin de bien montrer le fonctionnement de l’algorithme LTTA dans le chapitre précédent, plusieurs simulations ont été faites, pour évaluer l’exactitude du protocole LOCP basé sur LTTA, nous avons calculé le taux d’erreur, Les résultats obtenus sont présentés dans la figure suivant (figure 4.1) :

```
S (0.236074): send beacon by 79 its coordinate is (80.371938,11.166406)
0.149372
la position calculé de nœud 79 est (80.487294,11.071511) mais la réel position est (80.371938,11.166406) Taux d'erreur est 0.149372S (0.236074): send beacon by 65 its coordinate is (83.179863,3.950771)
1.442810
la position calculé de nœud 65 est (83.849750,2.672901) mais la réel position est (83.179863,3.950771) Taux d'erreur est 1.442810S (0.236074): send beacon by 87 its coordinate is (85.220853,6.884190)
1.033155
la position calculé de nœud 87 est (86.068996,6.294222) mais la réel position est (85.220853,6.884190) Taux d'erreur est 1.033155S (0.314154): send beacon by 24 its coordinate is (17.306683,73.418627)
0.275615
```

Figure 4.1: résultats de simulation de l’algorithme LTTA

N°	La réelle position	La position estimée	Taux erreur
79	(83.1793, 3.9507)	(80.4872, 11.0715)	1.4428
65	(85.2208, 6,8841)	(83,8497.2, 6729)	1 ,4428
87	(85,2208 .6, 8841)	(86,0689 .6, 2942	1,0331

Tableau 4.3: résultats de simulation de l’algorithme LTGA

3.2-Résultat d’algorithme LTGA

La figure suivante présente les résultats de simulation de LOCP associe avec LTGA :

```

S (0.838016): send beacon by 24 its coordinate is (36.627486,98.159427)
la position calculé de nœud 58 est (34.153947,97.826177) mais la réel position e
st (34.449705,96.183862) Taux d'erreur est 1.668733
S (0.838016): send beacon by 58 its coordinate is (34.449705,96.183862)
la position calculé de nœud 85 est (5.589008,88.996384) mais la réel position es
t (31.647225,94.905406) Taux d'erreur est 26.719790
S (1.148950): send beacon by 85 its coordinate is (31.647225,94.905406)
la position calculé de nœud 65 est (58.534554,30.310427) mais la réel position e
st (59.542372,28.649541) Taux d'erreur est 1.942740
S (1.302081): send beacon by 65 its coordinate is (59.542372,28.649541)
la position calculé de nœud 46 est (93.618861,99.943667) mais la réel position e
st (94.032206,99.293471) Taux d'erreur est 0.770461
S (1.320765): send beacon by 46 its coordinate is (94.032206,99.293471)
    
```

Figure 4.2: résultats de simulation de l’algorithme LTGA

N°	La réelle position	La position estimée	Taux erreur
58	(34,4497.96, 1838)	(34,1539. 97,8261)	1.6687
85	(31.6472, 94,9054)	(5,5890.88, 9963)	26 ,7197
65	(59,5423 .28, 6495)	(58,5345 .30, 3104)	1,9427

Tableau 4.4: résultats de simulation de l’algorithme LTGA

La figure montre que LTGA engendre un taux d’erreur considérable, L’avantage de cet algorithme est qu’il est suffisant pour chaque nœud de recevoir seulement deux messages pour localiser sa position à l’opposé de LTTA qui exiger 3 messages.

3.3-Comparaison entre LLTA et LTGA :

Pour évaluer la performance et faire la comparaison entre les deux algorithmes, la figure 4.3 montre les résultats obtenus concernant la métrique de taux d’erreur (précision).

