

TABLE DES MATIÈRES

	Page
INTRODUCTION	23
CHAPITRE 1 MISE EN CONTEXTE.....	25
1.1 Introduction.....	25
1.2 Objectifs du projet	27
1.3 Méthodologie.....	28
CHAPITRE 2 PRINCIPES DE LOCALISATION D'INTÉRIEUR.....	31
2.1 Introduction.....	31
2.2 État de l'art des systèmes de localisation d'intérieur.....	31
2.2.1 Systèmes basés sur le Wifi ou IEEE 802.11	34
2.2.2 Systèmes de localisation de type Bluetooth.....	35
2.2.3 Systèmes d'intérieur de type GPS.....	35
2.2.4 Localisation dans un réseau de capteurs sans fils ou WSN	36
2.2.5 Systèmes de localisation de type RFID	36
2.2.6 Autres systèmes	36
2.3 Mesures de distances	37
2.3.1 Techniques de triangulation.....	37
2.3.1.1 L'instant d'arrivée ou ToA	38
2.3.1.2 La différence des instants d'arrivées ou TDoA	40
2.3.1.3 Le temps d'aller-retour du signal ou RToF.....	42
2.3.1.4 La puissance du signal reçu ou RSS	42
2.3.2 L'angle d'arrivé	44
2.3.3 La phase du signal reçu ou PoA.....	45
2.3.4 Analyse de scène.....	46
2.4 Algorithmes de localisation	46
2.4.1 Algorithmes de localisation basés sur les RSS	47
2.4.1.1 Le voisin le plus proche ou <i>Closest Neighbor</i> (CN).....	49
2.4.1.2 L'estimation basée sur le maximum de vraisemblance	49
2.4.2 Algorithmes de localisation basés sur le ToA.....	50
2.4.2.1 Moindres carrées pour le ToA	50
2.5 Métriques de performances.....	51
2.5.1 Précision.....	51
2.5.1.1 Erreur circulaire probable	52
2.5.1.2 Dilution géométrique de précision.....	53
2.5.1.3 Erreur quadratique moyenne.....	54
2.5.1.4 Borne inférieure de Cramer-Rao.....	54
2.5.2 Exactitude	54

2.5.3	Robustesse.....	55
2.5.4	Complexité.....	55
2.5.5	Évolutivité.....	56
2.5.6	Coût.....	57
2.6	Sources d'erreurs.....	57
2.6.1	Les erreurs induites par la propagation.....	58
2.6.1.1	Rayons des trajets multiples.....	58
2.6.1.2	Phénomène d'obscuration ou <i>shadowing</i>	59
2.6.2	Les erreurs induites par les instruments et l'observation.....	59
2.6.2.1	Délai matériel.....	60
2.6.2.2	Limitations de la bande passante.....	60
2.7	Conclusion.....	62
CHAPITRE 3 LES RADIOS DÉFINIES PAR LOGICIEL.....		63
3.1	Introduction.....	63
3.2	Les radios définies par logiciels.....	63
3.3	Architecture matérielle.....	66
3.3.1	Carte mère USRP N210.....	67
3.3.2	Carte fille.....	69
3.3.3	Antennes.....	71
3.4	Architecture logicielle.....	72
3.4.1	Framework GNU-Radio.....	72
3.4.2	Pilotes des radios, ordonnancement.....	76
3.4.2.1	Les pilotes UHD.....	76
3.4.2.2	L'ordonnanceur.....	78
3.4.2.3	Architecture de l'ordonnanceur TPB.....	79
3.5	Conclusion.....	81
CHAPITRE 4 MISE À JOUR DU SYSTÈME DE LOCALISATION SUR GNU-RADIO.....		83
4.1	Introduction.....	83
4.2	Réseau local sans fil IEEE 802.11b.....	83
4.2.1	Implémentations de 802.11 sur radio définie par logiciel.....	85
4.2.1.1	BBN 802.11b.....	85
4.2.1.2	Récepteur 802.11b du laboratoire SPAN.....	85
4.2.1.3	Encodeur et émetteur 802.11p de FTW.....	86
4.2.2	Contributions au projet <i>BBN 802.11b</i>	86
4.2.3	Émetteur-récepteur OFDM pour IEEE 802.11 a/g/p.....	87
4.2.4	Projet <i>WifiLocalization</i>	87
4.3	Mise à jour des codes sources de la chaîne de communication.....	89
4.3.1	Mise à jour pour se conformer aux pilotes UHD.....	89

4.3.2	Codes émetteur et récepteurs « <i>neoTx.py</i> » et « <i>neoRx.py</i> »	90
4.4	Tests préliminaires et résultats obtenus	93
4.4.1	Tests préliminaires	93
4.4.2	Résultats obtenus	97
4.5	Conclusion	103
CHAPITRE 5 AMÉLIORATIONS DU SYSTÈME DE LOCALISATION		105
5.1	Introduction.....	105
5.2	Étude des éléments de la latence	105
5.3	Interprétation des résultats rapportés par le module de post-traitement	109
5.4	Améliorations apportées	112
5.4.1	Amélioration du module d'estimation des RSS.....	112
5.4.2	Mise à jour pour la compatibilité avec l'API C++ d'UHD.....	115
5.4.3	Conception de la solution d'horodatage au sein du FPGA	117
5.4.4	Preuve de concept d'un mécanisme de synchronisation par PTPd.....	122
5.5	Conclusion	123
CONCLUSION.....		125
RECOMMANDATIONS		127
ANNEXE I	NOUVEAU CODE SOURCE ÉMETTEUR « <i>neoTx.py</i> ».....	129
ANNEXE II	NOUVEAU CODE SOURCE RÉCEPTEUR « <i>neoRx.py</i> ».....	133
ANNEXE III	RÉSULTATS D'HORODATAGE	137
ANNEXE VI	EXEMPLE TYPE DE CALIBRATION	139
ANNEXE V	RÉSOLUTION MATHÉMATIQUE DU PROBLÈME DE LOCALISATION	141
LISTE DE RÉFÉRENCES BIBLIOGRAPHIQUES.....		143

LISTE DES TABLEAUX

	Page
Tableau 2.1	Topologies des systèmes locaux de positionnement sans fils [5]32
Tableau 3.1	Comparatif des caractéristiques des USRP d’Ettus67
Tableau 3.2	Comparatif des cartes filles utilisées70
Tableau 3.3	Caractéristiques électriques de l'antenne utilisée [77]71
Tableau 5.1	Spécificités de l'hôte et des outils de développement du projet.....108
Tableau 5.2	Champs des résultats collectés pour le post-traitement108
Tableau 5.3	Description des tables de la base de données de post-traitement110
Tableau 5.4	Description des requêtes de la base de données de post-traitement111
Tableau 5.5	Ressources libres sur le FPGA de N210.....122

LISTE DES FIGURES

		Page
Figure 1.1	Diagramme bloc des systèmes de géolocalisation sans fil [2].....	26
Figure 1.2	Système de localisation basé sur les radios USRP N210.....	27
Figure 2.1	Aperçu des systèmes actuels basés sur le positionnement sans fils [9]..	33
Figure 2.2	Localisation en 2-D avec ToA, cas idéal	38
Figure 2.3	Localisation en 2-D avec ToA, tenant	39
Figure 2.4	Localisation basée sur les mesures de TDoA	41
Figure 2.5	Positionnement par RSS	43
Figure 2.6	Principe de localisation par la méthode d'AoA.....	44
Figure 2.7	Localisation basée sur la phase du signal reçu	45
Figure 2.8	Erreur circulaire probable	53
Figure 2.9	Effet des multi-trajets sur l'estimation du ToA du DP [66].....	59
Figure 2.10	Erreur de distance dans les conditions UDP [66]	61
Figure 2.11	Profil du canal multi-trajets [2].....	61
Figure 3.1	Diagramme comparatif entre les radios traditionnelles et les SDR [56]	64
Figure 3.2	Structure générique en blocs d'une radio définie par logiciel [73].....	66
Figure 3.3	Diagramme bloc de la carte mère USRP N210 [71].....	68
Figure 3.4	Diagramme bloc de la carte fille RFX-2400 [71].....	70
Figure 3.5	Antenne utilisée au cours de ce projet	71
Figure 3.6	Diagrammes de rayonnement de l'antenne utilisée [77]	72
Figure 3.7	Structure en couche du <i>framework</i> GNU-Radio.....	73
Figure 3.8	Architecture hybride de la SDR incluant logiciel et matériel [75]	74
Figure 3.9	Diagramme d'une application GNU-Radio interfacée avec l'USRP	75
Figure 3.10	Diagramme d'une application GNU-Radio sans connexion RF.....	76

Figure 3.11	Fonctionnalités apportées par les pilotes UHD [71].....	77
Figure 3.12	Concept de diagramme de flux avec les mémoires tampons GR [76]....	80
Figure 4.1	Les canaux Wifi IEEE 802.11 [86].....	84
Figure 4.2	Débits et schémas de modulations du protocole IEEE 802.11b [86].....	84
Figure 4.3	Diagramme de flux du projet <i>WiFiLocalization</i>	88
Figure 4.4	Schéma bloc des modules relatifs au récepteur neoTx	92
Figure 4.5	Schéma bloc des modules relatifs à l'émetteur neoRx	92
Figure 4.6	Banc d'essai pour le réglage des paramètres nécessaires à la réception.	94
Figure 4.7	Spectre des signaux reçu par GRC (BPSK et QPSK).....	95
Figure 4.8	Constellations des signaux démodulés via GRC (BPSK et QPSK).....	95
Figure 4.9	Spectre du signal BPSK reçu, $F_{\text{éch.}} = 25$ MHz.....	96
Figure 4.10	Spectre du signal QPSK reçu, $F_{\text{éch.}} = 25$ MHz.....	96
Figure 4.11	Protocole implémenté, mode BPSK à 1Mbps	98
Figure 4.12	Protocole implémenté, mode QPSK à 2Mbps	98
Figure 4.13	Banc d'essai illustrant le récepteur fixe et l'émetteur mobile	99
Figure 4.14	Délais de transmission en fonction de la distance réelle entre radios...100	
Figure 4.15	Distances moyennes en se basant sur les délais de transmission.....100	
Figure 4.16	Distribution de la valeur moyenne des RSS	101
Figure 4.17	Distances brutes estimées à partir des RSS	101
Figure 4.18	Délais à distance fixe en fonction du nombre de paquets.....102	
Figure 4.19	RSS en fonction du nombre des paquets reçus à une distance fixe	103
Figure 5.1	Latences introduites par le système sur les signaux	106
Figure 5.2	Latences introduites par l'architecture de l'USRP N210 [73].....	107
Figure 5.3	Principe du procédé de calibration.....	109

XVIII

Figure 5.4	RSS en fonction de la distance entre les USRP	112
Figure 5.5	Estimées de distances en fonction de la distance entre les USRP	113
Figure 5.6	Diagramme de flux GRC du nouveau module d'estimation de RSS	114
Figure 5.7	Propagation des étiquettes comprenant l'horodatage logiciel [71].....	115
Figure 5.8	Histogramme des délais entre 2 interrogations du temps du FPGA	117
Figure 5.9	Format d'un long entête PPDU pour trame 802.11b [86].....	118
Figure 5.10	Format d'un court entête PPDU pour trame 802.11b [86]	119
Figure 5.11	Schéma bloc de la solution d'horodatage au sein du FPGA	120
Figure 5.12	Principe du corrélateur conçu	121
Figure 5.13	Décalages et dérives d'horloges entre les machines selon PTPd	123

LISTE DES ABRÉVIATIONS, SIGLES ET ACRONYMES

ADC	Analog to Digital Converter
AoA	Angle of Arrival
API	Application Programming Interface
BBN	Bolt, Beranek and Newman
BPSK	Binary Phase Shift Keying
BLPA	Bluetooth Local Positioning Application
BICR	Borne Inférieure de Cramer-Rao
BS	Base Station
BSSID	Basic Service Set IDentification
CGRAN	Comprehensive GNU Radio Archive Network
DAC	Digital to Analog Converter
DDC	Digital Down Converter
DQPSK	Differential Quadrature Phase-Shift Keying
DSP	Digital Signal Processor
DSSS	Direct Sequence Spread Spectrum
DUC	Digital Up Converters
DGP	Dilution Géométrique de Précision
EIRP	Equivalent Isotropically Radiated Power
EAS	Electronic Article Surveillance
ECP	Erreur Circulaire Probable
EKF	Extended Kalman Filter

EMD	Erreur de Mesure de Distance
EQM	Erreur Quadratique Moyenne
FAF	Floor Attenuation Factor
FPC	Fonction de Probabilité Cumulative
FPGA	Field Programmable Gate Array
GCC	GNU Compiler Collection
GUI	Graphical User Interface
IEEE	Institute of Electrical and Electronic Engineering
ISM	Industrial, Scientific and Medical
KNN	K-Nearest Neighbors
LAN	Local Area Network
LoS	Line of Sight
MS	Mobile Station
NCO	Numerically Controlled Oscillator
NLoS	Non Line of Sight
NRL	Naval Research Laboratory
NTPD	Network Time Protocol Daemon
PDU	Protocol Data Unit
PLCP	Physical Layer Convergence Procedure
PN	Pseudo Noise
PPDU	PLCP Protocol Data Unit
RSS	Received Signal Strength
RMS	Root Mean Square

RTT	Round Trip Time
RFID	Radio Frequency IDentification
RToF	Round Trip Time of Flight
RWGH	Residual Weighting
SDR	Software Defined Radio
SMP	Smallest M-vertex Polygon
SNR	Signal to Noise Ratio
SpB	Samples per Baud
SVM	Support Vector Machine
TDoA	Time Difference of Arrival
ToA	Time of Arrival
TPB	Thread Per Block
UHD	Universal Hardware Driver
USRP	Universal Software Radio Peripheral
UWB	Ultra Wide Band
VRT-49	VITA 49 Radio Transport
WAF	Wall Attenuation Factor
WLAN	Wireless Local Area Network
WPI	Worcester Polytechnic Institute
WSN	Wireless Sensor Network

INTRODUCTION

Depuis que les systèmes universels de radiolocalisation par satellites GPS ont été livrés pour le grand public, les avancées dans les domaines de localisation d'extérieur et d'intérieur n'ont cessé de se faire entendre tant en termes de services qu'en termes de systèmes. Étant donné le caractère obstruant des murs des édifices et le caractère encombrant des différents meubles, la localisation d'intérieur est devenue une exigence imposée pour pallier aux difficultés des signaux satellite à garantir un service de positionnement fiable et exact. Ceci dit, l'amélioration des qualités des produits et des services reste un défi ouvert et constant, les performances des systèmes se voient bornés par les limites théoriques énoncés par Shannon, sans oublier aussi les contraintes technologiques.

À cet effet est apparu le paradigme de radio définie par logiciel ou *Software Defined Radio* (SDR). Son but est de transposer les vicissitudes du monde analogique des signaux radio fréquences RF dans le monde numérique du logiciel en des difficultés plus claires et simples à gérer. Ce même paradigme s'inscrit dans un cadre plus large qui est celui de la radio cognitive. Il existe encore bien peu de radios se réclamant comme étant cognitives. Dans ce cadre, on est amené à concevoir un outil logiciel capable d'agréger de multiples techniques et algorithmes de positionnement à savoir les techniques basées sur la puissance des signaux reçus (RSS) et les techniques basées sur les différences des délais de propagations des signaux ou "*Time Difference of Arrival*" (TDoA).

CHAPITRE 1

MISE EN CONTEXTE

1.1 Introduction

La localisation est devenue une partie indissociable de notre vie quotidienne grâce à l'omniprésence des services offerts par les systèmes de positionnement globalement dans les récepteurs dédiés ou les téléphones intelligents. À la base de la localisation, on trouve les signaux provenant de sources radio à savoir les satellites, les tours cellulaires et les points d'accès sans fil. Les applications pour un tel système comprennent la navigation piétonne, la localisation des pompiers et autres personnels à l'intérieur d'un bâtiment, ainsi que la détection et l'isolement des menaces possibles à la sécurité du bâtiment.

Il existe plusieurs techniques pour la localisation, principalement celles utilisant les données de délais temporels des signaux ou leurs puissances. Bien que le comportement des ondes électromagnétiques dans l'espace libre soit connu pour être prévisible, les techniques de localisation doivent tenir compte des erreurs spécifiques à l'environnement afin d'atteindre un niveau de précision admissible. Alors que les systèmes d'extérieur disponibles aujourd'hui sont suffisants pour les applications qu'ils desservent, les environnements intérieurs exigent des estimations de position beaucoup plus précises étant donné les nombreuses sources d'erreurs que les applications extérieures ne sont pas obligées d'en tenir compte. Il existe deux approches de base pour la conception d'un système de géolocalisation sans fil. La première consiste à développer un système de signalisation et une infrastructure de réseau de capteurs dédiés à la localisation. La deuxième consiste à utiliser une infrastructure de réseau sans fil existante pour localiser une station mobile ou *Mobile Station MS*.

L'avantage de la première approche est que la spécification physique, et conséquemment la qualité des résultats de localisation, est sous le contrôle du concepteur. La MS peut être conçue sous la forme d'une petite balise portable, et la densité des capteurs peut être réglée en vue de la précision requise selon l'application et l'environnement [3]. Avec la seconde

approche, on évite le déploiement d'une infrastructure dédiée coûteuse et qui nécessite un temps considérable. Ces systèmes ont cependant besoin d'utiliser des algorithmes plus intelligents pour compenser la faible précision des paramètres mesurés [4].

À la figure 1 est illustré un schéma bloc fonctionnel de système de géolocalisation sans fil. Les principaux éléments du système sont:

- un ensemble de capteurs ou stations de bases aux emplacements connus mesurant la position relative d'une station mobile (MS) par rapport à un point de référence ;
- un algorithme de positionnement estimant les coordonnées de la station mobile ;
- un dispositif d'affichage illustrant l'emplacement des MS pour les utilisateurs.

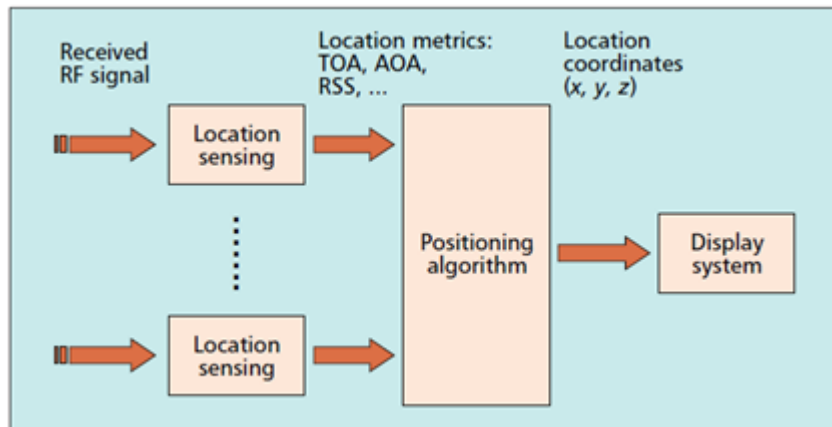


Figure 1.1 Diagramme bloc des systèmes de géolocalisation sans fil [2]

Le système d'affichage peut simplement indiquer les coordonnées de la MS, ou bien identifier son emplacement relatif dans une zone donnée. Cet affichage peut être logiciel sur un ordinateur privé ou serveur comme dans le cas des systèmes de suivi des employés dans une entreprise ou personnel monitoring. L'affichage peut aussi être embarqué sur la MS elle-même comme les applications téléphoniques du récepteur GPS.

1.2 Objectifs du projet

Les recherches sur les fondements théoriques du problème de localisation abondent, mais les implémentations à faible coût de ses solutions se font rares. À cet effet, l'objectif de ce travail est d'implémenter un système de localisation sur une plateforme de radios définies par logiciel qui est l'USRP N210. Étant dotée d'une grande capacité computationnelle, cette plateforme peut servir à couvrir pas mal d'applications dans ce sens. On vise durant ce projet à implémenter un algorithme de positionnement hybride servant à localiser les utilisateurs du réseau sans fil d'intérieur ou "WiFi" et ce en respectant un ensemble de métriques notamment la précision et l'exactitude. Ce travail propose une solution hybride dans le sens où elle combine deux sortes d'informations sur les signaux: la puissance du signal reçu (RSS) et la différence des instants d'arrivée du signal (TDoA). Ces deux informations sont communiquées à travers un réseau sans fil compatible avec IEEE 802.11. Trois radios USRP N210 aux emplacements connus créent un réseau de capteurs pour localiser un émetteur sans fil utilisant la même norme et le même type de radio. La configuration du système est illustrée à la Figure 1.2.

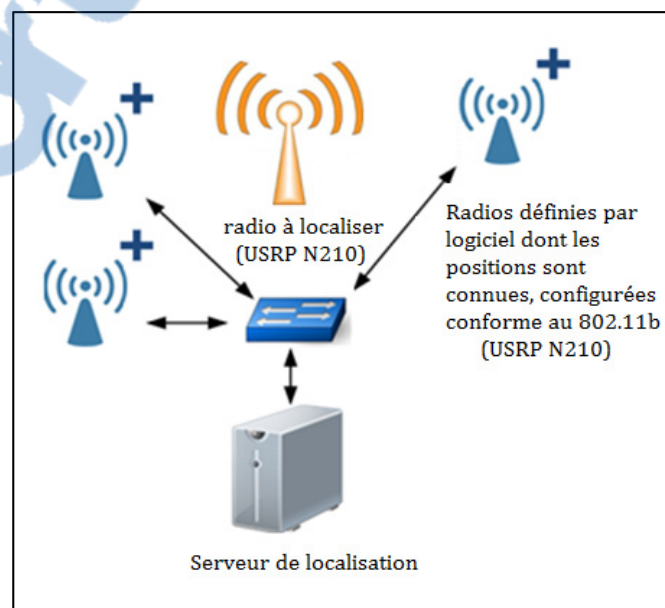


Figure 1.2 Système de localisation basé sur les radios USRP N210

En utilisant des méthodes de TDoA et RSS, la distance entre chacun des capteurs et l'émetteur peut être estimée. Les distances ainsi que les coordonnées des capteurs sont introduites dans un algorithme de trilatération afin de calculer les coordonnées de l'émetteur. On procède au cours de ce projet à l'exploitation de deux implémentations qui sont :

- le projet *BBN 802.11* qui met au point une version compatible avec le protocole 802.11b qui a été testée sur les radios définies par logiciel USRP 2 ;
- le projet *Wifilocalization* qui consiste en un ensemble de modules basiques de localisation d'intérieur exploitant le projet précédant BBN 802.11.

Le système *Wifilocalization* propose un module de post-traitement pour les données recueillis qui sert à compenser les altérations apportées par les caractéristiques complexes de l'environnement d'intérieur. Les objectifs du système à concevoir sont comme suit:

- mettre à niveau les deux projets à la dernière version de la bibliothèque GNU-Radio;
- garantir la compatibilité avec les nouveaux pilotes des radios USRP ;
- décrire les améliorations apportées au mécanisme de *time stamping* logiciel et la nouvelle méthode proposée pour la synchronisation des récepteurs ;
- montrer quel niveau de précision ces mécanismes sont à même de garantir dans un environnement d'intérieur ;
- concevoir une alternative de *time stamping* au sein du FPGA et présenter les métriques correspondantes ;
- intégrer une nouvelle méthode d'estimation des RSS au système et incorporer un modèle plus réaliste de la distance à partir de ces mesures.

1.3 Méthodologie

La première des étapes est de faire l'étude des différentes techniques de positionnement relatif et absolu ainsi que de survoler la panoplie des algorithmes existants portant sur la radio localisation. Vient ensuite la conception du module générique combinant les techniques

et agrégeant les résultats. Après, on valide l'implémentation du module conçu avec l'outil GNU-Radio. À ce stade-là, il est question d'exécution et de test des scénarios de positionnement sur ladite plateforme matérielle USRP N210.

Le chapitre 2 fournit une définition des thèmes clés liés à la localisation sans fil. Le chapitre suivant traite les aspects de la plateforme utilisée pour ce système tant au niveau logiciel que matériel. Le cadre du développement ainsi que les projets qui composent le système de localisation sont exposés au chapitre 4. Les étapes de mise à jour du système sur GNU-Radio sont aussi décrites au même chapitre. On y présente aussi les tests préliminaires ainsi que les résultats obtenus. Il importe de noter qu'à ce niveau-là du projet, on a réalisé que le mécanisme d'horodatage au niveau logiciel est loin de permettre d'atteindre une précision raisonnable pour la localisation d'intérieur. Le chapitre 5 est dédié en partie à l'étude des causes de la latence variable inhérente au système, des aspects d'amélioration des estimées de latences du système étudié et aux améliorations apportées ainsi que des résultats correspondants. Les détails de conception de la solution alternative d'horodatage au niveau du FPGA et la précision éventuelle à obtenir sont aussi montrés au chapitre 5. Le mémoire se termine avec une présentation des conclusions et des recommandations pour les travaux futurs à ce sujet.

CHAPITRE 2

PRINCIPES DE LOCALISATION D'INTÉRIEUR

2.1 Introduction

Ce chapitre contient les définitions des mécanismes, concepts et notions générales relatifs à la localisation de sources radios dans un environnement de réseaux locaux sans fil. Dans un premier lieu, l'état de l'art des systèmes existants est présenté. Par la suite, les différentes mesures de distance ainsi que les algorithmes communément utilisés pour la localisation d'intérieur sont examinés. La dernière section est consacrée à la définition des métriques de performances desdits systèmes et à l'étude des multiples sources d'erreurs altérant le mécanisme de positionnement.

2.2 État de l'art des systèmes de localisation d'intérieur

La localisation constitue la pierre angulaire pour d'innombrables applications sans fil. Les informations de positionnement sont exploitées par les opérateurs cellulaires pour repérer la source des appels d'urgence et gérer les ressources d'infrastructure. En outre, le service de localisation assuré par les téléphones cellulaires et les récepteurs GPS s'avère être une nécessité pour la navigation. On se sert aussi des informations de positionnement pour la gestion de flotte en faisant le suivi des objets tels que les véhicules et les biens précieux.

Principalement poussé par le mandat américain du *Enhanced 911*, qui exige d'être en mesure de localiser les téléphones mobiles avec une précision allant de 50 jusqu'à environ 100 m, mais aussi par les nouveaux services basés sur la localisation commerciale, maints systèmes et techniques de positionnement sont conçus et déployés [5]. De nombreuses techniques de localisation à l'intérieur exigent un processus connu sous le nom d'empreintes digitales ou *fingerprinting* à effectuer avant de pouvoir localiser avec précision les appareils mobiles. Ce processus, fastidieux et coûteux, doit être répété pour chaque nouvel environnement ou lorsque l'environnement varie. Ces contraintes limitent considérablement le nombre et les

types d'applications pouvant utiliser la localisation d'intérieur. En conséquence, une méthode robuste nécessitant peu ou pas de connaissance de la région est grandement nécessaire. Il importe de définir ici les topologies de systèmes locaux de positionnement sans fils et de se situer à l'intérieur de cette gamme. Ci-après, au Tableau 2.1, sont définies ces topologies.

Tableau 2.1 Topologies des systèmes locaux de positionnement sans fils [5]

Concept	Définition
Positionnement à distance	Mesures à partir d'un site distant vers la station mobile.
Auto-positionnement	Mesures à partir de la station mobile vers des points de repère généralement fixes.
Auto-positionnement indirect	Système d'auto-positionnement avec le transfert des données du résultat de mesure à un site distant.
Positionnement à distance indirect	Système de positionnement à distance avec le transfert des données du résultat de mesure à la station mobile.

Les dispositifs sans fil peuvent être localisés de plusieurs manières. Dans les applications où l'appareil mobile doit connaître son propre emplacement, le mobile compte sur les émetteurs sans fil existants et déployés à des emplacements connus. Le système de localisation globale (GPS) en est l'exemple. Globalement, on différencie trois méthodes pouvant fournir une infrastructure de trilatération. Une des solutions consiste à mettre en place une infrastructure sur-mesure qui est déployée au besoin. Toutefois, cette souplesse vient à grands frais, ce coût en temps et en argent est au-delà de la portée de ce projet.

Une autre solution courante consiste à utiliser les signaux RF existants qui sont à priori produits à d'autres fins pour la localisation, notamment les signaux de téléphonie cellulaire ou les signaux de données des réseaux locaux sans fils. Toutefois, ce type de système de localisation est limité par les propriétés du signal utilisé. Par exemple, pour exploiter le

réseau WiFi et faire usage de ces types de signaux, le concepteur est obligé d'utiliser l'étalement spectral à séquence directe (DSSS) sur la bande 2,4 GHz.

Une troisième solution consiste à développer une infrastructure dynamique à travers un réseau ad-hoc. Cette méthode introduit le concept des positions relatives. Cela présente des avantages dans des environnements hostiles où les infrastructures fixes ne peuvent être placés, comme dans le contexte des mines. D'autres techniques peuvent être classées comme des combinaisons des méthodes ci-dessus.

En outre, nombreux systèmes intègrent la localisation des signaux RF avec d'autres technologies telles que les ondes acoustiques, les accéléromètres et les gyroscopes. L'entrée de données supplémentaires est à même de fournir une précision améliorée, comme pour le GPS assisté (A-GPS) [8]. La Figure 2.1 présente la panoplie des systèmes de positionnement sans fils couvrant les milieux d'intérieur et d'extérieur accompagnés de leurs résolutions.

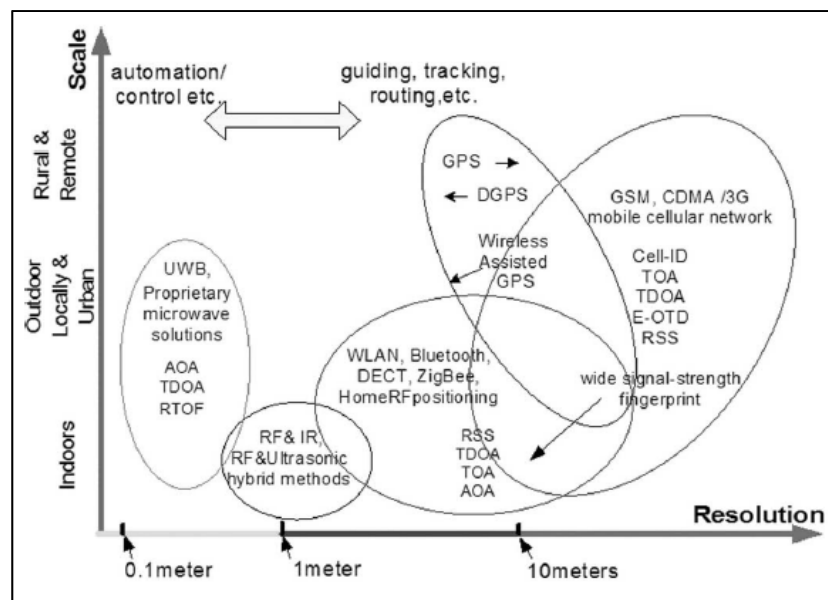


Figure 2.1 Aperçu des systèmes actuels basés sur le positionnement sans fils [9]

2.2.1 Systèmes basés sur le Wifi ou IEEE 802.11

Cette norme du réseau local sans fil (WLAN), fonctionnant dans la bande ISM des 2,4 GHz, est devenue très populaire dans les *hotspots* publics et les lieux des entreprises au cours des dernières années. Avec un débit binaire brut typique de 11, 54, ou 108 Mbps et une portée de 50 à 100 m, IEEE 802.11 est actuellement la norme dominante du réseau local sans fil. Par conséquent, l'utilisation de son infrastructure existante pour fins de localisation d'intérieur constitue une opportunité. La précision des systèmes de positionnement WLAN typiques utilisant les données RSS est approximativement de 3 à 30m, avec une fréquence de mise à jour de l'ordre de quelques secondes [9].

Bahl *et al.* [10] présentent un système d'intérieur de positionnement et de suivi ou *tracking* nommé RADAR. Les auteurs proposent deux types d'approches pour déterminer l'emplacement. La première dépend des mesures empiriques de la force des signaux du point d'accès lorsqu'il est hors ligne. La deuxième se base sur une modélisation du canal de propagation. Les modèles de propagation basés sur le facteur d'atténuation des murs WAF et celui du plancher FAF sont utilisés, à la place du modèle de Rayleigh ou de Rice, qui sont plutôt utilisés en milieu d'extérieur. La précision du système RADAR est d'environ 2-3 m. Dans leur deuxième travail [11], le système a été amélioré par un algorithme similaire à celui de Viterbi. La médiane de l'erreur ou le résultat est tel que le 50 percentile du système est d'environ 2,37 à 2,65 m et le 90 percentile est d'environ 5,93 à 5,97 m.

Maintes autres méthodes probabilistes sont utilisées pour la conception de systèmes de localisation d'intérieur à savoir la classification commune ou *joint clustering* [12, 13] les réseaux bayésiens basés sur les grilles [14, 15], les classificateurs basés sur les réseaux de neurones [16] ou bien la complexité stochastique et l'apprentissage concurrentiel en ligne [17]. Les systèmes de positionnement sans fils sont aussi exploités dans la robotique mobile. Déterminer la position d'un robot mobile donné, en utilisant multiples entrées provenant de différents capteurs, y compris les signaux GPS, sonar, la vision et des capteurs à ultrasons, est un sujet qui est abordé par de nombreux chercheurs [18].

Alors que la plupart des systèmes basés sur WLAN utilisent la puissance du signal reçu, AeroScout [19] utilise une solution de localisation par TDoA au sein du 802.11. Il requière une synchronisation rigoureuse de quelques nanosecondes entre les différents récepteurs d'un même signal pour aboutir à un résultat admissible.

2.2.2 Systèmes de localisation de type Bluetooth

Bluetooth ou IEEE 802.15 fonctionne dans la bande ISM 2,4 GHz. Par rapport au réseau local sans fil, le débit binaire brut est inférieur (1 Mbps), et la portée est plus courte (typiquement de 10 à 15 m). D'autre part, Bluetooth est omniprésent. Il est intégré dans la plupart des téléphones et assistants numériques personnels (PDA). *Topaz* est un système de positionnement local [20] basé sur l'infrastructure Bluetooth de la compagnie *Tadlys*. Ce système offre une précision spatiale de 2 m avec une fiabilité de 95%. Le délai de positionnement est de 15-30 s. Antti *et al.* [21] présentent la conception et l'implémentation d'une application locale de positionnement Bluetooth (BLPA). Tout d'abord, ils convertissent les niveaux de puissance des signaux reçus à des estimées de distance en fonction d'un simple modèle de propagation, et ensuite, ils utilisent un filtre de Kalman étendu (EKF) pour calculer l'estimée de position en 3-D. La précision de BLPA est rapporté à 3,76 m.

