



Université
de Toulouse

THÈSE

En vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par :

Institut National des Sciences Appliquées de Toulouse (INSA de Toulouse)

Présentée et soutenue par :

DALCE Réjane

le 26 juin 2013

Titre :

Méthodes de localisation par le signal de communication dans les réseaux de capteurs sans fil en intérieur

École doctorale et discipline ou spécialité :

SYSTEMES: Informatique et Systèmes Embarqués

Unité de recherche :

Institut de Recherche en Informatique de Toulouse (IRIT)

Directeur(s) de Thèse :

Pr. Thierry VAL

Dr. Adrien VAN DEN BOSSCHE

Jury :

Pr. Thierry DIVOUX, Université de Lorraine, rapporteur

Pr. François SPIES, Université de Franche-Comté, rapporteur

Dr. Laurence REDON, Université de Toulouse Le Mirail, examinateur

Dr. Alexandre GUITTON, Université Blaise Pascal, examinateur

Pr. Eric CAMPO, Université de Toulouse Le Mirail, invité

Remerciements

Il y a un peu plus de dix ans, je répondais par la négative à la question : « Tu ne veux pas être docteur ?! ». Je n'envisageais guère de passer plus de 5 ans à l'université. Etrangement, je finalise en 2013 ma thèse de doctorat : aventure passionnante, avec ses exaltations et ses doutes, ses montagnes de documents à lire et ses abîmes de code.

Si je suis ici aujourd'hui, c'est grâce à une succession de rencontres. Je veux donc exprimer ma reconnaissance à ces personnes et entités.

Je tiens à remercier MM. Divoux et Spies pour l'intérêt qu'ils ont porté à mes travaux en acceptant de les rapporter. Merci aux membres du jury pour m'avoir fait le plaisir d'examiner mon travail.

Puis mes directeurs de thèse, MM Val et van den Bossche, qui m'ont poussée à me poser des questions et m'ont donné les moyens techniques et scientifiques de tester mes idées. Ce fut une joie de recevoir votre encadrement.

Je remercie l'Ecole Doctorale Systèmes, non seulement d'avoir retenu ma candidature à cette thèse, mais également de m'avoir encadrée et soutenue durant ces quatre années.

Je ne saurais oublier l'Institut National des Sciences Appliquées de Toulouse ainsi que l'IUT de Blagnac qui m'ont offert la possibilité d'intervenir en tant qu'enseignant dans le cadre de mon monitorat.

A tout le personnel de l'IUT, chefs de département, personnels enseignants, administratifs et techniques, que j'ai eu la chance côtoyer durant ces quatre années. Vous avez constitué pour moi un écosystème favorable dans lequel j'ai pu expérimenter le rôle d'enseignant-chercheur.

A mes parents qui m'ont toujours encouragée à aller au bout de mes idées, aussi déroutantes soient-elles. Merci de m'avoir inculqué le goût de la lecture.

A Linqing et Juan, j'espère avoir été pour vous une source de motivation car vous avez joué ce rôle pour moi. A ceux qui commencent et à ceux qui poursuivent- Manzoor, Chiraz, Asma, Sabri- *FIGHTING* !!

A mes amis, Elizabeth, Tania, Fabian, Alexandra, Adrian, Sacha et Satoko, dont la présence m'a obligée à garder un équilibre dans mon existence.

Et à cette personne qui ne m'a jamais menée là où Sa grâce ne m'accompagnerait pas.

Merci !

RD

Table des matières

INTRODUCTION.....	9
1. Etat de l’art de la localisation dans les réseaux de capteurs sans fil	16
1.1. Connaissances fondamentales	16
1.1.1. Structure réseau : mobiles, ancres, <i>mesh</i> /arbre, repère global	16
1.1.2. Informations disponibles pour la localisation	20
1.1.2.1. Puissance du signal radio	21
1.1.2.2. Temps de vol	23
1.1.2.3. Angle d’incidence	26
1.1.2.4. Phase du signal	27
1.1.2.5. Nombre de sauts	27
1.1.2.6. Autres paramètres.....	28
1.2. Protocoles de mesure.....	28
1.2.1. Protocoles de collecte dans les méthodes <i>range-based</i>	29
1.2.1.1. Indicateur de puissance de signal reçu	29
1.2.1.2. Temps de vol du signal.....	30
1.2.2. Protocoles de collecte pour les méthodes <i>range-free</i>	34
1.3. Algorithmes de localisation.....	35
1.3.1. Algorithmes généraux	35
1.3.1.1. Trilatération	35
1.3.1.2. Triangulation	36
1.3.1.3. Reconnaissance d’empreinte	36
1.3.1.4. Multilatération	37
1.3.1.5. Filtre de Kalman.....	38
1.3.1.6. Filtre particulière [7]	39
1.3.1.7. Estimateur des moindres carrés.....	40
1.3.1.8. Inférence bayésienne	40
1.3.1.9. Sélection des plus proches voisins	40
1.3.1.10. Recherche séquentielle	41
1.3.2. Adaptations récentes des algorithmes à la localisation par WSN	41
1.3.2.1. Méthodes <i>range-based</i>	41
1.3.2.1.1. Temps de vol	41
1.3.2.1.2. RSSI et puissance reçue	42
1.3.2.1.3. TDOA et phase du signal	44
1.3.2.2. Méthodes Range-Free	46
1.3.2.2.1. Famille DV-Hop.....	46
1.3.2.2.2. APIT	49
1.3.2.2.3. <i>Centroid</i>	50
1.3.2.3. Conclusion.....	51
1.4. Pistes de recherche	51
1.4.1. Exactitude de la localisation.....	51
1.4.2. Algorithmes décentralisés	53
1.4.3. Algorithmes efficaces	54
1.5. Conclusion.....	54
2. Analyses exploratoires pour une auto-localisation <i>range-based</i> en intérieur.....	56
2.1. Sélection d’un signal : RSSI ou temps de vol	57
2.1.1. Plateformes utilisées pour l’exploration.....	57
2.1.2. RSSI	60
2.1.3. Temps de vol	65

2.1.4.	Conclusion.....	72
2.2.	Traitement des données.....	72
2.3.	Caractérisation énergétique.....	75
2.4.	Sources complémentaires.....	80
2.4.1.	Les ultrasons.....	80
2.4.2.	Les signaux infrarouges.....	81
2.4.3.	Les instruments de navigation inertielle.....	81
2.5.	Conclusion.....	83
3.	PROPOSITION D'UN SYSTEME DE LOCALISATION ORIGINAL ET EFFICIENT.....	86
3.1.	Proposition de protocole.....	87
3.1.1.	Négociation du <i>star interval</i>	87
3.1.2.	SSDS-TWR : échanges et formules.....	95
3.1.3.	PSDS-TWR : échanges et formules.....	98
3.1.4.	Comparaison de la charge réseau des deux protocoles.....	101
3.1.5.	Comparaison de la consommation énergétique induite par les deux protocoles	103
3.1.5.1.	Caractérisation temporelle de SSDS-TWR.....	103
3.1.5.2.	Caractérisation temporelle de PSDS-TWR.....	105
3.1.5.3.	Analyse de la consommation.....	107
3.1.6.	Conclusion.....	111
3.2.	Proposition d'un algorithme de localisation.....	111
3.2.1.	<i>iRingLA</i>	112
3.2.1.1.	Déroulement.....	112
3.2.1.1.1.	Création des anneaux.....	112
3.2.1.1.2.	Création de la matrice de départ.....	113
3.2.1.1.3.	Recherche des points candidats.....	117
3.2.1.2.	Durée d'exécution.....	117
3.2.1.3.	Complexité et optimisations.....	118
3.2.1.4.	Ordonnancement des anneaux.....	120
3.3.	Transposition sur d'autres technologies.....	126
4.	EVALUATION DU PROTOCOLE ET DE L'ALGORITHME PROPOSES.....	130
4.1.	Prototype.....	130
4.1.1.	Présentation de la plateforme matérielle.....	130
4.1.2.	Caractéristiques du <i>ranging</i> et corrections appliquées.....	131
4.1.3.	Résultats de localisation.....	134
4.1.3.1.	Précision et observations diverses.....	135
4.1.3.2.	Durée d'exécution du protocole.....	139
4.1.3.3.	Durée d'exécution de l'algorithme.....	139
4.1.4.	Conclusion.....	142
4.2.	Simulateur <i>DokoSim</i>	142
4.2.1.	Motivations.....	143
4.2.2.	Choix d'un simulateur.....	146
4.2.2.1.	NS-2 et NS-3.....	146
4.2.2.2.	OMNeT++.....	146
4.2.2.3.	WSNet+WSim.....	147
4.2.3.	Description technique.....	147
4.2.3.1.	Structure du simulateur.....	148
4.2.3.2.	Architecture de classes.....	148
4.2.3.2.1.1.	Classe <i>Clock</i>	149
4.2.3.2.1.2.	Classes <i>NodeAgenda</i> et <i>AgendaItems</i>	150

4.2.3.2.1.3.	Classe <i>WirelessMedium</i>	151
4.2.3.2.1.4.	Classe <i>WirelessNode</i>	152
4.2.3.2.1.5.	Classe <i>LocatorEngine</i>	154
4.2.4.	Choix d'implémentation.....	154
4.2.4.1.	Similitudes.....	155
4.2.4.2.	Différences	159
4.2.5.	Test de SSDS-TWR	161
4.2.5.1.	Paramétrage du <i>multipathFactor</i>	161
4.2.5.2.	Paramétrage du taux de pertes de trames	164
4.2.5.3.	Validité du simulateur	165
4.2.6.	Test de PSDS-TWR	165
4.2.6.1.	Simulation dans un environnement sans pertes.....	166
4.2.6.2.	PSDS-TWR face aux interférences	170
4.2.6.3.	Gain temporel de PSDS-TWR	171
4.2.7.	Conclusion sur PSDS-TWR	173
4.3.	Bilan sur le simulateur <i>DokoSim</i>	173
	CONCLUSION ET PERSPECTIVES	175
	GLOSSAIRE	181
	ANNEXE.....	183
	LISTE DES FIGURES.....	187
	LISTE DES TABLEAUX.....	191
	PUBLICATIONS PERSONNELLES	193
	BIBLIOGRAPHIE	195
	SUJET DE THESE	201

INTRODUCTION

La capacité à se déplacer est un avantage certain pour les organismes vivants : pouvoir changer d'environnement peut être la condition sine qua non de la survie d'une espèce. Mais avec ce pouvoir viennent aussi des obligations, comme la nécessité de contrôler ce déplacement de manière à le rendre véritablement utile. Ce contrôle s'exerce autant sur la manière de se déplacer que sur le choix et la reconnaissance du chemin emprunté. L'acteur cesse d'errer sitôt qu'il peut déterminer où il se trouve et, par voie de conséquence, où il va. Le premier problème constitue notre point focal que nous désignerons par la suite par le terme de localisation, qualifié parfois par la communauté scientifique par géo-localisation ou (géo)-positionnement. Cette thèse s'intéressera à une ou plusieurs méthodes pouvant être exploitées pour identifier l'endroit où se trouve l'acteur. Plus simplement, nous l'aiderons à répondre à la question fondamentale : « Mais où suis-je ? ».

Cette localisation utilisera des moyens divers, dépendant des possibilités de l'acteur. Elle interviendra dans différents types de situations que nous allons récapituler dans la suite de cette introduction.

Dans le passé, le chasseur-cueilleur typique avait besoin de savoir où il se trouvait simplement pour pouvoir regagner sa caverne. Aujourd'hui, la localisation sert à bien plus que ceci. Ses applications recouvrent des domaines très divers tels que :

- **La santé** Dans le contexte actuel de vieillissement de la population mondiale [6][51], les établissements spécialisés (maison de retraite, EHPAD¹...) doivent accueillir un nombre important de résidents ce qui impose une taille des structures adaptée. Les patients sont aujourd'hui fréquemment équipés d'unités permettant de contrôler à distance leur état de santé par la capture de certains paramètres vitaux [72][90]. Pour le bien-être des patients, ces unités communiquent généralement par une technologie sans fil. Cela implique que lors du déclenchement d'une alarme, le patient puisse se trouver ailleurs que dans son lit. Le personnel doit donc faire face à un défi de taille lors des interventions : pour arriver à temps, il lui faut déterminer où se trouve le patient.

De la même manière, l'augmentation des cas de dégénérescence cérébrale conduisant à des comportements à risque chez les personnes âgées, motive le

¹ Etablissement d'Hébergement pour Personnes Agées Dépendantes

développement des méthodes et moyens de localisation car il faut non seulement, si on ne peut les prévenir, reconnaître les fugues et retrouver très rapidement le patient [41].

- **La sécurité** Les services de localisation sont également souhaitables pour assurer la sécurité des biens et des personnes. Certaines activités professionnelles peuvent être dangereuses pour les personnes impliquées : par exemple, les employés chargés d'effectuer des rondes de contrôle dans les centrales nucléaires sont exposés aux radiations. Certes, il est nécessaire de pouvoir leur indiquer que le seuil de sécurité est atteint (dosimétrie) : des dispositifs existent permettant de mesurer cette exposition. Dans le cas où un accident risque de survenir, les secours doivent être en mesure de minimiser le délai d'intervention : ceci n'est possible que si la localisation de la victime est disponible.

De même, les individus susceptibles de fuguer ou de se perdre, enfants ou animaux, gagneraient à être équipés de dispositifs de localisation (tout en respectant l'éthique et la liberté individuelle).

- **L'environnement** Une perception traditionnelle de l'agriculture n'a que faire des nouvelles technologies de l'information et de la communication. Mais lorsque l'exploitation efficace des sols devient une priorité, différents domaines combinent leurs apports pour proposer des solutions permettant une production optimale tout en maintenant la qualité de l'environnement. Les travaux de l'université du Nebraska sur l'agriculture de précision (*Precision Agriculture*) en sont un exemple : dans [4], les avantages du contrôle automatique des machines utilisées dans l'agriculture sont présentés. L'ajout d'assistants au guidage permet entre autres de réaliser des actions même en cas de visibilité insuffisante. Ces équipements utilisent les données du GPS mais aussi un algorithme local de localisation afin d'améliorer le suivi de la machine et donc la qualité des passages sur le terrain.

Ce type d'application envisage l'ajout d'un récepteur GPS au niveau du tracteur de l'exploitant agricole. Les enjeux ainsi que le nombre restreint de machines rendent cet investissement acceptable. Néanmoins, dans d'autres situations, ce dispositif peut être inexploitable et ce pour diverses raisons. Tout d'abord, sans changer de domaine, le GPS peut ne pas être disponible sur une certaine zone du globe en

raison des relations entre les autorités locales et les administrateurs du système : certes, depuis 2000, la disponibilité sélective (*Selective Availability*) a été officiellement désactivée mais la possibilité existe encore [19]. De la même manière, l'emplacement *dans un environnement intérieur* des unités rend quasi impossible la réception du signal donc l'utilisation de ces données. Finalement, si l'installation sur une machine dotée d'un moteur permet de négliger la consommation énergétique du récepteur GPS, dans un contexte où cette ressource est très contrainte, le coût de la solution peut devenir rédhibitoire.

Il est par conséquent important de développer des solutions permettant une localisation autonome, non tributaire d'un opérateur spécialisé et adaptée aux conditions de l'expérience.

- **Le marketing** Actuellement se déploient en France [43] des services de localisation *d'intérieur* par *smartphone* Android exploitant la position d'antennes GSM² et de points d'accès publics ou *hotspots*³ Wifi. Si ce support est, dans d'autres pays (USA, Japon) [43], disponible dans les aéroports, il est permis d'envisager de coupler cette localisation avec des offres promotionnelles dans les grands magasins. Ceci permettrait d'orienter le client en fonction de ses besoins vers des produits spécifiques.
- **Le confort et les services à la personne** Bien qu'abordable en dernier, ce secteur fourmille d'applications : dans la vie quotidienne, certaines questions reviennent très souvent. Parmi elles, on retrouve :

« Où sont mes clés ? Mon téléphone portable ? La télécommande ...Je crois que je suis perdu : où suis-je ? ...Mais où ai-je garé la voiture ?!! »

Bien entendu, certains de ces problèmes sont résolus par des technologies existantes [19]. La technologie de RFID⁴ [37] est également un outil souvent rencontré dans ce domaine. Le RFID se base sur l'utilisation de circuits miniaturisés ou *tags*, placés sur des unités à contrôler, et de lecteurs. Les *tags* peuvent être passifs ou actifs : dans le premier cas, l'énergie reçue par le biais du message du lecteur est utilisée pour alimenter le *tag* et lui permettre de répondre. Dans le second cas, le

² *Global System for Mobile communications*

³ Points d'accès Wifi publics, souvent utilisés pour la navigation internet

⁴ *Radio Frequency Identification*

tag dispose d'une source d'énergie et peut donc communiquer sur de plus longues distances.

Cette technologie évoque les systèmes de *keyfinders* tels que [33]. Basés sur la technologie décrite dans le brevet 6774787 [15], ces dispositifs émettent une sonnerie et éventuellement un signal lumineux à la réception d'un message radio désigné comme signal de recherche. Bien que ces produits constituent une réponse à des problématiques données, il n'en demeure pas moins que dans certains cas, une forme d'intelligence est désirable au niveau du *tag*. Ce dernier se développera donc pour devenir un nœud de réseau de capteur sans fil.

La technologie *Near Field Communication* (NFC) est très similaire au RFID avec lequel elle peut s'interfacer. La NFC est principalement envisagée dans le cadre de paiements par téléphone mobile. Elle requiert une distance de 4cm ou moins entre les deux unités ce qui nous porte à conclure que moyennant une très forte densité de nœuds de référence, une localisation très précise est possible à partir de cette technologie. La question se transformerait alors en une étude du coût de pareille installation.

L'environnement actuel évolue progressivement vers un monde connecté. Les moyens de communications et les zones couvertes s'étendent de plus en plus. Les informations que peuvent générer ces outils sont parfois utiles pour la localisation, comme nous l'examinerons au chapitre 2. Il est donc avantageux d'exploiter leur présence plutôt que de déployer un *n-ième* système dédié uniquement à la localisation.

Cette exploitation peut être menée suivant les lignes suivantes [24] :

- **Utiliser les informations de connectivité et une connaissance préalable du réseau** Les méthodes relevant de ce type de stratégie sont dites *range-free*⁵ ou indépendantes de la mesure de la distance. Par exemple, si le mobile peut communiquer avec une entité dont on sait qu'elle se trouve à Londres, alors, le mobile est aussi très probablement à Londres selon la portée de la technologie utilisée. D'autres critères, en plus de la connectivité, nous analyserons au chapitre 1.1.2.5.
- **Obtenir du signal radio des données pertinentes** Le signal radio est employé pour mesurer la distance ou un angle. Ces mesures réelles sont utilisées pour

⁵ Solution basée sur des hypothèses concernant certaines caractéristiques et propriétés du système

déterminer la position, d'où l'appellation *range-based*⁶ [25]. Ces techniques mènent généralement à des résultats plus précis que dans le cadre *range-free*. Dans la suite, nous aurons à investiguer les différentes informations pouvant être exploitées par ces méthodes.

En ce qui concerne l'architecture matérielle devant supporter la solution, nous retiendrons les réseaux de capteurs sans fil au détriment des RFID et technologies assimilables. En effet, si l'avantage en termes de coût se situe du côté des RFID, la nécessité de disposer d'un mobile capable d'auto-localisation, donc de calculer sa position sans avoir recours à un serveur dédié, renforce l'attractivité des réseaux de capteurs. Par conséquent, dans la suite de ce document, nous allons nous centrer sur la localisation à l'aide d'un réseau de capteurs sans fil. Ce dernier sera constitué de deux types de nœuds, les ancres, formant l'infrastructure, et les mobiles. Les ancres connaissent leur position par configuration tandis que les mobiles doivent estimer la leur dynamiquement grâce aux mesures réalisées en collaboration avec les ancres. Si la localisation reste l'objectif premier de cette étude, des problématiques associées aux WSN⁷ ne seront pas ignorées : ainsi, les relations entre la solution, la consommation énergétique, l'utilisation de la bande passante comme des ressources propres des nœuds (mémoire...) seront prises en compte dans la formulation de notre proposition (architecture centré mobile ou infrastructure [21]).

Après avoir brossé le paysage tant protocolaire qu'algorithmique du problème au chapitre 1, nous présenterons les premiers résultats nous ayant permis d'orienter notre recherche. Le protocole ainsi que l'algorithme découlant de ce processus seront introduits au chapitre 3. Leurs résultats, collectés aussi bien dans le monde réel que dans un simulateur, seront présentés au chapitre 5. Finalement, les possibilités offertes par cette étude en termes d'approfondissement seront décrites.

⁶ Solution basée sur la collecte en temps réel de mesures de certaines caractéristiques du système

⁷ *Wireless Sensor Networks*

ÉTAT DE L'ART DE LA LOCALISATION DANS LES RESEAUX DE CAPTEURS SANS FIL

Nous commencerons notre étude par un chapitre examinant l'existant à plusieurs niveaux. Après une brève présentation du contexte sans fil, nous introduisons un échantillon représentatif des différents protocoles de mesure et algorithmes de calcul de position disponibles. Finalement, nous identifions les plus importantes pistes de recherche.

1.1.	Connaissances fondamentales	16
1.1.1.	Structure réseau : mobiles, ancres, <i>mesh</i> /arbre, repère global	16
1.1.2.	Informations disponibles pour la localisation	20
1.2.	Protocoles de mesure	28
1.2.1.	Protocoles de collecte dans les méthodes <i>range-based</i>	29
1.2.2.	Protocoles de collecte pour les méthodes <i>range-free</i>	34
1.3.	Algorithmes de localisation.....	35
1.3.1.	Algorithmes généraux	35
1.3.2.	Adaptations récentes des algorithmes à la localisation	41
1.4.	Pistes de recherche	51
1.4.1.	Exactitude de la localisation.....	51
1.4.2.	Algorithmes décentralisés	53
1.4.3.	Algorithmes efficaces	54
1.5.	Conclusion.....	54

Tout système de localisation peut être considéré comme une tour à trois niveaux : le matériel et ses propriétés, l'utilisation de ce matériel pour la collecte de mesures pertinentes, et enfin l'exploitation de ces mesures par un algorithme de localisation. Dans les sous-chapitres suivants, ces aspects seront traités. Nous commencerons donc par établir les bases de notre travail en présentant les connaissances fondamentales concernant le réseau de communications sous-jacent ainsi que les informations typiquement exploitées pour la localisation. Nous analyserons ensuite les protocoles exploités pour la récupération de ces données avant d'examiner les divers algorithmes sur lesquels se fondent la plupart des solutions rencontrées dans la littérature.

1.1. Connaissances fondamentales

Avant d'aborder les concepts protocolaires et algorithmiques directement liés à la localisation, nous avons jugé bon de rappeler les bases sur lesquelles ces solutions sont construites. Les paragraphes suivants s'intéresseront par conséquent à la présentation des plateformes matérielles typiquement ciblées par notre travail. L'environnement de communication sera également introduit.

1.1.1. Structure réseau : mobiles, ancres, *mesh*/arbre, repère global

Notre solution est destinée à une implémentation dans un réseau de capteurs sans fil, typique, c'est-à-dire communicant à travers des liens radio à courte portée. Ce type de réseau est constitué de deux types de nœuds : les coordinateurs, capables de routage, et les nœuds terminaux ou *end-devices*. Les fonctionnalités des routeurs étant plus étendues, un choix stratégique peut consister à les doper en termes de mémoire, de ressources énergétiques etc... Il n'en demeure pas moins qu'un nœud capteur n'aura pas les capacités d'un ordinateur portable et ceci pour diverses raisons parmi lesquelles le coût et les besoins. En effet, un WSN a une vocation d'omniprésence ; les membres de ce réseau doivent pouvoir s'intégrer dans l'environnement et produire une analyse détaillée des phénomènes qu'ils ont pour mission de suivre. Un nombre de modules proportionnel à la taille de la zone à surveiller est donc nécessaire. Plus le coût de chaque nœud sera élevé, plus il sera difficile de mettre en place le réseau idéal. D'un autre côté, les applications envisagées ne sont ni la diffusion de données multimédia ni l'échange de fichiers de taille importante : le but ici est de mesurer certaines

caractéristiques de l'environnement, de détecter l'occurrence d'événements prédéfinis et de notifier un client. Un haut débit n'est donc pas justifié dans notre contexte. Finalement, en vue de pouvoir répartir les modules de manière optimale, il est indispensable de se libérer des contraintes de l'environnement en matière d'accès à la ressource énergétique : les modules doivent par conséquent fonctionner même en étant détachés du réseau électrique. Une batterie ou pile devra donc être prévue dont la durée de vie sera maximisée par l'implémentation de protocoles judicieux. Cette autonomie permettra la mobilité, créant par là même le besoin d'un service de localisation.

A ces exigences, une réponse adaptée est fournie par les constructeurs à travers le format typique des modules. La version la plus simple d'un nœud est indiquée en figure 1(a) : les données provenant des capteurs sont collectées par un microcontrôleur qui les met en forme au sein d'un message à transmettre par l'émetteur / récepteur. Ce dernier dispose d'un accès aux interruptions du microcontrôleur, ce qui lui permet de signaler les événements se produisant au niveau de la PHY⁸. Les échanges de messages entrants et sortants prennent place sur le bus SPI⁹, à une cadence proportionnelle à la vitesse du microcontrôleur. L'ensemble est alimenté par une source d'énergie qui peut être une batterie ou une pile. Cette version correspond à une implémentation industrielle pour laquelle des modifications du code ne sont pas prévues et les seules interactions autorisées passent principalement par le lien radio.

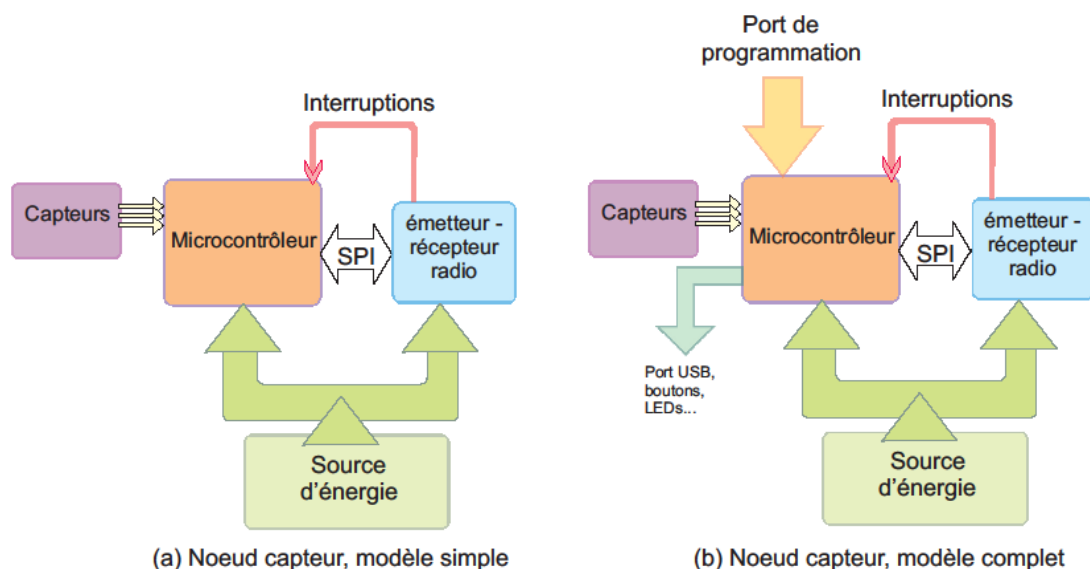


Figure 1 : Structure d'un nœud capteur

⁸ Couche physique du modèle OSI

⁹ *Serial Peripheral Interface*

Le microcontrôleur classique (8 bit) intègre une faible quantité de mémoire (64ko typiquement) et sa capacité de calcul est limitée. Concrètement, le nombre de variables requises par le logiciel peut empêcher le déploiement d'une solution trop complexe.

La figure 1(b) illustre la structure des nœuds utilisés dans le cadre de nos travaux : principalement, des interfaces supplémentaires sont disponibles telles que celle permettant de télécharger un nouveau code exécutable binaire sur la plateforme, ou encore le port série grâce auquel des données peuvent facilement être obtenues durant l'exécution du protocole.

Bien que nous ayons la possibilité de reprogrammer entièrement nos nœuds, nous avons néanmoins opté pour un développement basé sur le standard IEEE pour les réseaux de capteurs sans fil, la norme IEEE 802.15.4. Une autre norme de communication pour les réseaux personnels étant disponible, IEEE 802.15.1 [53], nous avons utilisé le critère de la puissance pour les départager et sélectionner le protocole le moins énergivore. Le tableau 1 présente un comparatif de ces standards. IEEE 802.11 [52] est ici mentionné comme référence. Si par l'adoption des couches physiques appropriées, des débits relativement élevés peuvent être supportés par ces trois normes, le standard 802.15.4 est nettement plus avantageux d'un point de vue énergétique.

Tableau 1 : Comparatif des standards IEEE 802.11, 802.15.1 et 802.15.4

	802.11	802.15.1	802.15.4
Réseau	WLAN ¹⁰	WPAN ¹¹	WPAN
Débit	1 – 450Mb/s	1 - 3Mb/s	250kb/s, 1Mb/s 110kb/s, 850kb/s, 6,81Mb/s, 27.24Mb/s
Puissance d'émission (mW)	750-1000	100	30
Bande de fréquence	2.4GHz	2.4GHz	2.4GHz 249.6 - 749.6MHz 3.1 - 4.8 GHz 6 - 10.6 GHz

En plus d'assurer l'interopérabilité avec l'ensemble des produits compatibles, cette norme ouverte a l'avantage de tenir compte de l'économie d'énergie au niveau protocolaire et de fournir, dans la version 802.15.4a, un support pour la mesure de distance à travers le signal

¹⁰ *Wireless Local Area Network*

¹¹ *Wireless Personal Area Network*

radio, plus précisément le temps de propagation dudit signal. Ainsi, le composant radio ou *transceiver*¹² utilisé sera compatible 802.15.4 et l'implémentation de la couche MAC¹³ respectera les indications du standard. Des mécanismes de conservation d'énergie tels que les différents types de sommeil des nœuds pourront être exploités par notre protocole. La figure 2 illustre l'agencement des périodes actives et des phases de sommeil tel que proposé par [54].

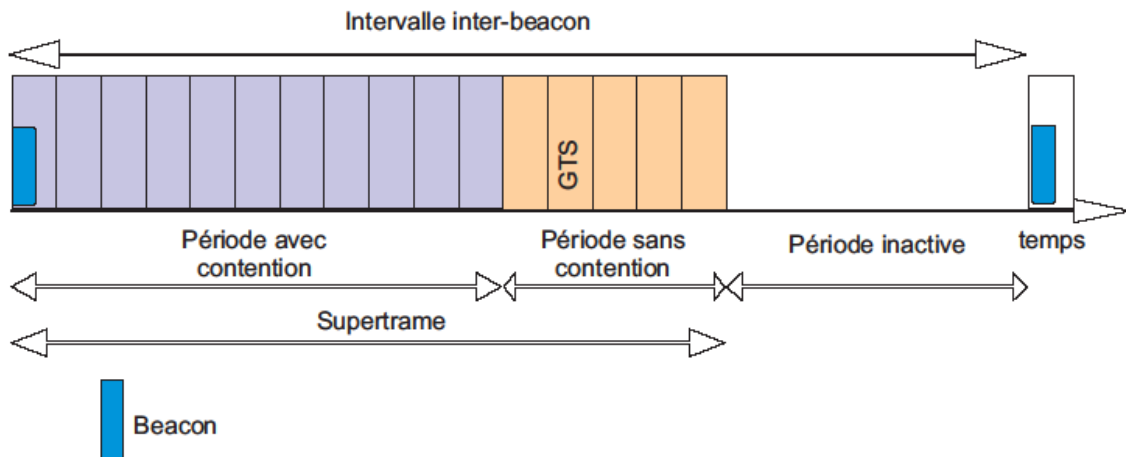


Figure 2 : Structure temporelle dans un réseau 802.15.4 utilisant des *beacons*

L'hypothèse fondamentale de nœuds de référence communiquant des informations relatives au réseau à travers des trames spéciales appelées balises ou *beacons* implique l'utilisation du modèle de la figure 2. L'écoulement du temps est rythmé par l'émission de ces *beacons*, l'intervalle les séparant étant appelé le BI¹⁴. Ce BI se divise en deux grandes parties : la première, la supertrame, permet les communications au sein du réseau tandis que la seconde, la période inactive, est dédiée à l'économie d'énergie. Durant cette phase, les nœuds entrent dans un mode de consommation réduite.

La supertrame est formée de 16 slots, répartis entre la CAP¹⁵ et la CFP¹⁶. La CAP, comme son nom l'indique, est exploitée pour les communications avec contention, où les nœuds sont en compétition pour l'accès au médium. Un protocole spécifique est alors nécessaire. Le choix de la norme a été d'utiliser le principe MAC nommé CSMA/CA¹⁷ durant cette période.

La CFP est également formée de slots qui ont la particularité d'être réservés à deux partenaires de communication et de n'autoriser les échanges que dans un sens. En d'autres

¹² Emetteur-récepteur, implémentation d'une couche physique spécifique

¹³ *Medium Access Control* ou contrôle de l'accès au médium

¹⁴ *Beacon Interval*

¹⁵ *Contention Access Period*

¹⁶ *Contention Free Period*

¹⁷ *Carrier Sense Multiple Access with Collision Avoidance*

termes, à la réservation du slot, nommé GTS¹⁸, la direction est spécifiée. Chacun des partenaires se voit assigner un rôle, soit émetteur, soit récepteur de données. La mise en place de ces conditions permet un accès direct au médium, sans crainte de provoquer des collisions. Dans [14], des aménagements sont proposés en vue d'optimiser l'utilisation des GTS.

IEEE 802.15.4 définit deux familles de topologies : l'étoile et le point à point. Des travaux subséquents ont proposé la création d'arbres d'étoiles [1][69] bénéficiant des propriétés d'auto-organisation du point à point et supportant la qualité de service (QoS¹⁹). Les travaux de [71] visent à concilier une structure maillée avec l'émission sans collision de *beacons* dans un réseau économe en énergie.

Ce standard inclut également l'adressage des nœuds. Une adresse longue sur 4 octets ainsi qu'une adresse courte sur 2 octets peuvent être utilisés pour désigner un élément du réseau. Selon la phase de l'existence du réseau, l'une ou l'autre sera employée dans les trames.

Dans notre travail nous avons choisi de considérer un réseau sans nœud central et associé à une structure maillée. Ceci ne signifie pas que le réseau soit isolé des autres réseaux : il est plus que probable qu'une passerelle existera entre le WSN et les réseaux avoisinants, au moins vers Internet. Mais le fonctionnement du WSN ne dépendra pas d'un nœud central responsable de toutes les décisions.

Dans la suite, le réseau considéré pour la localisation sera constitué de deux types de nœuds : les ancres ou balises, connaissant leur position par configuration ou calcul préalable, et les mobiles, devant récupérer des informations concernant le réseau pour déterminer leur position. Ces positions sont exprimées dans un repère global R et peuvent, par des transformations mathématiques, être ramenées à des coordonnées exploitables par des systèmes tels que les systèmes de navigation GPS. La détermination de ces matrices de passage reste en dehors des objectifs de cette thèse et ne sera donc pas abordée.

1.1.2. Informations disponibles pour la localisation

Cette section décrit les différentes informations pouvant être exploitées par les systèmes de localisation. Certaines sont des mesures réalisées par le système (puissance du signal, temps de vol...) tandis que d'autres sont du domaine des hypothèses relatives au réseau (connectivité...).

¹⁸ *Guaranteed Time Slot*

¹⁹ *Quality of Service*

1.1.2.1. Puissance du signal radio

Lorsqu'un équipement radio émet un signal, au fur et à mesure que celui-ci s'éloigne de la source, son énergie décroît. Ce phénomène s'explique par les interactions entre l'onde et le milieu de propagation. Divers modèles ont été proposés pour décrire cette relation.

1.1.2.1.1. 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 [21] 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.

1.1.2.1.2. Propagation avec réflexion sur le sol ou *Two Ray Ground reflection model*

Si l'équation de Friis est efficace dans le cadre des liaisons satellitaires, d'autres modèles sont nécessaires lorsqu'on s'intéresse à des communications proches du sol. Le *Two Ray Ground Reflection model* [81] considère que le signal suit deux chemins principaux (figure 3) pour arriver au récepteur. Le premier est direct et le second réfléchi par le sol.

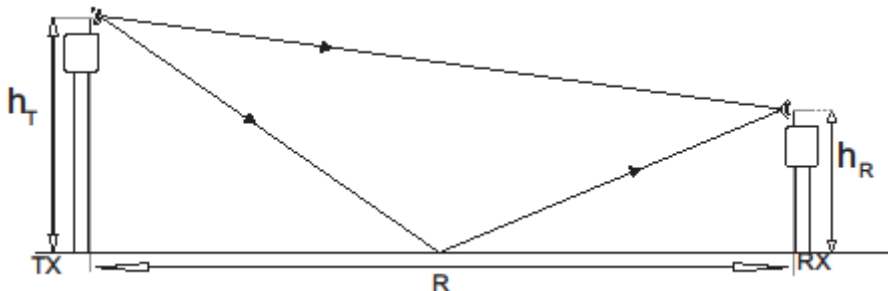


Figure 3 : Propagation avec une réflexion

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).

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

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

1.1.2.1.3. Modèle de *Shadowing*

Ce modèle [81] 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.

1.1.2.1.4. Niveau de puissance radio reçue ou *RSSI*²⁰

Certains équipements radio sont capables de fournir aux couches supérieures une estimation de la puissance reçue en dBm. Il est donc possible, à partir de cette valeur, de déduire la distance parcourue par le signal.

²⁰ *Received Signal Strength Indicator*

Sur les nœuds de WSN, l'indicateur de puissance de signal reçu ou RSSI est préféré à cette donnée. Ce paramètre se présente généralement sous la forme d'un octet et comme il dépend de la puissance, varie de la même manière en fonction de la distance. Néanmoins, les solutions développées à partir de la puissance reçue seront considérées comme plus faciles à transférer entre des plateformes diverses que celles à base de RSSI car la relation entre le RSSI et la puissance n'est pas normalisée. Ceci signifie que la mise en place de cette mesure est laissée à la discrétion du fabricant et que les modèles développés pour une plateforme ne seront pas utilisables sur une autre.

Dans la littérature, une multitude de solutions sont proposées pour estimer la position d'un nœud communicant à partir du RSSI [23] [17] [32] [47] [11] [31] [85] [12]. Cette stratégie a l'avantage de ne pas requérir un trafic dédié car toutes les trames échangées peuvent fournir une estimation du paramètre. Dans les cas où le nœud mobile est chargé de collecter les valeurs de RSSI à partir des trames périodiquement émises par les ancres, la récupération par plusieurs mobiles simultanément est possible et ne génère aucune collision.

D'un autre côté, cette information est très sensible à l'environnement : le corps humain, du fait de sa grande teneur en eau, atténue les signaux HF qui le traversent [66]. Ceci aura un impact en déploiement réel sur les solutions proposées pour des technologies exploitant la bande ISM 2.4GHz. Les positions relatives des antennes ont également une influence sur les niveaux de puissance détectés. Les informations brutes ne sont par conséquent pas directement exploitables : une étape de profilage, c'est-à-dire de constitution d'un cadre de référence à partir de données collectées dans l'environnement visé, doit donc être ajoutée et constitue un coût supplémentaire.

1.1.2.2. Temps de vol

Les informations temporelles ont la réputation d'être plus fiables [68] que le niveau de puissance reçue : la vitesse de propagation du signal dans l'air étant connue, il suffit d'appliquer la formule (équation 6) pour obtenir la distance.

$$\text{distance} = \text{vitesse} \times \text{temps} \quad (6)$$

Dans un réseau, l'échange de messages entre les équipements est un moyen d'estimer ce temps de vol. Dans le cas d'un réseau synchronisé, un seul message suffit. La figure 4 illustre cette situation : un message de B (comme Balise) vers M (comme Mobile) voit ses instants d'émission et de réception, t_a et t_b , capturés par les deux nœuds. Une fois les informations

réunies au niveau de B, par une autre transmission de M vers B, le temps de vol ou TOF²¹ peut être estimé (équation 7). D'autres protocoles permettant la mesure du temps de vol seront présentés dans la suite de ce document.

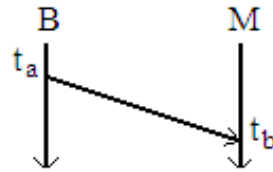


Figure 4 : Temps de vol

$$t_{vol} = t_b - t_a \quad (7)$$

A la base de la récupération des instants de réception des trames se trouvent les techniques de détection d'arrivée de message. Différentes méthodes sont étudiées dans [10] pour une couche physique *Ultra-wide Band* (UWB). La technique *Energy Detection* (ED) est basée sur la détection d'un message par son niveau d'énergie : le récepteur recueille des échantillons de RSSI du médium sur un intervalle d'intégration T. L'avantage principal de cette méthode est de ne requérir qu'un nombre limité d'échantillons pour parvenir à une décision. Par contre, lorsque le signal est difficile à discerner du bruit, les performances de ce type de récepteur sont fortement dégradées.

La méthode *Stored Reference* (SR), contrairement à l'ED, exige une collecte des échantillons très rapide. En effet, l'objectif est de corrélérer le signal reçu à une copie locale, le pic correspondant à l'arrivée d'une trame. Pour que le pic soit clairement identifiable, il est nécessaire d'aligner les deux copies à corrélérer d'où la fréquence d'échantillonnage élevée.

D'autres architectures ont également été analysées, notamment dans l'objectif de réduire les contraintes de coût engendrées par le type SR.

Dans les deux cas, un algorithme de détection vient identifier l'instant d'arrivée du message : ceci peut se faire en choisissant la valeur la plus élevée (*Maximum Energy Selection*), le premier échantillon dépassant un certain seuil (*Threshold Comparison*) etc.

Lorsque la couche physique n'est pas en mesure de fournir ces données aux couches supérieures, le microcontrôleur peut se risquer à estimer le temps de vol. Malheureusement, cette technique a très peu de chance d'être efficace car les horloges disponibles à ce niveau sont généralement plus lentes. De plus, la méthode d'accès au médium et les caractéristiques

²¹ *Time Of Flight*

du *transceiver* détermineront la précision avec laquelle les instants d'émission et de réception peuvent être prédits.

Quels que soient les moyens employés, les mesures seront affectées par des facteurs internes, propres à l'instrument, et des facteurs externes, liés à l'environnement. Si nous restreignons notre étude au cas d'un module communicant exploitant le signal radio pour mesurer la distance, nous nous intéresserons principalement au multitrajet et à la résolution des horloges utilisées.

Lorsqu'un signal radio se propage de l'émetteur vers le récepteur, il emprunte plusieurs trajets en plus du chemin direct. A l'issue de ces différents trajets, sont obtenues des versions de signal dont l'amplitude et la phase auront été affectées. De plus, le chemin parcouru étant plus long que la ligne droite, ces versions présenteront un certain retard. Au niveau du récepteur, l'une de ces versions sera reconnue. Il ne s'agit pas toujours du chemin direct d'où une surestimation automatique du temps de vol et donc de la distance estimée. Par exemple, dans le cas de l'algorithme *Maximum Energy Selection* appliqué à la sortie d'un *transceiver* ED, le chemin le plus court peut ne pas être reconnu au profit d'un chemin secondaire pour lequel le niveau d'énergie est plus élevé : ceci se produit typiquement lorsque le signal issu d'une réflexion est plus puissant que le signal originel.

Au niveau du module lui-même, les horloges utilisées pour générer les estampilles temporelles ont une influence sur la qualité de la mesure. Soit P_{CLK} la période de cette horloge, T_{impact} l'instant de réception et T_m l'instant d'impact mesuré. On suppose que T_m est un nombre entier. L'impossibilité de prédire l'instant d'impact conduit à (équation 8)

$$T_m > T_{impact} \quad (8)$$

Dans le meilleur des cas, T_{impact} est compris entre T_{m-1} et T_m . L'écart entre ces deux valeurs consécutives est P_{CLK} . Si P_{CLK} est grand, cela implique une erreur appréciable sur l'instant d'arrivée du signal. Par exemple, une résolution temporelle d' $1\mu s$ conduit à un niveau de précision de 300m. Une localisation très précise bénéficiera d'une horloge dont la fréquence est de l'ordre du gigahertz car l'erreur sera alors d'une trentaine de centimètres.

Le choix de la technologie à utiliser n'est pas anodin quand on sait qu'une relation (équation 9) existe entre la bande passante du signal et la résolution en distance, c étant la vitesse de la lumière.

$$d = \frac{c}{BP} [\text{m}] \quad (9)$$

Il est donc préférable de choisir des signaux de bande passante élevée. En ce sens, la couche physique *Ultra-Wide Band* (UWB) est très prometteuse [42]. Avec une bande passante d'au moins 500MHz, elle permet une résolution en distance de l'ordre de la soixantaine de centimètres. La norme IEEE 802.15.4 [54] pour les réseaux sans fil à faible débit et faible consommation énergétique a justement intégré cette technologie dans la liste des couches physiques supportées dans un objectif avoué de mesure de distance par le temps de vol. Trois des canaux définis par le texte de la norme disposent d'une bande d'1GHz. D'autre part, un cadre protocolaire pour la mesure de distance a été spécifié sur la base de l'UWB.

L'étude menée dans [84] exploite le TDOA²² et la mesure de phase. Pour augmenter sa précision, elle utilise l'ensemble des canaux définis par IEEE 802.15.4 dans la bande 2.4 GHz donc un signal de 80MHz.

1.1.2.3. Angle d'incidence

La localisation peut encore exploiter l'angle d'incidence du signal radio. Cette mesure est appropriée pour des signaux bande étroite, contrairement au temps de vol. Dans la littérature, deux méthodes principales sont employées.

Une antenne à faisceau étroit motorisée émettant, pour chaque pas de son angle de rotation, la valeur de cet angle. Du côté récepteur, l'angle spécifié par le message correspondant à la meilleure réception est considéré comme l'angle d'arrivée

Le récepteur est formé d'un vecteur d'antennes dont l'espacement est choisi pour éviter le couplage entre les voisins. Le signal incident est reçu par les différentes interfaces radio et traité par un algorithme adapté (MUSIC²³, ESPRIT²⁴...). Dans [79] une méthode générique de formulation du problème est proposée. Plus tard, dans [91], l'exploitation par un réseau GSM de l'AOA²⁵ a été étudiée avec une configuration matérielle et des algorithmes similaires.

²² *Time Difference Of Arrival*

²³ *Multiple Signal Classification*

²⁴ *Estimation of Signal Parameters via Rotational Invariance Technique*

²⁵ *Angle Of Arrival*

Les contraintes matérielles liées à la récupération de l'angle d'incidence rendent cette donnée difficile à exploiter dans le contexte d'un réseau de capteurs. De plus, cette information est très sensible au multitrajet et ne convient qu'à des situations en espace ouvert.

1.1.2.4. Phase du signal

La mesure de la phase du signal permet une évaluation très précise de la distance. Supposons que le signal de longueur d'onde λ soit une simple sinusoïde : si le dispositif de mesure renvoie un déphasage θ , cela indique qu'il se trouve à une distance λ^* de la source, écart donné par l'équation 10. Ainsi, pour un signal de 1GHz, la longueur d'onde λ sera de 30cm : θ prenant ses valeurs dans l'intervalle $[0 ; 2\pi]$, λ^* sera compris entre 0 et 30cm.

$$\lambda^* = \frac{\lambda\theta}{2\pi} \quad (10)$$

Cette solution repose sur une corrélation entre le signal venant du mobile et une copie locale de ce signal. Par conséquent, le réseau, au moins au moment de cette opération, doit être synchronisé.

Il faut noter que cette technique ne peut s'utiliser seule à des distances supérieures à λ . En raison des caractéristiques de la mesure des angles, il est impossible de différencier θ et $\theta+k2\pi$, $k \in \mathbb{N}$ [40]. Deux pistes existent : associer une autre technologie permettant de se placer en un point de l'espace pour ensuite affiner la mesure, ou travailler à basse fréquence [32]. Dans ce dernier cas de figure, la dimension de la longueur d'onde en basse fréquence est exploitée. Dans la bande LF (*Low Frequency*) allant de 30kHz à 300kHz, la longueur d'onde sera comprise entre 1 km et 10 km, ce qui est largement suffisant pour se localiser dans un bâtiment ordinaire.

1.1.2.5. Nombre de sauts

Dans le domaine *range-free*, l'un des paramètres souvent exploités est le nombre de sauts ou *hopcount* [75]. Il s'agit du nombre de liens radios empruntés par un message de la source à la destination à travers des nœuds relais. Ceci implique la capacité pour les intermédiaires de modifier certains champs de la trame. Ils doivent à minima pouvoir incrémenter le champ dédié au nombre de sauts avant de retransmettre le message. Si plusieurs chemins sont possibles, le message ayant été diffusé (*broadcast*), des mécanismes protocolaires doivent être mis en place pour assurer la cohérence des données. En particulier, éviter les boucles menant

à des mises à jour erronées de *hopcount* est l'une des conditions de succès dans la dissémination de l'information.

Ce *hopcount* est ensuite combiné à la taille moyenne d'un saut pour estimer la distance séparant les mobiles des ancrés du réseau. Dans un contexte réel, le concept de taille moyenne de saut devient la pierre d'achoppement de cette méthode : en effet, à moins d'une répartition régulière des nœuds, mobiles et ancrés, la taille des sauts varie dans le temps en raison du mouvement mais aussi des effets des perturbations électromagnétiques sur les transmissions. D'un autre côté, les chemins suivis par les messages n'étant pas rectilignes, les surestimations sont inévitables.

1.1.2.6. Autres paramètres

Finalement, il existe encore plusieurs informations pouvant être intégrées au processus de localisation. L'une des plus élémentaires est la connectivité : associée à une technologie à très faible portée, cette information peut mener à une très grande précision au prix d'une forte densité de nœuds de référence.

D'autres données ne sont pas directement utilisées dans le calcul de la position mais servent à assurer l'implémentabilité de la solution, comme les hypothèses de répartition homogène des ancrés.

1.2. Protocoles de mesure

Les paragraphes précédents ont introduit les différentes caractéristiques pouvant être exploitées dans le cadre de la localisation. Certaines résultent de mesures, d'autres sont liées à des hypothèses relatives à l'environnement. Les méthodes fondées sur les premières seront regroupées au sein de la famille de solutions *range-based* car dépendant de mesures pouvant principalement être converties en distance. Une seconde famille dite *range-free* utilisera des données telles que la connectivité ou le nombre de sauts.

Selon l'information retenue, un protocole de mesure spécifique doit piloter le matériel en vue de récupérer des échantillons. Nous considérons comme faisant partie du protocole tous les échanges de messages nécessaires à l'exécution de l'algorithme de localisation. Un examen des propositions venant des différentes familles de solutions a permis la classification présentée dans les paragraphes suivants.

1.2.1. Protocoles de collecte dans les méthodes *range-based*

Selon la caractéristique du signal à récupérer, le processus de collecte change. Nous présentons dans ce qui suit les deux principales caractéristiques physiques utilisables, la puissance du signal et le temps. Pour plus de détails, le lecteur peut se référer à [25].

1.2.1.1. Indicateur de puissance de signal reçu

La localisation par le *RSSI* implique le stockage de la valeur de ce paramètre mesuré pour une trame entrante dans une structure dédiée. Les nœuds peuvent exploiter les communications existantes pour générer leurs données et donc minimiser l'impact de la collecte sur la disponibilité du médium. Deux grandes tendances peuvent être identifiées :

Mobile centric : dans cette configuration, le mobile émet périodiquement des messages que les ancres recueillent. Une fois le *RSSI* mesuré, les divers échantillons doivent être renvoyés à un serveur central qui les traitera en vue de la localisation. Le protocole de collecte va donc opérer principalement durant la phase d'acheminement des données vers le serveur. Le travail de [17] se situe dans cette lignée.

Anchor centric : à l'inverse du cas précédent, le mobile reste à l'écoute du médium pour identifier ses voisins directs et le *RSSI* associé à leurs transmissions [47]. Dépendamment du type d'algorithme utilisé dans la solution, le mobile transfèrera ou non les informations récupérées à un serveur : si l'objectif est de transformer directement les mesures en distances et d'exécuter un algorithme léger de calcul de position, aucun autre échange n'est nécessaire. Au contraire, quand des moyens plus importants sont nécessaires, le mobile doit faire parvenir au serveur les informations qu'il détient.

Une classification plus générale existe et peut être obtenue dans [21]. Dans ce cadre sont définies différentes architectures (centrée mobile, centrée infrastructure, ad-hoc ...) mais aussi des critères permettant d'organiser les propositions de systèmes de localisation en fonction de leurs performances, des algorithmes utilisés etc. L'architecture étant dans ce contexte définie par rapport à l'acteur du calcul de localisation, un protocole décrit dans notre manuscrit comme *mobile centric* (car pour nous dans ce cas, le mobile est chargé d'émettre les données captées par l'infrastructure) sera classé dans [21] comme centré infrastructure (car l'infrastructure sera responsable du calcul de localisation). De la même manière, une exécution du protocole *anchor centric* sera considérée comme centrée mobile.

1.2.1.2. Temps de vol du signal

Nous classerons les protocoles de collecte d'informations temporelles suivant le nombre de signaux intervenant dans leur mise en œuvre. Nous distinguerons donc le mono-signal, et la combinaison de signaux :

1.2.1.2.1. Le mono-signal

Typiquement, le signal de communication est ici mis à contribution. On suppose que les nœuds sont capables de récupérer avec une précision suffisante les instants de départ et d'arrivée des messages. Si le réseau est complètement synchronisé, le *Time Of Arrival* peut être appliqué : ce protocole repose sur l'échange de deux messages : le premier réalisant la mesure et le second le rapatriement des résultats. La figure 5 illustre cet échange : les nœuds A et B partageant une horloge commune, l'estampille collectée par B s'exprime dans la même base temporelle que celle obtenue par A. Les deux valeurs peuvent donc être combinées par l'équation 11.

$$T_{vol} = T_1 - T_0 \quad (11)$$

La figure 5 présente aussi le protocole TWR²⁶ [54]. Ce dernier lève la contrainte de synchronisation sur le réseau. Au lieu de combiner des estampilles, la différence de durées est utilisée pour calculer le temps de vol (équation 12). Ce type de protocole suppose que la variation de la distance entre les deux partenaires est nulle durant l'exécution de la mesure : ainsi, le temps de propagation peut être considéré comme symétrique. Le travail de [55] présente l'exploitation de ce protocole pour une couche physique 802.11. Il est bon de noter que pour obtenir des résultats stables, 300 échantillons sont nécessaires. [65], quant à lui, décrit l'exploitation d'un signal modulé par un PN code ou *ranging signal* pour la mesure du temps de vol. Dans ce cas, il ne s'agit plus d'une paire de trames, données et acquittement, mais d'un code qui, une fois reçu par la cible, est retransmis et corrélé au signal originel pour identifier le temps de vol.

$$T_{vol} = \frac{(T_3 - T_0) - (T_2 - T_1)}{2} \quad (12)$$

Les nœuds A et B sont donc chargés de récupérer les instants de départ et d'arrivée des messages. Une seconde trame de données est requise pour le transfert des informations à A. TWR ne résout pourtant pas tous les problèmes. Il peut arriver que les horloges de A et B

²⁶ *Two-Way Ranging*

soient différentes : ceci signifie qu'une microseconde mesurée par A n'a pas la même durée qu'une microseconde mesurée par B. Pour lutter contre ce phénomène, une troisième procédure de collecte a été proposée, le *Symmetric Double-Sided Two-Way Ranging* (SDS-TWR). Présenté dans [ieee802154a], ce protocole repose sur deux exécutions symétriques de TWR (figure 6). Les huit estampilles sont rassemblées après réception de la trame DATA_REPLY et sont introduites dans l'équation 13.

$$t_{vol} = \frac{t_4 - t_1 + t_8 - t_5 - t_3 + t_2 - t_7 + t_6}{8} \quad (13)$$

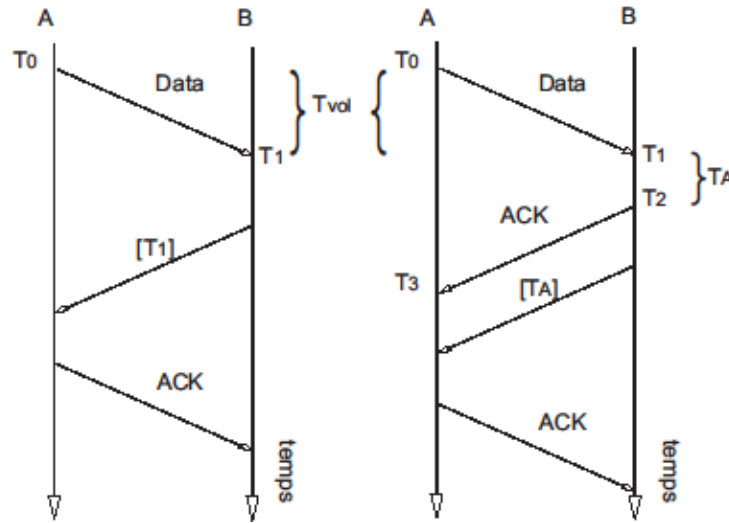


Figure 5 : Time Of Arrival et Two Way Ranging

Le SDS-TWR réduit l'écart entre la valeur estimée et la vraie valeur du temps de propagation. Si on appelle t_{reply1} la différence $t_4 - t_3$, t_{reply2} la différence $t_2 - t_1$, Δ_{reply} l'écart entre t_{reply2} et t_{reply1} , t_{vol} la vraie valeur du temps de vol et \hat{t} son estimation, la précision passe de l'expression donnée par l'équation 14 à celle de l'équation 15.

$$\hat{t}_{vol} - t_{vol} \approx \frac{1}{2} \times t_{reply2} \times (e_1 - e_2) \quad (14)$$

$$\hat{t}_{vol} - t_{vol} \approx \frac{1}{4} \times \Delta_{reply} \times (e_1 - e_2) \quad (15)$$

[64] utilise le protocole SDS-TWR et ajoute au processus de communication une phase de réservation d'un slot temporel dédié à l'exécution du *ranging*. L'affectation du slot est réalisée par un nœud coordinateur : ce dernier publie la liste des slots disponibles puis, une fois les requêtes des mobiles obtenues, signale à chaque mobile qu'il peut entamer sa mesure de distance. Avant de pouvoir contacter le mobile suivant, le coordinateur doit recevoir une

trame de libération provenant du mobile courant. Afin d'assurer le passage à l'échelle, le coordinateur est défini sur une cellule et non sur l'ensemble du réseau. Pendant la mesure d'un mobile, les autres mobiles entrent en mode *sleep*.