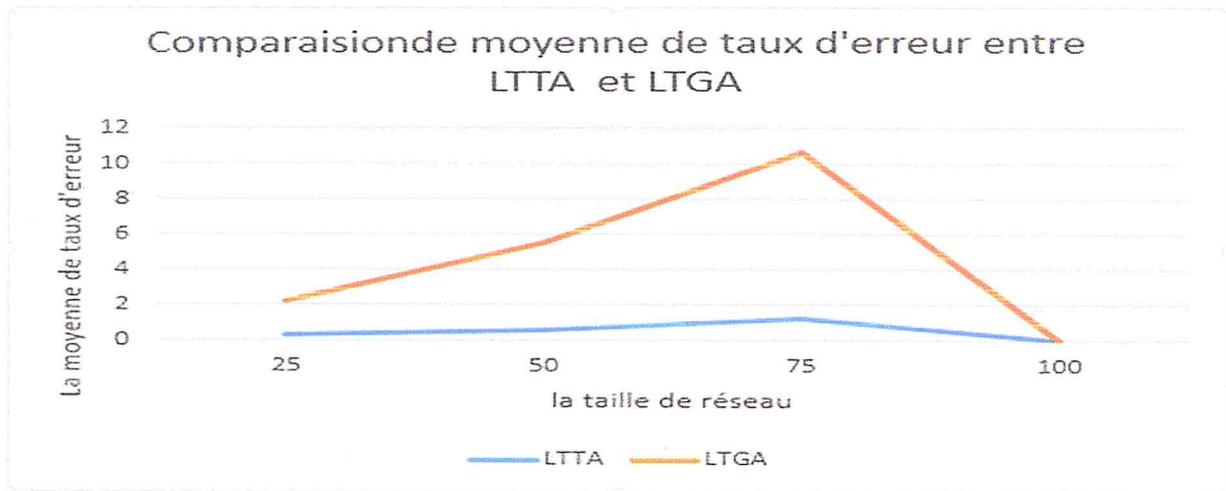


Figure 4.3 : comparaison de performance entre LLTA & LTGA

Discussion

Comme présenté dans la figure 4.3, il est évident que la performance de l'algorithme LLTA est meilleure que LTGA, le taux d'erreur est moyen dans tous les scénarios de simulation. D'autre part. Nous concluons que le taux d'erreur a une relation avec le rapprochement entre les nœuds.

3.4-Distribution des nœuds Ancre :

Pour garantir que tous les nœuds reçoivent les paquets à partir des nœuds ancrés, c'est-à-dire que tous les nœuds sont capables d'estimer leurs positions.

Si les nœuds ancrés sont distribués de façon aléatoire (figure 4.4), il est évident que certains nœuds ne reçoivent pas les paquets envoyés par les nœuds ancrés, ce qui veut dire que le nombre de nœuds localisés sera diminué. Donc, si nous avons placé les nœuds ancrés de manière organisée dans la surface de la simulation (voir figure 4.5), nous garantissons que tous les nœuds du réseau reçoivent les paquets à partir de différentes ancrés.

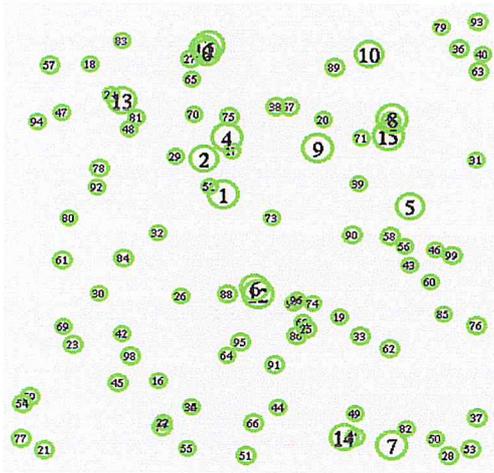


Figure 4.4: positionnement aléatoire des nœuds ancrés

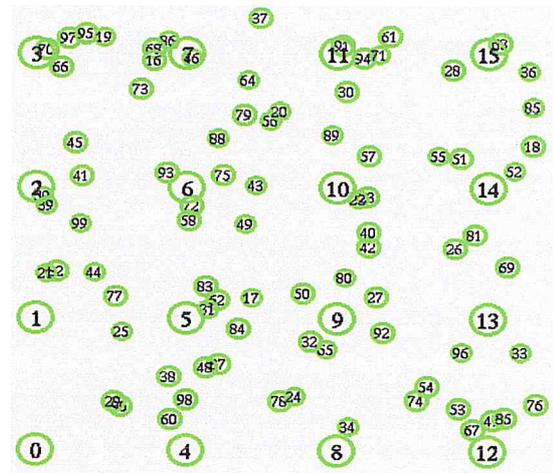


Figure 4.5: positionnement organisées des nœuds ancrés

3.5-Résultat de proposition :