2.2.3 Systèmes d'intérieur de type GPS

SnapTrack, [22] une société de *Qualcomm*, est la première à avoir introduit le GPS assisté au sans fil (A-GPS) pour surmonter les limitations du GPS classique, et à fournir une localisation d'intérieur de précision moyenne de 5 à 50 m. Récemment, Atmel [23] et U-blox [24] ont annoncé la disponibilité d'une nouvelle technologie de suivi ou *tracking* des faibles signaux GPS, appelé *SuperSense*.

Locata Corporation a inventé une nouvelle technologie de positionnement appelée *Locata*. Elle se compose en partie d'un *pseudolite* émetteur-récepteur synchronisé en temps appelé

LocataLite. Un réseau de *LocataLite* forme un *LocataNet*, qui transmet des signaux de type GPS, permettant de positionner un mobile en utilisant des mesures de phase de porteuse. Les tests relatifs révèlent une preuve de concept de cette technologie et la garantie d'un positionnement avec une précision en dessous des centimètres [25].

2.2.4 Localisation dans un réseau de capteurs sans fils ou WSN

Le système GPSless est une méthode de positionnement qui nécessite un certain nombre de nœuds ou capteurs placés à des positions connues formant un maillage et transmettant des signaux de balise périodiques contenant leurs positions respectives. L'algorithme employé n'a pas besoin de mesures de distance précises, il utilise seulement l'information selon laquelle un nœud est à portée ou hors de portée d'un certain nœud balise [26, 27].

2.2.5 Systèmes de localisation de type RFID

Les systèmes d'identification par radiofréquence ou RFID sont des systèmes passifs confinés à une zone de couverture radio à très courte portée. Ils reflètent le signal émis par un émetteur après l'avoir identifié individuellement par un récepteur généralement voisin. Le système en question est essentiellement basé sur la localisation cellulaire, mais à un niveau local, comme par exemple dans [28, 29], où l'on rapporte une portée de moins de un mètre. Ces systèmes sont peu coûteux et facile à intégrer, par exemple dans des étiquettes en papier imprimables pour les produits de détail. Un exemple de système utilisant cette technologie est SpotON [30]. Ce dernier utilise un algorithme d'agrégation pour la localisation 3D basé sur l'analyse des RSS. Un autre système du genre est appelé LANDMARC. Ce système, par contre, fait usage des étiquettes RFID actives [31].

2.2.6 Autres systèmes

Alors que la localisation d'extérieure est presque exclusivement réalisée en utilisant le GPS, les systèmes de localisation d'intérieur utilisent avec succès une variété de technologies. Le

prototype original '*Active Badge*', suivi par les systèmes commerciaux comme *Versus* [32, 33], utilisent des émetteurs/détecteurs infrarouge pour obtenir une précision de 5 à 10m. Les systèmes '*Cricket*' et '*Bat*' [34, 35], incorporent, quant à eux, les signaux ultrasons afin d'estimer les distances. En fonction de la densité de l'infrastructure et le degré d'étalonnage, les systèmes à ultrasons ont une précision entre quelques mètres et quelques centimètres.

Plus récemment, les émetteurs-récepteurs à ultra-large bande ont été utilisés pour permettre une localisation d'intérieur très précise [36]. L'inconvénient commun de tous ces systèmes est qu'ils nécessitent une infrastructure sur-mesure et donc coûteuse.

2.3 Mesures de distances

Les mesures peuvent être réparties principalement en trois classes : les mesures temporelles, celles qui reposent sur la puissance des signaux reçus et les mesures basées sur l'angle d'arrivée des signaux. Il existe aussi les mesures combinant plusieurs types de signaux. Voici la description des méthodes employées dans le cas de courte portée, environnement d'intérieur et antenne simple.

2.3.1 Techniques de triangulation

La triangulation, comme son nom l'indique, utilise les propriétés géométriques des triangles pour estimer l'emplacement cible. Elle a deux variantes: la multilatération et l'angulation. Dans cette partie, on se concentre sur la première variante qui estime la position d'un objet en mesurant sa distance à partir de points de référence. On différencie nombreuses techniques qu'on examine une à une.

2.3.1.1 L'instant d'arrivée ou ToA

Ces méthodes reposent sur l'estimation du délai de propagation entre l'émetteur et le récepteur. Ce délai est déterminé par l'inter corrélation entre le signal reçu et le signal transmis. L'emplacement du plus grand pic de cette corrélation indique l'arrivée du signal en ligne de vue directe ou *line-of-sight* (LoS), et donne la valeur d'un retard pris par rapport à un temps de référence. L'emplacement et la hauteur du pic sont fortement influencés par le bruit additif, qui dégrade le pic, et par l'interférence des signaux de trajets multiples, qui obscurcit le pic du signal LoS. La distance entre l'émetteur et le récepteur est directement proportionnelle à la durée de propagation de l'onde en question (acoustique ou électromagnétique) comme l'indique (2.1).

$$\text{Distance}_{\text{émetteur-récepteur}} = (T_{\text{réception}} - T_{\text{émission}}) * \text{célérité_de_l'onde} \quad (2.1)$$

Où T réfère à un instant donné dans le temps. Afin de permettre le positionnement en 2-D, les mesures ToA doivent être prises par rapport aux signaux provenant d'aux moins trois points de référence, comme indiqué dans la Figure 2.2. Les mesures imparfaites créent une région d'incertitude autour de chaque capteur, dans laquelle l'émetteur pourrait être contenu comme le montre la Figure 2.3.

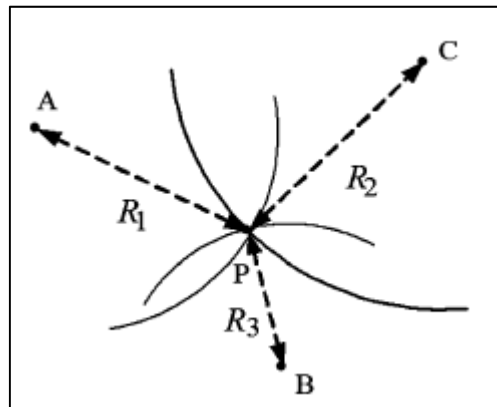


Figure 2.2 Localisation en 2-D avec ToA, cas idéal

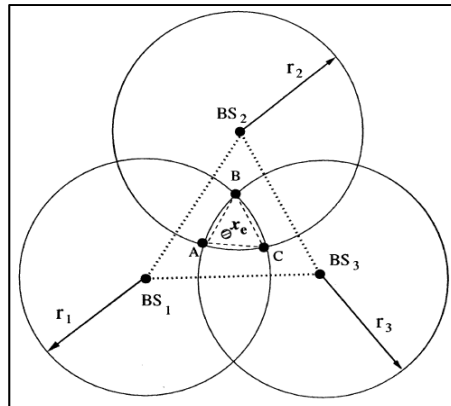


Figure 2.3 Localisation en 2-D avec ToA, tenant compte des imprécisions

Étant donné que la ToA repose sur la différence entre l'instant d'arrivée et celui d'émission, tous les récepteurs et les émetteurs doivent être synchronisés au niveau des horloges. Ils requièrent généralement un matériel plus sophistiqué et une référence de temps absolu. Cela se révèle être le cœur même de la recherche dans ce contexte, compte tenu de la célérité à laquelle les signaux se propagent et le besoin de synchronisation [37]. La ToA peut être mesurée par différentes techniques de signalisation comme l'étalement de spectre par séquence directe (DSSS) [38, 39] ou les mesures à ultra-larges bandes (UWB) [40]. La position de la cible peut également être calculée par minimisation de la somme des carrés de la fonction de coût. À la section 2.4.2.1, on décrit plus en détail cette méthode de minimisation. On suppose que la MS, située à (x_o, y_o) , transmet un signal à l'instant t_o , les N stations de base situées à (x_1, y_1) , (x_2, y_2) , ..., (x_n, y_n) reçoivent le signal aux instants t_1, t_2, \dots, t_n . En tant que mesure de performance, la fonction de coût peut être formée comme:

$$F(x) = \sum_{i=1}^N \alpha_i^2 f_i^2(x) \quad (2.2)$$

où α_i peut être choisi pour refléter la fiabilité du signal reçu à l'unité de mesure i et $f_i(x)$ est donnée comme suit: $f_i(x) = c * (t_i - t) - \sqrt{(x_i - x)^2 + (y_i - y)^2}$, où c est la célérité de la lumière, et $x = (x,y,t)^T$. Cette fonction est formée pour chaque unité de mesure, $i = 1, \dots, N$,

et $f_i(x)$ pourraient être apportées à zéro avec le bon choix de x , y et t . L'estimation de l'emplacement est déterminée en minimisant la fonction $F(x)$.

2.3.1.2 La différence des instants d'arrivées ou TDoA

La différence des instants d'arrivée (TDoA), utilise la multilatération, ou le positionnement hyperbolique. L'idée est de déterminer la position relative de l'émetteur mobile en examinant la différence dans le temps où le même signal connu arrive à de multiples récepteurs, plutôt que l'instant d'arrivée absolu ou ToA. Pour chaque mesure de TDoA, l'émetteur est situé sur un hyperboloïde à une différence de plage constante entre les deux unités de mesure. L'équation de l'hyperboloïde est donnée par :

$$R_{i,j} = \sqrt{(x_i - x)^2 + (y_i - y)^2 + (z_i - z)^2} - \sqrt{(x_j - x)^2 + (y_j - y)^2 + (z_j - z)^2} \quad (2.3)$$

où (x_i, y_i, z_i) et (x_j, y_j, z_j) représentent les coordonnées des récepteurs fixes i et j , et (x, y, z) représentent les coordonnées de la cible [5]. L'emplacement de la cible en 2D peut être estimé à partir de l'intersection de deux ou plusieurs mesures TDoA, comme indiqué dans la Figure 2.4. Deux hyperboles sont formées à partir de mesures TDoA à trois unités de mesure fixes (A, B, et C) afin de fournir un point d'intersection, qui localise la cible P [37, 43].

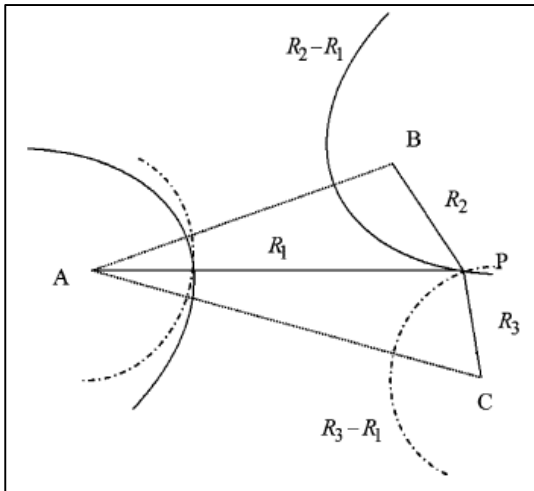


Figure 2.4 Localisation basée sur les mesures de TDoA

La TDoA est relativement immunisée contre les erreurs de timing. Tout biais d'horloge interne à chaque récepteur est éliminé puisque le calcul de la différence ne tient pas compte d'une référence de temps absolu. Cela donne place à une approche de localisation asynchrone moins coûteuse. En fait, l'émetteur n'est pas tenu d'être synchronisé avec le récepteur. La synchronisation est uniquement nécessaire entre les récepteurs. Il a été montré que les estimations de position utilisant la TDoA permettent d'obtenir de meilleures performances que les méthodes ToA en particulier dans des environnements à trajets multiples [37]. À l'exception des solutions précises de l'équation hyperbolique de TDoA (2.3) par régression non linéaire, une solution plus simple consiste à linéariser les équations en utilisant un développement de série de Taylor et un algorithme itératif [44].

Les méthodes conventionnelles de calcul des estimations TDoA sont basées sur la corrélation. TDoA peut être estimée à partir de l'inter corrélation entre les signaux reçus à une paire d'unités de mesure. Supposons que pour le signal transmis $s(t)$, le signal reçu à la station de base i est $x_i(t)$. Supposons que $x_i(t)$ est altéré par le bruit $n_i(t)$ et retardé par d_i , alors $x_i(t) = s(t - d_i) + n_i(t)$. De la même façon on obtient le signal arrivant à la station j , retardé par d_j et altéré par le bruit $n_j(t)$, $x_j(t) = s(t - d_j) + n_j(t)$. On définit ici la fonction d'inter corrélation des signaux reçus par les deux récepteurs i et j :

$$\hat{R}_{x_i, x_j}(\tau) = \frac{1}{T} \int_0^T x_i(t) x_j(t - \tau) dt \quad (2.4)$$

La TDoA estimée est la valeur de τ qui maximise $\hat{R}_{x_i, x_j}(\tau)$. Cette approche exige que les récepteurs partagent une référence de temps précise mais n'impose pas d'exigence sur la station mobile. Les techniques de traitement du domaine fréquentiel sont habituellement utilisées pour trouver τ . Exceptées les précédentes méthodes, une technique de mesure de TDoA basée sur le délai a été proposée dans [39] pour les réseaux 802.11, qui élimine le besoin de synchronisation initiale dans les procédés classiques.

2.3.1.3 Le temps d'aller-retour du signal ou RToF

Cette méthode consiste à mesurer le temps d'aller-retour d'un signal entre l'émetteur et la station de base. Pour RToF, la condition de synchronisation d'horloge est relativement moins stricte que celle requise dans le ToA, mais leurs mécanismes de mesure de distance est le même. Les algorithmes de positionnement pour ToA peuvent être directement appliqués pour RToF. La station de base est considérée comme un radar ordinaire. Un transpondeur cible répond au signal d'interrogation du radar. Toutefois, il est encore difficile de connaître le délai de traitement causé par la station de base. Dans les systèmes de longue ou moyenne portée, ce délai peut être ignoré s'il est négligeable devant le temps de transmission. Une autre approche consiste à utiliser le concept de réflexion modulée [45], adapté seulement aux systèmes à courte portée. Un algorithme de mesure de RToF des paquets du réseau local sans fil est présenté dans [46] rapportant une erreur de mesure de quelques mètres.

2.3.1.4 La puissance du signal reçu ou RSS

La distance peut aussi être estimée en utilisant l'indication d'intensité de signal reçu (RSSI). Cette méthode nécessite généralement beaucoup de travail à priori. La distance entre l'émetteur et le récepteur peut être estimée par la comparaison des RSS avec l'historique enregistré au préalable à différents endroits du site. La modélisation de perte de trajet peut

être utilisée en fonction du site de test. Une grande marge d'erreur et de variabilité persiste et nécessite la prise en compte de modèle plus réaliste et significatif. Les modèles théoriques et empiriques peuvent être appliqués pour convertir la perte de trajet en une estimation de la distance radiale [37]. Un modèle simple de perte de trajet permet de calculer la distance entre l'émetteur et le récepteur. Il s'agit du modèle log-linéaire qui est présenté dans (2.5).

$$RSSI = 10\alpha \log(d) \quad (2.5)$$

$$d = 10^{(RSSI - RSSI_{calibration}) / (10\alpha)} + d_{calibration}$$

où α représente le gradient de l'environnement et d la distance estimée. La puissance du signal reçu est communément considérée comme une mesure de tension, ou de manière équivalente calculée comme une puissance donnée par le carré de l'amplitude. Cependant, les mesures RSS sont connues pour leurs variations généralement causées par les conditions de fonctionnement et d'environnement. Les sources les plus influentes de l'erreur sont les rayons des trajets multiples et l'ombrage ou *shadowing*. Ces deux aspects sont examinés aux sections 2.6.1.1 et 2.6.1.2. Malgré ces erreurs, la simplicité relative et le faible coût des techniques RSS rendent ces solutions attrayantes pour la localisation. À la Figure 2.5, on présente le principe de triangulation par RSS. Les distances y sont données par les mesures de pertes de trajets ou *pathloss* L_{S1} , L_{S2} et L_{S3} .

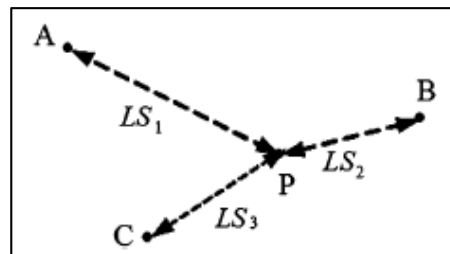


Figure 2.5 Positionnement par RSS

En raison du phénomène d'obstruction ou *shadowing* et d'évanouissement présents en environnement intérieur, les modèles de perte de trajet ne sont pas toujours valables. Les

paramètres utilisés dans ces modèles sont spécifiques au site. La précision de cette méthode peut être améliorée en utilisant des contours RSS pré mesurées centrées sur le récepteur [47]. Un algorithme de logique floue, montré dans [48], est en mesure d'améliorer drastiquement la précision de localisation en utilisant les RSS.

2.3.2 L'angle d'arrivé

Les mesures d'angle d'arrivée (AoA) identifient la direction d'origine d'un signal reçu. L'estimation d'AoA, communément appelé goniométrie ou *direction finding* (DF), peut se faire soit avec des antennes directionnelles soit avec un réseau d'antennes. Contrairement à l'approche de TDoA, la précision des estimées de direction dépend du chemin de vue directe LoS entre l'émetteur et les antennes réceptrices. Dans l'AoA, la position de la cible se trouve à l'intersection de plusieurs paires de lignes de direction d'angle, chacune formée par le rayon circulaire à partir d'une station de base ou une station balise à la cible mobile [37]. Comme le montre la Figure 2.6, l'AoA peut utiliser au moins deux points de référence connus A et B et deux angles (θ_1 , θ_2) pour dériver l'emplacement 2-D de la cible P [5]. Cependant, utiliser uniquement les informations d'angle pour déterminer la position n'est pas une pratique courante pour la localisation RF. Au lieu de cela, les mesures d'AoA sont utilisées pour compléter d'autres techniques.

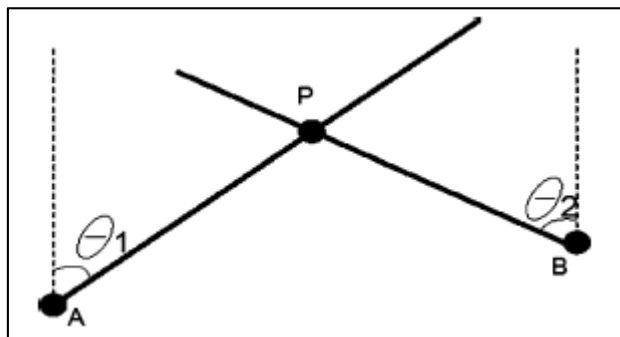


Figure 2.6 Principe de localisation par la méthode d'AoA

Les avantages de l'AoA sont tels qu'une estimation de la position peut être déterminée avec seulement trois unités de mesure pour un positionnement 3-D ou deux unités pour un positionnement 2-D, et qu'aucune synchronisation d'horloge entre les unités de mesure n'est obligatoire. Les inconvénients comprennent la configuration matérielle qui est relativement complexe et la dégradation significative de la précision de position quand la cible mobile s'éloigne de l'unité de mesure. Un positionnement précis avec AoA doit tenir compte des réflexions venant des directions trompeuses, ou par la mesure de directivité de l'ouverture d'antenne [49, 50].

2.3.3 La phase du signal reçu ou PoA

Cette méthode exploite les différences de phase de la porteuse du signal pour estimer la distance. Elle est aussi appelée la phase d'arrivée ou PoA. Dans la Figure 2.7, les stations de base A à D sont placées à l'intérieur d'un bâtiment cubique fictif. Le délai est exprimé comme une fraction de la longueur d'onde du signal, et est désigné par le symbole $\Phi_i = (2\pi f D_i) / c$ dans la formule $S_i(t) = \sin(2\pi f t + \Phi_i)$, où $i \in (A, B, C, D)$, et c étant la célérité de la lumière. Tant que la longueur d'onde du signal est plus longue que la diagonale du bâtiment cubique, c'est à dire, $0 < \Phi_i < 2\pi$, on peut obtenir l'estimée de distance $D_i = (c\Phi_i) / (2\pi f)$. Ensuite, on peut utiliser les mêmes algorithmes de positionnement de ToA et de TDoA. L'inconvénient de la PoA est sa sensibilité de mesure quant à l'ambiguïté de phase [9].

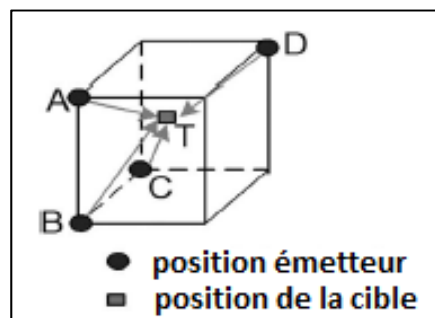


Figure 2.7 Localisation basée sur la phase du signal reçu

2.3.4 Analyse de scène

L'analyse de scène basée sur les ondes RF fait référence aux algorithmes qui recueillent d'abord les spécificités ou les empreintes d'une scène, puis estiment la position d'un objet en faisant correspondre les mesures trouvées avec la valeur la plus proche parmi les empreintes d'à priori. Ce procédé est appelé *fingerpringing* ou *pattern recognition*. Dans ce contexte, on utilise couramment les données RSS comme empreintes. On distingue deux phases pour ce genre de localisation: la phase hors ligne et la phase d'exécution. Lors de la phase hors ligne, une étude de site est réalisée. Il s'agit de recueillir les paires coordonnées/RSS depuis les stations de bases voisines. Lors de la seconde phase, une technique de localisation utilise les RSS présentement observés et précédemment recueilli pour trouver une correspondance et ainsi une estimée de position.

Bahl *et al.* ont trouvé que la puissance du signal à partir d'un point d'accès 802.11 ne varie pas de façon significative à un endroit donné. Ils ont utilisé cette observation pour construire le système RADAR [4]. Ce fut le premier système d'empreintes qui a montré la possibilité de localiser un ordinateur portable dans les couloirs d'un petit immeuble de bureaux avec une précision de 2 à 3 mètres, en utilisant quatre points d'accès 802.11. Plusieurs produits commerciaux de localisation ont été construits en utilisant les empreintes du 802.11 [51]. Le système global pour les communications mobiles ou GSM a, lui aussi, introduit la méthode de *fingerpringing* [12]. Le principal défi pour ces techniques réside dans le fait que les RSS peuvent être affectées par diffraction, la réflexion et la diffusion dans les milieux intérieurs.

2.4 Algorithmes de localisation

On distingue fondamentalement deux classes d'algorithmes de localisation: ceux basés sur la distance et ceux basés sur la reconnaissance de formes ou *pattern recognition*. Un algorithme de localisation à base de distance détermine la distance entre une MS et trois ou plusieurs points de référence qui peuvent ou pas être co-localisé avec cette première. Dans les algorithmes à base de reconnaissance de formes, le système procède à une phase à priori

appelée la reconnaissance du site ou *learning* afin de générer une base de données des références spécifiques au site. On distingue au moins cinq algorithmes de positionnement basés sur les empreintes: les méthodes probabilistes, les k voisins les plus proches ou *K-Nearest Neighbors* (kNN), les réseaux de neurones, la machine à vecteurs de support ou *Support Vector Machine* (SVM), et le plus petit polygone M -vertex ou *Smallest M-vertex Polygon* (SMP) [9].

2.4.1 Algorithmes de localisation basés sur les RSS

Soit (x,y) la position à déterminer de la MS, $O = [P_1 P_2 \dots P_m]$ est un vecteur des RSS observés à partir des m points d'accès AP_1, AP_2, \dots, AP_m situé aux positions $(x_1,y_1), (x_2,y_2), \dots, (x_m,y_m)$ respectivement. Soit $Z = [Z_1(x,y) Z_2(x,y) \dots Z_m(x,y)]$ le vecteur des mesures RSS prévues à la position (x,y) . La position de la MS peut être estimée comme (\hat{x}, \hat{y}) où $Z(\hat{x}, \hat{y})$ donne une approximation de $O = [P_1 P_2 \dots P_m]$. La fonction d'erreur est définie par:

$$\varepsilon(x, y) = \|(O(x, y) - Z(x, y))\|^2 = \sum_{i=1}^m (P_i - Z_i(x, y))^2 \quad (2.6)$$

L'emplacement estimé pour (\hat{x}, \hat{y}) doit minimiser (2.6) d'où :

$$\nabla \varepsilon(x, y) = 0 \quad (2.7)$$

$$\begin{pmatrix} \hat{x} \\ \hat{y} \end{pmatrix} = \text{ArgMin}(\varepsilon(x, y)) \quad (2.8)$$

En appliquant (2.7) à (2.6), on obtient:

$$\sum_{i=1}^m (P_i - Z_i(x, y)) \frac{\delta Z_i(x, y)}{\delta x} = 0 \quad (2.9)$$

$$\sum_{i=1}^m (P_i - Z_i(x, y)) \frac{\delta Z_i(x, y)}{\delta y} = 0 \quad (2.10)$$

ou encore sous la forme matricielle:

$$\begin{pmatrix} P_1 - Z_1(x, y) \\ P_2 - Z_2(x, y) \\ \vdots \\ P_m - Z_m(x, y) \end{pmatrix}^T \begin{pmatrix} \frac{\delta Z_1(x, y)}{\delta x} & \frac{\delta Z_1(x, y)}{\delta y} \\ \frac{\delta Z_2(x, y)}{\delta x} & \frac{\delta Z_2(x, y)}{\delta y} \\ \vdots & \vdots \\ \frac{\delta Z_m(x, y)}{\delta x} & \frac{\delta Z_m(x, y)}{\delta y} \end{pmatrix} = 0 \quad (2.11)$$

Les modèles de perte de trajets sont souvent représentés par une constante et une composante logarithmique [34, 52]. Il s'agit d'une réécriture de (2.5), si l'on ne considère pas l'effet du *shadowing*.

$$Z_i(x, y) = \beta_0 - \beta_1 \ln[(x - x_i)^2 + (y - y_i)^2] \quad (2.12)$$

où β_0 et β_1 sont des constantes propres au site. Ainsi:

$$\begin{cases} \frac{\delta Z_i(x, y)}{\delta x} = \frac{-2\beta_1(x - x_i)}{(x - x_i)^2 + (y - y_i)^2} \\ \frac{\delta Z_i(x, y)}{\delta y} = \frac{-2\beta_1(y - y_i)}{(x - x_i)^2 + (y - y_i)^2} \end{cases} \quad (2.13)$$

En remplaçant (2.13) et (2.12) dans (2.11) et supposant que $\beta_1 \neq 0$ on obtient [55]:

$$\begin{pmatrix} P_1 - \beta_0 + \beta_1 \ln[(x - x_1)^2 + (y - y_1)^2] \\ P_2 - \beta_0 + \beta_1 \ln[(x - x_2)^2 + (y - y_2)^2] \\ \vdots \\ P_m - \beta_0 + \beta_1 \ln[(x - x_m)^2 + (y - y_m)^2] \end{pmatrix}^T \cdot \begin{bmatrix} \frac{(x - x_1)}{(x - x_1)^2 + (y - y_1)^2} & \frac{(y - y_1)}{(x - x_1)^2 + (y - y_1)^2} \\ \frac{(x - x_2)}{(x - x_2)^2 + (y - y_2)^2} & \frac{(y - y_2)}{(x - x_2)^2 + (y - y_2)^2} \\ \vdots & \vdots \\ \frac{(x - x_m)}{(x - x_m)^2 + (y - y_m)^2} & \frac{(y - y_m)}{(x - x_m)^2 + (y - y_m)^2} \end{bmatrix} = 0 \quad (2.14)$$

2.4.1.1 Le voisin le plus proche ou *Closest Neighbor* (CN)

Dans cet algorithme, la station mobile compare un vecteur des RSS observés avec l'ensemble des empreintes disponibles dans la carte radio de référence et détecte le point de référence ayant la plus petite distance Euclidienne dans l'espace du signal [4]. Supposons que MS observe $O = [P_1 P_2 \dots P_m]$, la distance euclidienne entre ce vecteur et le $k^{\text{ième}}$ point de référence dans la carte est donnée par:

$$D_k = \left(\sum_i^m (P_i - Z_{ki})^2 \right)^{1/2} \quad (2.15)$$

Une autre variante de cet algorithme trouve les M points de référence les plus proches et estime la position en se basant sur la moyenne des coordonnées de ces points M [54].

2.4.1.2 L'estimation basée sur le maximum de vraisemblance

Cette technique est souvent employée pour affiner le résultat des estimées de distances par la méthode du voisin le plus proche [55].

2.4.2 Algorithmes de localisation basés sur le ToA

Pour des raisons de simplicité, on se limitera à la description de l'algorithme le plus utilisé.

2.4.2.1 Moindres carrés pour le ToA

La méthode des moindres carrés est un processus mathématique générique pour trouver une approximation de la solution d'un système d'équations où il y a plus d'équations que de variables. On tente de minimiser l'erreur qui dévie chaque équation du résultat approximé. Cet algorithme est généralement mis en œuvre sous forme matricielle, pour permettre la résolution d'un nombre quelconque d'équations. Une variante de cette technique est la méthode des moindres carrés pondérés ou *weighted least-squares*. On donne à chaque mesure de distance une pondération en fonction du niveau de confiance du récepteur en elle. Lorsqu'utilisée pour la localisation, cette méthode donne le poids le plus fort à la BS la plus proche de l'MS. Ceci est dû au fait que les erreurs liées aux trajets multiples par exemple, augmentent souvent avec la distance mesurée. L'erreur ε est calculée en utilisant l'équation suivante [56] :

$$\varepsilon = \delta - 2R_s d - 2Sx_s \quad (2.16)$$

$$\text{où } \delta \triangleq \begin{pmatrix} R_2^2 - d_{21}^2 \\ R_3^2 - d_{31}^2 \\ \vdots \\ R_N^2 - d_{N1}^2 \end{pmatrix}, \quad d \triangleq \begin{pmatrix} d_{21} \\ d_{31} \\ \vdots \\ d_{N1} \end{pmatrix}, \quad S \triangleq \begin{pmatrix} x_2 & y_2 & z_2 \\ x_3 & y_3 & z_3 \\ \vdots & \vdots & \vdots \\ x_N & y_N & z_N \end{pmatrix}$$

Cette équation d'erreur peut ensuite être utilisée dans l'une des nombreuses méthodes proposées par [57]. Les poids peuvent être appliqués à chacune des mesures sans beaucoup de difficulté, mais un certain arrangement est nécessaire pour trouver les poids appropriés. En [58], des références supplémentaires d'algorithmes des moindres carrés sont rapportées et qui utilisent un processus itératif à deux pas pour la résolution d'équations similaires.

Doherty [59] propose une méthode basée exclusivement sur les contraintes induites par la connectivité. Sachant que les communications d'égal à égal ou *peer to peer* dans le réseau sont modélisées comme un ensemble de contraintes géométriques sur les positions des nœuds. Les contraintes géométriques peuvent être de nature angulaire ou radiale. Le problème est résolu en utilisant un programme linéaire (LP) en évaluant toutes les contraintes géométriques. Il existe aussi l'algorithme de pondération résiduelle ou *residual weighting* RWGH qui est adapté pour les conditions de canal LoS, NLoS ou mixte [59].

Il importe de noter aussi que plusieurs combinaisons d'algorithmes dits hybrides ont été rapportées et permettent d'améliorer l'estimation de position.

2.5 Métriques de performances

Différents paramètres peuvent être utilisés pour décrire les aspects de qualité d'un système de positionnement. Tout d'abord, les paramètres importants sont ceux qui décrivent la configuration du réseau et de l'environnement. Deuxièmement, ceux qui sont directement liés à la localisation et les algorithmes utilisés. En se référant à l'étude de Hui *et al* [9], on dresse un comparatif des métriques de performances d'un système de localisation d'intérieur.

L'environnement se caractérise essentiellement par ses propriétés de propagation, sa densité, la distribution des nœuds et le degré de leur connectivité. La disponibilité d'une carte est aussi une propriété de l'environnement permettant le positionnement absolu alors que le positionnement relatif met l'accent uniquement sur la découverte de topologie.

2.5.1 Précision

La précision ou inversement l'erreur de localisation est le critère déterminant des systèmes de positionnement. Généralement, il s'agit de l'erreur moyenne sur la distance Euclidienne entre l'emplacement estimé et l'emplacement exact. La précision peut être considérée comme un biais potentiel ou un effet systématique ou encore un décalage du système de positionnement.

La précision constitue la première qualité du système. Mais, souvent, un certain compromis s'impose entre une précision «convenable» et d'autres caractéristiques.

Pour le cadre d'une application de localisation au sein d'un réseau de capteurs, il importe de noter que la taille de la base de données qui est tenue à jour à chaque nœud, la fréquence de mise à jour et la précision mathématique et numérique requises influencent la complexité de calcul et par conséquent la précision de positionnement [6]. Dans la littérature, on distingue de multiples estimateurs de position, à savoir:

- erreur circulaire probable (ECP) ;
- dilution géométrique de précision (DGP) ;
- erreur quadratique moyenne (EQM) ;
- borne inférieure de Cramer-Rao (BICR).

L'ECP est basée sur les variances de l'estimation de position dans les directions x et y . Cela donne une mesure globale de la précision de l'estimateur de position. DGP est une mesure de l'estimateur de performance en fonction de la position réelle du mobile par rapport aux stations de base. Chaque estimateur de performance de position peut être évalué en comparant son EQM à la BICR. Cette dernière représente la limite théorique de la variance de la sortie de l'estimateur. La définition et la formulation de chacune de ces mesures du rendement sont présentées ci-après.

2.5.1.1 Erreur circulaire probable

Si un estimateur est non biaisé, l'ECP décrit la distribution de l'estimé de position autour de la vraie position de la MS. ECP est définie comme le rayon d'un cercle autour du biais de l'estimateur de position et qui contient la moitié des estimations produites [60], comme l'illustre la Figure 2.8.