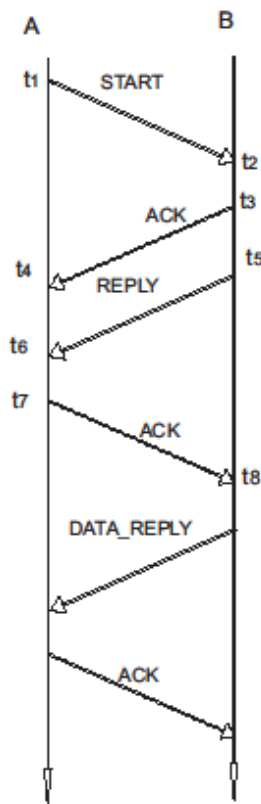


Figure 6 : Le protocole *Symmetric Double-Sided Two-Way Ranging*

Que l'on utilise TWR ou SDS-TWR, il existe une possibilité de corruption des données liées aux échecs de la communication sur le médium. La figure 7 illustre notre propos. Soient M et A les deux nœuds impliqués dans le *ranging*. Le temps de vol peut être calculé comme la moitié de la différence entre t_M et t_A . La mesure de t_M débute lorsque M soumet sa trame pour émission à la couche physique et prend fin à la réception de l'ACK. Dans le cas idéal, A commencerait la mesure de t_A à la réception de la trame **START** et arrêterait le processus à l'émission de l'ACK. Dans la réalité, la capture de l'instant d'émission étant très rarement supportée, il est critique de déléguer la génération et l'émission de l'ACK à la couche PHY de manière à assurer un t_A constant. D'autre part, impliquer les couches supérieures dans la génération de l'ACK introduit un autre problème. Le calcul final devra tenir compte du temps de traitement associé. Certes, ce dernier peut être mesuré grâce aux horloges du microcontrôleur mais il s'agira d'une estimation grossière, affectée par la granularité des instruments.

Lorsque des interférences affectent les communications, l'acquittement prend tout son sens. En effet, ce mécanisme permet de reconnaître qu'une perte de trame a eu lieu et provoque une réémission du message originel.

Si ce processus tend à assurer que le message arrive à destination, il a un impact négatif sur la mesure. En effet, du côté de l'émetteur, les retransmissions ne seront pas prises en compte dans la mesure du temps car la couche physique ne peut indiquer que le succès ou l'échec d'une communication. Par conséquent, la durée de t_M sera incohérente par rapport à celle de t_A : leur combinaison génèrera une surestimation de la distance parcourue par le signal.

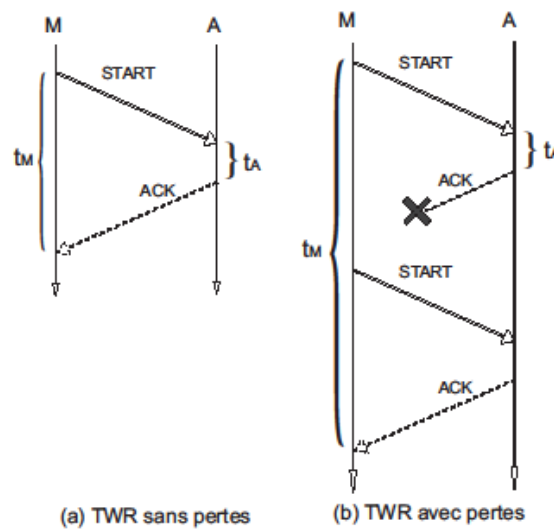


Figure 7 : Corruption de la mesure dans TWR

Le *Time Difference Of Arrival* (TDOA) est un cas particulier reposant sur un réseau partiellement synchronisé : seules les ancrs partagent une horloge commune. Typiquement, le mobile émet un message qui est reçu par un groupe d'ancres. Parmi ces dernières, une ancre de référence est désignée et les différences entre les instants d'arrivée du signal au niveau de la référence et d'une autre ancre i sont calculées. Le protocole régule alors l'émission de la première trame de la séquence par le mobile puis assure la mise en commun des différents échantillons au niveau de l'ancre de référence.

En termes de synchronisation, TDOA se place à mi-chemin entre TOA et TWR car exigeant uniquement la synchronisation des ancrs. Dans un contexte sans fil, établir et conserver l'homogénéité des horloges à la nanoseconde près n'est malencontreusement pas un problème trivial

1.2.1.2.2. La combinaison de signaux

Les capacités d'un nœud communicant peuvent être étendues à l'aide de divers modules. Certains de ces sous-systèmes sont capables de supporter la communication sans fil. Si la vitesse de propagation de ce signal additionnel est distincte de celle du signal radio, il est possible de calculer la distance en se basant uniquement sur la différence entre les instants d'arrivées. Connaissant la différence entre les instants d'émission t_0 , la relation entre la distance et la différence des instants τ est donnée par l'équation 16.

$$distance = (\tau - t_0) * \frac{(v_1 * v_2)}{(v_1 - v_2)} \quad (16)$$

Typiquement, on retrouve associé au signal radio un signal de vitesse inférieure comme l'ultrason ($v_{ultrason}=344\text{m/s}$ dans l'air à 20°C) [70][36].

1.2.2. Protocoles de collecte pour les méthodes *range-free*

Les méthodes *range-free* se dispensent de mesurer la distance séparant le mobile des balises. Elles exploitent des hypothèses sur la connectivité en vue d'estimer les écarts avant d'appliquer la trilatération (§ 2.1.4.1) ou une méthode géométrique équivalente. La famille DV-Hop (DV-Distance [75], CDV-Hop [58], *Selective 3-Anchor DV-Hop* [75] etc) se base sur l'inondation pour collecter les informations concernant le réseau comme l'adresse des balises et le nombre de sauts séparant le mobile de chacune des balises. Ce genre de protocole génère un nombre important de collisions : en sus des trames perdues, les chemins identifiés ne sont pas nécessairement les plus courts. Dans [46], l'inondation est implémentée sous la forme d'un Enhanced CSMA/CA ou eCSMA/CA permettant de réduire les collisions et limitée dans le temps par des *timers* adaptés.

La majeure partie des propositions *range-free* ne spécifie pas clairement le protocole de communication utilisé mais l'étude de leur fonctionnement fournit des indications quant aux pré-requis. Par exemple, [49] peut être mis en place au-dessus d'un réseau 802.15.4 dans lequel les balises émettent périodiquement des *beacons* à un saut.

D'autres, tels que [1] demandent une réglementation particulière : dans cet exemple précis, les balises disposent d'une antenne omnidirectionnelle et d'une antenne directionnelle à faisceau étroit. Tout en faisant tourner cette dernière, elles émettent leur position et l'angle actuel de rotation. L'agencement temporel des exécutions de cette manœuvre par les balises n'est que l'un des aspects demandant à être clairement défini.

1.3. Algorithmes de localisation

Comme nous l'avons vu dans l'introduction de la thèse, la localisation n'est pas une problématique récente. De fait, nous commencerons notre panorama par une description des méthodes fondamentales de détermination de la position puis nous passerons à des calculs plus modernes et directement liés au domaine de la localisation dans les réseaux de capteurs.

1.3.1. Algorithmes généraux

Tous les algorithmes supposent un ensemble de données à l'entrée qui doivent être transformées en vue d'obtenir une estimation de la position. Ils ne se valent pourtant pas tous en termes de robustesse. Certains, dans leur forme standard, attendent des données exactes et échouent lorsque les informations de la réalité leur sont fournies car ces dernières comportent typiquement une erreur.

Dans cette section, nous verrons d'abord les outils qui font l'hypothèse de données exactes puis ceux qui tiennent compte de la présence d'erreurs dans les mesures.

1.3.1.1. Trilatération

Cette méthode repose sur la connaissance des distances séparant la cible de différents points de référence et des coordonnées spatiales de ces ancres [5].

Soient trois balises B_1 , B_2 et B_3 , M le nœud mobile que l'on désire localiser. L'exécution du protocole de mesure de distance a produit le triplet (d_1, d_2, d_3) et les échanges de données ont permis au mobile de connaître les positions des balises. Les équations 17 et 18 décrivent la relation entre B_1 , B_2 et M .

$$(x - x_1)^2 + (y - y_1)^2 = d_1^2 \quad (17)$$

$$(x - x_2)^2 + (y - y_2)^2 = d_2^2 \quad (18)$$

La position recherchée se trouve donc à l'intersection des cercles $C_1 (B_1, d_1)$ et $C_2 (B_2, d_2)$. Dans le cas général, C_1 et C_2 se rencontrent en deux points distincts M et M' . Grâce aux données de l'ancre B_3 , l'un de ces deux candidats peut être retenu comme position du mobile (figure 8).

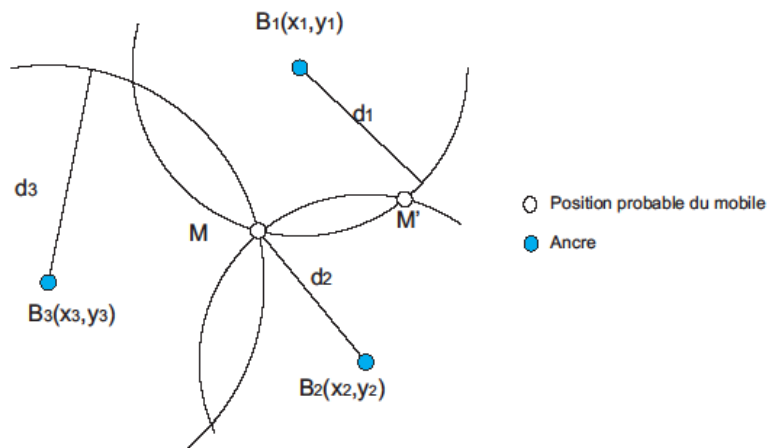


Figure 8 : Trilatération

1.3.1.2. Triangulation

L'angle d'incidence du signal émis par le mobile peut être utilisé pour retrouver sa position [5]. 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.

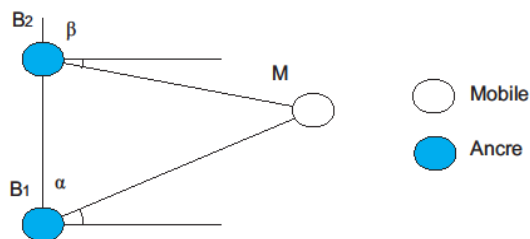


Figure 9 : Angle d'incidence et position

Le point M se retrouve à l'intersection des droites passant par les couples (B_1, M) et (B_2, M) (figure 9). En définissant un nouveau repère dont l'origine est B_1 et où (B_1, B_2) se confond avec l'axe des ordonnées, le rapprochement des équations des droites conduit à l'expression 19 de la position de M .

$$M\left(\frac{y_2}{\tan \alpha + \tan \beta}; \frac{y_2 \tan \alpha}{\tan \alpha + \tan \beta}\right) \quad (19)$$

1.3.1.3. Reconnaissance d'empreinte

Dans ce contexte, nous définirons une empreinte comme une collection d'échantillons d'une caractéristique susceptible de varier avec la position. Nous distinguerons les empreintes de référence des *online fingerprints*. Dans le premier cas, il s'agit de données capturées avant la mise en service du système de localisation (phase *offline*) : elles constituent un modèle de

l'évolution des composantes de l'empreinte dans l'espace. Une fois la phase active déclenchée, de nouvelles mesures, dites *online*, seront collectées. Ces dernières seront comparées au modèle disponible en mémoire : la position correspondant à l'empreinte de référence la plus « proche » du cliché récent sera sélectionnée comme solution.

Le *fingerprinting*²⁷ a engendré moult propositions grâce à sa modularité [22][32]. Les contributions ont surtout porté sur la traduction mathématique de la proximité entre deux échantillons en tenant compte des caractéristiques des échantillons. Malheureusement, la mise en place de cette technique est restreinte par la nécessité d'une phase d'apprentissage. Dans les cas où la solution ne peut mettre à jour dynamiquement et de manière autonome les clichés de référence, une nouvelle campagne de mesures sera nécessaire pour tout changement significatif de l'environnement. Des situations aussi ordinaires que la présence d'individus dans le local pourront exercer une influence significative sur la solution.

1.3.1.4. Multilatération

La multilatération exploite la différence entre les instants d'arrivée du signal ou TDOA pour calculer la position [73].

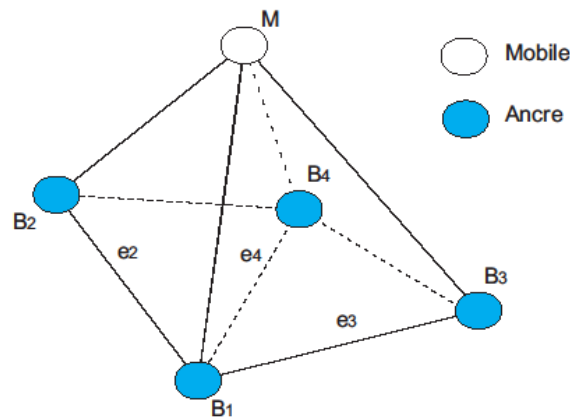


Figure 10 : Multilatération

Le mobile est ici en contact radio avec des balises $B_i (x_i, y_i)$, $i=1 \dots 4$. Le mobile émet un message et l'instant d'arrivée d'un message à la station i est noté t_i . Les ancres étant synchronisées, la différence entre ces instants peut être utilisée pour créer le système d'équations 20, 21, 22 où c est la vitesse de propagation, x_i et y_i les positions des nœuds B_i et e_i la distance entre le nœud B_i et une station choisie comme référence temporelle. Sur la figure 10, B_1 joue ce rôle.

²⁷ Reconnaissance d'empreinte

$$0 = c(t_2 - t_4) - \frac{e_2^2}{ct_2} + \frac{e_4^2}{ct_4} + 2x\left(\frac{x_2}{ct_2} - \frac{x_4}{ct_4}\right) + 2y\left(\frac{y_2}{ct_2} - \frac{y_4}{ct_4}\right) + 2z\left(\frac{z_2}{ct_2} - \frac{z_4}{ct_4}\right) \quad (20)$$

$$0 = c(t_2 - t_3) - \frac{e_2^2}{ct_2} + \frac{e_3^2}{ct_3} + 2x\left(\frac{x_2}{ct_2} - \frac{x_3}{ct_3}\right) + 2y\left(\frac{y_2}{ct_2} - \frac{y_3}{ct_3}\right) + 2z\left(\frac{z_2}{ct_2} - \frac{z_3}{ct_3}\right) \quad (21)$$

$$0 = c(t_3 - t_4) - \frac{e_3^2}{ct_3} + \frac{e_4^2}{ct_4} + 2x\left(\frac{x_3}{ct_3} - \frac{x_4}{ct_4}\right) + 2y\left(\frac{y_3}{ct_3} - \frac{y_4}{ct_4}\right) + 2z\left(\frac{z_3}{ct_3} - \frac{z_4}{ct_4}\right) \quad (22)$$

Partant d'une relation géométrique entre les nœuds, les méthodes présentées précédemment permettent de calculer la position d'un mobile dans le plan (2D). L'ajout d'une balise ou d'un moyen de mesure supplémentaire permettrait d'enrichir le système et donnerait une position dans l'espace (3D).

Alors que les algorithmes introduits jusqu'ici échouent si les données fournies en entrée sont imparfaites, ceux de la seconde partie de cette section tolèrent une certaine erreur. Par conséquent, ces derniers seront appliqués au traitement des données issues du monde réel.

1.3.1.5. Filtre de Kalman

Le filtre de Kalman [44] fait appel aux variables et concepts listés dans le tableau 2.

Tableau 2 : Variables du Filtre de Kalman

Variable	Sémantique
$x(t_k)=x_k$	Etat du système à l'instant t_k
$F(t_k)=F$	Dynamique du système
M_k	Mesures d'une caractéristique du système
H	Sensibilité des mesures à l'état courant du système
w_k	Bruit du système
v_k	Bruit des mesures

Pour alléger l'écriture, nous considérerons un système dont la dynamique n'évolue pas au cours du temps (équation 23).

$$F(t_k)=F \quad (23)$$

La variable x_k représente l'état du système que nous désirons suivre à l'instant t_k . Cet état ne peut être directement observé mais est lié aux mesures réalisées. Ce lien est modélisé par la

matrice H . Les perturbations v_k sont dues au processus de mesure. Les différents bruits correspondent à des variables gaussiennes.

Le système peut donc être décrit par le système d'équations 24.

$$\begin{aligned}x_k &= F \cdot x_{k-1} + w_k \\m_k &= H \cdot x_k + v_k\end{aligned}\tag{24}$$

Le principe du filtre peut être résumé comme suit : à partir d'un état x_{k-1} et du modèle d'évolution du système linéaire considéré, un nouvel état x_k est prédit. Le vecteur de mesures est alors mis à jour et son résultat permet de corriger l'estimation à priori de l'état du système. Ce filtre combine donc une connaissance théorique du système à des mesures en temps réel.

Le filtre de *Kalman* est optimal lorsqu'il est utilisé pour l'étude de systèmes linéaires perturbés par des bruits gaussiens. Il est dit optimal dans le sens de la minimisation de l'erreur statistique. La palette de cas peut être élargie en utilisant l'*Extended Kalman Filter* (EKF) [44] ou l'*Unscented Kalman Filter* [7]. Par exemple, dans le contexte de l'EKF, les équations d'état non linéaires sont transformées à l'aide d'un développement limité.

1.3.1.6. Filtre particulaire [7]

Si le filtre particulaire conserve les notions de loi d'évolution du système et de loi de correction, il supprime les contraintes liées à la distribution de probabilité suivie par les erreurs. Un ensemble de points appelées particules est généré, chacune associée à un poids. Chaque particule est susceptible d'explorer une des directions possibles à partir de l'état précédent et de la dynamique du système. Si au départ, les poids peuvent être équivalents, l'étape de mise à jour fait évoluer ces valeurs de telle manière que certaines particules disparaissent de l'ensemble de solution, au risque d'appauvrir significativement l'ensemble des solutions. Certaines adaptations ont été proposées pour éviter ce phénomène [32].

L'un des aspects critiques de la localisation est le temps de réponse du système. En effet, si le calcul est suffisamment long pour que le mobile ait à nouveau bougé, l'information obtenue sera généralement perçue comme de très mauvaise qualité. Dans le cas du filtre particulaire, si le nombre de particules doit être élevé, il implique une durée de traitement assez longue suivant la plateforme matérielle utilisée. La taille du nuage peut être réduite par l'implémentation de restrictions sur les déplacements possibles à travers des cartographies de l'environnement [32] [60].

1.3.1.7. Estimateur des moindres carrés

Soit un système linéaire pouvant s'écrire sous la forme de l'équation 25.

$$H \cdot x = z \quad (25)$$

où z est un vecteur de l lignes et H une matrice de format $l \times n$, deux structures de données connues, et x un vecteur $n \times 1$ que l'on cherche à estimer. Pour faciliter la compréhension, considérons, z comme un vecteur de mesures, x comme l'état du système et H comme la modélisation de la relation entre x et z . Dans les conditions idéales, les données disponibles permettent d'identifier l'état du système sans équivoque. Mais dans la réalité, l'estimation de x , notée \hat{x} , est influencée par les erreurs affectant le vecteur de mesures z . L'identification de l'état passe alors par la minimisation de l'erreur, donc de l'expression :

$$e = H \cdot \hat{x} - z \quad (26)$$

Quand l'écriture du membre de droite est connue, il est possible d'étudier la dérivée de sa norme en vue d'identifier la solution. Dans le cas où l'objectif n'est pas de forme linéaire, la résolution peut se faire par la méthode des Moindres Carrés Non-linéaires [20].

1.3.1.8. Inférence bayésienne

Considérée comme le cœur des méthodes probabilistes, l'inférence bayésienne (*Bayesian Inference* ou BI) [56] lie le paramètre recherché à la connaissance disponible. Dans le cas de la localisation, une probabilité conditionnelle basée sur les mesures est définie. En maximisant cette expression, la position peut être estimée.

1.3.1.9. Sélection des plus proches voisins

Souvent appliquée dans le cadre du *fingerprinting*, la méthode dite de *Nearest Neighbors* (NN) [67] sélectionne, parmi un ensemble de données de référence, un échantillon jugé le plus proche de la caractéristique mesurée durant la phase *online*. Différentes adaptations ont été proposées en exprimant de manière mathématique la distance entre deux échantillons dont les dimensions varient (valeur, vecteur, etc.). Cette distance peut être euclidienne, basée sur la distance de *Hamming* etc...

k-NN constitue une variante NN qui retourne les k plus proches voisins qui seront ensuite combinés pour obtenir l'estimation finale. Cette technique permet d'atténuer les effets d'un choix erroné en diversifiant les résultats intermédiaires.

Avec d'autres techniques de *data mining*, les algorithmes de recherche de voisins les plus proches constituent des outils de choix lors de la localisation déterministe à partir d'un ensemble de positions prédéfinies.

1.3.1.10. Recherche séquentielle

La recherche séquentielle est l'une des méthodes d'investigation les plus simples car elle repose que l'examen de tous les candidats, sans optimisations permettant de réduire le nombre de ces derniers. Elle constitue la base des algorithmes de type « analyse de matrice » ou *grid-scan* [49]. Ces algorithmes se décomposent en deux phases : l'identification d'une zone de recherche suivie de la création de la solution. Des tests sont définis qui permettent de limiter la portion de l'espace à analyser en fonction du type d'informations disponibles. Une fois la zone d'intérêt identifiée, elle est subdivisée en cellules. Chacune de ces dernières se verra affecter un poids qui évoluera suivant les résultats des tests exécutés sur la cellule. Finalement, les cases ayant le poids le plus élevé seront combinées pour former la solution.

1.3.2. Adaptations récentes des algorithmes à la localisation par WSN

Nous allons voir maintenant comment les différents algorithmes que nous avons investigués sont implémentés aujourd'hui au sein de réseaux de capteurs. D'autres outils seront également introduits à ce moment et leur relation aux algorithmes de la section précédente mise en évidence. Nous classerons ces propositions en fonction des familles de localisation et de l'information exploitée dans chaque cas.

1.3.2.1. Méthodes *range-based*

Comme indiqué précédemment, l'ensemble des solutions de localisation exploitant la mesure d'une caractéristique, typiquement du signal radio, est désigné sous le vocable de *range-based*. Dans le contexte des réseaux de capteurs sans fil, les données temporelles et relatives à la puissance sont les plus souvent utilisées car relativement faciles d'accès. Nous nous proposons, dans les paragraphes de cette sous-section, de décrire quelques implémentations représentatives de l'existant.

1.3.2.1.1. Temps de vol

L'information temporelle est communément exploitée pour estimer la distance \hat{d} entre le mobile et les balises. Les positions de ces dernières étant connues, plusieurs voies permettent de calculer les coordonnées du mobile. Dans [68], une fonction de coût combine les

estimations de la distance, les positions connues et les coordonnées recherchées (équations 27 et 28). L'exécution itérative de cet algorithme demande une implémentation centralisée mais aussi un point de départ du calcul ou *seed* proche de la vraie solution.

$$F(x) = \sum_{i=1}^N \alpha_i^2 f_i^2(x) \quad (27)$$

$$f_i(x, y) = c\Delta_i - \sqrt{(x-x_i)^2 + (y-y_i)^2} \quad (28)$$

Le travail décrit dans [55] comporte deux parties : la mesure de distance et le positionnement. La seconde phase exploite, à travers MATLAB, les algorithmes *Linear Least Squares*, *Non Linear Least Squares* et *Independent Time GPS Least Squares* pour calculer la position. Les estimations de distance sont générées à partir de résultats d'expérience utilisant la technologie Wifi. La répartition gaussienne des mesures a été introduite dans les simulations afin de reconstituer cet aspect de l'expérience réelle.

Dans [60], les auteurs utilisent la mesure du temps de vol par réalisée par une plateforme CSS²⁸ exécutant le protocole SDS-TWR. Les calculs sont réalisés par un filtre particulière incorporant un modèle de mouvement *brownien*²⁹. Le système développé requiert la connaissance de l'environnement physique donc une carte signalant les murs, les portes etc.. A partir de ces modèles, les nœuds reconstituent le chemin suivi par le signal et utilisent cette information pour mettre à jour le poids des particules.

Dans [64], le même protocole de mesure est utilisé mais le traitement est implémenté par un filtre de Kalman.

1.3.2.1.2. RSSI et puissance reçue

L'une des premières études exploitant le signal radio vient de [11]. L'étude de RADAR utilise la puissance du signal et le rapport signal à bruit (*Signal to Noise Ratio*, SNR) pour déterminer la position. Les nœuds du réseau sont soit des balises, soit des mobiles, tous synchronisés.

RADAR utilise la méthode de *fingerpinting* : durant la phase hors-ligne (*offline*), chaque balise enregistre la puissance reçue du mobile en y associant l'instant t où la trame a été reçue ainsi que la position et l'orientation du mobile au moment de l'émission. Au terme de cette phase, une base de données contenant toutes ces informations est générée.

²⁸ *Chirp Spread Spectrum*

²⁹ Mouvement aléatoire influencé par les chocs survenant au cours du déplacement rectiligne par ailleurs

Durant l'étape en ligne (*online*), les balises conservent une valeur moyenne de la puissance du signal et choisissent, pour retrouver la position, entre une combinaison des valeurs empiriques et un modèle de propagation. La recherche de la position se fait par la technique du *Nearest Neighbor in Signal Space (NNSS)* : cette méthode élit l'enregistrement pour lequel la distance entre le vecteur des *Signal Strength (SS)* stocké et celui mesuré est minimale. Une amélioration est considérée par le choix des k meilleurs NNSS et le calcul de leur centre de gravité comme la position du mobile. Cette amélioration est encore plus nette si les k plus proches voisins sont choisis en ignorant les orientations : les points sont alors physiquement distincts : les triplets $\langle \text{Position}_i, \text{Puissance}_i, \text{Nord} \rangle$ et $\langle \text{Position}_i, \text{Puissance}_i, \text{Sud} \rangle$ ne sont plus différenciés.

Si avec RADAR, les valeurs brutes de RSSI ont été utilisées, le travail réalisé dans [31] se propose de partir plutôt de la différence de RSSI entre les paires de références. La solution se basant sur le réseau sans fil IEEE 802.11, ces nœuds références sont des *Access Point (AP)*. Partant du constat que la lecture du RSSI dépend du nœud effectuant la mesure, les auteurs ont cherché une utilisation du RSSI qui puisse être employée par une grande variété de matériels. Leur méthode réduit l'espace à un ensemble fini discret de positions. Durant la phase *offline*, en chacun de ces points, le mobile récupère des valeurs de RSSI pour chaque AP détecté. Des paires d'AP (AP_i, AP_j) sont ensuite formées avec comme contrainte $i < j$. Pour chaque paire, la différence des valeurs de RSSI obtenues est calculée. Ce procédé a été validé par l'examen de relevés réalisés à partir de modules différents compatibles 802.11 : bien que les valeurs de RSSI obtenues ne soient pas identiques, les courbes tracées à partir de ces données ont la même allure.

Etablir ces différences n'étant pas une localisation en soi, les auteurs ont intégré leur proposition à deux méthodes connues : *Nearest Neighbor* et *Bayesian Inference* (§ 1.3.1.8 et 1.3.1.9). NN repose sur une base de données de mesures qui sert de référence durant la phase *online*. La position juste est celle correspondant au point de l'espace des RSSI dont la distance euclidienne avec les valeurs mesurées est minimale. Un ajustement est apporté pour calculer la distance entre les différences. BI permet de choisir la position la plus probable connaissant les valeurs actuelles. L'adaptation de cette méthode à la proposition demande l'utilisation d'un vecteur d'observation o lié à une position l . Le vecteur o_k aura la forme (identifiant de paire d'AP, différence de RSSI pour la paire d'AP). La mise en relation de ces valeurs et de la position occupée lors de la mesure durant la phase *offline* permettra d'aboutir à l'équation 29. Partant de là, la probabilité de se trouver à une position l_k sachant la capture courante

est donnée par l'équation 30. Retrouver la vraie position revient alors à maximiser cette probabilité en fonction de l_k .

$$\Pr(o | l_k) = \prod_{1 \leq i < j \leq m} \Pr(d(AP_i, AP_j) | AP_i \times AP_j, l_k) \quad (29)$$

$$P_r(l_k | o^*) = \frac{P_r(o^* | l_k) P_r(l_k)}{\sum_{k=1}^n P_r(o^* | l_k) P_r(l_k)} \quad (30)$$

L'expérience menée sur des points d'accès, un ordinateur portable et un *smartphone*, a montré l'impact positif de la proposition sur les méthodes existantes.

Le travail de [32] utilise un filtre particulière pour combiner *fingerprinting* et navigation inertielle. Un nouvel état est prédit par la loi du mouvement. La disponibilité d'un plan de l'environnement permet d'assurer des déplacements réalistes. La reconnaissance d'empreinte fournit une base de données liant puissance reçue et position. Ces informations sont exploitées pour la transformation du vecteur de mesures, un n-uplet de puissances reçues, en une position. Finalement, le poids de chacune des particules évoluera en fonction de sa proximité avec la position issue des mesures et de la probabilité d'avoir traversé un mur.

Dans [85], ce sont les données des sous-systèmes disponibles d'un *smartphone* qui sont fusionnées de manière opportuniste. Le filtre est donc dans ce cas-ci appelé à intégrer un *fingerprinting* réalisé à partir de la connexion à un réseau Wifi, les données provenant du GPS et du réseau GSM et les données issues du système embarqué de navigation inertielle.

1.3.2.1.3. TDOA et phase du signal

La solution proposée dans [84] exploite le TDOA et la mesure de phase dans un réseau totalement synchronisé.

Une hypothèse de base est la présence de deux émetteurs, T_1 et T_2 , placés sur la ligne reliant deux récepteurs, R_1 et R_2 (figure 11). T_1 et T_2 transmettent quasi simultanément leurs messages de mesure, chacun dans l'un des 16 canaux. La configuration est telle que les transmetteurs et les récepteurs sont à portée les uns des autres. Les émetteurs étant basés sur la norme IEEE 802.15.4, leur portée maximale sera environ de 30m en champ libre.

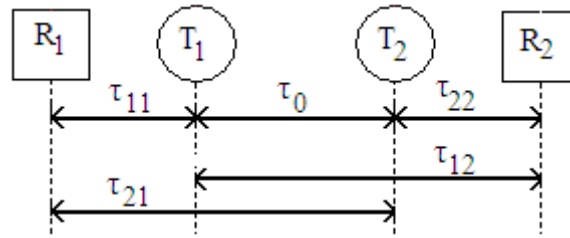


Figure 11 : Disposition des nœuds et mesures temporelles

Les différences de temps d'arrivée au niveau de chaque récepteur sont calculées et référencées par Δt_1 et Δt_2 (équations 31 et 32).

$$\Delta t_1 = \tau_{21} - \tau_{11} + \tau_{t_0} = \tau_0 + \tau_{t_0} \quad (31)$$

$$\Delta t_2 = \tau_{22} - \tau_{12} + \tau_{t_0} = -\tau_0 + \tau_{t_0} \quad (32)$$

Le terme τ_{t_0} représente ici la différence entre les instants d'émission des signaux et τ_0 correspond à l'écart temporel lié à la distance séparant les émetteurs. La combinaison $\Delta \tau$ de ces temps est proportionnelle à la distance séparant les émetteurs, d_0 , d'où l'équation 33.

$$d_0 = c_0 \tau_0 = \frac{c_0}{2} \Delta \tau \quad (33)$$

avec :

$$\Delta \tau = \Delta t_1 - \Delta t_2 \quad (34)$$

Si la position de l'un des deux émetteurs est connue, celle du second l'est automatiquement.

La mesure de la différence de phase s'effectue entre le signal reçu et un exemplaire contenu en mémoire au niveau de chacun des récepteurs. Ceci fournit une information sur la distance séparant la source de la destination. Cette distance sera comprise entre 0 et la longueur d'onde. Les caractéristiques de la mesure angulaire empêchent la distinction des distances associées à φ et $\varphi + 2\Pi$ mais ce résultat pourra être utilisé pour améliorer la précision de la mesure temporelle.

Dans [57], le filtre particulaire est utilisé par un algorithme de suivi de trajectoire pour combiner les différences de temps d'arrivée d'un signal émis par le mobile et les différences de fréquences introduites par ce mouvement.

1.3.2.2. Méthodes Range-Free

A l'instar des techniques *range-based*, les méthodes *range-free* disposent d'un arsenal d'algorithmes permettant de calculer la position à partir de simples déductions. L'un des avantages de cette famille tient au fait que des outils complexes ne sont généralement pas nécessaires pour traiter les informations. Dans la suite, l'utilisation du nombre de sauts et de la connectivité par les propositions décrites illustrera cet aspect. Il convient néanmoins de signaler que cette réduction de la complexité s'accompagne d'une perte significative de justesse sur les estimations.

1.3.2.2.1. Famille DV-Hop

DV-Hop est un algorithme de localisation *range-free* très populaire. Sa version originelle ainsi que plusieurs améliorations sont présentées dans cette section.

DV-Hop originel

La méthode de localisation DV-Hop [75] s'inspire des protocoles de routage à vecteur de distance pour proposer une estimation de la position des mobiles basée sur le nombre de sauts. Chaque ancre est ici chargée d'émettre une trame contenant sa position et un champ dédié au comptage de sauts. A la réception de ce message, tous les nœuds décideront de le retransmettre ou non en comparant le nombre de sauts vers l'ancre contenu dans la trame à l'information stockée en mémoire. Si le *hopcount* est inférieur au contenu de la mémoire, les données sont mises à jour, le champ de la trame est incrémenté et la trame retransmise. Une fois que toutes les balises ont récupéré la position et le nombre de sauts les séparant de leurs homologues, elles calculent une distance moyenne par saut (*hopsizel*). Par exemple, la balise A_1 (figure 12) peut calculer les distances géométriques entre elle et les balises A_2 et A_3 . Elle connaît également le nombre de sauts entre elle et ses voisines. L'équation 35 permet d'estimer la taille de saut. Ici, $d_{i,j}$ est la distance réelle entre les balises i et j et $hop_{i,j}$ correspond au nombre de sauts entre les balises i et j .

$$hopsizel_1 = \frac{d_{1,2} + d_{1,3}}{hop_{1,2} + hop_{1,3}} \quad (35)$$

Cette taille de saut est rediffusée à l'ensemble du réseau. Elle sera utilisée par un nœud mobile quelconque pour estimer sa distance par rapport à une ancre. Un nœud la détermine par l'équation 36 où $hopcount_{i,x}$ est le nombre de sauts entre le mobile x et la balise i . Une fois

les distances à trois balises connues, le mobile peut appliquer la trilatération (§ 1.3.1.1) pour trouver sa position.

$$d_{i,x} = hopsize_i \times hopcount_{i,x} \quad (36)$$

De cette manière, DV-Hop se propose de résoudre le problème de la localisation de tous les nœuds, même ceux hors de portée des balises, les relais de trames s'effectuant alors via les autres nœuds mobiles dans une topologie ad-hoc.

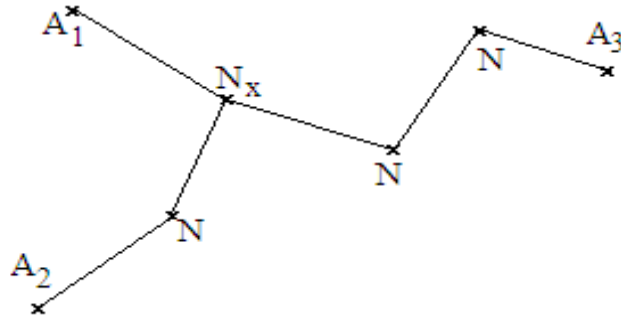


Figure 12 : Principe topologique de DV-Hop

Constraint DV-Hop

Une amélioration a été proposée [58] avec pour point de départ la relation implicite existant entre la portée de communication D des nœuds et la distance calculée par DV-Hop. En effet, si un nœud se trouve à un saut d'une ancre, la distance entre les deux est inférieure ou égale à D . De même, si l'écart est de deux sauts, la borne devient alors le double de la portée. Partant de cette remarque, *Constraint DV-Hop* propose de déterminer les coordonnées du mobile en observant les contraintes liées à la portée. Le système s'écrit alors à l'aide des équations 37 à 39.

$$\min \sum_{i=1}^M ((x_i - x_u)^2 + (y_i - y_u)^2 - d_i^2)^2 \quad (37)$$

$$\sqrt{(x_{v_j} - x_u)^2 + (y_{v_j} - y_u)^2} \leq D, L = 1 \quad (38)$$

$$D \leq \sqrt{(x_{v_j} - x_u)^2 + (y_{v_j} - y_u)^2} \leq DL, L \neq 1, j = 1, 2, \dots, M_1 \quad (39)$$

Cette proposition a été validée par une simulation de 100 nœuds, dispersés sur une surface de 50 m x 50 m. Les différentes expérimentations, avec un ratio d'ancres variables ou des modifications de la portée radio, ont indiqué la supériorité de CDV-Hop par rapport à DV-Hop.

Checkout DV-Hop

Dans la même optique d'amélioration de DV-Hop, le travail réalisé dans [45] distingue deux classes de mobiles : les mobiles de niveau 1 sont à portée d'au moins 3 balises et ceux de niveau 2 ne satisfont pas à cette condition. Un algorithme est proposé pour chacune des classes.

Pour les nœuds de niveau 1, la *Mid-Perpendicular Method* est utilisée. On considère les médiatrices des côtés du triangle formé par les ancres. Leur intersection désigne le centre du cercle circonscrit au triangle, point équidistant de tous les sommets. Les équations de deux médiatrices sont formées à partir des coordonnées de deux des ancres à portée.

Le travail réalisé représente également une amélioration de DV-Hop lorsque sont considérés les nœuds de niveau 2. Dans ce cas, *Checkout DV-Hop*, ajoute une étape de vérification (*checkout*). L'ancrage la plus proche en termes de nombre de sauts, notée A_{near} , est celle pour laquelle l'accumulation d'erreurs lors du calcul de la distance entre le mobile et cette ancre est la plus faible. L'écart entre A_{near} et le mobile servira alors de référence pour corriger la position estimée par DV-Hop (figure 13).

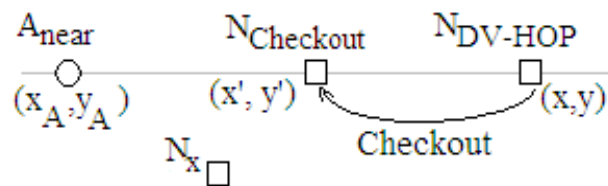


Figure 13 : Checkout DV-Hop

La figure 13 illustre l'effet de *Checkout DV-Hop*. La vraie position du nœud est N_x mais l'algorithme DV-Hop avait indiqué N_{DV-Hop} comme position probable. Connaissant la distance entre A_{near} et N_x , un nouveau point a été choisi par *Checkout DV-Hop*. Ce point, $N_{checkout}$, se trouve à la même distance d' A_{near} que N_x mais fait partie de la droite passant par A_{near} et N_{DV-Hop} .

La combinaison d'algorithmes a été comparée, en utilisant MATLAB, à d'autres associations de méthodes comme *Centroid + DV-Hop* ou *Convex Position Estimation (CPE)* [30] + DV-Hop. Avec différentes valeurs du ratio d'ancres, la nouvelle méthode présente de meilleures performances que ses concurrentes.

1.3.2.2.2. APIT

La méthode APIT [49] est basée sur le *Point-In-Triangulation test*. Ce test permet à un nœud de déterminer s'il se trouve ou non à l'intérieur d'un certain triangle. Dans sa version mathématique, le test stipule que tout déplacement d'un mobile situé à l'intérieur du triangle correspond à son rapprochement/éloignement d'au moins un des sommets du triangle (figure 14(a)). S'il se rapproche/s'éloigne de tous les sommets à la fois, le mobile est à l'extérieur du triangle (figure 14(b)).

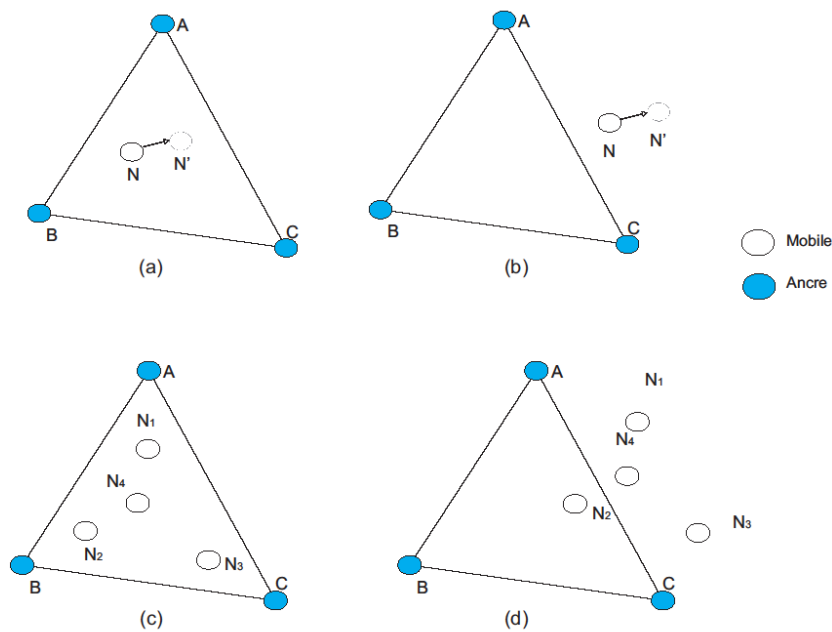


Figure 14 : Formulations géométrique et réseau de l'algorithme APIT

Ce test est adapté à un réseau en utilisant les relations entre les mobiles voisins du nœud d'intérêt et les ancres. Le mouvement du nœud n'étant pas forcément un paramètre contrôlé par l'application de localisation, les distances entre les voisins et les sommets sont utilisées pour émuler les déplacements. Ainsi, si tous les voisins d'un nœud N sont plus proches que N d'au moins un des sommets, N est à l'intérieur du triangle (figure 14(c)). Cette version approximative du PIT est sensible au placement des nœuds. Dans les cas où ce dernier est irrégulier et pour celui où au moins un des voisins se trouve à l'extérieur du triangle, une mauvaise décision est prise (figure 14(d)). APIT est dit *area-based* car son objectif est l'obtention d'une surface contenant probablement le point recherché.

Les nœuds mobiles collectent les trames balises émises (*beacons*) par les ancres. Une fois la liste d'ancres à portée construite, le mobile en sélectionne trois et détermine s'il se trouve à l'intérieur du triangle ainsi formé. Le test se répète jusqu'à l'épuisement des combinaisons ou

jusqu'à ce que l'intersection des triangles satisfasse le critère de taille de la surface de localisation obtenue. La surface obtenue correspond à la zone où se trouve très probablement le nœud. La position de ce dernier est alors associée au centre de gravité de cette surface.

1.3.2.2.3. *Centroid*

Les hypothèses de départ de *Centroid* [16] incluent une propagation sphérique isotrope ainsi qu'une portée identique pour toutes les stations.

La méthode se base sur un quadrillage de nœuds de référence émettant des *beacons* à un intervalle T sans collision. Selon une métrique de connectivité calculée localement (équation 40) et un seuil *Tresh*, le mobile décide de tenir compte d'un sous-ensemble de balises détectées. Les facteurs de cette caractéristique sont : $N_{recv}(i,t)$: le nombre de *beacons* émis par la balise i et reçus durant le temps t et $N_{sent}(i,t)$: le nombre de *beacons* émis par la balise i durant le temps t.

$$CM_i = \frac{N_{recv}(i,t)}{N_{sent}(i,t)} \times 100 \quad (40)$$

Les coordonnées des balises retenues sont utilisées pour calculer le centre de gravité de la zone, position que le nœud considèrera comme sienne. Sur la figure 15, une réalisation de *Centroid* est illustrée avec uniquement les balises retenues représentées par un point. La position estimée est représentée par une croix.

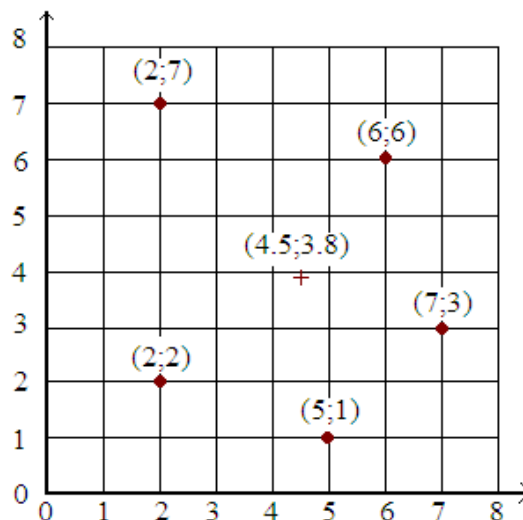


Figure 15 : Exemple de localisation avec *Centroid*

La validation par un prototype en environnement extérieur a impliqué quatre balises placées aux coins d'un carré de 10 m X 10 m et a mené à une erreur moyenne de 2m.

1.3.2.3. Conclusion

Les différentes propositions de localisation examinées peuvent être considérées comme représentatives des tendances observées pour la localisation dans les réseaux de capteurs sans fil. Les techniques *range-based*, par nature et grâce aux outils exploités conjointement, seront considérées comme plus exactes. Un compromis peut être trouvé entre justesse et complexité en combinant des solutions de ces deux familles, définissant ainsi des systèmes de localisation hybrides.

1.4. Pistes de recherche

Face à tant de propositions, il est bon d'examiner les apports et faiblesses de chacune en vue d'identifier la voie dans laquelle s'engager. Ceci est l'objectif des sections qui suivent où nous examinerons l'existant tour à tour en termes de performances et de coût matériel.

1.4.1. Exactitude de la localisation

Puisqu'il est question de localiser, le premier paramètre de comparaison des diverses solutions disponibles est la justesse de cette estimation. Autrement dit, quelle est en moyenne l'erreur observée sur les estimations de la position.

Cet aspect peut être étudié suivant deux angles : d'abord, une comparaison au niveau des moyens peut être réalisée. Les signaux d'entrée disponibles pour le *range-based* et le *range-free* seront alors comparés.

Le tableau 3 liste cinq moyens employés pour obtenir des distances et partant de là, la position. Les deux moyens les plus précis sont le temps de vol mesuré par UWB³⁰ et la phase du signal. Si ces deux technologies sont très convoitées, les difficultés d'accès au matériel constituent actuellement un frein aux projets de prototypage. En ce qui concerne l'UWB, peu de constructeurs proposent des *transceivers* implémentant la fonction de mesure de distance. La société Decawave [29] se prépare à une production de masse mais elle ne devrait être effective qu'à partir du second semestre de l'année 2013. D'ici là, le prix réel des composants reste prohibitif.

La mesure de phase, quant à elle, demande l'ajout de composants spécialisés et onéreux, ce qui nous ramène aux inconvénients des solutions nécessitant un ajout sur le matériel. De plus, elle doit être utilisée conjointement avec une autre technique en raison de l'ambiguïté de

³⁰ Ultra Wide Band

phase. Enfin, cette mesure exige qu'émetteur et récepteur soient en ligne de vue (LOS³¹) ; dans le cas contraire, le signal subit un déphasage supplémentaire qui fausse la mesure.

Tableau 3 : Comparaison des performances de diverses informations pertinentes pour la localisation

Caractéristique / Moyen	Ordre de la précision	Coût	Effet de l'environnement
Temps de vol UWB	Dizaine de centimètres	Synchronisation Accès au matériel	Atténuation hautes fréquences
Temps de vol Bande étroite	mètre	Accès au matériel Synchronisation	Multitrajet
Phase du signal	< longueur d'onde	Matériel	Ligne de vue
RSSI / Puissance	Mètre et variable dans le temps	Base de données	Humain
Angle d'incidence	-	Matériel	Multitrajet
Nombre de sauts	Portée	Inondation réseau Placement régulier	-

Si le temps de vol mesuré par des technologies radio à bande étroite est moins précis que celui obtenu par UWB en raison de sa sensibilité au multitrajet, en intérieur, ce temps de vol reste plus fiable que les indications de distances obtenues par RSSI. La variabilité temporelle de cette dernière caractéristique ainsi que sa vulnérabilité aux obstacles la rendent peu fiable pour la mesure de distance. Pour une exploitation profitable du RSSI, un *fingerprinting* incluant des mises à jour économiques de la base de données pour le réseau devrait être implémenté. En dépit de tout cela, cette caractéristique a encore de beaux jours devant elle car elle est accessible sur presque toutes les plateformes matérielles, contrairement au temps de vol.

Tout aussi facile à obtenir est le nombre de sauts. A l'origine utilisée pour le routage à vecteur de distance [82], cette information a fait son chemin dans les réseaux jusqu'à se créer une place dans le domaine de la localisation. Si l'une des failles tient au placement des nœuds, les hypothèses de symétrie des liens et d'absence de variabilité de la portée sont autrement plus restrictives car loin de la réalité. L'autre inconvénient de ce moyen est la nécessité d'inonder le réseau afin de parvenir à informer tous les nœuds.

³¹ *Line Of Sight*

1.4.2. Algorithmes décentralisés

Une autre piste de recherche concerne la décentralisation des algorithmes. Au-delà des possibilités inhérentes à l'information choisie, la structure de la solution algorithmique a un impact non négligeable sur la performance, en particulier sur la précision. Des outils comme le filtre de Kalman ou le filtre particulaire, en intégrant les sources d'erreur au calcul, peuvent limiter l'impact des perturbations sur l'estimation. Par contre, leur implémentation demande une plateforme relativement puissante. En effet, si on se limite au filtre particulaire, un certain nombre de points doivent être supportés par le système. Typiquement, quelques centaines de particules sont incluses dans le filtre. Les mises à jour de cet ensemble étant consommatrices en temps, dans bien des cas, un compromis est trouvé entre la population de points et l'ajout d'informations complémentaires (cartes de l'environnement...).

Le fait pour la solution de dépendre d'un serveur central de localisation implique la non disponibilité du service en cas de faille de cet équipement. Bien entendu, une ferme de serveurs peut être mise en place mais ceci entraîne d'autres problèmes, notamment la confidentialité, dans le cas où les serveurs sont mutualisés.

Si nous nous resituons dans le contexte de la localisation dans les réseaux de capteurs sans fil, nous ne pouvons négliger l'impact énergétique d'une solution basée sur un serveur central sur la survie de l'ensemble du réseau. Pour présenter cet aspect, nous nous baserons sur la topologie présentée à la figure 16. Considérons le cas où le mobile a exécuté les échanges permettant la mesure de distance avec ses voisins immédiats. Le calcul de position étant effectué au niveau du serveur, les résultats vont être acheminés à cette machine en suivant le chemin mis en évidence par un trait épais. Ici, par exemple, il est nécessaire de passer par deux nœuds spécifiques pour atteindre le serveur. Nous les rangerons dans la région C. La région B contiendra les nœuds permettant au mobile de rejoindre la région C. La zone A finalement englobera les ancres avec lesquelles le mobile a procédé à l'étape de *ranging*. Selon la position du mobile, les ancres incluses dans les régions A et B changeront. Ceci pourra permettre de ralentir l'épuisement des ressources énergétiques de ces nœuds. Par contre, en ce qui concerne les nœuds des différentes zones C_i , ils seront sollicités pour chaque demande de localisation émise par un mobile. Si on considère que le nombre de nœuds au voisinage du serveur, est faible, à moins de les alimenter en permanence, ils seront parmi les premiers éléments de réseau à cesser de fonctionner, entraînant l'arrêt du service de localisation.

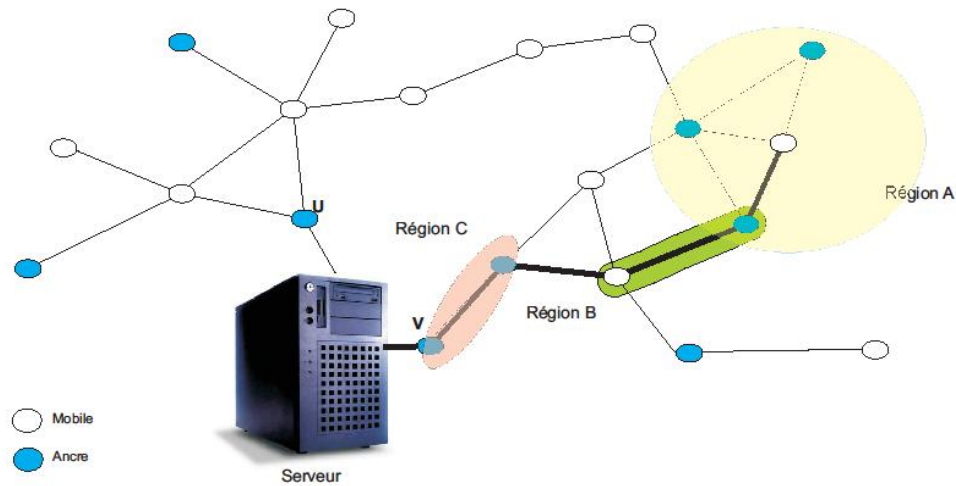


Figure 16 : Impact d'une solution centralisée sur l'ensemble du réseau

1.4.3. Algorithmes efficaces

Une dernière piste de recherche consiste à assurer que les algorithmes, tout en remplissant leurs objectifs, réclament le moins de ressources possibles. Effectivement, réaliser la localisation à travers un réseau de capteurs a un coût protocolaire et énergétique. Ces différents aspects sont liés car si un serveur est chargé de calculer la position, des messages doivent en effet être relayés à cette unité. Les écoutes et les émissions sur le canal radio liées à ces échanges génèrent une dépense d'énergie pour des nœuds qui ne sont pas forcément concernés par le processus. La charge peut être réduite en exigeant des nœuds capteurs la prise en charge du calcul. Afin de maintenir un coût raisonnable, plutôt que d'étendre les capacités des nœuds, il deviendra indispensable de développer des solutions robustes et simples pouvant être implémentées directement sur les modules du WSN.

1.5. Conclusion

Ce chapitre a introduit les différents types de données et les algorithmes communément rencontrés au cours de toute démarche de développement d'un système de localisation adapté à un réseau sans fil. Les contraintes imposées par les outils utilisés ont également été expliquées. En filigrane de cet exposé se distingue déjà l'orientation que nous donnerons à notre travail. En effet, dans l'objectif de doter un réseau de capteurs sans fil de la capacité de localiser les équipements mobiles avec un impact énergétique faible, nous aurons à définir un protocole de collecte adapté à l'information désirée et un algorithme capable de se satisfaire des ressources d'une plateforme réduite. La section suivante présentera les expérimentations ayant permis d'identifier le type de signal autour duquel notre solution sera construite.

ANALYSES EXPLORATOIRES POUR UNE AUTO-LOCALISATION *RANGE-BASED* EN INTERIEUR

Le chapitre 1 a offert une vue d'ensemble des travaux concernant la localisation pour les réseaux de capteurs. Les deux grandes familles de solutions, le range-based et le range-free, ont été introduites. De la même manière, les outils sur lesquels reposent ces implémentations ont été présentés.

Dans ce chapitre 2, nous décrirons les nœuds communicants autour desquels notre travail s'est articulé. Ce processus d'exploration abordera de nombreux aspects tels que les informations de localisation et l'énergie.

2.1.	Sélection d'un signal : RSSI ou temps de vol	57
2.1.1.	Plateformes utilisées pour l'exploration.....	57
2.1.2.	RSSI	60
2.1.3.	Temps de vol	65
2.1.4.	Conclusion.....	72
2.2.	Traitement des données	72
2.3.	Caractérisation énergétique	75
2.4.	Sources complémentaires	80
2.4.1.	Les ultrasons.....	80
2.4.2.	Infrarouge	81
2.4.3.	Instruments de navigation inertielle	81
2.5.	Conclusion.....	83

Après avoir examiné les différentes options disponibles en termes de signaux d'entrée, d'algorithme et de protocole, nous avons choisi de nous orienter vers les techniques *range-based* en exploitant l'un des deux signaux les plus faciles à obtenir, le RSSI et le TOF. En plus de permettre généralement d'atteindre une meilleure précision, nous avons retenu les méthodes *range-based* par opposition aux méthodes *range-free* ; cette dernière approche faisant également l'objet de d'une thèse réalisée durant la même période par notre collègue le Dr L. Gui [46] au sein de la même équipe. Naturellement, nous avons voulu intégrer ce service dans le type de réseaux sur lesquels nous travaillons habituellement, les réseaux de capteurs sans fil (§ 1.1.1). Nous tenions également à conserver le côté autonome de nos réseaux (auto-organisation, absence de point central, auto-cicatrisation ou *self-healing*³²...). La présence d'un serveur de localisation était donc hors de question. Ceci a motivé un choix dans la palette d'outils : les instances très consommatrices en calcul ou en mémoire, ont été mises de côté.

Finalement, nous considérons une solution au-dessus de la couche physique : la méthode est donc appelée à piloter la couche PHY, à récupérer ses données et générer une estimation de la position. Cette démarche a l'avantage d'être adaptée à nos compétences en réseaux et informatique, et permet un déploiement rapide de notre solution protocolaire et algorithmique sur des réseaux existants, par reprogrammation des nœuds, moyennant une couche PHY adéquate.

Après l'intégration de ces contraintes, notre objectif peut se formuler de la manière suivante : doter un nœud mobile d'un réseau de capteurs sans fil de la capacité de déterminer de manière autonome sa position dans un environnement intérieur, grâce aux mesures effectuées en temps réel avec des partenaires connaissant leurs positions respectives.

Dans la suite, nous nous intéresserons à la mise sur pied de cette méthode de localisation *range-based* centrée mobile. Cette analyse exploratoire est basée sur plusieurs plateformes matérielles, qui seront dans un premier temps présentées. Nous justifierons la sélection d'une information de localisation adaptée parmi les signaux disponibles, RSSI et TOF. Puis, nous discuterons des contraintes liées au traitement des données, de la relation entre la

³² Capacité du réseau à s'auto-réparer et à retourner à un fonctionnement normal après détection d'une faille

communication et la consommation énergétique avant de terminer par un bref panorama des éventuelles briques complémentaires.

2.1. Sélection d'un signal : RSSI ou temps de vol

L'un des avantages clés du RSSI est sa disponibilité : la plupart des plateformes proposent des méthodes permettant d'accéder à cette information à partir du logiciel. Le temps de vol est plus rarement implémenté en raison des exigences de précision des horloges pour un estampillage satisfaisant. Dans cette section, nous comparons ces deux informations à travers la métrologie réalisée sur deux plateformes réelles : Jennic et Nanotron. Ces dernières ont été retenues car elles permettent l'accès au temps de vol et au RSSI.

2.1.1. Plateformes utilisées pour l'exploration

Nous avons eu l'opportunité d'exploiter le matériel provenant de deux constructeurs différents durant notre étude. Deux plateformes ont été utilisées, elles sont présentées dans ce qui suit.

Dans un premier temps, nous avons effectué nos manipulations sur les nœuds du constructeur Jennic. La plateforme JN5148 [59] (figure 17) associe un microcontrôleur 32 bits à un *transceiver* opérant à 2,4GHz et compatible avec la norme IEEE 802.15.4 DSSS³³. Ce module dispose de 128Ko de ROM et 128Ko de RAM pour le stockage du code et des données, ainsi que de deux ports série.

