

## TABLE DES MATIERES

REMERCIEMENTS .....	i
TABLE DES MATIERES.....	ii
NOTATIONS.....	viii
1. Minuscules latines .....	viii
2. Majuscules latines .....	viii
3. Minuscules grecques .....	ix
LISTE DES ABREVIATIONS.....	x
INTRODUCTION GENERALE .....	1
CHAPITRE 1 ENVIRONNEMENT DE PROPAGATION ET GENERALITE SUR LES RESEAUX DE CAPTEURS SANS FILS .....	3
1.1 Environnement des systèmes de radiocommunication .....	3
1.1.1 La propagation en espace libre .....	3
1.1.2 La propagation hors espace libre .....	4
1.1.2.1 Les phénomènes de base en propagation.....	5
1.1.2.2 La propagation par trajets multiples .....	7
1.1.2.3 Les pertes moyennes hors espace libre.....	7
1.1.3 Le bruit radioélectrique .....	8
1.1.4 Canal de propagation.....	8
1.1.4.1 Modélisation du canal de propagation.....	8
1.1.4.2 Dispersion dans les canaux .....	10
1.1.4.3 Les dispersions temporelles.....	10
1.1.4.4 Les dispersions fréquentielles .....	11
1.1.4.5 La sélectivité des canaux .....	12
1.1.4.6 La diversité dans le canal .....	13
1.2 Réseaux de capteurs sans fil.....	14
1.2.1 Description d'un Capteur.....	14
1.2.1.1 Définition .....	14
1.2.1.2 Propriétés.....	15
1.2.1.3 Caractéristiques .....	15

1.2.1.4 Types de batterie .....	17
<b>1.2.2 Technologie de collecte d'énergie : .....</b>	<b>18</b>
<b>1.2.3 Méthodes de collecte d'énergie : .....</b>	<b>18</b>
<b>1.2.4 Architectures de récupération d'énergie : .....</b>	<b>19</b>
<b>1.2.5 Description d'un réseau de capteurs sans fil.....</b>	<b>20</b>
1.2.5.1 Définition .....	20
1.2.5.2 Domaine d'application .....	21
<b>1.2.6 Communication coopérative.....</b>	<b>24</b>
1.2.6.2 Coopération via un simple relais .....	26
1.2.6.3 Coopération via une communication directe .....	26
1.2.6.4 Coopération via des multi-Branches .....	27
1.2.6.5 Coopération via clustering .....	27
<b>1.2.7 Différents niveaux de coopération .....</b>	<b>27</b>
1.2.7.1 Couche physique .....	28
1.2.7.2 Couche MAC.....	28
1.2.7.3 Couche réseau .....	29
1.2.7.4 Etablir une route stable.....	29
<b>1.2.8 Contrôler la topologie du réseau .....</b>	<b>29</b>
<b>1.3 Conclusion .....</b>	<b>30</b>
<b>CHAPITRE 2 EVALUATION DES SYSTEMES MIMO.....</b>	<b>31</b>
<b>2.1 Introduction.....</b>	<b>31</b>
<b>2.2 Modèle théorique d'un système MIMO .....</b>	<b>31</b>
<b>2.3 Les techniques MIMO .....</b>	<b>32</b>
<b>2.3.1 Systèmes multi-antennes avec CSI à la réception .....</b>	<b>33</b>
2.3.1.1 Systèmes MIMO à multiplexage spatial .....	33
2.3.1.2 Systèmes MIMO à codage espace-temps .....	35
<b>2.3.2 Systèmes multi-antennes avec CSI à l'émission et à la réception.....</b>	<b>37</b>
2.3.2.1 Systèmes MIMO utilisant des techniques de water-filling.....	37