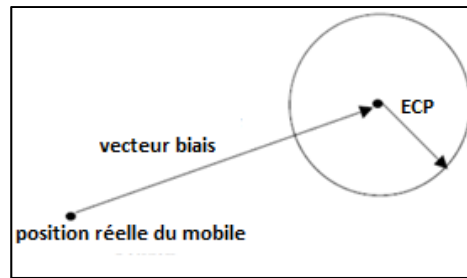


Figure 2.8 Erreur circulaire probable

L'ECP, à une précision de 10%, est donnée par [44] comme :

$$ECP \cong 0.75 \sqrt{\sigma_x^2 + \sigma_y^2} \quad (2.17)$$

où σ_x^2, σ_y^2 : *variances des estimées de position \hat{x}, \hat{y}*

2.5.1.2 Dilution géométrique de précision

On définit la dilution géométrique de précision DGP comme étant l'écart-type des mesures des distances [60]. Mathématiquement, elle est définie par le rapport entre la RMS de l'erreur de position par la RMS de l'erreur distance, comme suit:

$$DGP = \frac{\sqrt{\sigma_s^2 + \sigma_y^2}}{\sigma_s} \cong \frac{ECP}{0.75 \sigma_s} \quad (2.18)$$

De l'équation ci-dessus, il est évident que la DGP est directement proportionnelle à l'ECP, ce qui signifie que si la dispersion de l'estimation de position autour de la vraie position de la MS est petite, la variance de la sortie de l'estimateur sera elle aussi faible. Plus le RMS d'erreur de position est petit, meilleure est la performance de l'estimateur.

2.5.1.3 Erreur quadratique moyenne

L'EQM est le carré de la distance entre une vraie position (x, y) du mobile et une position estimée (\hat{x}, \hat{y}) . Mathématiquement, elle se définit comme :

$$EQM = \left(x - \hat{x}\right)^2 + \left(y - \hat{y}\right)^2 \quad (2.19)$$

$$\text{Moyenne quadratique (RMS)} = \sqrt{EQM}$$

2.5.1.4 Borne inférieure de Cramer-Rao

Afin de mesurer la précision de l'estimation de position, la RMS de l'erreur de position de l'estimateur peut être comparée à la limite théorique pour les estimateurs sans biais. Chan [61] a défini la BICR comme suit :

$$\Phi = c^2(G_t Q^{-1} G_t)^{-1} \quad (2.20)$$

où G_t : matrice des coefficients de Taylor évalués à une estimation initiale (x_o, y_o) ;

Q : matrice de covariance des mesures de distance corrigées servant d'entrée à l'estimateur ;

c : célérité de l'onde dans l'espace libre.

La limite théorique de l'erreur de position effective de l'estimateur est donnée par [60]:

$$RMS_{limit} = \sqrt{\text{trace}(\Phi)} \quad (2.21)$$

2.5.2 Exactitude

La précision ne considère que la valeur des erreurs moyennes sur la distance. Cependant, l'exactitude de positionnement traite de la constance avec laquelle le système fonctionne. En

effet, l'exactitude révèle la variation de rendement de ce dernier au cours de multiples observations et tests. Dans certaines littératures, l'exactitude de localisation est définie comme l'écart type de l'erreur de localisation ou de la dilution géométrique de précision (DGP). Dans ce travail, on considère l'exactitude comme la distribution de l'erreur entre l'estimée de position et la position exacte. Habituellement, les fonctions de probabilité cumulative (FPC) de l'erreur de distance décrivent la mesure de l'exactitude.

Lorsque deux techniques de positionnement sont comparées, si leurs précisions sont les mêmes, on opte pour le système avec le graphique FPC, qui atteint des valeurs de forte probabilité plus rapidement, parce que son erreur sur la distance est concentrée autour des petites valeurs. Dans la pratique, la FPC est décrite par le format percentile. Par exemple, soit un système ayant une précision de localisation de 90% dans les 2,3 m (la FPC d'erreur de distance de 2,3 m est de 0,9) et 95% à moins de 3,5 m. Un autre système a une précision de 50% dans les 2,3 m et 95% dans les 3.5 m. On pourrait choisir le premier système en raison de sa plus grande exactitude [9].

2.5.3 Robustesse

Une technique de positionnement robuste pourrait fonctionner normalement, même si certains signaux ne sont pas disponibles, ou lorsque certaines valeurs de RSS ou d'AoA ne figurent pas dans l'historique de valeurs enregistrées. Parfois, le signal provenant d'un émetteur est totalement bloqué, de telle sorte que le signal ne peut être obtenu qu'à partir d'autres stations. Parfois, certaines stations de base pourraient être hors de fonction ou endommagées dans un environnement difficile. Les systèmes de positionnement décrits comme robuste doivent utiliser cette information incomplète pour localiser correctement.

2.5.4 Complexité

La complexité d'un système de positionnement peut être attribuée à des facteurs matériels, logiciels ou de fonctionnement. On mettra plutôt l'accent sur la complexité des logiciels, c'est

à dire, la complexité de calcul de l'algorithme de positionnement. Si le calcul de l'algorithme de positionnement est réalisé du côté d'un serveur centralisé, le positionnement peut être calculé rapidement en raison de la puissance et de la capacité de traitement élevée de la machine. Si le calcul est effectué du côté du capteur de localisation, les effets de la complexité se feront remarquer.

La plupart des capteurs de localisation sont dénués de forte puissance de traitement ou d'une longue durée de batterie. Sur ce, on se tourne vers les algorithmes de positionnement de faible complexité. Il est généralement difficile de trouver une formule analytique de la complexité d'une technique de positionnement, d'autant plus que le temps de calcul doit y être considéré. La fréquence de localisation est aussi un indicateur important de la complexité. Le dual de la fréquence de localisation se présente comme le retard de localisation, qui est le délai entre qu'une cible mobile se déplace vers un nouvel endroit et que le système signale le nouvel emplacement. Dans les situations où l'on doit trouver l'erreur initiale de positionnement, le temps de première solution ou *time to first fix* qui est la durée avant laquelle une première estimation devient disponible, est un des aspects définissant la complexité d'un système. [6].

2.5.5 Évolutivité

Un système évolutif est un système qui assure son fonction de positionnement normalement lorsque sa portée ou son étendue devient grande. Habituellement, les performances de positionnement se dégradent lorsque la distance entre l'émetteur et le récepteur augmente. Un système de localisation doit s'échelonner sur deux axes: la géographie et la densité. L'échelle géographique signifie que la superficie ou le volume sont couverts. La densité signifie le nombre d'unités situées par zone géographique par période. Plus l'espace est couvert et les unités y sont amassées, plus les canaux sans fil deviennent encombrés, et de même pour le calcul nécessaire pour effectuer la localisation. Une autre mesure de l'évolutivité est la dimension de l'espace couvert. Certains systèmes permettent de localiser les objets en 2-D ou un espace 3-D, d'autres peuvent prendre en charge les deux.

2.5.6 Coût

Le coût d'un système de positionnement peut dépendre de plusieurs facteurs. Les facteurs importants comprennent l'argent, le temps, l'espace, le poids et l'énergie. Le facteur temps ici est lié à l'installation et la maintenance. D'autres littératures considèrent que le matériel, le stockage, les exigences de communication et la latence font aussi partie des coûts du système. La mesure de densité de l'espace et les contraintes de poids viennent aussi s'ajouter aux considérations de coût des capteurs de localisation. L'énergie, aussi, est un facteur important du coût d'un système. Certains capteurs comme les étiquettes de surveillance électronique d'articles (EAS) ou les étiquettes RFID passives réagissent à des champs extérieurs et, par conséquent, pourraient avoir une durée de vie quasi illimitée.

2.6 Sources d'erreurs

Après sa transmission, un signal traverse le milieu de propagation d'intérieur, qui introduit des déviations sur la direction d'arrivée et une dégradation de l'amplitude du signal. Sur son chemin, le signal rencontre de la dispersion naturelle et artificielle, des objets qui le rayonnent à leur tour et qui le dévient du chemin linéaire de propagation. Ceci dit, au niveau des systèmes d'émission et de réception, le signal peut rencontrer des erreurs instrumentales causées par des imperfections ou des limitations matérielles. Les systèmes de localisation sont donc sensibles à une grande variété de sources d'erreurs qu'on pourrait classer fondamentalement en deux catégories:

- les erreurs induites par la propagation et l'environnement ;
- les erreurs induites par les instruments et l'observation.

Les erreurs, provenant de causes indépendantes, sont cumulatives de point de vue statistique. Les statistiques d'erreur pour chaque catégorie d'erreur sont statistiquement combinées pour aboutir à une définition de l'erreur globale [5].

2.6.1 Les erreurs induites par la propagation

On liste à cet effet la perte de trajet (*pathloss*), les trajets multiples, le phénomène d'obstruction (*shadowing*) ainsi que l'évanouissement (*fading*). Ci-après sont décrits les effets des trajets multiples sur la mesure de distance.

2.6.1.1 Rayons des trajets multiples

Au moment où le premier bit d'un paquet arrive au récepteur, de nombreuses copies dues aux trajets multiples portant ce même paquet arrivent aussi. Ces rayons apparaissent en tant que pics de l'intensité du signal reçu, comme rapporté dans [62]. Le premier de ces pics correspond au trajet de ligne de vue directe ou LoS. Dans des conditions de multi trajets, un rayon secondaire peut être beaucoup plus fort que celui du LoS. Dans certains cas, la puissance du signal du trajet direct ou *direct path* (DP) passe en-dessous du seuil de détection et ne peut être détecté, on désigne ces conditions par UDP ou *Undetected Direct Path*. En tout état de cause, il est crucial que l'instant d'arrivée du premier rayon soit considéré et estimé avec précision [56]. Comme le premier rayon détectable d'un signal sans fil arrive au niveau du récepteur, l'amplitude du signal détecté augmente au-dessus d'un seuil de détection fixé à l'avance. Cette augmentation est suivie d'un pic, puis les deux autres rayons de trajets multiples et les autres bits. Le pic de ce signal est considéré comme étant l'instant où le rayon est arrivé. Toutefois, le signal n'a pas encore été décodé. Si un paquet n'a pas été reçu avec succès, il est probable que le bruit a aussi affecté l'horodatage (*timestamping*) et ainsi l'instant d'arrivé doit être rejeté. Les techniques de détection diffèrent par leur façon de déterminer l'instant où le premier rayon arrive et comment ils assurent qu'il commence à une transmission valide.

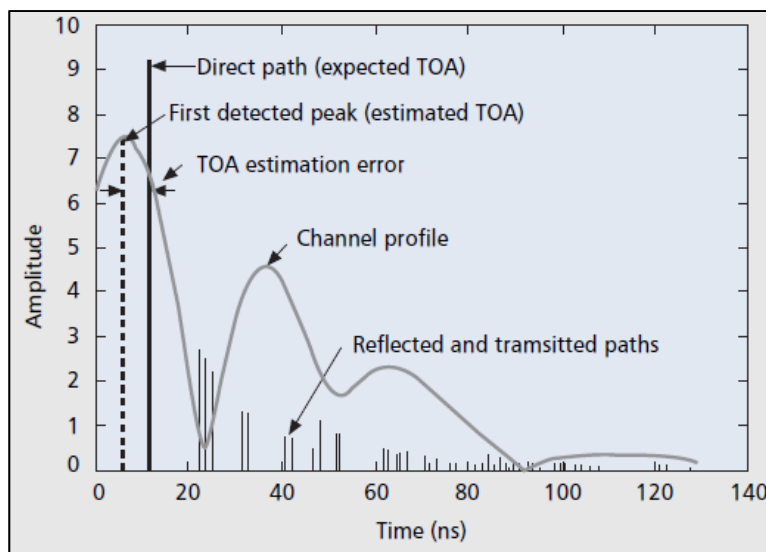


Figure 2.9 Effet des multi-trajets sur l'estimation du ToA du DP [66]

Dans la Figure 2.9, les lignes verticales pleines représentent la réponse impulsionnelle du canal idéal générée par un algorithme de *ray-tracing* pour deux emplacements arbitraires en environnement d'intérieur. Le trajet direct est également le trajet le plus fort *strongest path* (SP). Il correspond, dans ce cas, à l'estimée de ToA, qu'on utilise pour le calcul de la distance exacte entre l'émetteur et le récepteur.

2.6.1.2 Phénomène d'obscuration ou *shadowing*

Le *shadowing* est l'atténuation qui se produit lorsqu'un signal est obligé de passer autour des obstacles tels que les murs ou les arbres.

2.6.2 Les erreurs induites par les instruments et l'observation

Les erreurs instrumentales sont causées par des défauts, imperfections ou limitations des équipements. Parmi ces imperfections on note : le débalancement d'antennes, les diagrammes d'antennes distordus, les distorsions d'amplitude et les giges de phase, les imprécisions de temps et de fréquence, le non-alignement physique et le vieillissement des

instruments de mesure. On distingue aussi des erreurs d'observation qui sont dues à la faible précision de calcul du système de traitement ou de post-traitement des estimées. Les conversions de type de données comptent aussi parmi les causes de ces imprécisions.

À cet effet, une hypothèse de connaissance du site s'impose. Si le site de test ou de recherche est fixé à l'avance, un large historique des estimées de temps d'arrivée ou de puissance des signaux peut fournir des données quantitatives sur les effets éventuels d'erreur et procurer des paramètres de correction modélisant ce site [63].

2.6.2.1 Délai matériel

Même dans des conditions de parfaite synchronisation et en ligne de vue directe, le temps mesuré pour un paquet se propageant depuis un nœud à un autre ne reflète pas nécessairement la distance réelle entre ces nœuds. Ceci est dû aux délais de traitement au sein des dispositifs radio. Pour qu'un paquet soit démodulé, il doit tout d'abord se propager depuis le bout d'une antenne jusqu'à l'instant où l'on lui assigne un *timestamp*.

2.6.2.2 Limitations de la bande passante

Dans les systèmes de géolocalisation d'intérieur traditionnels, on utilise le premier pic détecté ou *first detected peak* (FDP) du profil du canal, reçu au-dessus du seuil de détection pour estimer le ToA du DP. Dans un environnement de chemin unique, le DP estimé concorde avec le DP réel. On désigne ici l'erreur de distance causée par une estimation erronée de la ToA par l'erreur de mesure de distance EMD. Dans les conditions d'UDP, le premier pic détecté est indépendant de l'instant d'arrivée du DP, causant ainsi une grande EMD [66].

À la Figure 2.10, on montre un exemple d'occurrence de cette erreur. Elle est due aux conditions UDP. L'exemple est pris à partir des résultats de *ray-tracing* pour une impulsion émise avec une bande passante de 200 MHz. La différence entre la puissance du signal le plus fort et celle du trajet direct est supérieure à la plage dynamique du récepteur. Cela

résulte en une EMD de 5.23 mètres. On désigne ici par plage dynamique d'un récepteur, la plage des signaux reçus en dessous du trajet de plus forte amplitude SP.

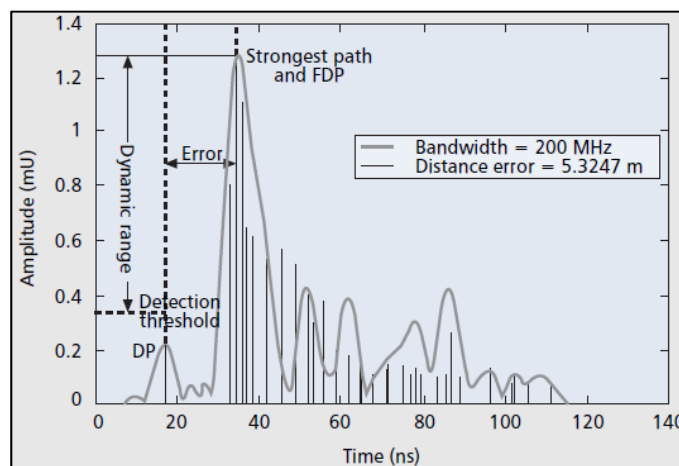


Figure 2.10 Erreur de distance dans les conditions UDP [66]

D'autres trajets arrivent après la DP suite à un certain nombre de réflexions et de transmissions. Ils arrivent avec des amplitudes plus faibles. Si la largeur de bande du système était infinie et que tous les chemins peuvent être résolus, les chemins générés par les algorithmes de *ray-tracing* auraient tous été observés au niveau du récepteur. Dans la pratique, la bande passante est limitée et le signal reçu est formé par l'addition de toutes les formes d'impulsion tels que représenté à la Figure 2.11.

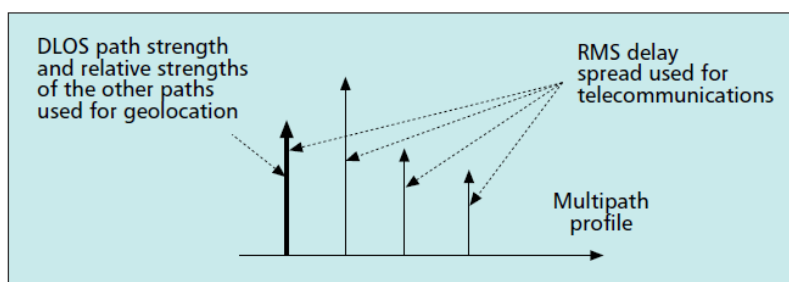


Figure 2.11 Profil du canal multi-trajets [2]

À mesure que la largeur de bande de transmission du système augmente, les impulsions en provenance de différents chemins deviennent plus étroites, et l'estimation de la ToA par le FDP devient plus proche de la ToA prévue par la DP, résultant en une plus petite EMD. Par conséquent, la première cause d'erreurs de distance dans les canaux à trajets multiples est la proximité des composantes de trajets multiples de la DP. L'augmentation de la bande passante du système, donne la possibilité de contrôler ces erreurs.

2.7 Conclusion

Ce chapitre passe en revue les différents systèmes et techniques de positionnement d'intérieur. Nombreux de critères et métriques sont discutés et plusieurs compromis entre eux sont observés. Notons par exemple, le compromis entre la complexité et la précision qui exige un examen attentif lorsque l'on choisit les systèmes de positionnement et les techniques pour différents milieux. Une fois appliqués dans des environnements réels, ces techniques et systèmes ont leurs caractéristiques importantes à considérer telles que la largeur de bande et les techniques de résolution des délais. Le choix de la technique et de la technologie affecte significativement la granularité et de l'exactitude de l'information de localisation. L'état de l'art de la localisation étant abordé et examiné, on passe au chapitre suivant à la définition de la plateforme matérielle et logicielle de la radio définie par logicielle dont on a fait usage.

CHAPITRE 3

LES RADIOS DÉFINIES PAR LOGICIEL

3.1 Introduction

L'environnement de développement GNU-Radio ainsi que la radio définie par logiciel USRP forment les éléments de base de ce travail de recherche. Dans les sections qui suivent, on donne un bref historique et descriptif de ses deux composantes.

3.2 Les radios définies par logiciels

Les radios traditionnelles sont généralement constituées d'un émetteur-récepteur en circuits intégrés élaboré en utilisant du matériel dédié. Les composants principalement utilisés sont des mélangeurs, des amplificateurs, des filtres, des modulateurs/démodulateurs et des convertisseurs. Même si la prolifération de la consommation du grand public des téléviseurs et des téléphones mobiles a abouti à une implémentation omniprésente des radios traditionnelles, un nouveau matériel a dû être développé pour chacune des variations [67].

À l'inverse de cette approche un concept novateur est apparu, la radio définie par logiciel ou "*Software Defined Radio*" (SDR). C'est un système de communication radio structurellement différent des radios traditionnelles tel qu'illustré à la Figure 3.1. Il implémente la plupart de ses fonctionnalités par logiciel au lieu d'une implémentation matérielle fixe. Pour définir la SDR, Eric Blossom, fondateur du projet GNU-Radio, cite : « Les radios définies par logiciel est la technique d'obtenir le code le plus près de l'antenne que possible. Elles transforment les problèmes matériels en des problèmes logiciels. » [73].

Contrairement aux radios traditionnelles, dont la modification est coûteuse et les fonctionnalités restent limitées à une configuration matérielle spécifique, la radio logicielle, elle, se base sur des composantes à moindre coût configurables par logiciel. Parmi ces composantes, on cite les processeurs numériques de signal (DSP), les *Field Programmable*

Gate Array (FPGA) et les processeurs à usage général (GPP). Maintenant que la vitesse des processeurs des ordinateurs est devenue plus rapide, le concept jadis théorique de radio définie par logiciel est devenu une réalité pratique [68]. En effet, le même processeur utilisé dans les ordinateurs personnels est employé par les SDR pour exécuter du traitement de signal. Un logiciel commun de conception de radio est défini pour mettre en œuvre le traitement de signal RF de haute fréquence sur un matériel analogique et le traitement des autres étages c'est-à-dire en fréquence intermédiaire et en bande de base sur du logiciel.

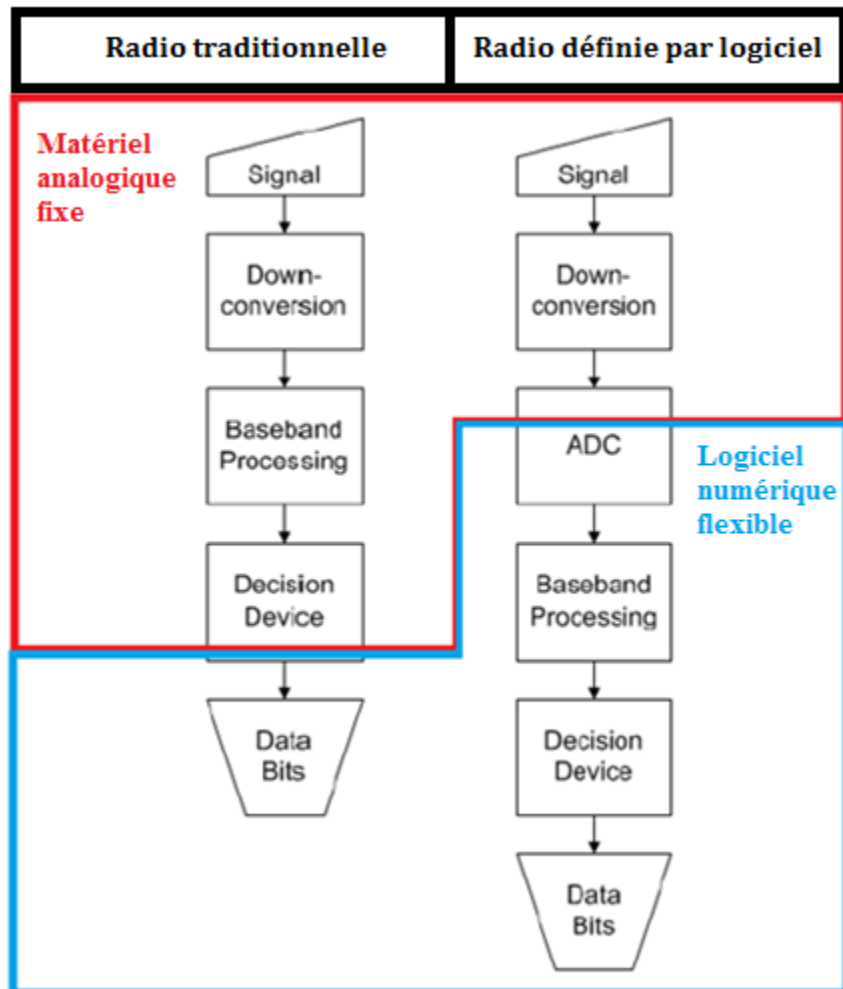


Figure 3.1 Diagramme comparatif entre les radios traditionnelles et les SDR [56]

De nombreux avantages de flexibilité sont obtenus en utilisant cette approche logicielle. Pour commencer notons la réduction significative du temps requis pour déployer ou mettre à jour de nouveaux produits. En effet, les dispositifs SDR déjà déployés peuvent simplement être reprogrammés pour profiter des normes émergentes et des nouvelles fonctionnalités. C'est pour cet effet qu'on décrit la SDR comme un outil de prototypage rapide des nouvelles applications avec laquelle on peut tester toutes sortes de variations et de configurations.

En outre, la SDR est une technologie clé pour la radio cognitive, l'accès opportuniste au spectre et l'auto-configuration des radios pour s'adapter aux conditions variables du canal [69]. La SDR a aussi l'avantage de fonctionner à différentes fréquences. Selon Blossom, un avantage important pour l'industrie est la possibilité de modifier des piles entières de traitement ou de schémas de modulation juste par une mise à jour logicielle [70].

L'un des inconvénients de ces radios est la taille importante des ordinateurs dont le rôle consiste à effectuer le traitement du signal. Ces ordinateurs sont beaucoup plus volumineux que les systèmes embarqués habituels. Ceci dit, selon [70], l'inconvénient majeur dans ce contexte concerne la consommation élevée de puissance. En effet, la partie logicielle dans un tel système nécessite d'énormes capacités de calcul d'où le caractère énergivore. En termes de chiffres, on doit différencier si la radio est utilisée pour une application à bande étroite ou à large bande. Pour les applications à bande étroite, un processeur Pentium régulier suffit pour répondre aux exigences. Toutefois, si l'application à mettre en œuvre utilise une large partie du spectre comme par exemple un récepteur de multiples canaux FM, on aura besoin d'ordinateurs puissants afin de traiter toutes les données fournis dans le temps imparti. C'est pourquoi peu de dispositifs portatifs basés sur les SDR ont été commercialisés. Par contre, la SDR a plutôt été ciblée pour développer les applications de type station de base, où les besoins actuels en énergie ne sont pas aussi critiques, comparé au terminal mobile [70].

3.3 Architecture matérielle

Historiquement, le développement des SDR sur une plateforme matérielle libre s'est organisé autour des systèmes USRP (*Universal Software Radio Peripheral*) produit par la compagnie Ettus Research sous la direction d'Eric Blossom et Matt Ettus [46]. Ses radios se caractérisent par un coût raisonnable, en comparaison aux autres systèmes proposés dans le marché. C'est pour cette raison qu'on les décrit comme des dispositifs *Commercially off the shelf* [70]. En effet, on pourrait relier l'avènement de ces radios à faible coût à la chute vertigineuse des coûts de composants RF développés notamment pour les applications de communication numérique sans fil tels que les synthétiseurs de fréquence, les modulateurs et démodulateurs I/Q sur des bandes allant de quelques MHz aux GHz offerts notamment par Analog Devices, Maxim ou Semtech [71].

À la Figure 3.2, on présente la structure générique en blocs d'une radio logicielle avec ses deux parties matérielle et logicielle.

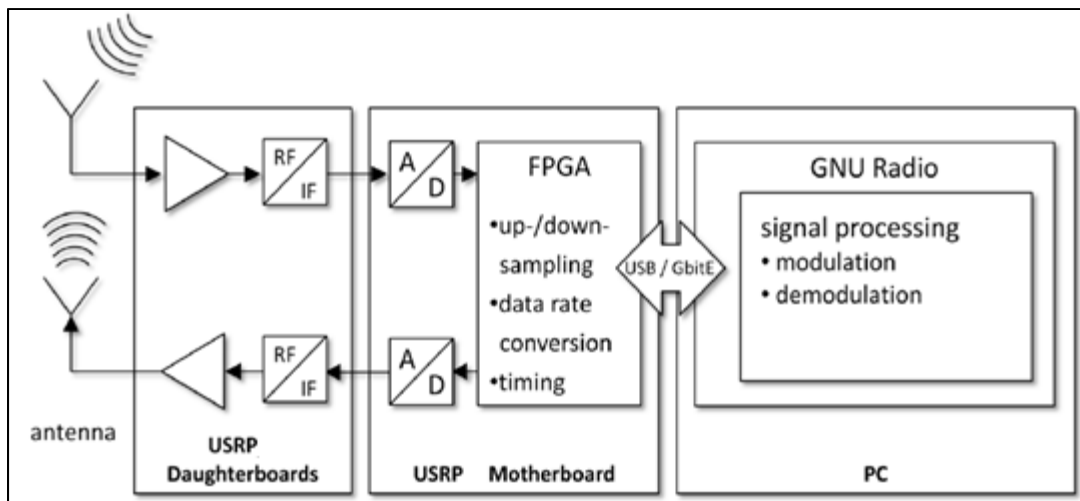


Figure 3.2 Structure générique en blocs d'une radio définie par logiciel [73]

Tout au long de ce projet de recherche, on utilise la dernière version d'USRP qui est la N210. Un comparatif des caractéristiques des radios USRP est dressé au Tableau 3.1.

Tableau 3.1 Comparatif des caractéristiques des USRP d’Ettus

	USRP 1	USRP 2	USRP N210	USRP B100	USRP E100
Nbr. de canaux	4 / 4	2 / 2	2 / 2	2 / 2	2 / 2
Résolutions CAN/CNA (bits)	12 / 14	14 / 16	14 / 16	12 / 14	12 / 14
Largeur de bande (M.éch./s) (USRP/hôte)	8	25 (16-bits) 50 (8-bits)	25 (16-bits) 50 (8-bits)	8 (16 bits) 16 (8 bits)	N/A
SFDR CAN/CNA (dBc)	85 / 83	88 / 80	88 / 80	83 / 83	85 / 83
Interface hôte	USB2	Gbit Eth.	Gbit Eth.	USB2	10/100 Eth. UART
Horloge externe / PPS	Non	10 MHz + 1PPS	5 ou 10 MHz + 1PPS	10 MHz + 1PPS	Non
Horloge interne	64 MHz / +/- 20 ppm	100 MHz / +/- 2.5 ppm	100 MHz / +/- 2.5 ppm	100 MHz / +/- 2.5 ppm	100 MHz / +/- 2.5 ppm
Processeur	µcontrôleur	Logiciel	Logiciel	Logiciel	Embarqué
FPGA	Altera Cyclone EP1C12	Spartan XC3S2000	Spartan 3A-DSP3400	Spartan 3A-1400	Spartan 3A-DSP 3400

Il importe de noter ici que la version USRP2 des radios a été abandonnée en faveur de la série N d’USRP. ‘N’, ici, fait référence au réseau Ethernet liant l’USRP et l’hôte. Les radios USRP N210 possède plus de ressources FPGA que la version USRP N200 [71].

3.3.1 Carte mère USRP N210

Le dispositif utilisé, dont on présente le diagramme bloc à la Figure 3.3, se compose d’une interface Gigabit Ethernet permettant de communiquer avec l’ordinateur hôte ainsi qu’une carte FPGA pour le traitement de signal à virgule flottante à grande cadence. L’USRP N210

est compatible avec une large gamme de modules RF appelées cartes filles (section 3.3.2). En mode réception, les échantillons récupérés par la carte fille sont numérisés par le convertisseur analogique-numérique (CAN), ensuite convertis à la fréquence IF désirée. Finalement, la séquence d'échantillons est multipliée par un sinus et un cosinus pour obtenir respectivement les deux composantes I et Q.

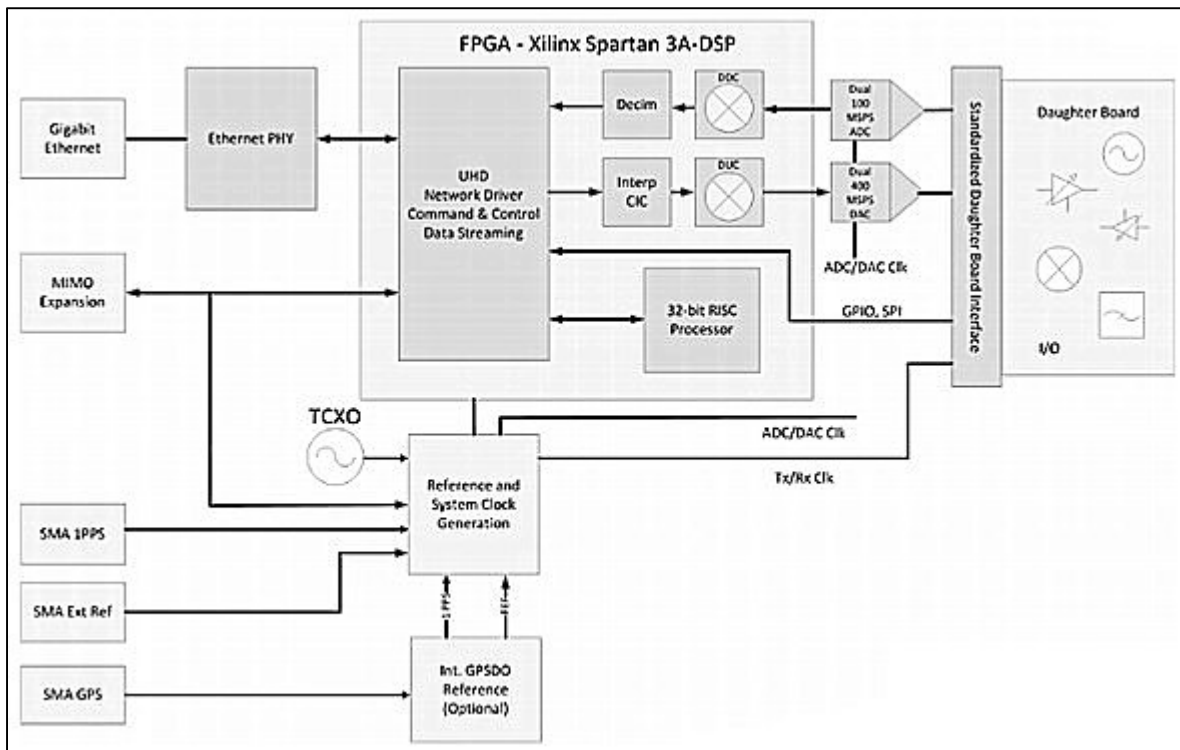


Figure 3.3 Diagramme bloc de la carte mère USRP N210 [71]

La fréquence est générée par un oscillateur commandé numériquement ou NCO qui synthétise une forme d'onde d'amplitude discrète et en temps discret au sein du FPGA. Après cela, une décimation par un facteur arbitraire N du taux d'échantillonnage (f_s) est effectuée donnant des échantillons qui seront envoyés à l'hôte avec un débit réduit. En émission, la procédure inverse est effectuée en utilisant des convertisseurs numériques ascendant ou "Digital Up Converter" (DUC) et un convertisseur numérique-analogique (CNA). Pour la liaison avec l'hôte, l'USRP N210 utilise la connexion Gigabit Ethernet, qui permet un débit significativement plus élevé que ce que l'USRP 1 offre via l'USB 2.0. La connexion USB,

elle, soutient 32 Mo/s en mode semi-duplex, de sorte que la transmission et la réception des échantillons n'est pas possible de manière synchrone. En utilisant des échantillons complexes à 4 octets (16 bits I et 16 bits Q) et en respectant le critère de Nyquist, on aboutit à une largeur de bande utilisable de 8 MHz. En contrepartie, l'USRP N210 permet avec l'Ethernet Gbit, un débit théorique de données de 1 Gbit/s soit 31.25 MHz pour 32 bits par échantillon complexe. Une valeur de 25 MHz de bande passante utilisable, tel qu'indiquée, peut servir de limite réaliste lorsque l'on considère les informations de gestion des paquets Ethernet ou "overhead" [71].

$$\text{débit théorique}_{\text{GigaBit_Eth.}} = 1 \text{ Gb/s} / 32 \text{ bits} = 31.25 \text{ MHz} \quad (3.1)$$

$$\text{débit utile}_{\text{GigaBit_Eth.}} = 1 \text{ Gb/s} / (32 \text{ bits} + 8 \text{ bits overhead}) = 25 \text{ MHz} \quad (3.2)$$

Tous les composants de la carte mère de l'USRP sont liés au FPGA comme le montre la Figure 3.3. Ils sont cadencés à la fréquence de l'horloge maitresse de la carte mère qui est de 100 millions de cycles par seconde (MHz). La fréquence d'échantillonnage du CAN définit combien de fois par seconde le signal analogique peut être mesuré. Pour le cas de l'USRP N210, il s'agit de 100 Méch/s. Une fois numérisé, le flux de bits du signal est passé à travers le FPGA et en paquets vers l'hôte à travers l'Ethernet.