³³ *Direct Sequence Spread Spectrum*



Figure 17 : Module radio-communicant Jennic

Dans un second temps, nous avons exploité les capacités de la plateforme désignée par « *AVRmodule* » [9] de Nanotron (figure 18). Du point de vue des communications sans fil, ce module implémente une couche physique plus récemment incluse dans le standard 802.15.4 [54]. Il s'agit de la technologie CSS opérant dans la bande de 2.4GHz. Le CSS est quelquefois qualifié de modulation de fréquence dans les hautes fréquences car sa caractéristique est de coder l'information par une variation de la fréquence. Deux symboles de base peuvent être ainsi définis : le *upchirp* (figure 19(a)) correspond à une variation d'une fréquence basse à une fréquence haute tandis que le *downchirp* (figure 19(b)) désigne la variation d'une fréquence haute à une fréquence basse. Finalement, la figure 19(c) représente le signal émis comme une succession de *chirps*.



Figure 18 : Module communicant Nanotron

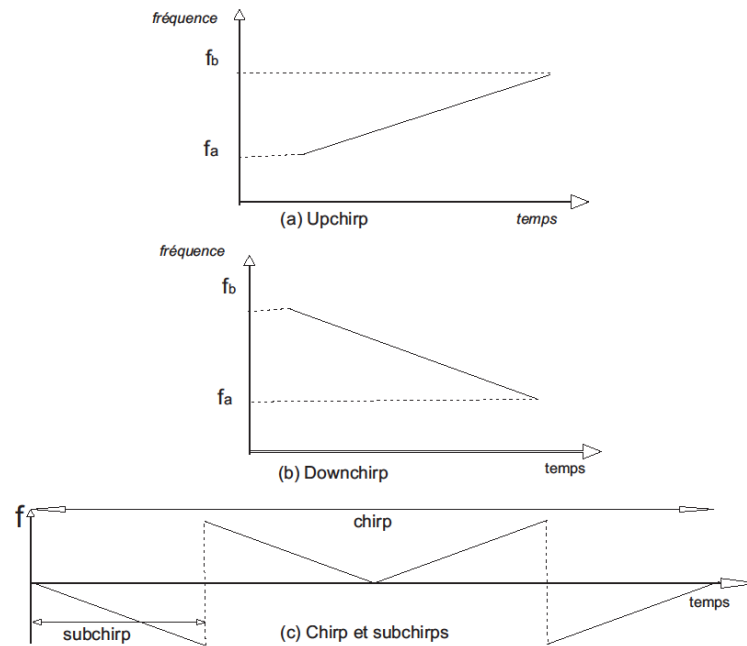


Figure 19 : Upchirps et downchirps dans la technologie CSS

A l'instar de la plateforme Jennic, l'*AVRmodule* supporte des liens à 250kb/s mais la fonction de *ranging* exige des communications à 1Mb/s. Cette couche physique est commandée par un microcontrôleur 8 bits ATMEL, l'ATMega644V [8]. Ce dernier dispose de 4ko de RAM, de 64ko de mémoire Flash et d'un port série.

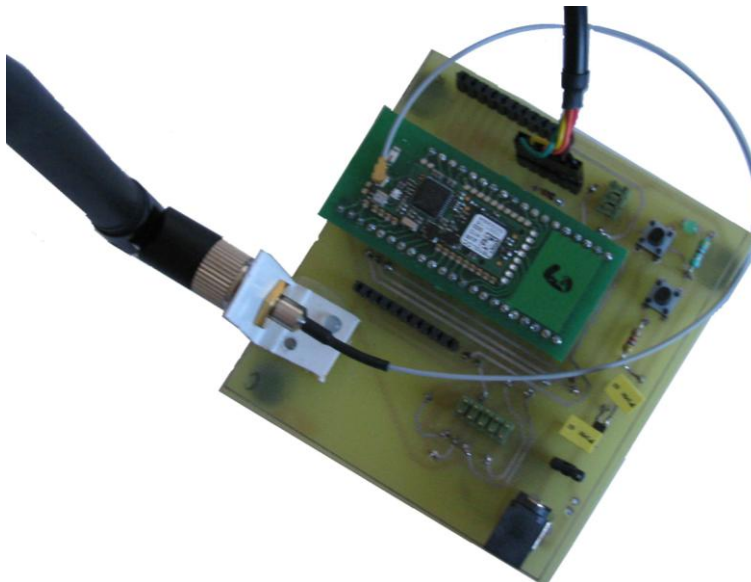


Figure 20 : Module radio-communicant Nanotron

La figure 20 présente la version finale du prototype dont la conception est présentée en annexe.

Les deux plateformes offrent l'accès au RSSI et au temps de vol. En ce qui concerne la seconde information, il est important de signaler que les deux plateformes n'utilisent pas des

Sélection d'un signal : RSSI ou temps de vol

canaux identiques pour les communications sans fil. Le DSSS se fonde sur des canaux de 2MHz tandis que le CSS exploite des bandes de 22MHz. L'utilisateur peut donc attendre du signal CSS une résolution temporelle plus fine, en vertu de la formule (équation 41). Cette dernière indique qu'augmenter la bande passante du signal améliore la finesse de la mesure de distance.

$$d = \frac{c}{BP} \quad (41)$$

Le tableau 4 résume les caractéristiques principales de nos deux plateformes matérielles. Il est intéressant de noter la faible taille de l'espace mémoire : cet aspect a une incidence sur les moyens pouvant être mis en œuvre.

Tableau 4 : Caractéristiques des deux plateformes matérielles disponibles

	Microcontrôleur/ microprocesseur	mémoire	Bande de fréquence (canal)	Débit
JN5148 <i>Jennic</i>	µP 32bit RISC	128ko ROM 128ko RAM	2.4GHz (5MHz)	250kb/s
<i>AVRmodule</i> <i>Nanotron</i>	ATMega644V	4ko RAM 64ko Flash	2.4GHz (22MHz)	250kb/s ou 1Mb/s

2.1.2. RSSI

La première piste envisagée a été celle du RSSI car, cette information étant disponible sur quasiment toutes les plateformes rencontrées, elle laissait pressentir la mise en place d'une solution transférable moyennant quelques petits ajustements pour pallier au caractère non standard du RSSI. Nous avons donc étudié le comportement de cette caractéristique en fonction de la distance.

Deux nœuds capteurs et un ordinateur portable sont utilisés pour récupérer les données. L'un des nœuds est désigné comme fixe et est relié à l'ordinateur par un port série. Ce module est chargé de transférer le RSSI de chacune des trames reçues au cours de l'expérience. Ces messages sont émis par un nœud mobile configuré en mode émission agressive. Ceci signifie que cet émetteur tente de saturer le médium en soumettant une nouvelle trame aussitôt que la transaction précédente a pris fin. Au fil du temps, le mobile s'éloigne du récepteur en

occupant des emplacements prédéfinis sur un axe reliant les deux partenaires de communication (figure 21). Les résultats sont présentés par les figure 22 et figure 23.

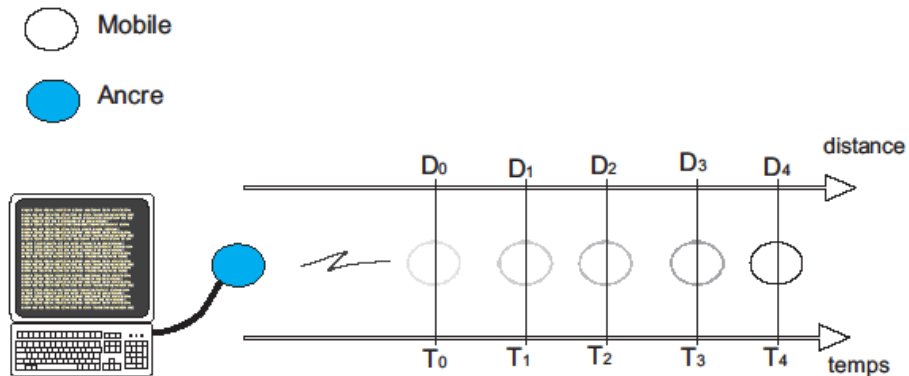


Figure 21 : Protocole de mesure de distance

La figure 22 a été générée à partir du matériel *Jennic*. Pour une même distance, elle montre les différentes valeurs de RSSI mesurées par le récepteur. De 6 à 10m, par exemple, les valeurs de RSSI reçues ne permettront pas de distinguer la position réelle du nœud émetteur.

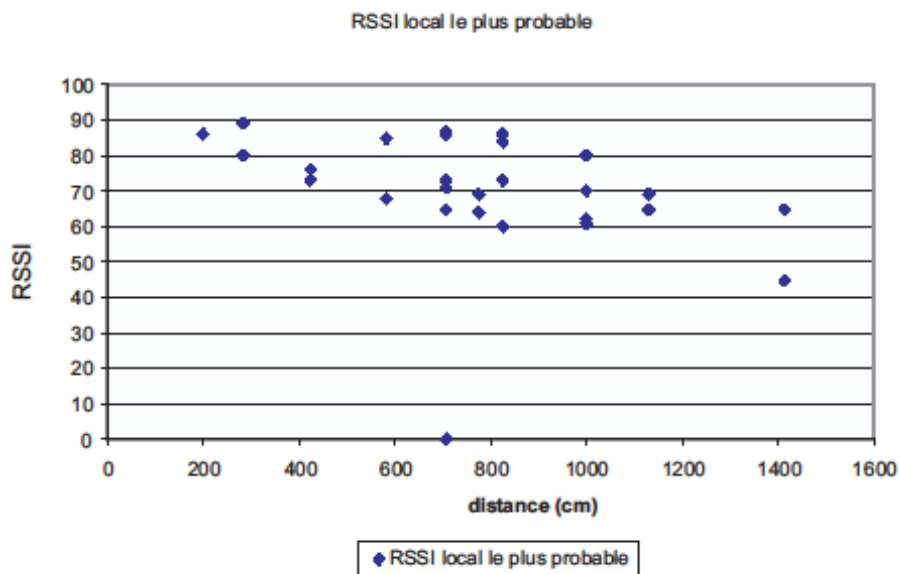


Figure 22 : Mesures de RSSI en ligne droite (Jennic)

Sur la figure 23, les résultats de la collecte de RSSI sur la plateforme Nanotron sont présentés. Les valeurs moyennes sont reliées par la courbe bleue. Autour de chacune de ces valeurs, l'intervalle occupé par les échantillons est également illustré. La situation est donc la même que pour la plateforme Jennic, le recouvrement des intervalles liés à différentes distances ne permettant pas de les distinguer.

Sélection d'un signal : RSSI ou temps de vol

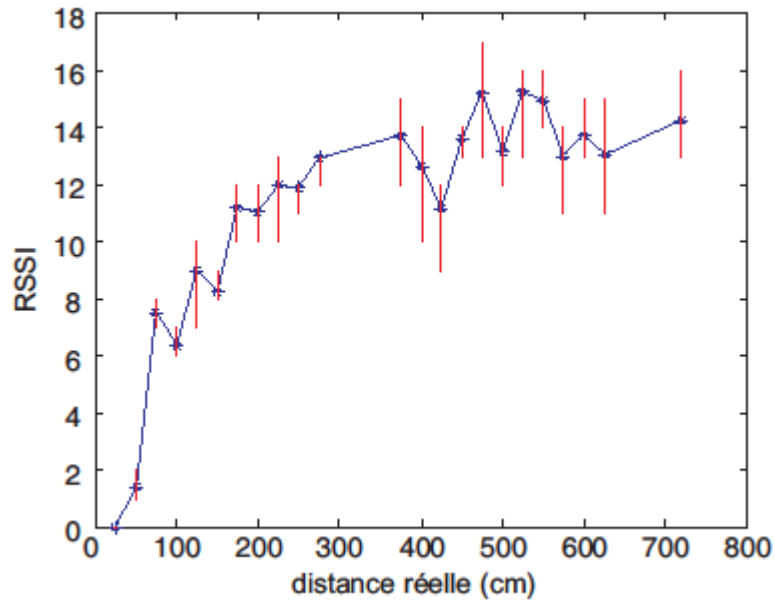


Figure 23 : Mesures de RSSI en ligne droite (Nanotron)

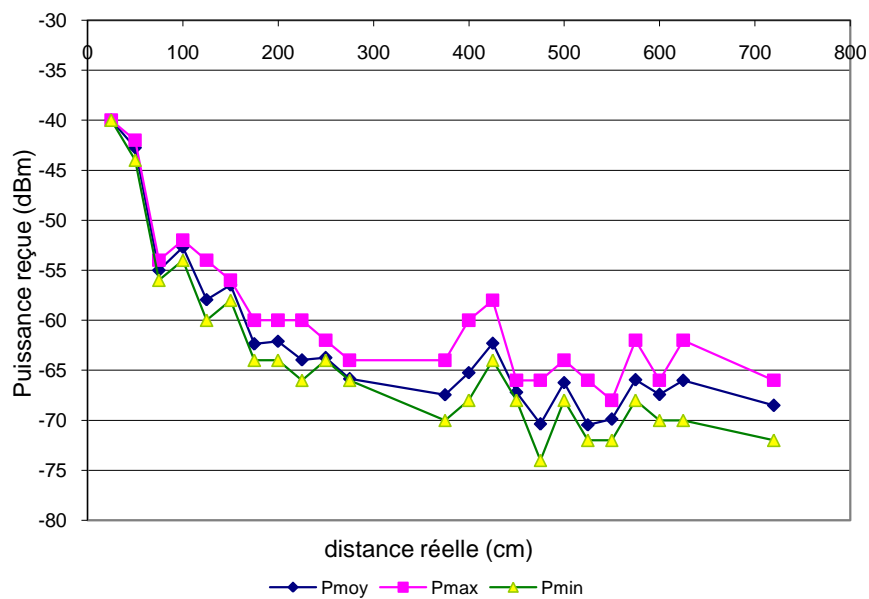


Figure 24 : Mesure de puissance reçue en ligne droite (Nanotron)

Il est bon toutefois de noter que contrairement à Jennic, la plateforme Nanotron propose d'entrée de jeu une formule permettant de transformer le RSSI en une valeur de puissance reçue (équation 42).

$$P(\text{dBm}) = -40\text{dBm} - 2\text{dBm} * \text{RSSI} \quad (42)$$

Les résultats de cette transformation sont illustrés par la figure 24.

Dans un second temps, le mobile et l'ancre ont été placés à une distance constante l'un de l'autre. On déplace le mobile selon une trajectoire représentant un arc de cercle centré sur l'ancre de collecte (figure 25). Durant ce parcours, l'orientation du mobile ne varie pas : le mouvement correspond donc à une suite de translations. La récupération des valeurs d'angles est facilitée par l'emploi d'un trépied disposant de cette graduation.

La figure 26 indique la variation du RSSI suivant l'angle. Nous avons tenu à étudier le comportement de cette caractéristique car en situation normale, les antennes ne seront pas parfaitement isotropes : l'impact de la directivité des antennes ne saurait donc être négligé. La figure 26 confirme l'existence de cet impact : alors que la distance reste constante, les valeurs mesurées changent avec l'angle. Par conséquent, un modèle simple ne peut être utilisé pour transformer le RSSI en distance car il varie avec au moins deux paramètres. Il faudrait déplacer la contrainte sur le matériel en imaginant les ancres capables d'indiquer l'orientation absolue de leurs antennes et les mobiles capable de reconstruire celle de la leur.

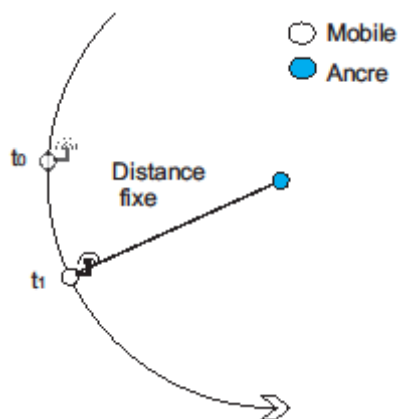


Figure 25 : Mesure du RSSI durant un déplacement circulaire

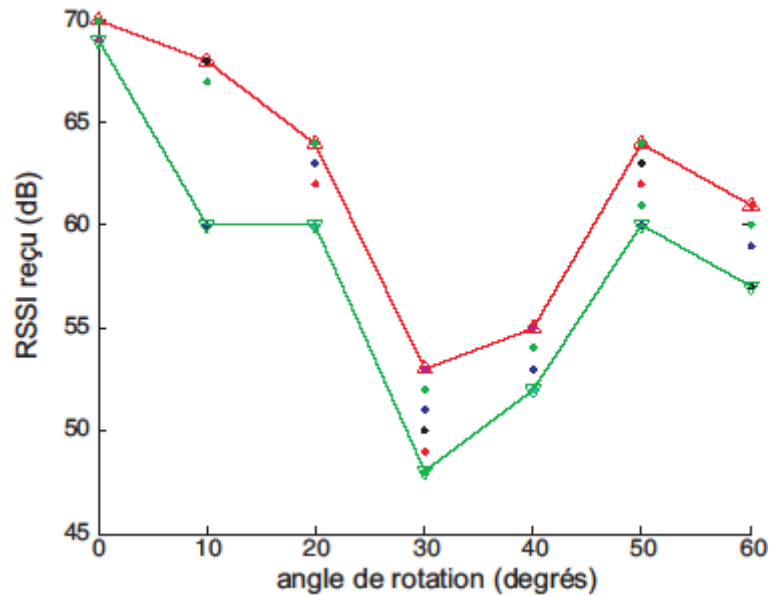


Figure 26 : Variation du RSSI avec l'angle de rotation (Jenic)

La synthèse de ces deux expériences suggère que la méthode la plus fidèle de représenter l'évolution du RSSI avec la position est une surface. En effet, si l'on raisonne sur une seule ancre, pour chaque direction, une courbe représentant les valeurs de RSSI attendues peut être générée (Figure 27). La combinaison de ces courbes produit une surface.

Malheureusement, même en négligeant l'aspect aléatoire, stocker les données dans ce format serait très coûteux. De plus, une surface devrait être créée par rapport à chaque nœud de référence : la recherche de la position estimée demanderait de croiser des résultats sur plusieurs de ces représentations. L'approche existante de *fingerprinting* en des points de référence permet de combiner taille raisonnable des données et collecte pertinente des informations.

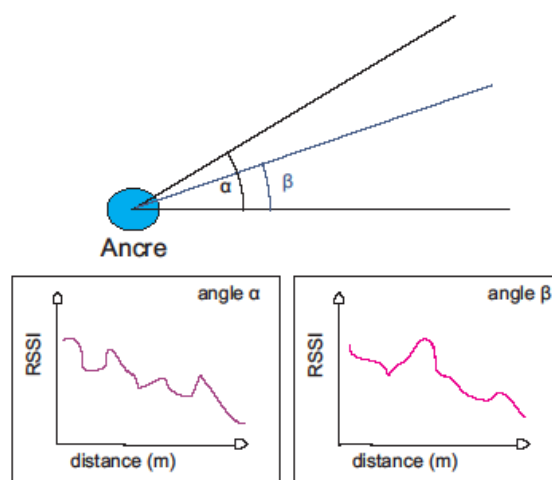


Figure 27 : Exemple de courbes de RSSI pour des directions différentes

En dépit du caractère prometteur de cette méthode, nous ne nous sommes pas attardés sur le RSSI car son exploitation ne saurait être que centralisée : en effet, la faible taille mémoire des modules (tableau 4) interdit la mise en place d'une base de données réellement utile à leur niveau.

2.1.3. Temps de vol

Les plateformes utilisées [9][59] permettent l'accès au RSSI mais aussi à une estimation du temps de vol. Le matériel ainsi que le protocole de mesure n'étant pas les mêmes dans les deux cas, les performances sont très différentes.

Dans un premier temps, nous avons caractérisé les performances de la plateforme [59] basée sur le DSSS. En intérieur, les résultats ont été moins que satisfaisants : en effet, la documentation fournie par le constructeur indiquait une impossibilité de réaliser des mesures en intérieur en dessous de 6m. Nous avons donc procédé à une première étude du comportement de l'outil en extérieur. Deux nœuds sont à nouveau utilisés. L'un est fixe et placé sur un trépied. Le second est mobile et s'éloigne du premier au cours de l'expérience. Le mobile communique avec un ordinateur portable à travers un port USB. A chaque pression sur une touche, il exécute 10 échanges protocolaires de type TWR (*Two-Way Ranging*) avec le nœud fixe et affiche les mesures sur une console. Ces données sont recueillies sur une distance variant de 4 à 100m. La figure 28 illustre ces résultats. En moyenne, pour chaque distance, les valeurs sont dispersées sur un intervalle de 8m, l'écart-type variant entre 1.5m et 3m. Les résultats de l'expérience constituaient un encadrement de la valeur réelle de distance mais l'incertitude restait trop élevée pour une utilisation en intérieur. Nous avons donc décidé de travailler sur une seconde plateforme.

Sélection d'un signal : RSSI ou temps de vol

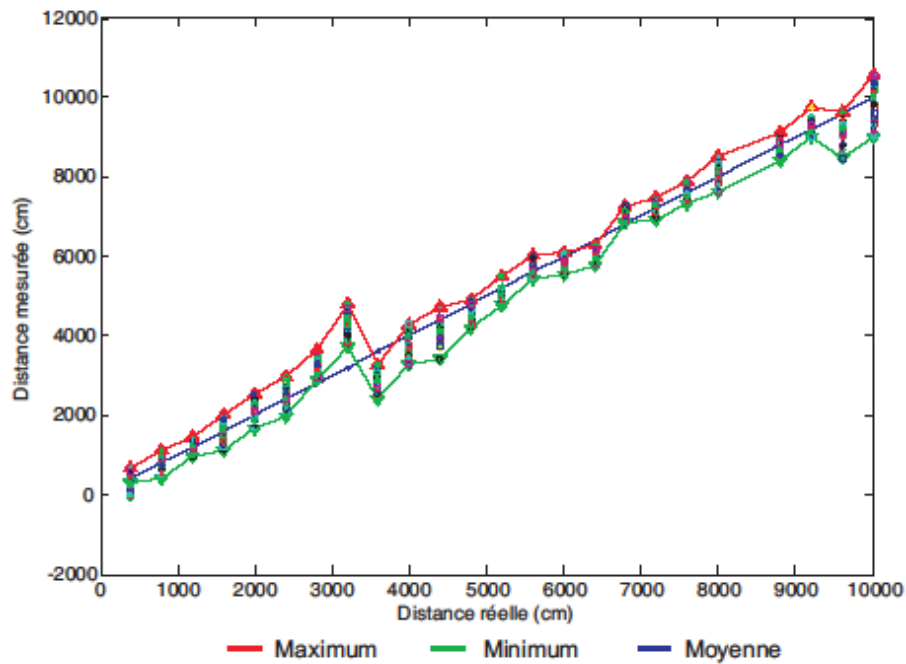


Figure 28 : Caractérisation de la mesure de distance (signal DSSS, en extérieur)

La promesse des mesures basées sur la technologie CSS tenait aux caractéristiques du signal et au protocole implémenté. Comme nous l'avons vu précédemment, SDS-TWR a été élaboré en vue de limiter l'impact des différences d'horloge sur les résultats. De plus, l'augmentation de la largeur de bande de 5 à 22MHz laissait entrevoir une amélioration de la résolution en distance. L'examen de la méthode de calcul utilisée par la plateforme Nanotron a mis en évidence l'utilisation de données relatives à la phase pour corriger l'information temporelle obtenue. En effet, à l'issue des échanges de messages de SDS-TWR, l'application fournie traite les informations collectées par la formule donnée par l'équation 43.

$$T = \frac{\frac{TxRespTime}{4} - \frac{PhaseOffsetACK + PhaseOffsetData}{32}}{2} \quad (43)$$

$$- \frac{(TOAOffsetMeanData + TOAOffsetMeanACK) \times \frac{LOD20}{48}}{2} - rangingConstant$$

Avec

TxRespTime : le temps écoulé entre l'envoi d'une trame et la réception de l'acquiescement

PhaseOffsetAck : le déphasage d'un acquiescement reçu

PhaseOffsetData : le déphasage de la trame de données reçue

TOAOffsetMeanData : le décalage temporel associé à une réception de données

TOAOffsetMeanAck : le décalage temporel associé à une réception d'acquiescement

Ces différents facteurs correspondent à des tops d'horloge et sont ramenés au domaine temporel en les multipliant par la période de l'horloge d'origine, d'où les différents termes (1/4, 1/32 et Lod20/48).

En examinant les valeurs des paramètres de la formule durant les mesures, nous avons noté que les seuls éléments variant de manière significative étaient TxGateOff, RxGateOff, TxUCSum et RxUCSum.

La figure 29 indique les instants de capture de ces valeurs.

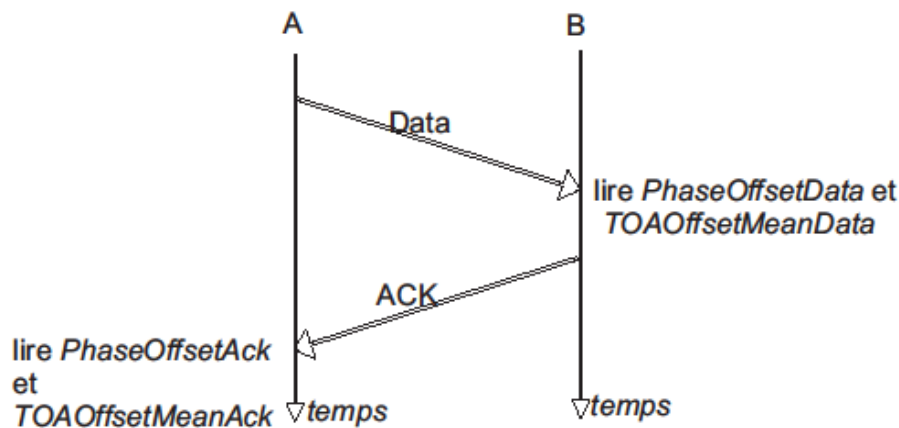


Figure 29 : Accès aux registres

Un zoom sur la figure 29 mène à la figure 30 où sont représentées les durées d'émission et de réception. L'instant où la trame arrive réellement au niveau de la couche physique ne correspond pas exactement à celui où les valeurs sont stockées dans les registres, que ce soit pour la réception de la trame ou celle de l'acquittement. En désignant par t_i les instants réels, nous pouvons écrire les équations 44 et 45 :

$$t_4 = t_1 + \text{TxRespTime} - e \quad (44)$$

avec $e = f(\text{PhaseOffsetAck}, \text{TOAOffsetMeanAck})$

$$t_3 - T_{ACK} = t_2 + f(\text{PhaseOffsetMeanData}, \text{TOAOffsetMeanData}) = t_2' \quad (45)$$

La plateforme utilisée charge la couche physique d'envoyer automatiquement l'acquittement sans notifier les couches supérieures. T_{ACK} devient donc une constante englobant la durée des contrôles sur la trame reçue, le temps de retournement de la radio et la circulation du signal vers l'antenne.

Sélection d'un signal : RSSI ou temps de vol

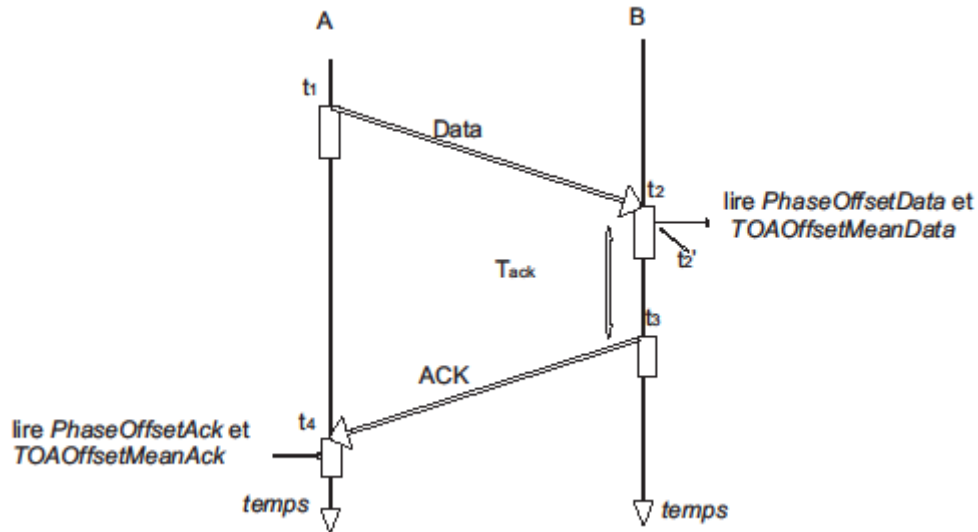


Figure 30 : Accès aux registres et instants d'émission/réception réels

Le temps de vol peut être calculé par l'équation 46 :

$$\begin{aligned}
 2T &= (t_4 - t_1) - (t_3 - t_2) \\
 &= TxRespTime - f(PhaseOffsetMeanAck, TOAOffsetMeanAck) \\
 &\quad - (f(PhaseOffsetMeanData, TOAOffsetMeanData) + T_{ACK}) \\
 &= TxRespTime - f(PhaseOffsetMeanAck, TOAOffsetMeanAck) \\
 &\quad - f(PhaseOffsetMeanData, TOAOffsetMeanData) - T_{ACK}
 \end{aligned} \tag{46}$$

Le terme $f(PhaseOffsetMeanData, TOAOffsetMeanData)$ permet de corriger l'estimation de $t_3 - t_2$: une fois la trame complète reçue, l'ACK est généré et émis. Par conséquent, T_{ACK} ne peut seul représenter la durée en question : la couche physique se synchronise pour pouvoir reconnaître les symboles. Pour reconstituer le vrai début de la trame, elle utilise les informations de l'offset de phase et de l'offset temporel. Un raisonnement identique explique la nécessité de corriger $TxRespTime$ à l'aide de $f(PhaseOffsetMeanAck, TOAOffsetMeanAck)$.

Nous avons donc mené des expériences de caractérisation, cette fois-ci en *intérieur*. Une démarche similaire de collecte a été suivie : un nœud fixe a été placé sur une table et branché à un ordinateur via le port USB (figure 31).

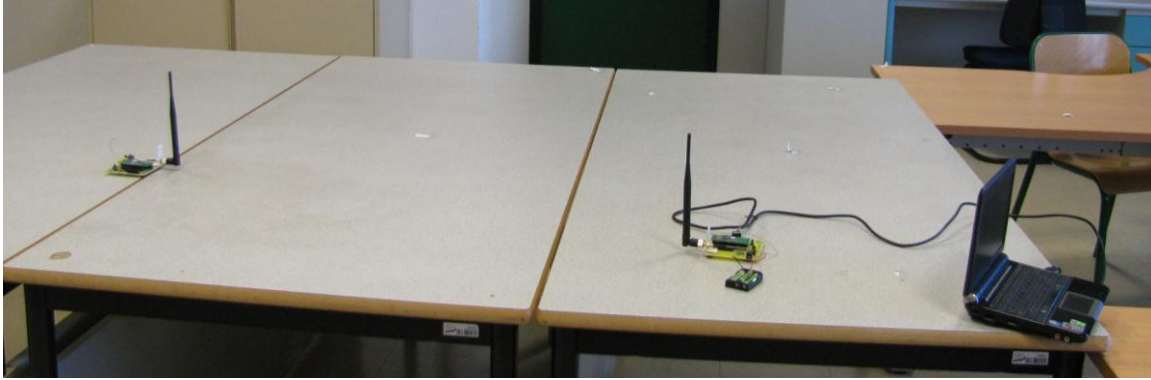


Figure 31 : Contexte de test de la mesure de distance avec les nœuds *Nanotron*

Le mobile s'est éloigné au fur et à mesure du nœud fixe et pour chaque position occupée, un certain nombre d'échantillons a été récupéré. Sur la figure 32, les résultats bruts sont représentés par les courbes bleues et magenta. La courbe cyan résulte de la moyenne des deux précédentes tandis que la droite verte correspond à la réalité donc à la première bissectrice en vert sur la figure 32. Cette dernière représente la fonction de mesure parfaite et est donnée par l'équation 47.

$$y_{mesure} = f(x_{réalité}) = x_{réalité} \quad (47)$$

La première remarque concernera l'intervalle contenant les échantillons : une nette amélioration peut être observée puisque sa largeur a été réduite à 1.16m. Par contre, on notera l'écart significatif entre les résultats de la mesure et les valeurs attendues. L'exploitation de cet outil exigera donc une phase de correction de l'information afin de la ramener à proximité des valeurs exactes.

Cette première mesure a été exécutée sur une distance allant jusqu'à 4.5m. Cette limite a été imposée par les caractéristiques du module. En effet, l'antenne par défaut offrait de piètres performances. Nous avons procédé à une modification du code ayant pour objet l'activation de la mesure du RSSI. Ceci a eu un impact positif sur la portée : en effet, afin de rendre possible la récupération du RSSI, nous avons dû activer le circuit de CAG³⁴, désactivé par défaut. Le résultat est illustré en figure 33.

³⁴ Contrôle Automatique de Gain

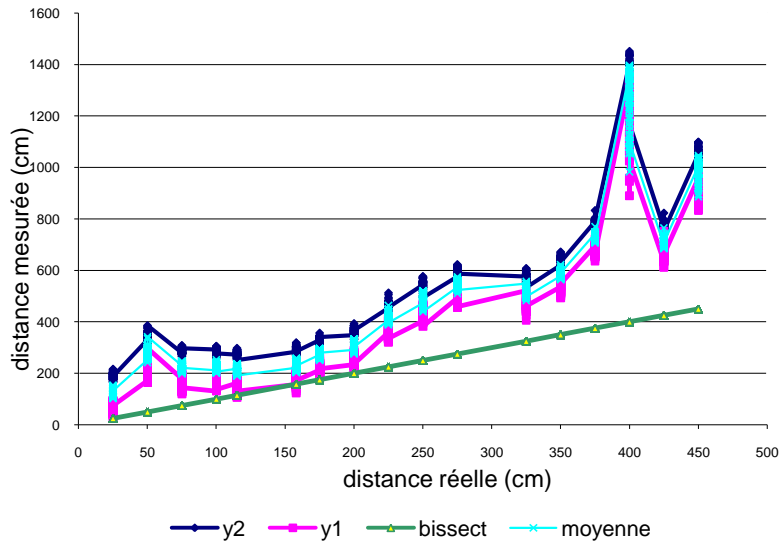


Figure 32 : Mesures de distance par temps de vol à l'aide d'une plateforme CSS

La figure 34 résume les expériences menées dans la chambre anéchoïque³⁵ de l'IUT de Blagnac (figure 35). En raison de la taille de la salle, nous n'avons pas pu effectuer des mesures au-delà de 4m. On observe notamment ici que les estimations sont meilleures car inférieures à celles obtenues en environnement ordinaire et bruité. Si l'influence du multitrajet a pu être éliminée grâce à la salle anéchoïque, une erreur non négligeable demeure. Elle est due alors aux limitations du matériel : d'un côté, la couche physique dispose d'une horloge de 32MHz pour l'estampillage des trames lorsque les acquittements sont activés. D'un autre côté, la largeur de bande d'un signal CSS étant de 22MHz, la résolution en distance possible avec cette technologie reste limitée (cf. équation 9). Les effets néfastes de ces deux caractéristiques ont été limités par le constructeur durant le processus de transformation en distances des valeurs capturées. A la section 4.2.6.3, nous proposerons une méthode de correction supplémentaire permettant de ramener les estimations à proximité des valeurs réelles.

³⁵ Ne produisant pas de réflexions du signal grâce à un traitement des surfaces

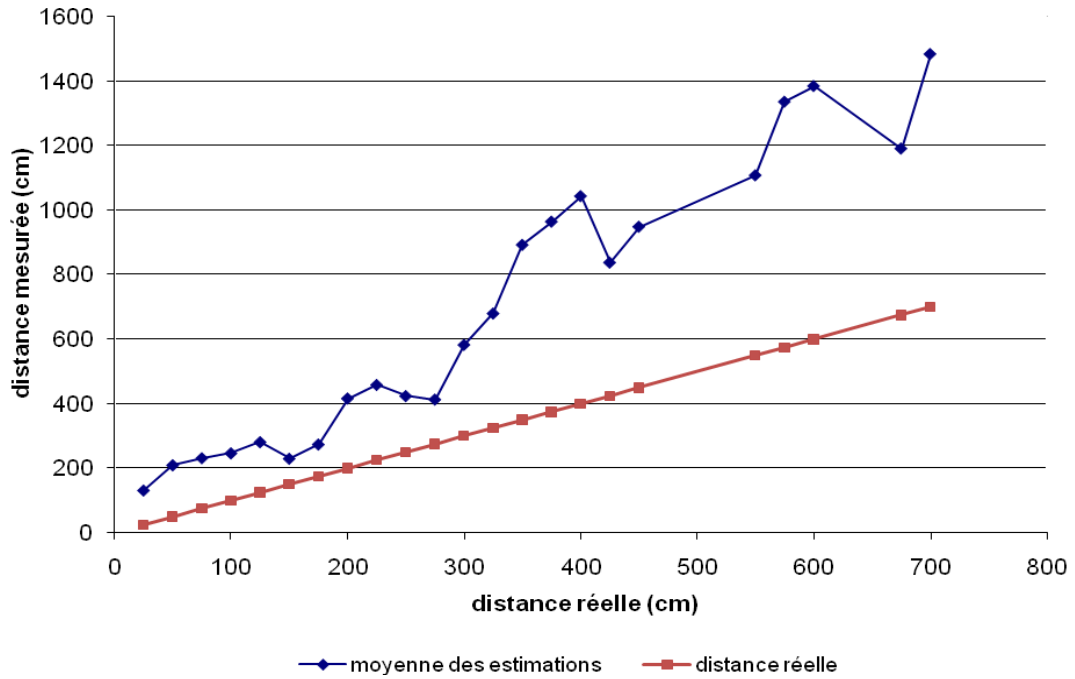


Figure 33 : Mesures de distance par temps de vol (AVR Module)

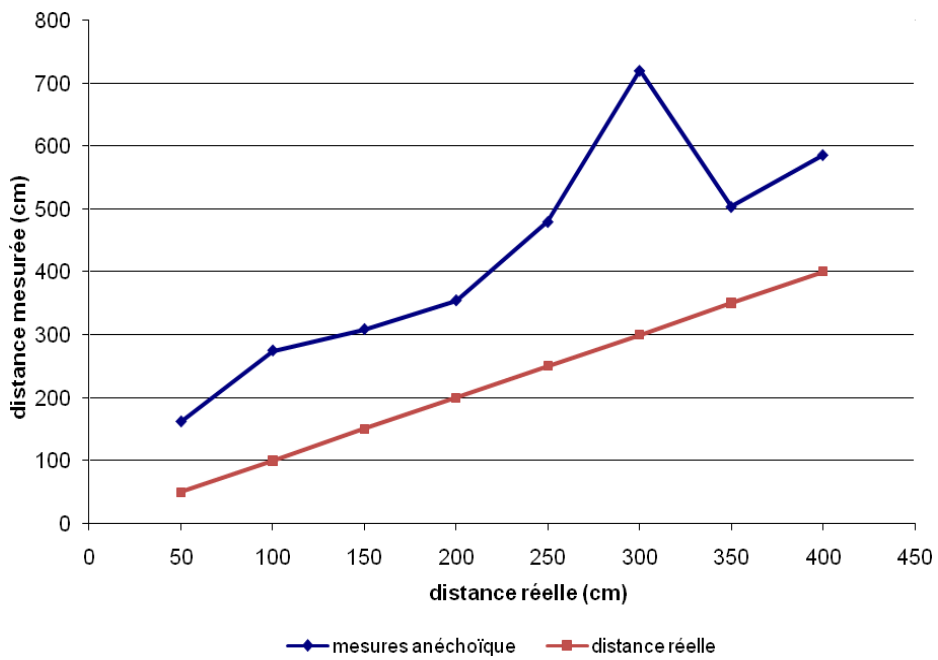


Figure 34 : Mesures de distance par temps de vol en salle anéchoïque (AVR Module)

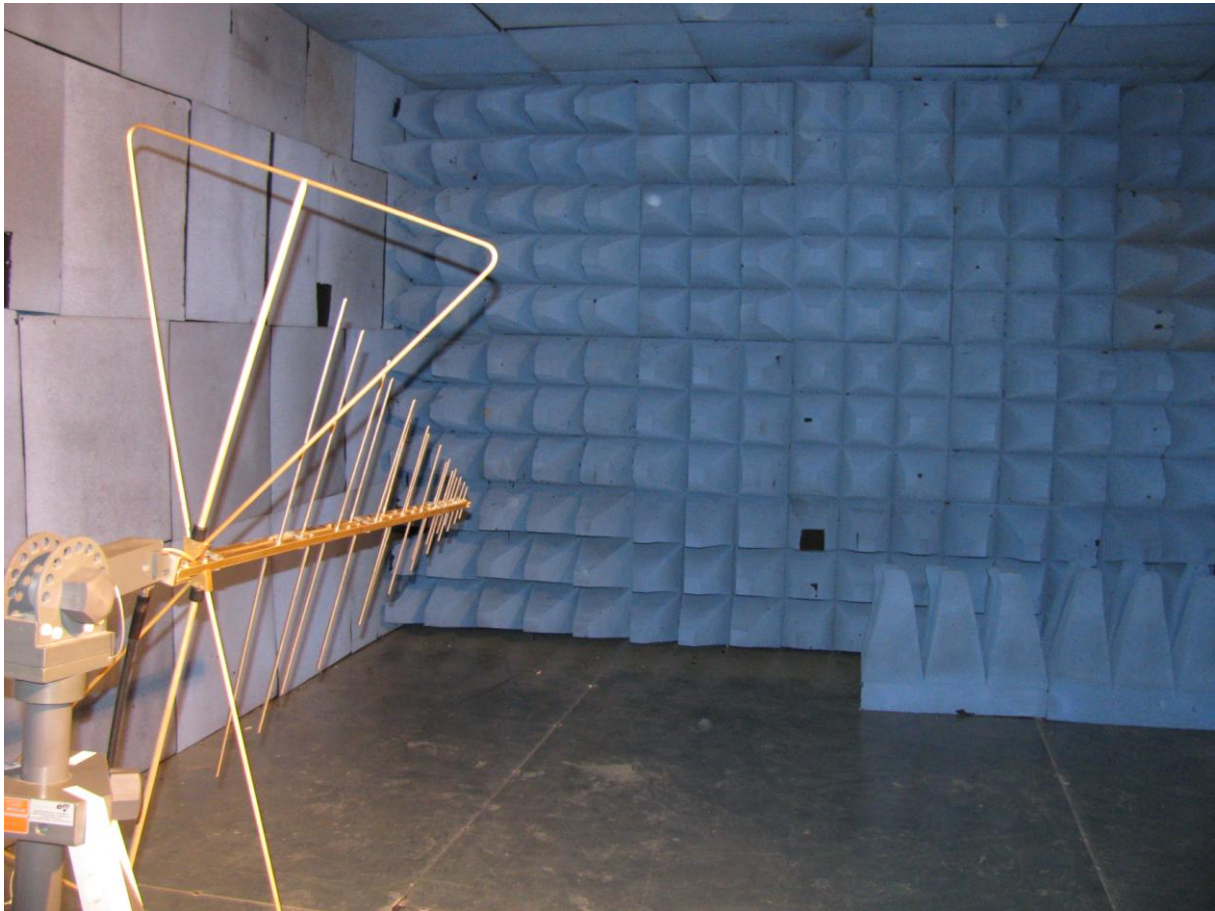


Figure 35 : Salle anéchoïque de l'IUT de Blagnac

2.1.4. Conclusion

Nous avons choisi de travailler avec la technologie de communication CSS et d'effectuer nos mesures par le protocole SDS-TWR car cette combinaison offrait les meilleurs résultats et était plus appropriée pour la localisation en intérieur. Dans le futur, nous espérons avoir accès à des modules intégrant un *transceiver* UWB capable de mesures de temps de vol. Nous y reviendrons dans la présentation de nos perspectives à l'issue de la thèse.

2.2. Traitement des données

Une fois le type de données identifié, nous nous sommes tournés vers l'algorithme à mettre en place pour calculer la position. L'une des expériences significatives que nous avons eu à mener implique le logiciel Matlab.

Notre démarche peut être décrite par la figure 36. Un traitement algorithmique combine les informations concernant le réseau (les positions des balises) et les mesures collectées par le mobile. Bien que nous ayons choisi de nous concentrer sur le temps de vol, le RSSI a également été inclus dans ce traitement, moyennant un modèle de transformation en distance.

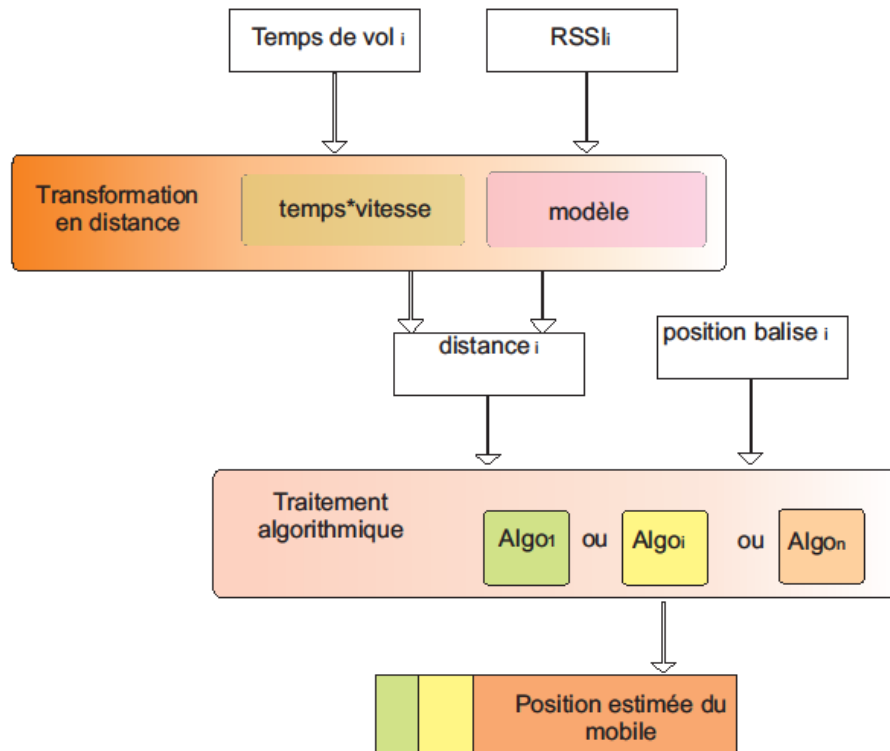


Figure 36 : Procédure de traitement des données

En ce qui concerne le bloc de traitement, nous avons opté pour une démarche similaire à celle présentée dans [68] : une fonction de coût a été définie (équation 48) et l'algorithme a pour objectif de la minimiser. Pour chaque ancre $B_i(x_i, y_i)$, le coût s'exprime sous la forme de l'équation 49, où c représente la vitesse du signal et σ_i la confiance à accorder aux estimations issues de l'ancre i . Dans notre cas, nous avons choisi de considérer l'écart-type des estimations de distance générées à partir de l'ancre i comme la valeur de σ_i .

$$G = \sum_{i=1}^n \alpha_i^2 * f_i(x, y)^2 \text{ avec } \alpha_i^2 = \frac{1}{\sigma_i^2} \quad (48)$$

$$f_i(x, y) = c\Delta t_i - \sqrt{(y - y_i)^2 + (x - x_i)^2} \quad (49)$$

La résolution de cette équation non linéaire fait appel à la fonction *fsolve* de Matlab. Cette dernière requiert l'expression mathématique du problème mais également un point initial ou *seed* à partir duquel la solution finale est recherchée par itérations successives.

Deux problématiques ont donc été rencontrées, et tout d'abord, celle évidente de la sélection du point de départ ou *seed*. En effet, un *seed* trop éloigné de la vraie position peut conduire à une non-convergence vers la solution. L'utilisation d'un point fixe préconfiguré ne peut donc être envisagée. D'autre part, le mobile, en choisissant le *seed*, est incapable de déterminer si celui-ci est proche ou non de la position recherchée. Il est donc indispensable de

doter l'instance de calcul des moyens de déterminer son *seed* en fonction des informations disponibles.

Diverses pistes ont été examinées en ce sens :

- **Le barycentre des ancrés à portée** Dans ce cas, seules les positions des ancrés utilisées pour la mesure de distance sont nécessaires au mobile. Ces informations sont accessibles à travers les trames *beacons* émises périodiquement par les ancrés. Si les mêmes ancrés sont utilisés pour une diversité de positions du mobile, nous nous retrouvons dans la situation où un point fixe est utilisé : pour certains points, l'algorithme échoue et les coordonnées ne peuvent être estimées.
- **Le barycentre pondéré des ancrés à portée** Pour éviter d'avoir un point fixe, nous permettons aux ancrés d'avoir une influence sur le *seed* en fonction de leur RSSI détecté par le mobile. La valeur maximale du RSSI ainsi que le nombre d'occurrences de cette valeur sont utilisées pour ajuster le *seed*. Nous pouvons ici noter que nous utilisons conjointement plusieurs signaux, comme imaginé au début de la thèse.
- **La balise la plus proche** Ce cas correspond à une simplification du précédent : la balise dont le RSSI maximal sur la période est détecté le plus souvent est considérée comme la plus proche du mobile

L'application de ces diverses stratégies nous a permis d'observer que l'exploitation du barycentre pondéré représentait le meilleur choix. Il présente toutefois deux faiblesses :

- **Performance** L'erreur moyenne est de 6.8m et l'écart-type de 4m. Ces données correspondent aux valeurs prédites par le constructeur des modules utilisés durant cette période. Ces résultats ont motivé la recherche d'un autre algorithme. Ce dernier devrait être capable d'utiliser en entrée des données bien moins que parfaites et tout de même produire un résultat. D'autre part, il ne devait réclamer aucune connaissance étendue du réseau, les données récupérées à travers les *beacons* devant suffire.
- **Durée d'exécution** Utiliser Matlab pour la résolution du problème implique une exécution centralisée donc la possibilité d'exploiter un maximum de données a posteriori sans se soucier des contraintes de taille de mémoire. En dépit du fait que cette démarche ne coïncide pas avec notre objectif, elle nous a permis de glaner des informations intéressantes. En particulier, l'importance de la durée d'exécution

de l'algorithme. Dans notre cas et selon la proximité entre le *seed* et la solution, le calcul durait entre 30 s et 1 minute, sur un poste de travail de type Pentium disposant de 2Go de RAM et cadencé à 3GHz. Transposer le processus sur des plateformes à base de microcontrôleurs, tournant à quelques MHz et disposant de moins de 1 Mo de RAM était un défi que nous ne tenions pas à relever sans apporter des modifications significatives à l'algorithme de localisation.

Cet ensemble d'expérimentation, s'il ne nous a pas permis d'identifier dans l'immédiat l'algorithme désiré, a tout de même contribué à dessiner le contour de notre solution. Tout d'abord, la quantité de données requises doit être minimale. En effet, la faible taille mémoire de nos cibles ne les autorise pas à manipuler de grandes quantités d'information. D'un autre côté, donner accès à tous les mobiles à une connaissance étendue du réseau implique une circulation intense de trames qui, du point de vue de la localisation, s'apparenteraient à de la signalisation. La consommation en bande passante induite conduirait à considérer la localisation non comme un service mais comme l'un des clients principaux du réseau.

D'autre part, les opérations nécessaires à la détermination de la position devront s'exprimer le plus simplement possible et mener à une exécution rapide. Ainsi l'impact de l'implémentation sur la disponibilité des ressources des nœuds capteurs sera minimisé.

2.3. Caractérisation énergétique

Toute proposition de solution à base de réseaux de capteurs sans fil doit tenir compte de l'impact de cette mise en œuvre sur la ressource énergétique du nœud. Dans la suite, nous aurons à comparer l'effet de l'utilisation du protocole proposé sur notre réseau. En ce sens, cette section s'attachera à étudier la relation entre les différentes phases de communication et la consommation. Nous nous baserons sur les documents techniques décrivant le microcontrôleur ATmega644V [8] et le *transceiver* NATR51 [74] et intégrerons les résultats de nos propres mesures.

A chaque état du système correspond un niveau d'énergie. Dans notre étude, nous considérerons principalement l'énergie consommée par le *transceiver* car il est sans conteste le sous-système le plus énergivore de notre module communicant. Ce *transceiver* peut emprunter plusieurs états ; pour chacun d'eux, la notice du constructeur indique le délai d'activation à partir de l'état spécifié. Le tableau 5 décrit ces options.

Tableau 5 : Etats du *transceiver* et consommation

Etat	Description	Courant	Délai d'activation vers un état cible
<i>PwrDownModeFull</i>	Sommeil profond	< 2.5µA	Cible : <i>PowerUp</i>
<i>PwrDownModePad</i>		< 600µA	Entre 1 et 32ms (programmable) + activation du microcontrôleur
<i>PowerUp</i>		< 700µA	Cible : <i>standBy</i> < 5ms
<i>StandBy</i>	La distribution de l'horloge bande de base est désactivée. Pour passer en mode <i>Ready</i> , cette distribution est enclenchée. Le délai dépend du débit de l'interface SPI	< 2.5mA	Cible : <i>Ready</i> 6µs (sans reconfiguration de la radio, pour un débit SPI de 4Mb/s)
<i>Ready</i>	Dans cet état, le µC peut demander à tout moment d'émettre ou de lui remettre une trame.	< 4mA	--

Sur notre plateforme, le µC fonctionne sous 2.5V, ce qui implique une vitesse d'environ 8MHz. Nous pourrions donc adopter le délai spécifié de 6µs pour le transit de l'état *StandBy* à l'état *Ready*.

L'information manquante concerne l'évolution de la consommation durant les émissions et les réceptions. Pour étudier cet aspect de la question, nous avons ajouté à notre carte de développement un module optionnel permettant de placer une résistance en série avec la charge. La mesure de la tension à ses bornes nous indiquera le courant la traversant. Ce courant représentera le courant global alimentant principalement le microcontrôleur et le *transceiver*.

Pour réaliser ces mesures, nous avons utilisé un *AVRmodule* configuré pour émettre une trame dont la *payload* est de 43 octets à destination d'un nœud absent. Ceci provoquera l'attente en réception pendant *aMACACKWaitDuration* puis un certain nombre de réémissions, 3 dans notre cas. On notera qu'ici, les nœuds radio accèdent directement au médium : le protocole d'accès au médium CSMA-CA n'est pas utilisé. Un second signal est

capturé simultanément. Le front montant de ce dernier indique la fin du transfert des données à travers le bus *SPI* et le front descendant la notification d'échec de la transaction à la couche MAC. Les figure 37, 38 et 39 présentent les différents aspects du résultat.

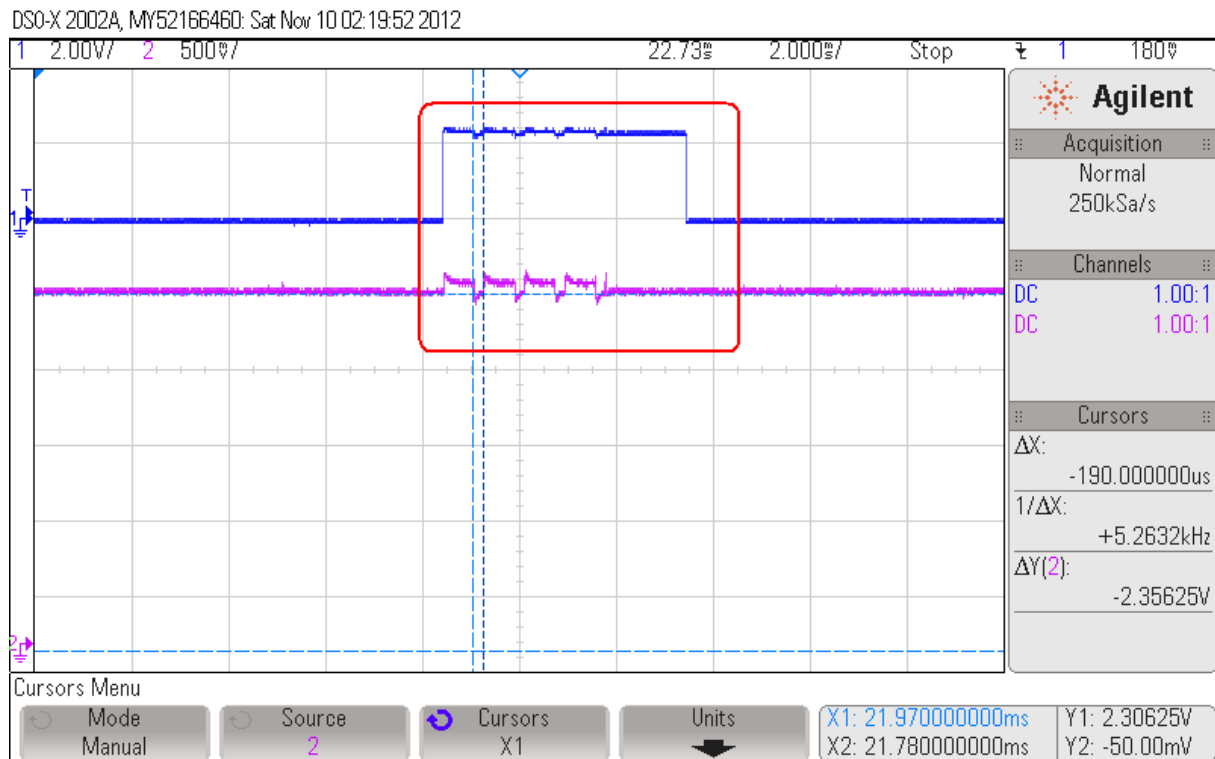


Figure 37 : Evolution de la tension aux bornes de la charge durant la phase de transmission

La figure 37 illustre la séquence globale d'exécution. La courbe bleue représente le signal de référence tandis que la tension aux bornes de la charge, V_{charge} , correspond à la courbe magenta. On observe la chose suivante : une première transmission a lieu et est suivie d'une écoute du médium dans l'attente de l'ACK. Dans la première phase, V_{charge} augmente jusqu'à la valeur V_{TX} , puis diminue jusqu'à la valeur V_{RX} . Ces valeurs sont obtenues à partir des figure 38 et de la figure 39.

La tension d'alimentation de notre dispositif étant fixée à 2,5V, le courant traversant notre circuit variera en fonction de l'état (émission, réception, éteint). Le système d'équations 50 représente la situation.