La figure suivante présente les résultats de simulation de LOCP associées avec notre proposition :

```

la position calculé de nœud 62 est (63.330372,83.064670) mais la réel position e
st (63.741084,96.390711) Taux d'erreur est 13.332368
S (0.837916): send beacon by 62 its coordinate is (63.741084,96.390711)
la position calculé de nœud 19 est (44.847643,97.065274) mais la réel position e
st (44.951257,95.784072) Taux d'erreur est 1.285385
S (0.837916): send beacon by 19 its coordinate is (44.951257,95.784072)
la position calculé de nœud 20 est (42.745888,95.191020) mais la réel position e
st (42.898393,93.291542) Taux d'erreur est 1.905591
S (0.837916): send beacon by 20 its coordinate is (42.898393,93.291542)
la position calculé de nœud 54 est (99.916189,87.426916) mais la réel position e

```

Figure 4.6 : résultats de simulation de la proposition

N°	La réelle position	La position estimée	Taux erreur
62	(63,7410.96, 3907)	(63,3303. 83,0646)	13.3323
19	(44.9512, 95,7840)	(44.8476, 97,0652)	1 ,2853
20	(42,8983 .93, 2915)	(42,7458 .95, 1910)	1,9055

Tableau 4.5: résultats de simulation de l’algorithm proposition.

-Comparaison entre LLTA et LTGA avec la méthode proposée :

L'évaluation de ces algorithmes a été réalisée par rapport aux métriques de performance mentionnées précédemment qui sont :

1-Le taux d'erreur :

La figure suivante résume cette comparaison :

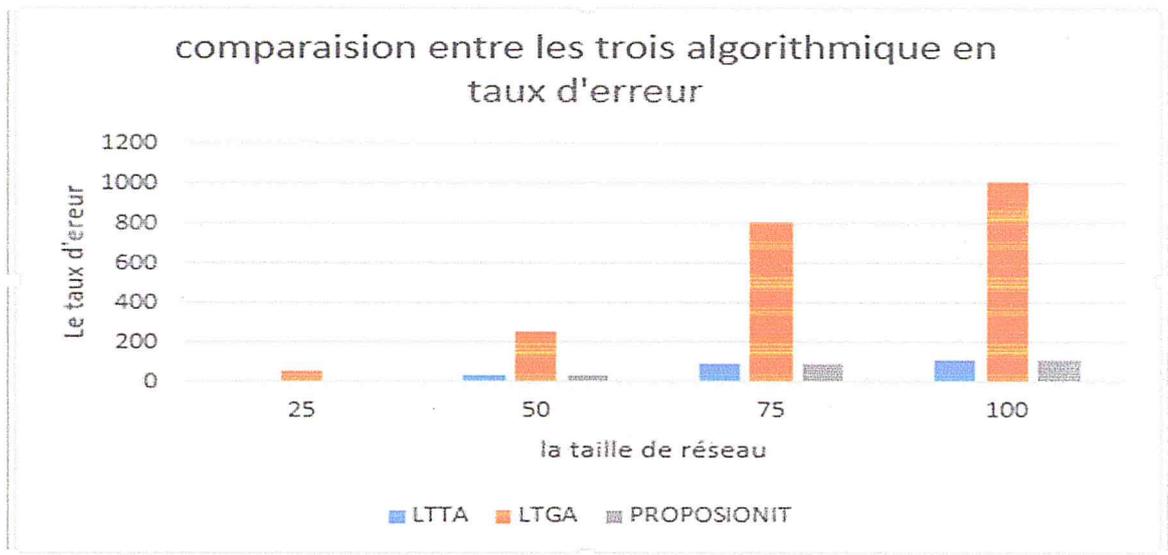


Figure 4.7: comparaison en taux d'erreur.