3.3.2 Carte fille

La carte mère est connectée à une carte fille lui permettant l'émission et la réception des signaux RF analogiques. Sur ses cartes filles, le signal est filtré, amplifié et mis sur une fréquence de bande de base qui dépend de la bande IF de la carte et de la fréquence de l'oscillateur local. Parmi ces cartes, on compte aussi les cartes Rx/Tx de base sans conversion de fréquence ou filtrage. Elles ne fournissent qu'une connexion RF directe à la carte mère. Tandis que la carte mère effectue les conversions analogique/numérique, l'échantillonnage décimation/interpolation, et l'interfaçage avec l'hôte, les cartes filles, elles, contiennent des têtes RF fixes ou des interfaces directes aux convertisseurs de la carte mère.

Dans le présent travail, la carte fille utilisée doit couvrir la bande d'intérêt, comme indiqué par les objectifs du projet, c'est-à-dire la bande ISM 2.4GHz. Les trois cartes filles couvrant

cette bande sont: RFX2400, XCVR2450 et SBX. Ces cartes peuvent opérer en mode duplex intégral pour une émission et une réception simultanées. Elles offrent une largeur de bande allant jusqu'à 50 MHz. Leurs caractéristiques sont résumées au Tableau 3.2.

Tableau 3.2 Comparatif des cartes filles utilisées

	Bande (MHz)	$P_{\max} \text{Tx}$ (mW)	Synthétiseurs	Caractéristiques
RFX2400	2300-2900	50	Double pour les fréquences indépendantes Tx et Rx	ISM 2.4 GHz
XCVR2450	2400-2500 4900-5900	100	1	ISM 2.4GHz et 4.9 GHz, Sécurité publique, UNII, sans fil japonais
SBX	400- 4400	100	2	Cellulaire, WiFi, S et ISM 2.4

À la Figure 3.4 on présente le modèle en blocs de la carte RFX2400, la plus utilisée durant ce projet. On distingue les convertisseurs ascendants DUC, les convertisseurs descendants DDC, les oscillateurs locaux de synthèse de fréquences et les filtres [71].

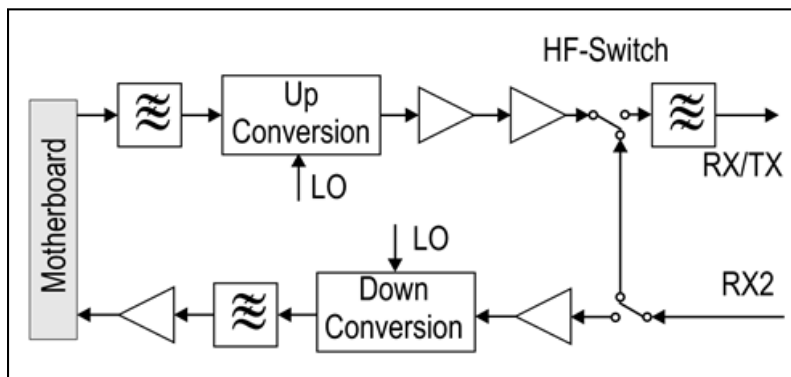


Figure 3.4 Diagramme bloc de la carte fille RFX-2400 [71]

3.3.3 Antennes

Au cours de ces travaux, l'antenne utilisée est la *Tri-Band Hyperlink sans fil TNC-Male* modèle HG2458RD-SM illustrée à la Figure 3.5. C'est une antenne omnidirectionnelle en caoutchouc de large couverture. Elle utilise une polarisation verticale et est reliée à l'USRP via un connecteur SMA. En raison de la nature inclinable et pivotant des connecteurs, elle peut être déployée verticalement, horizontalement ou à un angle entre les deux.



Figure 3.5 Antenne utilisée au cours de ce projet

Au Tableau 3.3, on présente les caractéristiques électriques de cette antenne.

Tableau 3.3 Caractéristiques électriques de l'antenne utilisée [77]

Caractéristique	Valeur
Gammes de fréquences	2400-2500 MHz 4900-5350 MHz 5725-5850MHz
Gain	3dBi
Largeur horizontale du faisceau	360°
Impédance	50 Ohm
Puissance Maximale	50W

Les diagrammes de rayonnement de l'antenne à 3 dBi de gain sont présentés à la Figure 3.6.

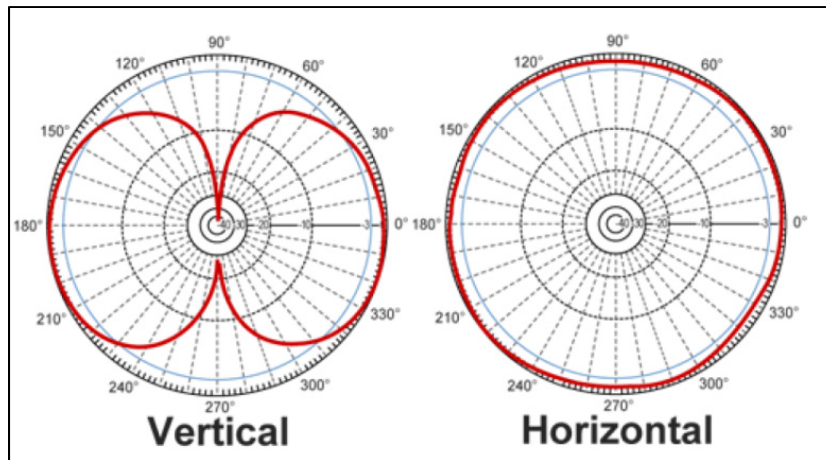


Figure 3.6 Diagrammes de rayonnement de l'antenne utilisée [77]

L'USRP, les cartes filles ainsi que les autres éléments connexes forment une plateforme matérielle entièrement libre pour le développement et pouvant aussi être prise en charge entièrement avec un logiciel libre. Les schématiques des composantes de cette plateforme sont fournis aux utilisateurs sur le site web d' Ettus [71]. L'USRP est vendu comme équipement de test mais le constructeur laisse à l'utilisateur la responsabilité du respect de la réglementation du spectre radio.

3.4 Architecture logicielle

3.4.1 Framework GNU-Radio

Le projet GNU-Radio consiste en une boîte à outils de logiciels pour le développement d'applications SDR. Le projet a été fondé en 1998 par Eric Blossom. En 2001, il a été introduit par ce dernier avec John Gillmore. GNU-Radio est un environnement libre, ouvert, gratuit et est distribué sous la licence GNU General Public Licence. Le projet est utilisé par une grande variété d'organismes universitaires, gouvernementaux et commerciaux, par des chercheurs ainsi que des radioamateurs. Il fournit différents blocs de traitement de signal préassemblés pour la création de forme d'onde et l'analyse de ses applications [73]. Le système d'exploitation recommandé pour l'installation de GNU-Radio est Linux, mais il peut

aussi être installé sous MS Windows en utilisant l'un des émulateurs d'environnements Linux, tel que Cygwin ou MinGW, ainsi que sur MAC OS et NetBSD. Au cours des quatre années suivant sa création, GNU-Radio est devenu une multiplateforme largement utilisée. Une partie de son succès dérive du concept d'abstraction des blocs qui permet aux développeurs de manipuler les signaux en ajoutant une série d'événements individuels de traitement de signal. L'environnement de programmation est basé sur un système d'exécution constitué par un diagramme et des blocs de traitement. Ce diagramme décrit le flux de données dans tout le système GNU-Radio. Il est habituellement implémenté en utilisant le langage orienté objet Python. Les blocs de traitement de signal sont des entités fonctionnelles implémentées en C++ et son plus exigeants du point de vue computationnel. Ils opèrent sur les flux découlant d'un certain nombre de ports d'entrées à un certain nombre de ports de sorties. L'outil SWIG (*Simplified Wrapper and Interface Generator*) est utilisé pour créer des fonctions de connexions (*wrappers*) avec Python autour des blocs C++. GNU-Radio est donc caractérisé par une structure en couches qu'on présente à la Figure 3.7.

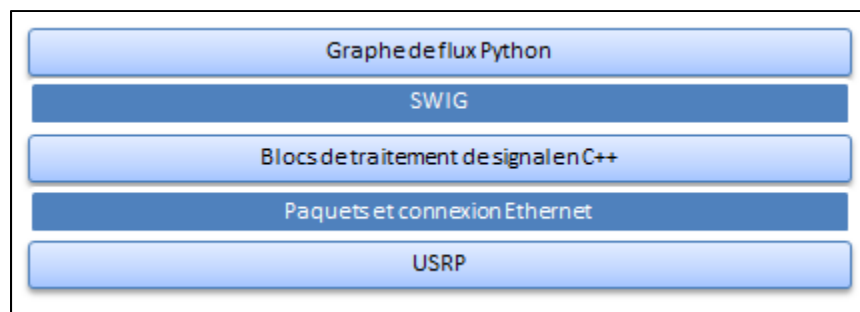


Figure 3.7 Structure en couche du *framework* GNU-Radio

L'environnement GNU-Radio permet aussi son intégration avec le système d'exploitation Linux afin d'en utiliser les services notamment *Linux standard pipeline* ou *Inter-Process Communication* (IPC). Une couche d'abstraction matérielle fournit un support pour les pilotes et pour la gestion des ressources matérielles comme l'illustre la Figure 3.8.

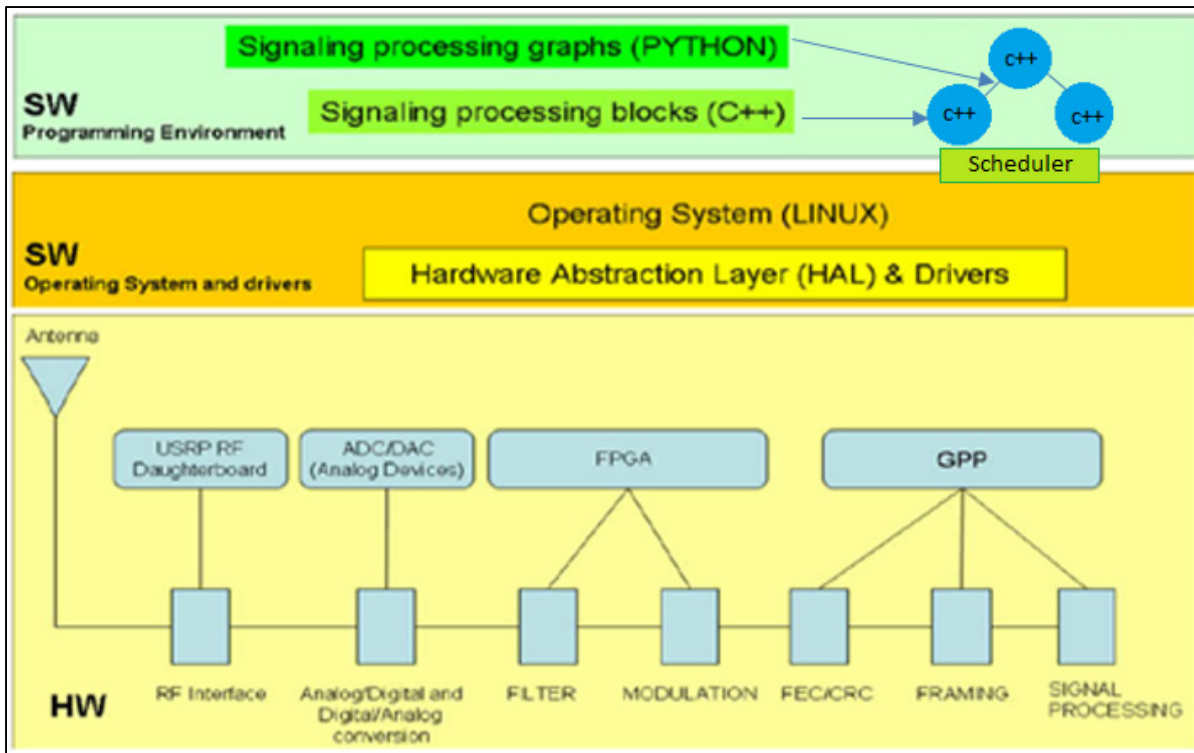


Figure 3.8 Architecture hybride de la SDR incluant logiciel et matériel [75]

GNU-Radio offre une importante et croissante bibliothèque logicielle de routines individuelles ou des blocs entiers de traitement de signal. Le système d'exécution prévoit l'allocation dynamique de tampons et prévoit aussi l'ordonnancement en fonction des taux fixes ou dynamiques des Entrées/Sorties des blocs. Le fonctionnement de l'ordonnanceur est décrit à la section 3.4.2.2. L'application GNU Radio consiste la plupart du temps en un diagramme de flux comme le montre la Figure 3.9. GNU Radio est doté d'une interface graphique appelée *GNU Radio Companion* ou GRC qui apporte plus de simplifications, même si les applications GNU-Radio sont implémentées en utilisant différents langages de haut niveau. Cet outil graphique GRC est basé sur une conception orientée blocs. Il met en exergue la modularité du système. Une fois qu'un diagramme bloc conçu par GRC, est lancé, le diagramme de flux correspondant est compilé. Un code source Python est ensuite généré et une exécution séquentielle est effectuée.

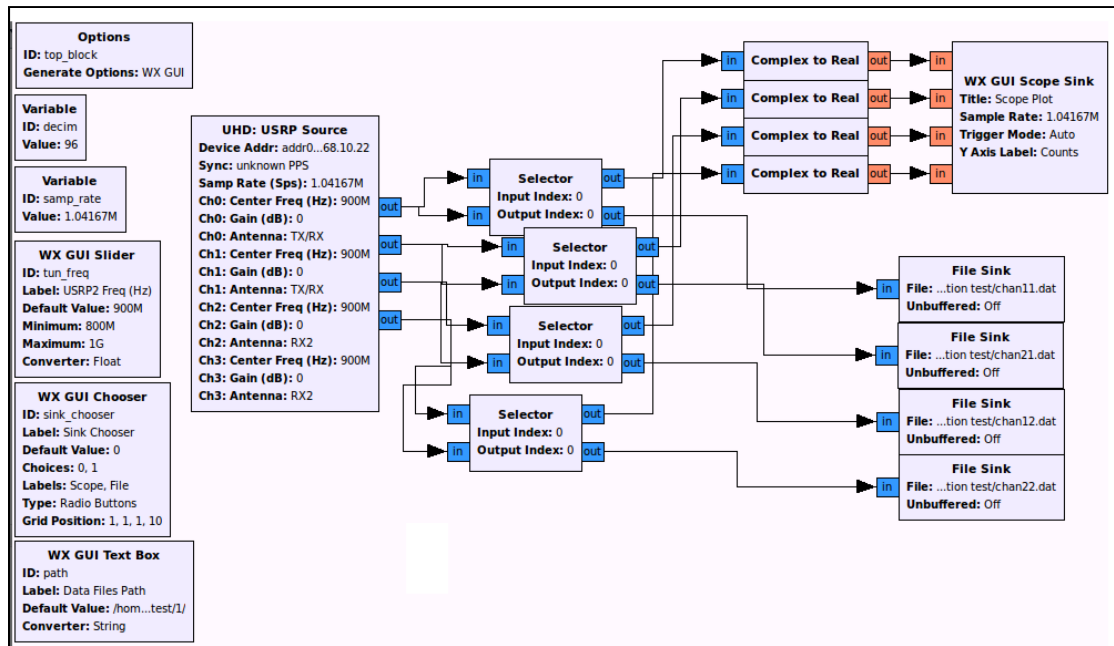


Figure 3.9 Diagramme d'une application GNU-Radio interfacée avec l'USRP

Cette architecture permet aux développeurs d'écrire du code hautement optimisé en C++ et de faciliter la réutilisation des blocs à l'aide du langage Python qui est beaucoup plus convivial. Conceptuellement, les blocs de traitement sont mis en cascade en une chaîne de traitement ou diagramme de flux de signal qui élabore un flux fini ou infini de données en bande de base. Ces ports d'Entrées/Sorties sont par la suite interfacé avec le matériel du monde réel, ce qui convertit et déplace les signaux de bande de base à passe-bande.

Bien que communément utilisée avec l'USRP, la GNU-Radio peut également être utilisée sans interfaçage avec le matériel comme un logiciel autonome ou *standalone* comme le montre la Figure 3.10. De même, la GNU-Radio soutient également le développement d'algorithmes de traitement du signal à partir de données enregistrées à l'avance.

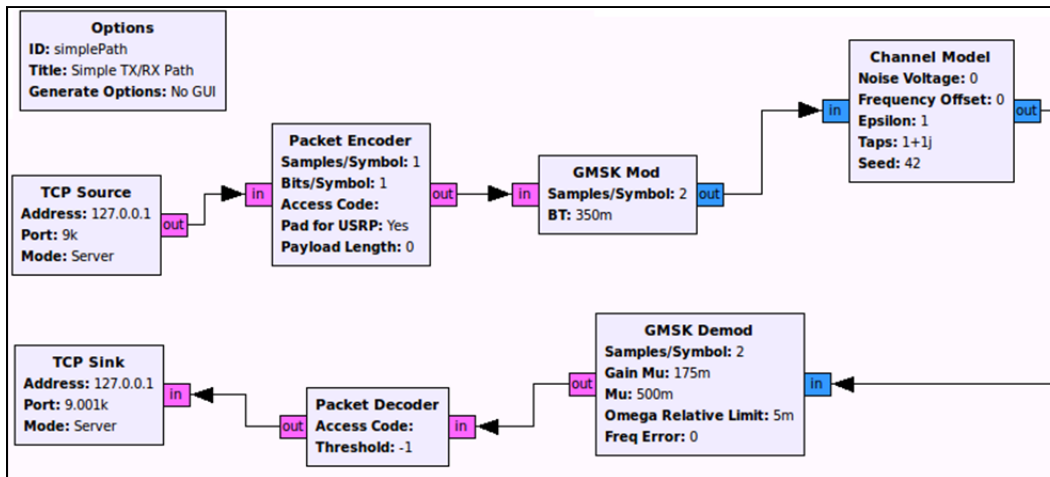


Figure 3.10 Diagramme d'une application GNU-Radio sans connexion RF

Comme avec l'outil de conception Simulink, les modules dans GRC sont insérés et connectés pour former le diagramme de l'application. GRC est doté d'une liste de modules prêts à utiliser. L'utilisateur peut aussi étendre cette liste en y ajoutant ses propres modules personnalisés. Les paramètres de chaque module sont définis dans un fichier XML que l'interface GRC utilise pour évaluer la validité des paramètres de configuration dans le diagramme. Alors que le code source de la GNU-Radio est entièrement ouvert "*open source*" et ayant une solide base d'utilisateurs, il reste en développement continu. La documentation relative à GNU-Radio se compose de quelques tutoriels, une API, des archives de courriels et commentaires de code. Tout support à l'utilisateur final est réalisé par des bénévoles. D'importants bogues sont régulièrement découverts et corrigés, tandis que la rétrocompatibilité est souvent rompue [20].

3.4.2 Pilotes des radios, ordonnancement

3.4.2.1 Les pilotes UHD

UHD réfère à *USRP Hardware Driver*. C'est l'ensemble des pilotes pour le support des radios USRP. UHD intègre plusieurs fonctionnalités à GNU-Radio, parmi lesquelles on note:

les blocs sources pour la connexion d'une USRP en mode récepteur ainsi que ses routines *work()* et *recv()*. On note en contrepartie, les blocs destinations ou *sink* pour la connexion d'une USRP en mode émetteur et ses routines *work()* et *send()*.

Comme le montre la Figure 3.11, l'UHD permet la manipulation des aspects matériels des USRP à savoir les multicanaux, l'alignement des échantillons et la synchronisation temporelle [71]. Les pilotes des USRP donnent la main à l'utilisateur pour configurer la radio. En effet, l'utilisateur peut configurer ou lire le gain, configurer la fréquence centrale, lire les valeurs des capteurs embarqués sur la carte fille ou la carte mère, vérifier le verrouillage de l'oscillateur local de la carte fille RF, configurer ou lire les valeurs de synchronisation, la fréquence d'échantillonnage, sélectionner l'antenne, la tête RF ou l'interface de flux. L'UHD permet la découverte de dispositifs connectés au système, l'instanciation de périphérique ainsi que l'envoi et la réception des échantillons.

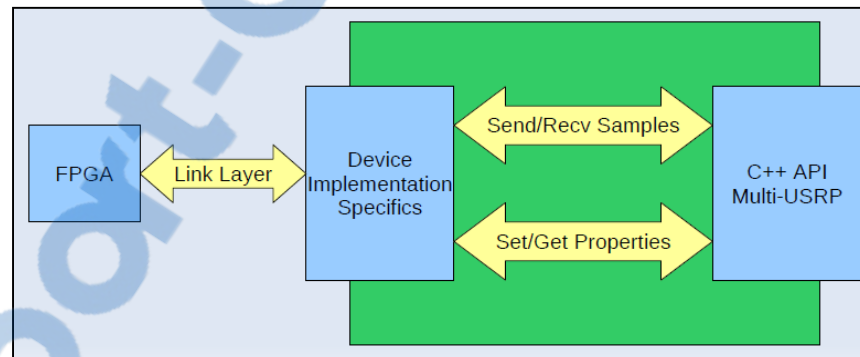


Figure 3.11 Fonctionnalités apportées par les pilotes UHD [71]

UHD est un requis indispensable pour l'interfaçage avec les USRP et les cartes filles sur les plateformes GNU-Radio, MATLAB, Simulink et LabVIEW. UHD donne à l'USRP la capacité de transmettre ou de recevoir à un moment précis si on connecte à bord de l'USRP une carte supplémentaire appelée GPSDO ou si l'ordinateur hôte est synchronisé à un récepteur GPS externe.

Par opposition à la version obsolète USRP 2, la version utilisée dans ce projet USRP N210 fonctionne via les pilotes UHD. Avant, l'USRP 2 utilisait l'interface Ethernet "brute" et USRP 1 utilisait l'interface USB 2.0. Parmi les avantages des nouveaux pilotes UHD on cite : la neutralité de plateforme quant à l'USRP utilisé d'où le terme "Universal". En effet, ces pilotes apportent une couche d'abstraction plus cohérente pour les systèmes basés sur la SDR et à la fin permettent à ces dispositifs d'être «plus reconnu» dans le monde de la technologie SDR [79]. UHD est doté d'une API C++ unique pour tous les USRP. Il a aussi introduit la possibilité de développer sous Linux mais aussi sur Mac et Windows. Les possibilités de compilations ont été étendues pour couvrir à la fois le compilateur GNU ou GCC sous tous les OS, Clang, MSVC et Cmake pour une compilation multi plateformes. UHD permet la génération d'un système de compilation natif. Il offre les fonctionnalités des bibliothèques Boost, Cross platform C++, ASIO ainsi que les tests unitaires et les options de programme qui facilitent l'exécution de tests multiples.

Cependant, au moment de rédaction de ce mémoire, UHD est encore en développement. Beaucoup de fonctions et de projets GNU-Radio compatibles avec les anciens pilotes n'étaient pas encore traduits pour se conformer à UHD. Cela a représenté pour cette recherche un facteur d'accroissement de la courbe d'apprentissage de GNU-Radio et ses différents environnements de développement connexes. De même, au début de ce projet, les outils Matlab et Simulink ne supportaient pas les fonctionnalités d'UHD. Les fonctionnalités permettant de soutenir l'outil Labview étaient encore en version expérimentale.

3.4.2.2 L'ordonnanceur

Un ordonnanceur est utilisé par GNU-Radio pour allouer les ressources de calcul entre les blocs au sein du diagramme de flux à l'exécution. Il supporte les modifications de diagramme de signaux et la reconfigurabilité en temps réel. Il existe deux types d'ordonnanceur: l'ordonnanceur à un fil d'instruction unique pour tout le diagramme de flux appelé "*single threaded scheduler*" et l'ordonnanceur à un fil d'instruction par bloc ou "*thread-per-block*" (TPB). Le premier type d'ordonnanceur alloue un seul fil d'instruction. GNU-Radio découpe

chaque flux de données en morceaux et alimentent chaque bloc du diagramme, un à la fois, par ces morceaux. En appelant la fonction d'exécution de chaque bloc, l'ordonnanceur parcourt le diagramme de flux de la source à la destination à la recherche d'un bloc avec un tampon de sortie disponible et des données nécessaires à l'entrée. Quand il trouve ce bloc, il le traite pour la transmission. Il continue ensuite au bloc suivant. Quand il atteint la destination, qui est le bloc final du diagramme, il reprend de la source. L'ordonnanceur est essentiellement un interrogateur cyclique ou *cyclic poller*, appelant chaque bloc à son tour pour effectuer sa fonction d'exécution, toujours dans le même ordre.

Par défaut, GNU-Radio utilise l'ordonnanceur TPB qui alloue un fil d'instruction séparé pour l'exécution de chaque bloc. Lors de l'exécution, il n'y a pas d'ordonnement global parmi les blocs. Chaque fil d'instruction associé à un bloc s'exécute en boucle jusqu'à ce que le code GNU-Radio prenne fin. Pour chaque boucle, le fil d'instruction appelle d'abord l'exécuteur de ce bloc qui vérifie si son tampon de sortie est disponible et que les données d'entrée nécessaires le sont aussi. Si c'est le cas, l'exécuteur appelle la fonction principale *work* pour le traitement de ces données, puis il informe les blocs voisins des changements d'état. Dans le cas contraire, ce bloc attend le changement d'état de ses voisins, puis il vérifie à nouveau son état. Suivant le mécanisme précédemment décrit, tous les blocs du diagramme traitent ensemble les données entrantes morceaux par morceaux [76].

3.4.2.3 Architecture de l'ordonnanceur TPB

GNU-Radio contient de nombreux types de blocs, notés GR blocs, dépendamment de la façon dont ils traitent le rapport signal/traitement de données. Brièvement, chaque bloc reçoit un ou plusieurs flux de données d'entrée, les traite et génère un ou plusieurs flux de données de sortie dont la taille peut être différente du nombre de flux d'entrée. Il y a deux exceptions à cette règle: le premier bloc dans un diagramme de flux ou le bloc source a uniquement des flux de sortie, à l'inverse, le dernier bloc du diagramme appelé bloc destination ou *sink bloc* admet uniquement des flux d'entrée. Généralement un bloc dispose d'une fonction principale, appelée fonction de travail, qui effectue le calcul réel. Pour un nombre donné d'éléments

d'entrée à partir des flux d'entrée, cette fonction calcule un nombre correspondant d'éléments de sortie et les met aux flux de sortie. Différents blocs ont une représentation variable selon l'implémentation détaillée de la fonction de travail. Comme l'illustre la Figure 3.12, à chaque flux de sortie d'un tampon noté *GR buffer*, une mémoire tampon est allouée pour contenir ses données. Pour contenir les données, chaque tampon utilise un *circular buffer* [76]. Chaque bloc attribue également un lecteur de mémoire tampon pour chacun des flux d'entrée. Chaque lecteur tampon lit les données à partir du tampon en amont pour lequel il se réfère. Les tampons et les lecteurs tampons fournissent les canaux des blocs adjacents pour l'échange des données et des changements d'état.

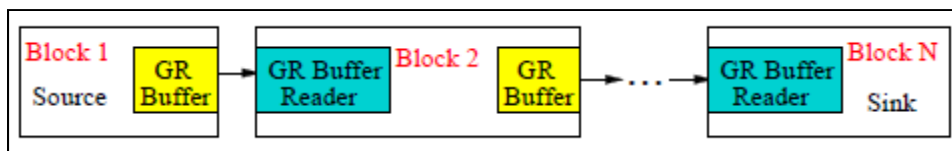


Figure 3.12 Concept de diagramme de flux avec les mémoires tampons GR [76]

Dans chaque bloc, une fonction dite exécuteur est utilisée pour aider au fonctionnement de l'ordonnanceur. Une fois appelé par l'ordonnanceur, l'exécuteur d'un bloc de traitement, détermine le tampon de sortie minimal disponible à ses flux de sortie. S'il n'y a pas de tampon, l'exécuteur signale à l'ordonnanceur que ce bloc est suspendu et est en attente pour la disponibilité du tampon de sortie. S'il existe un tampon de sortie disponible, l'exécuteur estime la quantité de données d'entrée nécessaires à chaque flux d'entrée pour générer suffisamment de données de sortie pour remplir la mémoire tampon de sortie disponible. S'il n'y a pas assez de données d'entrée disponibles à n'importe quel flux d'entrée, l'exécuteur peut essayer de réduire la taille des tampons de sortie et refaire l'estimation. En contrepartie, un bloc source doit uniquement vérifier s'il y a un tampon de sortie disponible, et vice versa pour le bloc destination. S'il n'y a pas de blocage, l'exécuteur appelle la fonction de travail qui exécute le calcul sur les données d'entrée disponibles et génère la quantité correspondante des données de sortie. Ensuite l'exécuteur notifie à l'ordonnanceur la progression du traitement [76].

3.5 Conclusion

Dans ce chapitre, on a décrit de près les éléments constituant la plateforme matérielle et logicielle. Les aspects relatifs au développement et le fonctionnement des applications sur GNU-Radio ont aussi été énoncés. Ces informations descriptives permettent de fixer le vocabulaire et les concepts qui seront utilisés pour la réalisation de ce projet. Au chapitre suivant, on décrira les étapes de mise à jour sur GNU-Radio des implémentations qui ont servi de points de départ pour notre projet.

CHAPITRE 4

MISE À JOUR DU SYSTÈME DE LOCALISATION SUR GNU-RADIO

4.1 Introduction

Au cours de ce chapitre, on définit les éléments de la solution de localisation servant de point de départ à ce projet à savoir l'implémentation de "*BBN 802.11*" et le système de localisation d'intérieur "*Wifilocalization*". La description des étapes suivies pour la mise à jour du système est exposée. La compatibilité du système avec la dernière mise à niveau des pilotes des USRP N210 étant réalisée, on expose les différents tests préliminaires conduits ainsi que les résultats obtenus à cet effet.

4.2 Réseau local sans fil IEEE 802.11b

IEEE802.11b supporte des débits de données variant de 11 Mbps à 5.5 Mbps, 2 Mbps et 1 Mbps. La plupart des ordinateurs portables disposent d'une interface WLAN 802.11b intégrée en plus de son interface de base IEEE 802.3 10Base-T. Le protocole IEEE802.11b fonctionne à la bande de fréquence ISM de 2,4 GHz, qui ne nécessite pas de licence pour l'utilisation, par contre elle nécessite que les fabricants respectent les exigences relatives aux dispositifs du réseau local sans fil pour leurs produits [87]. Les canaux 802.11 comprennent 22 MHz de largeur de bande et sont organisés comme le montre la Figure 4.1.

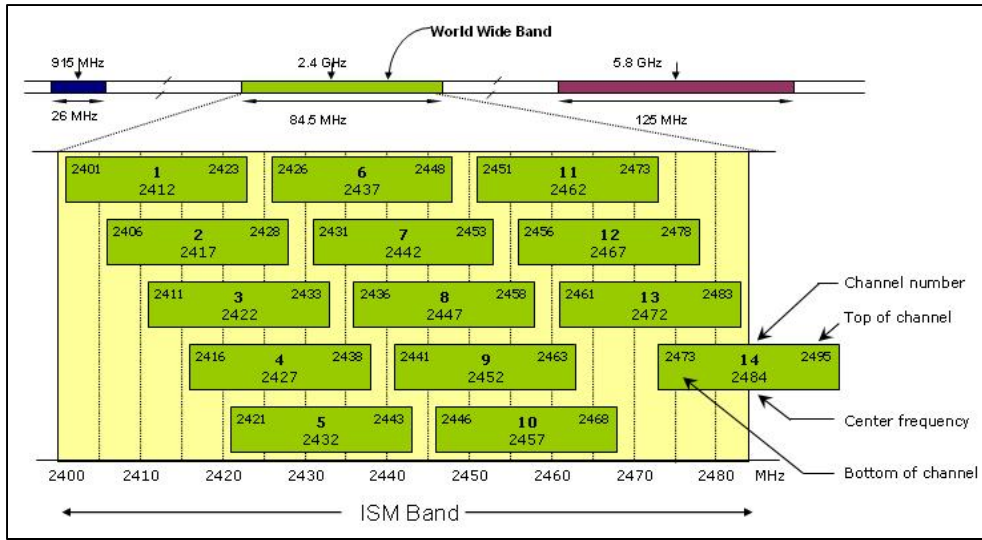


Figure 4.1 Les canaux Wifi IEEE 802.11 [86]

À la Figure 4.2, sont présentés les différents formats de données définis pour le protocole 802.11b avec les débits correspondants.