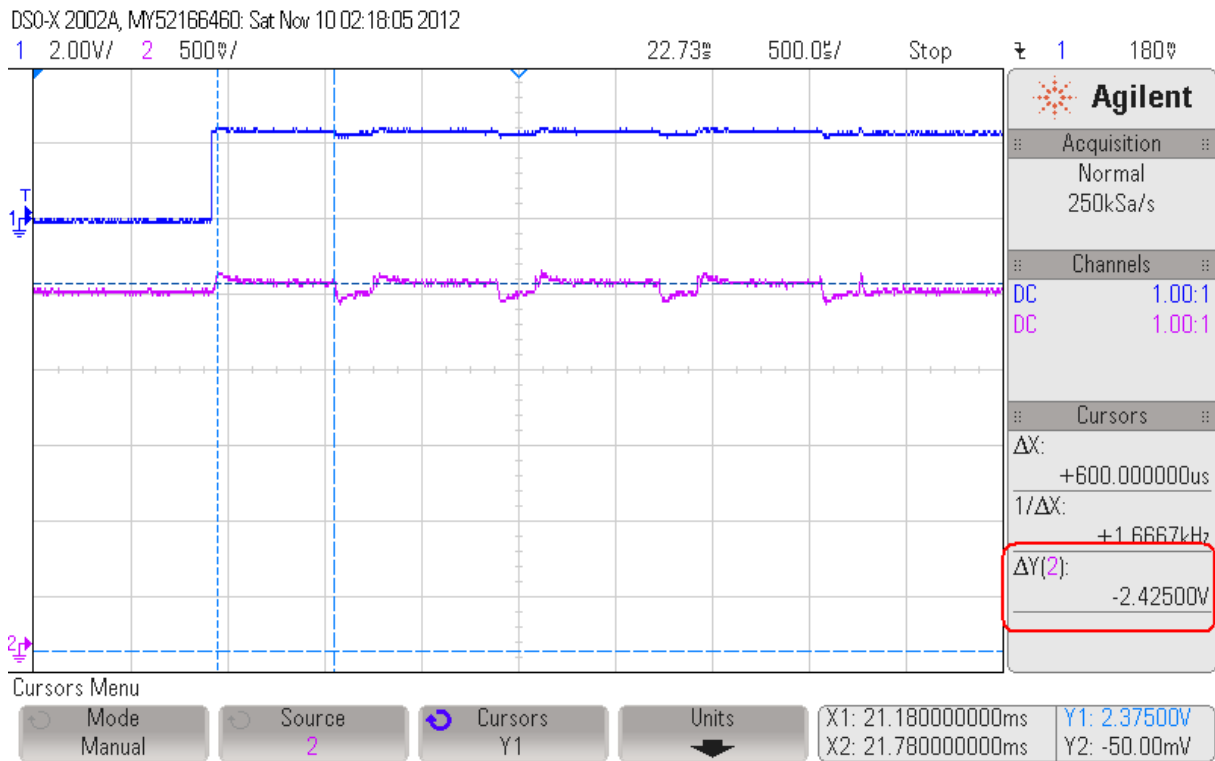


Figure 38 : Mesure de la tension aux bornes de la charge durant la transmission

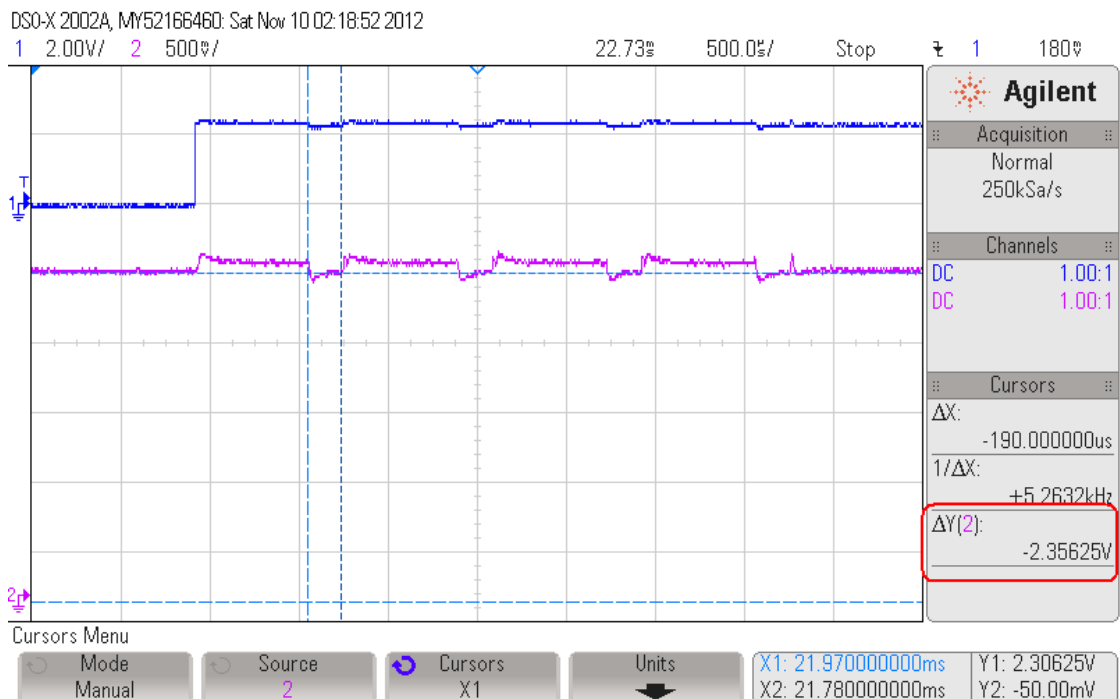


Figure 39 : Mesure de la tension aux bornes de la charge durant la réception

$$\begin{aligned}
 V_{totale} &= V_R + V_{charge} \\
 V_R &= RI \\
 I_{TX} &= \frac{V_{totale} - V_{TX}}{R}, I_{RX} = \frac{V_{totale} - V_{RX}}{R}
 \end{aligned}
 \tag{50}$$

Les valeurs de R, V_{TX} , V_{RX} et V_{totale} sont extraites des zones encadrées de la figure 38 et de la figure 39 et sont rappelées dans le tableau 6. Elles permettent d'aboutir aux courants des états de transmission et de réception donnés dans le tableau 7. A la différence de technologies telles que IEEE 802.11, nous pouvons noter que la réception est ici plus énergivore que l'émission. Ceci est dû au fait que pour les réseaux 802.15.4, la puissance d'émission est réduite : plutôt que d'utiliser, à l'instar de 802.11, des liens couvrant de grandes distances, cette technologie s'appuie sur le relais de trames pour acheminer les messages, dans des réseaux multisauts. A complexités analogues des récepteurs, la transmission est donc moins consommatrice que la réception.

Tableau 6 : Paramètres de la consommation durant la période active de l'émetteur-récepteur

Paramètre	Valeur
Résistance R	4,1 ohms
V_{totale}	2,5 V
V_{TX}	2,425 V
V_{RX}	2,356 V

Tableau 7 : Courants traversant l'équipement durant la phase active de l'émetteur-récepteur

Courant par état	Valeur
Réception/écoute du médium : I_{RX}	35,12 mA
Transmission : I_{TX}	18,3 mA

A partir de ces données, nous formulerons la consommation énergétique induite par l'exécution d'un protocole donné comme (équation 51)

$$\text{énergie}_{consommée} = V_{TX} \times I_{TX} \times \text{durée}_{TX} + V_{RX} \times I_{RX} \times \text{durée}_{RX}
 \tag{51}$$

Dans la suite, nous utiliserons ce modèle pour évaluer la consommation de notre protocole de localisation

2.4. Sources complémentaires

La section précédente a introduit certaines des expérimentations auxquelles nous nous sommes livrés au cours de la phase de découverte et d'études exploratoires préliminaires. En particulier, face à l'impossibilité d'appliquer un algorithme aux données brutes, nous nous sommes intéressés à la possibilité de les corriger à partir de sources complémentaires. La première de ces informations était le RSSI mais des outils tels ceux utilisés par la navigation inertielle pourraient se révéler utiles. Dans cette section, nous présentons ces différents éléments ainsi que les avantages et les contraintes liés à leur utilisation.

2.4.1. Les ultrasons

Le son peut également être utilisé pour la mesure de distance. Comparativement à l'onde électromagnétique, il a l'avantage de se propager à des vitesses largement inférieures dans des milieux similaires. Cette caractéristique implique que la résolution de la mesure temporelle peut être plus faible dans le cadre de l'utilisation des ultrasons tout en maintenant une précision appréciable. Par exemple, soient c_1 et c_2 les vitesses respectives de l'onde électromagnétique et de l'ultrason dans l'air. Pour mesurer une distance d à l'aide de ces signaux, les horloges sont soumises à une contrainte de précision dépendant de c_i . Dans l'équation 52, pour une distance donnée, le terme Δt_i représente la durée minimale pouvant être mesurée avec cette horloge. Cette durée est fonction de la fréquence de l'horloge mais aussi des performances du mécanisme d'accès aux estampilles temporelles. Plus simplement, si deux accès consécutifs sont nécessairement distants de plus de Δt_i , Δt_i n'est pas mesurable.

$$d = c_1 \Delta t_1 = c_2 \Delta t_2 \quad (52)$$

Si nous revenons à la comparaison de l'ultrason et de l'onde électromagnétique, la mesure d'1m requiert une horloge capable de mesurer précisément 3,33ns tandis que l'ultrason se satisfait, dans l'air à 25°C, d'une horloge précise à la milliseconde.

Bien qu'il soit plus facile d'identifier une plateforme matérielle pouvant supporter l'usage de l'ultrason, nous avons décidé de ne pas suivre cette piste pour des raisons d'encombrement, de coût et d'énergie. En effet, le transducteur permettant de manipuler en émission et en réception des ultrasons n'est pas par défaut disponible sur les nœuds : la nécessité d'ajouter un élément ayant longtemps été le fer de lance des méthodes range-free, nous avons jugé bon de nous confiner aux moyens disponibles au sein même de la plateforme.

D'autre part, la présence de cette unité supplémentaire implique une augmentation de la consommation énergétique ainsi que du coût de l'unité.

2.4.2. Les signaux infrarouges

Le signal infrarouge a également été associé à l'ultrason en vue d'améliorer la performance de la localisation. Dans [70], les deux signaux sont émis simultanément mais la différence entre leurs vitesses de propagation respective implique un décalage dans leurs arrivées. La réception du premier signal provoque alors le démarrage d'un compteur cadencé à 70kHz, stoppé par le message ultrasonique.

Si les résultats de cette association sont prometteurs, l'utilisation de l'infrarouge soulève certains problèmes [61]. Les surfaces polies réfléchissent ce signal mais il ne traverse pas les objets opaques. D'un côté, cela facilite la sécurisation des communications mais de l'autre, cette limitation physique de la portée implique une installation dense au niveau de l'infrastructure pour assurer la communication en mobilité. Bien qu'il soit possible de configurer les émetteurs et récepteurs de manière à exploiter les réflexions du signal, la distorsion du signal due aux versions multiples reçues et l'impact de l'éclairage ambiant sur la communication ne peuvent être négligés. Finalement, étant un type d'onde lumineuse, ce signal souffre d'un très fort *path loss* et ne convient que pour des liens à faible portée. Consacrer plus de puissance à l'émission peut permettre de contrebalancer cet effet mais ce choix ne convient pas pour des réseaux de mobiles et à plus forte raison pour les réseaux de capteurs sans fil.

2.4.3. Les instruments de navigation inertielle

L'idéal, lors du *design* d'un protocole pour les réseaux de capteurs sans fil, consiste à accomplir sa tâche de manière efficiente, en utilisant le moins de ressources possible. Dans cette perspective, la localisation par navigation inertielle constitue une solution de choix. Son principe peut être énoncé de la manière suivante : connaissant la position initiale et les caractéristiques du déplacement d'un mobile, il est possible de reconstituer sa trajectoire à partir des équations du mouvement et par conséquent de déterminer sa position. Par exemple, sur la figure 40, la position initiale correspond à P0. Connaissant l'orientation du mobile et la taille de son déplacement, une estimation P1 de la nouvelle position peut être générée.

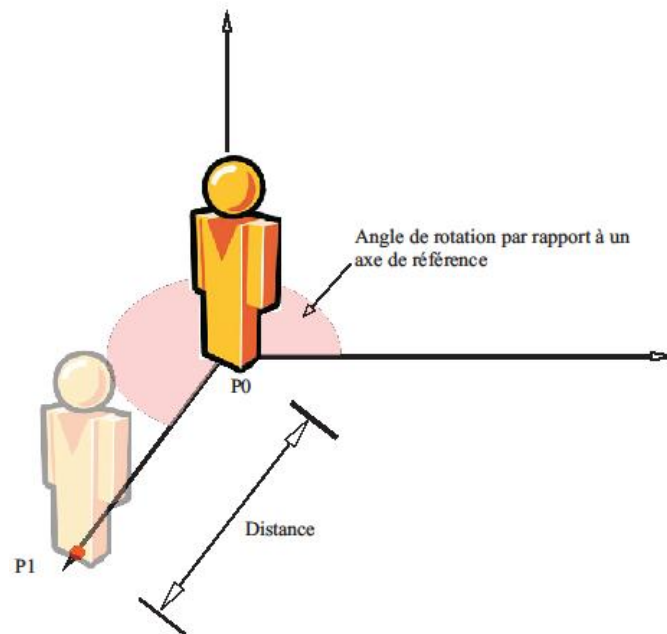


Figure 40 : Prédiction de la prochaine position basée sur l'orientation et le déplacement

Une école se base sur des statistiques concernant le mouvement du mobile tandis qu'une autre préconise l'utilisation de capteurs adaptés en vue de récolter des informations plus précises. Typiquement, des accéléromètres et des gyroscopes sont exploités dans cet objectif.

L'accéléromètre, comme son nom l'indique, permet de mesurer l'accélération subie par l'objet suivant un, deux ou trois axes. En combinant les intégrations successives de cette information, il est possible de remonter à la vitesse ainsi qu'à la distance parcourue. Dans le cas de la localisation de personnes, moyennant la connaissance de la démarche du mobile, ces mêmes accéléromètres peuvent servir à détecter les impacts liés aux pas et ainsi à mesurer la distance à travers le nombre de pas effectués.

L'exemple de la figure 40 a montré que la distance ne suffit pas pour localiser à partir des équations du mouvement. Il est nécessaire d'adjoindre au système un capteur capable de déterminer l'orientation du mobile. Deux pistes sont disponibles : la première est le compas. Cet instrument retourne directement la direction mais est perturbé par la présence de champs magnétiques parasites ou de masses métalliques dans l'environnement. Le second capteur est le gyroscope. Ce dernier diffère du compas en ceci qu'une intégration continue est nécessaire pour estimer l'angle de rotation. En effet, le gyroscope mesure la vitesse de rotation angulaire : la prise en compte du temps permet d'identifier l'orientation finale du mobile.

Exploiter les équations du mouvement accéléré non rectiligne et non uniforme permettrait de se dispenser de mesures radio. Ceci est vrai, mais seulement dans un monde idéal. En effet, ces capteurs sont tous sujets à des imperfections de conception qui se traduisent par des

erreurs de mesures. Ces dernières peuvent être dues au biais qui est lui-même fonction de la température, à la non linéarité de la réponse, au bruit de mesure. Pour toutes ces raisons, les systèmes utilisant la navigation inertielle ont besoin d'un moyen additionnel de correction des données afin de compenser la dérive par accumulation d'erreurs. Cette correction provient généralement d'un réseau sans fil, par exemple le GPS. Cette fusion de données peut par exemple reposer sur un filtre particulière.

2.5. Conclusion

Au terme de cette phase d'exploration, nous avons jeté notre dévolu sur le temps de vol en raison de la stabilité de cette information. Notre plateforme basée sur la technologie CSS sera mise à contribution car sa performance est bien meilleure. Bien que nous ayons souhaité initialement intégrer différentes sources de données à notre solution, les contraintes associées nous ont porté à renoncer à ajouter à nos nœuds d'autres sources d'information. Néanmoins, une donnée supplémentaire comme le RSSI, facilement accessible à travers le *transceiver*, pourra être exploitée au niveau protocolaire.

Forts de ceci, nous pouvons partir à la recherche du protocole ainsi que de l'algorithme adaptés. Ces derniers seront introduits au chapitre suivant.

PROPOSITION D'UN SYSTEME DE LOCALISATION ORIGINAL ET EFFICIENT

La phase d'exploration liée à notre thèse a été présentée dans les pages précédentes. La sélection d'une plateforme ainsi que d'un signal en vertu de ses qualités pour la mesure de distance ont été exposées. Le chapitre 3 sera par conséquent consacré à l'algorithme et au protocole qui ont été bâtis pour l'exploitation d'une couche physique capable de ranging par une couche MAC basée sur les propositions de la norme IEEE 802.15.4.

3.1.	Proposition de protocole.....	87
3.1.1.	Négociation du <i>star interval</i>	87
3.1.2.	SSDS-TWR : échanges et formules	95
3.1.3.	PSDS-TWR : échanges et formules	98
3.1.4.	Comparaison de la charge réseau des deux protocoles	101
3.1.5.	Comparaison de la consommation énergétique induite par les deux protocoles	103
3.1.5.1.	Caractérisation temporelle de SSDS-TWR	103
3.1.5.2.	Caractérisation temporelle de PSDS-TWR	105
3.1.5.3.	Analyse de la consommation.....	107
3.1.6.	Conclusion.....	111
3.2.	Proposition d'un algorithme de localisation.....	111
3.2.1.	<i>iRingLA</i>	112
3.2.1.1.	Déroulement	112
3.2.1.2.	Durée d'exécution	117
3.2.1.3.	Complexité et optimisations	118
3.2.1.4.	Ordonnancement des anneaux.....	120
3.3.	Transposition sur d'autres technologies	126

Le principe « diviser pour mieux régner » peut s’appliquer à la localisation. En effet, le système à créer peut se représenter comme la structure illustrée en Figure 41. La solution est répartie entre trois tâches principales qui sont : la sélection des nœuds de référence par rapport auxquels la mesure de distance devra être réalisée, la collecte des estimations de ces distances et le calcul de la position. Au terme de la première tâche, le mobile et les ancres conviennent d’un instant dédié à la mesure. Une fois ce rendez-vous fixé avec les meilleures ancres, un protocole adapté à l’information à recueillir est mis en place. Comme nous l’avons vu dans l’état de l’art au chapitre 1.2, ce protocole de *ranging* dépend de la caractéristique physique à récupérer. Il décidera du comportement des ancres et du mobile de manière à obtenir une approximation de la distance. Ces estimations seront injectées au bloc de calcul qui, grâce à un algorithme efficace, déterminera la position la plus probable du nœud mobile.

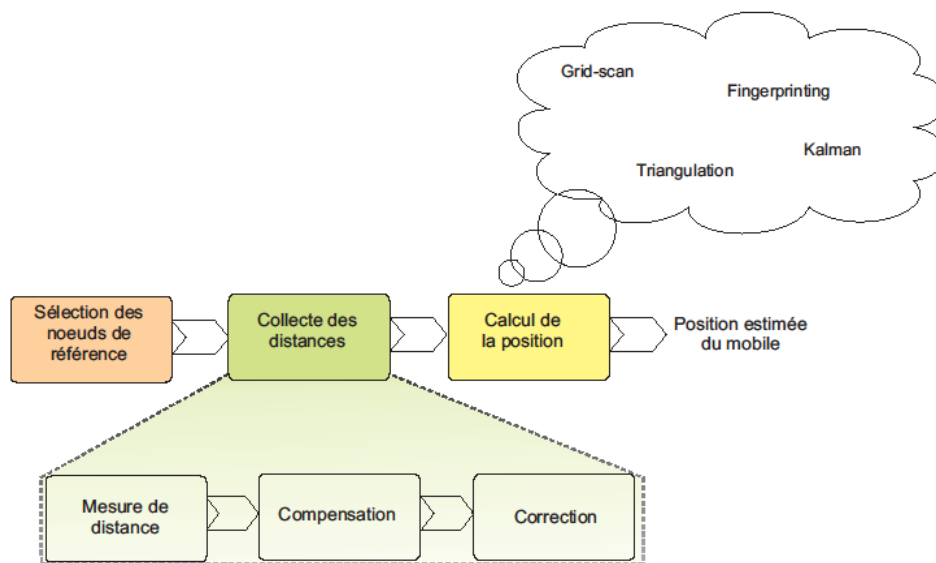


Figure 41 : structure d'un système de localisation

Ce chapitre 3 décrira notre système de localisation : il s’agit d’une solution *range-based* dont l’originalité consiste à être peu consommatrice de ressources CPU tout en étant efficace au-dessus d’une couche physique disposant d’une fonction de mesure distance (dans notre cas, CSS) sans intervention d’un serveur central de localisation.

Nous présenterons, dans la suite de ce chapitre, le protocole proposé et le comparerons, en termes d’*overhead* et de consommation énergétique, à d’autres solutions. Nous décrirons

ensuite l'algorithme utilisé pour le calcul de la position. Cet algorithme intégrera les conclusions de notre phase exploratoire, présentée au chapitre 2.2.

3.1. Proposition de protocole

Nous avons vu au chapitre 1.2.1.2 les principaux protocoles pouvant être exploités pour estimer le temps de vol. Nous allons proposer deux adaptations du meilleur d'entre eux, SDS-TWR, dans le cas où des mesures de distance doivent être effectuées avec n partenaires.

Avant d'entrer dans le vif du sujet, nous discuterons de la mise en place d'un cadre dans lequel la localisation puisse s'exécuter.

3.1.1. Négociation du *star interval*

Que le réseau exploite SSDS-TWR³⁶ (§ 3.1.2) ou un autre protocole de mesure, une phase de négociation est nécessaire. Tout d'abord, le risque de corruption des données évoqué en 1.2.1.2 indique qu'une exécution du protocole de mesure de distance dans un intervalle sans contention serait préférable : l'élimination des interférences générées par d'autres technologies étant impossible, il faut chercher à minimiser les perturbations provenant du même réseau. Particulièrement dans le cas de PSDS-TWR³⁷ (§ 3.1.3), la parallélisation des réponses appelle une exécution dans un intervalle temporel dédié.

Les études sur l'allocation des GTS [50][62] ont montré qu'il n'était pas intéressant de rendre disponible en permanence un GTS car il se retrouvait fréquemment inutilisé. Partant du même raisonnement, nous considérons que cet intervalle devrait être mis à disposition du mobile et des ancres à la demande, d'où la nécessité d'une négociation de cet intervalle. La mise en place de cette négociation, dans le cadre d'un réseau basé sur le standard IEEE 802.15.4 avec *beacons*, peut exploiter les communications existantes ou encore exiger des échanges dédiés ayant lieu dans la CAP. Une autre piste consisterait à supprimer la négociation et à exécuter le protocole de mesure de distance directement dans la CAP. Ceci impliquerait une dilatation de la mesure induite par l'utilisation du protocole d'accès au médium CSMA/CA et par conséquent une augmentation du coût énergétique de la localisation. Ceci étant incompatible avec notre objectif d'intégration de ce service avec un impact minimal sur le réseau, cette approche a été mise de côté. De plus, l'étalement temporel du processus accentuerait la détérioration de la précision liée à la dérive des horloges.

³⁶ *Sequential Symetric Double-Sided Two-Way Ranging*

³⁷ *Parallel Symetric Double-Sided Two-Way Ranging*

Dans notre travail, nous avons préféré considérer une négociation s'appuyant sur des échanges de *beacons* (transportant les informations nécessaires), de manière à minimiser l'impact protocolaire de la localisation. Il serait intéressant de déterminer le gain engendré par rapport aux solutions pour lesquelles la négociation est mise en œuvre dans la CAP. Nous en reparlerons dans les perspectives de cette thèse.

Dans la suite, nous considérons un réseau maillé autorisant la réservation d'un intervalle temporel, dédié à la localisation, appelé *star interval* [28]. Par définition, durant le *star interval*, le nœud mobile ayant effectué la demande est autorisé à communiquer directement avec tous ses voisins à un saut (les ancres ou éventuellement des mobiles associés à une localisation consistante) et demeure le seul à pouvoir initier un échange. Les liens utilisés forment une étoile, d'où le nom choisi pour cet intervalle.

La figure 42 illustre le concept du *star interval* : soit M le mobile et A,B,C et D des ancres à portée. Soient A, C et D les ancres choisies comme partenaires. Cette sélection sera basée sur la qualité des liens établis entre le mobile et ces ancres. Cette caractéristique sera fonction du RSSI et du pourcentage des trames périodiques émises par l'ancre et reçues par le mobile, ce dernier pouvant détecter l'absence d'un message par un saut dans le numéro de séquence. Alors que dans le fonctionnement général du réseau de capteurs, i.e. hors phases de localisation, tous les liens, matérialisés par une ligne reliant les nœuds, sont susceptibles d'être utilisés (figure 42 (a)), durant le *star interval*, seuls les liens établis entre le mobile propriétaire du *star interval* et les ancres participantes peuvent être exploités (figure 42(b)).

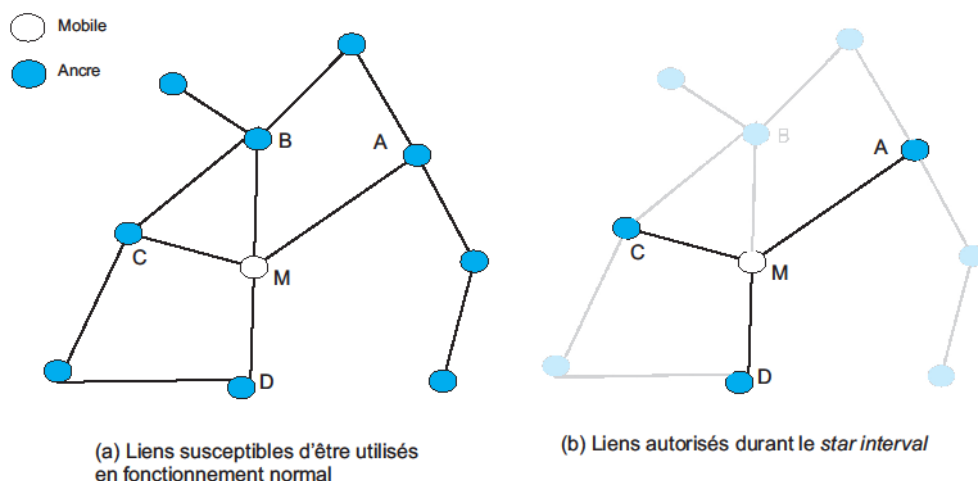


Figure 42 :Utilisation des liens radios dans le *star interval*

Afin de faciliter l'intégration de notre solution à des réseaux de capteurs existants et basés sur le standard IEEE 802.15.4 [54], nous nous proposons de respecter le découpage temporel

de base entre une période active et une période inactive. Cette structure temporelle du protocole est maintenue à travers la diffusion par les ancres de trames appelées *beacons* qui contiennent également dans notre cas la position de ces mêmes ancres. L'émission sans collision des *beacons* dépassant le cadre de cette thèse, nous invitons le lecteur intéressé à se référer aux travaux de [14] [63] pour des réponses à cette problématique. La figure 43 illustre les positions relatives du *star interval* et des périodes d'activités définies par le standard.

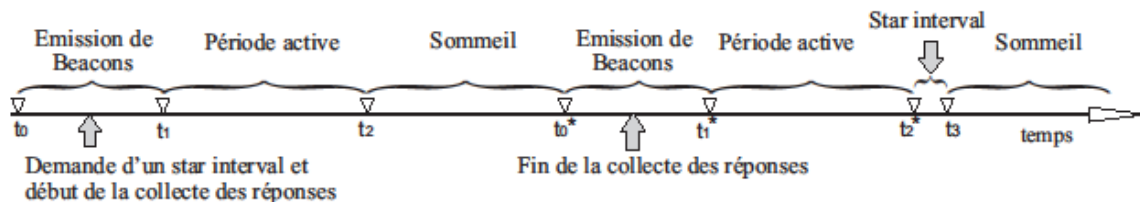


Figure 43 : Intégration du *star interval* dans la structure temporelle du protocole

On considère que le mobile est aussi un coordinateur et peut donc émettre des *beacons*. Ces derniers s'inscrivent dans une séquence impliquant les sources voisines, il est possible, par un réglage approprié de la durée d'écoute, d'assurer la collecte des *beacons* de tous les voisins. Le format général des *beacons* est donné dans la figure 44.

Octets: 2	1	4/10	0/5/6/10/14	2	1	0/1	variable	variable	variable	2
Champ de contrôle	Numéro de Séquence	Champs d'adresses	En-tête de sécurité	Spécification de la supertrame	Spécification des GTS	Direction des GTS	Liste des GTS	Liste d'adresses	Charge utile	FCS
En-tête MAC				Charge utile MAC						En-queue MAC

Figure 44 : Format général d'un beacon

Le *beacon* peut se découper en trois parties : l'en-tête MAC ou *MAC Header* (MHR), la charge utile de niveau MAC et le pied MAC ou *MAC Footer* (MFR) contenant le code de redondance cyclique (CRC). Le champ de contrôle spécifie, entre autres aspects, le type de trame (sur 3 bits) parmi les options suivantes :

- *Beacon*
- Données
- Acquittement
- Commande MAC

Dans le cadre de notre solution de localisation, quatre types de *beacons* seront considérés, la différenciation étant réalisée à partir de leur *payload* :

- **Beacon général** Ce *beacon* est émis par tous les nœuds en dehors d'un processus de localisation. Il remplit uniquement des fonctions de coordination temporelle de

l'ensemble du réseau, tout en transportant l'information de position évoquée ci-dessus. L'ensemble des voisins à un saut a l'opportunité de mettre à jour la position connue du nœud émetteur à chaque réception de *beacon*.

- **Beacon de demande de localisation** En plus de participer à la conservation de la structure temporelle, ce *beacon* contient dans sa *payload* une liste de nœuds avec lesquels le mobile émetteur désire réaliser une opération de localisation, ainsi que l'identifiant du protocole à utiliser.

Tableau 8 : Codage du protocole de mesure et des réponses des ancres

Valeur	Sémantique
0	TWR
1	SDS-TWR
2	PSDS-TWR
3	Réservé

Nous avons décidé de consacrer deux bits à cette information, les différentes valeurs étant listées dans le tableau 8.

- **Beacon de réponse de localisation** les nœuds sollicités pour la localisation émettent leur réponse dans leur prochain *beacon*. Cette réponse peut correspondre à l'un des trois cas suivants :
 - **Requête acceptée** Lorsque l'ancre cible va participer à un exercice de localisation et que le *star interval* visé est libre, elle accepte de participer à l'exécution du protocole. Cet accord est matérialisé par l'émission, dans la *payload*, du code du protocole demandé (tableau 8) et de l'adresse du nœud mobile requérant la participation de l'ancre,
 - **Requête rejetée** Dans ce cas, l'ancre ne peut pas participer à la localisation pour des raisons diverses (le protocole n'est pas implémenté, la configuration l'interdit...),
 - **Opposition** L'ancre s'oppose à la demande de localisation car, selon les données dont elle dispose, le *star interval* visé est déjà assigné à un autre mobile. Pour éviter des collisions, la seconde demande d'utilisation sera bloquée. Cette réponse sera typiquement émise par une ancre commune à deux îlots (figure 45) et sollicitée pour le même *star interval* par deux mobiles différents. Ici, si on suppose que E a déjà accepté de coopérer durant le *star interval* avec M1, rejeter simplement la demande de M2 ne

suffit pas car alors, M2 pourrait engager la procédure avec B, F et D et perturber les communications au niveau de E (terminal caché).

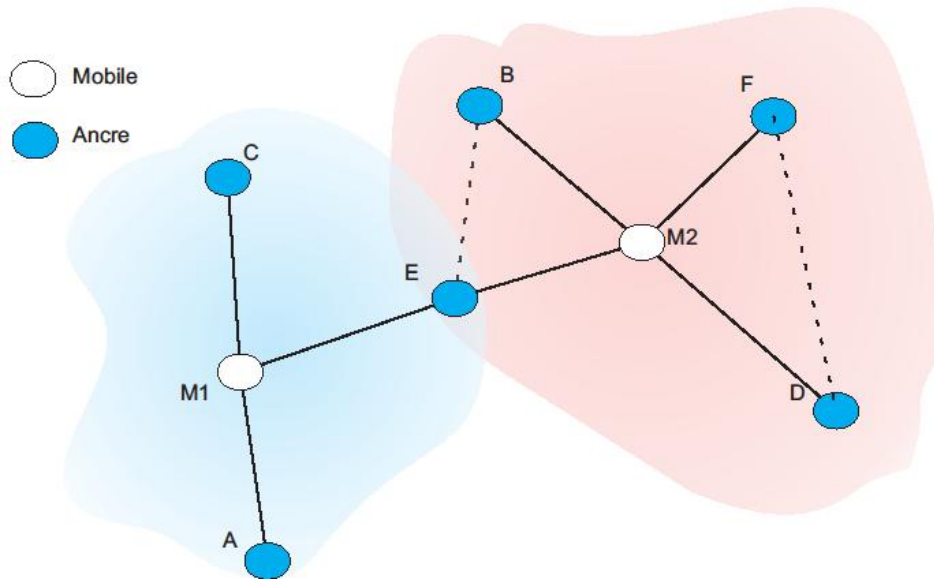


Figure 45 : Ancre commune à deux îlots de localisation potentiels

- **Beacon de confirmation** Ce *beacon* permet au mobile d'indiquer à l'ensemble de ses partenaires que le processus de localisation sera ou non exécuté, suivant l'ensemble des réponses reçues.

Pour se localiser, le mobile se met à l'écoute des *beacons* pendant un temps déterminé. A l'issue de cette phase, il dispose d'une liste d'ancres avec lesquelles il pourra communiquer durant le *star interval*. Dans son prochain *beacon*, il inclut sa demande de localisation pour un *star interval* placé à la fin du prochain *beacon interval*. Chaque ancre diffusera sa réponse dans son prochain *beacon*. Suivant la configuration et les décisions préalablement prises par l'ancre sollicitée, cette réponse sera de l'un des trois types vus précédemment. Le mobile, après avoir collecté l'ensemble des réponses, décidera de confirmer ou d'annuler la réservation. Si l'une des réponses des ancres correspond à une opposition, le mobile annule le rendez-vous. Sinon, il émet un code de confirmation. Il est possible d'inclure le nombre de partenaires ayant accepté la requête de manière à permettre aux ancres de limiter dynamiquement la taille du *star interval*.

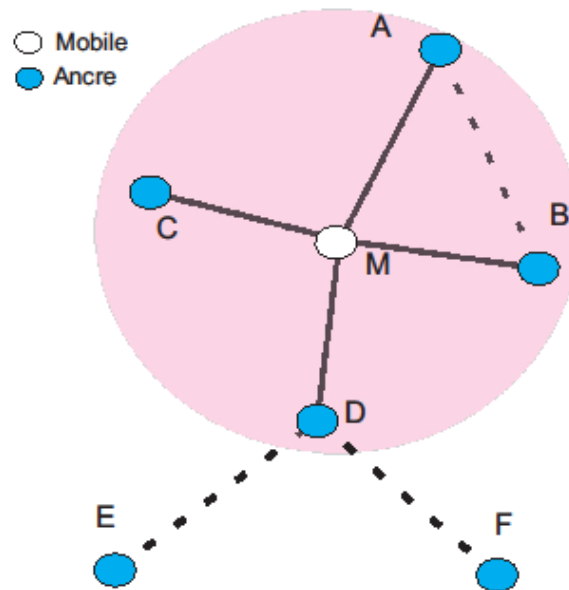


Figure 46 : Exemple de topologie pour la négociation du *star interval*

La séquence de négociation du *star interval* est illustrée à la figure 47 pour la topologie correspondant à la figure 46. Les liens entre les nœuds sont représentés par des segments de droite. Dans la situation schématisée, le mobile a déterminé que les voisins directement accessibles sont A, B, C et D. Ils seront sollicités pour la localisation.

Les échanges de localisation sont répartis sur deux *beacons intervals* correspondant aux blocs 2 et 3 de la figure 47, le premier illustrant l'identification des voisins par la réception de leurs *beacons*. Les trames échangées sont représentées par des segments de droite sur lesquels un triangle a été dessiné au niveau de chaque nœud récepteur. Le nœud émetteur de la trame est identifié par la présence d'un rond dont la couleur dépend de son rôle dans la localisation (mobile ou ancre). On notera que l'axe temporel vertical n'est pas à l'échelle.

Afin d'implémenter cette négociation, nous proposons la structure de trame illustrée en figure 48 (a) pour le message de localisation transporté par les *beacons*. Ce message sera donc divisé en 2 ou 3 parties. Le premier octet contiendra le descripteur des suivants. Le premier bit indique si la position de l'émetteur, codée sur 4 octets, est annoncée ou non dans cette trame. Les deux suivants identifient le rôle de ce champ de localisation : il peut s'agir d'une simple annonce de l'adresse (code 00), d'une requête de localisation (code 01), d'une réponse de localisation (10) ou encore d'une confirmation (code 11). Deux autres bits servent à identifier le protocole (tableau 8). Finalement, les trois derniers bits représentent un nombre entier n . Dans le cas d'une requête de localisation, cette valeur de n indique le nombre d'adresses courtes d'ancres listées dans la suite du message. Lorsqu'une ancre utilise ce format, n correspond au nombre de mobiles auquel l'ancre en question répond : la suite du

message sera constituée de n réponses dont le format est donné en figure 48 (c). Chacune de ces réponses commence par un octet de contrôle suivi de l'adresse du mobile auquel s'adressent les informations de l'octet. Dans cet octet de contrôle, 4 bits seront utilisés : les bits 0 et 1 serviront à rappeler le protocole demandé tandis que les bits 2 et 3 codent la réponse. Le tableau 9 indique le codage des réponses.

La relation entre les différents champs est expliquée à travers cet exemple : on suppose que deux mobiles veulent effectuer la localisation et ciblent le même *star interval*. On imagine également qu'une des ancrs est commune aux deux îlots. La figure 49(a) représente la requête typiquement émise par les mobiles. Le descripteur commençant par zéro, la position du nœud émetteur, connue ou pas, n'est pas annoncée. La suite de l'octet permet de voir qu'il s'agit d'une requête d'utilisation de PSDS-TWR concernant 4 ancrs. A la réception, chaque ancre détermine si elle est concernée ou pas.

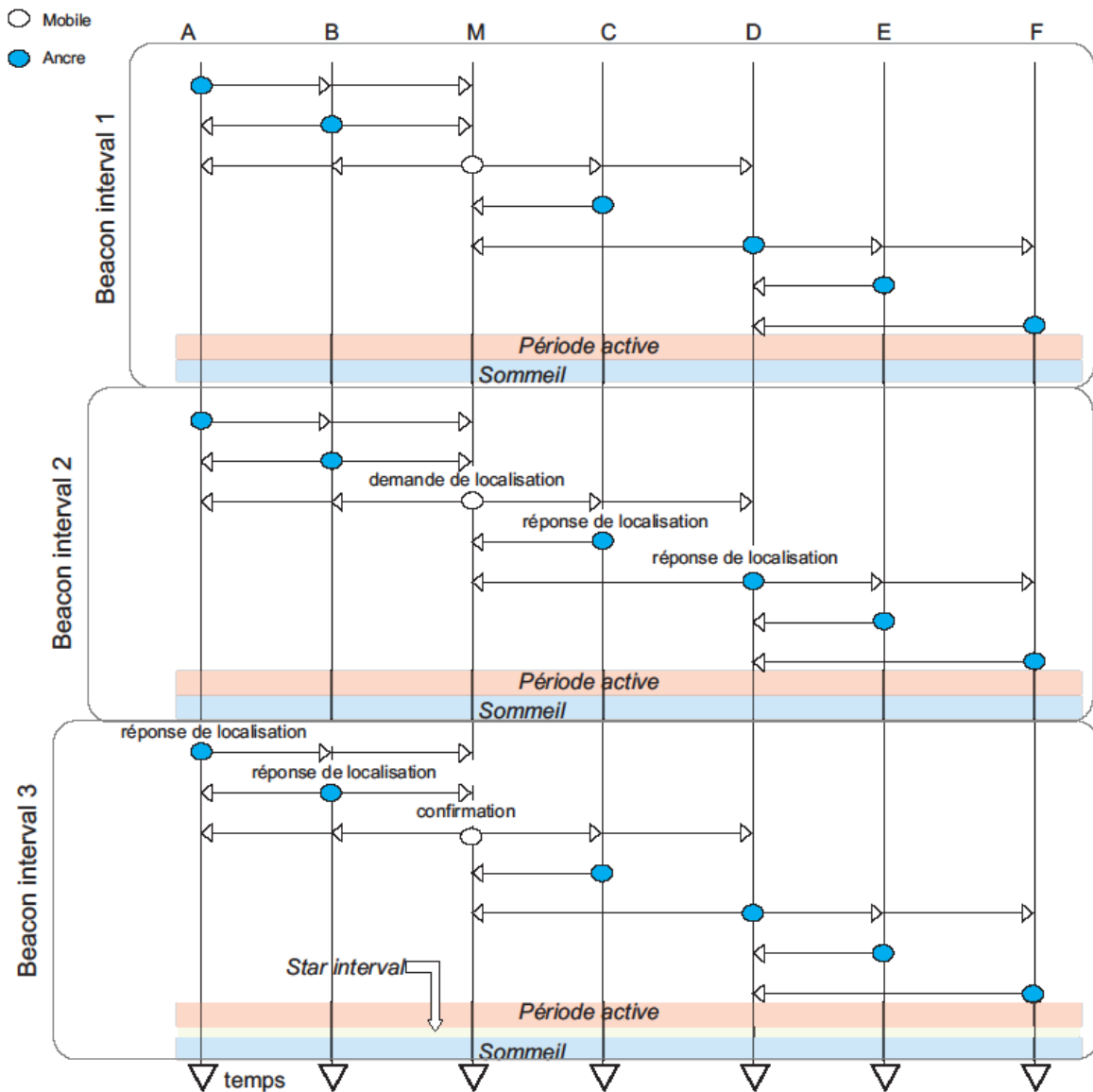


Figure 47 : Négociation du *star interval*

Tableau 9 : Codage des réponses à une demande de localisation

bits 2-3	Sémantique
00	Requête acceptée
01	Requête rejetée : protocole inconnu
10	Requête rejetée : localisation interdite
11	Opposition à la requête

Soit C l'ancrage commun aux deux flots. On suppose que C est amené à participer à des exercices de localisation. Les deux mobiles l'ayant sollicité pour deux *star interval* concomitants, il devra répondre positivement à l'un des mobiles tout en s'opposant à l'autre

demande. La trame jouant ce rôle correspond à la figure 49(b) : l’ancre annonçant sa position, le descripteur débute par un 1 suivi du code représentant les réponses (10). Le sous-champ protocole est ici mis à 0 et n’est pas interprété par les récepteurs. La valeur de n correspond au nombre de mobiles auxquels l’émetteur répond dans la suite du message.

Octets : 1	0/4	Variable
Descripteur	Position de l'émetteur	Données complémentaires

(a) Format général du message de localisation

7	6	5	4	3	2	1	0
Position	Type de message	Protocole		N			

(b) Descripteur du message localisation

7	6	5	4	3	2	1	0
Réservé				Type de message		Protocole	

(c) Descripteur de la réponse à une requête de localisation

Figure 48 : Format des différentes parties du message de localisation

Octets : 1	2	2	2	2
0 0 1 0 1 1 0 0	adresse 1	adresse 2	adresse 3	adresse 4

(a) Exemple de payload: requête de localisation avec PSDS-TWR avec 4 ancres

Octets : 1	4	1	2	1	2
1 1 0 0 0 0 1 0	Position	X X X X 0 0 1 0	adresse1	X X X X 1 1 1 0	adresse 2

(b) Exemple de payload: réponse de localisation à deux demandes, la première étant acceptée et la seconde bloquée

Figure 49 : Exemples d'utilisation du message de localisation inclus dans les beacons

3.1.2. SSDS-TWR : échanges et formules

Le *Sequential Symmetric Double Sided Two-Way Ranging* (SSDS-TWR) peut être considéré comme la suite naturelle de SDS-TWR. Les échanges de trames de SDS-TWR sont exécutés avec n ancres séquentiellement (figure 50).

Le mobile, après avoir contacté les ancres, détient un n -uplet de distances estimées à combiner pour obtenir la position. Dans le cas de la figure 51, il s’agit du quadruplet [3,5 ; 4,5 ; 7,8 ; 1,5].

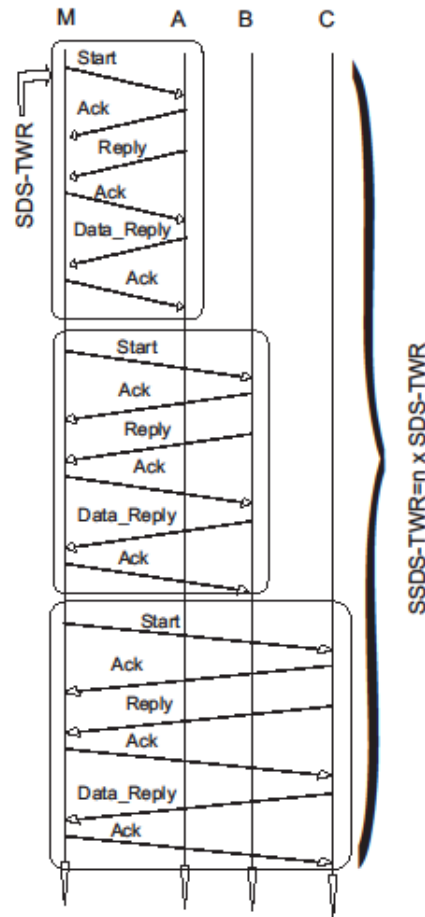


Figure 50 : Le protocole SSDS-TWR

Une fois que le *star interval* aura été retenu, le mobile et les balises exécuteront le protocole *Sequential SDS-TWR* (SSDS-TWR). Il consiste, comme son nom l'indique, à exécuter SDS-TWR avec chacune des balises disponibles, tour à tour (figure 50).

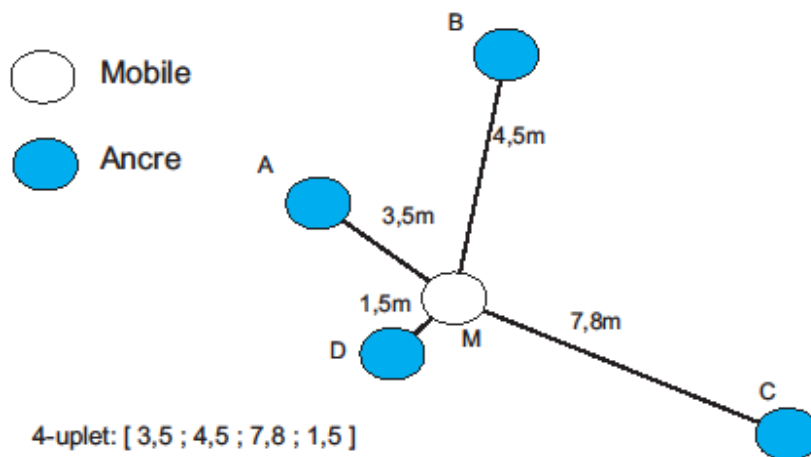


Figure 51 : Liens radio et quadruplet de distances

Supposons que la configuration de l'ancre D lui interdise de participer à la localisation. Le mobile exécutera le protocole SDS-TWR à trois reprises, avec les partenaires A, B puis C

(figure 50). A chaque reprise, deux TWR consécutifs peuvent être observés. Le premier est constitué de la trame *Start* du mobile et de l'acquiescement émis par l'ancre tandis que le second repose sur la trame *Reply* de l'ancre et l'acquiescement du mobile. Ces deux TWR permettent d'obtenir des estampilles temporelles au niveau du mobile et de l'ancre. Pour pouvoir calculer le temps de vol, il est nécessaire que l'ensemble de ces informations soit disponible à un même équipement. Le rôle de la dernière trame de la séquence est donc de rapatrier vers l'initiateur la moitié des données détenue par l'ancre.

Comme indiqué précédemment, l'intervalle durant lequel le protocole de localisation s'exécute est désigné comme *star interval* car seul le mobile, centre de l'étoile, peut initier les échanges avec les ancres choisies. Ce groupe de nœud forme donc géographiquement un îlot disposant d'une tranche temporelle réservée durant laquelle la compétition sur l'accès au médium entre les nœuds sera désactivée.

Dans la suite, nous ferons référence à la charge utile des trames échangées durant l'exécution de SSDS-TWR. Nous avons décidé de conserver la définition originelle de ces données. Les trames *Start*, *Reply* et *Data_Reply* transportent une structure de 11 octets dont le format est spécifié dans le tableau 10.

Tableau 10 : Structure de la payload dans SSDS-TWR

Champ	Taille (octets)
PHASEOFFSETACK	1
TOAOFFSETMEANACK	2
TXRESPTIME	2
PHASEOFFSETDATA	1
TOAOFFSETMEANDATA	2
RANGING_ERROR	1
COUNTER	1
ACTTXID	1

Si SSDS-TWR permet d'effectuer les mesures de temps de vol avec les balises tour à tour, il n'est pas sans inconvénient. Tout d'abord, 3 trames de données et 3 acquiescements sont nécessaires pour chacune des ancres. Le nombre de trames échangées augmentera donc linéairement avec le nombre de partenaires. Bien que dans notre situation ne nous amène à manipuler que des liens de taille homogène (quelques dizaines de mètres maximum), il est intéressant de noter que lorsque l'homogénéité n'est pas respectée, la durée d'exécution de SSDS-TWR ne croît pas qu'avec n . En effet, si l'on considère le triplet de distances exprimées en mètres (10,15, 20000), la durée d'exécution du protocole avec les deux premiers partenaires sera différente de celle observée avec le troisième, la taille du lien introduisant un

décalage de l'ordre de 10^{-4} s. La durée ne correspond alors pas à une constante que multiplie le nombre d'ancres.

Finalement, d'un point de vue énergétique, quand le nombre de partenaires de localisation du mobile est grand, les dernières balises de la séquence sont en réception tant que le mobile ne s'adresse pas à elles. Comme nous l'avons vu dans nos explorations et métrologies préalables (§ 2.3), la réception est plus consommatrice que l'émission sur les plateformes visées. Il serait donc judicieux de définir un protocole réduisant la durée d'occupation de cet état afin de contribuer à la longévité du réseau liée à la consommation énergétique.

3.1.3. PSDS-TWR : échanges et formules

Si on considère que le mobile a pris rendez-vous pour un *star interval* auprès de n balises, cela signifie que ces n balises vont demeurer en réception durant toute la durée de l'intervalle en attendant qu'arrive le tour de chacune. Selon le nombre de balises, la dernière de la liste peut rester à l'écoute pendant un temps plus ou moins long. Ceci est regrettable quand le seul nœud qui puisse la contacter est occupé avec une autre ancre.

D'un autre côté, l'îlot formé par le mobile et ses balises bénéficie d'une réservation du médium, le *star interval*. Durant cet intervalle, le mobile est le seul initiateur réel. Cela évoque une méthode basique de l'exécution de multiples processus simultanément, le *round-robin* : dans notre cas, avec un ordonnancement approprié, le mobile pourrait indiquer aux balises quand répondre sans occasionner de collisions.

Afin de réduire le nombre de trames ainsi que la durée totale d'exécution de la phase de *ranging*, nous avons imaginé d'exécuter la mesure entre un mobile unique et plusieurs ancres, en parallèle. Nous avons ainsi nommé notre protocole, *Parallel SDS-TWR* (PSDS-TWR). La figure 52 illustre la différence entre PSDS-TWR et SSDS-TWR. Nous avons choisi de définir les messages comme des commandes MAC : cela signifie qu'au niveau de la *payload*, un octet sera réservé pour l'identification de la commande spécifique associée à la trame. Quatre trames sont définies : *Location_Start*, *Reply1*, *Data_Request* et *Reply2*. A chaque type sera associée un code de commande MAC. La trame *Location_Start* contiendra la liste ordonnée des adresses des ancres concernées par la localisation et le code de commande. Cet ordonnancement pourra se baser sur le RSSI, favorisant ainsi les ancres dont le signal est le meilleur. La trame *Reply1* contiendra simplement un code indiquant la participation effective au processus. La trame *Data_Request* contiendra l'identifiant de cette phase du protocole, les ancres connaissant déjà leur position dans la liste à partir de la trame *Location_Start*.

Finalement, *Reply2* permettra à chaque ancre de faire parvenir au mobile les trois estampilles collectées pour *Location_Start*, *Reply1* et *Data_Request* (figure 52).

Du point de vue temporel, un décalage doit être calculé par chaque ancre afin de déterminer l'instant d'émission de sa réponse sans provoquer de collision. Il s'agit de T_A pour la balise A. Soient :

t_S : l'instant d'émission, mesuré par M, de *Location_Start*

$t_{1,A}$: l'instant de réception, mesuré par A, de *Location_Start*

$t_{R,A}$: l'instant de réception, mesuré par M, de *Reply1* envoyé par A

$t_{2,A}$: l'instant de réception, mesuré par A, de *Data_Request*

$t_{D,S}$: l'instant d'émission, mesuré par M, de *Data_Request*

$t_{V,A}$: le temps de vol entre M et A

Le temps de vol est alors lié à ces différentes informations par les équations 53 et 54.

$$t_{V,A} = \frac{t_{R,A} - t_S - T_A}{2} \quad (53)$$

$$t_{V,A} = \frac{t_{2,A} - t_{1,A} - T_A - (t_{D,S} - t_{R,A})}{2} \quad (54)$$

D'où :

$$t = \frac{2t_{R,A} + t_{2,A} - 2T_A - t_S - t_{D,S} - t_{1,A}}{4} \quad (55)$$

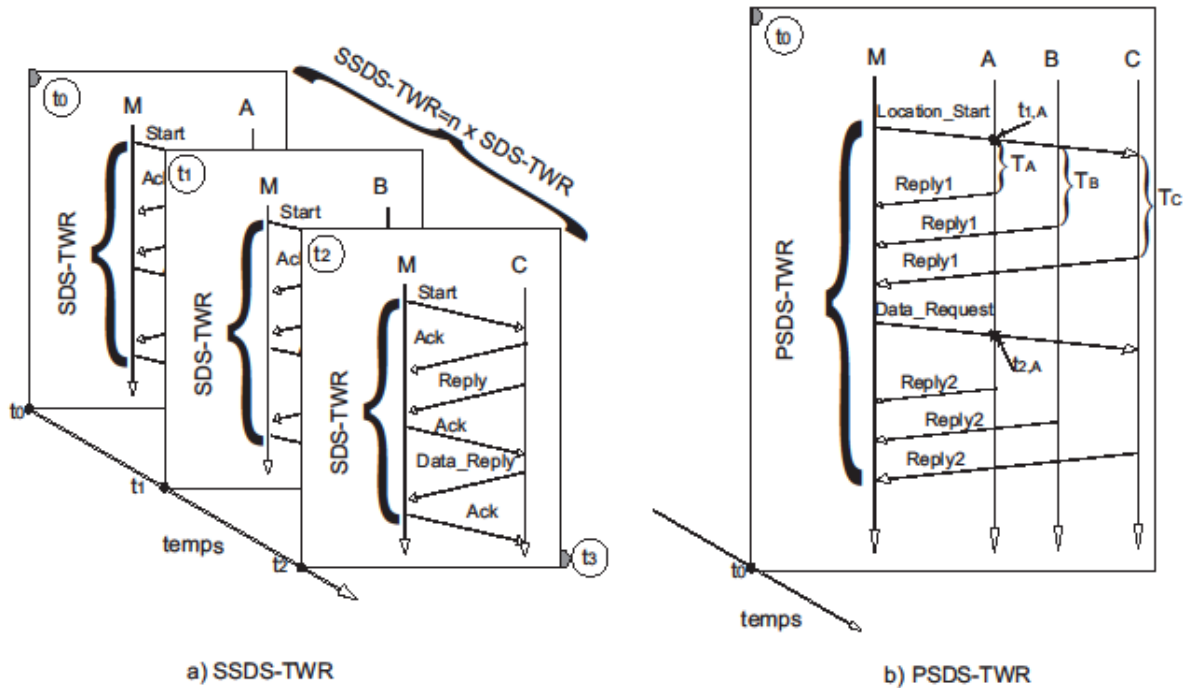


Figure 52 : SSSD-TWR et PSDS-TWR

La figure 52 permet de comparer les deux protocoles sur un exemple. Les deux situations font intervenir 3 ancres et un mobile. Dans l'exécution de SSSD-TWR, trois échanges, dissociés dans le temps, ont lieu. On peut les considérer comme chacun borné dans le temps, ce qui rend le temps total maximal égal au produit de cette borne temporelle par le nombre d'ancres. Cette dissociation entre les 3 exécutions implique que les messages échangés avec A sont répliqués pour effectuer une mesure similaire avec B et C. D'un autre côté, différencier les communications à destination de chacune des ancres permet l'utilisation du mécanisme de réémission automatique en cas d'absence de l'acquiescement. Le fait d'attendre un ACK est alors mentionné dans la trame de données. Lorsque cet ACK n'est pas reçu, l'émetteur suppose que la trame initiale a été perdue, retransmet le message et se remet en attente de l'ACK. Ce processus se déroule un certain nombre de fois jusqu'à ce que le nombre de tentatives maximal soit atteint ou que l'ACK soit reçu. L'ACK permet donc de détecter les échecs de transmission au prix d'une augmentation de la durée de l'exécution de SDS-TWR.

Le protocole PSDS-TWR repose sur la diffusion de deux des quatre messages, *Location_Start* et *Data_Request*. Il permet, à deux reprises, à toutes les ancres concernées, de profiter d'un seul et même message du mobile. Ce dernier se comporte donc comme un chef d'orchestre : ses messages sont interprétés par toutes les ancres impliquées qui répondent de manière harmonieuse. Dans le cas de la mesure de temps, les TWR des différentes ancres sont imbriquées. Par exemple, la trame *Location_Start* du mobile est commune au premier TWR

de chacune des ancrs tandis que la trame de retour est matérialisée par la trame *Reply1* de chacun des partenaires.

L'évaluation des apports de PSDS-TWR constitue le sujet des sections suivantes. Nous examinerons la réduction de la charge réseau induite ainsi que la diminution de la consommation énergétique.

3.1.4. Comparaison de la charge réseau des deux protocoles

Nous pouvons comparer la taille totale des données échangées avec celle de l'implémentation existante de SDS-TWR. Soit n le nombre d'ancres à contacter. Pour SDS-TWR, chacune des trois trames de données transporte 11 octets de charge utile. La quantité totale pour n exécutions de SDS-TWR successives est donnée par l'équation 56.

$$Qt_1(n) = 3 \text{ [trames]} * 11 \text{ [octets]} * n \text{ [balises]} = 33n \quad (56)$$

Dans le cas de PSDS-TWR, la trame *Start* contient une liste de n adresses courtes 802.15.4 (§ 1.1.1) donc $2n$ octets et un octet identifiant le type de commande. Les codes respectifs des trames *Reply1* et *Data_Request* seront codés sur 1 octet. Chaque estampille temporelle pourra être codée sur 4 octets. Ces 32 bits serviront à coder un nombre d'unités temporelles dépendant de la plateforme et de l'horloge utilisée. Un octet supplémentaire sera ajouté à ces estampilles pour préciser qu'il s'agit d'une trame de type *Reply2*. La quantité de données totale en fonction de n s'écrit :

$$\begin{aligned} Qt_2(n) &= \text{somme des charges utiles de (Start+ n*Reply_1+ Data-Request+n*Reply_2)} \\ &= 2n+1+n+1+13n = 2+16n \end{aligned} \quad (57)$$

Le gain est donc :

$$\text{Gain} = (17n-2)/33n \quad (58)$$

La dérivée du gain étant positive, il augmentera avec le nombre de balises contactées jusqu'à sa limite $17/33$.