Discussion

Comme présenté dans la figure 4.7, nous remarquons dans tous les scénarios de simulation que le taux d'erreur dans les deux algorithmes (LTTA & proposition) est toujours le même. D'autre part, le taux d'erreur dans LTGA est toujours supérieur (Cela est dû à leurs principe expliquées précédemment)

Nous concluons que LTTA est plus performant que LTGA en taux d'erreur. notre proposition est plus performante que LTTA et LTGA en taux d'erreur est respective

3.6-Simulation des ancrs mobiles

L'utilisation des ancrs mobile peut réduire le cout d'équipement GPS. Le nœud ancre doit suivre une trajectoire spécifique. A chaque instant, ce nœud broadcast le paquet a ses voisins, les résultats sont visualisés dans la figure suivante :

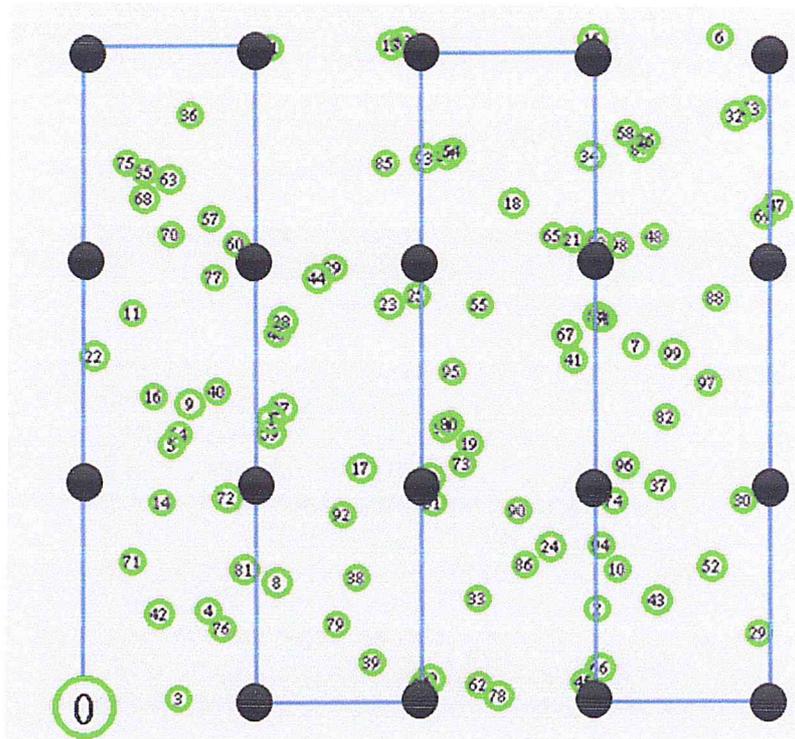


Figure 4.9 : la simulation par l'utilisation de nœud ancre

Comme présenté dans la figure précédente qu'à chaque période (1 second), le nœud ancre 0 broadcast le paquet à ces voisins, il y a des nœuds qui ont réussi de calculer leurs positions.

N°	La réelle position	La position estimée	Taux erreur
30	(31,1084.96, 3907)	(33,7018. 39,3995)	2.5969
38	(29.4992, 95,7840)	(31,9017. 93,827)	2 ,4155
32	(32,5931 .92, 2524)	(32,7453 .76, 2415)	3,9920

Tableau 4.6: résultats de simulation des ancrs mobile.

4-Conclusion

Nous avons exposé les résultats des deux algorithmes LTGA & LTGA et de notre méthode proposée. D'après l'implémentation, nous avons comparé les résultats obtenus entre les deux algorithmes LTGA & LTGA et notre algorithme proposée avec l'étude du positionnement des nœuds ancrés de façon aléatoire et organisée (augmentation des nœuds localisés dans le cas où les nœuds ancrés distribués de manière organisée), et l'étude de l'influence du changement de la portée de communication sur le nombre des nœuds localisés. Nous concluons que notre proposition est plus performante que LTGA en nombre des nœuds localisés, et elle est plus performante que LTGA en taux d'erreur. Enfin, nous avons utilisé le principe de la mobilité des nœuds ancrés dans le processus de localisation.