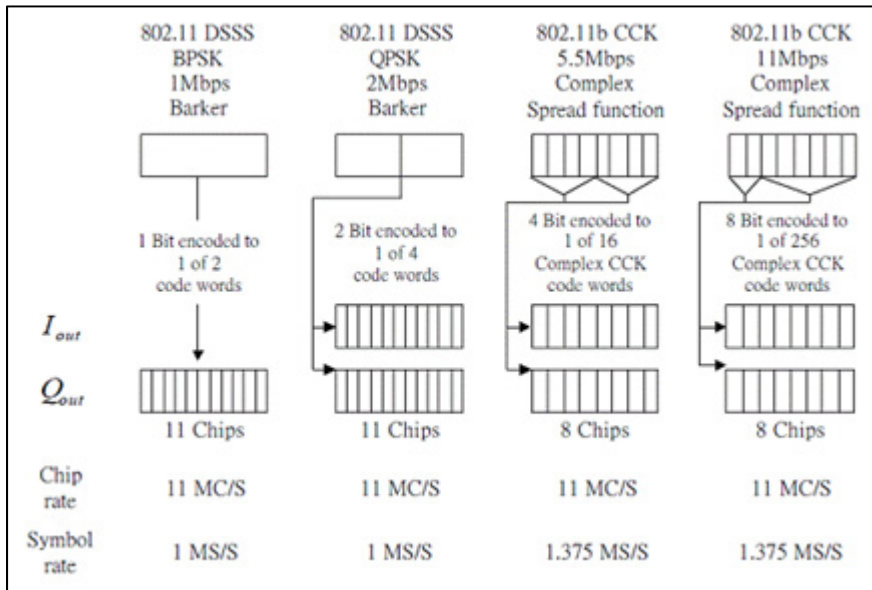


Figure 4.2 Débits et schémas de modulations du protocole IEEE 802.11b [86]

4.2.1 Implémentations de 802.11 sur radio définie par logiciel

4.2.1.1 BBN 802.11b

Plusieurs tentatives de reproduction des signaux 802.11 ont été effectuées avec GNU-Radio. La firme *BBN Technologies* a développé une application partielle d'IEEE 802.11b pour les USRP dans le cadre du projet ADROIT, financé par le programme ACERT du DARPA [84]. L'objectif de ce projet est de créer un réseau sans fil cognitif capable de reconfigurer les radios en fonction des caractéristiques du réseau et des conditions du canal. Le projet ADROIT a développé une large gamme de logiciels, dont beaucoup ont été rendus publics en vertu des licences open source. Parmi ces logiciels, un émetteur-récepteur pour IEEE 802.11b a été produit pour les radios USRP1 ainsi que des améliorations pour le cadre de développement GNU-Radio 3.1.1 [84]. Cette implémentation pilote a ouvert la voie pour la communauté de recherche sur SDR pour apporter leurs contributions.

Étant donné que le spectre du signal IEEE 802.11 couvre 22 MHz et selon le critère de Nyquist, la fréquence d'échantillonnage minimale serait de 44 MHz. La bande passante totale requise est donc de $2 \times 22 = 44$ MHz. Ceci dépasse la limite utilisable de la bande passante USB de 8 MHz énoncée auparavant au chapitre 3 [67]. Avec *BBN 80211b*, la largeur de bande de signal est réduite à 4 MHz, avant d'être transmise au contrôleur USB. De cette façon, le signal est fondamentalement sous-échantillonné et donc le rapport signal à bruit se dégrade. Plusieurs affichages dans la liste de diffusion ou *mailing list* de GNU-Radio rapportent le succès d'une réception de données IEEE 802.11 à 1Mbps en BPSK.

4.2.1.2 Récepteur 802.11b du laboratoire SPAN

Motivé par les limites de *BBN 80211b* à l'époque, M. Firooz et N. Patwari du laboratoire "Signal Processing Across Networks" (SPAN) de l'Université de l'Utah ont proposé une alternative. Rappelons qu'IEEE 802.11 utilise l'étalement spectral par séquence directe ou

DSSS, avec une séquence de code de Barker à 11 éléments. *BBN80211b* sous-échantillonne délibérément le signal et le réduit à un format de 8 bits pour sa transmission via USB 2.0 à l'ordinateur hôte, le débit de données ainsi que le SNR sont compromis. Ce problème est contourné par la mise en œuvre de l'opération dés-étalement du signal, avant la connexion USB, au sein du FPGA *Altera* sur la carte mère de l'USRP. Ainsi, le débit symbole est réduit à $1M_{\text{symboles/seconde}}$, ce qui est supporté par l'USRP. Une fois arrivé à l'hôte, le signal sera démodulé par la couche *Physical Layer and Convergence Protocol* (PLCP) suivant la norme IEEE 802.11 [81]. Pour le décodage de la trame, les auteurs ont utilisé la solution de BBN. Leur résultat final est un récepteur basé sur la version 1 des USRP et les travaux de BBN qui est en mesure de recevoir des paquets 802.11b transmis à 2Mbps en QPSK [85]. Ensuite, une contribution à cette implémentation a abouti à une mise à niveau pour les USRP 2 basé sur l'Ethernet Gigabit.

4.2.1.3 Encodeur et émetteur 802.11p de FTW

Une autre implémentation du 802.11 sur radio définie par logiciel est un émetteur 802.11 des versions a, g et p développé par des chercheurs du centre de recherche en télécommunications de Vienne FTW. Cet émetteur est compatible avec GNU-Radio 3.2.2, emploie la plate-forme USRP2 et était fonctionnel à partir de Février 2010. Le code source est disponible sur la plateforme des projets libres GNU-Radio appelée *Comprehensive GNU Radio Archive Network* ou CGRAN [85]. Bien que, conçu pour créer des trames 802.11p pour une transmission avec USRP2, l'émetteur 802.11p et son encodeur utilisent la même couche physique de 802.11a et g.

4.2.2 Contributions au projet *BBN 802.11b*

Ultérieurement, des membres de la communauté GNU-Radio ont contribué au *BBN 802.11* pour qu'il soit compatible avec la plateforme USRP2 et GNU-Radio 3.2.2. Le problème de bande passante a été contourné à l'aide de l'interface Gigabit Ethernet. Ceci dit, cette version amendée du code est maintenant obsolète étant donné qu'elle supporte les pilotes Ethernet

brutes qui ont laissé la place à UHD comme mentionné au chapitre 3. L'examen du code source *BBN802.11* indique que ce dernier n'est pas sans limites. On note l'absence des mécanismes de correction de la fréquence et de la phase, d'où sa sensibilité vis-à-vis des conditions indésirables du canal. *De même*, aucune commande automatique de gain n'est mise en œuvre, ce qui rend les performances du système sensible au gain spécifié par l'utilisateur [85]. En résumé, l'implémentation SDR de *BBN802.11* choisie, comprend un émetteur et un récepteur 802.11b de base. Le récepteur est capable de décoder complètement les paquets 802.11b transmis à 1 Mbps et partiellement les paquets envoyés à 2Mbps [65, 85]. L'implémentation comprend la couche physique et une couche MAC partielle.

4.2.3 Émetteur-récepteur OFDM pour IEEE 802.11 a/g/p

Il s'agit du plus récent travail réalisé sur SDR. Cette implémentation a été publiée à la fin de rédaction de ce mémoire. C'est un émetteur-récepteur 802.11ag/p pour GNU-Radio. Il est testé avec l'USRP N210 et les cartes filles XCVR2450 et SBX [85].

4.2.4 Projet *WifiLocalization*

Ce projet consiste en une méthode hybride de localisation d'intérieur, passive, en temps réel et est basée sur deux informations : les délais temporels et les RSS. Les capteurs de localisation sont placés à des endroits connus et collectent les informations. Ensuite, un certain traitement en temps-réel est exécuté suivi d'un post-traitement via une base de données. En outre, le projet rapporte l'affichage en temps réel de la position de la cible sur une carte du site. À la Figure 4.3, on représente le diagramme de flux de la solution rapportée par *WifiLocalization* [56].

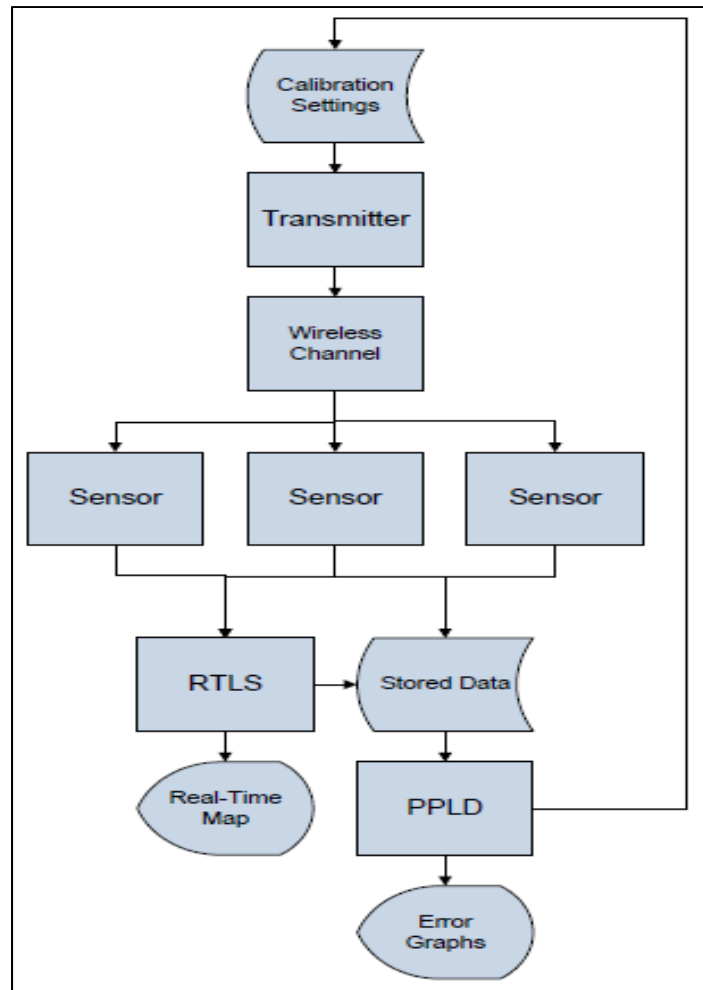


Figure 4.3 Diagramme de flux du projet *WiFiLocalization*

Le projet comprend un module de localisation temps-réel «*RTLS*» avec une carte géographique du site, un module de simulation du scénario de localisation, des *add-in* pour les récepteurs sans fils de BBN 802.11, notés *sensor*, pour la collecte de données et leur transmission sur le réseau afin de les traiter sur un serveur central. Le projet comprend aussi un module de post-traitement et d'améliorations des résultats dénoté "PPLD" ou *Post Processing Localization Database*. Cet outil est décrit et discuté à la section 5.3.

4.3 Mise à jour des codes sources de la chaîne de communication

L'application qu'on réalise dans le présent travail est caractérisée par une haute complexité puisqu'elle comprend un nombre important d'opérations devant être exécutées en temps-réel.

4.3.1 Mise à jour pour se conformer aux pilotes UHD

La première tâche est de réécrire le code source pour qu'il puisse être utilisé avec les pilotes UHD et ainsi promouvoir sa portabilité. La modification du code comprend entre-autre, le remplacement des blocs "usrp_sink" et "usrp_source" par les blocs "UHD_usrp_sink" et "UHD_usrp_source", ainsi que la mise au point des paramètres de ces derniers. On cite les différents points de modifications :

- définir la configuration de l'horloge de l'USRP à *internal*, via la commande `self.u.set_clock_config(uhd.clock_config.internal(), uhd.ALL_MBOARDS)` ;
- remplacer les facteurs d'interpolation/décimation avec les nouvelles fréquences d'échantillonnage des USRP avec UHD via la commande `self.u.set_samp_rate(options.sample_rate)`. Cette valeur, qui peut pratiquement aller jusqu'à 25 MSps, est passée à l'USRP N210 comme option lors de l'exécution ;
- contrôler le facteur gain du récepteur via la commande `self.u.set_gain(options.gain)`. Pour la carte fille RFX 2400, le gain Rx peut être une valeur comprise entre 0 et 70 dB caractérisant l'amplificateur programmable de gain (PGA). Le réglage du gain Tx, par contre, n'est pas supporté par cette carte fille. Pour la carte SBX, les gains Rx et Tx sont compris entre 0 et 31.5 dB ;
- régler la fréquence centrale du signal avec `self.u.set_center_freq(options.cordic_freq)` ;

- supprimer les lignes qui traitent de la sélection de USRP *subdevice*; de la commutation automatique entre les modes d'émission et de réception, étant donné que les USRP N210 fonctionnent en mode duplex intégral ;
- introduire l'option de contrôle du format de données sur le medium ou *otw_format* pour *over the wire format*. Dans ses dernières mises à jour, UHD a en fait introduit le support d'un format de données de moindre résolution permettant plus de bande passante entre l'hôte et l'USRP. De ce fait, on a pu dépasser la limite théorique de 25MSps avec 16 bits de résolution jusqu'à 50MSps à 8 bits uniquement de résolution.

4.3.2 Codes émetteur et récepteur « *neoTx.py* » et « *neoRx.py* »

Le code consiste en un ensemble de fichiers reliés entre eux écrits en python et faisant appel à d'autres fichiers écrit en C++ implémentant des fonctionnalités de calcul avancées. Parmi ces fichiers on note « *bbn_80211b.py* » qui instancie les blocs GR relatifs au codage et décodage. « *bbn_80211b_pkt.py* » sert d'enveloppe pour les classes de *bbn_80211b.py*. En effet, on doit instancier les classes dans *bbn_80211b_pkt.py*, et non pas directement dans *bbn_80211b.py*. Les classes inscrites dans *bbn_80211b_pkt.py* sont *bbn_80211b_mod_pkts* et *bbn_80211b_demod_pkts*.

La classe *bbn_80211b_mod_pkts* instancie la file d'attente des paquets pour la transmission et assure la fonction *send_pkt*. Cette dernière prend une charge utile ou *payload*, ajoute le préambule de 802.11, l'entête PLCP ainsi que le contrôle de redondance cyclique (CRC) correspondant. La classe *bbn_80211b_demod_pkts* instancie la file d'attente de réception et crée le fil d'instruction ou *thread* pour la surveillance les paquets reçus.

neoRx.py et *neoTx.py* sont les scripts à exécuter pour envoyer et recevoir des paquets avec l'USRP N210. À l'annexe I et II, on présente les codes sources de ces deux scripts. Lors de leurs exécutions, on doit choisir le filtre de mise en forme comme étant un cosinus surélevé ou bien une impulsion en bande de base de Barker [85]. En dépit du fait que le taux de

symbole de 802.11 BPSK et QPSK est à 1 MSps et 2MSps respectivement, l'énergie du signal, elle, est répartie sur 22 MHz en bande de base en utilisant un code d'étalement Barker à 11 bits comme le montre la Figure 4.2.

Pour envoyer des données à 1 Mbps, l'émetteur est configuré à utiliser 4 échantillons par symbole `./neoTx.py -S 4` et faire de même pour le récepteur: `./neoRx.py -S 4`. On a réussi à récupérer des paquets 802.11 en utilisant un taux d'échantillonnage de 25 MSps en utilisant `./neoRx.py -d 25 -b -f 2412e6`

L'option "-d 25" indique au récepteur de fonctionner à 25 MSps, et "-f 2412e6 " signifie l'écoute sur le canal N°1 du 802.11, le "-b" signifie un l'étalement selon le code de Barker. L'option "-p", si utilisée, désactive le contrôle CRC. Pour chaque paquet reçu ayant un CRC valide, le récepteur affiche le message "PKT" comme le montre la Figure 4.11 et la Figure 4.12 pour les modulations BPSK et QPSK respectivement.

Le script à exécuter pour lancer l'émetteur est :

```
python neoTx.py -a "addr=192.168.10.2" -r 5000 -f 2.412e9 -S 4 -s 25e6 -m
a0:36:fa:25:31:2a -R 2 -w Sc8
```

où:

- a : l'option de configuration de l'adresse IP ;
- m : l'adresse MAC ;
- r : le nombre de paquets à transmettre ;
- f : la fréquence en Hz ;
- S : le nombre d'échantillons par symbole ;
- s : fréquence d'échantillonnage ;
- R : schéma de modulation (1 Mbps BPSK ou 2Mbps QPSK) ;
- w : Résolution des données sur le moyen de communication (16 bits ou 8 bits).

À la Figure 4.5, on représente la structure en bloc de fonctionnement de l'émetteur *neoTx*.

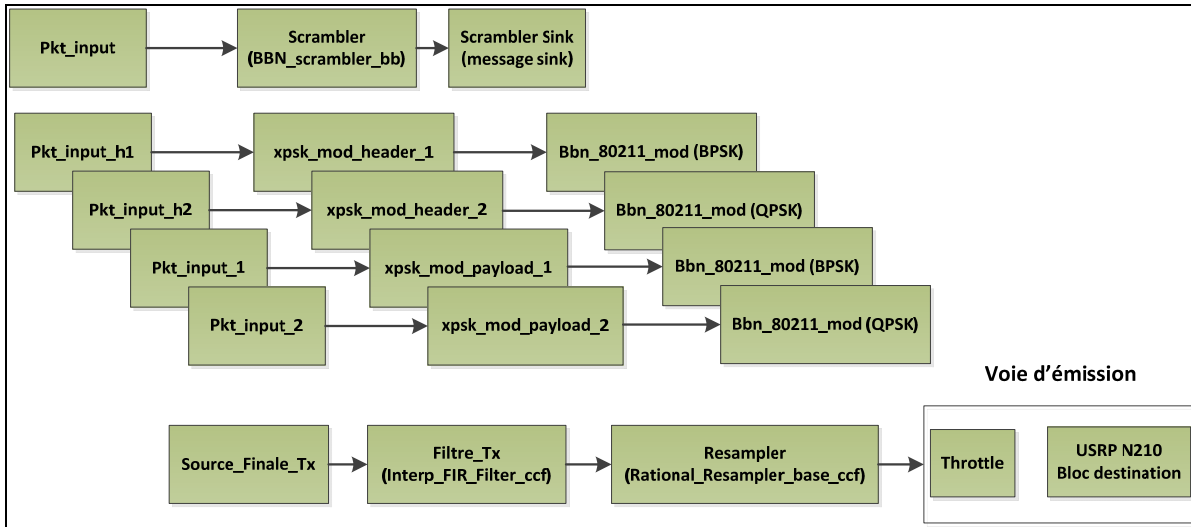


Figure 4.4 Schéma bloc des modules relatifs au récepteur neoTx

À la Figure 4.5, on illustre le diagramme de flux pour le récepteur *BBN 802.11*. Le diagramme fait état des différents étages de réception.

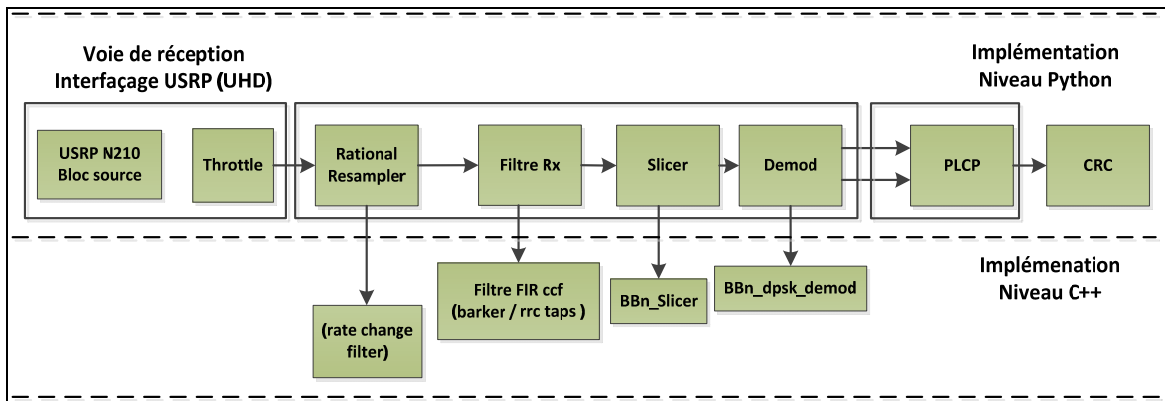


Figure 4.5 Schéma bloc des modules relatifs à l'émetteur neoRx

Le bloc « BBN.DPSK_DEMOD » effectue la démodulation selon le schéma choisi et met à jour la valeur de RSS selon les points obtenues de la constellation reçue. Le bloc

« BBN.SLICER » effectue le découpage en morceaux de données ou *chunks* dépendamment du paramètre d'échantillons par symbole ou SPB.

La combinaison des données avec le code pseudo aléatoire généré passent au niveau de l'émetteur par le bloc « BBN_SCRAMBLER » et au niveau du récepteur par le bloc « BBN_DESCRAMBLER » afin de rendre aléatoire le spectre de sortie et donc supprimer les lignes spectrales discrètes quand un symbole est répété. Le bloc « BBN.PLCP80211 » représente le bout de la chaîne de réception. Le décodage des contenus des paquets est effectué à ce niveau. PLCP, par référence à *Physical Layer Convergence Protocol*, constitue une sous couche de la couche physique communicant avec la couche MAC. Le bloc « *Throttle* » est utilisé dans le but de cadencer le flux de données entre les blocs GR, selon le choix de l'utilisateur du taux d'échantillonnage et non pas au maximum possible de la fréquence du processeur de la machine hôte.

Pour le récepteur, on exécute le script suivant avec les paramètres définis à la section 4.3.2.

```
python neoRx.py -a "addr=192.168.20.5" -g 30 -f 2.412e9 -S 25 -s 25e6
-m a0:36:fa:25:31:2a -w -Sc8
```

Aux figures précédentes, les structures en bloc du récepteur et de l'émetteur montrent les deux niveaux d'abstraction : C++ et python et mettant l'accent sur l'interaction entre les fichiers écrits en langages différents pour l'implémentation du nouveau émetteur-récepteur.

4.4 Tests préliminaires et résultats obtenus

4.4.1 Tests préliminaires

Au départ de ce travail, on a examiné la possibilité d'utiliser GRC pour développer le système de localisation. Après, on a constaté que la conception du système était trop complexe à mettre en œuvre sur GRC et qu'elle nécessite de nouveaux blocs de traitement.

Sur ce, uniquement une aide à la conception est réalisée par GRC. Ensuite, le travail a été dirigé vers du codage C++ et Python. Même si GRC n'a pas été utilisé pour ces expériences, on a eu recours à quelques blocs de traitement de signal pour réaliser des tests préliminaires.

Le GRC a été utilisé pour la calibration des niveaux de gains nécessaires à l'obtention d'une figure acceptable de constellation pour valider la réception. Les blocs utilisés pour ces tests sont la *FFT sink* qui a servi d'analyseur de spectre, l'*USRP source* comme récepteur, les variables et leurs curseurs ou *slider* pour la configuration en temps réel. On s'est servi aussi de blocs *sink* pour visualiser la constellation, le comportement temporel ainsi que la figure *waterfall*. Les tests consistent à lancer le script émetteur sur une machine notée Hôte_{Tx} et recevoir sur une autre machine selon la Figure 4.6. On obtient à la Figure 4.7 et Figure 4.8 les spectres des signaux reçus BPSK et QPSK respectivement. On a réglé les valeurs nécessaires aux paramètres afin d'avoir un signal reçu acceptable.

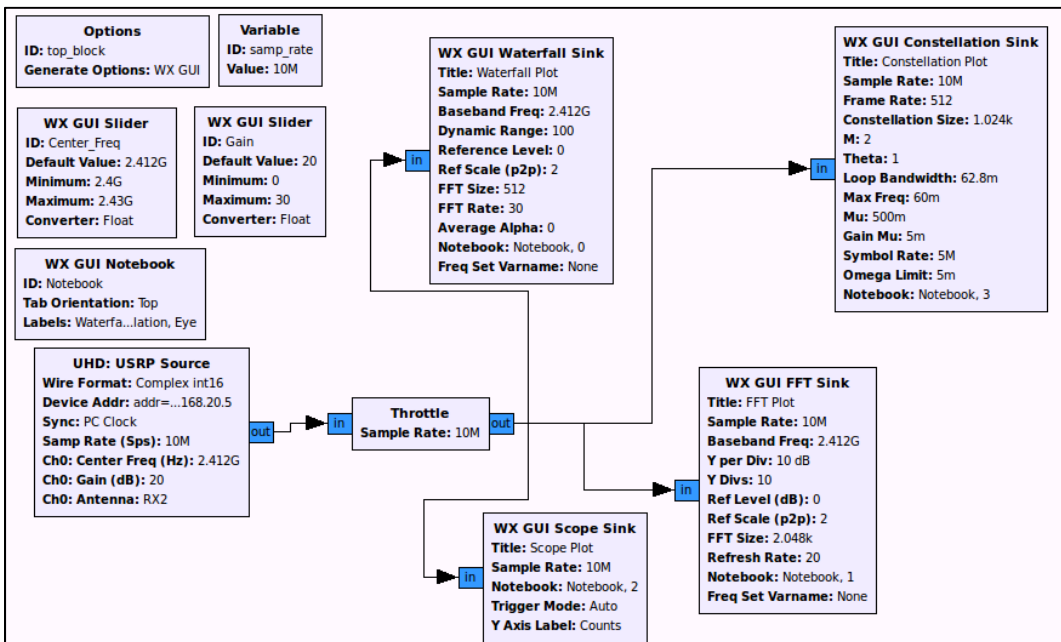


Figure 4.6 Banc d'essai pour le réglage des paramètres nécessaires à la réception

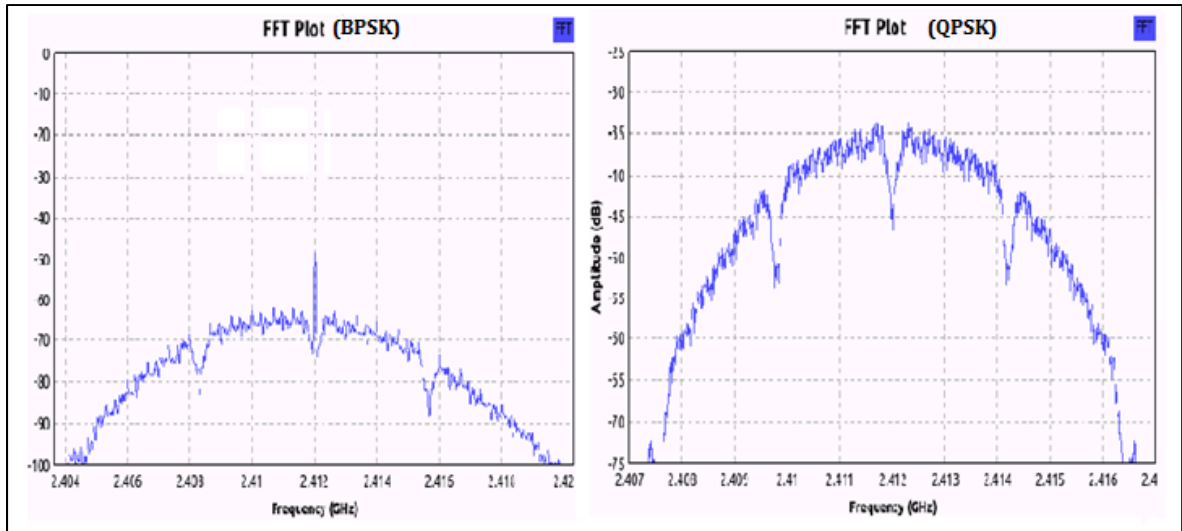


Figure 4.7 Spectre des signaux reçu par GRC (BPSK et QPSK)

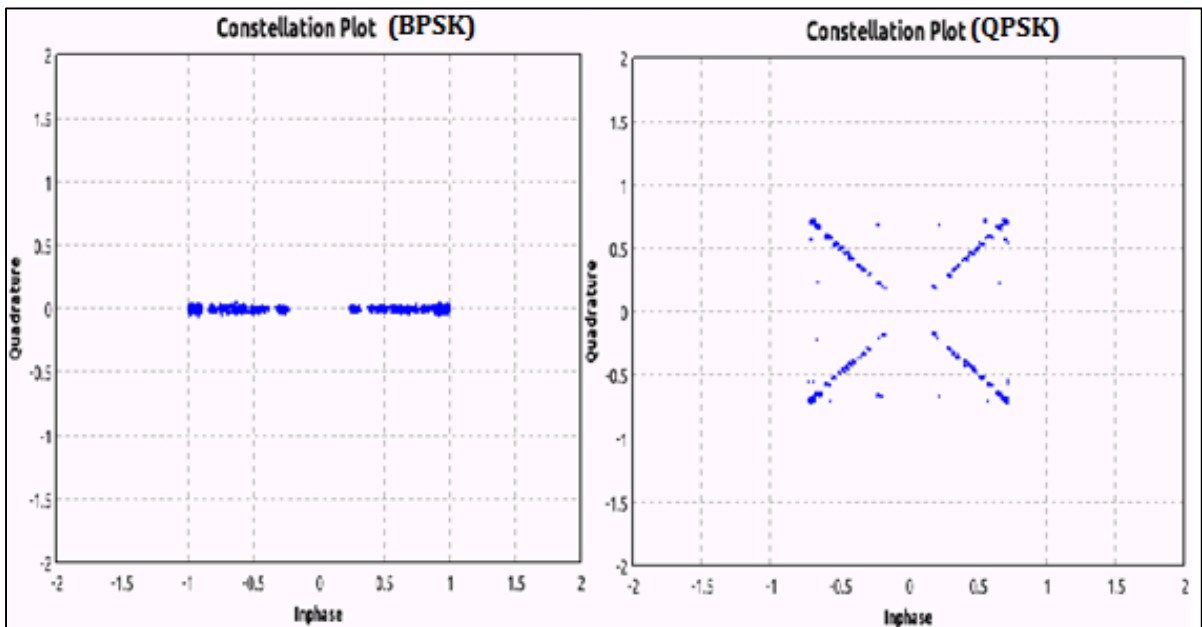


Figure 4.8 Constellations des signaux démodulés via GRC (BPSK et QPSK)

Pour valider encore la réception et la démodulation, on a aussi eu recours à un analyseur de spectre et on a obtenu la Figure 4.9 et la Figure 4.10.

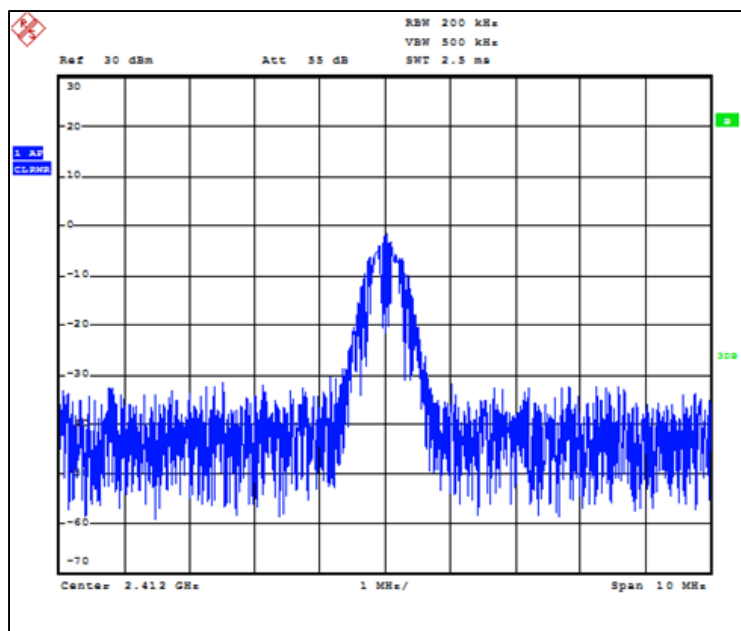


Figure 4.9 Spectre du signal BPSK reçu, $F_{\text{éch.}} = 25$ MHz

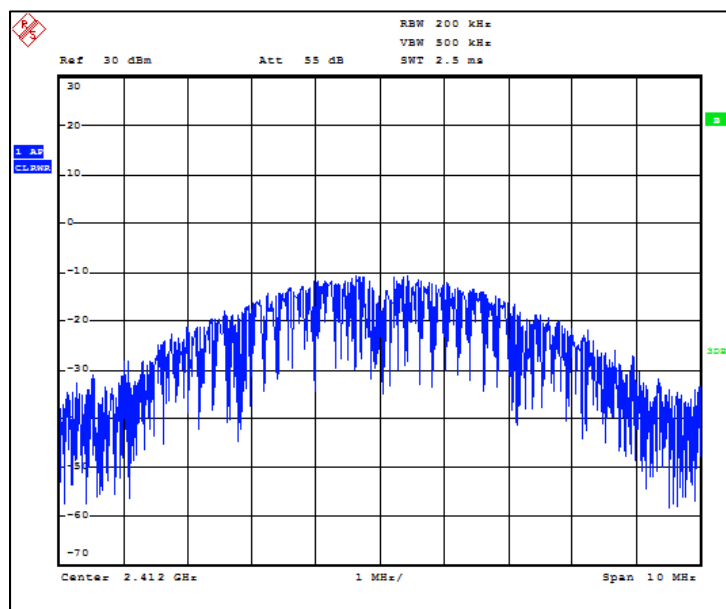


Figure 4.10 Spectre du signal QPSK reçu, $F_{\text{éch.}} = 25$ MHz

4.4.2 Résultats obtenus

Le protocole de paquets aide à déterminer comment décoder correctement les données au niveau du récepteur. Plusieurs étapes de tests de protocoles de paquets ont été développées, à savoir en Python et en C++. Les tests en python forment l'étape préliminaire qu'on a utilisée pour valider la conception du système. En regardant le code BBN, on reconnaît la longueur des paquets, la valeur RSS, l'ID source, l'instant à partir duquel le récepteur a été activé et le débit de données. Les paquets contiennent également une charge utile ou *payload*, qui est le champ où nous avons décidé de passer les informations dont nous avons besoin. L'information supplémentaire à transmettre est l'adresse MAC et un horodatage ou *timestamp* au moment de la transmission.