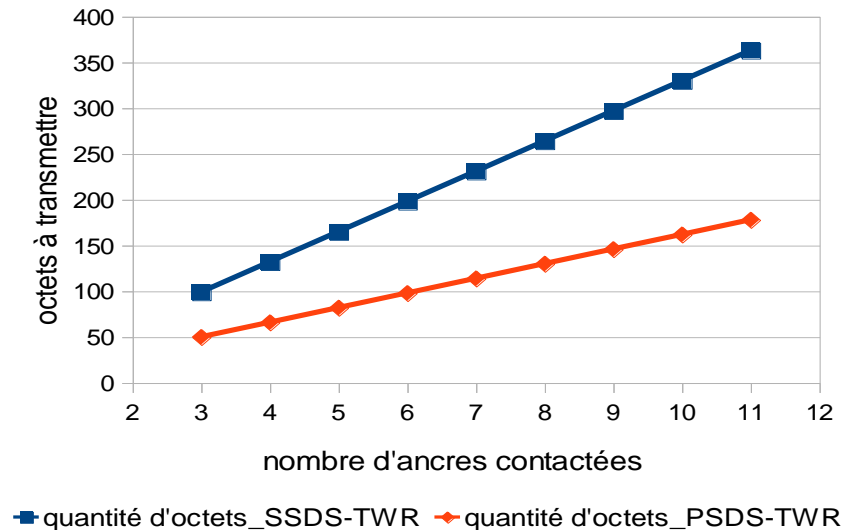


Figure 53 : Volumes d'information échangés par SSDS-TWR et PSDS-TWR

Utiliser PSDS-TWR apporte également des avantages en termes de durée d'exécution. Durant la phase de simulation du protocole PSDS-TWR (§ 4.2.6), des mesures seront effectuées quant à sa durée d'exécution. Ces informations seront ensuite rapprochées des résultats de SSDS-TWR, collectés sur notre prototype.

D'un autre côté, les ACK étant supprimés, un message perdu l'est définitivement. S'il s'agit de la trame *Location_Start* et qu'aucune ancre ne l'ait reçue, le mobile peut détecter cette situation par une absence de réponse de la part des ancres de la liste. Dans cette situation, il peut et va réémettre la trame *Location_Start*. Dans le cas où une réponse d'une ancre est perdue, les informations temporelles liées à cette ancre sont inexploitable, ce qui implique la nécessité de contacter plus de balises que strictement demandé par le type de localisation (2D ou 3D) ou de trouver un moyen de reconstruire les données manquantes. En effet, trois trames étant utilisées pour générer les 6 instants nécessaires, la perte de *Reply1* rend impossible l'estimation du temps de vol tandis que celle de *Reply2* empêche l'application de la formule de l'équation 11 et donc le calcul de la distance, les données provenant des balises ne pouvant être récupérées. Par conséquent, si trois ancres sont contactées et que l'une des trames est perdue, l'ensemble des données de référence sera insuffisant pour une localisation en 2D.

L'impact de la nécessaire augmentation du nombre d'ancres contactées sur la quantité de données à transmettre reste tout de même limité. La figure 53 indique que PSDS-TWR pourrait contacter deux fois plus d'ancres avant de parvenir à rivaliser avec SSDS-TWR sur ce point.

Dans cette étude, nous nous sommes limités exclusivement à l'impact protocolaire de nos protocoles dans des conditions optimales. La prise en compte des en-têtes ainsi que des retransmissions ne ferait qu'accentuer les avantages apportés par l'utilisation de PSDS-TWR.

3.1.5. Comparaison de la consommation énergétique induite par les deux protocoles

Comme nous l'avons vu à la section 2.3, la consommation induite par un protocole dépend des caractéristiques de la plateforme matérielle ciblée mais aussi de la durée d'occupation des états d'émission, de réception, etc. Par conséquent, nous commencerons notre analyse en caractérisant les durées pour chacun des protocoles. Nous examinerons ensuite l'impact de SSDS-TWR et de PSDS-TWR sur la consommation.

3.1.5.1. Caractérisation temporelle de SSDS-TWR

Nous considérerons un ensemble formé du mobile M et de $n=3$ ancres, désignées par A, B et C, collaborant pour déterminer la position de M. Nous rappelons ci-dessous la structure temporelle de l'exécution du protocole SSDS-TWR.

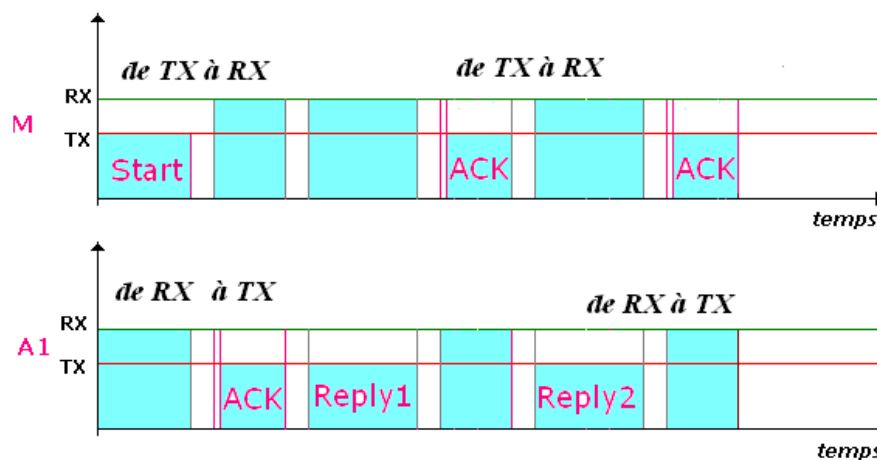


Figure 54 : Echanges avec une ancre durant SSDS-TWR

La figure 54 indique que trois échanges de trames doivent avoir lieu pour chacune des ancres partenaires. Les communications sont ici systématiquement acquittées.

Deux catégories de durées peuvent être ici identifiées : celles qui ont été mesurées sur le prototype et celles qui sont calculées à partir de formules classiques.

Concernant la première catégorie, toutes auraient pu être prédites mais, dans le cas des créations de trames, il aurait fallu par exemple intégrer la décomposition en langage machine de chacune des instructions appelées et ensuite estimer le temps d'exécution de ce code pour

une plateforme donnée. Nous avons donc préféré récupérer ces informations sur notre prototype à l'aide d'un oscilloscope. Nous avons matérialisé les durées des différentes phases par des changements d'état sur les sorties binaires de notre microcontrôleur. Le tableau 11 indique les valeurs obtenues pour les différentes phases en faisant appel à notre version du code optimisée pour la localisation.

Tableau 11 : Durées des différentes étapes pour l'exécution de SSDS-TWR

Phase	Durée
Création de la trame <i>Start</i>	1,6ms
Création de la trame <i>Reply1</i>	600µs
Création de la trame <i>Reply2</i>	40µs
Durée totale d'une opération de <i>ranging</i>	5,6ms
Délai séparant la fin d'un <i>ranging</i> et le début de la suivante	6,4ms

Le délai séparant les opérations de *ranging* utilisé par le microcontrôleur pour sauvegarder les résultats de l'échange précédent dans les structures dédiées.

Les durées de la seconde catégorie sont calculées en exploitant les valeurs et relations indiquées dans le standard [54], ainsi que les *datasheets* des équipements utilisés.

- **Durée d'émission des trames** Le temps nécessaire à l'émission d'une trame de n bits par une interface de débit r (bits/s) est n/r .
- **Délai d'arrivée du premier bit du message à un récepteur situé à une distance d sans perturbations** Cette durée est calculée par l'équation 59.

$$\begin{aligned} \text{délai} &= \text{émission du premier bit} + \text{traversée de } d \text{ par le premier bit} \\ &= \frac{1}{\text{débit de l'interface}} + \frac{d}{\text{vitesse de propagation}} \end{aligned} \quad (59)$$

- **Temps d'attente de l'acquittement** Il s'agit du délai accordé pour que parvienne l'ACK d'une trame précédemment émise. Le calcul de cette durée dépend du paramètre *macAckWaitDuration*. Son calcul se base sur la formule suivante (équation 60).

$$\begin{aligned} \text{macAckWaitduration}_{1M} &= aUnitBackoffPeriod + aTurnaroundTime + \\ & \text{phySHRDuration}_{1M} + [1.5 + 3/4 \times \text{ceiling}(4/3 \times 5)] \times \text{phySymbolsPerOctet}_{1M} \end{aligned} \quad (60)$$

avec :

aUnitBackoffPeriod : le nombre de symboles formant une période de base dans la norme IEEE 802.15.4. Sa valeur est fixée à 20.

aTurnaroundTime : durée maximale des transitions du mode réception au mode transmission et vice versa, exprimée en symboles. Sa valeur est 12.

phySHRDuration : durée de l'en-tête de synchronisation exprimée en symboles de la couche PHY utilisée. Pour une couche physique CSS à 1Mb/s, sa valeur est 12.

phySymbolsPerOctet : le nombre de symboles par octet. Ce paramètre dépend de la couche physique choisie. Dans le cas de la couche physique CSS à 1Mb/s, cette valeur est de 4/3.

La constante *macAckWaitDuration* indique le nombre de symboles correspondant à l'attente de l'ACK. Dans la norme, la durée d'un symbole pour la couche physique CSS à 1Mb/s est de 6µs. En remplaçant l'ensemble de ces informations dans l'équation 60, nous aboutissons à une durée maximale d'attente de l'ACK de 318µs.

Nous pouvons donc résumer dans le tableau 12 l'occupation des états de transmission et de réception des différents nœuds impliqués. Dans le cas de SSDS-TWR, l'exploitation d'un mode d'économie d'énergie ne fait pas partie du protocole bien que les ancres, une fois leur participation terminée, soient autorisées à entrer dans un état de sommeil. Par conséquent, si on désigne par *o* la position d'une ancre quelconque dans la séquence des *n* ancres à contacter par le mobile, cette ancre devra rester en réception jusqu'à ce qu'elle soit contactée, c'est-à-dire pendant *o-1* exécutions de SDS-TWR. Un nouveau paramètre fait son apparition dans le tableau 12 : il s'agit de *padding*, surdimensionnement des fenêtres de réception.

Tableau 12 : Répartition du temps entre émission et réception pour SSDS-TWR

Type de nœud	Durée de réception	Durée d'émission
Mobile	$(T_{\text{Reply1}} + T_{\text{Reply2}} + 2*\text{padding}) * n$	$(T_{\text{Start}} + 2*T_{\text{ACK}}) * n$
Ancre	$T_{\text{Start}} + \text{padding} + 2*T_{\text{ACK}} + (n-o)*$ $\text{SSDS-TWR}_{\text{durée moyenne}}$	$T_{\text{ACK}} + T_{\text{Reply1}} + T_{\text{Reply2}}$

3.1.5.2. Caractérisation temporelle de PSDS-TWR

Ce protocole n'exploite pas d'acquittements mais repose sur des durées convenues entre le mobile et les ancres. Le mobile entame la séquence en diffusant une trame *Location_Start* et

en se plaçant immédiatement après en réception. Il demeurera dans cet état jusqu'à l'expiration de *Timer1*. La valeur de *Timer1* dépendra du nombre d'ancres concernées par cette opération de localisation (§ 4.2.6.1). A l'expiration du *Timer1*, le mobile considère que toutes les réponses lui sont parvenues et s'apprête à démarrer la seconde phase du processus. Pour ce faire, il reformate la trame *Data_Request* puis commence son émission. De la même manière, une fois cette transmission terminée, il arme *Timer2* et se place dans l'état de réception jusqu'à la fin du délai d'attente configuré. Lorsque toutes les réponses ont été récupérées, le mobile peut éteindre son *transceiver* et se consacrer au calcul de sa position.

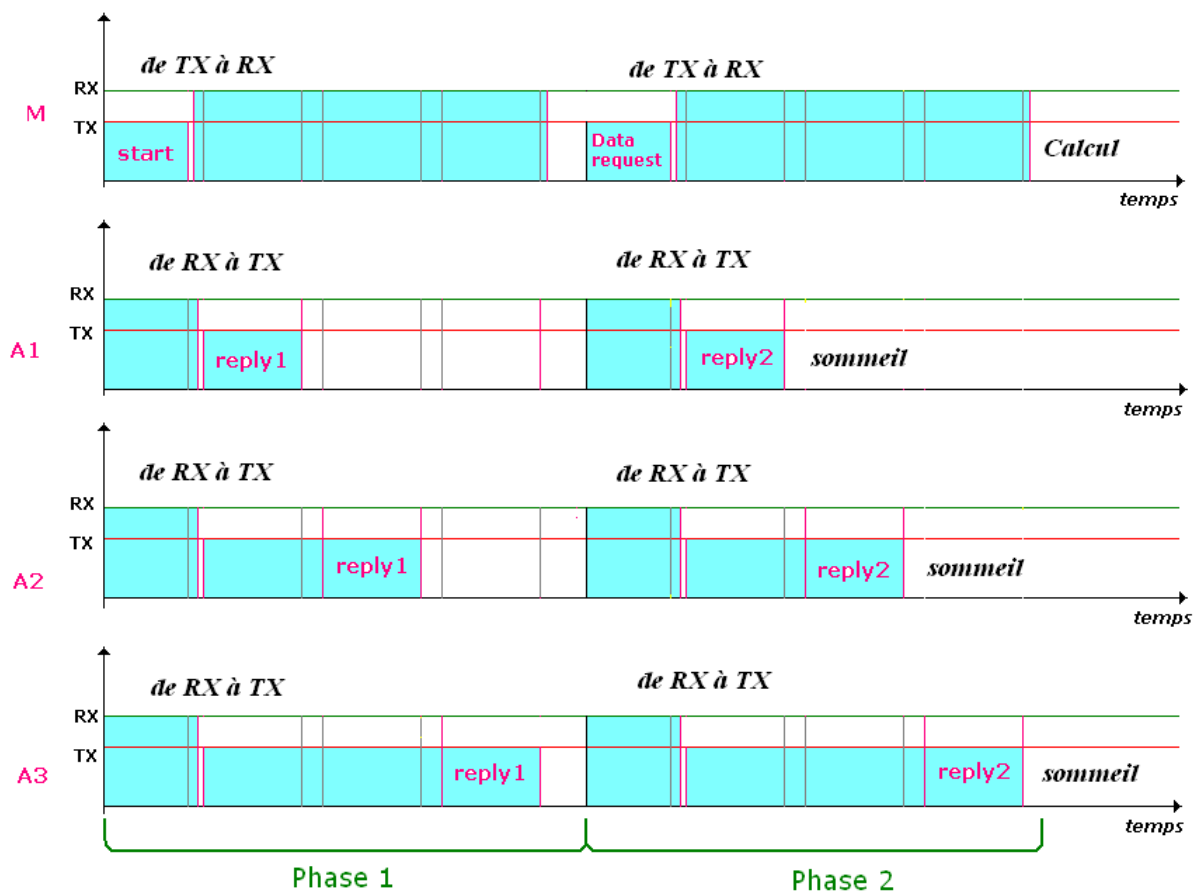


Figure 55 : Séquence des échanges durant l'exécution de PSDS-TWR

Les ancres, au début du *star interval*, sont en réception. Une fois le message du mobile reçu, elles déterminent leur rang dans la liste des réponses et, en fonction de cette information, déterminent l'instant d'envoi de leur réponse, la trame *Reply1*. Théoriquement, certaines d'entre elles pourraient profiter de la période séparant la fin de l'émission de *Reply1* et la fin prévue de la première phase pour entrer en mode économie d'énergie. Avant la fin de la première phase, les ancres configurent à nouveau leur *transceiver* en mode réception afin de détecter la trame *Data_Request*. Elles respecteront la place assignée dans la liste des ancres

pour émettre leur seconde réponse. Au terme de cette action, elles entreront en mode économie d'énergie.

Sur la figure 55, la succession des états occupés par les équipements radio a été représentée.

L'implémentation de PSDS-TWR réutilise les données temporelles collectées pour la mise en œuvre de SSDS-TWR. Notamment, les durées de création des messages sont conservées. Deux nouvelles valeurs apparaissent également : il s'agit des deux *timers* activés par le mobile et qui définissent les deux phases de la localisation. Le choix des valeurs à affecter à ces *timers* se fera par approximations successives lors de l'exécution du protocole, en prenant garde à ne provoquer ni collision ni perte de la réponse de la dernière ancre de la liste. Mathématiquement, ces valeurs peuvent s'exprimer en fonction du nombre n de réponses attendues et d'un temps élémentaire T_e (équation 61).

$$Timer_i = T_{e,i} * n \quad (61)$$

Nous pouvons donc résumer l'occupation des états de transmission et de réception des différents nœuds impliqués suivant le tableau 13. Les ancres peuvent passer en mode *Stand-by* une fois leur trame *Reply1* émise si le délai entre la fin de la transmission et la fin de *Timer1* est suffisant pour exécuter les deux transitions (endormissement puis réveil) et conserver l'état d'économie d'énergie, même pour un temps très court. L'expression de la durée de l'occupation par les ancres de ce mode est donnée dans la table 3 avec o la position dans la liste et n le nombre d'ancres partenaires.

Tableau 13 : Répartition du temps entre émission et réception pour PSDS-TWR

Nœud	Durée de réception	Durée d'émission	Economie d'énergie
Mobile	Timer1+Timer2	$T_{Start} + T_{Data_Request}$	n/a
Ancre	$T_{Start} + T_{Data_Request} + padding$	$T_{Reply1} + T_{Reply2}$	$N * T_{e,1} - (n-o) * T_{e,1} + nT_{e,2} - (n-o) * T_{e,2}$

3.1.5.3. Analyse de la consommation

Dans cette section, forts de la connaissance relative aux états empruntés par l'équipement durant le processus de mesure de distance, nous allons comparer la consommation énergétique

induite par l'exploitation de SSDS-TWR et de PSDS-TWR. Nous exploiterons également les données fournies par le constructeur du matériel.

Le tableau 14 rappelle les durées de l'occupation de chaque état pour les deux types de nœuds formant notre réseau et les deux protocoles à comparer. Pour chaque protocole, la mesure de distance implique n ancrs, la variable o représente la place d'une ancre dans la séquence.

Bien que la définition de PSDS-TWR fasse appel à des trames de taille variable, en moyenne, elles sont réduites par rapport à celles utilisées dans SSDS-TWR. Pour étudier le pire cas, nous considèrerons que les trames utilisées par PSDS-TWR et SSDS-TWR sont identiques. Nous réécrivons aussi les durées des *timers* en fonction de la durée de transmission des trames et du nombre d'ancres. Ces décisions mènent au système d'équations 62.

$$T_{Start} = T_{Reply1} = T_{Reply2} = T_{Data_Request} = T \quad (62)$$

$$Timer1 = Timer2 = n * T$$

Tableau 14 : Résumé des durées pour les protocoles SSDS-TWR et PSDS-TWR

Protocole Type	SSDS-TWR		PSDS-TWR	
	Mobile	TX	$(T_{Start} + 2 * T_{ACK}) * n$	TX
RX		$(T_{Reply1} + T_{Reply2} + 2 * padding) * n$	RX	Timer1 + Timer2
Eco		n/a	Eco	n/a
Ancre	TX	$T_{ACK} + T_{Reply1} + T_{Reply2}$	TX	$T_{Reply1} + T_{Reply2}$
	RX	$T_{Start} + padding + 2 * T_{ACK} + (n - o) * SSDS-TWR_{durée\ moyenne}$	RX	$T_{Start} + T_{Data_Request} + padding$
	Eco	n/a	Eco	$N * T_{e,1} - (n - o) * T_{e,1} + n T_{e,2} - (n - o) * T_{e,2}$

L'expression de la consommation pour chacun des protocoles est donnée par les équations 63 et 64.



$$C_{SSDS-TWR} = P_{TX} * (nT + 2nT_{ACK} + n(2T + T_{ACK})) + P_{RX} * ((2T + padding) * n + (T + padding + 2T_{ACK}) * n + \frac{n(n-1)}{2} * SSDS) \quad (63)$$

$$= P_{TX} * 3n(T + T_{ACK}) + P_{RX} * (3nT + 3n * padding + 2nT_{ACK} + \frac{n(n-1)}{2} * SSDS)$$

$$C_{PSDS-TWR} = P_{TX} * (2T + 2nT) + P_{RX} * (2nT + 2nT + n * padding) + P_{eco} * nT(n+1) \quad (64)$$

$$= P_{TX} * 2T * (1+n) + P_{RX} * (4nT + n * padding) + P_{eco} * nT(n+1)$$

Le paramètre P_{eco} est donné par la tension aux bornes du module communicant en mode économie (eco) d'énergie et le courant le traversant. Ce courant peut être calculé comme la somme des courants consommés par le *transceiver* en mode *StandBy* (tableau 5) et celui du microcontrôleur. A partir de la documentation technique, nous avons déterminé un courant total maximal de 6mA pour cet état (tension aux bornes du module dans la même configuration 2,475V).

A partir de ces informations, nous avons étudié la consommation induite par l'exécution de chacun des protocoles de localisation en fonction du nombre d'ancres impliquées. Les figure 56, 57 et 58 présentent ces résultats. La courbe associée à un marqueur en forme de losange  correspond aux résultats de SSDS-TWR tandis que le motif de sablier  permet de reconnaître les prédictions de PSDS-TWR.

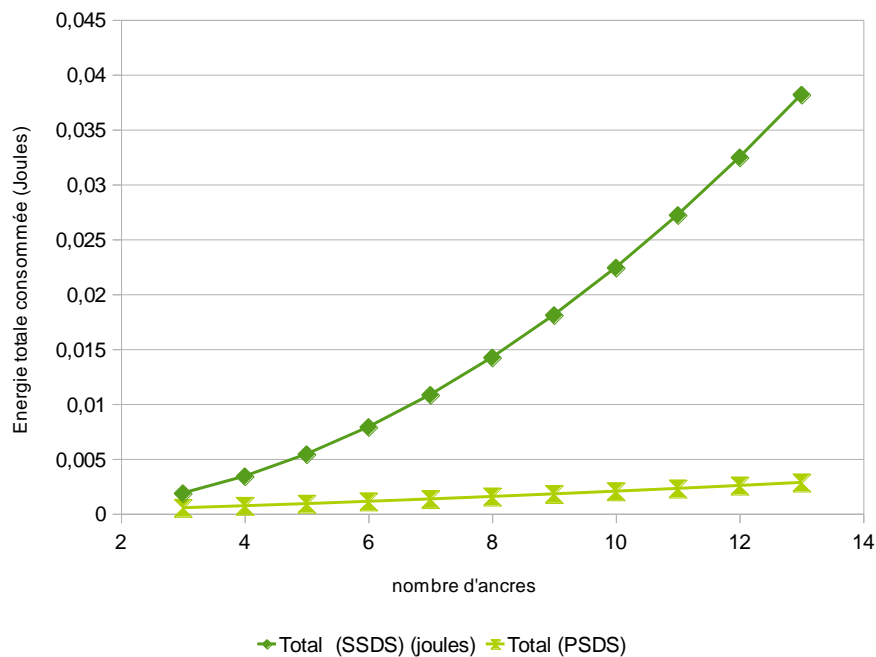


Figure 56 : Evolution de la consommation totale selon le nombre d'ancres et le protocole

La figure 56 donne une vue générale de la réduction de la consommation : le point de dépense énergétique principal étant la réception, une diminution drastique de la durée de cet état a permis de ramener la consommation en dessous de 5mJ par cycle protocolaire. On notera que le gain, c'est-à-dire l'écart entre les deux courbes vertes, augmente avec le nombre d'ancres.

La figure 57 se concentre sur l'amélioration de la situation au niveau du mobile. Dans ce cas-ci, le gain provient de la réduction du nombre de trames émises donc du temps passé en transmission. En effet, on peut observer que les droites correspondant à la réception et au mode eco pour les deux protocoles sont pratiquement superposées tandis que les droites illustrant la contribution de la transmission sont faciles à distinguer.

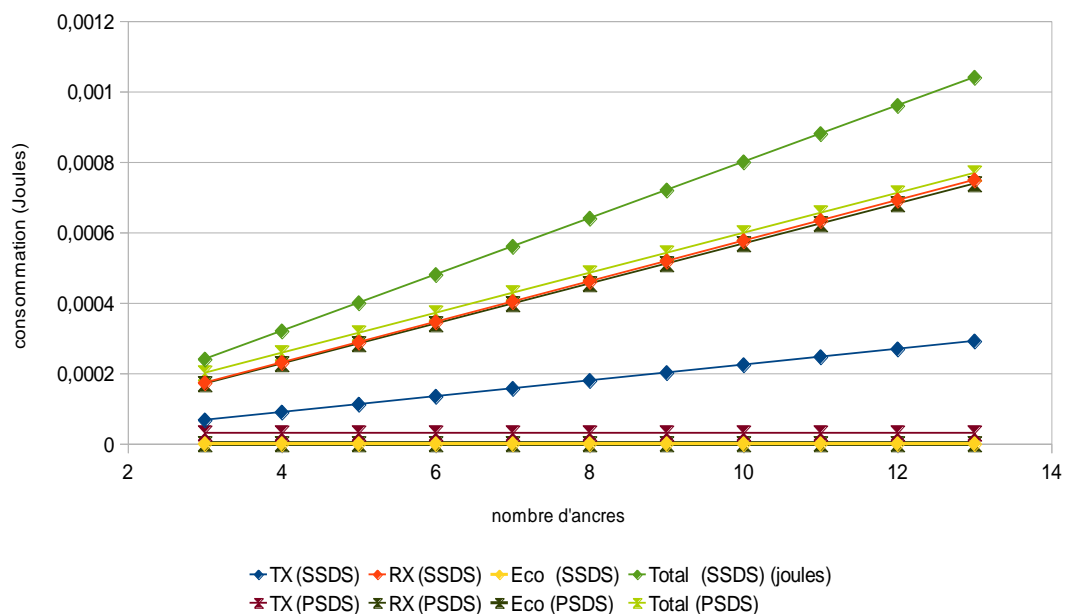


Figure 57 : Evolution de la consommation du mobile selon le nombre d'ancres et le protocole

Finalement, la figure 58 révèle que l'économie d'énergie significative observée en figure 56 est due aux ancres. Ces dernières passant moins de temps en réception durant l'exécution de PSDS-TWR, leur consommation est réduite par rapport à SSDS-TWR.

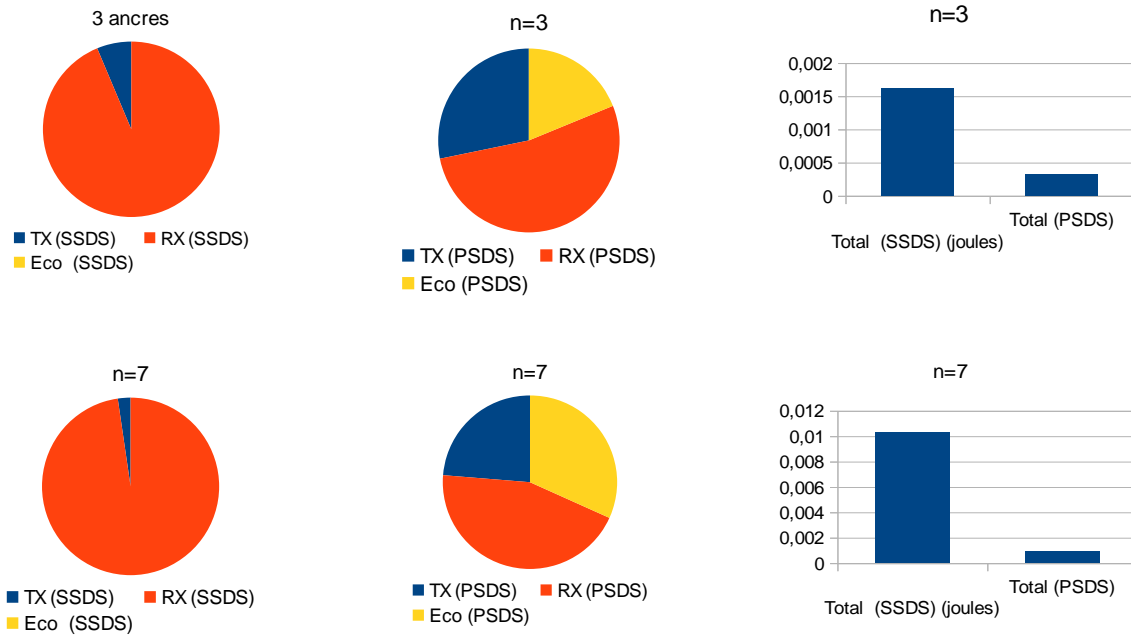


Figure 58 : Evolution de la consommation des anres selon le nombre d'anres et le protocole

D'une manière générale dans les réseaux de capteurs, il peut paraître inadéquat de favoriser la survie des anres, celles-ci étant généralement considérées comme alimentées. Mais si on se place dans la perspective de localisation collaborative où un mobile localisé peut aider d'autres mobiles à calculer leur position, cet effet du protocole ne peut être qu'intéressant pour la longévité globale du réseau.

3.1.6. Conclusion

La section 3.1 a introduit nos propositions en ce qui concerne le protocole de mesure de distance. Si SSDS-TWR apparaît comme le successeur naturel de SDS-TWR, PSDS-TWR se positionne comme la meilleure proposition en termes de performances. En effet, il s'engage à réduire la charge réseau liée à la localisation mais aussi la consommation énergétique. Un ordonnancement basique des anres est proposé pour l'agencement temporel des réponses. Dans le futur, cet aspect pourra être revu de manière à optimiser la durée d'exécution. En ce qui concerne la durée d'exécution, un modèle sera proposé à la section 4.2.6.3 et paramétré en fonction des résultats de la simulation.

3.2. Proposition d'un algorithme de localisation

Réaliser la mesure de distance avec n anres n'est que la première moitié du travail de localisation. En effet, un algorithme de calcul de position doit à présent prendre la relève et

Proposition d'un algorithme de localisation

fournir une estimation fiable de la position à partir des données de référence. En plus de sa capacité à déterminer correctement la position, cet algorithme devra également satisfaire des contraintes de coût, notamment en termes de ressources exigées. Pour les plateformes visées, cette efficacité est une condition *sine qua non* d'implémentabilité.

Dans cette section, nous présenterons la méthode de calcul utilisée dans le cadre de nos travaux.

3.2.1. *iRingLA*

La forme complète de l'acronyme *iRingLA* est *inter Ring Localization Algorithm*. Elle implique que notre algorithme calcule l'intersection d'anneaux en vue d'identifier la position du mobile. Les paragraphes suivants décriront le passage des données brutes, obtenues par l'exécution de l'un des protocoles introduits à la section précédente, aux anneaux puis à la position et aborderont différentes problématiques liées au traitement de ces données comme l'ordre d'inclusion des anneaux.

3.2.1.1. Déroulement

L'algorithme *iRingLA* se base sur les mesures réalisées pour créer des anneaux centrés sur les ancres puis identifie l'intersection de ces anneaux ayant le plus de chances de contenir la position réelle du mobile. Une matrice de points est alors construite et les éléments appartenant à l'intersection sont qualifiés de candidats. Finalement, à partir des positions candidates retenues, une estimation de la position est réalisée. Ces différentes étapes seront détaillées à travers les sections suivantes.

3.2.1.1.1. Création des anneaux

Dans la littérature, il est admis que la mesure de la distance à partir du temps de vol dans un environnement *intérieur*, à l'aide d'un signal à bande limitée, a très peu de chances de fournir une estimation exacte de la distance. Heureusement, des outils perfectionnés (§ 1.3.1.6) permettent de retrouver les informations originelles à partir de données bruitées. Dans notre cas, ces moyens étant inadaptés (§ 1.4.2 et 1.4.3), nous avons choisi de représenter ces distances connues du mobile par un anneau construit de la manière suivante : soient R_{in} et R_{out} les rayons intérieur et extérieur de l'anneau centré sur le point $P(x,y)$. Soit par ailleurs d_{moy} , la distance moyenne du mobile à la balise B positionnée en P et e , une erreur moyenne caractérisée durant l'étalonnage de l'instrument de mesure.

Dans la réalité, d_{moy} et e étant les données connues du mobile, on écrira (équation 65).

$$\begin{aligned} R_{out} &= d_{moy} + e \\ R_{in} &= d_{moy} - e \end{aligned} \quad (65)$$

Plus loin dans ce document, nous indiquerons la valeur ainsi que l'origine de ce paramètre d'erreur e .

3.2.1.1.2. Création de la matrice de départ

Une fois les anneaux définis, il est temps de créer la matrice recouvrant l'intersection. La recherche d'une solution par tests des éléments d'une matrice, si elle a l'avantage d'être simple à implémenter, présente divers risques comme la non détection due à la relation entre les géométries de l'intersection et du quadrillage (la résolution n'est pas assez fine par rapport aux dimensions de l'intersection).

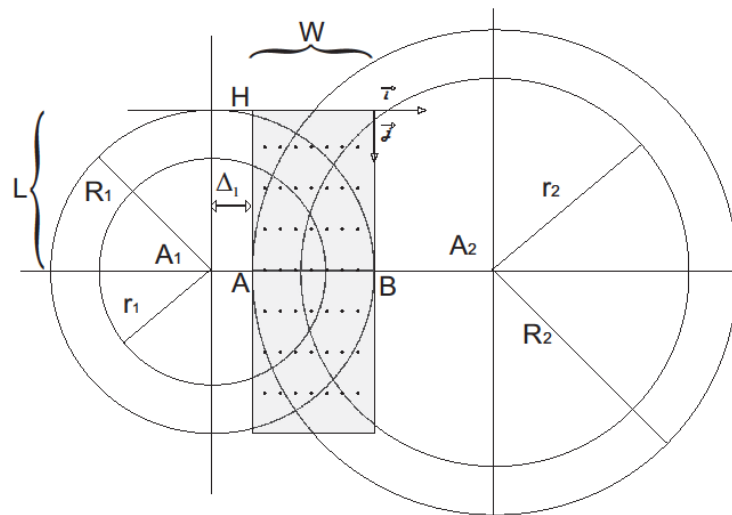


Figure 59 : Relation entre la matrice et les anneaux

Il est donc nécessaire de construire la matrice la plus dense possible, donc d'augmenter le nombre de lignes et/ou de colonnes. Mais ceci n'est possible qu'au détriment de la mémoire. Afin de minimiser le nombre de points à stocker, une matrice de superficie réduite sera définie pour le même nombre de points : ces derniers étant concentrés sur une surface plus faible, la résolution s'en trouvera améliorée.

Nous distinguerons dans la suite trois cas généraux dénommés A, B et C. La figure 60 indique les différentes variantes pour chaque situation tandis que le tableau 15 liste les critères permettant de les reconnaître.

Proposition d'un algorithme de localisation

On désignera comme anneau 1 le plus petit des deux et anneau 2 le plus grand. Les variables suivantes seront utilisées :

R_i : rayon extérieur de l'anneau i

r_i : rayon intérieur de l'anneau i

d : distance entre les centres des anneaux

Pour chacun de ces cas, un certain nombre de points d'intersection peut être calculé. Les cas A et C présentent entre 2 et 6 points tandis que 8 points sont obtenus dans le cas B (figure 60). Dans le cas B, la répartition des points permet de définir une matrice recouvrant l'intersection et incluant un minimum de surface inutile dans laquelle aucun point candidat ne pourrait se trouver. Pour les instances des cas A et C, les informations fournies par les points d'intersection seront complétées par les relations entre les anneaux. Le processus de création de la matrice a donc la forme illustrée par la figure 61.

Tableau 15: critères d'identification du type d'intersection

Cas	Description	Critères
A	Intersection mono-surface en amande	$R_1+R_2>d$ $R_i<d$ $r_1+r_2<d$
B	Intersection en double surface	$R_1+R_2>d$ $r_1+r_2>d$ $R_2<d+r_1$
C	Intersection mono-surface en croissant	$R_2> d+r_1$

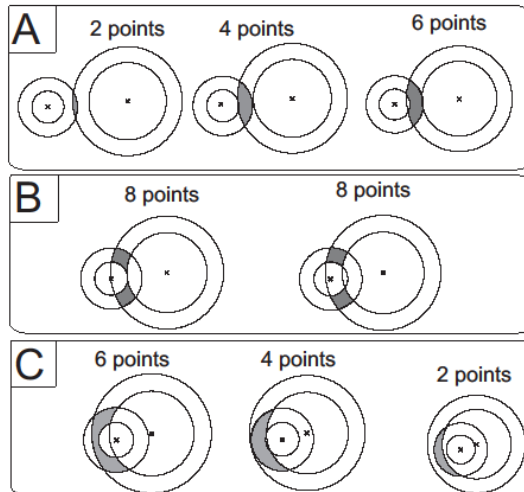


Figure 60 : Différentes formes de l'intersection de deux anneaux

Une fois les coordonnées des points A et H obtenues (figure 59), les éléments de la matrice sont initialisés. Afin de faciliter le processus, un nouveau repère est défini par le point A et les vecteurs libres \vec{i} et \vec{j} . Le vecteur \vec{i} est colinéaire à \overline{AB} et le vecteur \vec{j} est colinéaire à \overline{AH} .

Soient W la largeur de la matrice, $2L$ sa hauteur et m le nombre de lignes (et de colonnes) prévues. Deux nouvelles variables dh et dc sont définies (équation 66).

$$\begin{aligned} dh &= \frac{2L}{m} \\ dc &= \frac{W}{m} \end{aligned} \tag{66}$$

A partir de l'ensemble de ces informations, les coordonnées du point H dans le nouveau repère sont calculées. A partir de ces deux valeurs, l'ensemble des points de la matrice est généré par l'algorithme de la figure 62. Finalement, les points sont exprimés dans le repère global (§ 1.1.1).

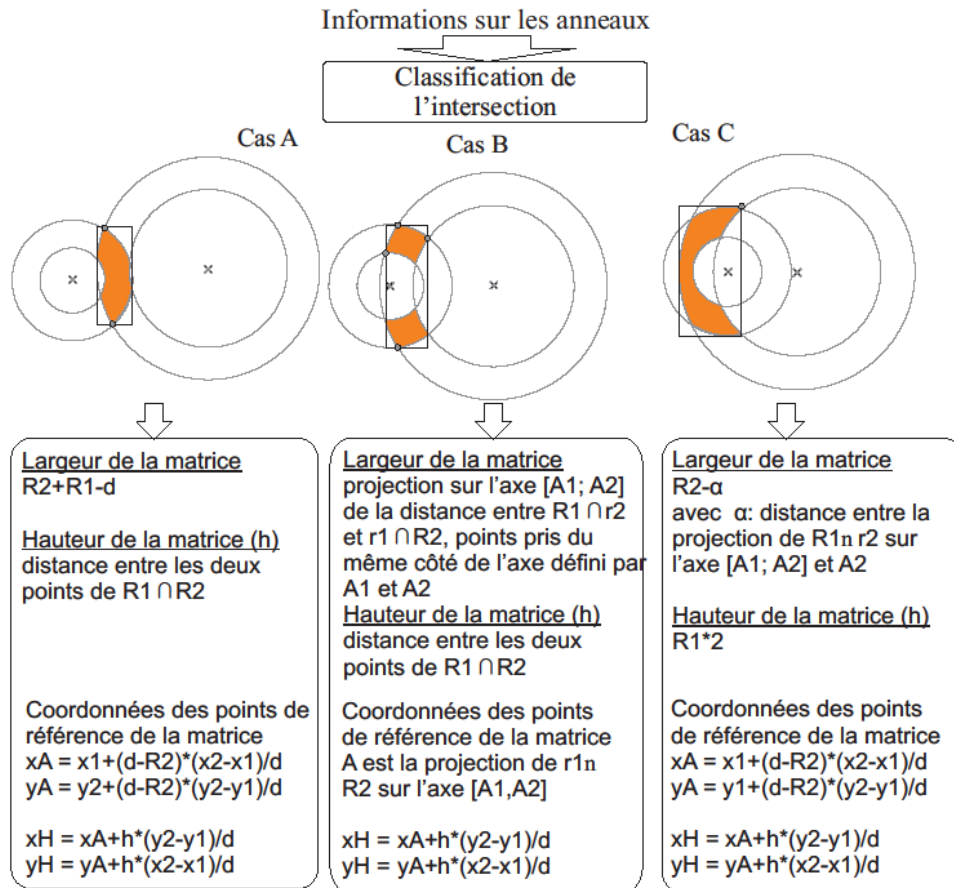


Figure 61 : Création de la matrice

```

for(line=1;line<this.nlines;line++){
  for(col=1;col<=this.ncols;col++){
    points[line][col].ordonnee=H.ordonnee-line*dh;
    points[line][col].abcisse=H.abcisse+col*dc;
    points[line][col].currentBase=b1;
  }
}
    
```

Figure 62 : Initialisation des points de la matrice

Les deux changements de repère permettent de simplifier l'écriture de l'algorithme car dans le cas général, les centres des ancrs ne forment pas un vecteur colinéaire à un axe du repère global. Typiquement, leur disposition sera celle indiquée par la figure 63. On observe que pour former la même matrice, il est nécessaire d'exprimer les vecteurs \vec{i} et \vec{j} en fonction de \vec{e}_1 et \vec{e}_2 , ce qui a pour effet d'alourdir l'écriture.

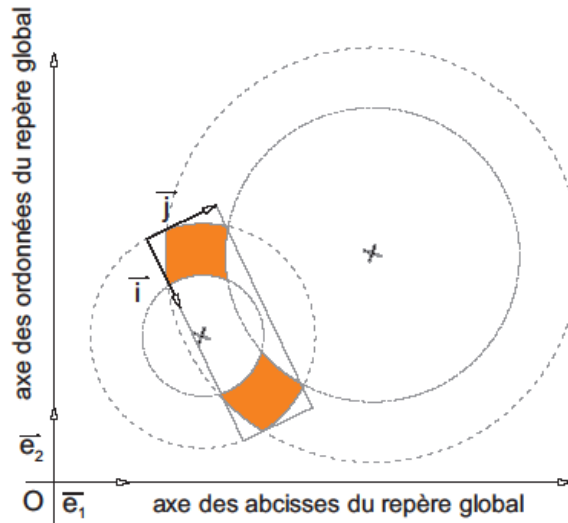


Figure 63 : Changement de repère

3.2.1.1.3. Recherche des points candidats

La fin de la section précédente coïncide avec le début du test des points de la matrice. L'objectif est d'identifier, parmi les éléments du quadrillage de points, ceux appartenant à l'intersection des anneaux. La question est simple : étant donné un point (x,y) , ses coordonnées le placent-elles, pour chacun des anneaux, à l'extérieur de l'anneau intérieur et à l'intérieur de l'anneau extérieur ? Si la réponse est oui, le point candidat fait partie de l'intersection.

La recherche linéaire est la méthode la plus simple pour répertorier les points à conserver ; mais si le test doit être répété pour k anneaux sur la même matrice, les $m*m$ éléments de la matrice carrée seront tous traités k fois. Bien que les contraintes matérielles limitent la taille des structures de données employées, la durée associée à ces répétitions n'est pas négligeable. Plus loin, nous investiguerons différentes pistes permettant de réduire le nombre de points traités par notre algorithme.

3.2.1.2. Durée d'exécution

Notre algorithme comporte les blocs suivants :

- **Création de la matrice** La matrice est parcourue intégralement à deux reprises, pour l'initialisation et le changement de base
- **Test sur les deux premiers anneaux** Toute la matrice est traitée.
- **Test pour chaque anneau subséquent** L'ensemble des points est analysé.

- **Récupération de tous les points ayant réussi les tests et combinaison pour trouver une estimation de la position** Cette phase finale nécessite de nouveau un passage dans la matrice.

On observe donc que la matrice de $m*m$ éléments est un élément clé. Elle est parcourue $k+4$ fois. Si les trois premières peuvent être considérées comme essentielles, il serait intéressant de réduire la charge au fur et à mesure que des anneaux sont intégrés. D'autre part, il ne faut pas perdre de vue le fait que, suivant le nombre de surfaces distinctes formant l'intersection, un sous-ensemble des points peut être qualifié d'inutile puisque que ces points ne feront pas partie de la solution. Eviter de manipuler ces éléments peut être une stratégie d'optimisation de l'algorithme. La section suivante abordera les différentes pistes examinées.

3.2.1.3. Complexité et optimisations

L'une des pistes que nous avons choisi de suivre consiste à réduire le nombre de points traités par notre algorithme, ceci permettant de diminuer la durée de l'exécution.

Dans un premier temps, nous avons voulu éviter de générer tous les points de la matrice et avons simplement défini deux points, dits *représentants*. Chacun d'eux est représentatif d'une des deux moitiés de la matrice. Ces deux points *Top* et *Bottom* (figure 64(a)) ont donc été calculés en fonction de A, W et L. La procédure à ce moment est simple : l'appartenance des points *Top* et *Bottom* au troisième anneau est déterminée et la recherche des points candidats est restreinte à la moitié correspondant au succès du test des points représentants. Sur l'exemple de la figure 64(a), la recherche de la position estimée du mobile s'exécutera au voisinage de *Bottom*. Ceci, malheureusement, est le cas idéal : l'un des représentants est contenu dans le troisième anneau et il est proche de la position réelle du mobile, M. Dans la réalité, la situation se présente plutôt comme celle dépeinte par la figure 64(b) : aucun des représentants n'appartient au troisième anneau. Nous avons alors adapté notre algorithme en choisissant la moitié dont le représentant est le plus proche de la troisième ancre. Par un choix judicieux des ancres partenaires, la situation où *Top* et *Bottom* sont équidistants de la troisième ancre peut être évitée. Les résultats sont défavorables une fois sur deux, la figure 64(b) illustrant cette situation. Dans ce cas, la moitié correspondant à *Top* sera systématiquement retenue. Lorsque M est proche de *Top*, la position estimée sera aussi proche de la réalité. Mais quand la vraie position sera proche de *Bottom*, comme cela se produit pour M', l'erreur sera non négligeable.

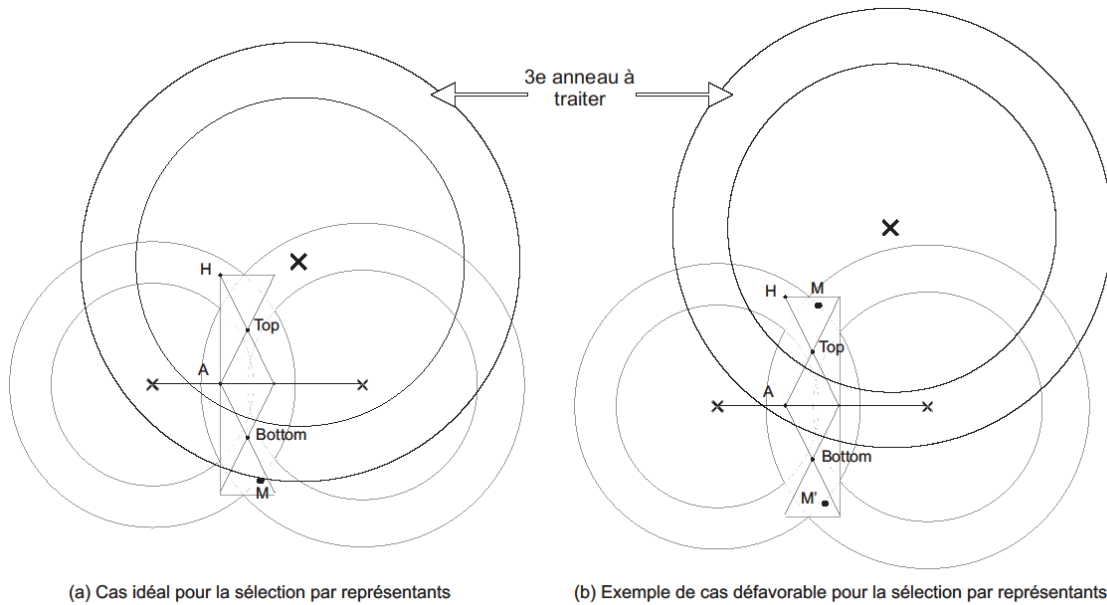


Figure 64 : Exemple de sélection à partir des représentants

Nous avons testé également l'ajout de maintes contraintes mais sommes parvenus aux conclusions suivantes :

- du fait de l'impossibilité pour le mobile de prédire s'il est réellement proche de *Top* ou de *Bottom*, sélectionner une surface en fonction de ces deux représentants introduit un risque d'élimination complète de la zone correcte de recherche. Cette incapacité à sélectionner provient du fait que peu de données sont disponibles : en effet, l'exécution du protocole de *ranging* fournit, au plus, deux estimations de la distance. Il est donc difficile pour le mobile d'identifier laquelle est la plus plausible,
- il est possible de définir une démarche de test permettant à la méthode basée sur les représentants de fournir un résultat acceptable sur un certain sous-ensemble des points de tests, mais ceci a un coût : l'algorithme se complexifie grandement car il s'emploie à faire face à une succession de cas particuliers.

Nous avons donc décidé de changer de méthode et de tenter de réduire le nombre de points pour les tests liés à l'ancre n , $n=3, 4, 5 \dots$. Notre choix s'est porté sur la technique du tri à bulles : durant le test impliquant les deux anneaux les plus petits, tous les points ayant réussi ce test sont déplacés vers le haut de la matrice. Leur nombre p est comptabilisé de sorte que, pour tous les anneaux subséquents, le test commence par le haut de la matrice et ne tienne compte que des p premiers éléments. En termes de temps d'exécution, une nette amélioration a été observée avec une réduction de 623ms à 430ms, soit un gain de 30%.

Cette solution présente tout de même un risque puisque l'algorithme se limite aux points retenus par le premier test. La décision reste pourtant pertinente puisque ces deux anneaux sont généralement les plus fiables.

En termes de précision, l'application de ce filtrage n'a rien changé car le déplacement des points dans la matrice n'influence par leur position dans le repère.

3.2.1.4. Ordonnement des anneaux

Si la réduction du nombre de points traités tend à améliorer les performances de notre algorithme en termes de durée d'exécution, il nous a fallu ajouter une phase supplémentaire afin de le rendre robuste face aux aléas de la mesure réelle. Sur la figure 64(a), la situation idéale est représentée : deux anneaux ont servi de base pour la construction de la matrice et le troisième présente une intersection avec cette même matrice. Dans la réalité, ceci est exceptionnel. Nous faisons plus souvent face à des configurations telles que celle décrite par la figure 65 : le mobile M a calculé trois anneaux centrés sur les ancrs A1, A2 et A3. Si la matrice est construite sur l'intersection de A2 et A3, l'ancre A1 n'apportera aucune contribution et la position finale sera très éloignée de l'emplacement réel de M. Si la paire d'ancres A1 et A2 est utilisée comme point de départ, l'ancre A3 ne permettra pas de raffiner les résultats mais la position sera nettement plus proche de M. Se pose alors le problème pour le mobile de décider de la paire d'anneaux à exploiter pour bâtir sa matrice de candidats.

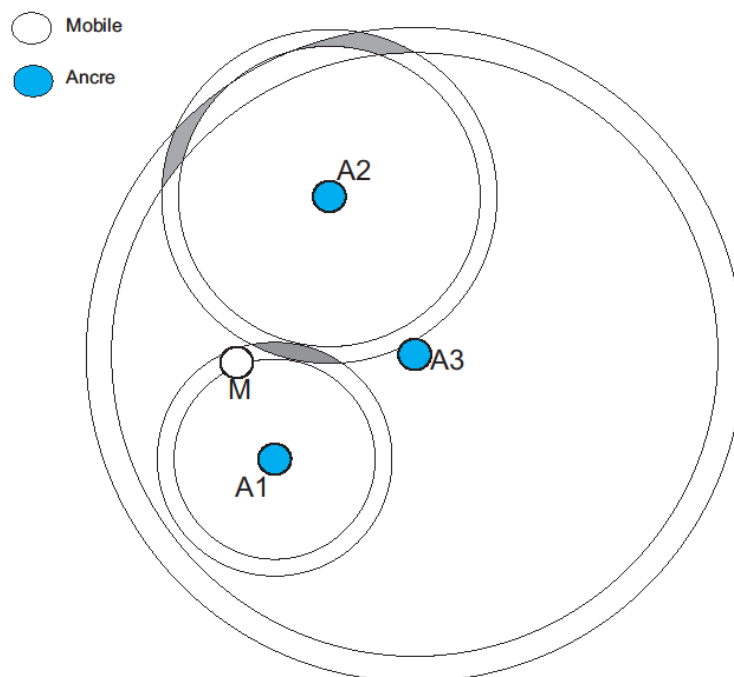


Figure 65 : Modèle de résultat réel de mesure de distances par SSDS-TWR

Pour munir le mobile des outils permettant de discerner les anneaux les plus fiables, nous avons étudié la probabilité qu'un anneau soit correct. Cette probabilité correspond à l'expression 67.

$$P(R_{in} \leq \text{distance et } R_{out} \geq \text{distance}) \quad (67)$$

R_{in} et R_{out} désignent respectivement les cercles intérieur et extérieur de l'anneau. Soit x l'estimation de la distance du mobile à l'ancre fournie par l'outil de mesure, l'expression peut être écrite comme l'équation 68.

$$P(d - c \leq x \leq d + c), \quad c = cte \quad (68)$$

Afin de pouvoir calculer cette probabilité, nous avons caractérisé le comportement réel de la variable aléatoire x , pour notre plateforme basée sur la technologie CSS. Une campagne de mesures a été menée pendant laquelle 1000 échantillons ont été recueillis pour chaque distance (figure 66). Nous avons fait déplacer le mobile le long d'une droite en occupant une vingtaine de positions différentes. Les résultats indiquent que x suit approximativement une gaussienne dont la moyenne μ et l'écart-type σ dépendent de la vraie distance.

On peut donc écrire l'expression 69 :

$$P(d - c \leq x \leq d + c) = P(a \leq x \leq b) = P\left(\frac{a - \mu}{\sigma} \leq y \leq \frac{b - \mu}{\sigma}\right) = \Phi\left(\frac{b - \mu}{\sigma}\right) - \Phi\left(\frac{a - \mu}{\sigma}\right) \quad (69)$$

avec $\Phi()$ la fonction de densité d'une normale centrée réduite. En étudiant le signe de la dérivée par rapport à la distance de cette expression, nous pourrions savoir comment varie la probabilité que l'anneau soit correct.

Commençons par réécrire l'expression 69 en la représentant par $u(d)$ (équation 70).

$$u(d) = \Phi\left(\frac{b - \mu(d)}{\sigma(d)}\right) - \Phi\left(\frac{a - \mu(d)}{\sigma(d)}\right) \quad (70)$$

La dérivée $u'(d)$ peut s'écrire :

$$u'(d) = \frac{\delta}{\delta d} \left(\int_{\frac{a - \mu(d)}{\sigma(d)}}^{\frac{b - \mu(d)}{\sigma(d)}} \frac{1}{\sqrt{2\pi}} e^{-\frac{t^2}{2}} dt \right) \quad (71)$$

Soit $A(d)$ et $B(d)$ tels que :

Proposition d'un algorithme de localisation

$$A(d) = \frac{a - \mu(d)}{\sigma(d)} \quad \text{et} \quad B(d) = \frac{b - \mu(d)}{\sigma(d)} \quad (72)$$

Sur la figure 67, $\mu(d)$ correspond à la ligne brisée (jaune). La première bissectrice, en bleu, ainsi qu'une droite de pente $p=2$, en rouge, sont également représentées. Nous choisirons d'approcher $\mu(d)$ par l'expression :

$$\mu(d) \approx k * d, k \geq 0 \quad (73)$$

Afin de simplifier le calcul, nous considérerons $\sigma(d)$ comme une constante positive. La dérivée peut alors s'écrire :

$$u'(d) = \left(-\frac{k}{\sigma}\right) \left(f\left(\frac{b - k * d}{\sigma}\right) - f\left(\frac{a - k * d}{\sigma}\right) \right) \quad (74)$$

Avec $f(t) = \varphi(t)$.

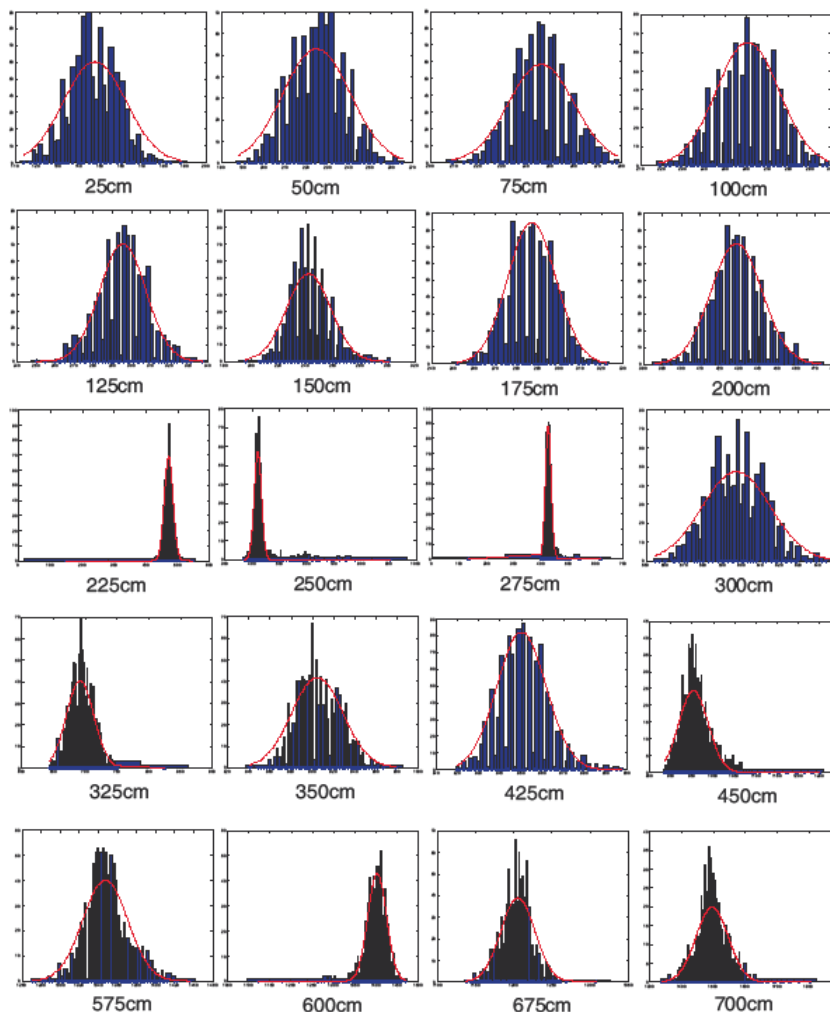


Figure 66 : Caractéristiques de la mesure de temps de vol par CSS en intérieur et en ligne de vue.