Conclusion général

Conclusion général :

Les réseaux de capteurs ont connu une grande évolution au cours des dernières années. Cette évolution a rencontré plusieurs contraintes dont la plus importante était la localisation. Cette nouvelle technologie qui a surgit après les grands progrès technologiques concernant le développement des capteurs intelligents capables de détecter des événements et de préciser sa position, Plusieurs recherches ont été faites pour la conception de l'algorithme qui tiennent compte de cette contrainte et en plus qui minimisent la consommation d'énergie. En effet, c'est dans le cadre de ce thème que s'oriente l'objectif de notre projet de fin d'études

Dans ce projet, nous avons donné un aperçu sur les réseaux de capteurs et présenté certaines de leurs applications. Nous avons aussi exposé les différents défis qui doivent surmonter la conception des protocoles de communication dans ces réseaux. Nous nous sommes intéressés aux algorithmes basés sur les méthodes Range-based qui utilisent les technologies (AOA, TOA et RSSI) et Les algorithmes basés sur Les méthodes range-free

Aussi nous avons étudié les deux algorithmes de localisation dans les RCSFs (*LTTA & LTGA*) qui sont pour la méthode (Trilatération & Triangulation), respectivement, dans le but de faire une comparaison entre eux à l'aide de simulateur NS2. Nous avons proposé un autre algorithme qui a-t-il combine les caractéristiques de les autres précédents algorithmes

Après avoir effectué plusieurs simulations et analysé les résultats obtenus, nous avons constaté que le *LTTA* est plus précis que le *LTGA*. En effet, en augmentant le nombre de nœuds dans le réseau, on obtient des résultats plus Précis, donc ils introduisent un lourd surcoût de communication ce qui entraîne une consommation d'énergie très importante.

Comme perspective, il serait très intéressant de joindre les avantages des algorithmes range-based en termes de précision aux algorithmes range-free. Par exemple rajouter le RSSI à l'algorithme *LTGA* afin de minimiser l'erreur de précision tout en gardant sa simplicité. Une bonne hybridation permettra d'avoir un algorithme de géo localisation

Nous avons faisi des comparaisons entre les trois algorithmes (*LTTA & LTGA & notre algorithme proposé*) nous concluions que l'algorithme proposé est plus précis que les autres deux algorithmes. Ayant comme avantage une meilleure précision, à moindre coût.

1. Annexe A : Installation NS2 sous Ubuntu :

- Installer les logiciels complémentaires avec la commande suivante :

```
sudo apt-get install build-essential autoconf automake libx11-dev libxmu-dev
```

- Installer le compilateur C++ avec la commande suivante:

```
sudo apt-get install g++-4.3
```

- Copier le package d'installation « ns-allinone-2.34.tar.gz » dans votre dossier personnel (/home/micman).

Extraire le fichier dans le même répertoire personnel. 5- Entrer dans le répertoire de ns-allinone-2.34/otcl-1.13 et modifier les fichiers suivants :

- Le fichier Makefile.in : remplacer

```
CC = @CC@ Par CC = gcc-4.3
```

- Le fichier Configue.in : remplacer `CC = @CC@` par `CC = gcc-4.3` 6- Lancer l'installation de NS2 :

```
- cd ns-allinone-2.34  
- ./install  
- ./validate
```

- Editer quelle que chemine. Dans le fichier "~/.bashrc"

```
- sudo gedit .bashrc
```

- Ajouter les lignes suivantes au fichier : Mettre à la place de « micman » votre répertoire personnel

```
1 # LD_LIBRARY_PATH  
2 OTCL_LIB=/home/micman/ns-allinone-2.34/otcl-1.13  
3 NS2_LIB=/home/micman/ns-allinone-2.34/lib  
4 X11_LIB=/usr/X11R6/lib  
5 USR_LOCAL_LIB=/usr/local/lib  
6 export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$OTCL_LIB:$NS2_LIB:$X11_LIB:$USR_LOCA  
7  
8 # TCL_LIBRARY  
9 TCL_LIB=/home/micman/ns-allinone-2.34/tcl8.4.18/library  
10 USR_LIB=/usr/lib  
11 export TCL_LIBRARY=$TCL_LIB:$USR_LIB  
12  
13 # PATH  
14 XGRAPH=/home/micman/ns-allinone-2.34/bin:/home/micman/ns-allinone-2.34/tcl8.4  
15 NS=/home/micman/ns-allinone-2.34/ns-2.34/  
16 NAM=/home/micman/ns-allinone-2.34/nam-1.14/  
17 PATH=$PATH:$XGRAPH:$NS:$NAM
```