Une fois l'information transmise avec succès au récepteur, le message est analysé pour pouvoir utiliser ces données en conséquence. La Figure 4.11 montre à la fois le message transmis et le message à la réception pour une communication à 1MBps avec BPSK et la Figure 4.12 montre une communication réussie à 2MBps avec QPSK.

```

root@naceur-HP-Compaq-6005-Pro-SFF-PC: /home/naceur/D
root@na
Sending
ÉMETTEUR
a0:36:fa:25:31:2a1384968582.933223009
    Sending pkt 1
a0:36:fa:25:31:2a1384968583.934580088
    Sending pkt 2
a0:36:fa:25:31:2a1384968584.935955048
    Sending pkt 3
a0:36:fa:25:31:2a1384968585.936669111
    Sending pkt 4
a0:36:fa:25:31:2a1384968586.937794924
    Sending pkt 5
a0:36:fa:25:31:2a1384968587.939081907
    Sending pkt 6
a0:36:fa:25:31:2a1384968588.940215111
    Sending pkt 7
a0:36:fa:25:31:2a1384968589.940670967
    Sending pkt 8
Adresse MAC émetteur   Instant d'émission (UNIX)
a0:36:fa:25:31:2a1384968589.940670967

root@naceur-H...
root@naceur-H...
root@naceur-H...
root@naceur-H...
#-----#
Received header!
signal: 0x0A
service: 0x00
length: 0x0148
crc: 0xB95F
Calculated crc: 0xB95F
Running at 1Mbps
Packet Length: 41 bytes
#-----#
97 48 58 51 54 58 102 97 58 50 53 58 51 49 58 50 97 49 51 56 52 57 54 56
53 56 57 46 57 52 48 54 55 48 57 54 55 237 17 216 175
PKT: len=37, rssi=-96, src=35:3a:33:31:3A:32, time=3737664, rate=1 Mbps
Payload: a0:36:fa:25:31:2a1384968589.940670967
Tx MAC:      a0:36:fa:25:31:2a
Tx timestamp: 1384968589.940670967

Received header!
signal: 0x0A
service: 0x00
length: 0x0148
crc: 0xB95F
Calculated crc: 0xB95F
Running at 1Mbps
Packet Length: 41 bytes
#-----#
97 48 58 51 54 58 102 97 58 50 53 58 51 49 58 50 97 49 51 56 52 57 54 56
53 57 48 46 57 52 49 56 48 48 49 49 55 172 18 63 32
PKT: len=37, rssi=-96, src=35:3a:33:31:3A:32, time=4405088, rate=1 Mbps
Payload: a0:36:fa:25:31:2a1384968590.941800117
Tx MAC:      a0:36:fa:25:31:2a
Tx timestamp: 1384968590.941800117

```

Figure 4.11 Protocole implémenté, mode BPSK à 1Mbps

```

root@naceur-HP-Compaq-6005-Pro-SFF-PC: /home/naceur/Docu
root
ÉMETTEUR
UHD transmitter:
*****
IP @:      addr=192.168.10.2
MAC @:     a0:36:fa:25:31:2a
Freq:     2.412GHz
Sample Rate: 25M samples per sec.
Antenna:   TX/RX
Wire-Format: sc16
Using Barker Code spreading

Sending pkt 0
a0:36:fa:25:31:2a1384968419.179184914
    Sending pkt 1
a0:36:fa:25:31:2a1384968420.180469990
    Sending pkt 2
a0:36:fa:25:31:2a1384968421.181787968
    Sending pkt 3
a0:36:fa:25:31:2a1384968422.183099985
    Sending pkt 4
a0:36:fa:25:31:2a1384968423.184371948
    Sending pkt 5
Adresse MAC émetteur   Instant d'émission (UNIX)
a0:36:fa:25:31:2a1384968421.181787968

root@naceur-HP-Compaq-6005-Pro-SFF-PC: /home/naceur/Documents/wifiLocalizati
root@naceur-H...
root@naceur-H...
root@naceur-H...
root@naceur-H...
Récepteur
Received header!
signal: 0x14
service: 0x00
length: 0x00A4
crc: 0x850C
Calculated crc: 0x850C
Running at 2Mbps
Packet Length: 41 bytes
#-----#
97 48 58 51 54 58 102 97 58 50 53 58 51 49 58 50 97 49 51 56 52 57 54 56
52 50 49 46 49 56 49 55 56 55 57 54 56 243 28 222 44 0
PKT: len=37, rssi=-96, src=35:3a:33:31:3A:32, time=9048768, rate=2 Mbps
Payload: a0:36:fa:25:31:2a1384968421.181787968
Tx MAC:      a0:36:fa:25:31:2a
Tx timestamp: 1384968421.181787968

Received header!
signal: 0x14
service: 0x00
length: 0x00A4
crc: 0x850C
Calculated crc: 0x850C
Running at 2Mbps
Packet Length: 41 bytes
#-----#
97 48 58 51 54 58 102 97 58 50 53 58 51 49 58 50 97 49 51 56 52 57 54 56
52 50 46 49 56 51 48 57 57 56 53 211 84 43 2 0
PKT: len=37, rssi=-96, src=35:3a:33:31:3A:32, time=9716248, rate=2 Mbps
Payload: a0:36:fa:25:31:2a1384968422.183099985
Tx MAC:      a0:36:fa:25:31:2a
Tx timestamp: 1384968422.183099985

```

Figure 4.12 Protocole implémenté, mode QPSK à 2Mbps

Une fois les scripts sont exécutés et le scénario de localisation est lancé, on récupère les résultats numériques suivants : adresse Mac de l'émetteur, les instants d'envoi ou $timestamp_{Tx}$ en deux parties : une en seconde et une partie fractionnelle, les instants de réception ou $timestamp_{Rx}$ et les RSS en dBm.

Un exemple de scénario de localisation selon les paramètres énoncés à la section 4.3.2 a permis de récupérer les données à l'annexe III. On a reconduit la même expérience à différentes distances entre l'émetteur et le récepteur en commençant par 1 mètre jusqu'à 6.5 mètres avec des pas de 0.5 mètre. Le banc d'essai est illustré à la Figure 4.13. La disposition des radios reste toujours en vue directe et sans modification de paramètres.

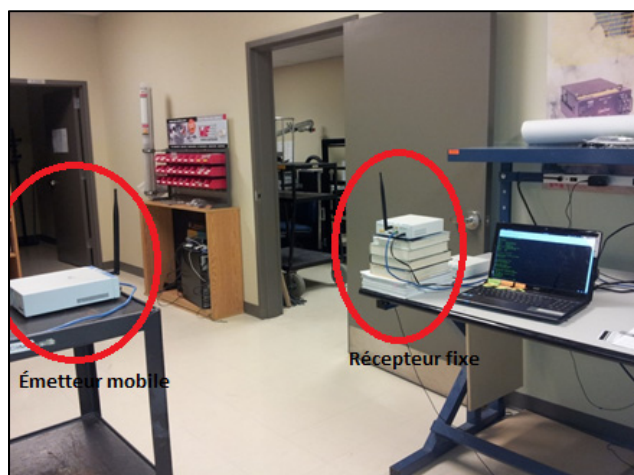


Figure 4.13 Banc d'essai illustrant le récepteur fixe et l'émetteur mobile

En collectant les différents fichiers de résultats, on trace en fonction de la distance réelle entre les radios, la distribution des délais temporels à la Figure 4.14, la distance estimée résultante à la Figure 4.15, la distribution des puissances moyennes reçues à la Figure 4.16 et finalement les distances brutes estimées par RSS à la Figure 4.17. Les lignes de meilleurs ajustements de ses variables sont incluses sur les mêmes figures. Ces ajustements traduisent le comportement global des variables étudiées.

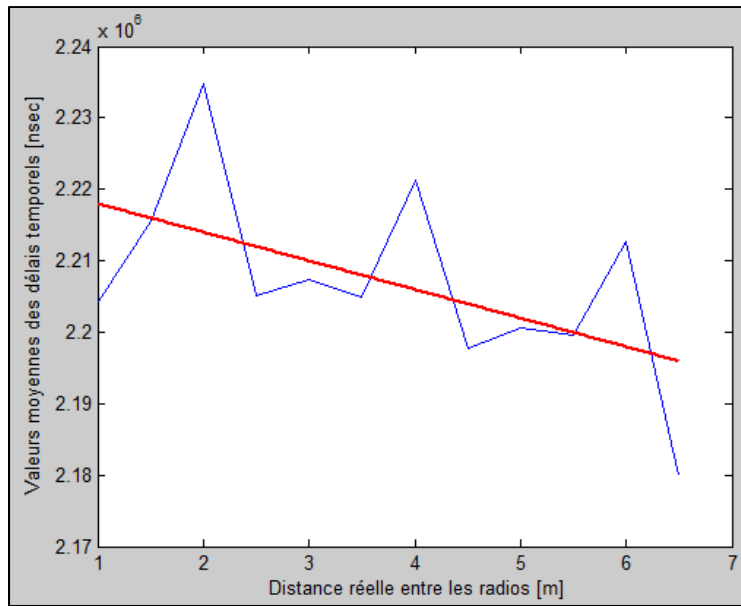


Figure 4.14 Délais de transmission en fonction de la distance réelle entre radios

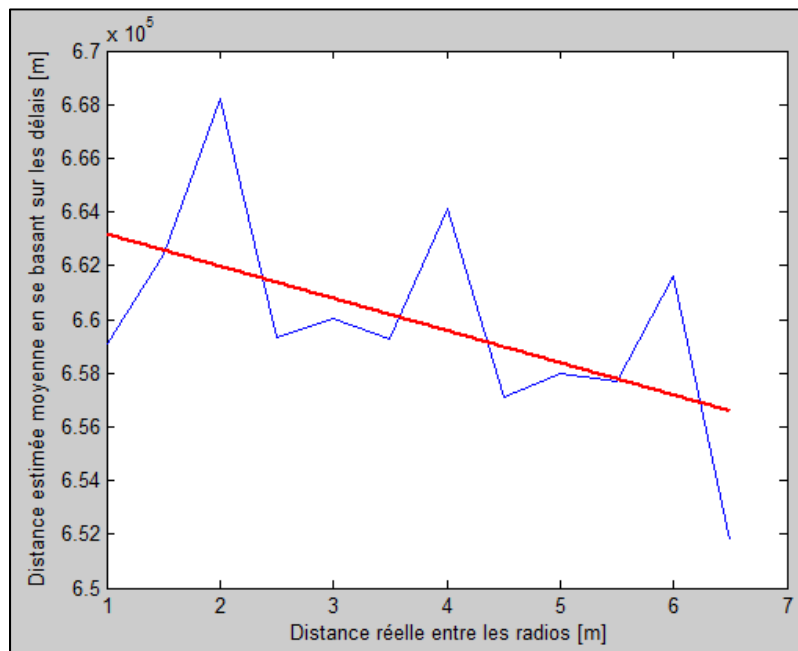


Figure 4.15 Distances moyennes en se basant sur les délais de transmission

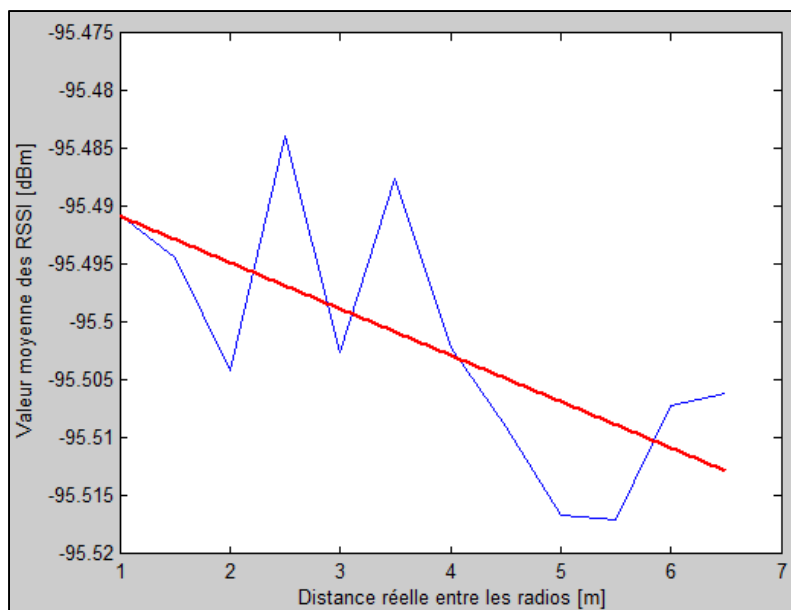


Figure 4.16 Distribution de la valeur moyenne des RSS

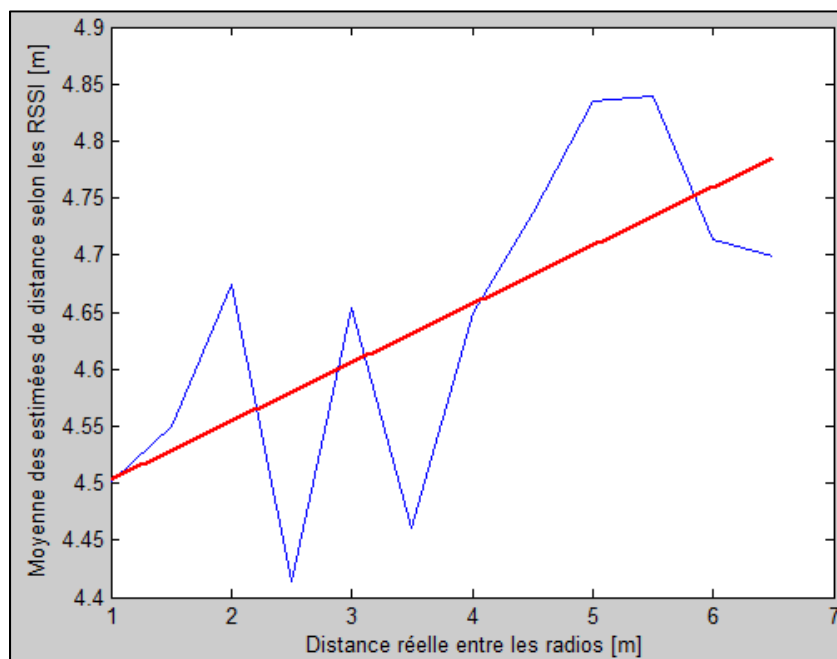


Figure 4.17 Distances brutes estimées à partir des RSS

À la section 5.4.1, on présente le modèle utilisé pour dériver les estimées de distances depuis les valeurs de RSS collectées.

En analysant les Figure 4.14 et Figure 4.15, on remarque que la résolution temporelle est variable et est de l'ordre de 2.2 millisecondes donnant ainsi une estimée de distance de 660 Km ce qui est non admissible pour la localisation d'intérieur. Pour dégager les causes de variabilité des données de délais de transmission, on trace les délais en fonction du nombre de paquets reçus. Les radios sont à une distance fixe de 4.5 mètres. Le résultat de cette expérimentation avec la ligne de meilleur ajustement apparaissent à la Figure 4.18.

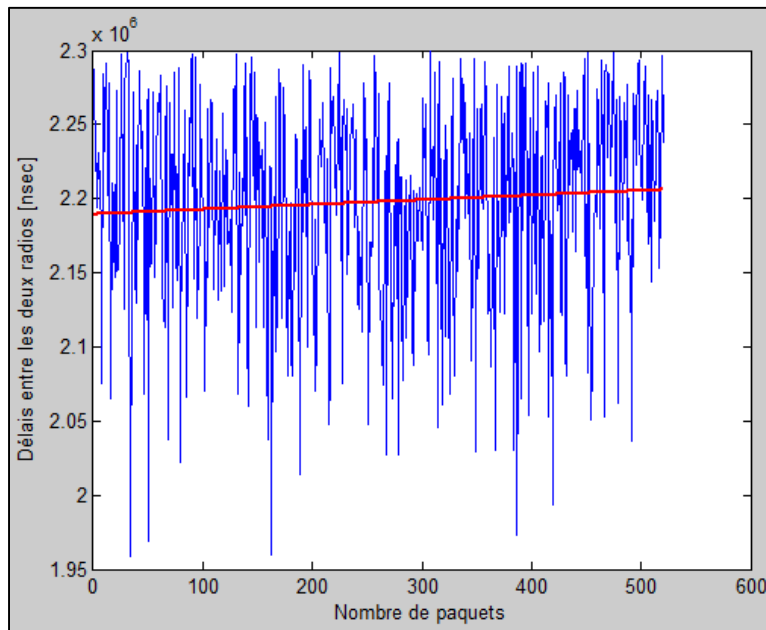


Figure 4.18 Délais à distance fixe en fonction du nombre de paquets

Suivant la même logique, on trace la figure de distribution des valeurs de RSS à la même distance 4.5 mètres entre les radios et on obtient la Figure 4.19.

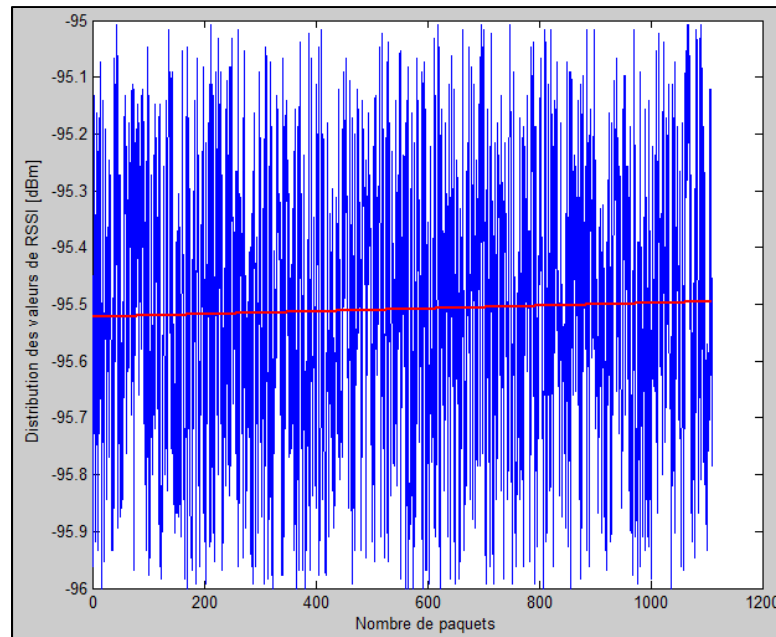


Figure 4.19 RSS en fonction du nombre des paquets reçus à une distance fixe

4.5 Conclusion

Dans ce chapitre, on a étalé la technique d'implémentation des nouveaux codes de transmission et de réception. De plus, on a présenté les résultats préliminaires de calibration du système, ainsi que les résultats se rapportant à l'objectif de localisation. On a tracé les courbes de distribution du RSS moyen en fonction des distances réelles entre les radios ainsi qu'en fonction des paquets reçus à une distance fixe entre les radios. Les mêmes étapes sont exécutées pour les délais temporels. Les estimés de distances à partir des deux méthodes précédentes sont calculés et tracés en fonction des distances entre les radios. On aperçoit clairement la disparité entre les résultats obtenus et les prédictions. Pour la méthode basée sur les délais cette disparité est due principalement aux délais cumulés par les paquets tout au long de la chaîne d'émission et de réception. Les valeurs de RSS, elles, comptent pour des variations minimales qui ne reflètent pas le comportement attendu. Pour cela, il importe de bien cerner les causes de cette disparité et d'élaborer des améliorations considérables au système. Tout cela est présenté au chapitre suivant.

CHAPITRE 5

AMÉLIORATIONS DU SYSTÈME DE LOCALISATION

5.1 Introduction

Au cours de ce chapitre, on expose les améliorations apportées au mécanisme de localisation. On commence par analyser les différentes causes affectant la résolution des valeurs d'horodatage rapportées au chapitre 4. Ensuite, on décrit l'intégration d'un nouveau module d'estimation de RSS. De même, on décrit les étapes de mise à jour du code source de localisation pour faire l'usage de l'API d'UHD en C++ afin d'améliorer la fonctionnalité d'horodatage du système. Par ailleurs, on expose les résultats de cette dernière approche et on décrit les détails de conception d'une solution d'horodatage au sein du FPGA.

5.2 Étude des éléments de la latence

En examinant les résultats d'estimation de délais de transmission rapportés à la section 4.4.2 de ce mémoire, on réalise que l'ordre de variabilité de ces délais excède de loin l'ordre admissible pour pouvoir réaliser un post-traitement capable d'établir une loi de prédiction du comportement de ces variations. La prédiction du délai qu'effectue le paquet depuis l'insertion de l'horodatage jusqu'à sa conversion en un flux de paquets Ethernet est à même de résoudre la problématique de localisation d'intérieur par TDoA.

Ceci dit, on note aussi le délai variable de la communication entre l'hôte et l'USRP. Ce délai est d'ordre supérieur à l'objectif de dix nanosecondes qu'on s'est fixé au départ. Ajoutant à cela, les autres latences introduites par le traitement du signal numérique de l'hôte au sein du FPGA et du même signal converti sur la voie analogique de l'USRP. Les différentes latences sont illustrées à la Figure 5.1.

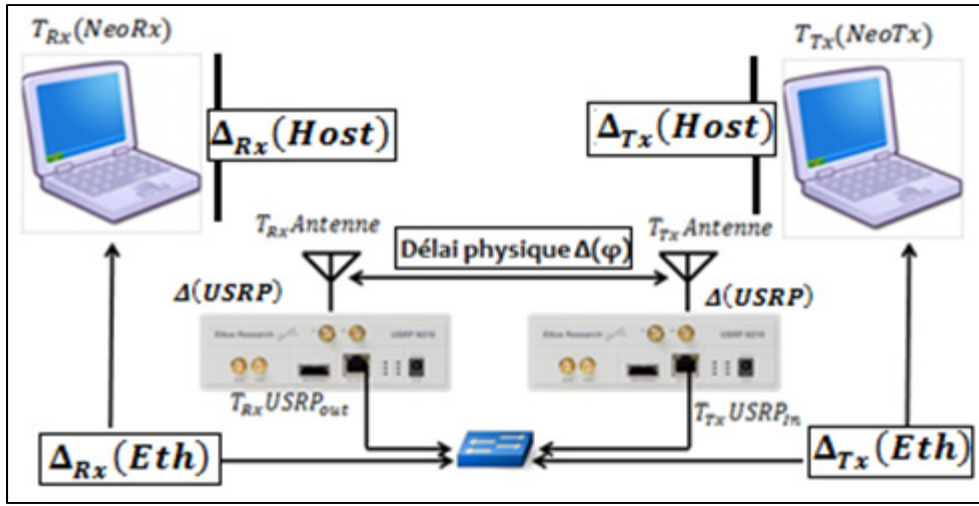


Figure 5.1 Latences introduites par le système sur les signaux

Le délai physique en fonction des latences du système est donné par (5.1).

$$\begin{aligned}
 \text{délai physique: } \Delta(\varphi) &= T_{Rx}(\text{antenne}) - T_{Tx}(\text{antenne}) \\
 &= T_{Rx}(\text{NeoRx}) - [\Delta_{Rx}(\text{Host}) + \Delta_{Rx}(\text{Eth}) + \Delta_{Rx}(\text{USRP})] - \\
 &\quad (T_{Tx}(\text{NeoTx}) + [\Delta_{Tx}(\text{Host}) + \Delta_{Tx}(\text{Eth}) + \Delta_{Tx}(\text{USRP})])
 \end{aligned} \tag{5.1}$$

Où l'on définit :

- $T_{Rx}(\text{antenne})$ et $T_{Tx}(\text{antenne})$: respectivement les instants de réception et d'émission du signal au niveau de l'antenne ;
- $T_{Rx}(\text{NeoRx})$: l'instant d'insertion de l'horodatage au niveau applicatif de Python et ce relativement à un paquet reçu et décodé correctement;
- $T_{Tx}(\text{NeoTx})$: l'instant d'insertion de l'horodatage au contenu d'un paquet au niveau applicatif de Python;
- $\Delta_{Tx}(\text{Host})$: le délai accumulé par un paquet depuis son insertion au contenu du paquet avec Python jusqu'à sa manipulation avec le noyau du système d'exploitation pour le convertir en un flux de trames formatées pour Ethernet et vice versa pour la quantité $\Delta_{Rx}(\text{Host})$. À la Figure 5.2, on illustre ces délais par le passage entre le noyau ou *Kernel* vers le niveau de l'application ou *userspace* ;

- $\Delta_{Rx}(Eth)$: le délai accumulé par le même paquet encapsulé sur le moyen de connexion Ethernet de l'USRP vers l'hôte ou inversement pour la quantité $\Delta_{Tx}(Eth)$. Le chemin parcouru par le paquet en question peut inclure un commutateur Ethernet si la radio n'est pas directement connectée à l'hôte ;
- $\Delta_{Rx}(USRP)$ et $\Delta_{Tx}(USRP)$: le délai accumulé par le paquet au sein de l'USRP. On distingue le délai dans la portion de traitement numérique : le FPGA et le délai en aval, lors du parcours de la section analogique de la radio. On montre à la Figure 5.2, que la première section est régie par la cadence de 100 MHz de la carte mère de l'USRP N210 et donc le délai y est proportionnel. À la section reliant les antennes des convertisseurs CAN/CNA, la dimension temporelle n'est plus une fonction des impulsions d'horloge mais on l'exprime plutôt par le temps "physique". Le délai y est déterministe. Il est proportionnel aux propriétés physiques des éléments de cette voie ainsi que de la longueur d'onde en question. Le délai y est exprimé par le délai de groupe ou *group delay*.

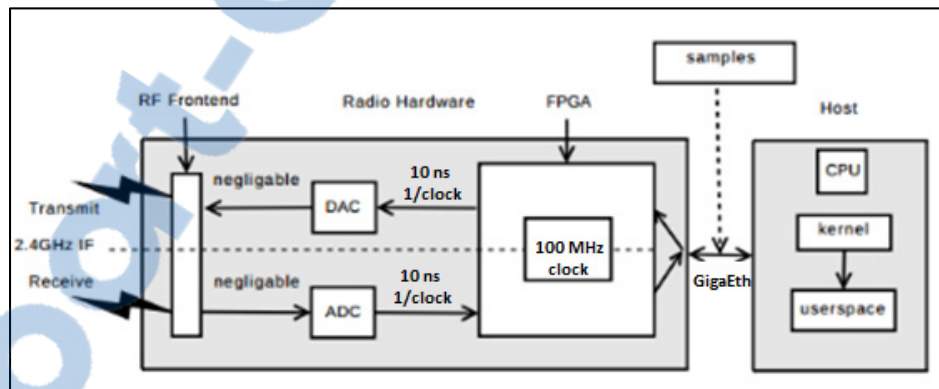


Figure 5.2 Latences introduites par l'architecture de l'USRP N210 [73]

Selon la documentation d' Ettus et le forum GNU-Radio [71], si l'application visée a des exigences strictes de latence, on doit prendre en considération le temps requis par l'hôte pour alimenter le medium Ethernet en paquets. Ce temps est proportionnel au taux d'échantillonnage qu'on configure lors d'exécution des scripts. Par conséquent, pour diminuer la latence en réception, la couche transport doit être configurée pour diminuer au plus le nombre d'interruptions causé par le cœur du système d'exploitation afin de manipuler

la file d'attente des paquets reçu par la carte réseau. D'où l'augmentation de la taille des buffers via les commandes Bash suivantes :

```
$sudo sysctl -w net.core.rmem_max = 50000000
```

```
$sudo sysctl -w net.core.wmem_max = 1048576
```

Au Tableau 5.1, on résume les spécificités du système d'exploitation, de la machine hôte et des outils de développement dont on a fait l'usage tout au long de ce projet

Tableau 5.1 Spécificités de l'hôte et des outils de développement du projet

Système d'exploitation	Ubuntu 12.04 LTS
Version de GNU-Radio	3.6.5
Version d'UHD	3.5.3
Mémoire RAM	6 GigaOctet

Une des expériences menées est l'estimation en temps réel de la latence par le biais de la commande *ping*, de la portion hôte-USRP. Selon la documentation du système d'opération Linux, ce test est à même de donner une valeur de l'ordre des millisecondes. Au Tableau 5.2, on illustre les champs de résultats enregistrés par le système en temps réel. Les tests ont montrés que la valeur moyenne de RTT ou *Round Trip Time* entre l'hôte et l'USRP est de l'ordre de 1.1 millisecondes comme le rapporte l'annexe III.

Tableau 5.2 Champs des résultats collectés pour le post-traitement

Adresse Mac Tx	Horodatage_{Tx} [s, nsec]	Délai Tx (nsec)	Horodatage_{Rx} [s, nsec]	Délai Rx (nsec)	RSS (dBm)
---------------------------	--	----------------------------	--	----------------------------	----------------------

Avec un horodatage logiciel, la valeur obtenue ne garantit pas une précision suffisante pour la localisation d'intérieur vu le caractère aléatoire prédominant de ces estimés. En même temps que l'on récupère l'écho ou *ping* d'un paquet, le paquet suivant aura une valeur différente. Ceci dit, même avec cette quantification, l'estimée en temps réel de la latence au noyau même du système d'exploitation n'est pas encore obtenue.

Greffer ce test d'écho-réponse à l'instant où l'on insère l'horodatage dans le paquet à transmettre s'avère donc une piste non-exploitable pour répondre à l'exigence de précision admissible. D'autre part, l'instruction *ping* fonctionne à la voie asynchrone de la radio USRP et donc ne réfère pas à la voie la plus rapide de traitement de donnée. En fait, une réponse à l'écho *ping* est générée, non pas au niveau du FPGA, mais à partir d'un processeur GPP embarqué à la carte mère qui n'est pas en communication avec les paquets envoyés depuis l'hôte vers l'USRP [80].

5.3 Interprétation des résultats rapportés par le module de post-traitement

Ayant reconstruit le contenu de l'unité de post-traitement dénotée dans [56] par PPLD, la calibration telle que rapportée a été examinée de près. La base de données fonctionne selon le principe illustré au diagramme à la Figure 5.3.

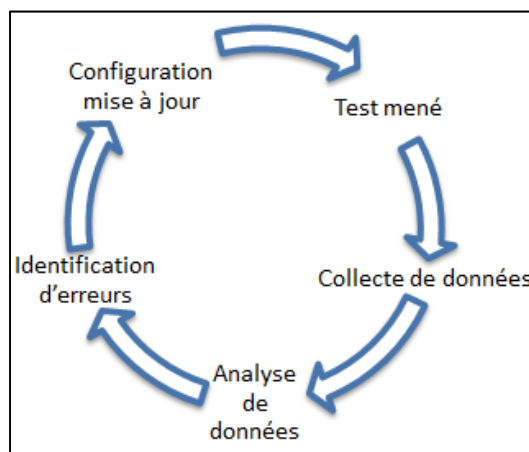


Figure 5.3 Principe du procédé de calibration

On illustre au Tableau 5.3 les tables constituant la PPLD

Tableau 5.3 Description des tables de la base de données de post-traitement

Tables	Description
PPLD_Master_Table (le point d'entrée)	Toutes les données importées associés à tous les projets.
Input_Controller	Les fonctions de contrôle servant aux <i>initialisations</i> (ex. décalages de synchronisation, lieu de test (lors de la calibration))
Environment_Table	Liste des environnements et leur gradient de perte de trajet.
Mac_Table	Liste des adresses Mac précédemment saisies et leur emplacement (X, Y, Z).
BSSID_Frequency_Offset	Liste des ID des stations de base attribué à chaque adresse MAC et les décalages de fréquence de chacune.
Master_Distance_Calculator	Les distances prédites calculées et les erreurs identifiées.
Calibration_RSS_ToA	Liste des erreurs utilisées pour suivre l'erreur moyenne de la distance du test en cours.

Parmi ces tables, on distingue trois importantes à décrire davantage : La "*PPLD_Master_Table*" est la table qui contient les résultats bruts des fichiers CSV générés et remplis dynamiquement par le module de localisation en temps réel. Une fois ces résultats importés, on passe à l'exécution des requêtes "*Distance vs RSS Measurement*" et "*Distance vs Toffset Comparison*" qui mettent à jour les deux tables "*Master_Distance_Calculator*" et "*Calibration_RSS_ToA*". Au Tableau 5.4, on décrit les requêtes régissant la configuration des paramètres d'amélioration et de correction des multiples erreurs du système.

Tableau 5.4 Description des requêtes de la base de données de post-traitement

Requête	Description
Actual Distance Calculations	Compare les distances mesurées aux distances post-traitées.
Actual Time Delay Calculations	Compare les délais temporels.
Calibration RSS ToA	Contient les résultats de calibration.
Auto Correct Distance Filter	Corrige les distances en se basant sur l'environnement et l'erreur.
Distance vs RSS Measurement	Contient les distances en fonction des RSS.
Distance vs Toffset Comparison	Compare les divers décalages temporels.
Mac Specific Time Statistics	Contient les décalages temporels moyens par adresse Mac.
Pathloss interpretation	Déterminer la pente et l'ordonnée à l'origine du coefficient de perte de trajet.
ToA/RSS Average distance Error	Calcule les erreurs moyennes associées aux distances.
Localization	Calcule séquentiellement en différé la position de l'émetteur.

À l'annexe IV, on illustre un exemple type de calibration du système de localisation. Ce dernier consiste en le calcul des décalages inhérents entre une radio en mode émetteur et une en mode récepteur en sachant à priori le délai qui devrait exister. Ce même exemple est reconduit au début de chaque scénario de localisation à exécuter et ce pour chaque couple de radio Tx-Rx. Tout compte fait, en réalisant une multitude de tests avec la PPLD, on se rend compte que le mécanisme de calibration pour les distances de 1 à 10 mètres, une fois élaboré produit un ensemble de correctifs. Ces derniers lorsqu'ils sont exploités par un scénario de localisation sans connaissance au préalable, ne donnent pas de résultats admissibles.

5.4 Améliorations apportées

Étant donné l'ensemble des imprécisions inadmissibles du système antérieur, on se fixe l'objectif d'apporter un nouveau module d'estimation des RSS, de concevoir une solution d'horodatage garantissant une résolution de 10 nanosecondes et finalement de suggérer une meilleure alternative au mécanisme de synchronisation des machines hôtes déjà en place.

5.4.1 Amélioration du module d'estimation des RSS

La méthode déjà implémentée avec *BBN80211*, donne des valeurs de faible résolution. À la Figure 5.4, on illustre la distribution des valeurs de RSS mesurées par le système. Selon le modèle illustré dans **Erreur ! Source du renvoi introuvable.**, on trace l'estimée de distance moyenne à la Figure 5.5.