Le signe de $f\left(\frac{b-k*d}{\sigma}\right) - f\left(\frac{a-k*d}{\sigma}\right)$ déterminera celui de la dérivée.

On se rappelle que :

$$a = d - c \quad (75)$$

et

$$b = d + c \quad (76)$$

On a donc :

$$u'(d) = \left(-\frac{k}{\sigma}\right) \left(f\left(\frac{d+c-k*d}{\sigma}\right) - f\left(\frac{d-c-k*d}{\sigma}\right) \right) = \left(-\frac{k}{\sigma}\right) \left(f\left(\frac{(1-k)d+c}{\sigma}\right) - f\left(\frac{(1-k)d-c}{\sigma}\right) \right) \quad (77)$$

Si k est égal à 1, le signe de la dérivée est nul et l'étude est inutile car la moyenne des mesures suit parfaitement la réalité.

Dans notre cas, k est supérieur à 1, et le problème revient à étudier la relation entre $\left(\frac{(1-k)d+c}{\sigma}\right)$ et $\left(\frac{(1-k)d-c}{\sigma}\right)$. Les termes k , d , c et σ étant positifs, $\left(\frac{(1-k)d-c}{\sigma}\right)$ est négatif. On se rappellera que f correspond à la densité de probabilité d'une normale centrée réduite. Par conséquent, $\left(f\left(\frac{(1-k)d-c}{\sigma}\right)\right)$ se trouve du côté croissant de la cloche de Gauss (figure 68). Nous désignerons l'emplacement de cette valeur par L .

Si

$$\left(\frac{(1-k)d+c}{\sigma}\right) > |L| \quad (78)$$

la dérivée sera positive. En remplaçant k , c et σ par les valeurs choisies pour notre plateforme matérielle, nous obtenons la courbe de la figure 69 pour laquelle :

$$m(d) = \frac{(1-k)d+c}{\sigma} \quad (79)$$

et

$$n(d) = \left| \frac{(1-k)d-c}{\sigma} \right| \quad (80)$$

Proposition d'un algorithme de localisation

Quelle que soit la valeur de d , on constate que $m(d)$ est inférieur à $n(d)$ donc le signe de la dérivée est négatif. Ceci implique que la probabilité qu'un anneau soit correct décroît avec la distance et appuie la stratégie de construire la matrice à partir des plus petits anneaux d'abord.

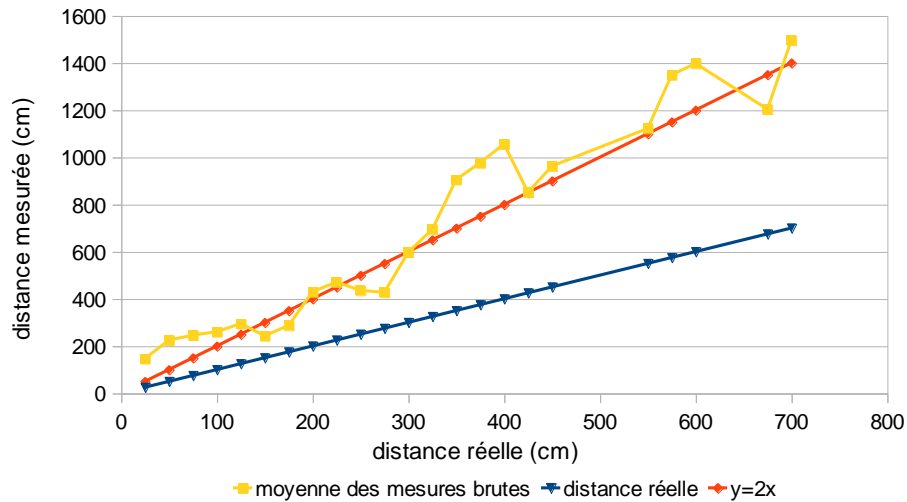


Figure 67 : Evolution de la moyenne des échantillons avec la distance

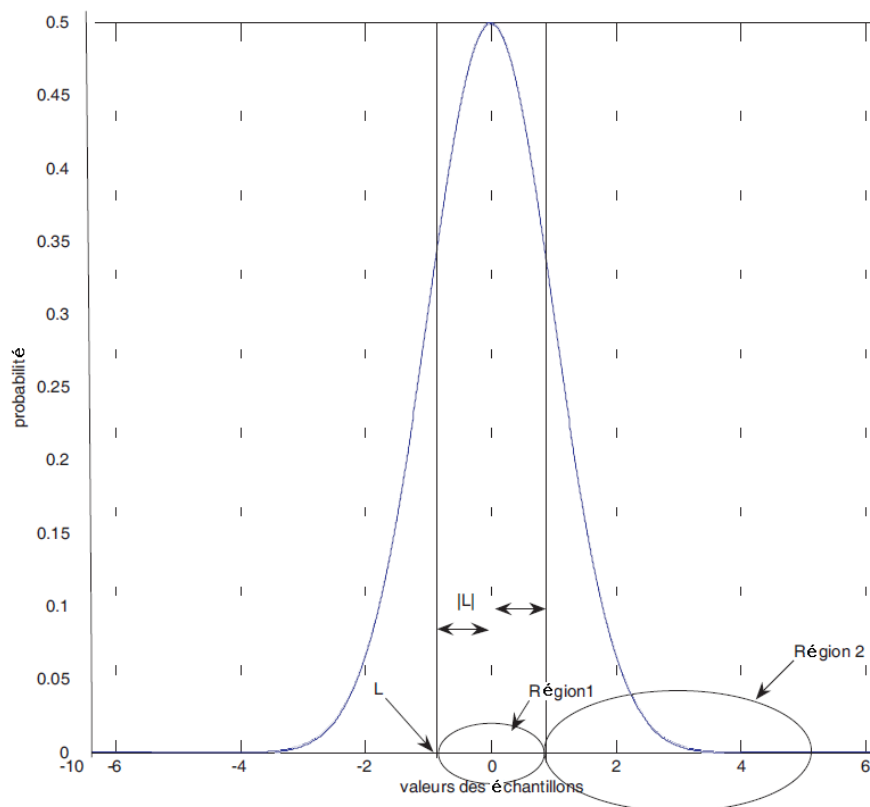


Figure 68 : Courbe de Gauss et régions

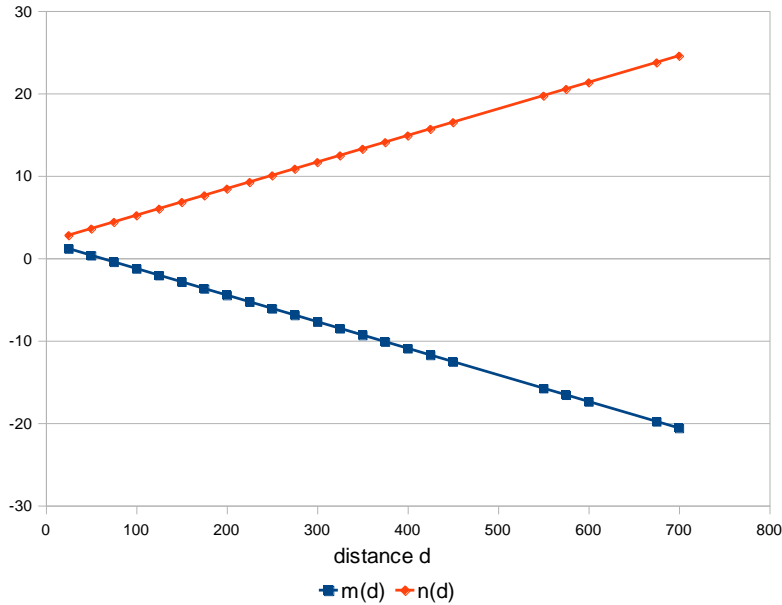


Figure 69 : Evolution de $n(d)$ et $m(d)$ avec la distance

En vue d'évaluer les performances de cette proposition d'ordonnement des anneaux, nous l'avons appliquée à des données réelles collectées de la manière suivante : une session de localisation a généré 5 anneaux par ancre pour 21 positions distinctes du mobile. Dans le cas du prototype, nous n'avons pas implémenté l'ordonnement. L'erreur sur la position ainsi obtenue est répertoriée et comparée à celle indiquée après traitement des anneaux par un programme implémentant leur ordonnancement. Sur la figure 70, les données représentées en jaune proviennent du prototype (sans stratégie particulière d'ordonnement) tandis que celles correspondant aux barres rouges sont produites grâce à l'application de notre classement. Chaque segment correspond au résultat d'une expérience, l'ordre n'ayant pas de signification particulière.

Bien que l'erreur de localisation soit toujours présente, classer les anneaux améliore globalement les résultats. Dans la plupart des cas, la hauteur des barres rouges est inférieure à celle des jaunes : l'erreur est donc réduite par l'ordonnement.

Certains échantillons mettent pourtant cette correction en échec : ceci est dû au non respect des hypothèses de départ. En effet, il peut arriver que la vraie position ne soit pas contenue dans le plus petit des anneaux. Cet événement reste tout de même rare (en moyenne inférieur à 10%). L'ordonnement reste donc approprié dans la plupart des cas.

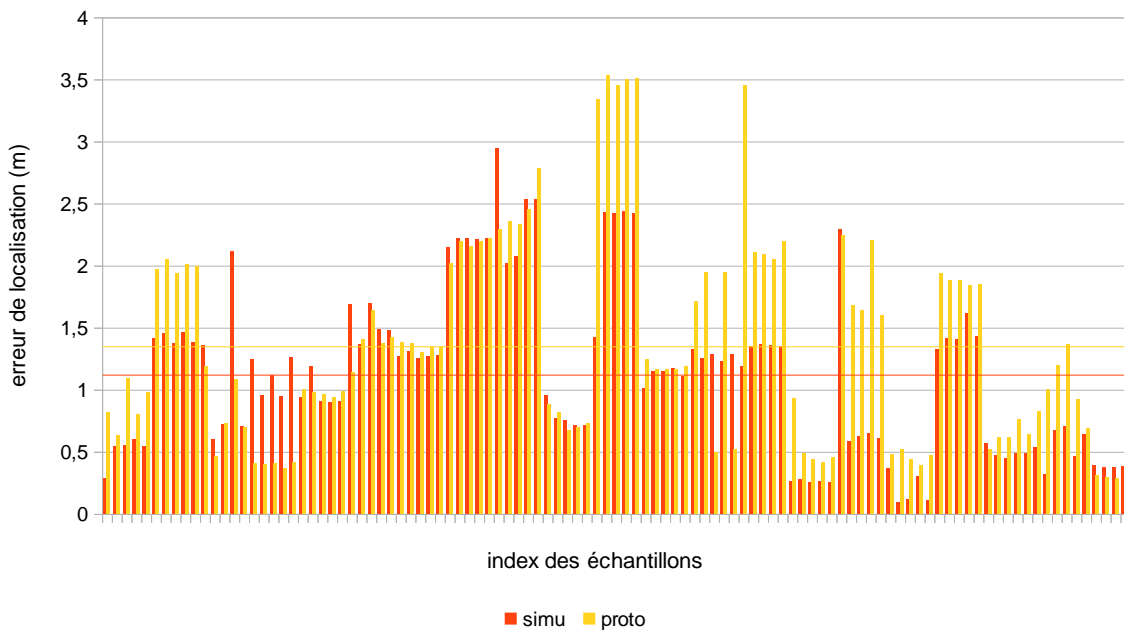


Figure 70 : Comparaison des erreurs avec et sans ordonnancement des anneaux

De même, lorsque l'ordre idéal et l'ordre par défaut coïncident, l'avantage de l'ordonnement disparaît. Les deux points sont alors confondus.

Finalement, bien que la moyenne du gain, illustrée par la distance entre les deux droites horizontales sur la figure 70 soit assez faible, dans les cas où l'ordonnement a un rôle à jouer, une réduction importante peut être observée.

3.3. Transposition sur d'autres technologies

La simplicité présentée ici d'*iRingLA* peut être perçue par certains comme une faiblesse et ce, pour diverses raisons. Tout d'abord, les plateformes très intégrées et pourtant relativement puissantes existent [85], [86]. De plus, des technologies prometteuses telles que UWB émergent : en raison de la précision inhérente à ces dispositifs, un algorithme tel qu'*iRingLA* pourrait sembler inutile.

Nous nous proposons, dans cette section, d'analyser ces *a priori*.

Tout d'abord, en ce qui concerne l'évolution technologique des nœuds communicants, la situation peut être envisagée sous plusieurs angles. D'un point de vue économique, l'augmentation de la puissance de chaque module entraîne souvent une élévation du prix. Comme nous l'avons dit en introduction du manuscrit, utiliser des équipements onéreux pour

un réseau de capteurs, dont la vocation est de faire collaborer un nombre élevé de nœuds, rendrait inaccessible ces solutions dans la plupart des cas.

Du point de vue de la mission du réseau et donc des communications, il est nécessaire de garder à l'esprit que l'objectif est l'échange de données. Même en se situant en dehors du domaine des applications critiques, les communications liées au processus de localisation ne doivent pas dégrader le service réseau. Par conséquent, bien que le serveur de localisation unique puisse être remplacé par un sous-ensemble des ancres, l'impact protocolaire induit par cette solution répartie ne doit pas être négligé.

Finalement, il ne faut pas non plus oublier une des lois du monde informatique : quand la quantité de ressources disponibles croît, les attentes en termes de nombre de services rendus croissent également. Par conséquent, sur une plateforme plus puissante, il n'est pas garanti qu'une capacité de traitement plus importante soit dévolue à la localisation. A notre avis, il est plus probable qu'une multitude d'autres services applicatifs, plus proche de l'utilisateur soient ajoutés à la plateforme.

La question de la pertinence d'un algorithme aussi simple peut également être abordée d'un point de vue technologique. A la base, *iRingLA* a été élaboré pour permettre à des nœuds dont la capacité de calcul est limitée de déterminer leur position à partir de données de qualité moyenne, produites par la couche physique (typiquement CSS). L'utilisation d'anneaux, plutôt que de cercles, permet de tenir compte de l'erreur de mesure sans exploiter un modèle très complexe qui permettrait de compenser cette même erreur. Aujourd'hui, la perspective de composants radio capables de mesures quasi exactes laisse croire que des algorithmes tels *qu'iRingLA* n'ont plus lieu d'être. Il s'agit, selon nous, d'une position qui découle d'une incompréhension sur les promesses techniques des nouvelles plateformes : elles seront probablement plus précises car la dispersion des échantillons successifs pourra être réduite, ce qui ne signifie pas pour autant que la mesure retournée sera exacte. L'incertitude sur la distance réelle subsistera et donc, un moyen de calculer la position à partir de ces données restera nécessaire. Si l'on replace cette conclusion dans le contexte de l'avènement imminent de *l'Internet des objets*, le développement de solutions logicielles adaptées à des plateformes légères a encore de beaux jours devant lui.

EVALUATION DU PROTOCOLE ET DE L'ALGORITHME PROPOSES

Le chapitre 3 a été consacré à la présentation détaillée du protocole de mesure de distance ainsi qu'à celle de l'algorithme de détermination de la position proposés. Dans le chapitre 4 seront exposés la démarche d'évaluation de ces solutions ainsi que leurs résultats.

4.1.	Prototype	130
4.1.1.	Présentation de la plateforme matérielle	130
4.1.2.	Caractéristiques du <i>ranging</i> et corrections appliquées.....	131
4.1.3.	Résultats de localisation	134
4.1.3.1.	Précision et observations diverses.....	135
4.1.3.2.	Durée d'exécution du protocole	139
4.1.3.3.	Durée d'exécution de l'algorithme.....	139
4.1.4.	Conclusion.....	142
4.2.	Simulateur <i>DokoSim</i>	142
4.2.1.	Motivations.....	143
4.2.2.	Choix d'un simulateur	146
4.2.2.1.	NS-2 et NS-3	146
4.2.2.2.	OMNeT++	146
4.2.2.3.	WSNet+WSim.....	147
4.2.3.	Description technique.....	147
4.2.3.1.	Structure du simulateur	148
4.2.3.2.	Architecture de classes	148
4.2.3.2.1.	Classe <i>Clock</i>	149
4.2.3.2.2.	Classes <i>NodeAgenda</i> et <i>AgendaItems</i>	150
4.2.3.2.3.	Classe <i>WirelessMedium</i>	151
4.2.3.2.4.	Classe <i>WirelessNode</i>	152
4.2.3.2.5.	Classe <i>LocatorEngine</i>	154
4.2.4.	Choix d'implémentation.....	154
4.2.4.1.	Similitudes.....	155
4.2.4.2.	Différences	159
4.2.5.	Test de SSDS-TWR	161
4.2.5.1.	Paramétrage du <i>multipathFactor</i>	161
4.2.5.2.	Paramétrage du taux de pertes de trames	164
4.2.5.3.	Validité du simulateur	165
4.2.6.	Test de PSDS-TWR	165
4.2.6.1.	Simulation dans un environnement sans pertes.....	166
4.2.6.2.	PSDS-TWR face aux interférences	170
4.2.6.3.	Gain temporel de PSDS-TWR	171
4.2.7.	Conclusion sur PSDS-TWR	173
4.3.	Bilan sur le simulateur <i>DokoSim</i>	173

Après avoir présenté les bases de nos propositions, nous nous intéressons maintenant à leurs performances. Pour étudier celles-ci, nous ferons appel à un prototype et à un simulateur de notre réalisation, *DokoSim*. Sur le prototype, nous pourrions observer le comportement de notre algorithme de localisation, appliqué à des données réelles. L'exécution d'*iRingLA* dans un environnement significatif, c'est-à-dire non exempt de perturbations liées à la présence d'autres réseaux exploitant la même bande de fréquence mais aussi de multitrajet, nous fournira une indication sur sa robustesse.

Par la suite, nous décrirons le simulateur *DokoSim*, utilisé pour tester la solution globale incluant *iRingLA* et le protocole de localisation, et analyserons les résultats de PSDS-TWR implémenté dans cet environnement. L'impact protocolaire ayant été analysée théoriquement en section 3.1.4, nous nous focaliserons sur les aspects temporels de l'utilisation de notre protocole.

4.1. Prototype

Dans cette section, nous introduirons le dispositif utilisé pour l'évaluation de nos propositions. Les corrections particulières apportées aux données seront présentées avant l'examen des résultats, tant en termes de précision que de complexité.

4.1.1. Présentation de la plateforme matérielle

Nous avons utilisé les nœuds *Nanotron* présentés en section 2.1.1 associés à des cartes de développement dont la création est décrite dans [26] et rappelée à la section 2.1.1. Les expérimentations ont pris place dans un milieu intérieur et ont été de deux types : en une dimension (1D) puis en 2 dimensions (2D). Bien que la suite de ce document présente une solution applicable au cas 2D, une adaptation au cas 3D est possible grâce à l'ajout de contraintes sur la hauteur par rapport au sol des nœuds à localiser.

Tout d'abord, nous avons effectué une simple mesure de distance ou *ranging*, qui n'implique que deux nœuds A et B. Soit A le nœud fixe, relié à un ordinateur par une connexion série. B se déplace le long d'une droite le reliant à A et, pour différentes valeurs de distance, des échantillons de la mesure effectuée par le matériel sont enregistrés.

Dans un deuxième temps, nous avons réalisé des tests de localisation en 2 dimensions. Quatre nœuds participent à ce scénario où trois jouent le rôle de balises fixes tandis que le

quatrième occupe diverses positions dans la pièce (figure 71). A chaque emplacement, un certain nombre d'échantillons sont collectés.



Figure 71 : Environnement de collecte de données

Dans un premier temps, nous examinerons le comportement de notre plateforme en mode *ranging* et en déduirons les aménagements à effectuer avant de l'exploiter en mode localisation.

4.1.2. Caractéristiques du *ranging* et corrections appliquées

Dans un cadre théorique, les mesures collectées, bien qu'affectées d'une certaine erreur, devraient pouvoir être fournies telles quelles à l'outil de calcul de position. Dans la réalité, une étape supplémentaire doit être franchie correspondant à la correction ou mise en forme des données.

Cette phase tient compte des caractéristiques du signal utilisé. Dans notre cas, nous exploiterons la courbe brute des mesures obtenues pendant la caractérisation de notre plateforme (figure 73).

Afin que chaque mobile puisse accomplir la tâche de rectification de manière autonome, nous avons privilégié une méthode simple, ayant un coût minimal en termes de puissance de calcul et de mémoire. En effet, cette correction devra être appliquée par le nœud à chacun des échantillons obtenus sans nécessiter d'échange avec un serveur central. Nous avons donc opté pour une correction faisant usage de la pente a de la régression linéaire associée. Bien que ce choix exige une étape de collecte de données préalable à la mise en service du système, notre solution reste distincte du *fingerprinting* car nous ne procédons pas à un *mapping* entre la cartographie de l'environnement visé et les valeurs obtenues.

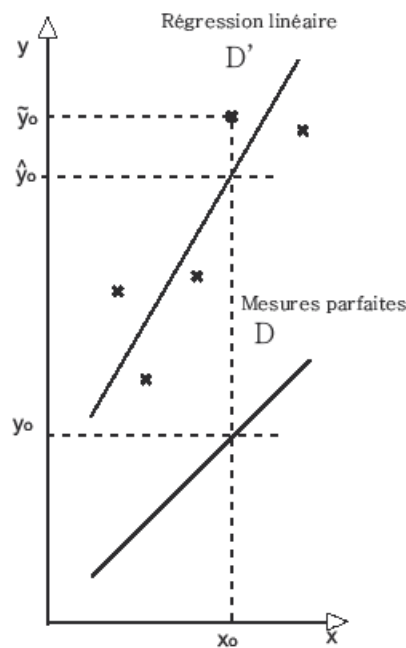


Figure 72 : Mesures réelles et régression linéaire

L'objectif de notre traitement est de retrouver la distance à laquelle une certaine valeur \tilde{y}_0 a été obtenue. Dans un premier temps, la régression linéaire correspondant à nos résultats a été calculée (droite des \hat{y}). Elle est utilisée pour estimer les distances à l'origine de ces mesures : soient les deux droites D et D' (figure 72). D correspond à une mesure parfaite tandis que D' représente la régression linéaire associée au nuage de points issus des mesures réelles (équation 81).

$$\begin{aligned} f_D(x) &= x \\ f_{D'}(x) &= \hat{y} = ax + b \end{aligned} \tag{81}$$

La variable \tilde{y} représente la mesure correspondant effectivement à x . En remplaçant \hat{y} par \tilde{y} dans la réciproque de $f_{D'}$, la valeur de x peut être estimée. Les effets de ce traitement sont décrits sur la figure 73 : puisque nous utilisons SDS-TWR entre deux nœuds A et B, deux

estimations de la distance peuvent être recueillies. Ces estimations forment les courbes a2b et b2a. En traitant les valeurs brutes par la réciproque de la régression linéaire, nous avons obtenu les courbes « a2b reg » et « b2a reg », déjà nettement plus proches de la réalité. Finalement, nous avons retranché de ces nouvelles estimations la valeur moyenne de l'erreur calculée durant cette caractérisation et avons abouti à la courbe « corrigés moy ». La mesure brute ayant été effectuée en dehors de la chambre anéchoïque, l'erreur ainsi corrigée résulte des effets combinés de l'environnement (multitrajet) et de la qualité même de l'outil de mesure.

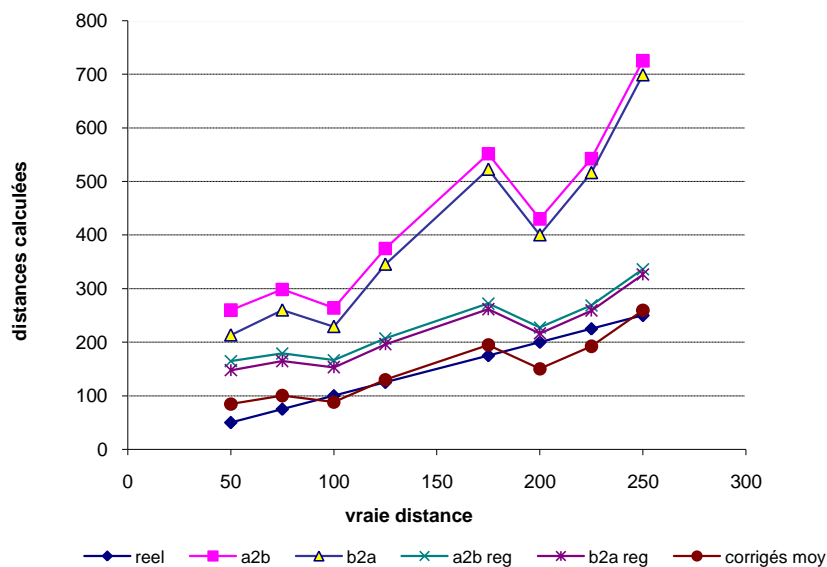


Figure 73 : Première correction des valeurs brutes de distance

Cette démarche satisfaisait aux contraintes de simplicité de notre correction. Pourtant, nous avons dû modifier notre outil par la suite. Une fois le verrou de la portée levé (§. 2.1.3), nous avons pu effectuer des mesures de distances dépassant 2,5m et avons réalisé que la réciproque de la régression linéaire ne conviendrait pas pour compenser les surestimations commises dans l'intervalle allant de 4 à 8m. Nous avons calculé l'incertitude sur l'ordonnée à l'origine b . Son amplitude était telle qu'il nous aurait fallu la recalculer régulièrement. L'épreuve de son utilisation par le prototype a confirmé cette tendance : entre deux sessions de mesure, une mise à jour du paramètre b était nécessaire. Nous avons donc choisi une nouvelle expression, donnée par l'équation 82. Elle fait intervenir la pente a de la régression linéaire mais également, l'incertitude σ_a sur cette même pente et permet d'estimer la distance à laquelle la valeur brute x' a pu être collectée.

$$f(x') = \frac{x'}{a * (1 + \sigma_a)} \tag{82}$$

L'impact de cette nouvelle fonction de correction est illustré par la figure 74. En rouge est dessinée la courbe issue des mesures brutes. La droite noire correspond à la première bissectrice. La courbe bleue représente la correction par réciproque de la régression linéaire tandis que la verte montre les résultats de l'application de la transformation donnée par l'équation 82.

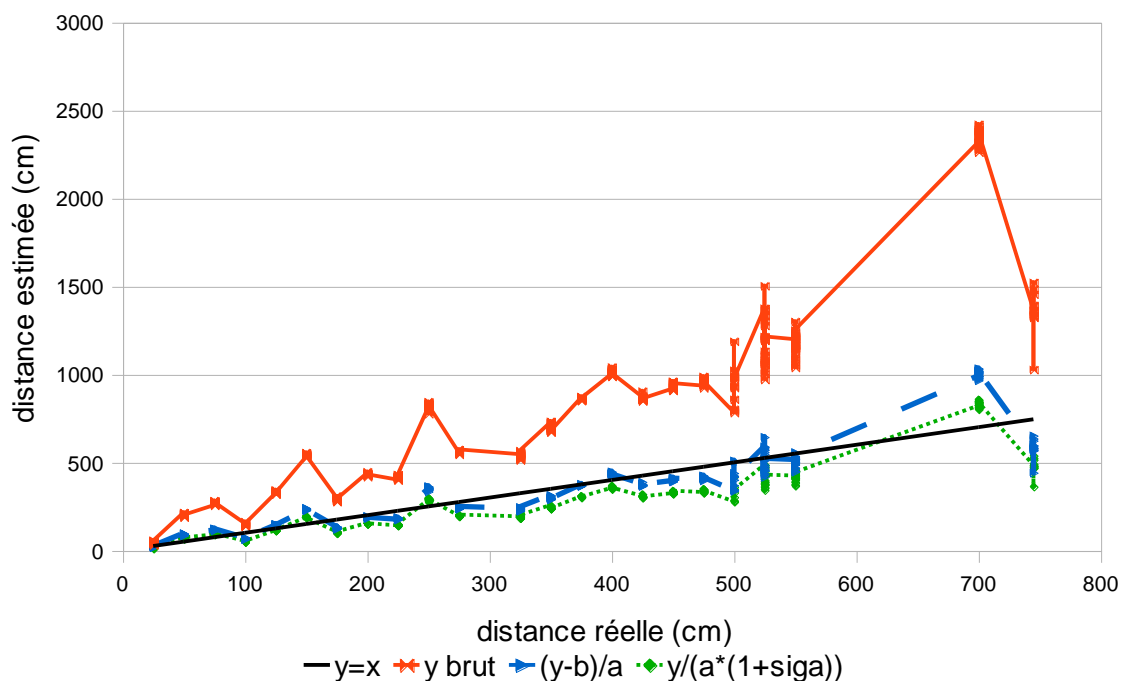


Figure 74 : Deuxième correction des estimations de distance

Le coût pour le mobile étant le même et cette équation n'exigeant pas une recalibration fréquente de ses paramètres, nous avons décidé de la conserver comme outil de correction des données brutes.

4.1.3. Résultats de localisation

Notre plateforme matérielle étant constituée de quatre nœuds, trois d'entre eux ont joué le rôle d'ancres durant nos expérimentations. Ces nœuds de référence ont été placés de manière à former un triangle dans une pièce de 8,5m x 5,5m (figure 71). En différents points de la pièce, le mobile a calculé sa position à 5 reprises. L'ensemble de ces résultats a été collecté sur un ordinateur portable à travers une liaison série. La figure 75 illustre la topologie de test adoptée.

Dans un premier temps, en raison de difficultés techniques, la localisation a eu lieu à l'intérieur du triangle formé par les balises. En effet, la portée des équipements étant très faible dans leur configuration initiale (§ 2.1.3), il était pratiquement impossible de maintenir les communications en dehors du triangle. Une fois ce problème résolu, le mobile a pu adopter des positions plus éloignées tout en restant dans la même pièce.

En ce qui concerne les perturbations, la configuration de la salle et la présence de plusieurs armoires métalliques ont été des facteurs propices au multitrajets. Finalement, le passage de personnes, bien qu'il n'ait pas été très intense, a également eu lieu durant les mesures.

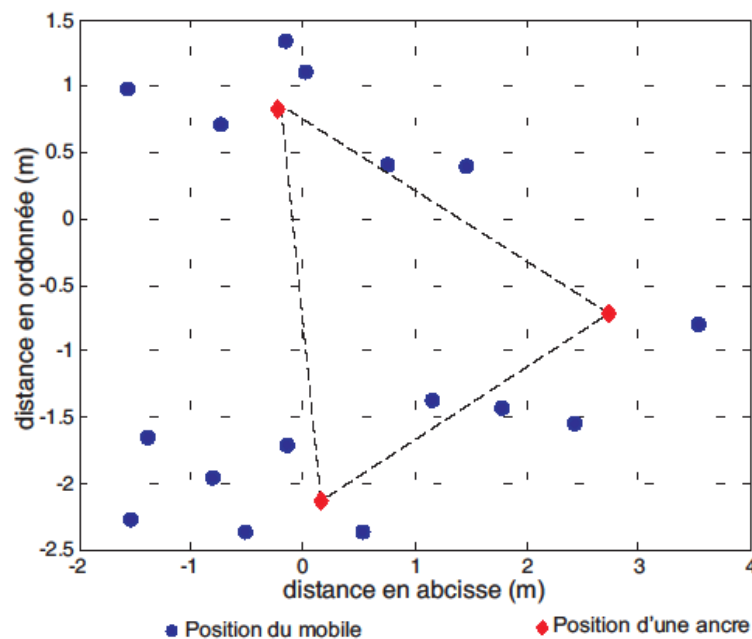


Figure 75 : Topologie de test

4.1.3.1. Précision et observations diverses

Nous avons testé le protocole SSDS-TWR sur le prototype. L'ordre dans lequel les ancres sont contactées a été prédéfini comme l'ordre croissant des adresses MAC de celles-ci. Nous avons pu observer les phénomènes suivants.

- **Variation sur le nombre d'intersections** Selon les tailles relatives des rayons des cercles utilisés dans la création de la matrice, la solution peut correspondre à une ou deux surfaces (§ 3.2.1.1.2),
- **Variation sur le nombre d'anneaux** Dans le meilleur des cas, le mobile obtiendra autant d'anneaux que de partenaires de localisation. Malheureusement,

les pertes de trames peuvent provoquer l'échec de la mesure de distance (§ 1.2.1.2) et réduire le nombre d'anneaux,

- **Besoin d'un ordonnancement des anneaux** (§ 3.2.1.4) : Comme nous l'avons vu précédemment, classer les anneaux selon leur taille au moment de la création de la matrice de points permet généralement d'améliorer la précision,
- **Absence de solution** Cette situation peut être rencontrée pour des raisons très diverses, par exemple :

- l'intersection des anneaux n'existe pas : quel que soit le couple d'ancres (A_i, A_j) , on a :

$$R_{\max,i} + R_{\max,j} < \text{distance}(A_i, A_j) \quad (83)$$

- les anneaux sont tangents : l'intersection ne peut donc servir de base à la définition d'une matrice.,
- Les perturbations ont provoqué la génération de valeurs de distance irrationnelles : généralement, ces distances sont négatives et en valeur absolue supérieures au kilomètre. Elles apparaissent lorsque le médium est perturbé comme le présente la figure 76. La trame de M ayant atteint A, cette ancre stocke l'instant de réception du message, émet un acquittement dont il conserve l'instant d'émission et transfère la trame à la couche supérieure. Si l'acquittement est perdu, une réémission de la trame 1 est nécessaire. En la recevant, la couche physique de A détecte un doublon (trame 3) : un ACK sera émis à nouveau (trame 4) mais la couche supérieure ne sera pas sollicitée, conformément à la norme IEEE 802.15.4. Les instants ne seront donc pas mis à jour, ce qui conduira à combiner des valeurs incohérentes.

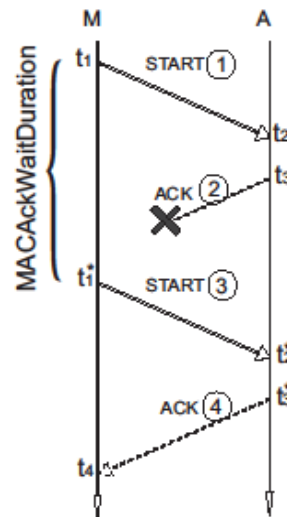


Figure 76 : Impact des collisions sur TWR

Nous avons obtenu une erreur moyenne de localisation de 116cm sur le prototype.

Dans un second temps, nous nous sommes livrés à une campagne de mesures similaires mais ici, 80 mesures en moyenne ont été récoltées pour chacune des 27 positions du mobile définies dans la pièce (figure 77). L'objectif, dans cette situation, était de recenser une grande variété d'échantillons sur lesquels tester l'ordonnement des anneaux. Nous avons pu observer qu'avec notre solution de localisation, l'erreur était inférieure à 150 cm dans 65% des cas, ceci sans correction liée aux anneaux. La moyenne de l'erreur calculée sur ces résultats est de 151cm.

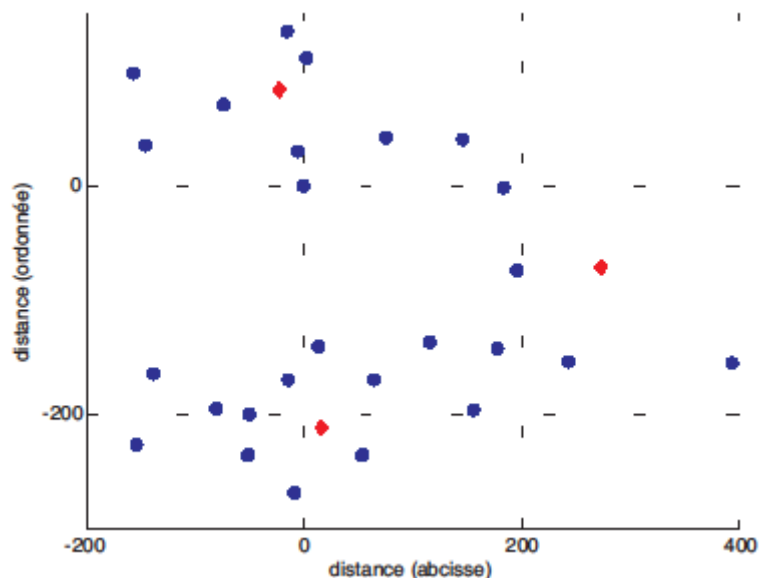


Figure 77 : Répartition des 27 positions de test

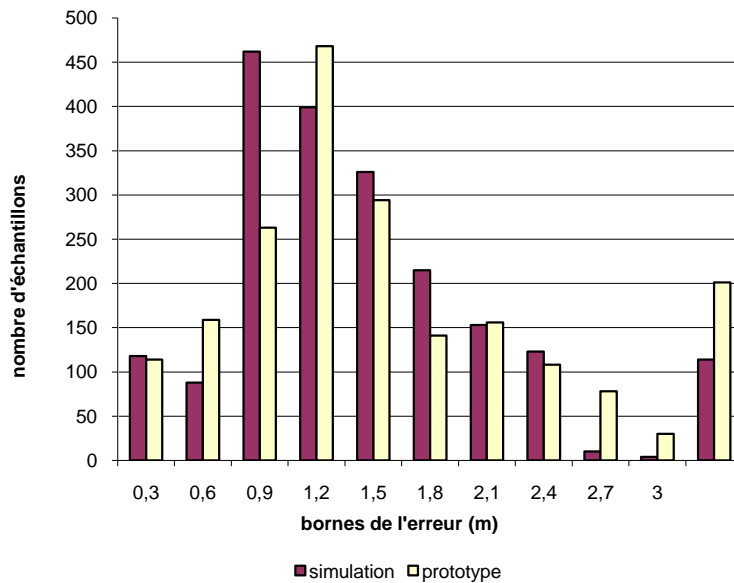


Figure 78 : Mise en évidence de l'effet de l'ordonnement des anneaux

Ces mêmes mesures ont été fournies à un programme chargé d'appliquer l'ordonnement des anneaux puis de calculer la position avec *iRingLA*. Il joue le rôle d'un simulateur d'algorithme en ceci qu'il est la traduction en java du code C dont le binaire est exécuté par les modules matériels. La figure 78 compare les erreurs brutes aux erreurs obtenues après tri des anneaux. Cette transformation a permis d'obtenir une erreur moyenne de 131cm.

Les données représentées en ivoire sur la figure 78 correspondent aux erreurs issues du prototype. On observe que la majorité des échantillons se retrouvent dans l'intervalle 0-120 cm. Une fois l'ordonnement appliqué, la majorité se retrouve décalée vers les valeurs inférieures, ce que représente le graphique violet. Cette translation est limitée car dans les échantillons que nous avons eu à traiter, le nombre de points pour lesquels l'ordre devait être modifié était restreint. Néanmoins, dans ces cas-ci, l'erreur a typiquement été réduite de deux tiers.

Il serait intéressant de pouvoir caractériser cette erreur observée sur la position, en attribuant une part au multitrajet et une seconde aux limites de la technologie utilisée. Une des pistes serait d'exécuter ce scénario dans un environnement exempt de multitrajet mais dans l'état actuel des choses, cette étude n'est pas réalisable en raison des dimensions réduites de la chambre anéchoïque.

4.1.3.2. Durée d'exécution du protocole

Dans la section 3.1.5.3, nous avons utilisé des informations temporelles afin d'étudier l'impact de notre protocole sur la consommation énergétique des nœuds. Nous présenterons dans les paragraphes qui suivent la manière dont cette durée a été mesurée.

Cette expérience fait intervenir, conjointement aux 4 nœuds du réseau, un oscilloscope dont la mission est de visualiser graphiquement les signaux pilotés par le logiciel. La mesure s'effectue principalement au niveau du mobile et pour chaque étape du processus, le fragment de code concerné est encadré par une instruction demandant la mise à 1 ou à 0 d'une broche de sortie du microcontrôleur. Dans un premier temps, nous avons considéré la durée séparant le démarrage du protocole de la réception de l'ultime trame mais cette mesure ne permettait pas d'observer les différentes phases comme les créations de message, les accès au médium, etc. Nous avons modifié notre programme de manière à encadrer chacune des exécutions de SDS-TWR. La courbe obtenue indique qu'une exécution de SDS-TWR dure 5.4ms, durée allant de la création de la trame *Start* à la réception de la trame *Reply2*. Un temps non négligeable s'écoule entre deux opérations de *ranging*. Il est consacré en priorité à la mise à jour des structures contenant les résultats de mesure de distance. En combinant toutes ces informations, nous obtenons le tableau 11 (page 104). Globalement, la mesure de distance avec 3 ancrs dure 30ms.

4.1.3.3. Durée d'exécution de l'algorithme

Dans cette section, nous examinons d'un point de vue théorique la complexité de notre algorithme. Le tableau 16 rappelle les différentes phases de notre algorithme. Soient les variables :

i : initialisation d'un élément de la matrice

c : changement de base d'un élément de la matrice

t1 : test à l'aide des deux ancrs de base sur un élément de la matrice

tk : test à l'aide d'une ancre supplémentaire pour un élément de la matrice

d : déplacement d'un élément dans la matrice

r : récupération d'un point dans la matrice

M : nombre total de points dans la matrice

Prototype

N : nombre d'éléments ayant réussi $t1$

P : nombre d'éléments ayant réussi le k -ème test.

Tableau 16 : Nombre d'opérations pour chacune des phases de l'exécution d'*iRingLA*

Phase	Exécution par défaut	Version améliorée
Initialisation de la matrice	$i*M$	$i*M$
Changement de base	$c*M$	$c*M$
Test1	$t1*M$	$t1*M+d*N$
Test $_k$	$t2*M$	$N*t2*k$
Récupération des points	$r*M$	$P*r$

L'initialisation et le changement de base consistent en des affectations du résultat de deux formules aux coordonnées x et y de chaque point.

Les tests permettent de déterminer l'appartenance d'un point à un anneau. Ils sont formés de 2 comparaisons par anneau, l'un indiquant si le point est à l'extérieur de l'anneau intérieur et l'autre si le point se trouve à l'intérieur de l'anneau extérieur. Selon le résultat de ces comparaisons, un attribut du point est modifié afin d'indiquer s'il a réussi le test.

Dans la version améliorée de l'algorithme, lors du test sur les deux premières ancres, tous les points ayant réussi l'épreuve sont déplacés vers le haut de la matrice.

```
lastline=0;
lastcol=0;
mat[lastline][lastcol].x=mat[line][col].x;
mat[lastline][lastcol].y=mat[line][col].y;
mat[lastline][lastcol].isInSolution=1;
if(lastcol==nbColMax-1){
    lastcol=0;
    lastline++;
}else{
    lastcol++;
}
```

Figure 79 : Algorithme de déplacement d'un point dans la matrice

Le deuxième test ne s'appliquera alors qu'aux N points retenus à la fin de $t1$. Dans le pire des cas, l'exploitation des anneaux subséquents ne réduira pas N . Les 2 comparaisons liées au second test seront donc exécutées $N*k$ fois, k étant le nombre d'ancres restantes.

La récupération consiste à identifier dans la matrice les points faisant partie de la solution. Dans le cas de la solution par défaut, l'intégralité de la matrice est parcourue et une comparaison de l'attribut indiquant le statut du point avec une valeur de référence est réalisée. Si la valeur obtenue suggère l'appartenance de l'élément à la solution, ses coordonnées sont intégrées à l'estimation finale.

La solution optimisée utilise une variable supplémentaire contenant la valeur P pour réduire le nombre d'opérations. En effet, seuls les P premiers éléments seront combinés pour obtenir l'estimation de la position. Notons qu'ici, aucun test sur attribut n'est nécessaire.

Afin d'écrire les expressions des deux temps d'exécution, nous définirons une variable supplémentaire α , correspondant à la réduction du nombre de points après chaque test. Les équations 84 et 85 indiquent les durées respectives des algorithmes par défaut et amélioré. Les variables $i, c, t1, d, tk$ et r sont remplacées par leur équivalent en nombre d'opérations.

$$\begin{aligned} T_{\text{defaut}} &= M(i + c + t1 + k * tk + r) = M(7 + 13 + 21 + 14 * k + 17) \\ &= M(14 * k + 58) \end{aligned} \tag{84}$$

$$\begin{aligned} T_{\text{amélioré}} &= M(i + c + t1 + \frac{d}{\alpha} + \frac{tk * k}{\alpha} + \frac{r}{\alpha}) = M(7 + 13 + 21 + \frac{8}{\alpha} + \frac{14k}{\alpha} + \frac{17}{\alpha}) \\ &= M(41 + \frac{25 + 14 * k}{\alpha}) \end{aligned} \tag{85}$$

Les deux implémentations restant des recherches linéaires, l'expression de leurs temps d'exécution est de la forme $O(n)$ avec $M=n$. Notre modification a essentiellement modifié la pente de la courbe (figure 80).

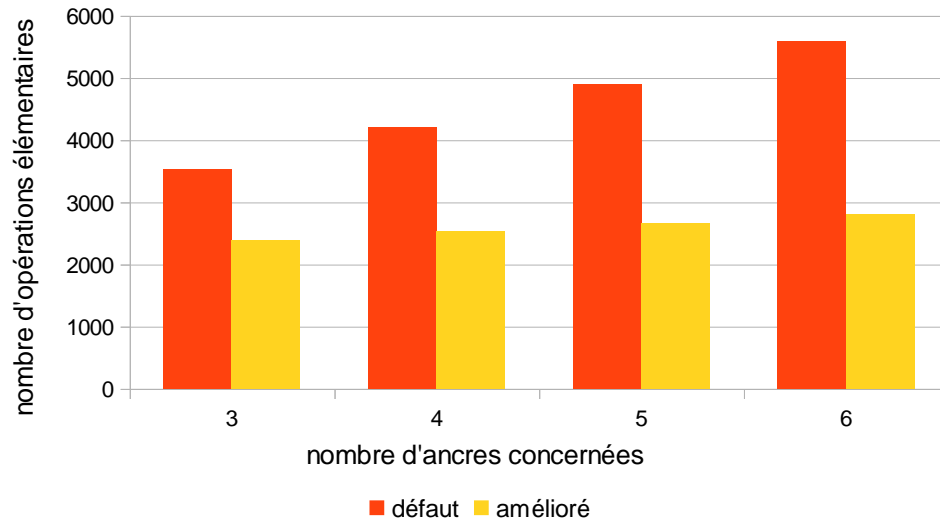


Figure 80 : Impact de la modification d'*iRingLA* sur le nombre d'opérations

Nous avons choisi pour cette figure une matrice de 49 points telle qu'utilisée par notre prototype. La valeur du paramètre α (facteur de réduction du nombre de points) a été fixée à 5 afin de correspondre aux résultats réels. Nous pouvons observer que l'amélioration de 30% de la durée d'exécution mesurée sur le prototype lors de la localisation avec 3 ancres correspond à la différence de 32% indiquée par le graphique.

4.1.4. Conclusion

La solution, en termes de protocole de mesure de distance et d'algorithme de calcul de la position, a été mise en œuvre sur une plateforme réelle. Afin de permettre l'exploitation d'un algorithme aussi simple, nous avons mis en forme les données avant de les fournir au traitement logiciel. Dans notre simulateur (§ 4.2), le protocole et l'algorithme seront implémentés mais il est évident que la phase de correction sera omise : l'objectif de notre simulateur n'est pas de modéliser les erreurs introduites dans la mesure pour ensuite proposer différents algorithmes de rectification de la mesure.

4.2. Simulateur *DokoSim*

Si l'implémentation d'une proposition sur un démonstrateur a l'avantage de prouver de manière indiscutable la faisabilité de la solution, elle ne saurait rendre la simulation obsolète. En effet, les simulateurs ont un rôle à jouer à différentes phases du travail de recherche. Par exemple, au tout début de la phase de conception, lorsque la solution n'en est qu'au stade d'idée, l'environnement virtuel fournit des indicateurs quant à la faisabilité de la solution. Bien qu'il soit souvent nécessaire de procéder à des adaptations au moment d'implémenter le

protocole sur un prototype réel, l'outil de simulation offre la possibilité de procéder à un dimensionnement préalable.

Dans certains cas, le simulateur peut être conçu pour faciliter le transfert de la solution vers des plateformes matérielles. Finalement, la problématique du passage à l'échelle, en raison de la nécessité de mettre en place un grand nombre d'équipements, peut être étudiée avec un coût raisonnable à l'aide d'un simulateur.

4.2.1. Motivations

La section 3.1.3 a présenté le principe du protocole PSDS-TWR. Le lecteur aura noté que ce protocole renonce à utiliser les acquittements au profit du parallélisme. La suppression de ces trames rend malheureusement le test de PSDS-TWR impossible sur notre prototype. En effet, la désactivation des ACK implique le recours à une horloge de niveau haut pour mesurer les instants et les durées. Sur notre plateforme matérielle, l'horloge disponible présente une résolution de $30,5\mu\text{s}$. Par conséquent, en attendant d'avoir accès à des équipements plus performants, nous avons dû nous tourner vers un simulateur pour étudier la performance de PSDS-TWR.

Le protocole PSDS-TWR se base sur le fait que seul le mobile initie les communications dans le *star interval* pour se dispenser de CSMA/CA. Néanmoins, une contrainte supplémentaire existe sur l'agencement temporel des réponses des balises. La figure 81 illustre l'importance de ce délai.

Considérons les quatre nœuds A, B, C et M répartis dans une pièce de 15m x 15m. Imaginons les ancres A, B et C placées en trois des quatre coins de la pièce et M occupant un emplacement quelconque dans la salle. La différence moyenne des distances séparant M des ancres est de 4m, ce qui correspond à une différence de temps de vol de 13ns. M initie PSDS-TWR en envoyant une trame de START contenant l'ordre de réponse [A, B, C]. Cette trame est représentée par la flèche de M vers les ancres (figure 81 (a)). Soit T_t le temps de traitement de la trame START et de préparation de la réponse. A $t_{\text{impact,A}}+T_t$, A est prêt à répondre, de même que B à $t_{\text{impact,B}}+T_t$ et C à $t_{\text{impact,C}}+T_t$. La différence entre ces instants ramenés à une horloge absolue est de 13ns donc si les ancres répondent toutes immédiatement, les transmissions vont se perturber (figure 81 (a)). Une solution serait de prévoir un délai correspondant à D tel que :

$$D_i = (\text{temps de transmission d'une trameReply}_i) \times (\text{emplacement du noeud } i \text{ dans la liste}), \quad (86)$$

$$i = 0, 1 \dots n$$

Ceci pourtant ne résout pas le problème. Bien que la collision sur le médium soit évitée, elle risque de se produire dans la mémoire du *transceiver* (figure 81 (b)). En effet, si k *buffers* sont disponibles et que $k+1$ messages distincts sont envoyés, un message sera nécessairement perdu : soit le *transceiver* réécrit dans un *buffer* occupé (*overwrite*), soit la réception est désactivée et le message arrivant ignoré. Par conséquent, le calcul du temps d'attente entre les réponses doit tenir compte du temps de propagation d'une trame du même type mais aussi du temps de traitement d'une de ces trames donc du délai de libération d'un *buffer* (figure 81 (c)). Notons que dans le cas d'un nœud de WSN, ces deux temps peuvent être confondus car les limites matérielles de ces plateformes impliquent le traitement séquentiel des trames entrantes qui sont toujours stockées dans une zone mémoire prédéfinie par le microcontrôleur : tant que le microcontrôleur n'a pas terminé le traitement de la trame en cours, il ne peut pas en récupérer une nouvelle de la couche physique.

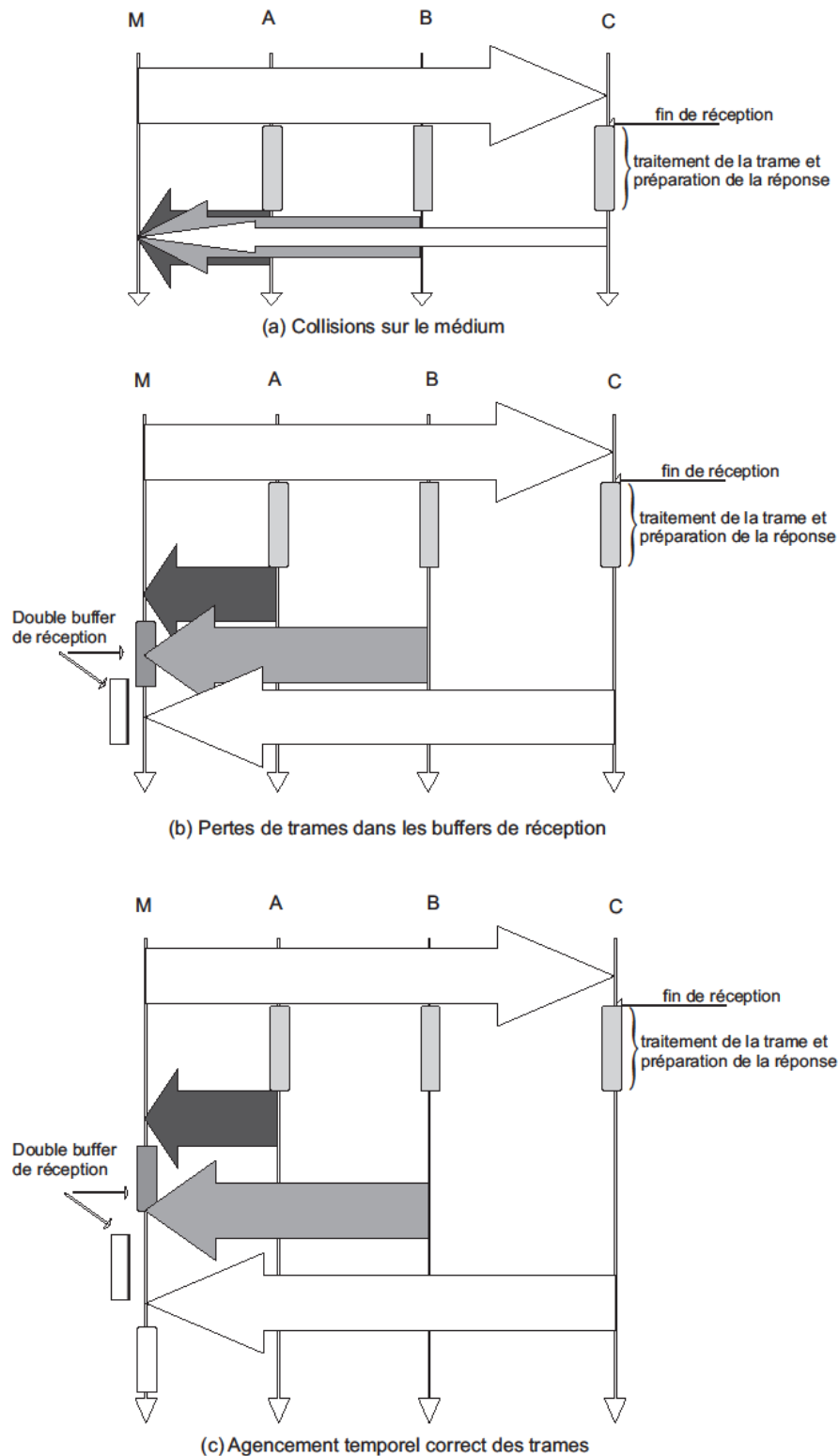


Figure 81 : Impact du délai entre les réponses

Pour fournir une appréciation utile des performances, l'outil choisi devra tenir compte des temps de traitements au niveau des nœuds du réseau de capteurs. Cette contrainte n'est pas triviale sachant que bien des simulateurs ont été créés à la base pour des réseaux filaires

d'ordinateurs et se sont progressivement adaptés par l'ajout d'implémentations de protocoles variés.

4.2.2. Choix d'un simulateur

Il existe plusieurs simulateurs de réseau connus de la communauté scientifique. Laissant de côté les outils payants, nous avons le choix entre la combinaison WSNet+WSim [18], OMNeT++ [78], ns-2[76] et ns-3 [77]. Nous présenterons dans un premier temps ns-2 et OMNeT++ ainsi que les raisons pour lesquelles nous ne les avons pas retenus avant d'introduire la plateforme WSNet+WSim.

4.2.2.1. NS-2 et NS-3

Ce simulateur orienté réseau est très riche en termes de protocoles implémentés. En dépit des doutes émis en ce qui concerne son adéquation à l'étude des WSN [89], bien des travaux se basent sur cet outil pour évaluer leur performance. Dans le travail de [3], la mesure de distance est réalisée à partir du RSSI sous NS-2 et l'effet du bruit sur la mesure représenté par une variable aléatoire normale dont la moyenne et l'écart-type dépendent de la distance réelle. Dans [80], la mesure de distance à partir de la différence des temps d'arrivée est implémentée en ajoutant à la mesure de temps une erreur suivant une distribution gaussienne de moyenne nulle et d'écart-type $3ns$. Les autres sources d'erreurs sont négligées dans cette étude de la localisation par l'utilisation d'un réseau IEEE 802.11.

La nouvelle version de NS-2, désignée comme NS-3, est aussi riche que sa précédente en termes de protocoles et structures implémentées mais ne fournit pas de support pour les réseaux de capteurs sans fil.

4.2.2.2. OMNeT++

MIXIM est né de OMNET++ et a pour objectif d'offrir les outils pour simuler tous types de réseaux, filaires ou non. Notamment, il dispose d'une implémentation des deux versions du standard IEEE 802.15.4a UWB et 802.15.4 DSSS. Les efforts d'adaptation au cas réel sont illustrés dans [82]. Les modèles de couches physiques fournis sont riches en détails.

Les simulateurs NS-2 et OMNET++ ne tiennent pas compte du temps de traitement des trames. Pour NS-2, cette particularité s'est faite jour au cours de l'étude décrite en [27] tandis que pour OMNET++, ce choix d'implémentation est clairement indiqué, notamment dans [82].

4.2.2.3. WSNNet+WSim

La plateforme de simulation WorldSens [18] se distingue des autres simulateurs en abordant la question du *design* à plusieurs niveaux. Ceci est possible par l'exploitation de deux simulateurs : WSNNet, qui est un simulateur de réseau, et WSim qui émule le comportement du matériel sur lequel la solution sera implémentée. Le travail débute sur WSNNet où le dimensionnement des protocoles est réalisé. Ensuite, le code conçu pour la plateforme choisie est transformé en un fichier binaire qui est injecté dans WSim. Ce dernier est configuré pour émuler très précisément la plateforme cible. Une description complète du nœud, de ses interfaces et de ses périphériques étant implémentée, le code permet de vérifier les propriétés de la solution en temps réel. Cette description se fait par combinaison de blocs élémentaires disponibles sous formes de bibliothèques.

Ce travail approfondi est pourtant une épée à double tranchant : si la plateforme cible ne peut être reconstituée à l'aide des briques disponibles sous WorldSens, la question de son intérêt peut se poser. Dans l'état actuel de nos connaissances, les microcontrôleurs ATmega644V ainsi que la couche physique 802.15.4a CSS ne sont pas prises en charge par WorldSens. Simuler notre solution conduirait au choix d'une plateforme différente "équivalente", ce qui va à l'encontre de l'objectif d'un simulateur aussi détaillé.