- Sauvegarder.
- Redémarrer l'ordinateur.
- Taper dans un terminal « ns », il doit s'afficher le symbole « % », dans ce cas le NS2 a été bien installé.
- Installer xgraph avec la commande : `sudo apt-get install xgraph ygraph.`

2. Annexe B : Intégration du protocole LOCP sous NS2

L'intégration de protocole LOCP nécessite la modification des fichiers suivants :

- \$NS_ROOT/Makefile
- \$NS_ROOT/queue/priqueue.cc
- \$NS_ROOT/common/packet.h
- \$NS_ROOT/trace/cmu-trace.h
- \$NS_ROOT/trace/cmu-trace.cc
- \$NS_ROOT/tcl/lib/ns-packet.tcl
- \$NS_ROOT/tcl/lib/ns-lib.tcl
- \$NS_ROOT/tcl/lib/ns-agent.tcl
- \$NS_ROOT/tcl/lib/ns-mobilenode.tcl

- Nous commençons avec fichier Makefile `~/ns-allinone-2.34/ns-2.34/Makefile` ajouter la ligne suivant dans la ligne 269 : **mkdir LOCP.**
- ajouter les lignes suivantes dans `~/ns-allinone-2.34/ns-2.34/queue/priqueue.cc` dans la ligne line 93 :

```
// LOCP patch
case PT_LOCP:
```

- modifier le fichier `~/ns-allinone-2.34/ns-2.34/common/packet.h` :

```
// LOCP packet
static const packet_t PT_LOCP = 62;

// insert new packet types here
static packet_t PT_NTTYPE = 63;
```

- Puis ajouter dans la ligne 254 de `~/ns-allinone-2.34/ns-2.34/common/packet.h` :

```
type == PT_AODV ||
type == PT_LOCP)
```

- et dans la ligne 390 dans le même fichier :

```
// LOCP patch
name_[PT_LOCP] = "LOCP";
```

- Pour ajouter fonction de trace nous avons ajoutées la ligne suivant dans `~/ns-allinone-2.34/ns-2.34/trace/cmu-trace.h` dans la ligne 163:

```
void format_LOCP(Packet *p, int offset);
```

- Dans le fichier `~/ns-allinone-2.34/ns-2.34/trace/cmu-trace.cc` nous ajoutons les lignes suivantes dans la ligne 1071 :

```
// LOCP patch
void
```

Annexe

```
CMUTrace::format_LOCP(Packet *p, int offset)
{
    struct hdr_LOCP *wh = HDR_LOCP(p);
    struct hdr_LOCP_beacon *wb = HDR_LOCP_BEACON(p);
    struct hdr_LOCP_error *we = HDR_LOCP_ERROR(p);

    switch(wh->pkt_type) {
        case LOCP_BEACON:

            if (pt_->tagged()) {
                sprintf(pt_->buffer() + offset,
                    "-LOCP:t %x -LOCP:h %d -LOCP:b
%d -LOCP:s %d "
                    "-LOCP:px %d -LOCP:py %d -
LOCP:ts %f "
                    "-LOCP:c BEACON ",
                    wb->pkt_type,
                    wb->beacon_hops,
                    wb->beacon_id,
                    wb->beacon_src,
                    wb->beacon_posx,
                    wb->beacon_posy,
                    wb->timestamp);
            } else if (newtrace_) {
                sprintf(pt_->buffer() + offset,
                    "-P LOCP -Pt 0x%x -Ph %d -Pb %d
-Ps %d -Ppx %d -Ppy %d -Pts %f -Pc BEACON ",
                    wb->pkt_type,
                    wb->beacon_hops,
                    wb->beacon_id,
                    wb->beacon_src,
                    wb->beacon_posx,
                    wb->beacon_posy,
                    wb->timestamp);
            } else {
                sprintf(pt_->buffer() + offset,
                    "(BEACON)",
                    "[0x%x %d %d [%d %d] [%d %f]]
                    wb->pkt_type,
                    wb->beacon_hops,
                    wb->beacon_id,
                    wb->beacon_src,
                    wb->beacon_posx,
                    wb->beacon_posy,
                    wb->timestamp);
            }
            break;

        case LOCP_ERROR:
            // TODO: need to add code
            break;

        default:
#ifdef WIN32
            fprintf(stderr,
                "CMUTrace::format_LOCP: invalid LOCP
packet typen");
#else
            fprintf(stderr,
                "%s: invalid LOCP packet typen",
                __FUNCTION__);
#endif
            abort();
    }
}
```