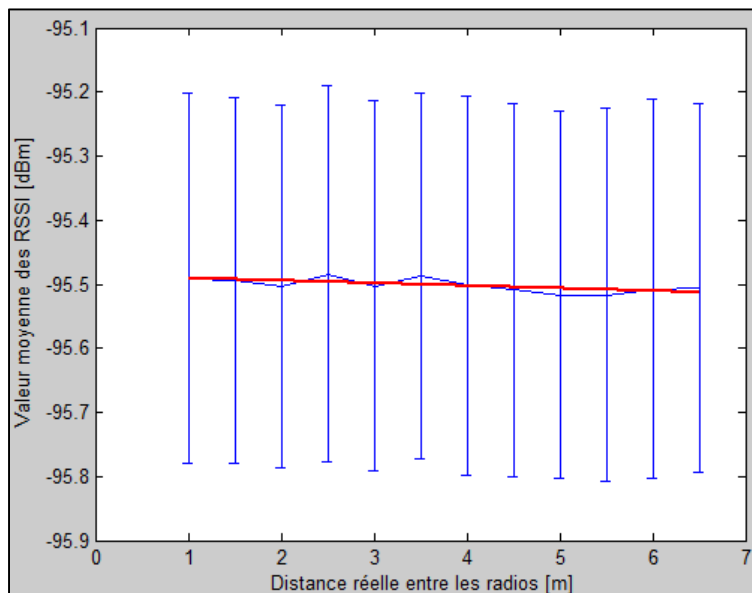


Figure 5.4 RSS en fonction de la distance entre les USRP

À l'équation **Erreur ! Source du renvoi introuvable.**, on montre l'estimée de distance en se basant sur les RSSI mesurés :

$$RSSI [dBm] = -10 * pathloss_{Grad} * \log_{10}(distance[m]) + RSSI_{calib}[dBm] \quad (5.2)$$

$$Distance_{par\ RSSI} = \left(distance_{calib} * 10^{\left(\frac{pathloss[dB] - pathloss_{calib}[dB]}{10 * pathloss_{Grad}} \right)} \right) \quad (5.3)$$

Où:

- $pathloss_{Grad}$: le gradient du milieu caractérisant la perte de trajet. Pour notre cas du milieu d'intérieur on l'a considéré constant et égal à 3.5 ;
- $pathloss$: la différence entre la puissance du signal transmis en dBm et celle du signal reçu ou RSSI ;
- $pathloss_{ref}$: la différence entre la puissance du signal transmis et celle du signal reçu lors de la phase de calibration ;
- $distance_{calib}$: la distance entre les radios lors de la phase de calibration ;

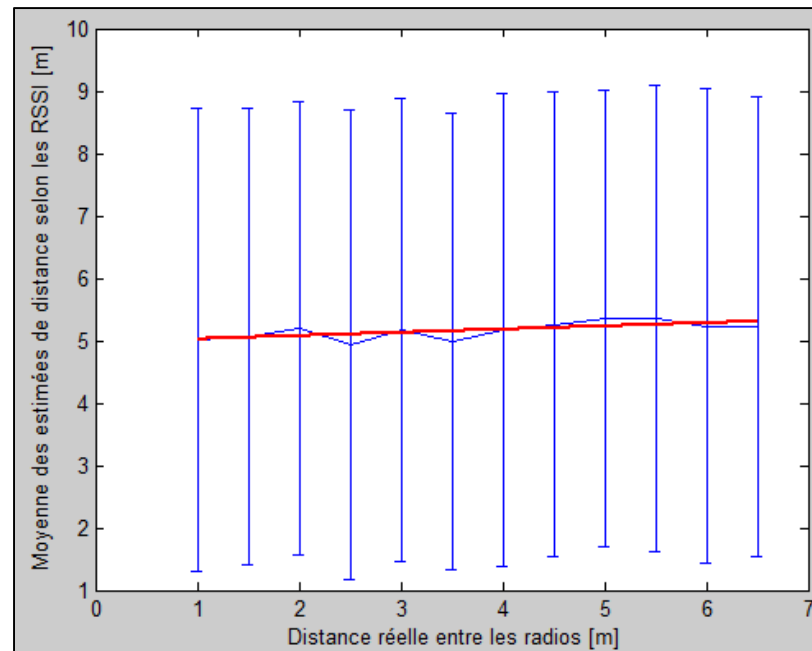


Figure 5.5 Estimées de distances en fonction de la distance entre les USRP

Le modèle de perte de trajet implémenté pour l'estimation de distance par RSS est un modèle basique et ne peut traduire de manière fiable et précise les variations de l'environnement. Seul l'effet de la distance y est considéré. Ni les évanouissements de signal, ni le phénomène d'obscurité ne sont pris en considération. On remarque aussi clairement que les valeurs rapportées ne comptent que pour des légères variations enregistrées au niveau des deux chiffres après la virgule.

À cet effet, on suggère d'utiliser un gradient de perte de trajet variable, et on présente à la section suivante un module d'estimation de RSS plus fiable et réaliste.

Selon l'un des affichages dans la liste de diffusion de GNU-Radio [80], on peut retrouver la valeur de RSS depuis un signal reçu par le calcul de la fonction (5.4)

$$10 * \text{Log}_{10}(\text{Decimate}(\text{Single_Pole_IIR_Filter}(\text{Complex_To_Mag}^2(\text{Signal})))) \quad (5.4)$$

Pour cela, on met en place un diagramme de flux avec l'outil graphique GRC de GNU-Radio traduisant la fonction précédente et on récupère les résultats enregistrés dans un fichier.

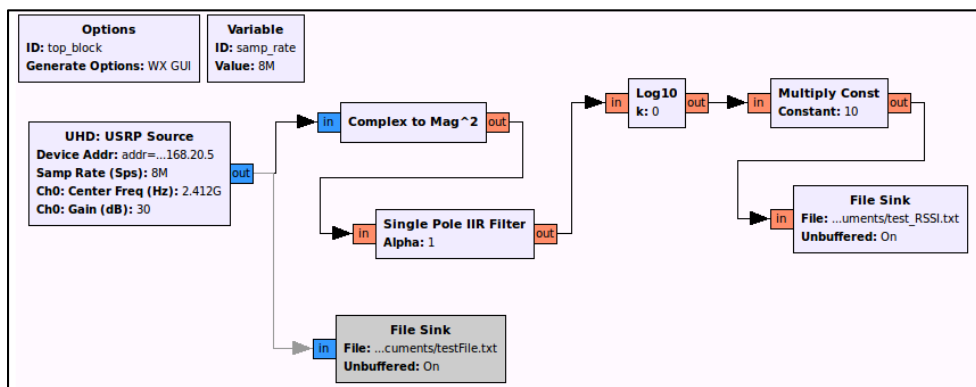


Figure 5.6 Diagramme de flux GRC du nouveau module d'estimation de RSS

Étant donné les contraintes de temps pour ce projet, on se restreint à la conception de cette alternative de calcul de RSS. L'implémentation et le test sont laissés pour des travaux futurs.

5.4.2 Mise à jour pour la compatibilité avec l'API C++ d'UHD

Afin d'exploiter les fonctionnalités avancées qu'offre le FPGA dans tout ce qui a trait à la stabilité de son horloge et le caractère déterministe qu'offre le traitement numérique, et pour pouvoir commander l'envoi et la réception à des instants donnés, on opte pour la migration du code source des chaînes d'émission et de réception vers l'API d'UHD en C++.

Parmi les traits avancés offerts par cette migration on note l'exploitation d'un compteur interne du FPGA qui permet aux échantillons entrants venant de l'hôte d'être envoyés à des instants strictement définissables. Pour cet effet, l'API C++ a permis de faire l'usage du mécanisme de propagation des étiquettes du flux ou *Stream tags* tel qu'illustré à la Figure 5.7. En effet, l'UHD permet de manipuler et exploiter ces métadonnées véhiculées simultanément avec les données en question [71].

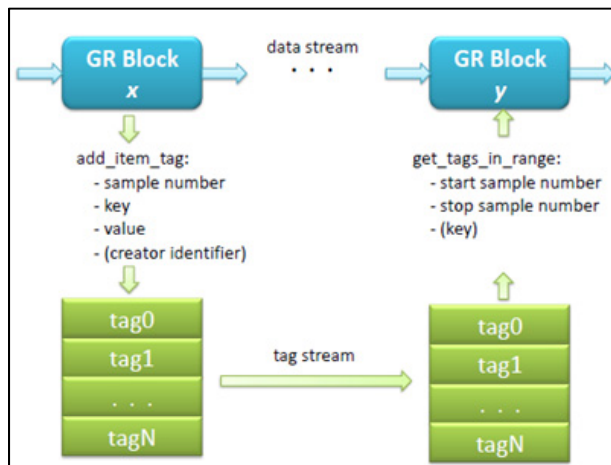


Figure 5.7 Propagation des étiquettes comprenant l'horodatage logiciel [71]

Plus précisément, dans notre application, on a besoin de connaître la valeur du registre de l'instant actuel du FPGA lors de l'envoi d'un paquet. Cette fonctionnalité est garantie via la commande `get_time_now()`. Ce registre sera accédé par lecture. On doit aussi régler, au départ de la transmission, la valeur de l'instant courant via la commande `set_time_now(foo)`.

Une autre fonctionnalité non moins importante est celle qui est assurée par la commande *set_time_next_pps (foo)*. Cette dernière assure le verrouillage du FPGA à l'instant "foo" au prochain front du PPS ou *pulse per second*.

Ainsi, on retranscrit le code source de la totalité des composantes des chaînes d'émission et de réception en C++. Plus précisément, on a développé le cœur des blocs suivants : le générateur d'entête 802.11b, le modulateur configurable en BPSK ou QPSK, la mise en forme configurable avec la séquence de Barker ou avec un cosinus surélevé. De même du côté récepteur, on a développé le code des blocs de démodulation, de reconstruction du signal ainsi que l'interfaçage entre les différents blocs créés, et entre ces mêmes blocs avec ceux précédemment élaborés par *BBN802.11* ainsi que le respect des règles du paradigme du langage orienté objet ont tous exigés un temps considérable durant ce projet.

Afin de vérifier et valider la stabilité des méthodes proposées par l'API en C++ vis-à-vis de notre application, on exécute une boucle de 1000 itérations de la commande *get_time_now()* et l'on trace la distribution des délais entre chaque 2 interrogations successives du registre de temps du FPGA tel qu'illustré à la Figure 5.8:

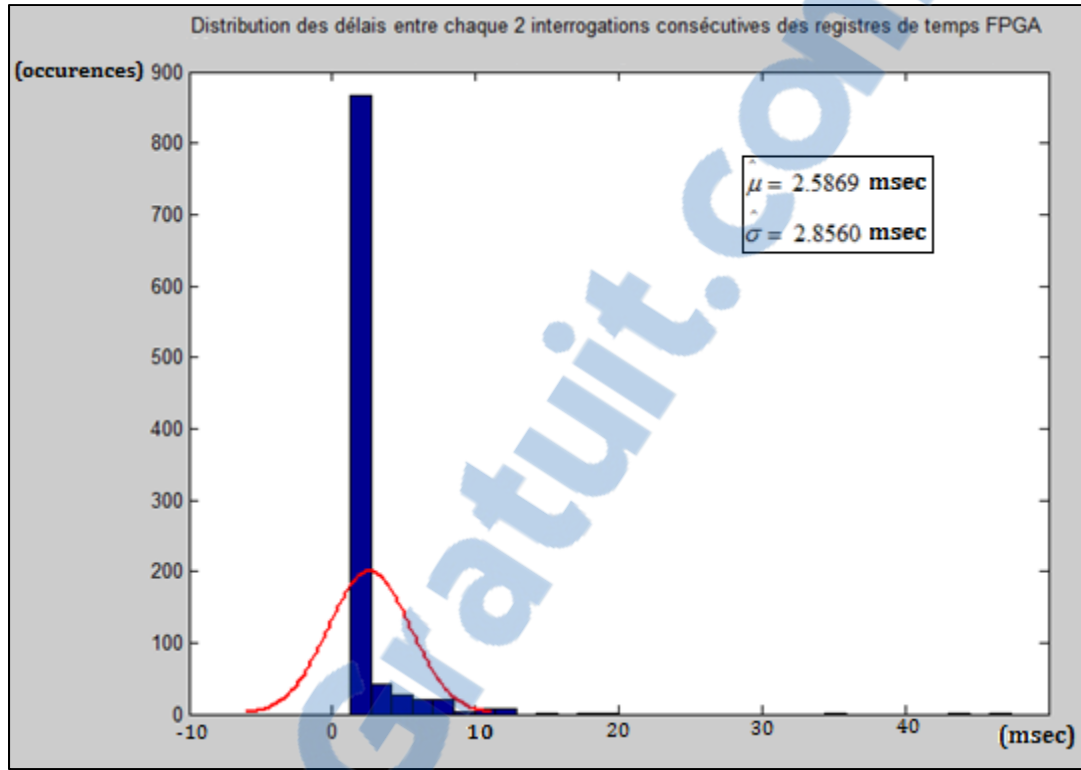


Figure 5.8 Histogramme des délais entre 2 interrogations du temps du FPGA

Selon la figure, la moyenne des délais se situe à l'ordre de 2.58 ms et l'écart-type est de 2.85 ms. Étant donné ces valeurs, on conclut qu'il est impossible, statistiquement parlant, de définir une loi régissant la variabilité des délais dans la section séparant l'hôte de l'USR. En conséquence, ni les résultats enregistrés par la propagation des étiquettes de flux, ni l'interrogation et insertion en temps réel du temps du FPGA n'ont garanti le niveau de précision nécessaire pour l'objectif de ce projet. En effet, la résolution temporelle résultante est toujours variable à un ordre très supérieur aux fins de ce projet. Sur ce, on se tourne vers une méthode d'estimation des instants d'envoi et de réception au niveau matériel du FPGA.

5.4.3 Conception de la solution d'horodatage au sein du FPGA

Comme dans tous les systèmes sensibles au temps, la principale source d'erreur est la latence au niveau du système. Ces retards importants du matériel doivent être comptabilisés avec une

précision d'ordre égale à celui des objectifs de localisation. La logique de résolution de latence, qu'elle soit mise en œuvre au sein du FPGA ou en logiciel sur la machine hôte, introduit du retard de mesure et de calcul.

Néanmoins, lorsque la latence est estimée au sein du matériel, elle connaît moins de variabilité face à l'ordre des délais admissibles par la localisation d'intérieur. En effet, cette dernière exige une résolution de l'ordre de 10 nanosecondes afin de garantir une résolution spatiale de 3.3 mètres. Une latence estimée au niveau matériel est facilement compensée lors d'un post-traitement [64]. Cette latence est fonction du nombre d'impulsions d'horloges qui se sont produites depuis l'arrivée du signal vers son émission par l'antenne.

Pour implémenter un horodatage précis au niveau matériel il importe d'étudier la structure d'entêtes des paquets suivant la norme 802.11b. En effet, il existe deux types d'entêtes associés à la sous-couche PLCP : un long entête PPDU ou *PLCP Protocol Data Unit* illustré à la Figure 5.9 et un court entête PPDU illustré à la Figure 5.10. Dans le format de l'entête long, le champ SYNC est mélangé ou *scrambled* avec des uns, tandis que pour l'entête court il est mélangé avec des zéros.

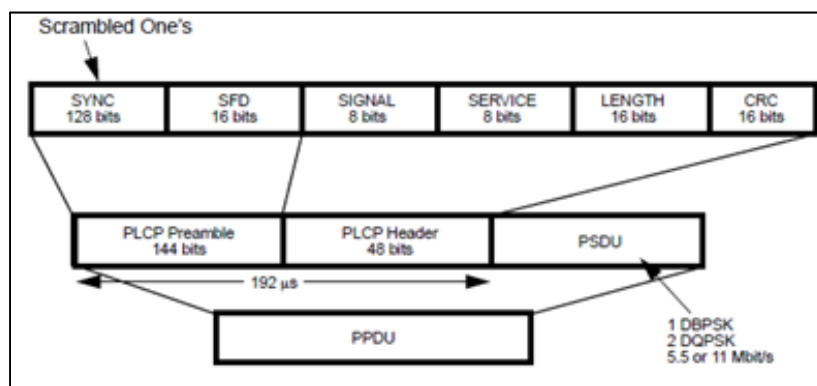


Figure 5.9 Format d'un long entête PPDU pour trame 802.11b [86]

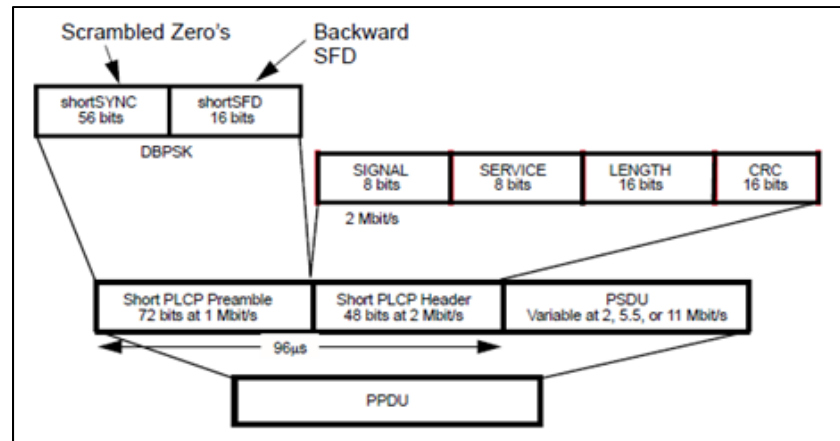


Figure 5.10 Format d'un court entête PDU pour trame 802.11b [86]

Comme le montre la Figure 5.9 et la Figure 5.10, le préambule est composé d'un champ SYNC et d'un champ SFD. Le premier est principalement utilisé pour assurer la synchronisation entre l'émetteur et le récepteur et peut être utilisé pour estimer le canal. Le champ SFD (*Start of Frame Delimiter*) indique que les données suivantes constituent le début de l'entête. Une fois le SFD détecté, le récepteur sait qu'un paquet de données a été reçu. L'entête 802.11b comprend les champs : Signal, Service, Longueur et CRC. Le champ *Signal* : indique le taux de transmission qui est utilisé pour la PDU. Le champ *Service* : conçu pour accroître la flexibilité du PDU dans le futur. Le champ *Length* : informe le récepteur de la durée nécessaire pour envoyer le PDU et non pas la longueur réelle des données comme dans 802.11a. Le champ *CRC* : utilisé pour vérifier si les 3 champs précédents (*Signal, Service, Length*) sont corrects. Il est utilisé pour épargner le circuit de bande de base les démodulations incorrectes.

La solution proposée consiste en l'horodatage de l'instant de détection du début d'une trame ou SFD. Ceci dit, peu importe l'entête utilisé, le champ SFD est le même pour les toutes les trames 802.11b. La Figure 5.11 illustre le principe de la solution proposée pour aboutir au niveau de précision désiré.

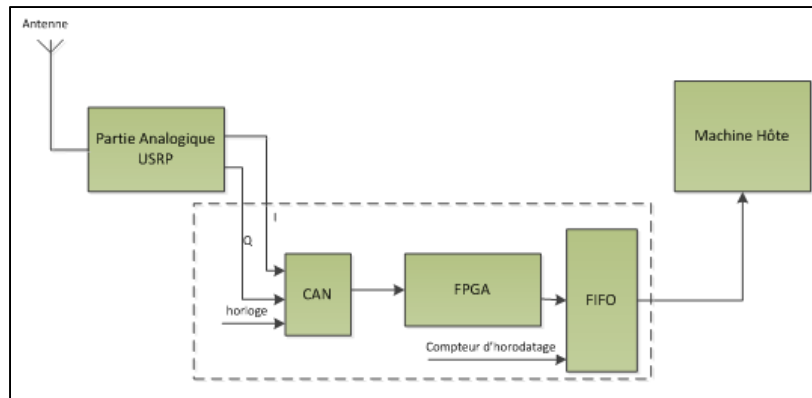


Figure 5.11 Schéma bloc de la solution d'horodatage au sein du FPGA

Pour ce faire, on procède comme suit :

Les données analogiques reçues par le *front-end* convertis à la fréquence intermédiaire, échantillonnées et numérisées à la sortie du CAN sont enregistrées dans un tampon FIFO. Ces mêmes données sont entrées dans un corrélateur cherchant l'entête de la trame 802.11b. Une fois un tel entête trouvé, l'instant présent ou horodatage est envoyé à un tampon FIFO de sortie pour la transmission via l'interface GigaEthernet vers l'ordinateur hôte. L'adresse MAC de l'émetteur du signal peut être décodé et mise à la sortie de l'USRP. Ainsi le contenu des données véhiculées sera structuré en une séquence de ((horodatage, adresse MAC source), (horodatage, adresse MAC source),...).

L'unique traitement en temps réel est l'horodatage de la détection du SFD; tout autre traitement peut être fait de façon asynchrone ultérieurement. En outre, les effets des rayons multi-trajets ne sont pas pris en compte par le corrélateur. Ce dernier cherche la première occurrence du début d'une nouvelle trame. À la machine hôte, tous les horodateurs venant de plusieurs récepteurs pour la même trame sont traitées, afin de localiser l'émetteur cible.

Lorsqu'un signal arrive au niveau du récepteur, une convolution est effectuée entre le signal entrant et une chaîne de caractères prédéfinie. On peut utiliser la détection d'une valeur de crête pour savoir quand le SFD est détecté. Ce modèle de référence est stocké dans un registre à décalage et ne sera pas modifié pendant le processus de détection. Dans un autre registre à décalage, on stocke le signal entrant dans un tampon FIFO. Pour effectuer la

corrélation croisée on effectue un calcul du OU exclusif ou *XOR* avec le motif de référence prédéfinie comme l'illustre la Figure 5.12.

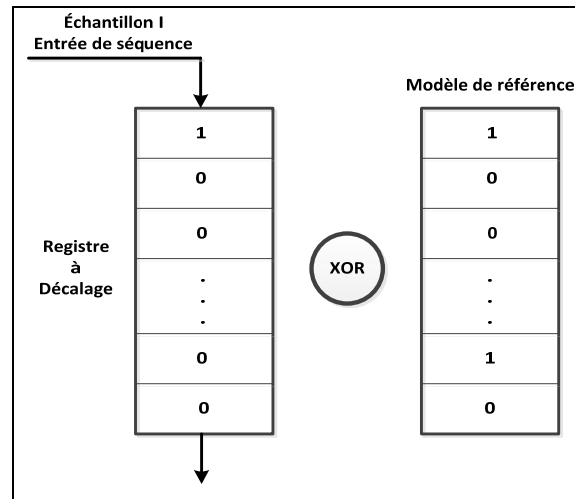


Figure 5.12 Principe du corrélateur conçu

Lorsque la corrélation dépasse un certain seuil, on enregistre l'horodatage et on le place dans un tampon de sortie FIFO pour être envoyé vers la machine hôte. Une fois le modèle prédéfini ou "*match*" trouvé, on remarque un grand pic dans le résultat de corrélation, ce qui indique que l'entête est trouvé.

Pour résumer, la solution consiste à choisir l'événement suivant la synchronisation du récepteur en se basant sur le préambule, comme une indication bien définie de l'arrivée d'une trame. On se propose de détecter par corrélation avec la séquence de puces ou *chips* du modèle de bit du SFD. Cela permet d'avoir une résolution proportionnelle à la fréquence d'échantillonnage de l'horloge principale sur la carte mère qui est de 100 Mch/s plutôt que le débit binaire de 1 ou 2 Mbps. Cela correspond à une résolution temporelle de 10 ns et donc une résolution spatiale d'environ 3.34 m. Ce qui comble l'ordre de grandeur de notre résolution spatiale désirée. À l'annexe V, on présente la résolution mathématique du problème de localisation d'intérieur en ayant obtenu les horodateurs des différents récepteurs aux endroits connus.

Le code Verilog de la solution conçue peut être chargé dans le FPGA. L'écriture de ce code et la génération du fichier binaire correspondant nécessite les outils "*Xilinx ISE Design Suite*". Au Tableau 5.5, on représente les ressources disponibles sur le FPGA de type Spartan 3A DSP 3400 qui est situé sur la carte mère USRP N210 pour les traitements embarqués.

Tableau 5.5 Ressources libres sur le FPGA de N210

Ressource	% de libre
Bascules	59%
LUTs	37%
Mémoire	68%
Ressources DSP	76%

5.4.4 Preuve de concept d'un mécanisme de synchronisation par PTPd

La synchronisation temporelle des récepteurs est actuellement réalisée par le protocole NTP ou *Network Time Protocol* via son démon Linux dénoté NTPD. Ce dernier s'appuie sur la connexion Internet de l'hôte pour régler l'horloge système à l'heure exacte. Le mécanisme garantit une précision de l'ordre des millisecondes et ne prévoit pas une mise à jour fréquente et acceptable de ces dérives étant donné la sensibilité temporelle de l'application. Ceci dit, les paquets véhiculés ne contiennent pas de données de synchronisation. Selon la littérature, *Precision Time Protocol* ou PTP via le démon Linux noté PTPd sert de meilleur outil pour la synchronisation des machines dans un réseau local Ethernet. Suivant la spécificité TDoA de ce projet, le besoin d'avoir une référence temporelle absolue ne se présente pas. Il y a besoin d'une synchronisation d'horloge entre les hôtes des récepteurs. PTPd permet de mesurer les décalages d'horloge en temps réel, les récupérer directement et les introduit dans l'enregistreur ou *logger* du système d'exploitation. À la Figure 5.13, on illustre les champs des résultats de synchronisation d'une machine hôte maître avec une machine hôte esclave.


```

root@naceur... * root@naceur... * root@naceur... * root@naceur... * root@naceur... * root@naceur... * root@naceur... * root@naceur... * root@naceur... * root@naceur...
# Timestamp, State, Clock ID, One Way Delay, Offset From Master, Slave to Master, Master to Slave, Drift, Last packet Received
2013-11-22 20:57:48.729145, init
2013-11-22 20:57:48.794313, lstn_init 1
2013-11-22 20:58:00.730616, mst 78acc0fffea67a4a(unknown)/00
Master

root@naceur-Aspire-5755: /home/naceur/Documents/Testing_directory/Latency/ptpd-2.2.0/src
root@naceur... * root@naceur... * root@naceur... * root@naceur... * root@naceur... * root@naceur... * root@naceur... * root@naceur... * root@naceur... * root@naceur...
# Timestamp State Clock ID One Way Delay Offset From Master Slave to Master Master to Slave Drift Last packet Received
13 2013-11-22 20:58:13.148388, init
14 2013-11-22 20:58:13.211299, lstn_init 1
15 2013-11-22 20:58:25.211357, lstn_init 1
16 2013-11-22 20:58:37.211349, lstn_init 1
17 2013-11-22 20:58:49.211320, lstn_init 1
18 2013-11-22 20:58:01.211362, lstn_init 1
19 2013-11-22 20:58:13.211321, lstn_init 1
20 2013-11-22 20:58:25.211290, lstn_init 1
21 2013-11-22 20:58:37.211360, lstn_init 1
22 2013-11-22 20:58:49.211362, lstn_init 1
23 2013-11-22 20:58:01.211240, lstn_init 1
24 2013-11-22 20:58:02.726170, slv 78acc0fffea67a4a(unknown)/01, 0.000000000, 0.000000000, 0.000000000, 0.000000000, 0, I
25 2013-11-22 20:58:03.726227, slv 78acc0fffea67a4a(unknown)/01, 0.000000000, -0.002322923, 0.000000000, -0.004645847, -2322, S
26 2013-11-22 20:58:04.711637, slv 78acc0fffea67a4a(unknown)/01, 0.000006049, -0.002322923, 0.004670043, -0.004645847, -2322, D
27 2013-11-22 20:58:04.726388, slv 78acc0fffea67a4a(unknown)/01, 0.000006049, -0.004516281, 0.004670043, -0.004380668, -6838, S
28 2013-11-22 20:58:05.211708, slv 78acc0fffea67a4a(unknown)/01, 0.000011829, -0.004516281, 0.004426916, -0.004380668, -6838, S
29 2013-11-22 20:58:05.726852, slv 78acc0fffea67a4a(unknown)/01, 0.000011829, -0.004137031, 0.004426916, -0.003875517, -10975, D
30 2013-11-22 20:58:06.727288, slv 78acc0fffea67a4a(unknown)/01, 0.000011829, -0.003678283, 0.004426916, -0.003457392, -14653, S
31 2013-11-22 20:58:06.961870, slv 78acc0fffea67a4a(unknown)/01, 0.000033353, -0.003678283, 0.003716750, -0.003457392, -14653, D
32 2013-11-22 20:58:07.727747, slv 78acc0fffea67a4a(unknown)/01, 0.000033353, -0.003263909, 0.003716750, -0.003025246, -17916, S
33 2013-11-22 20:58:08.728028, slv 78acc0fffea67a4a(unknown)/01, 0.000033353, -0.002907295, 0.003716750, -0.002722639, -20823, S
34 2013-11-22 20:58:08.899133, slv 78acc0fffea67a4a(unknown)/01, 0.000055891, -0.002907295, 0.002957316, -0.002722639, -20823, D
35 2013-11-22 20:58:09.728386, slv 78acc0fffea67a4a(unknown)/01, 0.000055891, -0.002589038, 0.002957316, -0.002366193, -23412, S
36 2013-11-22 20:58:10.149260, slv 78acc0fffea67a4a(unknown)/01, 0.000070311, -0.002589038, 0.002643512, -0.002366193, -23412, D
37 2013-11-22 20:58:10.211646, slv 78acc0fffea67a4a(unknown)/01, 0.000077954, -0.002589038, 0.002553566, -0.002366193, -23412, D
38 2013-11-22 20:58:10.728596, slv 78acc0fffea67a4a(unknown)/01, 0.000077954, -0.002314821, 0.002553566, -0.002129605, -25726, S

```

Figure 5.13 Décalages et dérives d'horloges entre les machines selon PTPd

5.5 Conclusion

Ce chapitre englobe les travaux réalisés pour apporter les améliorations au système de localisation. On commence par examiner les portions du système et leurs différents délais. On décrit ensuite le module de post-traitement. On s'arrête sur les points critiques à améliorer pour apporter ensuite un ensemble de techniques intégré au système que l'on teste en essayant d'aboutir à un résultat acceptable pour la localisation d'intérieur. Finalement, on suggère une solution basée sur l'horodatage matériel sur le FPGA de l'USRP dont on conçoit le mécanisme et prédit les éventuels résultats.

CONCLUSION

La contribution du présent travail touche à la fois les aspects de radio positionnement et les outils de prototypage des radios définies par logiciel. Ceci dit, la plateforme et l'algorithme de localisation élaborés peuvent servir de référence pour les travaux futurs du LaCIME. En somme, un système de localisation sans fil pour les environnements d'intérieur combinant les informations de ToA et de RSS a été conçu, testé et implémenté sur SDR. Les mesures à priori, se sont avérées très imprécises. Un horodatage de l'ordre de dix nanosecondes ne peut avoir lieu sans le recours au traitement numérique du FPGA. En effet, les imprécisions sont difficiles à quantifier en temps réel étant donné le caractère aléatoire du délai de la portion Ethernet-hôte ainsi que du caractère préemptif du système d'exploitation. L'erreur considérable est due au délai entre l'instant d'insertion de l'horodatage et l'instant d'envoi du paquet. Pour cela, on a suggéré et conçu un mécanisme d'horodatage matériel capable de combler les erreurs de la solution en logiciel. Outre, la synchronisation entre les capteurs constitue elle aussi l'un des défis majeur de ce projet. Pour conclure, on réalise que les méthodes de localisation d'intérieur basées sur les délais temporels sont très sensibles à la synchronisation entre les récepteurs et aux latences inhérentes de chacune de ces radios. Par contre, la méthode exploitant les RSS se révèle être un meilleur compromis, bien qu'elle soit sensible aux conditions de l'environnement. Le système est à présent entièrement compatible avec la série N des radios. L'ajout de traits significatifs au code source modulaire du protocole *BBN802.11* a ouvert la voie pour la communauté de recherche dans la localisation ainsi que celle des SDR vers de nouveaux défis. Les recherches utilisant les plateformes SDR ont considérablement boosté et poussé d'un cran les perspectives et les attentes du domaine de télécommunications. La maturité des implémentations de ces applications demeure l'objectif essentiel de la communauté de recherche sur les SDR. L'entière reconfigurabilité et l'accès aux codes sources libres, a donné à ces radios une notoriété grandissante. Les SDR permettent à la fois le prototypage rapide ainsi que des approches adaptatives et cognitives. Le sujet de localisation d'intérieur demeure un défi de recherche, en restant une priorité pour maintes applications.

RECOMMANDATIONS

Afin d'accroître la précision et réaliser plus de mesures pour valider la robustesse du système, on recommande plusieurs éléments qui doivent être abordés et améliorés. Tout d'abord, la sensibilité des horodateurs se révèle être le plus grand obstacle à l'obtention des mesures de distance précises. Ensuite, on note que la synchronisation est également cruciale lors de l'examen d'un tel système. Encore une fois, il est très difficile d'obtenir une synchronisation entre l'émetteur et chacun des capteurs. La mise en place d'un réseau maillé point-à-point entre les différents récepteurs entre eux pourrait aboutir à une synchronisation fiable pour les algorithmes de TDoA. Un algorithme avancé qui examine plusieurs paquets à la fois pourrait fournir plus de précision.