Nous avons donc décidé de développer notre propre simulateur. Certes, dans un premier temps, il n'aura pas la richesse de ses prédécesseurs mais au fur et à mesure, comme cela s'est d'ailleurs vu pour NS-2, d'autres modules pourront s'ajouter à sa bibliothèque.

4.2.3. Description technique

Si bien des simulateurs sont écrits en C, nous avons opté pour Java pour le développement de *DokoSim*. Premièrement, si nous sommes très familiers de la programmation en C, la réalisation de projets en Java est nettement plus confortable. Elle correspond à notre approche naturelle des problèmes qui consiste à identifier les différents acteurs, à définir leurs rôles, les actions possibles et enfin leurs propriétés ou attributs.

D'autre part, nous nous intéressons à de la simulation et non de l'émulation : l'utilisation d'un langage de haut niveau n'est donc pas gênante dans la mesure où le temps dans la simulation est mesuré indépendamment de celui de l'ordinateur hôte.

Dans les paragraphes suivants, nous présenterons les différents éléments en interaction dans notre logiciel ainsi que les grandes lignes de leur implémentation.

4.2.3.1. Structure du simulateur

Le simulateur permet de visualiser les interactions qui ont lieu au fil du temps entre des nœuds d'un réseau sans fil. Pour ceci, on dispose d'une entité modélisant le médium qui prend en charge l'acheminement du signal radio en un temps dépendant principalement de la distance séparant les nœuds. Une émission consiste à confier au médium une trame à destination d'un certain sous-ensemble du réseau. Le message est délivré à tous les membres du sous-ensemble en fonction des propriétés du lien radio existant entre la source et chacune des destinations. Ces propriétés sont la distance séparant les nœuds et la probabilité de pertes de trame. Dans notre implémentation, une probabilité d'erreurs globale est définie qui affecte tous les liens. Suivant cette probabilité, une fraction des messages émis ne parvient pas à destination en raison des interférences ou des erreurs de transmission liées au Taux d'Erreur Bit (TEB). Ce modèle convient pour un simulateur se positionnant à l'extérieur de la couche physique : la distinction entre les différents phénomènes pouvant conduire à une perte de message n'est plus nécessaire car au-dessus de la couche MAC, ils se résument à une trame perdue. Finalement, si la distance entre les partenaires est supérieure à la portée maximale configurée, le message n'est pas délivré.

Une boucle finie permet de simuler l'écoulement du temps. Les durées associées aux différents traitements ont été configurées soit à partir des valeurs collectées sur le prototype, soit en utilisant des formules théoriques (§ 3.1.5.1).

4.2.3.2. Architecture de classes

Cette section décrit les différentes classes utilisées durant l'implémentation. Nous commencerons par les plus génériques avant d'aborder celles qui réalisent une fonction très spécifique.

L'architecture de classes est présentée en figure 82. Les différentes relations peuvent être traduites sous la forme suivante :

- Les nœuds sans fil (instances de `WirelessNode`) utilisent un unique médium pour leurs transmissions. Ce médium existe indépendamment de la présence de nœuds sans fil. Suivant les positions des nœuds et ayant configuré une portée radio fixe, des zones de diffusion distinctes peuvent être définies au sein de ce médium.

- Un nœud sans fil quelconque exploite deux horloges et deux objets de type Frame durant son fonctionnement : leurs rôles respectifs seront décrits dans la section consacrée au modèle du nœud sans fil (§ 4.2.3.2.4).
- Un nœud sans fil peut implémenter un algorithme de localisation. Il possède alors un objet LocatorEngine qui implémente cet algorithme.
- Pour organiser ses activités, le nœud sans fil utilise un unique Agenda contenant les différents événements dont l'occurrence a été prévue. Ces événements sont appelés des AgendaItems et leur nombre est variable.

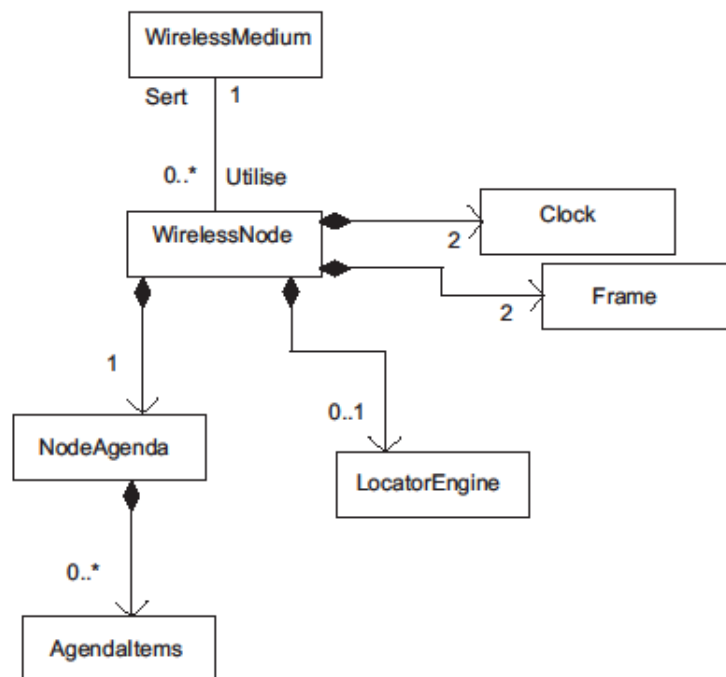


Figure 82 : Architecture de classes

4.2.3.2.1. Classe *Clock*

Tous les équipements du réseau sont dotés d'une horloge. A la création, une fréquence est définie et un *offset*³⁸ peut également être appliqué comme valeur de démarrage. Ceci permet de différencier les instants de démarrage des différentes horloges. Cet *offset* s'exprime en nanosecondes.

Deux méthodes sont disponibles pour ce type d'objet :

- `getCurrentTime()`

Cette méthode renvoie la valeur courante de l'horloge de l'élément.

³⁸ Décalage appliqué à la valeur de l'horloge

- `stepThroughTime()`

Cette méthode incrémente la valeur de *Clock*. Elle est appelée au sein de la boucle de base. Chaque exécution de la boucle de base correspond à la plus petite unité de temps mesurable par le simulateur. Il s'agit donc de sa granularité *g*. Pour chaque horloge, un rapport existe entre *g* et la période de l'horloge : pour parcourir une période de *Clock*, un certain nombre *k* d'exécution de la boucle est nécessaire. Aussi, toutes les *k* exécutions, la méthode `stepThroughTime()` sera appelée. Par exemple, pour une horloge à 1MHz, si la granularité est d'une nanoseconde, un millier d'exécutions de la boucle sont nécessaires pour parcourir la plus petite unité de temps mesurable par cette horloge.

4.2.3.2.2. Classes *NodeAgenda* et *AgendaItems*

Les nœuds d'un réseau sans fil exécutent des calculs internes et pilotent leur interface radio pour la communication. Dans le cadre d'un réseau de capteurs, nous avons à manipuler des modules ne pouvant généralement exécuter qu'une action à la fois. Pour chaque action, un ensemble d'étapes prédéfinies a été codé. Par exemple, la version la plus simple de l'émission d'une trame peut se subdiviser en la création de la trame, le transfert de la trame par le bus SPI (§ 1.1.1) et la transmission des bits sur le médium. Chacune de ces étapes a une certaine durée et est identifiée par un code. Selon ce code, il est possible de décider de la prochaine action à exécuter. Cette décision survient à la fin de l'étape en cours : cet instant est prédit grâce à la valeur courante de *Clock* et à la durée de l'opération.

Pour représenter la succession des tâches, nous avons choisi de créer des objets *NodeAgenda*. A l'instar des agendas traditionnels, un objet de cette classe liste les différentes activités que le nœud doit exécuter, ainsi que l'instant prévu de démarrage. Il inclut également des méthodes permettant de gérer les activités représentées par des *AgendaItems*. La fin de l'exécution de la tâche correspondant à l'objet *AgendaItems* *i* coïncide avec le début de la suivante et est désignée par le terme « rendez-vous ».

- `addItem()`

Cette méthode permet d'ajouter un objet *AgendaItems* à *NodeAgenda*

- `getNextItemTime()`

Cette méthode permet de connaître l'heure du plus prochain rendez-vous. Du point de vue du nœud, la tâche précédente est en cours jusqu'à l'heure de démarrage de la

suivante, qui constitue le prochain rendez-vous. L'absence de *multithreading* permet cette simplification.

- `popItemOut()`

Une fois qu'une tâche a été entamée, sa référence doit être supprimée de l'objet *NodeAgenda*. La méthode *popItemOut()* supprime donc le premier élément de la liste.

- `popFirstItemOut(int actionCode)`

Le fait d'avoir un agenda global pour le nœud rend parfois nécessaire de supprimer un élément qui n'est pas le premier de la liste. Par exemple, dans le cadre d'une communication acquittée, l'expiration du *timer* lié à l'attente de l'ACK est un rendez-vous figurant dans l'agenda. Si l'ACK n'est jamais reçu, au moment de ce rendez-vous, une réémission pourra avoir lieu. Si l'ACK est reçu, ce rendez-vous doit être supprimé.

- `printAgenda`

Pour des besoins de vérification par l'utilisateur, il peut être intéressant de récupérer le contenu de l'agenda du nœud à un instant *t*.

4.2.3.2.3. Classe WirelessMedium

Le médium a la responsabilité d'acheminer, en un temps dépendant de la distance, les messages de la source à la destination. Pour ceci, il dispose d'une certaine connaissance du réseau, matérialisée par la liste des nœuds connus et la matrice de connectivité, *C*. Cette dernière comporte une ligne et une colonne pour chaque équipement réseau. Dans notre implémentation, la valeur C_{ij} avec $i \neq j$, dépend uniquement de la distance : si les nœuds sont à portée, $C_{ij}=1$. Sinon, $C_{ij}=0$. Cette table *C* est mise à jour de la manière suivante : à la création d'un nœud, une position lui est assignée puis il est ajouté au vecteur *network*, qui représente la connaissance du médium pour ce qui a trait aux équipements radio. La variable *Position* du nœud est définie de manière à être accessible au médium. Ce dernier, au moment d'ajouter le nouveau nœud à *network*, calcule la distance le séparant de chacun des nœuds connus et se base sur cette distance pour configurer les cellules de *C*. Dans le cadre des communications sans fil pour les réseaux de capteurs, la taille réduite des nœuds rend difficile l'utilisation d'antennes multiples directement implémentées sur le module. Par conséquent, lorsque le circuit pilotant l'antenne disponible est configuré en émission, il est impossible de recevoir simultanément des messages. Ceci se traduit dans la matrice par une diagonale $C_{ii}=0$: un équipement sans fil du réseau de capteurs ne peut détecter les messages dont il est la source.

Les méthodes implémentées sont :

- `addNode(WirelessNode node)`

Durant la configuration du scénario, tous les nœuds créés sont ajoutés à la base de connaissance du médium de manière à pouvoir interagir avec ce dernier et donc avec leurs éventuels voisins. Un vecteur nommé *network* contiendra la référence vers les différents nœuds. Le médium est donc capable d'interroger un nœud pour obtenir sa position. Cette information est alors utilisée pour peupler et mettre à jour la matrice de connectivité.

- `freeMedium()`

Lorsqu'un nœud entame une transmission, une variable interne de l'objet *WirelessMedium* indique que le support sans fil est occupé. Le nœud radio, à la fin de la transmission, indique au médium qu'il a terminé en appelant cette méthode.

- `getFirstBitDelayNS(double distance)`

Cette fonction calcule le délai de transmission associé au premier bit d'un message émis. Ce décalage dépend de la distance et de la couche physique utilisée. Son calcul est donné par l'équation 87.

$$\text{DélaiPremierBit} = \frac{1}{\text{débit}} + \frac{\text{distance}(\text{source}, \text{destination})}{\text{vitesse}} \quad (87)$$

- `getInterNodeDistance(int addr1, int addr2)`

Afin de réguler les échanges, le médium doit connaître la distance séparant la source de la destination. Cette fonction parcourt la base de connaissance avec les adresses de la source et de la destination comme paramètres et, grâce aux données stockées, retourne la distance entre ces équipements.

- `processPHYFrame(Frame frame)`

Lorsqu'un émetteur/récepteur est prêt à transmettre une trame, il la soumet au médium par le biais de cette fonction. Cette dernière retourne une variable indiquant le succès ou l'échec de la transmission. Si cet aspect ne rejoint pas la réalité, il a l'avantage de faciliter, durant l'interprétation de la trace, la reconnaissance des collisions.

4.2.3.2.4. Classe WirelessNode

Les équipements réseaux sont créés à partir de la classe *WirelessNode*. Suivant leur configuration, ils peuvent agir comme des ancres ou des mobiles. Dans les deux cas, le protocole de localisation à exploiter peut être choisi entre SSDS-TWR et PSDS-TWR.

Chaque nœud dispose d'une horloge de haut niveau et d'une horloge dédiée à la couche physique. La première est utilisée pour agencer les différents rendez-vous tandis que la seconde intervient lors de l'estampillage temporel des messages.

- `liveTimeUnitAnchor()` / `liveTimeUnitMobile()`

Suivant sa nature, un nœud aura à exécuter l'une de ces deux méthodes à chaque incrémentation de son horloge de niveau haut. Il pourra ainsi déterminer si une action a été prévue pour l'heure courante et commencer son exécution. Dans le cadre de notre simulateur, les actions sont divisées en fragments atomiques, indivisibles et de durée connue. L'exécution de `liveTimeUnitXXX()` mène à la réalisation complète d'un fragment et la création d'un point de rendez-vous dans le temps, éloigné de la durée du même fragment.

- `makeFrame()`

Suivant le type des trames, un certain nombre d'étapes doit être réalisé. Parmi elles, le choix du destinataire, la composition de la charge utile, la configuration des propriétés de la trame... Ces actions déterminent la durée de la phase de création du message. Dans notre contexte, nous utilisons des messages de taille fixe dont la structure est prédéfinie. Nous pouvons donc borner le temps de création de la trame suivant son type. Lors de la préparation d'un envoi, le nœud complète les différents champs de l'objet *Frame* puis, à travers `makeFrame()`, il fixe le rendez-vous de fin de création tout en mettant à jour des champs tels que le numéro de séquence.

Une version surchargée de `makeFrame()` existe pour les trames de taille variable : un paramètre est alors inclus lors de l'appel de la méthode pour indiquer le délai supplémentaire.

- `notifyCollision()`

Bien que l'objectif soit de développer un protocole de localisation exempt de collisions, pendant la phase de dimensionnement, ces dernières restent inévitables : nous avons donc inclus une méthode simple permettant de signaler l'occurrence d'une collision. Bien entendu, les nœuds à l'origine de la collision ne seront pas notifiés. Seuls les modules en réception à cet instant seront avertis.

- `receiveStartIndication(long delayNS, int length, Frame frame)`

Cette méthode est appelée par le médium lors du traitement d'une trame pour indiquer à tous les destinataires potentiels que dans *delayNS* nanosecondes, une trame de longueur *length* arrivera au niveau de leur récepteur.

- `receiveCompleteDelay()`

Dans la réalité, une fois une réception achevée par la couche PHY, la couche MAC en est informée par le biais d'une interruption. Elle identifie alors le type de l'interruption et, ayant reconnu une notification de message entrant, elle demande le transfert du message des *buffers* de la couche physique à la mémoire du μ C. Plutôt que de créer des interruptions, nous avons utilisé une astuce simple : nous avons permis à la couche physique de connaître la taille du message en cours de réception. Ainsi, elle peut indiquer à la couche MAC que dans un temps déterminé, elle peut récupérer le nouveau message.

- `waitATurnAroundtime(Frame frame)`

Dans le cas des messages exigeant un acquittement, il est nécessaire pour le nœud récepteur de modifier l'état de son *transceiver* de la réception à la transmission. Cette opération n'étant pas instantanée, la méthode `waitATurnAroundTime` s'assure de la permanence des données durant la commutation. Ceci revient à créer un *AgendaItems* spécifiant qu'une trame d'acquiescement doit être émise pour la dernière trame reçue.

- `spiTransmit()` / `spiTransmitUp()`

Les paragraphes précédents ont illustré la communication entre la couche MAC et la couche physique. Elle prend place sur le bus SPI, au rythme défini pour cette interface. Les méthodes `spiTransmit()` et `spiTransmitUp()` modélisent respectivement ces transferts d'informations de la MAC vers la PHY et de la PHY vers la MAC.

- `startLocalization()`

Dans le cadre des deux protocoles mis en place sur notre prototype, SSDS-TWR et PSDS-TWR, le nœud mobile joue le rôle d'initiateur de la localisation. Cette méthode est donc appelée exclusivement par les nœuds mobiles. Selon le protocole activé, elle forme la première trame de la séquence et initialise les variables appropriées.

4.2.3.2.5. Classe LocatorEngine

Cette classe implémente le traitement des données par l'algorithme *iRingLA*. Il est l'adaptation en Java du code en C écrit pour les modules réels.

4.2.4. Choix d'implémentation

A présent, nous spécifions le positionnement de *DokoSim* par rapport au prototype au moment de son *design*. Les points sur lesquels les plateformes logicielles et matérielles se

rejoignent seront introduits et les aspects non pris en compte seront présentés tandis que ces choix seront justifiés.

4.2.4.1. Similitudes

Cette section décrit les aspects de la réalité pris en compte dans notre simulateur ainsi que la manière dont ils ont été implémentés.

- **Zones de portée à travers la matrice de connectivité** Comme nous l'avons indiqué dans les paragraphes précédents, le médium dispose d'une matrice de connectivité basée sur la distance séparant les nœuds et la portée de la technologie radio utilisée. Grâce à cet outil, des zones de portée peuvent être définies : soit C la matrice de connectivité : la zone de portée du nœud i est l'ensemble des nœuds j tels que $C[i,j]=1$.

Un exemple est fourni par la figure 83 et le tableau 17. En configurant une portée maximale de 20m, les nœuds 5 et 6 forment deux îlots isolés. On notera les 0 de la diagonale principale : en sans fil, un nœud radio ne peut pas détecter sa propre transmission sur le médium.

Tableau 17 : Matrice de connectivité construite par le médium

	N1	N2	N3	N4	N5	N6
N1	0	1	1	0	0	0
N2	1	0	1	0	0	0
N3	1	1	0	1	0	0
N4	0	0	1	0	0	0
N5	0	0	0	0	0	0
N6	0	0	0	0	0	0

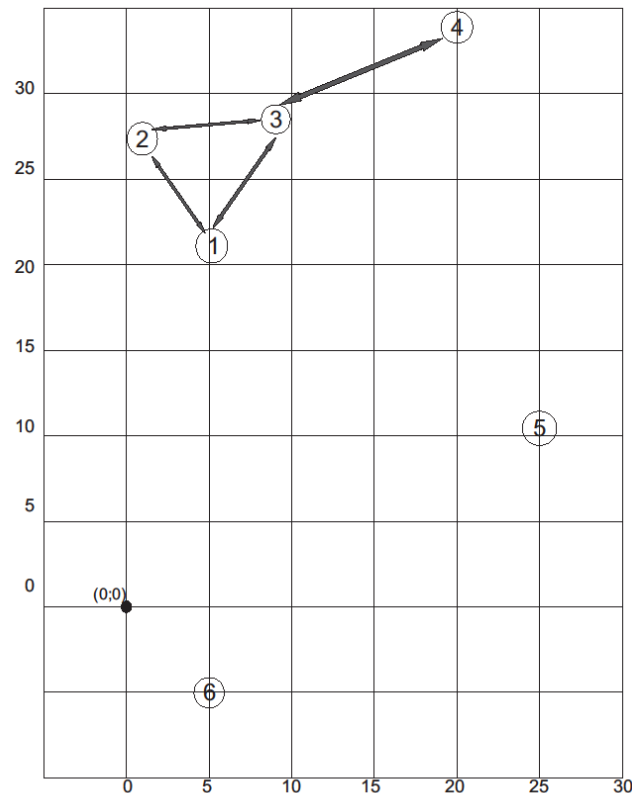


Figure 83 : Topologie d'exemple pour les zones de portée

- Effets du multitrajet** Soient deux modules radio séparés par une distance d . Un message de l'un à l'autre n'arrivera à destination qu'en t secondes, t étant directement lié à la distance. En combinant les instants de réception et d'émission du message selon les formules du TOA, il est aisé de retrouver ce temps de vol. Ceci n'est malheureusement correct que dans un environnement parfait. Lorsque le multitrajet intervient, le signal détecté et décodé par la couche physique n'est pas forcément le signal originel, issu du chemin direct. Il s'agit parfois d'une copie ayant été réfléchié par l'environnement et donc souffrant d'un retard supplémentaire. Nous avons mis en place un mécanisme qui permet à notre simulateur de prendre en compte les effets du multitrajet sur l'instant de réception. Le temps de vol est alors modélisé par :

$$t_{vol} = \frac{dist(A, B)}{c} + multipathFactor \times random[ns] \quad (88)$$

La variable c correspond à la vitesse de la lumière et $random$ est une variable aléatoire prenant ses valeurs dans l'intervalle $]0 ; 2[$. Le paramètre $multipathFactor$ influence l'amplitude possible du retard subi par la transmission dans un

environnement donné. Sa valeur sera choisie de manière à renforcer la ressemblance entre les plateformes réelle et virtuelle. Sa sélection sera présentée dans la phase d'ajustement de *DokoSim* à la réalité.

- **Collisions** Le médium possède une variable lui indiquant qu'une transmission est en cours. Une fois l'émission terminée, la source provoque la mise à jour de ce champ. Lorsqu'un nœud décide de transmettre une trame, cette même variable, avec la matrice de connectivité, est utilisée pour déterminer si une collision sera générée ou pas.
- **Pertes de trames sur les liens** Un certain taux de pertes de trames a été configuré. Pour chaque transmission, le médium se réfère à cette information afin de décider s'il délivre la notification de message entrant à la destination.

Dans l'implémentation de SDS-TWR, la présence des ACK permet de mettre à jour les estampilles temporelles en fonction du succès ou de l'échec de la transmission : considérons par exemple, sur la figure 84, la perte de l'acquittement lié à la trame *START*. Une exécution normale mènerait au stockage de t_1 et t_3 après réception de l'ACK et de t_2 et t_4 après émission du même ACK. Si l'acquittement émis en t_3 est perdu, le délai d'attente de la réponse correspondant au paramètre *MACAckWaitDuration* de la norme expire et une nouvelle trame *START* est émise. L'information t_1 est mise à jour au niveau de M. Le nœud A, à la réception de cette trame, aura déjà entamé la préparation de la trame *Reply1*. Le numéro de séquence de *START* permettra de reconnaître un doublon et de ne pas le faire passer aux couches supérieures. Un acquittement sera néanmoins émis et t_4 généré. Par conséquent, le taux de pertes, s'il a une incidence sur la durée de l'échange, peut également altérer la qualité des résultats car les paires d'estampilles qui seront combinées ne seront pas cohérentes. Cette situation mène typiquement à des valeurs extrêmement élevées qui peuvent être filtrées par le code.

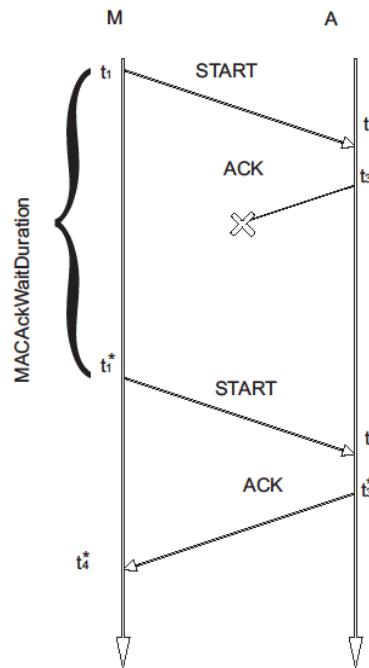


Figure 84 : Exemple de pertes de trames au cours de TWR

- **Respect des durées réelles de traitement** Nous avons voulu nous rapprocher au maximum de la réalité dans les aspects temporels du problème. Pour ceci, nous avons tenu compte des lignes directrices du standard mais aussi des contraintes imposées par le matériel. Dans le premier groupe, nous retrouvons des spécifications comme le débit de l'interface air, le délai séparant la fin de la réception d'une trame et la génération de l'acquittement associé.

Le second groupe considère les durées liées à la plateforme que nous voulons simuler :

- création des différents types de trames : SSDS-TWR utilise des trames de taille fixe pour le contrôle de l'exécution de la localisation et l'échange des structures de données. Nous avons donc mesuré ces durées sur une plateforme réelle à l'aide d'un oscilloscope et d'un analyseur d'états logique et avons utilisé les résultats pour configurer les délais de notre simulateur. L'implémentation de PSDS-TWR bénéficie de ces informations car il est critique que le délai, entre la réception de l'invitation à transmettre et le début de la réponse, permette aux ancres précédant le nœud courant dans la liste de répondre sans collision,
- transfert des trames à travers le port SPI : la traversée de ce lien introduit un décalage entre l'instant de transmission tel qu'il est vu par la couche MAC et

celui connu par la couche physique. Il est fonction du débit du port SPI et de la taille des données transmises.

- **Transceiver occupé** Nous avons également implémenté le comportement de l'émetteur-récepteur radio en ce qui concerne l'unicité de l'état. En effet, un *transceiver* ne peut émettre et recevoir en même temps. Si une émission est programmée mais qu'une réception commence avant que la transmission ne commence, cette dernière sera retardée pour permettre à la réception de se terminer.
- **Tailles des entêtes** Pour calibrer les en-têtes de trames, nous avons exploité l'amendement de 2007 du standard IEEE pour les réseaux de capteurs [54]. Ce document spécifie entre autres, pour une interface radio et un débit donnés, la taille des symboles ainsi que leur nombre dans les différents champs d'entête.
- **Buffers de couche PHY** Comme nous l'avons déjà dit, notre plateforme physique possède, en plus de sa capacité à mesurer la distance, une caractéristique la mettant à part des autres composants que nous avons rencontrés. Il s'agit du double *buffer* de la couche physique. Généralement, les *transceivers* maintiennent un unique espace de stockage pour les trames entrantes. Cet espace est souvent calibré pour ne contenir qu'une seule trame. Si une nouvelle trame est reçue avant que la précédente n'ait été entièrement transmise à la couche supérieure, l'ancienne se perd. Pour éviter ce problème, deux buffers sont disponibles en réception. Pour les mêmes raisons, nous avons doté la couche PHY de nos nœuds virtuels de deux *buffers* dans *DokoSim*.

4.2.4.2. Différences

Cette section liste les aspects que nous avons choisi de ne pas implémenter ainsi que les différences notables par rapport au matériel.

- **Capture de l'instant d'émission des trames** La capacité à déterminer les instants d'émission et de réception des trames avec une résolution d'une nanoseconde est donnée aux nœuds de notre simulateur alors qu'il s'agit du verrou majeur imposé par le matériel : une fois l'acquittement désactivé, l'instant d'émission ne peut être connu qu'avec une précision de $30.5\mu\text{s}$, ce qui ne convient pas pour les échelles auxquelles nous voulons réaliser la localisation. Avec notre simulateur, nous nous permettons une incursion sur les plateformes à venir [29], sur lesquelles il devrait

être possible de détecter avec une erreur acceptable l'instant de transmission d'une trame.

- **Aspects non implémentés**

- **Puissance du signal** Bien que l'information de puissance reçue soit très commune dans les réseaux sans fil, nous n'avons pas inclus son développement dans notre simulateur car pour le moment, nous ne l'exploitons pas. Néanmoins, le médium a tous les outils en main pour informer le récepteur de la puissance à détecter : la distance séparant les nœuds et la possibilité d'effectuer des tirages de variables aléatoires. Il suffira de lui fournir la puissance d'émission et le modèle choisi pour intégrer aux valeurs manipulées le niveau de puissance reçue.
- **Mobilité** Nous envisageons des applications où le mobile se déplace à vitesse réduite, de l'ordre d'1m/s (personne âgée, robot ménager...). Du point de vue de la localisation, le mouvement est statique durant l'échange de localisation. Son impact sur les résultats est donc négligeable. Par contre, la succession des exécutions du protocole et de l'algorithme pourra permettre de suivre le déplacement global du porteur du nœud mobile.
- **Réservation du *star interval*** Nous n'avons pas mis en place la pile protocolaire permettant d'organiser l'émission sans collisions des *beacons* et la réservation du *star interval* car notre objectif n'est pas la validation de cet aspect du protocole. Le lecteur intéressé par cette problématique pourra se référer à [14], [69] ou [71].
- **Modèle énergétique des nœuds** Limiter la consommation énergétique des nœuds d'un WSN est une condition essentielle de la viabilité de la solution. Cette consommation est fonction notamment des durées d'écoute sur le médium. Bien que nous n'ayons pas implémenté une classe représentant la batterie du module et capable d'émuler l'évolution de la charge dans le temps, nous avons proposé dès le départ de limiter l'*idle listening* par le biais de PSDS-TWR.
D'autre part, une étude théorique de cet aspect du protocole a été réalisée en section 3.1.5.
- **Le *skew* des horloges** Bien que la variation dans le temps de la fréquence des horloges puisse avoir une incidence sur les résultats, ce comportement n'a pas été intégré à notre solution.

- **Méthode d'accès au médium** Bien que nous fournissions des indications au sujet des étapes de préparation de la localisation, nous nous sommes concentrés sur les interactions ayant lieu durant le *star interval*. Par conséquent, nous n'avons pas eu à implémenter une méthode d'accès, le mobile étant le seul autorisé à initier un échange.

4.2.5. Test de SSDS-TWR

Cette opération sert deux objectifs : tester le protocole est le premier mais l'ensemble des expériences devra également permettre de configurer le simulateur. Au terme de cette phase, DokoSim sera à même de supporter le test de PSDS-TWR. Nous allons donc, dans les sous-chapitres suivants, paramétrer le *multipathFactor* ou MP ainsi que la probabilité de pertes par interférences.

4.2.5.1. Paramétrage du *multipathFactor*

Nous avons configuré les balises de manière à respecter la disposition des nœuds dans notre implémentation réelle. Nous avons fait varier les valeurs du *multipathFactor* afin d'identifier la valeur qui reflète le mieux la réalité. Dans chaque cas, nous examinons l'erreur de *ranging* puis celle de localisation.

multipathFactor=0

Cette configuration résulte en un encadrement systématique de la vraie position par les anneaux. Le recouvrement de l'intersection des anneaux par une matrice mène à une solution très proche de l'objectif (figure 85(a)). L'erreur de position n'est pourtant pas nulle. Ceci s'explique par la nature discrète du simulateur : les instants capturés correspondent à la prochaine valeur entière du temps en nanosecondes et non à la valeur réelle. Par exemple, si le nœud A1 envoie une trame à A2, le signal doit atteindre le récepteur en distance/vitesse nanosecondes. Cette valeur étant décimale, l'instant d'impact détectable est calculé suivant l'équation 89.

$$t_{impact} = t_{courant} + \text{ceiling}\left(\frac{\text{distance}}{\text{vitesse}}\right) \quad (89)$$

Ici la fonction *ceiling(u)* retourne le plus petit entier supérieur à *u*. Par conséquent, même dans les conditions parfaites, les distances mesurées par les nœuds virtuels ne sont pas totalement exactes.

multipathFactor=3 et 4

L'effet du multitrajet commence à se faire sentir : le mobile se retrouve moins souvent à mi-chemin entre $R_{\min,i}$ et $R_{\max,i}$. Ceci est dû au fait que les distances mesurées s'éloignent de la réalité. L'erreur reste néanmoins limitée et la solution existe généralement (figure 85(b)).

multipathFactor=5 et multipathFactor=7

Avec cette configuration, les résultats sont nettement plus dégradés. Certaines situations du monde réel commencent à se produire (§ 4.1.3.1). Par exemple, la solution correspondant à l'ensemble vide devient plus probable : ceci se produit lorsque les intersections n'existent pas. De même, la position réelle du mobile peut ne pas être contenue par un ou plusieurs anneaux. Sur la figure 85(c), nous avons choisi de représenter des situations où la vraie position se retrouve à l'extérieur de la zone de recherche.

Par exemple, au niveau du prototype, il arrive parfois que les résultats de localisation soient particulièrement dégradés lorsque la quantité d'énergie résiduelle passe en-dessous d'un certain seuil. Les résultats obtenus avec un MP de 7 se rapprochent de cette situation. Bien que l'on cherche à éviter cette dernière, elle constitue un cas d'étude pour la localisation.

Le tableau 18 résume les résultats d'erreur moyenne pour la situation réelle et le simulateur. Dans la suite, nous utiliserons un *multipathFactor* de 7 en raison de la similitude entre les résultats et le comportement réel de notre système. Nous utiliserons comme références les résultats de la deuxième campagne de mesure (§ 4.1.3.1)

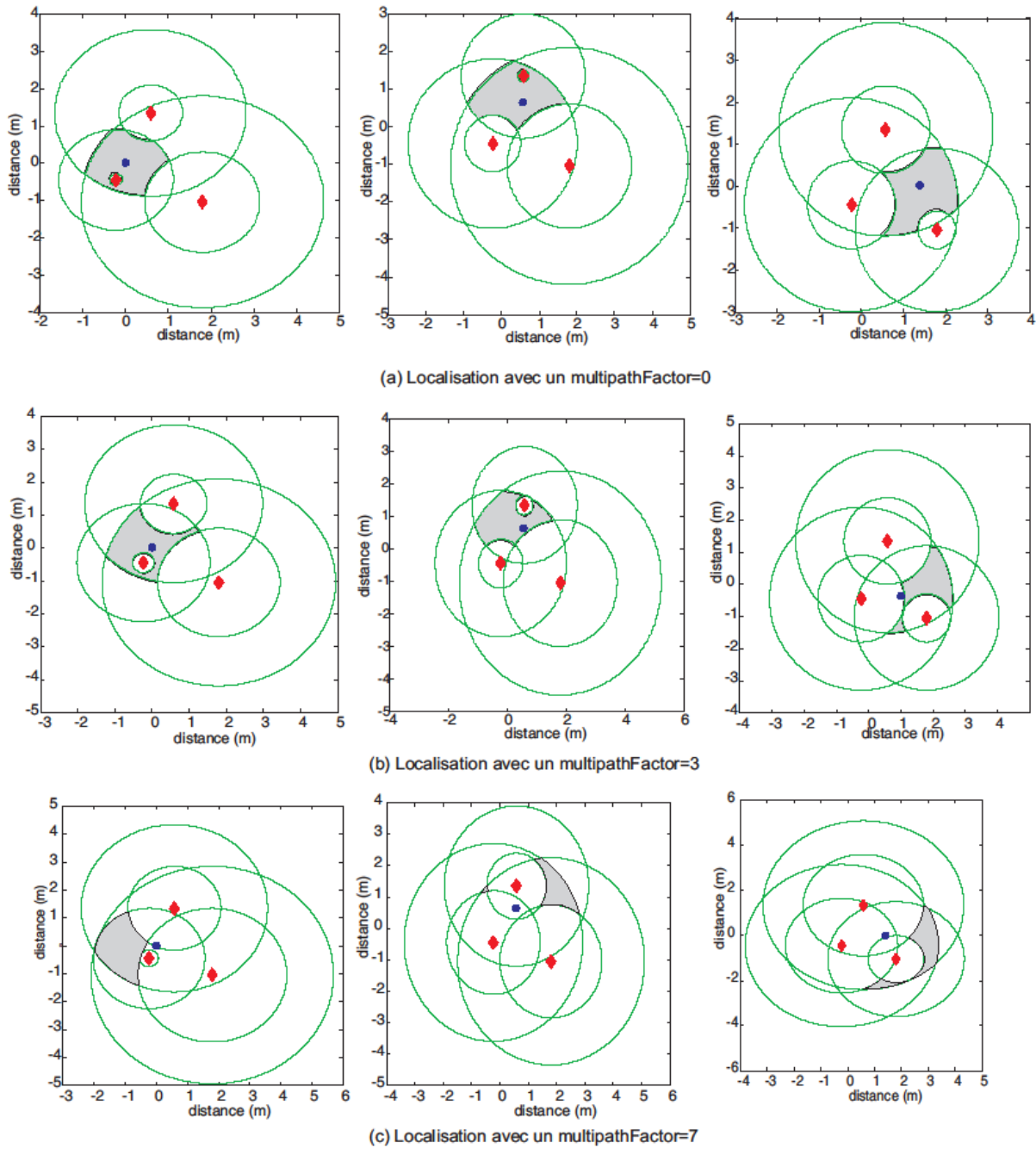


Figure 85 : Effet de la variation du multipathFactor sur les résultats

Tableau 18 : Comparaison des erreurs de localisation en fonction du *multipathFactor*

Situation	Réalité (prototype)	MP=0	MP=3	MP=4	MP=5	MP=7
Erreur moyenne (mètres)	1,16	0,269	0,6014	0,89	1,04	1,38

4.2.5.2. Paramétrage du taux de pertes de trames

Dans la suite du manuscrit, nous considérons le cas où tous les nœuds opérant sur les mêmes fréquences et utilisant les mêmes deux couches basses, respectent le protocole de négociation du *star interval*. Un nouvel arrivant se met d'abord à l'écoute des messages échangés afin de déterminer les intervalles durant lesquels il est autorisé à émettre. Ceci permet de négliger les collisions provenant du réseau lui-même au cours du *star interval* car seul le mobile a le droit d'initier un échange. Nous nous pencherons donc sur les effets des interférences sur la performance de SSDS-TWR. Nous examinerons l'impact sur la qualité de la localisation mais aussi sur la durée de l'échange. La configuration est donnée dans le tableau 19. Nous avons choisi de considérer qu'une trame sur 100 est perdue en raison des interférences.

Tableau 19 : Configuration du simulateur pour SSDS-TWR

Nombre de positions occupées par le mobile	16
<i>Frame error rate</i>	10^{-2}
<i>Multipath Factor</i>	7

Tableau 20 : Résultats de simulation pour SSDS-TWR avec interférences

Critère	Pas d'interférences	Taux de pertes 1%
Durée d'exécution du protocole (ms)	26,63	26,70 (std 0.21)
Erreur moyenne (m) sans ordonnancement	1,46	1,41
Erreur moyenne (m) avec ordonnancement	1,16	1,07

Afin d'évaluer la durée du processus de mesure, nous avons configuré notre simulateur à l'aide des informations données au chapitre 3.1. Sur le prototype, la durée séparant le début et la fin de la phase de *ranging* est de 30ms.

Lorsque les interférences sont ignorées, la durée de l'échange de SSDS-TWR correspond à 26.63ms pour 3 balises. Un taux de pertes de 10^{-2} n'a pas une incidence significative sur la durée du processus (tableau 20). Dans le pire des cas, 200 μ s ont été ajoutées à la durée d'exécution du protocole par la perte d'une trame de données.

Pour ce qui est de la précision, les résultats sont similaires, qu'il y ait ou non des interférences. Ceci est dû au fait que l'influence du multitrajet est plus notable que celle des pertes. En effet, le paramètre *multipathFactor* a un impact sur chaque transmission alors que notre configuration ne provoque la perte que d'une trame sur 100.

Finalement, le tableau 20 indique la valeur moyenne de l'erreur de localisation lorsque l'ordonnement est appliqué. Le gain est comparable à ce qui a été observé dans la réalité (figure 86). Des intervalles d'erreurs ont été définis pour permettre de classer les résultats de l'utilisation de *iRingLa* avec (en rouge) puis sans ordonnancement (en bleu) des anneaux. La même translation de l'effectif des échantillons vers la gauche peut être constatée, de même que l'existence de cas auxquels l'ordonnement n'apporte guère de contribution. Dans l'intervalle [2m ; 2,5m] peuvent être relevés des cas où l'ordonnement est mis en échec en raison du non respect de l'hypothèse d'appartenance de la vraie position aux plus petits anneaux.

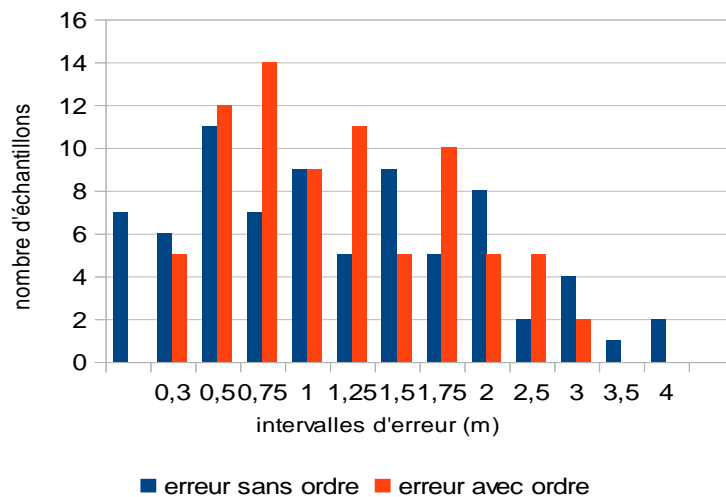


Figure 86 : Evolution de l'erreur de localisation par ordonnancement des anneaux

4.2.5.3. Validité du simulateur

Les expériences précédentes ont montré la similitude des comportements de notre simulateur et du prototype. Particulièrement, la configuration du *multipathFactor* à 7 et du taux de pertes de trames à 10^{-2} permet de simuler le fonctionnement de notre plateforme dans l'environnement qui nous concerne. Dans la suite, cette configuration sera conservée pour tous les scénarii.

4.2.6. Test de PSDS-TWR

Le protocole PSDS-TWR a pour objectif de paralléliser l'exécution de SDS-TWR entre un mobile unique et un ensemble d'ancres ayant accepté de participer à l'opération de localisation. Le verrou de ce protocole provient de l'élimination des acquittements : il n'existe pas de recours pour les trames perdues. Par conséquent, après avoir illustré son

fonctionnement dans un monde virtuel parfait, nous examinerons l'impact des pertes de trames sur la performance. Dans cette section, la comparaison avec une plateforme réelle ne sera pas possible.

4.2.6.1. Simulation dans un environnement sans pertes

Dans un premier temps, les pertes liées aux interférences seront négligées. Le seul effet du multitrajet sera pris en compte. Une première analyse a été réalisée pour déterminer la valeur des délais D_i à respecter entre les réponses successives des ancres.

Le déclenchement de la deuxième phase du processus au niveau du mobile est contrôlé par un *timer* initialisé par une valeur $D_{1,2}$. A l'expiration du *timer*, M entame la création de la trame *Data_Request*. En tenant compte du temps de création et de son transfert à la couche physique, son émission ne doit pas venir perturber l'arrivée des trames de réponses venant des ancres. Ces envois étant réalisés sans méthode d'accès au médium et les acquittements n'étant pas requis, ces collisions seraient désastreuses [46]. $D_{1,2}$ s'étend de la fin de la transmission de la trame *LocationStart* au début de la constitution de *DataRequest*.

Ces différents paramètres sont illustrés sur la figure 87. Dans la suite, nous considérerons 3 ancres.

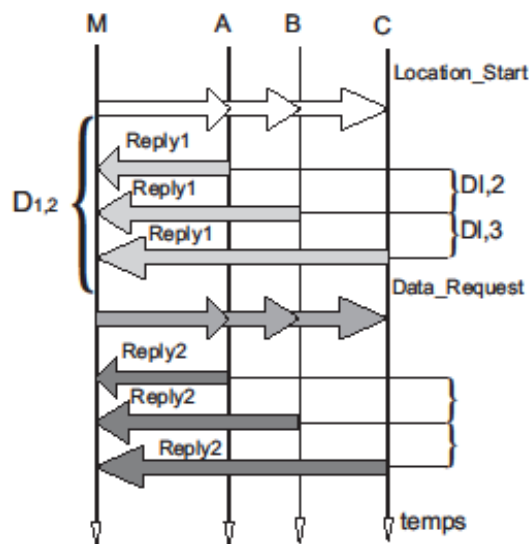


Figure 87 : Paramètres temporels de PSDS-TWR

Les différentes trames *Reply1* étant toutes de la même taille, nous considérerons que les D_i sont constants et égaux à D_1 . En considérant que la trame *Location_Start* est reçue simultanément par les 3 ancres, le formatage des trois trames se terminera au même instant sur les trois nœuds ancres. A partir de cet instant, chaque ancre configurera un *timer* avec un

délai correspondant à sa position p_i dans la liste proposée par M dans la trame *Location_Start*. Le calcul est réalisé par l'équation 90.

$$delay_i = (p_i - 1) * DI \tag{90}$$

De cette manière, la première ancre de la liste émet immédiatement sa trame et les autres suivent à *DI* secondes d'intervalle. La valeur de *DI* a été choisie de manière à permettre à chaque ancre de terminer la transmission de sa trame. Nous avons choisi un *DI* de 500µs entre deux transmissions pour éviter les collisions. Cette valeur tient compte du temps pendant lequel le médium est occupé (entre 304 et 352 µs selon le type de trame) mais aussi du temps nécessaire à la libération d'un des *buffers* de couche physique.

L'intervalle séparant la fin de la transmission de *Location_Start* du début de la création de *Data_Request* dépend du nombre de balises partenaires (équation 91)

$$D_{1,2} = n * D_e \tag{91}$$

avec :

n : nombre de partenaires

D_e : un délai élémentaire

Une fois la trame *Data_Request* émise, un second *timer* est armé : à son expiration, M cesse d'écouter le médium et exécute son algorithme de localisation avec les informations disponibles.

Dans la réalité, la fin de chaque phase serait contrôlée par l'occurrence du premier des deux événements : réception du message provenant de la dernière balise de la liste ou expiration du *timer*.

Le tableau 21 résume les valeurs des différents paramètres temporels tandis que le tableau 22 résume les résultats obtenus dans cette configuration. On se souviendra que la configuration du *multipathFactor* est conservée (MP=7).

Tableau 21 : Paramétrage temporel

Délai élémentaire <i>D_e</i> : Attente de Reply1	550µs
Délai inter-balise <i>DI</i>	500µs
Délai élémentaire <i>D_e</i> : Attente de Reply2	750µs

Tableau 22 : Résultats

Durée d'exécution du protocole (3 ancrés)				7,483ms
Erreur moyenne obtenue sans ordonnancement (m)				1,71
Erreur moyenne obtenue avec ordonnancement (m)				1,35

L'extraction de l'erreur et de la durée d'exécution de chacune des expériences a fourni les données du tableau 22. Du point de vue de la performance de localisation, la situation s'est dégradée avec une erreur moyenne de 1,71m et une erreur inférieure à 2m dans 70% des cas. Cette dégradation est due au paramètre *multipathFactor*. Les équations 92 et 93 rappellent la formule permettant d'estimer le temps de vol en tenant compte des délais introduits par le multitrajét. Les variables t_i représentent les instants d'émission et de réception tandis que t_{vol} représente le temps de vol réel. Dans chacun des cas, T correspond à l'estimation.

$$2T_S = \frac{t_3 - t_0 - (t_2 - t_1)}{2} + \frac{t_7 - t_4 - (t_6 - t_5)}{2} = 2t_{vol} + \frac{\alpha + \beta + \theta + \rho}{2} \quad (92)$$

$$2T_P = \frac{t_3 - t_0 - (t_2 - t_1)}{2} + \frac{t_7 - t_4 - (t_6 - t_5)}{2} = 2t_{vol} + \frac{\alpha + 2\beta + \rho}{2} \quad (93)$$

Les expressions de T_S et T_P diffèrent au niveau des délais additionnels. Nous considérons que ces variables α , β , ρ et θ sont des gaussiennes, donc des normales décalées de leur moyenne. Afin de simplifier les calculs, la moyenne et la variance sont les mêmes pour toutes ces variables que nous considérons indépendantes. Les expressions 94 et 95 indiquent les lois suivies par les normales équivalentes.

$$\alpha + \beta + \theta + \rho \sim N(0,4) \quad (94)$$

$$\alpha + 2\beta + \rho \sim N(0,6) \quad (95)$$

Les délais seront pris dans un intervalle plus vaste dans le cas de PSDS-TWR, ce qui explique la différence de performance des deux protocoles dans les mêmes conditions.

La figure 88 indique la répartition de l'erreur pour l'ensemble des points. Dans quelques situations, l'erreur dépasse 3m. L'analyse de ces cas a mis en évidence la ressemblance entre le comportement du simulateur et celui de la mesure réelle : alors que certaines erreurs auraient pu être réduites par l'application d'un ordonnancement sur les anneaux, dans d'autres cas, cette phase n'aurait pas amélioré les résultats car l'une des hypothèses fondamentales n'était pas respectée, à savoir que la vraie position est contenue dans le plus petit des anneaux. La figure 89 illustre ces situations, obtenues par simulation. Le cas où l'ordonnancement

n'introduit aucune amélioration correspond à la figure 89(a) tandis que la figure 89(b) indique la possibilité de réduire l'erreur en entamant les recherches dans l'intersection des deux plus petits anneaux, centrés sur les balises A et B.

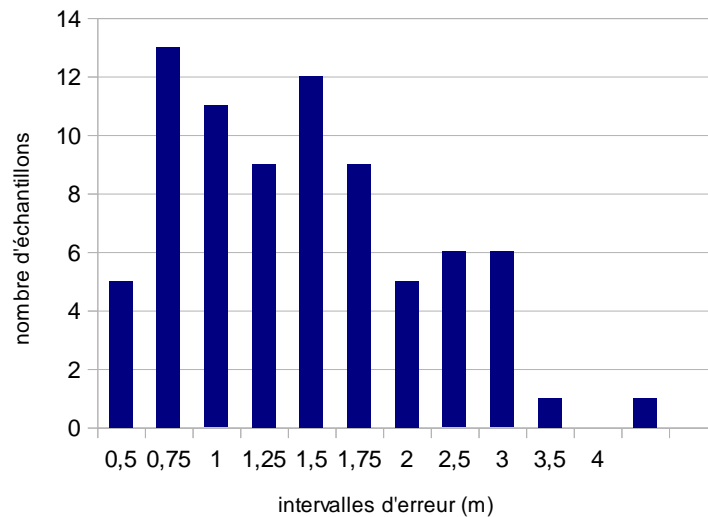


Figure 88 : Répartition de l'erreur pour PSDS-TWR

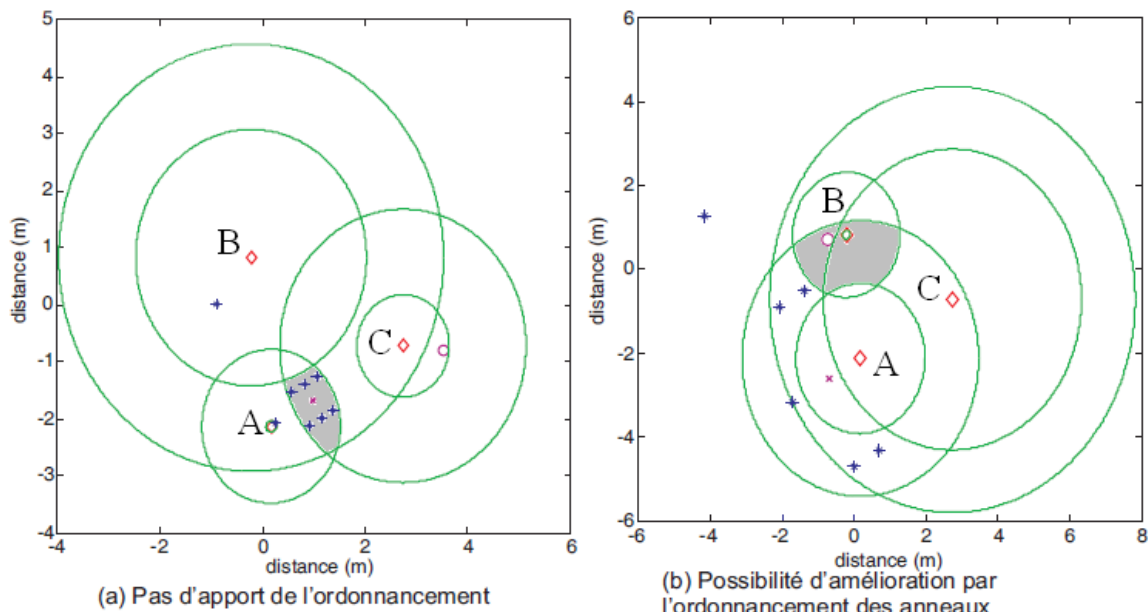


Figure 89 : Cas particuliers observés pendant la simulation de PSDS-TWR

De ces mêmes résultats, nous avons extrait la durée d'exécution de PSDS-TWR pour 3 ancres. Le gain de temps est appréciable, la durée totale des échanges ayant été réduite à 7,48ms contre 30ms pour SSDS-TWR. Ceci permettrait par exemple de contrer la détérioration de la précision par deux à trois exécutions successives de PSDS-TWR dont les résultats seraient combinés pour aboutir à la position finale. D'un autre point de vue, le fait de rendre très bref l'échange de localisation réduit les chances d'interception des données par

une entité mal intentionnée tout en augmentant la disponibilité du médium pour des transmissions de données.

4.2.6.2. PSDS-TWR face aux interférences

Les acquittements ayant été supprimés, aucune réémission n'est prévue pour les trames. Ces pertes conduisent à l'exclusion d'une ou plusieurs ancres de l'algorithme de localisation. En effet, le nombre de trames étant réduit, chaque message devient critique. Par conséquent, le nombre d'ancres à contacter devrait, dans un environnement réel, être fonction du taux de pertes sur le médium. La collecte par les nœuds du WSN de statistiques liées aux pertes sur le médium permettrait d'adapter automatiquement la valeur de n à l'environnement.

Dans notre simulateur, nous n'avons pas mis en place ce dispositif d'adaptation au milieu.

Tableau 23: Résultats de simulation pour PSDS-TWR avec ajout d'interférences

Durée d'exécution du protocole (3 ancres)				7,483ms (std 0.000)
Erreur	moyenne	obtenue	sans	1,87
ordonnancement (m)				
Erreur	moyenne	obtenue	avec	1,45
ordonnancement (m)				

Le tableau 23 présente les résultats de la localisation à l'aide de PSDS-TWR pour deux versions de *iRingLA* : l'une intégrant l'ordonnancement des anneaux et l'autre non.

Tout d'abord, on notera qu'en termes de durée d'exécution, l'ajout d'interférences ne modifie pas les performances de PSDS-TWR. En effet, les réémissions étant supprimées, le phénomène d'accroissement de la durée de l'échange ne se produit pas.

En ce qui concerne la précision de la localisation, l'impact de l'ordonnancement se fait à nouveau sentir. La figure 90 présente la répartition des erreurs en fonction de la distance. Les données issues de l'exécution sans ordonnancement des anneaux correspondent aux barres bleues tandis que les rouges symbolisent les résultats après classification. Nous pouvons observer une recrudescence de cas pour lesquels l'erreur dépasse 350cm. Il s'agit une fois de plus de situation où le besoin d'un ordonnancement des anneaux se manifeste. Ceci est démontré par le décalage à gauche du graphique bleu.

Un second phénomène se produit durant ces simulations sans affecter directement les résultats : il s'agit de pertes de messages conduisant à une réduction du nombre d'échantillons. Dans notre cas, près de 8% des exécutions n'ont pas mené à une position

calculée par *iRingLA*. Ce choix a été motivé par la conclusion qu'*iRingLA* était inutile et excessif dans ces situations. En effet, si deux ancrs sont disponibles, il suffit de sélectionner un point sur le segment de droite reliant les partenaires de localisation en fonction des informations de distance obtenues. Lorsque trois ancrs sont disponibles, l'impact des interférences peut donc être qualifié de binaire : la position peut ou ne peut pas être calculée par *iRingLA*.

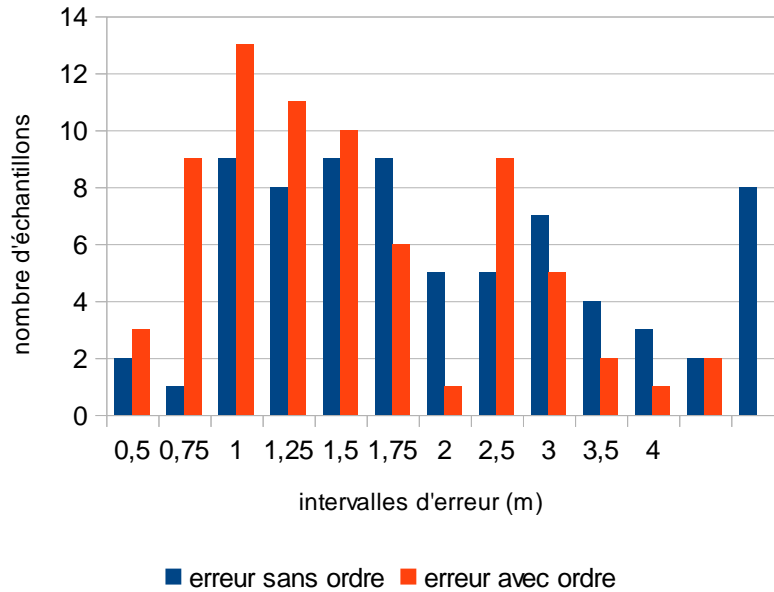


Figure 90 : Répartition des erreurs pour PSDS-TWR avec intégration des pertes de trames

4.2.6.3. Gain temporel de PSDS-TWR

Le gain temporel permis par PSDS-TWR, par rapport à SSDS-TWR, se conserve lorsque le nombre de balises concernées augmente. La figure 91 représente cette évolution selon le nombre des balises. Les durées peuvent s'écrire suivant les formules 96 et 97.

$$T_{PSDS-TWR} = cte + n * t_1 + n * t_2 \tag{96}$$

$$T_{SSDS-TWR} = n * T \tag{97}$$

avec :

n : le nombre d'ancres

t_1 et t_2 : les délais élémentaires dans les phases 1 et 2 respectivement de PSDS-TWR

T : la durée d'un échange SDS-TWR.

Dans le cas de SSDS-TWR, selon l'hypothèse de liens radio à faible portée (inférieure à 100m), la différence de durée entre les exécutions successives du protocole peut être considérée comme négligeable. En effet, l'intervention du microcontrôleur au cours de l'exécution implique des durées de l'ordre de la milliseconde alors que des liens de moins de 50m introduisent des délais de l'ordre de la centaine de nanosecondes. La durée totale ne dépend donc que du nombre d'ancres.

Le protocole PSDS-TWR, quant à lui, fait intervenir deux périodes pendant lesquelles le mobile émet les trames *Location_Start* et *Data_Request*, puis *n* intervalles pendant lesquels les ancres émettent *Reply1* et encore *n* intervalles dédiés au transfert de *Reply2*.