Annexe

- Toujours la modification : ~/ns-allinone-2.34/ns-2.34/tcl/lib/**ns-packet.tcl** @ ligne 172 :

```
# LOCP patch
LOCP
```

- Modification de ~/ns-allinone-2.34/ns-2.34/tcl/lib/**ns-lib.tcl** @ ligne 633 :

```
LOCP {
    set ragent [$self create-LOCP-agent $node]
}
```

- Le même fichier ajouté dans la ligne 860 :

```
Simulator instproc create-LOCP-agent { node } {
    # Create LOCP routing agent
    set ragent [new Agent/LOCP [$node node-addr]]
    $self at 0.0 "$ragent start"
    $node set ragent_ $ragent
    return $ragent
}
```

- Modifier ~/ns-allinone-2.34/ns-2.34/tcl/lib/**ns-agent.tcl** line 202 :

```
Agent/LOCP instproc init args {
    $self next $args
}
```

```
Agent/LOCP set sport_ 0
Agent/LOCP set dport_ 0
```

- Modifier : ~/ns-allinone-2.34/ns-2.34/tcl/lib/**ns-mobilenode.tcl** line 201 :

```
# Special processing for LOCP
set LOCPonly [string first "LOCP" [$agent info class]]
if {$LOCPonly != -1 } {
    $agent if-queue [$self set ifq_(0)] ;# ifq between LL and MAC
}
```

- Nous avons fini. arrivé à ~/ ns-allinone-2.34 / ns-2.34 / répertoire et faire :

```
MAKE CLEAN
```

```
MAKE
```

BIBLIOGRAPHIQUE

[1] :Y. Yasser, « Routage pour la Gestion de l'Energie dans les Réseaux de Capteurs Sans Fil », Thèse de Doctorat, Université de Haute Alsace, juillet 2010.

[2] Akyildiz, I. F., W. Su, Y. Sankarasubramanian, E. Cayirci, "A Survey on Sensor Networks", IEEE communication Magazine, August, 102114(2002).

[3] S.J., Pister Kristofer. <http://webs.cs.berkeley.edu/tos/>. TinyOS. [En ligne] 2009.

[4] Kristofer, Pister. <http://robotics.eecs.berkeley.edu/~pister/SmartDust/>. SmartDust. [En ligne] 2001.

[5] Crossbow. MICA2 Data sheet. [En ligne] 2009.

[6] HABI Abdelmalek, Robustesse des nœuds dans un Réseau de Capteurs Sans Fil Rechargeable, diplôme MASTER, UNIVERSITE FERHAT ABBES SETIF 1, 2013

[7] : Louise lamont, louise.lamont@crc.ca centre de recherche sur la communication canada CRC. juin. 2006

[8] A. Perrig, R. Szewczyk, J. D. Tygar, V. Wen, and D. E. Culler, —Spins : Security Protocols for Sensor Networks, Wirel. Netw. 8 (2002), no. 5, 521–534.X

[9] : moodle.utc.fr/file.php/498/SupportWeb/co/Module_RCSF_1.html

[10] Michael Fitzgerald. Technology Review : Tracking a Shopper's Habits. Technology Review. [En ligne] 04 August 2008. <http://www.technologyreview.com/computing/21161/>

[11] Giuseppe Anastasi, Marco Conti, Andrea Passarella, and Luciana Pelusi, —Mobile-relay forwarding in opportunistic networks, In M. Ibnkahla, editor, Adaptation and Cross Layer Design in Wireless Networks. CRC Press, New York, USA, 2008.

[12] y_challal.esi.dz/wp-content/uploads/2016/05/PDF-RCSF

[13] E.M. Petriu, N.D. Georganas, D.C. Petriu, D. Makrakis, and V.Z. Groza. Sensor-based information appliances. IEEE Instrumentation Measurement Magazine. December 2000, Vol. 3, 4, pp. 31-35.