ANNEXE I

NOUVEAU CODE SOURCE ÉMETTEUR « neoTx.py »

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-
#
# Modified by: Naceur El Ouni
#
# Copyright 2004,2005 Free Software Foundation, Inc.
#
# Copyright (c) 2006 BBN Technologies Corp. All rights reserved.
# Effort sponsored in part by the Defense Advanced Research Projects
# Agency (DARPA) and the Department of the Interior National Business
# Center under agreement number NBCHC050166.
#
# This file is part of GNU Radio
#
#####
from gnuradio import gr, gru, blks2, eng_notation
from gnuradio import uhd
from gnuradio.eng_option import eng_option
from bbn_80211b_pkt import *
from optparse import OptionParser

import random, time, struct, sys
#####

class bbn_80211b_transmit_path(gr.top_block):
    def __init__(self, args, verbose, freq, antenna, gain, samp_rate, spb, barker, mac_addr, otw):
        gr.top_block.__init__(self)

        self.u = uhd.usrp_sink(device_addr = args , stream_args=uhd.stream_args( cpu_format="fc32",
otw_format=otw))
        self.u.set_clock_config(uhd.clock_config.internal(), uhd.ALL_MBOARDS)

        self._spb = spb
        self.normal_gain = 1.0

        # transmitter
        self.packet_transmitter = bbn_80211b_mod_pkts(spb=spb, alpha=0.5, gain=self.normal_gain,
use_barker=barker)

        # Set the antenna
        if(antenna):
            self.u.set_antenna(antenna, 0)
```

```

self.u.set_samp_rate(samp_rate)
rate = self.u.get_samp_rate()

# Set the center frequency
self.u.set_center_freq(freq, 0)
frequency = self.u.get_center_freq()

# Set the gain : if no gain was specified, use the mid-point in dB
if gain is None:
    g = self.u.get_gain_range()
    gain = float(g.start()+g.stop())/2
self.u.set_gain(gain)

self.gr_throttle_0 = gr.throttle(gr.sizeof_gr_complex, samp_rate)

self.connect(self.packet_transmitter,self.gr_throttle_0, self.u)

"""
Prints information about the UHD transmitter
"""
print "\nUHD transmitter:"
print "*****\n"
print "IP @:    %s" % (args)
print "MAC @:    %s" % (mac_addr)
print "Freq:    %sHz" % (eng_notation.num_to_str(frequency))
print "Sample Rate: %s samples per sec." % (eng_notation.num_to_str(rate))
print "Spb:    %s samples per symb." % (spb)
print "Antenna:  %s" % (antenna)
print "Wire-Format: %s" % (otw)
if barker == True:
    print "Using Barker Code spreading\n"

def send_pkt(self, payload="", send_rate=0, eof=False):
    return self.packet_transmitter.send_pkt(payload, send_rate, eof)

# //////////////////////////////////////
#                               main
# //////////////////////////////////////

def main():

    def send_pkt(payload="", send_rate=1, eof=False):
        tb.send_pkt(payload=payload, send_rate=send_rate, eof=eof)

    parser = OptionParser(option_class=eng_option)
    parser.add_option("-a", "--args", type="string", default="192.168.10.2",
        help="UHD device address args , [default=%default]")

```



```

parser.add_option("-A", "--antenna", type="string", default="TX/RX",
                  help="select Rx Antenna where appropriate")

parser.add_option("-s", "--samp_rate", type="eng_float", default=1e6,
                  help="set sample rate (bandwidth) [default=%default]")

parser.add_option("-g", "--gain", type="eng_float", default=30,
                  help="set gain in dB (default is midpoint)")

parser.add_option("-f", "--freq", type="eng_float", default=2.4e9,
                  help="set frequency to FREQ")

parser.add_option("-S", "--spb", type="int", default=4,
                  help="set samples/ baud [default=%default]")

parser.add_option("-b", "--barker", action="store_true", default=False,
                  help="Use Barker Spreading [default=%default]")

parser.add_option("-p", "--no-crc-check", action="store_true", default=False,
                  help="Check payload crc [default=%default]")

parser.add_option("-v", "--verbose", action="store_true", default=False,
                  help="Verbose Output")

parser.add_option("-m", "--mac-addr", type="string", default="a0:36:fa:25:31:2a",
                  help="use USRPN210 at specified MAC address [default=%default]")

parser.add_option("-r", "--reps", type="int", default=1000,
                  help="Number of packets to send [default=%default]")

parser.add_option("-R", "--modulation", type="int", default=1,
                  help="Modulation scheme DBSPK or DQPSK [default=DBPSK]")

parser.add_option("-w", "--wire-format", type="string", default="sc16",
                  help="Set wire format from USRP [default=%default]")

(options, args) = parser.parse_args()

if len(args) != 0:
    parser.print_help()
    sys.exit(1)

if options.freq < 1e6:
    options.freq *= 1e6
#####
# build the graph

```

```

#####

tb = bbn_80211b_transmit_path(options.args, options.verbose, options.freq,
                              options.antenna, options.gain, options.samp_rate, options.spb,
                              options.barker, options.mac_addr, options.wire_format)

#####
# start flow graph
#####
tb.start() #tb.start(max_noutput_items=5000)
#####
# generate and send packets
#####
n = 0
bit_rate = options.modulation
print ""
while n < options.reps:
    print "Sending pkt ", n
    send_pkt(options.mac_addr + "%.9f" % time.time() , bit_rate, False);
    n = n + 1
    time.sleep(.7)

print ""
time.sleep(1)
send_pkt(eof=True)
tb.wait() # wait for it to finish
#####
if __name__ == '__main__':
    try:
        main()
    except KeyboardInterrupt:
        raise SystemExit
#####

```

ANNEXE II

NOUVEAU CODE SOURCE RÉCEPTEUR « neoRx.py »

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-
#
# Modified by: Naceur El Ouni
#
# Copyright 2004,2005 Free Software Foundation, Inc.
#
# Copyright (c) 2006 BBN Technologies Corp. All rights reserved.
# Effort sponsored in part by the Defense Advanced Research Projects
# Agency (DARPA) and the Department of the Interior National Business
# Center under agreement number NBCHC050166.
#
# This file is part of GNU Radio
#
#####
from gnuradio import gr, gru, uhd, eng_notation, bbn
from gnuradio.eng_option import eng_option
from optparse import OptionParser
from bbn_80211b_pkt import *
from bbn_80211b import *

from grc_gnuradio import wxgui as grc_wxgui
#####
import sys, struct, os, time, math
#####
def rx_callback(ok, payload):
    toa = "%.9f" % time.time();
    size = struct.calcsize("@qHBB");
    packet_data = payload[size:];
    hdr = struct.unpack("@qHbB", payload[0:size]);
    if len(packet_data) > 16:
        data_hdr = struct.unpack("@BBBBBB", packet_data[10:16])

        mac_str = "%02x:%02x:%02x:%02x:%02X:%02X" % \
            (data_hdr[0], data_hdr[1], data_hdr[2],
             data_hdr[3], data_hdr[4], data_hdr[5],)

    else:
        mac_str = "UNKNOWN"

    print ""
    print "#####"
```

```

print "PKT: len=%d, rssi=%d, src=%s, time=%ld, rate=%d Mbps" \
      %(hdr[1], hdr[2], mac_str, hdr[0], hdr[3]/ 10.0)

print "Payload: ", packet_data
macadd = packet_data[0:17]
timeSent = packet_data[17:]
print ""
print "Tx MAC:      ", macadd
print "Tx timestamp: ", timeSent
print "ToA:        ", toa

c=299792458;
tof = float(toa) - float(timeSent)
dist = c*tof
print "ToF:        ", tof
print "Distance:   ", dist

class receive_path(gr.hier_block2):
    def __init__(self, args, verbose, freq, antenna, gain, sample_rate, mac_addr, spb, barker, otw):
gr.hier_block2.__init__(self, "receive_path", gr.io_signature(0, 0, 0), # Input signature,
gr.io_signature(1,1,gr.sizeof_gr_complex)) # Output signature

        self.u = uhd.usrp_source(device_addr = args , stream_args=uhd.stream_args(cpu_format="fc32",
otw_format=otw, channels=range(1)))

        self.u.set_clock_config(uhd.clock_config.internal(), uhd.ALL_MBOARDS)

        # Set the antenna
        if(antenna):
            self.u.set_antenna(antenna, 0)

        # Set the center frequency
        self.u.set_center_freq(freq, 0)

        # Set the gain # if no gain was specified, use the mid-point in dB
        if gain is None:
            g = self.u.get_gain_range()
            gain = float(g.start()+g.stop())/2
        self.u.set_gain(gain)
        self.u.set_samp_rate(sample_rate)
        rate = self.u.get_samp_rate()
        rxantenna = self.u.get_antenna()
        rxgain = self.u.get_gain()
        frequency = self.u.get_center_freq()
        self.connect(self.u, self)

        """"
Prints information about the UHD receiver

```

```

"""
print "\nUHD Receiver:"
print "*****"
print "IP @:    %s "    % (args)
print "MAC @:    %s"    % (mac_addr)
print "Freq:    %s Hz" % (eng_notation.num_to_str(frequency))
print "Gain:    %f dB" % (rxgain)
print "Sample Rate: %s samples per sec." % (eng_notation.num_to_str(rate))
print "Spb:     %s samples per symb." % (spb)
print "Antenna:  %s"    % (rxantenna)
print "Wire-Format: %s" % (otw)
if barker == True:
    print "Using Barker Code spreading\n"

# //////////////////////////////////////
# The Flow Graph
# //////////////////////////////////////
class app_flow_graph(grc_wxgui.top_block_gui):
    def __init__(self):
        grc_wxgui.top_block_gui.__init__(self)

        parser = OptionParser(option_class=eng_option)
        parser.add_option("-a", "--args", type="string", default="192.168.20.5",
                          help="UHD device address args , [default=%default]")

        parser.add_option("-f", "--freq", type="eng_float", default=2.4e9,
                          help="set frequency to FREQ", metavar="FREQ")

        parser.add_option("-g", "--gain", type="eng_float", default=30,
                          help="set gain in dB (default is midpoint)")

        parser.add_option("-S", "--spb", type="int", default=4,
                          help="set samples/ baud [default=%default]")

        parser.add_option("-b", "--barker", action="store_true", default=False,
                          help="Use Barker Spreading [default=%default]")

        parser.add_option("-p", "--no-crc-check", action="store_true", default=False,
                          help="Check payload crc [default=%default]")

        parser.add_option("-v", "--verbose", action="store_true", default=False,
                          help="Verbose Output")

        parser.add_option("-A", "--antenna", type="string", default="RX2",
                          help="select Rx Antenna where appropriate")

        parser.add_option("-s", "--sample_rate", type="eng_float", default=1e6,
                          help="set sample rate (bandwidth) [default=%default]")

```

```

parser.add_option("-w", "--wire-format", type="string", default="sc16",
                  help="Set wire format from USRP [default=%default]")

parser.add_option("-m", "--mac_addr", type="string", default="a0:36:fa:25:32:9b",
                  help="use USRPN210 at specified MAC address [default=None]")

(options, args) = parser.parse_args()

if len(args) != 0:
    parser.print_help()
    sys.exit(1)

self.options = options
#####
# build the graph
#####
self.u = receive_path(options.args, options.verbose, options.freq, options.antenna, options.gain,
options.sample_rate, options.mac_addr, options.spb, options.barker, options.wire_format)

self.bit_receiver = bbn_80211b_demod_pkts (spb = options.spb, alpha = 0.5, use_barker =
options.barker, check_crc = not options.no_crc_check, callback = rx_callback)

self.connect(self.u, self.bit_receiver)
#####
# start flow graph
#####
def main ():
    app = app_flow_graph()
    app.start() #app.start(max_noutput_items=100)
    os.read(0,10)

if __name__ == '__main__':
    main ()

```

ANNEXE III

RÉSULTATS D'HORODATAGE

Tableau-A III-1 Résultats numériques obtenus lors d'un scénario de localisation effectué

Timestamp Tx	Ping Tx	Timestamp Rx	Ping Rx	Délai (nsec)	Estimé par les délais (m)	RSSI (dBm)	Estimé par RSSI (m)
556674957	569000	568704405	545000	10915448	3271360	-95.2539	1.465672132
562908888	555000	565241311	531000	1246423	373553	-95.1328	-0.092708995
564671993	542000	569537341	529000	3794348	1137166	-95.6406	6.428708238
569170952	570000	571601561	565000	1295609	388294	-95.7109	7.328805743
572621107	524000	574954641	557000	1252534	375384	-95.6953	7.12912569
573741913	518000	576160492	530000	1370579	410763	-95.375	3.022073286
576016903	522000	578613048	547000	1527145	457685	-95.5352	5.077961824
576663971	2257000	578997501	1204000	-1127470	-337903	-96	11.02340807
580050945	534000	582857560	535000	1737615	520763	-95.1289	-0.142929297
580666065	542000	584191219	561000	2422154	725920	-95.7773	8.178362042
581780910	568000	589158433	539000	6270523	1879276	-95.2656	1.616129172
584667921	565000	587049259	569000	1247338	373827	-95.6953	7.12912569
586920977	535000	589660524	540000	1664547	498865	-95.4648	4.174925134
589796066	531000	592907816	557000	2023750	606518	-95.75	7.829142894
592057943	562000	594904929	567000	1717986	514880	-95.8242	8.778070303
592675924	568000	595433006	559000	1630082	488536	-95.0547	-1.098795151
594914913	561000	597363765	547000	1340852	401853	-95.1211	-0.243376077
596029997	563000	598594248	563000	1438251	431044	-96	11.02340807
597152948	1621000	599487261	549000	164313	49245	-95.0391	-1.299853786
603959084	571000	606580057	562000	1487973	445946	-95.2969	2.018543082
604671955	561000	607106721	571000	1302766	390439	-95.5781	5.627923558
605792046	584000	608716787	531000	1809741	542379	-95.4141	3.524172207
608046055	524000	610693698	533000	1590643	476716	-95.3867	3.172339298
610902071	557000	613022137	563000	1000066	299720	-95.9648	10.57415903
612009048	580000	614319587	556000	1174539	352009	-95.75	7.829142894
612664938	548000	614754873	562000	979935	293687	-96	11.02340807
613789082	527000	616226071	560000	1349989	404592	-95.1172	-0.293602556
616040945	540000	618583002	549000	1453057	435481	-95.3828	3.122252675
621789932	553000	624436937	544000	1550005	464536	-95.0273	-1.451958525
624032974	564000	626365417	594000	1174443	351981	-95.2812	1.816709817
629754066	577000	632557222	561000	1665156	499047	-95.5469	5.227975876
632649899	557000	634822203	542000	1073304	321669	-95.6562	6.628502485
633745909	547000	636194724	522000	1379815	413531	-95.2461	1.365357171
638223886	576000	640956232	538000	1618346	485018	-95.1914	0.661635458

ANNEXE VI

EXEMPLE TYPE DE CALIBRATION

Before Calibration							Distance(Input)
Mac address	TimeSent(s)	TimeSent(ns)	TimeArvl(s)	TimeArvl(ns)	Time-difference	Distance(m)	DistanceError
A0:36:FA:25:32:98	1356050785	971518040	1356050785	974930972	3412932	1023171.27326686	-1023169.27326686
A0:36:FA:25:32:98	1356050792	980264902	1356050792	981983610	1718708	515255.695904264	-515253.695904264
A0:36:FA:25:32:98	1356050793	681121111	1356050793	682831569	1710458	512782.408125764	-512780.408125764
A0:36:FA:25:32:98	1356050795	82441092	1356050795	85841813	3400721	1019510.50756222	-1019508.50756222
A0:36:FA:25:32:98	1356050795	783318043	1356050795	784789295	1471252	441070.353417416	-441068.253417416
						2342814.2	614143.97697182
Average TOA Results							Initial Variables
Avn Time Dif	Avg_Distance	Avg_Dist_Err	Avg_timingOffset	SD	1- Timing offset supposé nul au départ		Timing Offset
2342814.2	2342814.2	-702356.0276553	-2342807.53	N/A	2- Distance moyenne incorrecte (devrait être égal à 2)		0
Output	Output	Output	Output	Input	3- Les résultats révèlent l'offset qu'on doit introduire après pour corriger		Avg_Timing offset
							0
After ToA Calibration							Distance(Input)
Mac address	TimeSent(s)	TimeSent(ns)	TimeArvl(s)	TimeArvl(ns)	Time-difference	Distance(m)	Distance Error
A0:36:FA:25:32:98	1356050785	971518040	1356050785	974930972	1070124.47	320815.245227247	-320813.245227247
A0:36:FA:25:32:98	1356050792	980264902	1356050792	981983610	-624099.53	-187100.332135345	187102.332135345
A0:36:FA:25:32:98	1356050793	681121111	1356050793	682831569	-632349.53	-189573.619913845	189575.619913845
A0:36:FA:25:32:98	1356050795	82441092	1356050795	85841813	1057913.47	317154.479522609	-317152.479522609
A0:36:FA:25:32:98	1356050795	783318043	1356050795	784789295	-871555.53	-261285.374622193	261287.774622193
						6.6700000002	1.9996156949
Average TOA Results							Initial Variables
Avn Time Dif	Avg_Distance	Avg_Dist_Err	Avg_timing	SD	4- On considère les valeurs trouvées		Timing Offset
6.6700000002	9996156949	0.000384305	-2.0489121510536E-010	N/A	5- La déviation est l'erreur système		2342814.2
Output	Output	Output	Output	Input	6- Le timing offset est minuscule		Avg_Timing offset
							System's error
							7- La distance est corrigée

Figure-A VI-1 Étapes suivies pour la calibration élémentaire du système

ANNEXE V

RÉSOLUTION MATHÉMATIQUE DU PROBLÈME DE LOCALISATION

Dans ce projet, on utilise trois récepteurs pour détecter le signal émis, comme le montre la Figure A V-1. Du point de vue algébrique, l'exigence minimale pour la détection est de trois récepteurs. Évidemment, plusieurs récepteurs peuvent donner une meilleure résolution.

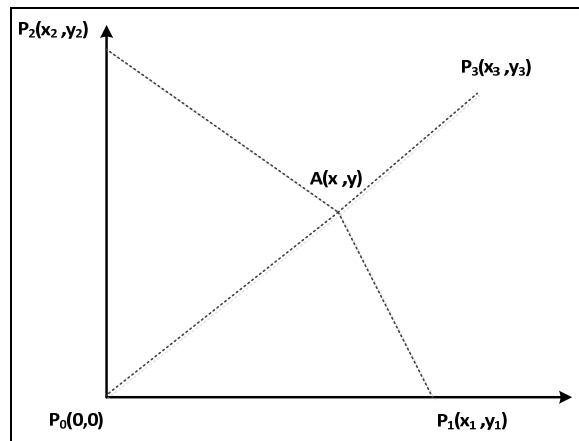


Figure A V-1 Principe mathématique de la localisation de l'émetteur

Étant donné P_0 , P_1 , P_2 et P_3 les emplacements des récepteurs, t est l'instant lorsque le signal a été transmis, et t_0 , t_1 , t_2 et t_3 sont les instants d'arrivée du signal, c la célérité de transmission du signal, qui est de $3 \cdot 10^8$ m/s. La résolution du système d'équations suivant nous donne les coordonnées inconnues de l'émetteur à localiser.

$$x^2 + y^2 = (t_0 - t) * c \quad (\text{A V-1})$$

$$(x_1 - x)^2 + y^2 = (t_1 - t) * c \quad (\text{A V-2})$$

$$x^2 + (y_2 - y)^2 = (t_2 - t) * c \quad (\text{A V-3})$$

$$(x_3 - x)^2 + (y_3 - y)^2 = (t_3 - t) * c \quad (\text{A V-4})$$

LISTE DE RÉFÉRENCES BIBLIOGRAPHIQUES

- [1] J. Caffery and G. Stuber, "Subscriber Location in CDMA Cellular Networks," *IEEE Trans. VT*, vol. 47, no. 2, May 1998, pp. 406–16.
- [2] K. Pahlavan and P. Krishnamurthy, *Principles of Wireless Networks — A Unified Approach*, Prentice Hall, 2002.
- [3] J. Werb and C. Lanzl, "Designing a Positioning System for Finding Things and People Indoors" *IEEE Spec.*, vol. 35, no. 9, Sept. 98, pp. 71–78.
- [4] P. Bahl and V. Padmanabhan, "RADAR: An In-Building RF-Based User Location and Tracking System," *IEEE INFOCOM*, Israel, Mar. 2000, 775 - 784 vol.2
- [5] C. Drane, M. Macnaughtan, and C. Scott. Positioning GSM telephones. *IEEE Communications Magazine*, 36(4):46–54, 1998.
- [6] Design and Deployment of Wireless Networked Embedded Systems.
- [7] J. Zhang. *Handbook of Wireless Networks And Mobile Computing*, chapter Location Management in Cellular Networks, pages 27–50. John Wiley & Sons, New York, 2002.
- [8] CS98 J.J. Caffery and G.L. Stüber. Overview of radiolocation in CDMA cellular systems. *IEEE Communications Magazine*, 36(4):38–45, April 1998.
- [9] Hui, L., H. Darabi, P. Banerjee and L. Jing (2007). "Survey of Wireless Indoor Positioning Techniques and Systems." *Systems, Man, and Cybernetics, Part C: Applications and Reviews*, IEEE Transactions on 37(6): 1067-1080.
- [10] Bahl, P. and V. N. Padmanabhan (2000). RADAR: an in-building RF-based user location and tracking system. *INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings.* IEEE.
- [11] P. Bahl, A. Balachandran, and V. Padmanabhan. Enhancements to the RADAR User Location and Tracking System, Microsoft Research, Technical Report, Feb. 2000.
- [12] Otsason, V., A. Varshavsky, A. LaMarca and E. Lara (2005). Accurate GSM Indoor Localization. *UbiComp 2005: Ubiquitous Computing*. M. Beigl, S. Intille, J. Rekimoto and H. Tokuda, Springer Berlin Heidelberg. 3660: 141-158.

- [13] M. Youssef, A. Agrawala, and A. Udaya Shankar, "WLAN location determination via clustering and probability distributions," *IEEE Int. Conf. Pervasive Comput. Commun.*, Mar. 2003, pp. 143–151.
- [14] M. Youssef, A. K. Agrawala, "Handling samples correlation in the Horus system," *IEEE INFOCOM 2004*, Hong Kong, vol. 2, pp. 1023–1031, Mar. 2004.
- [15] P. Castro, P. Chiu, T. Kremenek, and R. R. Muntz, "A probabilistic room location service for wireless networked environments," in *Proc. 3rd Int. Conf. Ubiquitous Comput.*, Atlanta, GA, Sep. 2001, pp. 18–34.
- [16] R. Battiti, T. L. Nhat, and A. Villani, "Location-aware computing: A neural network model for determining location in wireless LANs," Tech. Rep. DIT-02–0083, 2002.
- [17] P. Kontkanen, P. Myllymäki, T. Roos, H. Tirri, K. Valtonen, and H. Wetteg, "Topics in probabilistic location estimation in wireless networks," in *Proc. 15th IEEE Symp. Pers., Indoor, Mobile Radio Commun.*, Barcelona, Spain, Sep. 2004, vol. 2, pp. 1052–1056.
- [18] S. Thrun, "Probabilistic algorithms in robotics" *AI Mag.*, vol. 21, no. 4, pp. 93–109, 2000.
- [19] AeroScout Company, www.aeroscout.com.
- [20] Topaz local positioning solution. www.tadlys.com.
- [21] A. Kotanen, M. Hannikainen, H. Leppakoski, and T. D. Hamalainen, "Experiments on local positioning with Bluetooth," in *Proc. IEEE Int. Conf. Inf. Technol.: Comput. Commun.*, Apr. 2003, pp. 297–303.
- [22] SnapTrack. www.snaptrack.com.
- [23] Atmel Corporation. www.atmel.com.
- [24] U-blox AG. www.u-blox.com.
- [25] J. Barnes, C. Rizos, J. Wang, D. Small, G. Voigt, and N. Gambale. (2003). Locata: The positioning technology of the future? presented at 6th Int. Symp. Satellite Navig. Technol. Incl. Mobile Positioning Location Services, Melbourne, Australia, pp. 49–62. www.gmat.unsw.edu.au/snap/snap.html.
- [26] N. Bulusu, J. Heidemann, and D. Estrin. GPS-less low-cost outdoor localization for very small devices. *IEEE Personal Communications*, 7(5):28–34, October 2000.

- [27] N. Bulusu. *Selfconfiguring Localization Systems*. PhD thesis, Dept. Computer Sciences, Univ. of California, Los Angeles, CA, 2002.
- [28] L.M. Ni, Y. Liu, Lau Y.C., and A.P. Patil. LANDMARC: Indoor location sensing using active RFID. In *Proc. 1st IEEE Int'l Conf. Pervasive Computing and Communications (PerCom 2003)*, pages 407–415. IEEE CS Press, Los Alamitos, CA, March 2003.
- [29] F. Siegemund and C. Flörkemeier. Interaction in pervasive computing settings using Bluetooth-enabled active tags and passive RFID technology together with mobile phones. In *Proc. 1st IEEE Int'l Conf. Pervasive Computing and Communications (PerCom 2003)*, pages 378–387. IEEE CS Press, Los Alamitos, CA, March 2003.
- [30] J. Hightower, R. Want, and G. Borriello, “SpotON: An indoor 3D location sensing technology based on RF signal strength,” Univ. Washington, Seattle, Tech. Rep. UW CSE 2000–02-02, Feb. 2000.
- [31] L. M. Ni, Y. Liu, Y. C. Lau, and A. P. Patil, “LANDMARC: Indoor location sensing using active RFID,” *Wireless Netw.*, vol. 10, no. 6, pp. 701–710, Nov. 2004.
- [32] A. Hopper, A. Harter, and T. Blackie. The active badge system. In *Proc. Of INTERCHI-93*, pages 533–534, Amsterdam, The Netherlands, 1993.
- [33] Versus Technologies, www.versustech.com.
- [34] N.B. Priyantha, A. Chakraborty, and H. Balakrishnan. The Cricket location support system. In *Proc. 6th ACM/IEEE Ann. Int'l Conf. Mobile Computing and Networking (MobiCom 2000)*, pages 32–43. ACM Press, New York, 2000.
- [35] M. Addlesee, R. Curwen, S. Hodges, J. Newman, P. Steggle, A. Ward, and A. Hopper. Implementing a sentient computing system. *IEEE Computer*, 34(8):50–56, August 2001.
- [36] Ubisense, www.ubisense.net.
- [37] Honore, A. A., R. W. Thomas, R. K. Martin and S. H. Kurkowski (2009). Implementation of collaborative RF localization using a software-defined Radio network. Military Communications Conference, 2009. MILCOM 2009. IEEE.
- [38] B. B. Peterson, C. Kmiecik, R. Hartnett, P. M. Thompson, J. Mendoza, and H. Nguyen, “Spread spectrum indoor geolocation,” *J. Inst. Navigat.*, vol. 45, no. 2, pp. 97–102, 1998.

- [39] X. Li, K. Pahlavan, M. Latva-aho, and M. Ylianttila, "Comparison of indoor geolocation methods in DSSS and OFDM wireless LAN," in *Proc. IEEE Veh. Technol. Conf.*, Sep. 2000, vol. 6, pp. 3015–3020.
- [40] N. S. Correal, S. Kyperountas, Q. Shi, and M. Welborn, "An ultrawideband relative location system," in *Proc. IEEE Conf. Ultra Wideband Syst. Technol.*, Nov. 2003, pp. 394–397.
- [41] B. Fang, "Simple solution for hyperbolic and related position fixes," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 26, no. 5, pp. 748–753, Sep. 1990.
- [42] M. Kanaan and K. Pahlavan, "A comparison of wireless geolocation algorithms in the indoor environment," in *Proc. IEEE Wireless Commun. Netw. Conf.*, 2004, vol. 1, pp. 177–182.
- [43] R. Exel and P. Loschmidt, "High Accurate Timestamping by Phase and Frequency Estimation," in *ISPCS 2009 International IEEE Symposium on Precision Clock Synchronization for Measurement, Control and Communication*, Brescia, Italy, 2009, pp. 126–131.
- [44] D. Torrieri, "Statistical theory of passive location systems," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 20, no. 2, pp. 183–197, Mar. 1984.
- [45] M. Kossel, H. R. Benedickter, R. Peter, and W. Bachtold, "Microwave backscatter modulation systems," *IEEE MTT-S Dig.*, vol. 3, pp. 1427–1430, Jun. 2000.
- [46] A. Gunther and C. Hoene, "Measuring round trip times to determine the distance between WLAN nodes," in *Proc. Netw. 2005.*, Waterloo, ON, Canada, May 2005, pp. 768–79.
- [47] J. Zhou, K. M.-K. Chu, and J. K.-Y. Ng, "Providing location services within a radio cellular network using ellipse propagation model," in *Proc. 19th Int. Conf. Adv. Inf. Netw. Appl.*, Mar. 2005, pp. 559–564.
- [48] A. Teuber and B. Eissfeller, "A two-stage fuzzy logic approach for wireless LAN indoor positioning," in *Proc. IEEE/ION Position Location Navigat. Symp.*, Apr. 2006, vol. 4, pp. 730–738.
- [49] B. D. Van Veen and K. M. Buckley, "Beamforming: A versatile approach to spatial filtering," *IEEE ASSP Mag.*, vol. 5, no. 2, pp. 4–24, Apr. 1988.

- [50] B. Ottersten, M. Viberg, P. Stoica, and A. Nehorai, "Exact and large sample ML techniques for parameter estimation and detection in array processing," in *Radar Array Processing*, S. S. Haykin, J. Litva, and T. J. Shepherd, Eds. New York: Springer-Verlag, 1993, pp. 99–151.
- [51] Ekahau, www.ekahau.com.
- [52] K. Pahlavan and A. Levesque, *Wireless Information Networks*, 2nd ed., John Wiley & Sons, 2005.
- [53] T. S. Rappaport, *Wireless Communications: Principles and Practice*, 2nd ed. , Prentice Hall, Upper Saddle River, NJ, 2002.
- [54] P. Prasithsangaree, P. Krishnamurthy, P.K. Chrysanthis, "On indoor position location with wireless LANs," *IEEE PIMRC2002*, vol. 2, pp. 720 - 724 , Sept. 2002.
- [55] Hatami, Ahmad, PhD Dissertation "Application of Channel Modeling for Indoor Localization Using TOA and RSS", Electrical & Computer Engineering Department.
- [56] Dobbins, Ryan. "Software Defined Radio Localization Using 802.11-style Communications". WORCESTER POLYTECHNIC INSTITUTE.
- [57] J. O. Smith and J. S. Abel, —Closed-Form Least-Squares Source Location Estimation from Range-Difference Measurements,|| *IEEE Trans. on Acous., Speech, Signal Process.*, vol. 35, pp. 1661-1669, Dec 1987.
- [58] P. Sadhukhan and P. K. Das, —MGAL: A Modified Geometry-Assisted Location Estimation Algorithm Reducing Location Estimation Error in 2D Case under NLOS Environments,|| in *Mobile Entity Localization and Tracking in GPS-Less Environments*, Orlando, Fl., USA, 2009, pp. 1-18.
- [59] L. Doherty, K.S.J. Pister, and L. El-Ghaoui. Convex position estimation in wireless sensor networks. In *Proc. 20th Ann. Joint IEEE Conf. Computer Communication Soc. (Infocom 2001)*, volume 3, pages 1655–1663. IEEE, Piscataway, NJ, 2001.
- [60] Data Fusion For Improved TOA/TDOA Position Determination in Wireless Systems by Rahman I. Reza, PhD thesis, Faculty of the Virginia Polytechnic Institute and State University.

- [61] Y. T. Chan, K. C. Ho, .A Simple and Efficient Estimator For Hyperbolic Location., *IEEE Transactions on Signal Processing*, vol. 42, no. 8, pp 1905-1915, August 1994.
- [62] K. Wang, M. Faulkner, J. Singh, and I. Tolochko, —Timing Synchronization for 802.11a WLANs under Multipath Channels,|| Victoria University Centre for Telecommunications and Microelectronics, Melbourne, Australia, 2003.
- [63] H. H. Jenkins. Small-Aperture Radio Direction-Finding. Artech House, Inc., 1991.
- [64] P. Loschmidt, R. Exel, A. Nagy, and G. Gaderer, —Limits of Synchronization Accuracy Using Hardware Support in IEEE 1588,|| in *Int. IEEE Symposium on Precision Clock Synchronization for Measurement, Control, and Communication*, Ann Arbor, MI., USA, 2008, pp. 12-16.
- [65] M. Youssef *et. al.*, —PinPoint: An Asynchronous Time-Based Location Determination System,|| in *MobiSys 2006*, Uppsala, Sweden, 2006, pp. 165-176.
- [66] Pahlavan, K., F. O. Akgul, M. Heidari, A. Hatami, J. M. Elwell and R. D. Tingley (2006). "Indoor geolocation in the absence of direct path." *Wireless Communications*, IEEE 13(6): 50-58.
- [67] D. Valerio, —Open Source Software-Defined Radio: A survey on GNUradio and its applications,|| Forschungszentrum Telekommunikation Wien, Vienna, Austria, Rep. *FTW-TR-2008-002*, 2008.
- [68] Scaperoth, D., 2005, —Cognitive Software Defined Radio.
- [69] Slah, A., 2002, An Introduction to Software Radio, www.signallake.com/innovation.
- [70] Majo Boter, Marcos. "Design and implementation of an OFDM-based communication system for the GNU Radio platform." (2011).
- [71] Ettus, Matt. "Universal software radio peripheral (USRP)." *Ettus Research LLC* www.ettus.com.
- [73] Fähnle, Matthias. 2010. « Software-Defined Radio with GNU Radio and USRP/2 hardware frontend: setup and FM/GSM applications ». *Hochschule Ulm University of Applied Sciences, Institute of Communication Technology*.
- [74] Site officielle de GNU Radio www.gnuradio.org.

- [75] Baldini, Gianmarco, Raimondo Giuliani et Dimitrios Symeonidis. 2008. « Reconfigurable Radio Systems for Public Safety Based on Low-Cost Platforms ». In *Intelligence and Security Informatics*, sous la dir. de Ortiz-Arroyo, Daniel, Henrik Legind Larsen, Daniel Dajun Zeng, David Hicks et Gerhard Wagner. Vol. 5376, p. 237-247. Coll. « Lecture Notes in Computer Science »: Springer Berlin Heidelberg. < http://dx.doi.org/10.1007/978-3-540-89900-6_24 >.
- [76] Feng, Ge, C. J. Chiang, Y. M. Gottlieb et R. Chadha. 2011. « GNU Radio-Based Digital Communications: Computational Analysis of a GMSK Transceiver ». In *Global Telecommunications Conference (GLOBECOM 2011), 2011 IEEE*. (5-9 Dec. 2011), p. 1-6.
- [77] Fiche technique de l'antenne : www.l-com.com/wireless-antenna-24-49-58-ghz-3-dbi-tri-band-rubber-duck-antenna-sma-male.
- [78] IEEE 802.15.4 Implementation on an Embedded Device April-2012- Master's Thesis - Thandee, Rithirong - Master of Science.
- [79] K. Dabcevic, "Evaluation of Software Defined Radio platform with respect to implementation of 802.15.4 Zigbee," ed, 2011, p. 83.
- [80] Mailing list de GNU Radio, lists.gnu.org/archive/html/discussgnuradio.
- [81] Hamed Firooz, Implementation of Full-Bandwidth 802.11b Receiver, span.ece.utah.edu/pmwiki/pmwiki.php?n=Main.80211bReceiver.
- [82] IEEE 802.15.4 Implementation on an Embedded Device Rithirong Thandee, Thesis - Virginia Polytechnic Institute and State University, Master of Science in Electrical and Computer Engineering, Carl B. Dietrich, Chair, Jeffrey H. Reed, Co-Chair, S.M. Shajedul Hasan - April 10th, 2012, Blacksburg, Virginia.
- [84] G. D. Troxel, E. Blossom, S. Boswell, A. Caro, I. Castineyra, A. Colvin, *et al.*, "Adaptive dynamic radio open-source intelligent team (ADROIT): Cognitively-controlled collaboration among SDR nodes," in *Networking Technologies for Software Defined Radio Networks, 2006. SDR'06.1st IEEE Workshop on*, 2006, pp. 8-17.
- [85] *Comprehensive GNU Radio Archive Network* CGRAN site web.
- [86] Standard IEEE 802.11b.
- [87] A. Luo and L. Ge, "Indoor Location Detection using WLAN," KTH, 2010.