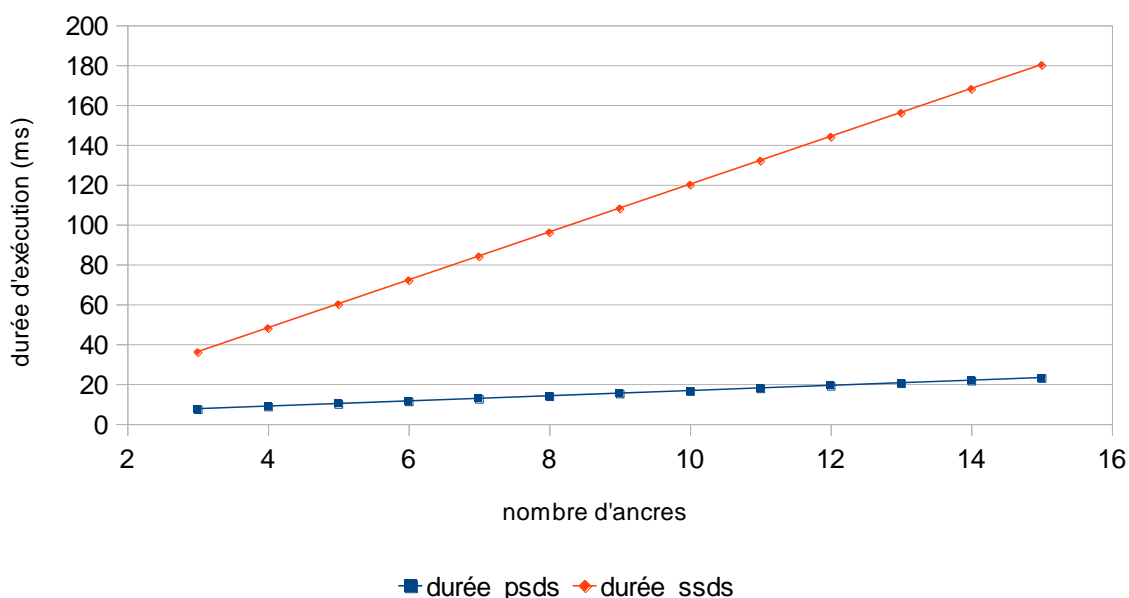


Figure 91 : Durée d'exécution selon le protocole.

A partir des résultats obtenus et des configurations adoptées, nous avons choisi les valeurs de T , t_1 et t_2 (tableau 24). La constante associée à l'exécution de PSDS-TWR tient compte de la préparation des trames et de leur émission après le transfert par le bus SPI.

Tableau 24 : Paramètres temporels pour la comparaison de PSDS-TWR et SSDS-TWR

Paramètre	Valeur
t_1	550 μ s
t_2	550 μ s
T	12ms

Bien que les deux droites soient affines, la croissance de la courbe bleue, correspondant à PSDS-TWR est beaucoup plus lente car les durées élémentaires associées à chaque ancre sont faibles comparativement aux valeurs utilisées pour SSDS-TWR.

4.2.7. Conclusion sur PSDS-TWR

Dans ce chapitre, nous avons procédé à une évaluation du protocole PSDS-TWR à l'aide d'un simulateur dont les paramètres ont été préalablement configurés grâce au protocole SSDS-TWR. Le principe de PSDS-TWR consiste à supprimer les trames d'acquittement afin de paralléliser les opérations de mesure de distance. Grâce à ce procédé, nous avons pu réduire de manière significative la durée d'exécution du protocole. Les simulations indiquent une réduction du temps d'exécution du protocole de 27ms à 7.5ms pour des scénarii impliquant 3 ancrés.

De la même manière, la quantité de trames échangées pour un même nombre d'ancres partenaires est deux fois moindre que lors de l'utilisation de SSDS-TWR. Ces gains sont obtenus aux prix d'une certaine diminution de la robustesse : en effet, toute perte de réponse de la part des ancrés implique l'élimination des informations liées à cette dernière durant le calcul de la position. Dans certains cas, ceci peut conduire à un ensemble de données insuffisant pour la localisation en 2D.

L'impact négatif de l'utilisation de PSDS-TWR sur les résultats de localisation est mesuré : l'erreur moyenne lors de son utilisation est supérieure d'une trentaine de centimètres à celle obtenue avec SSDS-TWR.

L'algorithme *iRingLA* et l'impact positif de l'ordonnancement des anneaux a pu être observé. Si, sur l'erreur moyenne, la réduction n'est que de quelques centimètres, dans les situations où cet ordonnancement est appliqué, les erreurs peuvent être trois fois plus faibles qu'avec un ordonnancement statique.

Dans le futur, nous souhaitons prototyper le protocole PSDS-TWR afin de mesurer en environnement réel sa performance et le comparer à SSDS-TWR. Il sera alors intéressant d'étudier des algorithmes permettant d'ajuster le nombre d'ancres contactées en fonction de l'environnement radio ou encore de l'énergie disponible au niveau des nœuds.

4.3. Bilan sur le simulateur *DokoSim*

Au vu des besoins, nous avons choisi de développer notre propre simulateur plutôt que d'en modifier un existant. Au-delà de cette expérience profitable de modélisation du comportement matériel, nous avons pu réaliser une étude du protocole et de l'algorithme proposés, tant sur son comportement dans un environnement perturbé que sur des aspects de

performances comme la durée d'exécution et la précision. De plus, le simulateur nous a permis de contourner les limitations matérielles imposées par l'architecture Nanotron, notamment les ACK obligatoires, pour implémenter notre contribution, PSDS-TWR, qui a clairement montré sa supériorité.

Dans le cadre d'un transfert de technologie, certaines améliorations pourront sans doute être apportées à notre outil comme une intégration plus complète du standard IEEE 802.15.4 pour la couche MAC ou encore la prise en compte du comportement de la puissance émise avec la distance.

CONCLUSION ET PERSPECTIVES

A différentes échelles, les réseaux se sont intégrés à la vie humaine. Hier, le LAN, le MAN et le WAN s'ingéniaient à nous interconnecter. Aujourd'hui, les PAN s'intéressent au confort, à la protection et au service à la personne. D'un côté, par les capacités de mesures de paramètres environnementaux embarquées, les nœuds de ces infrastructures sont capables de suivre l'évolution tant de la température d'une pièce que l'état de santé d'un malade ou encore l'évolution d'un incident de type « marée noire » [48]. Un complément à la richesse de ces offres est la localisation. Bien qu'il s'agisse d'une problématique aussi ancienne que le concept de mobilité, la détermination de la position d'un élément donné reste un sujet d'étude intéressant. Spécifiquement, lorsque ce service doit être rendu par un réseau de capteurs sans fil, l'investigation devient passionnante.

Dans le premier chapitre, nous avons introduit différentes solutions proposées par le passé. Une approche modulaire a été retenue dans laquelle nous considérons tout système de localisation comme formé d'un bloc responsable des échanges d'information entre les collaborateurs, d'un second chargé de la correction des données brutes et finalement d'un troisième implémentant l'algorithme de localisation à proprement parler. Dans le cadre des solutions dites *range-based*, le protocole sera principalement dédié à la mesure en temps réel de caractéristiques utiles tandis que dans le cas des méthodes *range-free*, il sera chargé de la dissémination des informations relatives à la connectivité.

Les algorithmes exploités pour la localisation vont de la simple application de règles de géométrie au filtrage du signal pour l'élimination des erreurs. Ces concepts sont résumés ainsi que les exigences liées à leur utilisation. La diversité des pistes, tant sur le signal à retenir que dans la manière de manipuler les informations, nous a porté à redéfinir notre problématique comme la recherche d'une solution d'auto-localisation *range-based* pour réseau de capteurs sans fil. A partir de cette formulation, nous avons proposé différentes stratégies dont les principes mais également les possibles débouchés et perspectives sont introduits dans les paragraphes suivants.

Conclusion

Notre première contribution concerne l'algorithme de calcul de la position. Le chapitre 2.2 nous a permis de cerner la nécessité d'une méthode efficiente, c'est-à-dire capable de mener à bien sa mission avec un minimum de ressources. Dans le contexte de réseaux de capteurs sans fil, mémoire et puissance de calcul étant drastiquement réduites, l'algorithme devait se satisfaire d'une connaissance très limitée du réseau et d'un maigre jeu d'opérations. Notre proposition a été présentée sous le sigle *iRingLA*. Derrière ces consonances amusantes se cache un algorithme de localisation basé sur l'intersection des anneaux. Cette représentation de la distance mesurée a été retenue car elle intègre simplement la réalité des erreurs de mesures. Sur la base de ces anneaux, une recherche linéaire a été définie. Elle permet d'identifier la zone où se trouve le plus probablement le mobile.

Dans un second temps, nous nous sommes penchés sur le protocole de *ranging*. A l'origine, nous exploitions une adaptation d'un protocole décrit dans [54] et disponible sur notre plateforme matérielle. Au cours des expérimentations nous est venue l'idée de réduire le coût induit par les échanges. Ce coût peut se décliner en plusieurs versions. Nous avons adressé les aspects énergétiques et protocolaires du problème en proposant un nouveau protocole de mesure de distance par temps de vol, le *Parallel Symmetric Double-Sided Two-Way Ranging* (PSDS-TWR). En parallélisant les échanges, nous avons considérablement réduit l'impact du service de localisation sur la bande passante utilisée. En effet, la durée d'exécution du protocole croît significativement moins vite avec le nombre de balises que lorsque le protocole par défaut (SSDS-TWR) est utilisé. Ceci permet d'entrevoir des économies d'énergie appréciables pour l'ensemble du réseau.

Un concept nouveau a été également introduit. Il s'agit du *star interval*. Ressemblant par l'absence de mécanismes d'évitement des collisions aux GTS spécifiés dans la norme IEEE 802.15.4, il se distingue par le partage de ce slot temporel entre des coordinateurs, les nœuds ayant ce rôle pouvant exceptionnellement réserver un *star interval*. La négociation de cet intervalle fait l'objet de la section 3.1.1.

Finalement, nous avons réussi à nous passer du serveur de localisation classiquement utilisé dans les solutions *range-based* tout en ayant la possibilité d'annoncer, à l'instar de propositions nettement plus coûteuses, une erreur inférieure à 2m dans 70% des cas.

Perspectives

La recherche scientifique a pour vocation d'ouvrir de nouveaux champs. Nous présentons donc ici les principales possibilités offertes par notre étude.

Tout d'abord, l'étude de l'**adaptation à une faible densité d'ancres et/ou de voisins** est souhaitable. Notre solution exige que le mobile soit à portée d'au moins 3 nœuds de références. Ces références peuvent être par nature des ancres comme des mobiles connaissant leur position. En effet, il est possible d'imaginer que d'autres mobiles localisés puissent jouer le rôle d'ancres pour un mobile non - localisé.

La situation se complique quand aucun nœud de référence n'est disponible dans le voisinage immédiat du mobile. A dire vrai, cette situation embarrassante pour une solution *range-based* fait la force de propositions de type DV-hop telles que [75] et [46]. Ces travaux proposent des méthodes d'estimation de la position en fonction d'ancres lointaines, atteignables par plusieurs sauts. Il serait donc pertinent de développer une solution hybride, éventuellement désignée par rDV-Hop où le « r » représenterait l'aspect *range-based*. La combinaison de ces outils pourrait mener à une estimation plus précise de la distance dans des environnements défavorables. Il faudra toutefois veiller à limiter l'accumulation des erreurs de localisation de références en références.

Le problème de la diminution du nombre d'ancres, à portée du mobile, se présente également lorsque le nœud s'éloigne de la zone de couverture du réseau. Il est envisageable, pour un nœud disposant de capteurs adaptés, de définir un **fonctionnement en mode dégradé** dans lequel les dernières informations obtenues par notre procédé de localisation sont combinées avec des mesures de caractéristiques, éventuellement indépendantes du signal de communication, pour fournir une estimation de la position. Une combinaison de gyromètres et d'accéléromètres constitue un exemple de ces capteurs adaptés.

Parallèlement, l'**adaptation de nos travaux sur des plateformes plus performantes** constitue une seconde piste de recherche. L'objectif serait d'étudier l'apport de notre protocole ainsi que de l'algorithme *iRingLA* à des technologies de communication et de mesures plus sophistiquées. Dans cette optique, nous avons commencé à examiner des plateformes matérielles basées sur l'UWB. Nous nous attendons à ce que cette couche physique, mise en avant dans [54] pour son débit et ses capacités de *ranging*, améliore de manière significative les résultats de la localisation via notre approche.

A l'instar d'un système comme le GPS, il serait profitable de fournir à l'utilisateur une **indication de la qualité de la localisation**. Après nous être concentrés sur la mise en place de notre méthode de localisation, nous pouvons nous intéresser à doter le système de capacités d'autocritique. En effet, disposer d'une information de localisation ne suffit pas. Il est nécessaire de pouvoir indiquer aux clients la confiance pouvant être accordée aux estimations. Cette confiance, souvent qualifiée de consistance, doit pouvoir se déduire de la connaissance limitée détenue par le mobile. Les travaux actuels visent à déterminer les paramètres à combiner afin de développer cet aspect.

L'application de notre solution au monde réel appelle une étude de l'**adaptation dynamique du protocole à l'environnement**. Nous avons, dans nos expériences, souvent exploité le cas où trois ancres sont disponibles pour la localisation. En tenant compte de l'impact des pertes de messages sur PSDS-TWR, il peut être intéressant de doter le mobile d'un mécanisme de détermination dynamique du nombre d'ancres à contacter afin de maximiser la probabilité de succès d'une opération de localisation.

Dans le même ordre d'idée, nous pouvons envisager une **cohabitation des deux protocoles, PSDS-TWR et SSDS-TWR**. Dans un premier cas, la mobilité des nœuds indiquerait le protocole à utiliser : SSDS-TWR lorsque la mobilité est nulle ou réduite, et PSDS-TWR dans le cas d'une mobilité élevée. Une seconde approche consisterait à analyser le niveau de perturbations radios issues d'équipements externes au réseau. Dans le cas d'une influence forte sur les communications, une version améliorée de SSDS-TWR, robuste face aux pertes de messages de *ranging*, pourra être utilisée.

Dans le contexte humain, il est souvent question de *power naps* : il s'agit de brèves périodes de sommeil capables de restaurer une portion importante de l'énergie d'un individu. PSDS-TWR nous fournit une occasion d'étudier l'**intérêt énergétique de ces micro-sommeils** pour des nœuds sans fil.

Nous aimerions également affiner le processus de **négociation du *star interval***. Nous avons défini un intervalle dédié à la localisation, désigné par le terme *star interval*. Nous avons également spécifié le protocole de négociation de ce slot avec les ancres partenaires. Durant cette phase, nous avons choisi de configurer les ancres de sorte qu'elles bloquent l'utilisation du même *star interval* dans les îlots voisins. Une question intéressante serait l'étude de l'impact de cette décision sur la localisation à travers le réseau, en le comparant aux performances dans le cas où l'ancre choisirait de s'abstenir de participer lors de demandes simultanées. D'autre part, nous pourrions aussi comparer notre solution aux cas où la

négoiation est **mise en œuvre durant la CAP**. Le *beacon* étant le vecteur de dissémination privilégié au sein du réseau, une stratégie pourrait consister à minimiser sa surcharge.

Un objectif à moyen terme est l'**extension des capacités de DokoSim**. Si en l'état, DokoSim permet de tester les protocoles de mesure de distance exploités par cette thèse, une couche MAC complète serait la bienvenue car elle permettrait d'étudier en situation la localisation au sein du réseau. Cette implémentation, ainsi que de nombreuses autres constituent une partie des travaux à venir.

Finalement, l'**intégration de notre solution à des systèmes actuels** ne saurait être négligée. Au-delà des aspects passionnants de l'élaboration de protocoles et d'algorithmes, il est important pour toute proposition de localisation de trouver sa place dans le monde réel. Ce peut être au sein de projets variés et soumis à des contraintes diverses. Tout près d'ici, la maison intelligente de l'IUT de Blagnac constitue un environnement idéal pour tester l'exploitation de la connaissance de la position [87] par les modules d'adaptation au handicap. Toujours dans la Ville Rose, les matches de rugby pourraient bénéficier du suivi de la position des joueurs. L'étude et le monitoring des volcans feraient intervenir des capteurs robustes et dédiés à la collecte d'informations dans les coulées (terre, boue, roches). Finalement, dans des environnements à risque où les individus présents doivent pouvoir être évacués en urgence, par exemple une centrale nucléaire, découvrir facilement la position d'un rondier permet son guidage efficace vers les sorties de secours. Pour ces mêmes environnements, dans le cas d'une personne équipée d'un dosimètre, il peut être intéressant de localiser la zone dans laquelle un dépassement du niveau recommandé de radioactivité s'est produit.

GLOSSAIRE

Anéchoïque (salle)	Ne produisant pas de réflexions du signal grâce à un traitement des surfaces
AOA	<i>Angle Of Arrival</i>
BI	<i>Beacon Interval</i>
CAG	Contrôle Automatique de Gain
CAP	<i>Contention Access Period</i>
CFP	<i>Contention Free Period</i>
CSMA/CA	<i>Carrier Sense Multiple Access/ Collision Avoidance</i>
CSS	<i>Chirp Spread Spectrum</i>
DSSS	<i>Direct Sequence Spread Spectrum</i>
ESPRIT	<i>Estimation of Signal Parameters via Rotational Invariance Technique</i>
<i>Fingerprinting</i>	Reconnaissance d'empreinte
EHPAD	Établissement d'hébergement pour personnes âgées dépendantes
GSM	<i>Global System for Mobile communications</i>
GTS	<i>Guaranteed Time Slots</i>
Hotspot Wifi	Point d'accès Wifi publics, souvent utilisés pour la navigation internet
LOS	<i>Line Of Sight</i>
MAC	<i>Medium Access Control</i> , ou contrôle de l'accès au médium
Mouvement brownien	Mouvement aléatoire influencé par les chocs survenant au cours du déplacement rectiligne par ailleurs
MUSIC	<i>MUltiple SIgnal Classification</i>
Offset	Décalage appliqué à la valeur de l'horloge
PHY	Couche physique du modèle OSI
PSDS-TWR	<i>Parallel Symmetric Double Sided Two-Way Ranging</i>
QoS	<i>Quality of Service</i>
<i>Range-based</i>	Solution basée sur la collecte en temps réel de mesures de certaines caractéristiques du système
<i>Range-free</i>	Solution basée sur des hypothèses concernant certaines caractéristiques et propriétés du système
RFID	<i>Radio Frequency IDentification</i>
RSSI	<i>Received Signal Strength Indication</i>
<i>Self-healing</i>	Capacité du réseau à s'auto-réparer et à retourner à un fonctionnement normal après détection d'une faille
SPI	<i>Serial Peripheral Interface</i>
SSDS-TWR	<i>Sequential Symmetric Double Sided Two-Way Ranging</i>
TDOA	<i>Time Difference Of Arrival</i>
TOF	<i>Time Of Flight</i>
TWR	<i>Two Way Ranging</i>
<i>Transceiver</i>	Émetteur-récepteur de signal, implementation d'une couche physique spécifique
UWB	<i>Ultra Wide Band</i>
WLAN	<i>Wireless Local Area Network</i>

Glossaire

WPAN	<i>Wireless Personal Area Network</i>
WSN	<i>Wireless Sensor Network</i>

ANNEXE

Conception de la carte de développement nanoD

Alors que le JN5148 était livré avec une carte de développement, nous avons dû mettre en œuvre notre propre architecture afin de pouvoir interagir avec « *l'AVRmodule* » [26].

L'objectif premier de la carte de développement (figure 20) est de donner accès aux différentes interfaces du module communicant. Ces entrées/sorties peuvent être classées en 3 grands groupes : la communication entre le nœud radio et un ordinateur, l'accès aux ports de programmation et l'interface homme- machine

Communication avec l'ordinateur

L'interface de base pour la communication avec l'ordinateur reste le port série asynchrone du module. La stratégie usuelle consiste à inclure sur la carte de développement un composant tel que le ICL3223, capable d'adapter les niveaux de tension entre le port série d'un ordinateur et celui de *l'AVRmodule*. L'inconvénient majeur de cette configuration est la nécessité d'alimenter en permanence un bloc non vital. De plus, l'ordinateur cible doit disposer d'une interface série de type RS-232. Afin de nous affranchir de ces contraintes, nous avons décidé de déporter la fonctionnalité de l'ICL3223 sur un câble d'adaptation. Pour ceci nous avons utilisé le composant du FTDI Basic Breakout [39] : centré sur le FT232RLSSOP [38], il permet de ramener le flux de données sur un port USB, nettement plus répandu de nos jours sur les ordinateurs portables. Les broches RxD, TxD et GND du module seront acheminées à un connecteur sur lequel nous brancherons le module. La transformation effectuée, les données seront disponibles pour tout ordinateur disposant d'un port USB, via un port série virtuel COMx pour la partie logicielle.

Une adaptation (figure 92) a dû être réalisée car les niveaux de tensions fournis par le FTDI dépassent les limites fixées pour *l'AVRmodule*. En effet, la broche TxD du FTDI fournit un signal de 3.3V. Une diode est donc rajoutée, pour réduire la tension fournie au pin RXD de *l'AVRmodule*.

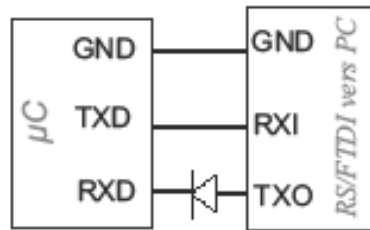


Figure 92 : Adaptation port RS-232

Programmation et débogage

Dans un contexte de conception de protocoles et d'algorithmes, il est indispensable d'avoir aisément accès aux interfaces de programmation du nœud. Notre carte de développement permet de se relier sur les deux ports de programmation supportés par l'AVRmodule, le SPI ainsi que le port *Joint Test Action Group* (JTAG).

Visualisation et interaction

Si le port série donne accès à une trace au format texte ASCII via un émulateur de terminal sur un ordinateur tiers, doter la carte de diodes électroluminescentes et de boutons poussoirs facilite une interprétation rapide du déroulement du programme sur le module. Avec ses 16 broches d'entrées/sorties, le module nanoLOC AVR est bien équipé pour supporter une riche combinaison de diodes et de boutons. Nous avons pourtant choisi de déporter notre bloc visualisation/interaction sur une carte d'extension. Cette décision part d'un double constat : premièrement, la visualisation n'est pas indispensable au fonctionnement. En réalité, elle représente une consommation supplémentaire pour des nœuds contraints énergétiquement. Une diode et un bouton seront implémentés directement sur la carte tandis que les 14 broches restantes seront routées vers deux ports d'extension. La possibilité de raccorder à travers ces ports des unités totalement différentes constitue la seconde motivation à implanter les interfaces d'extension.

En plus de valider les propositions de localisation basées sur le temps de vol, le prototype rejoindra les autres problématiques actuelles de l'équipe comme l'étude de la conservation de l'énergie [35].

En ce sens, nous avons ajouté un bloc de suivi de l'état de la source d'alimentation (figure 93). L'objectif est de récupérer la valeur de tension aux bornes de la résistance et, connaissant la différence de potentiel aux bornes de la diode, retrouver la tension délivrée par la source. Nous pourrons donc ainsi suivre en temps réel l'évolution de la tension de la batterie.

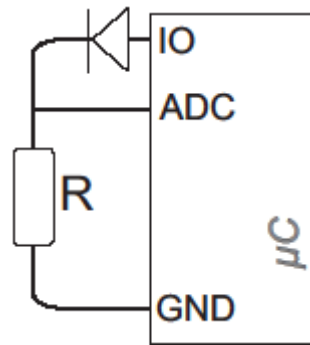


Figure 93 : Montage de suivi de l'alimentation

L'entrée VCC n'est pas directement utilisée pour cette fonction : nous avons préféré recourir à une broche d'entrées/sorties du microcontrôleur configurée en sortie. La possibilité de contrôler logiquement son état permettra de limiter les dépenses d'énergie liées à cette mesure.

La figure 94 correspond à la carte implémentée.

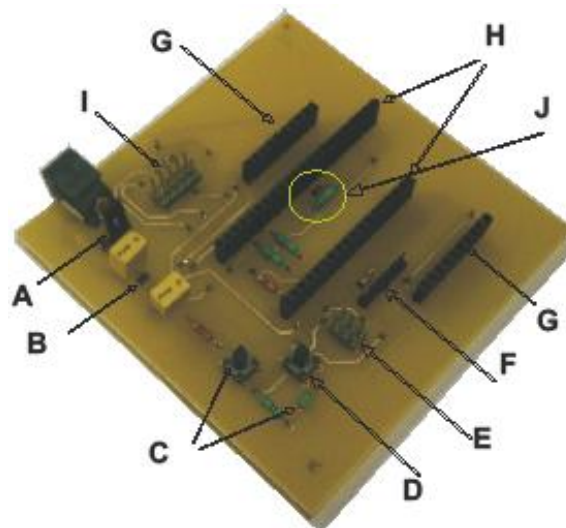


Figure 94 : Carte nanoM

On remarquera le sélecteur d'alimentation (A), le régulateur (B) et le circuit de mesure de la tension (J). Le module nanoLOC AVR se connecte par les embases H tandis que ses broches d'entrée/sortie, son port PWM et l'ADC libre sont accessibles par l'interface G. Les broches RXD et TXD du port série ont été ramenées en F ainsi que la masse dans la disposition correspondant au FTDI. La programmation par SPI se fait par le port E et l'accès JTAG par le port I. Finalement, en C se retrouvent deux E/S du module, l'un correspondant à un bouton (entrée) et le second à une LED (sortie).

LISTE DES FIGURES

Figure 1 : Structure d'un nœud capteur	17
Figure 2 : Structure temporelle dans un réseau 802.15.4 utilisant des <i>beacons</i>	19
Figure 3 : Propagation avec une réflexion	21
Figure 4 : Temps de vol.....	24
Figure 5 : <i>Time Of Arrival</i> et <i>Two Way Ranging</i>	31
Figure 6 : Le protocole <i>Symmetric Double-Sided Two-Way Ranging</i>	32
Figure 7 : Corruption de la mesure dans TWR	33
Figure 8 : Trilatération	36
Figure 9 : Angle d'incidence et position.....	36
Figure 10 : Multilatération	37
Figure 11 : Disposition des nœuds et mesures temporelles	45
Figure 12 : Principe topologique de DV-Hop	47
Figure 13 : <i>Checkout DV-Hop</i>	48
Figure 14 : Formulations géométrique et réseau de l'algorithme APIT	49
Figure 15 : Exemple de localisation avec <i>Centroid</i>	50
Figure 16 : Impact d'une solution centralisée sur l'ensemble du réseau.....	54
Figure 17 : Module radio-communicant Jennic	58
Figure 18 : Module communicant Nanotron	58
Figure 19 : <i>Upchirps</i> et <i>downchirps</i> dans la technologie CSS	59
Figure 20 : Module radio-communicant <i>Nanotron</i>	59
Figure 21 : Protocole de mesure de distance.....	61
Figure 22 : Mesures de RSSI en ligne droite (Jennic).....	61
Figure 23 : Mesures de RSSI en ligne droite (Nanotron).....	62
Figure 24 : Mesure de puissance reçue en ligne droite (Nanotron).....	62
Figure 25 : Mesure du RSSI durant un déplacement circulaire	63
Figure 26 : Variation du RSSI avec l'angle de rotation (Jennic).....	64
Figure 27 : Exemple de courbes de RSSI pour des directions différentes	64
Figure 28 : Caractérisation de la mesure de distance (signal DSSS, en extérieur)	66
Figure 29 : Accès aux registres	67
Figure 30 : Accès aux registres et instants d'émission/réception réels.....	68
Figure 31 : Contexte de test de la mesure de distance avec les nœuds <i>Nanotron</i>	69
Figure 32 : Mesures de distance par temps de vol à l'aide d'une plateforme CSS	70
Figure 33 : Mesures de distance par temps de vol (AVR Module).....	71
Figure 34 : Mesures de distance par temps de vol en salle anéchoïque (AVR Module).....	71
Figure 35 : Salle anéchoïque de l'IUT de Blagnac	72
Figure 36 : Procédure de traitement des données.....	73
Figure 37 : Evolution de la tension aux bornes de la charge durant la phase de transmission	77
Figure 38 : Mesure de la tension aux bornes de la charge durant la transmission	78
Figure 39 : Mesure de la tension aux bornes de la charge durant la réception	78
Figure 40 : Prédiction de la prochaine position basée sur l'orientation et le déplacement.....	82
Figure 41 : structure d'un système de localisation.....	86
Figure 42 :Utilisation des liens radios dans le <i>star interval</i>	88
Figure 43 : Intégration du <i>star interval</i> dans la structure temporelle du protocole.....	89
Figure 44 : Format général d'un beacon	89
Figure 45 : Ancre commune à deux îlots de localisation potentiels.....	91
Figure 46 : Exemple de topologie pour la négociation du <i>star interval</i>	92
Figure 47 : Négociation du <i>star interval</i>	94

Figure 48 : Format des différentes parties du message de localisation	95
Figure 49 : Exemples d'utilisation du message de localisation inclus dans les beacons	95
Figure 50 : Le protocole SSDS-TWR	96
Figure 51 : Liens radio et quadruplet de distances	96
Figure 52 : SSDS-TWR et PSDS-TWR	100
Figure 53 : Volumes d'information échangés par SSDS-TWR et PSDS-TWR	102
Figure 54 : Echanges avec une ancre durant SSDS-TWR	103
Figure 55 : Séquence des échanges durant l'exécution de PSDS-TWR.....	106
Figure 56 : Evolution de la consommation totale selon le nombre de d'ancres et le protocole	109
Figure 57 : Evolution de la consommation du mobile selon le nombre d'ancres et le protocole	110
Figure 58 : Evolution de la consommation des ancres selon le nombre d'ancres et le protocole	111
Figure 59 : Relation entre la matrice et les anneaux	113
Figure 60 : Différentes formes de l'intersection de deux anneaux	115
Figure 61 : Création de la matrice	116
Figure 62 : Initialisation des points de la matrice	116
Figure 63 : Changement de repère	117
Figure 64 : Exemple de sélection à partir des représentants	119
Figure 65 : Modèle de résultat réel de mesure de distances par SSDS-TWR	120
Figure 66 : Caractéristiques de la mesure de temps de vol par CSS en intérieur et en ligne de vue.	122
Figure 67 : Evolution de la moyenne des échantillons avec la distance	124
Figure 68 : Courbe de Gauss et régions	124
Figure 69 : Evolution de $n(d)$ et $m(d)$ avec la distance	125
Figure 70 : Comparaison des erreurs avec et sans ordonnancement des anneaux	126
Figure 71 : Environnement de collecte de données.....	131
Figure 72 : Mesures réelles et régression linéaire	132
Figure 73 : Première correction des valeurs brutes de distance	133
Figure 74 : Deuxième correction des estimations de distance	134
Figure 75 : Topologie de test.....	135
Figure 76 : Impact des collisions sur TWR.....	137
Figure 77 : Répartition des 27 positions de test	137
Figure 78 : Mise en évidence de l'effet de l'ordonnancement des anneaux.....	138
Figure 79 : Algorithme de déplacement d'un point dans la matrice	140
Figure 80 : Impact de la modification d' <i>iRingLA</i> sur le nombre d'opérations.....	142
Figure 81 : Impact du délai entre les réponses	145
Figure 82 : Architecture de classes	149
Figure 83 : Topologie d'exemple pour les zones de portée	156
Figure 84 : Exemple de pertes de trames au cours de TWR	158
Figure 85 : Effet de la variation du <i>multipathFactor</i> sur les résultats	163
Figure 86 : Evolution de l'erreur de localisation par ordonnancement des anneaux	165
Figure 87 : Paramètres temporels de PSDS-TWR	166
Figure 88 : Répartition de l'erreur pour PSDS-TWR	169
Figure 89 : Cas particuliers observés pendant la simulation de PSDS-TWR	169
Figure 90 : Répartition des erreurs pour PSDS-TWR avec intégration des pertes de trames	171
Figure 91 : Durée d'exécution selon le protocole.....	172
Figure 92 : Adaptation port RS-232	184
Figure 93 : Montage de suivi de l'alimentation	185

Figure 94 : Carte nanoM 185

LISTE DES TABLEAUX

Tableau 1 : Comparatif des standards IEEE 802.11, 802.15.1 et 802.15.4	18
Tableau 2 : Variables du Filtre de Kalman.....	38
Tableau 3 : Comparaison des performances de diverses informations pertinentes pour la localisation	52
Tableau 4 : Caractéristiques des deux plateformes matérielles disponibles.....	60
Tableau 5 : Etats du <i>transceiver</i> et consommation.....	76
Tableau 6 : Paramètres de la consommation durant la période active de l'émetteur-récepteur	79
Tableau 7 : Courants traversant l'équipement durant la phase active de l'émetteur-récepteur	79
Tableau 8 : Codage du protocole de mesure et des réponses des ancres	90
Tableau 9 : Codage des réponses à une demande de localisation	94
Tableau 10 : Structure de la payload dans SSDS-TWR	97
Tableau 11 : Durées des différentes étapes pour l'exécution de SSDS-TWR	104
Tableau 12 : Répartition du temps entre émission et réception pour SSDS-TWR.....	105
Tableau 13 : Répartition du temps entre émission et réception pour PSDS-TWR.....	107
Tableau 14 : Résumé des durées pour les protocoles SSDS-TWR et PSDS-TWR.....	108
Tableau 15: critères d'identification du type d'intersection	114
Tableau 16 : Nombre d'opérations pour chacune des phases de l'exécution d' <i>iRingLA</i>	140
Tableau 17 : Matrice de connectivité construite par le médium.....	155
Tableau 18 : Comparaison des erreurs de localisation en fonction du <i>multipathFactor</i>	163
Tableau 19 : Configuration du simulateur pour SSDS-TWR.....	164
Tableau 20 : Résultats de simulation pour SSDS-TWR avec interférences.....	164
Tableau 21 : Paramétrage temporel.....	167
Tableau 22 : Résultats	168
Tableau 23: Résultats de simulation pour PSDS-TWR avec ajout d'interférences	170
Tableau 24 : Paramètres temporels pour la comparaison de PSDS-TWR et SSDS-TWR.....	172

PUBLICATIONS PERSONNELLES

Articles de revues internationales

Rejane Dalce, Thierry Val, Adrien Van den Bossche. *Comparison of Indoor Localization Systems based on Wireless Communications*. Dans : *Wireless Engineering and Technology, Scientific Research Publishing*, Irvine - USA, Vol. 2 No 4, octobre 2011.

Conférences et workshops internationaux

Rejane Dalce, Adrien Van den Bossche, Thierry Val. *Optimization of a CSMA/CA based MAC protocol designed for confined WSNs (regular paper)*. Dans : *IEEE International Conference on Wireless Communications in Unusual and Confined Areas (ICWCUCA 2012)*, Clermont-Ferrand, France, 28/08/2012-30/08/2012, IEEEExplore digital library, 2012.

Rejane Dalce, Adrien Van den Bossche, Thierry Val. *Towards a new range-based localization method for WSNs: Challenges, Constraints and Correction (regular paper)*. Dans : *IEEE International Conference on Wireless Communications in Unusual and Confined Areas (ICWCUCA 2012)*, Clermont-Ferrand, France, 28/08/2012-30/08/2012, IEEEExplore digital library, 2012 (Best Paper).

Rejane Dalce, Adrien Van den Bossche, Thierry Val. *Une plateforme de prototypage évolutive pour la localisation dans un réseau IEEE 802.15.4a CSS (regular paper)*. Dans : *Conférence Internationale Francophone d'Automatique (CIFA 2012)*, Grenoble, 04/07/2012-06/07/2012, IEEE, juillet 2012.

Adrien Van den Bossche, Thierry Val, Rejane Dalce. *SISP: a lightweight Synchronization Protocol for Wireless Sensor Networks (short paper)*. Dans : *Emerging Technologies and Factory Automation (ETFA 2011)*, Toulouse, 05/09/2011-09/09/2011, *IEEE Computer Society - Conference Publishing Services*, 2011.

Rejane Dalce, Linqing Gui, Thierry Val, Adrien Van den Bossche, Anne Wei. *Localisation par méthodes « range-based » et « range-free » de stations mobiles communicantes dans un réseau sans fil (poster)*. Dans : *Colloque Francophone sur l'Ingénierie des Protocoles (CFIP 2011)*, Sainte-Maxime, France, 10/05/2011-13/05/2011, CFIP 2011, mai 2011.

Soumissions à des conférences internationales/ journaux

« *Indoor Self-Localization in a WSN, based on Time Of Flight: Propositions and Demonstrator* » soumis à IPIN 2013

« *A complete decentralized range-based localization method for wireless sensor networks* » soumis à *ACM Transactions on Sensor Networks*

« *A new MAC-level negotiation protocol for indoor TOF-based localization of mobile nodes* » soumis à *International Journal of Sensor Networks*

Conférences sans actes publiés

Rejane Dalce, Thierry Val, Adrien Van den Bossche. *Méthodes et systèmes de localisation combinant plusieurs technologies de transmission et de positionnement*. Dans : *Congrès des doctorants EDSYS*, Toulouse, 10/05/2011-11/05/2011 (conférencier invité).

BIBLIOGRAPHIE

1. K. Alagha, G. Chalhoub, A. Guitton, E. Livolant, S. Mahfoudh, P. Minet, M. Misson, J. Rahme, T. Val, A. van den Bossche, "Cross-layering in an industrial wireless sensor network: case study of OCARI" Journal of Networks : JNW Vol. 4 - issue 6, Collection special issue: Wireless Sensor Networks: Theory and Practice, Academy Publisher ISBN 1796-2056 (Août 2009)
2. H. S. AbdelSalam, S. Olariu, "HexNet: Hexagon-Based Localization Technique for wireless Sensor Networks", IEEE International Conference on Pervasive Computing, mars 2009
3. A.M. Abu-Mahfouz, G.P. Hancke, "NS-2 extension to simulate localization system in wireless sensor networks", AFRICON 2011
4. V. Adamchuk, "Satellite-based Auto-Guidance", Institute of Agriculture and Natural Resources, 2008
5. Amundson I., Koutsoukos X. D., "A Survey on Localization for Mobile Wireless Sensor Networks", International Workshop on Mobile Entity Localization (MELT), septembre 2009
6. http://aoa.gov/AoARoot/Aging_Statistics/future_growth/docs/By_Age_Total_Population.xls
7. M. S. Arulampalam, S. Maskell, N. Gordon, T. Clapp, 'A tutorial on Particle Filters for Online Nonlinear/Non-Gaussian, Bayesian Tracking', IEEE transactions on Signal Processing vol 50, No 2, février 2002
8. ATMega 644V <http://www.atmel.com/Images/doc2593.pdf> (Accès octobre 2012)
9. Nanotron AVR Module http://www.nanotron.com/EN/pdf/Factsheet_nanoLOC-AVR-Module.pdf (Accès décembre 2011)
10. R. Badorrey, Á. Hernández, J. Chóliz, A. Valdovinos, I. Alastruey, "Evaluation of TOA estimation algorithms in UWB receivers", European Wireless Conference, juin 2008
11. P. Bahl, V. N. Padmanabhan, "RADAR : An In-Building RF-Based User Location and Tracking System" IEEE Infocom, mars 2000
12. P. Barsocchi, F. Furfari, P. Nepa, F. Potorti, "RSSI localisation with sensors placed on the user", Indoor Positioning and Indoor Navigation (IPIN), 2010, septembre 2010
13. Bluetooth Low Energy https://www.bluetooth.org/docman/handlers/downloaddoc.ashx?doc_id=229737
14. A. Van den Bossche, « Proposition d'une nouvelle méthode d'accès déterministe pour un réseau personnel sans fil à fortes contraintes temporelles », Université de Toulouse Le Mirail, juillet 2007
15. <http://patft.uspto.gov/netacgi/nph-Parser?Sect1=PTO1&Sect2=HITOFF&d=PALL&p=1&u=%2Fnethtml%2FPTO%2Fsrchnum.htm&r=1&f=G&l=50&s1=6774787.PN.&OS=PN/6774787&RS=PN/6774787>
16. N. Bulusu, J. Heidemann, D. Estrin, "GPS-Less Low-Cost Outdoor Localization for Very Small Devices" IEEE Personal Communications, octobre 2000

17. E. Cassano, F. Florio, F. de Rango, S. Marano, "A Performance Comparison between ROC-RSSI and Trilateration Localization Techniques for WPAN Sensor Networks in Real Outdoor Testbed", Wireless Telecommunications Symposium, juin 2009
18. G. Chelius, A. Fraboulet, E. Fleury, "WorldSens: a Fast and Accurate Development Framework for Sensor Network Applications", SAC'07, mars 2007
19. Guide Pratique du GPS (Paul Correia) Editions Eyrolles(5e édition)
20. SIAM "Iterative Methods for Optimization", CT Kelley, North Carolina State University, Raleigh, North Carolina, Society for Industrial and Applied Mathematics,1999 [online] http://www.siam.org/books/textbooks/fr18_book.pdf (Accès août 2012)
21. M. Cypriani, Géopositionnement Wi-Fi autocalibré en milieu hétérogène, Université de Franche-Comté, décembre 2012
22. M. Cypriani, P. Canalda, F. Spies, "OwlPS: A Self-calibrated Fingerprint-Based Wi-Fi Positioning System", in "Evaluating AAL Systems Through Competitive Benchmarking. Indoor Localization and Tracking", pp 36-51, Springer Berlin Heidelberg 2012, DOI 10.1007/978-3-642-33533-4_4
23. V. Daiya, Jemimah Ebenezer, S.A.V. Satya Murty, Baldev Raj, Experimental Analysis of RSSI for Distance and Position Estimation, IEEE-International Conference on Recent Trends in Information Technology, ICRTIT 2011 MIT, Anna University, Chennai, juin 2011
24. Rejane Dalce, Linqing Gui, Thierry Val, Adrien Van den Bossche, Anne Wei. Localisation par méthodes « range-based » et « range-free » de stations mobiles communicantes dans un réseau sans fil (poster). Dans : Colloque Francophone sur l'Ingénierie des Protocoles (CFIP 2011), Sainte-Maxime, France, 10/05/2011-13/05/2011, CFIP 2011, (en ligne), mai 2011.
25. Rejane Dalce, Thierry Val, Adrien Van den Bossche. Comparison of Indoor Localization Systems based on Wireless Communications. Dans : Wireless Engineering and Technology, Scientific Research Publishing, Irvine - USA, Vol. 2 No 4, (en ligne), octobre 2011.
26. Rejane Dalce, Adrien Van den Bossche, Thierry Val. Une plateforme de prototypage évolutive pour la localisation dans un réseau IEEE 802.15.4a CSS (regular paper). Dans : Conférence Internationale Francophone d'Automatique (CIFA 2012), Grenoble, 04/07/2012-06/07/2012, IEEE, (support électronique), juillet 2012.
27. Rejane Dalce, Adrien Van den Bossche, Thierry Val. Optimization of a CSMA/CA based MAC protocol designed for confined WSNs (regular paper). Dans : IEEE International Conference on Wireless Communications in Unusual and Confined Areas (ICWCUCA 2012), Clermont-Ferrand, France, août 2012
28. Rejane Dalce, Adrien Van den Bossche, Thierry Val. Towards a new range-based localization method for WSNs: Challenges, Constraints and Correction. IEEE International Conference on Wireless Communications in Unusual and Confined Areas (ICWCUCA 2012), Clermont-Ferrand, France, août 2012

29. Real Time Location Systems. Retrieved August 20, 2012 from http://www.decawave.com/assets/files/products/RTLS_110307.pdf
30. Doherty L., Pister K. S. J., El Ghaoui L., "Convex Position Estimation in Wireless Sensor Networks", IEEE Infocom, avril 2001
31. Dong F., Chen Y., Li K., Guo D., "A Calibration-Free Localization Solution for Handling Signal Strength Variance", International Workshop on Mobile Entity localization MELT, septembre 2009
32. F. Evennou, "Techniques et technologies de localisation avancées pour terminaux mobiles dans les environnements indoor", Ph.D. Dissertation, Université Joseph Fourier, Grenoble, 2007
33. [ezFind] <http://ez-find.com/ez-find/>
34. W. Fahs, "Diffusion d'Informations Partagees Entre Mobiles Cooperants Evoluant Sous une Meme Cellule d'un Reseau Sans fil Avec Infrastructure," Ph.D. Dissertation, Université de Clermont-Ferrand, 2009
35. N. Fourty, A. van den Bossche, T. Val, Etude de l'impact énergétique de l'algorithme d'accès au médium pour un réseau de capteurs sans fil industriel, Sixième Conférence Internationale Francophone d'Automatique, Nancy, France, juin 2010
36. Embedded wireless system for pedestrian localization in indoor environments, Sensors & Transducers Journal, Vol.14-2, pp.211-227, février 2012, Special Issue
37. G. Fritz, Simulation de fautes pour l'évaluation du test en ligne de systèmes RFID, Université de Grenoble, décembre 2012
38. FT232RL http://www.ftdichip.com/Support/Documents/DataSheets/ICs/DS_FT232R.pdf (Accès janvier 2012)
39. Ftdi <https://www.sparkfun.com/products/9716?> (Accès janvier 2013)
40. R. Fuller. Tutorial on location determination by RF means. In Mobile entity Localization and Tracking in GPS-less Environments, 2009.
41. Geotonome <http://www.geotonome.fr/> (Accès octobre 2012)
42. M. Ghavami, L. B. Michael, R. Kohno, "Ultra wideband signals and systems in communication engineering" Wiley Editions 2007
43. GoogleLocalisation <http://support.google.com/gmm/bin/answer.py?hl=en&answer=1685827> (Accès octobre 2012)
44. M. S. Grewal and A. P. Andrews, KalmanFiltering, Wiley Editions 2008
45. L. Gui, A. Wei, T. Val, "A Two-level Range-free Localization Algorithm for Wireless Sensor Networks", WiCom, septembre 2010
46. L. Gui, Amélioration de la Localisation dans les Réseaux de Capteurs sans Fil par Méthodes "Range-free". Thèse de doctorat, INSA Toulouse - EDSYS - Laboratoires LATTIS puis IRIT, février 2013.

Bibliographie

47. Y. Gwon, R. Jain, T. Kawahara, "Robust Indoor Location Estimation of Stationary and Mobile Users", Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies, mars 2004
48. S. Harchi, J.-P. Georges, T. Divoux, "WSN dynamic clustering for oil slicks monitoring" 3rd International Conference on Wireless Communications in Unusual and Confined Areas, ICWCUCA'12 - 3rd International Conference on Wireless Communications in Unusual and Confined Areas, ICWCUCA'12, France (2012) [hal-00751839 - version 1]
49. T. He, C. Huang, B. M. Blum, J. A. Stankovic, T. Abdelhazer "Range-Free Localization Schemes for Large Scale Sensor Networks" MobiCom, septembre 2003
50. Yu-Kai Huang, Ai-Chun Pang, Hui-Nien "An Adaptive GTS Allocation Scheme for IEEE 802.15.4" IEEE Transactions on parallel and distributed systems, vol. 19, no 5, mai 2008
51. INsee http://www.insee.fr/fr/themes/document.asp?ref_id=ip1089 (Accès octobre 2012)
52. Specific requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications 2012, IEEE Computer Society
53. Specific requirements Part 15.1: Wireless medium access control (MAC) and physical layer (PHY) specifications for wireless personal area networks (WPANs), IEEE Computer Society, juin 2005
54. Specific requirements Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs) Amendment 1: Add Alternate PHYs, IEEE Standard for Information technology Telecommunications and information exchange between systems, août 2007
55. F. Izquierdo, M. Ciurana, F. Barcelo, J. Paradells, E. Zola, (2006) 'Performance evaluation of a TOA-based trilateration method to locate terminals in WLAN'. Paper presented at the 1st International Symposium on Wireless Pervasive Computing. 16-18 Janvier 2006
56. F. V. Jensen, Introduction to Bayesian Networks, Springer 1997
57. Jeong A.C., Hanbyeul Na, Sunwoo Kim, Chunsoo Ahn (2012), "Moving-Target Tracking Based on Particle Filter with TDOA/FDOA Measurements", ETRI Journal, Vol 32, No 2, avril 2012
58. Wei-Wei Ji, Zhong Liu "An improvement of DV-HOP algorithm in Wireless Sensor Networks", International conference on Wireless Communications, Networking and Mobile Computing, septembre 2006
59. JN5148 Jennic. <http://www.jennic.com/files/productbriefs/jn-ds-jn5148-1v2.pdf>. (Accès octobre 2010)
60. Jongdae J., Hyun M., "Indoor Localization Using Particle Filter and Map-based NLOS Ranging Model", IEEE International Conference on Robotics and Automation (ICRA), 2011
61. J. M. Kahn, J. R. Barry, Wireless Infrared Communications, Proceedings of the IEEE, vol. 85, no. 2, février 1997

62. Koubaa, M. Alves, E. Tovar, "i-GAME: An Implicit GTS Allocation Mechanism in IEEE 802.15.4 for Time-Sensitive Wireless Sensor Networks" 18th Euromicro Conference on Real-Time Systems (ECRTS 06), Dresden, Germany, juillet 2006
63. A. Koubaa, M. Alves, M. Attia, A. Van Nieuwenhuysse, 'Collision-Free Beacon Scheduling Mechanisms for IEEE 802.15.4/Zigbee Cluster-Tree Wireless Sensor Networks'. Paper presented at the 7th International Workshop on Applications and services in Wireless Networks. mai 2007
64. Kwang Hyun Lee, Sung Ho Cho, CSS based localization system using Kalman Filter for Multi-Cell Environment, Adanced Technologies for Communications 2008
65. S. Lanzisera, D.T. Lin, K. S. Pister, "RF Time of Flight Ranging for Wireless Sensor Network Localization", Workshop on Intelligent Solutions in Embedded Systems (WISES), june 2006, pp. 1-12
66. D. Lieckfeldt, J. You, D. Timmermann, "Characterizing the influence of human presence on bistatic passive rfid-system" Wireless and Mobile Computing, Networking and Communications, november 2009, pp. 338 - 343
67. H. Lin, Y. Zhang, M. Griss, I. Landa "WASP: An Enhanced Indoor Locationing Algorithm for a Congested Wi-Fi Environment" Second International Workshop, MELT 2009
68. H. Liu, H. Darabi, P. Banerjee, "Survey of Wireless Indoor Positioning Techniques and Systems", IEEE Transactions on Systems, Man and Cybernetics_ Part C: Applications and Reviews, Vol 37, no 6, november 2007
69. E. Livolant, A. van den Bossche, T. Val, "MAC specifications for a WPAN allowing both energy saving and guaranteed delay - Part B: Optimization of the intra-star exchanges for MaCARI", IFIP Conference on Wireless Sensor and Actor Networks (WSAN 2008) Ottawa, Canada (July 2008)
70. E. Llusca, T. Val, C. Normand, J. Mercier, "Location of mobiles stations in a wireless LAN", Local Computer Networks (LCN 2000) Tampa, Etats-Unis (Novembre 2000)
71. J. Lu, Modeling, simulation and implementation of an 802.15.4 based adaptive communication protocol in wireless sensor network: application to monitoring the elderly at home, Université de Toulouse, février 2013
72. C. Mailhes, B.Comet, H. de Bernard, E. Campo, A. Prieto, S. Bonhomme, Telemedicine applications in OURSES project, IEEE International Workshop onSatellite and Space Communications, octobre 2008
73. Multilatération <http://en.wikipedia.org/wiki/multilateration> (Accès juin 2011)
74. NA5tr1 [http://www.gaw.ru/pdf/Nanotron/nanoLOC-NA5TR1-RegDesc Prelim_V101_2006-12-21.pdf](http://www.gaw.ru/pdf/Nanotron/nanoLOC-NA5TR1-RegDesc_Prelim_V101_2006-12-21.pdf) (Accès janvier 2011)
75. D. Niculescu and B. Nath, "Position and Orientation in Ad Hoc Networks" Elsevier Ad Hoc Network, vol. 2, no. 2, Apr. 2004
76. NS-2 <http://www.isi.edu/nsnam/ns/> (Accès janvier 2013)
77. NS-3 <http://www.nsnam.org/docs/release/3.16/manual/ns-3-manual.pdf> (Accès janvier 2013)

78. OMNET++ <http://www.omnetpp.org/doc/omnetpp/manual/usman.html> (Accès janvier 2013)
79. H. Ouibrahim, D.D. Weiner, T.K. Sarkar, A generalized approach to direction finding 1986
80. S. Park, H. Ahn, and W. Yu, "A Simple Object Tracking System Using ITDOA without Time Synchronization", ICACT 2007
81. T. S. Rappaport, "Wireless communications, principles and practice", Prentice Hall, 1996.
82. RIP protocol <http://tools.ietf.org/html/rfc2453> (Accès décembre 2012)
83. J. Rousselot, M. Aoun, R. Serna Oliver, « Accurate Timeliness Simulations for Real-Time Wireless Sensor Networks », Computer Modeling and Simulation 2009
84. S. Schwarzer, M. Vossiek, M. Pichler, A. Stelzer, "Precise Distance Measurement with IEEE 802.15.4 (ZigBee) Devices" IEEE Radio and Wireless Symposium, march 2008, pp. 779 - 782
85. STM32F205ZG <http://www.st.com/st-web-ui/static/active/en/resource/technical/document/datasheet/CD00237391.pdf> (Accès janvier 2013)
86. STM32F427II http://www.st.com/st-web-ui/static/active/en/resource/technical/document/data_brief/DM00053484.pdf (Accès janvier 2013)
87. T. VAL, E. BOUGEOIS, A. VAN DEN BOSSCHE, N. CAZENAVE, L. REDON, A. SOVEJA, T. VILLEMUR, « Projet CANet : un système de suivi de personnes à mobilité réduite grâce à leur canne de marche », revue EsprIUT, Numéro spécial Recherche, Vol. Hors-série, p. 10-11, février 2013
88. Weyn, M. Klepal, Didyawan, "Adaptive Motion Model for a Smart Phone Based Opportunistic Localization System", MELT 2009
89. Y. Xue, H. S. Lee, M. Yang, P. Kumarawadu, H. H. Ghenniwa, W. Shen, "Performance evaluation of NS-2 Simulator for Wireless Sensor Networks", Electrical and Computer Engineering, 2007.
90. G.Z. Yang, "Body Sensor Networks", Springer-Verlag London, 2006
91. L. Zimmermann, A. Goetz, G. Fischer, R. Weigel, GSM Mobile Phone Localization using Time Difference of Arrival and Angle of Arrival Estimation, Multiconference on Systems, Signals and Devices, 2012

SUJET DE THESE

Méthodes et systèmes de localisation combinant plusieurs technologies de transmission et de positionnement

Contexte scientifique : Ingénierie des réseaux sans fil pour des applications de géo-localisation

Mots clefs : sans fil, géo-localisation, positionnement, protocoles, UWB, GPS, contraintes temporelles critiques, modélisation, caractérisation, prototypage

Description :

L'objectif de cette thèse est de contribuer à la recherche de méthodes de combinaison de données de localisation dans un contexte d'ingénierie des réseaux sans fil. En effet, il n'est pas rare que, de nos jours, un équipement terminal de réseau mobile dispose d'un nombre important d'informations interprétables dans un objectif de géo-localisation : en plus du signal du réseau lui-même qui porte des informations pertinentes (force du signal radio, désynchronisation sur la couche physique, adresse de la station de base, etc.) il n'est pas rare que le terminal dispose également d'un récepteur GPS, d'un accès vers un ou deux autres WLAN/WPAN (802.11, bluetooth et demain 802.15.4/ZigBee ou UWB), etc. L'équipement peut également comporter des composants spécifiques tels que des accéléromètres, sources potentielles de données pertinentes à localisation du terminal.

Dans ce contexte, il paraît intéressant d'imaginer un traitement global de l'ensemble de ces informations qui sont de toutes manières présentes dans le terminal. La combinaison de l'ensemble de ces données devrait soit permettre une précision inédite (l'exemple du D-GPS en est une preuve), soit un fonctionnement optimal dans des circonstances particulières (exemple : à l'intérieur d'un bâtiment où des techniques de positionnement par satellites sont inefficaces).

Méthodes de localisation par le signal de communication dans les réseaux de capteurs sans fil en intérieur

Depuis quelques années, la thématique de la localisation a connu un regain d'intérêt, motivé en grande partie par le développement des réseaux de capteurs sans fil. Les propositions ayant pour objectif d'apporter une réponse à cette problématique peuvent être classées en deux catégories : les méthodes *range-based*, retenues pour cette thèse, dont la caractéristique est de se fonder sur des mesures en temps réel du signal pour générer une estimation relativement fiable de la distance, et les solutions *range-free*, économes en ressources car se limitant à l'exploitation d'hypothèses concernant la connectivité des nœuds dans le réseau. Les contributions peuvent se focaliser sur l'un des trois aspects fondamentaux de la question : le développement d'une couche physique performante, la proposition d'un algorithme de calcul permettant des résultats plus précis, et la mise en place d'un protocole de collecte de mesures.

La contribution de cette thèse est par conséquent multiple : en premier lieu, nous proposons un nouveau protocole de mesure du temps de vol, nommé *Parallel Symmetric Double-Sided Two-Way Ranging* (PSDS-TWR), dont l'objectif est la réduction de la charge protocolaire du service. Deuxièmement, nous avons mis en place un algorithme de calcul de la position désigné par *interRing Localization Algorithm* (*iRingLA*). Basé sur une recherche linéaire, *iRingLA* accepte des données inexacts et en extrait une estimation de la position dont l'erreur est inférieure à 2m dans 70% des cas, ceci en s'exécutant directement sur un nœud mobile léger. Les données concernant l'algorithme ont été collectées grâce à un prototype utilisant la technologie *Chirp Spread Spectrum* tandis que l'étude de performance du protocole a impliqué la conception d'un simulateur nommé DokoSim.

Mots clés : localisation *range-based*, algorithme décentralisé, *Time of Flight*, réseaux de capteurs sans fil, couche physique radio, prototype, simulation, mesures, IEEE 802.15.4a

Localization methods using the communication signal in indoor wireless sensor networks

The development of Wireless Sensor Networks has given a new life to research in the field of localization. Many proposals have been made which can be classified as either range-free or range-based solutions. The range-free category relies on a priori knowledge of the network while the latter uses the available hardware to measure signal characteristics from which distance information can be derived. Although the information origin can vary, all proposals either introduce a new protocol, a novel algorithm or a new and improved physical layer.

Our work led to the definition of a new protocol and an efficient algorithm. Aside from allowing the nodes to collect Time Of Flight related data, the Parallel Symmetric Double-Sided Two-Way Ranging protocol (PSDS-TWR) reduces overhead and energy consumption, making the localization service affordable for the network. The performance of this protocol, in terms of duration, has been studied using a homemade simulator named DokoSim. We also introduce an algorithm based on rings and linear search. This inter-Ring Localization Algorithm (*iRingLA*) achieves a localization error of less than 2m in 70% of the cases while being tested on our Chirp Spread Spectrum based prototype.

Keywords: decentralized algorithm, range-based localization, Time of Flight, Wireless Sensor Network, physical layer, prototype, simulation, measurements, IEEE 802.15.4a