[14] : eprints.univ-batna2.dz/165/1/Adel%20CHOUHA

[15] fr.slideshare.net/MilagrosRomanEngjrMA/presentationwsn

[16] B. Kechar, « Problématique de la consommation de l'énergie dans les réseaux de capteurs sans fil », Séminaire LIUPPA, Université de Pau et des Pays de l'Adour, 14 Octobre 2007.

[17] MASTER/memoire_YACINE_Younes

[18] Abdallah Makhoul Réseaux de capteurs : localisation, couverture et fusion de données. DOCTEUR de l'Université de Franche-Comté UNIVERSITÉ DE FRANCHE-COMTE le Vendredi 14 novembre 2008

[19] dspace.univ-tlemcen.dz/bitstream/112/3814/1/THESE-LABRAOUI-Nabilai.pdf

[20] tel.archives-ouvertes.fr/tel-00724810/document

[21] Messaoud BELLOULA La géolocalisation dans les réseaux de capteurs sans fils Etude de cas : utilisation en agriculture Université Hadj Lak dar – Batna Faculté des Sciences Département d'Informatique.

[22] AZZEDINE BOUKERCHE, LOCALIZATION SYSTEMS FOR WIRELESS SENSOR NETWORKS UNIVERSITY OF OTTAWA HORACIO A. B. F. OLIVEIRA, UNIVERSITY OF OTTAWA, FEDERAL UNIVERSITY OF AMAZONAS EDUARDO F. NAKAMURA, FEDERAL UNIVERSITY OF MINAS GERAIS AND FUCAPI ANTONIO A. F. LOUREIRO, FEDERAL UNIVERSITY OF MINAS GERAIS

[23] dspace.univ-tlemcen.dz/bitstream/112/7157/1/La-Geo-localisation-dans-les-Reseaux-de-Capteurs-sans-Fil

[24] M. Cypriani, Géopositionnement Wi-Fi autocalibré en milieu hétérogène, Université de Franche-Comté, décembre 2012

[25] T. S. Rappaport, "Wireless communications, principles and practice", Prentice Hall, 1996.

[26] tel.archives-ouvertes.fr/tel-00724810/document

[27] N. Priyantha, A. Miu, H. Balakrishnan, and S. Teller, —The cricket compass for context-aware

mobile applications, in 6th ACM MOBICOM, July 2001, Rome, Italy.

[28] Azzedine Boukerche, —Algorithms And Protocols For Wireless Sensor Networks, Wiley, ISBN:

978-0-471-79813-2, 11-2008

[29] LABRAOUI Nabila, LA SÉCURITÉ DANS LES RÉSEAUX SANS FIL AD HOC, A L'UNIVERSITE DE TLEMCEM FACULTE DES SCIENCES ,Pour l'obtention du diplôme de DOCTORAT, en 2012

[30] Ivan Stojmenovic, —Handbook Of Sensor Networks: Algorithms And Architectures, Wiley-

Intersciencene, ISBN: 978-0-471-68472-5, 10-2005

[31] sed.inrialpes.fr/stages/stages2007/rapportEsteves.pdf

[32] Oguejiofor O.S, Aniedu A.N, Ejiofor H.C, Okolibe A.U Trilateration Based localization Algorithm for Wireless Sensor Network International Journal of Science and Modern Engineering (IJISME) ISSN: 2319-6386, Volume-1, Issue-10, September 2013

[33] Mme LABRAOUI Nabila LA SÉCURITÉ DANS LES RÉSEAUX SANS FIL AD HOC A L'UNIVERSITE DE TLEMCEM FACULTE DES SCIENCES Pour l'obtention du diplôme de DOCTORAT Spécialité : “ Informatique” Soutenue en 2012.

[34] T.He, C. Huang, B.M. Blum, J.A. Stankovic, and T. Abdelzaher, “Range-free localization schemes for large scale sensor networks,” in *Proceedings of the 9th ACM Annual International Conference on Mobile Computing and Networking (MobiCom '03)*, pp. 81–95, September 2003.

[35] Yannick Darcillon Sujet : Localisation des nœuds dans les Réseaux de capteurs sans fil Master 2 RTM 2007/2008.