2.3.3	<i>Systèmes multi-antennes sans CSI ni à l'émission ni à la réception</i>	39
2.4	<b>Critères d'analyses de performance des systèmes</b>	40
2.4.1	<i>SNR moyen dans un canal à évanouissements rapides</i>	40
2.4.2	<i>Probabilité d'erreur moyenne</i>	40
2.4.2.1	Evanouissements rapides	40
2.4.2.2	Evanouissements lents	41
2.5	<b>Evaluation de système MIMO-coop</b>	41
2.5.1	<i>Architecture</i>	41
2.5.2	<i>Description du modèle</i>	43
2.5.2.1	Modèle	43
2.5.2.2	Sélection des relais coopéreront	43
2.5.3	<i>Analyse de la consommation énergétique pour les WSNs</i>	45
2.5.3.1	Modèle énergétique	46
2.5.3.2	Consommation énergétique (SISO et MIMO)	47
2.6	<b>Conclusion</b>	51
<b>CHAPITRE 3 LE CODAGE SPATIO-TEMPOREL</b>		52
3.1	<b>Introduction</b>	52
3.2	<b>Construction des codes</b>	52
3.2.1	<i>L'efficacité spectrale</i>	52
3.2.2	<i>La robustesse</i>	53
3.2.3	<i>La complexité</i>	53
3.2.4	<i>Hypothèses</i>	53
3.2.5	<i>Critères de construction des codes espace-temps</i>	53
3.2.6	<i>Architecture générale d'un système de codage spatio-temporel</i>	55
3.2.7	<i>Différentes Classes des codes spatio-temporels</i>	56
3.2.7.1	Codes ST en Treillis	56
3.2.7.2	Codes ST en blocs	57
3.2.7.3	Codes ST orthogonaux	58

3.2.7.4 Code d'Alamouti .....	58
3.2.7.5 Codes Spatio Tempo Fréquentiels (STF) .....	60
<b>3.3 Algorithmes de décodage associés aux techniques MIMO .....</b>	<b>62</b>
3.3.1 <i>L'égaliseur de forçage à zéro (ZF)</i> .....	62
3.3.2 <i>L'égaliseur minimisant l'erreur quadratique moyenne (MMSE)</i> .....	62
3.3.3 <i>L'égaliseur à annulations successives d'interférences ordonnées</i> .....	63
3.3.4 <i>L'égaliseur à Maximum de Vraisemblance</i> .....	64
3.3.5 <i>Le décodage par sphère</i> .....	64
<b>3.4 La performance des codages .....</b>	<b>64</b>
3.4.1 <i>Performance du codage d'Alamouti</i> .....	64
3.4.2 <i>Performance du codage STTC (Space Time Treillis Coding)</i> .....	65
3.4.2.1 Améliorations apporté par le STTC .....	65
3.4.2.2 Performance des codages STTC.....	66
<b>3.5 Conclusion .....</b>	<b>69</b>
<b>CHAPITRE 4 LES METHODES DE LOCALISATION DANS UN RESEAU DE CAPTEURS SANS FILS ...</b>	<b>70</b>
<b>4.1 Introduction.....</b>	<b>70</b>
<b>4.2 Protocoles de collecte dans les méthodes <i>range-based</i>.....</b>	<b>70</b>
4.2.1 <i>Indicateur de puissance de signal reçu</i> .....	70
4.2.2 <i>Temps de vol du signal</i> .....	71
4.2.2.1 Le mono-signal.....	71
4.2.2.2 La combinaison de signaux .....	75
<b>4.3 Protocoles de collecte pour les méthodes <i>range-free</i>.....</b>	<b>75</b>
<b>4.4 Algorithmes de localisation .....</b>	<b>76</b>
4.4.1 <i>Algorithmes généraux</i> .....	76
4.4.1.1 Trilatération .....	76
4.4.1.2 Triangulation .....	77
4.4.1.3 Reconnaissance d'empreinte .....	78
4.4.1.4 Multilatération .....	78

<b>4.5 Adaptations récentes des algorithmes à la localisation par WSN.....</b>	<b>79</b>
<b>4.5.1 Méthodes range-based .....</b>	<b>79</b>
4.5.1.1 Temps de vol .....	79
4.5.1.2 RSSI et puissance reçue .....	80
4.5.1.3 TDOA et phase du signal .....	82
<b>4.5.2 Méthodes Range-Free.....</b>	<b>83</b>
4.5.2.1 Famille DV-Hop.....	83
4.5.2.2 APIT .....	86
4.5.2.3 Centroid.....	87
<b>4.6 Conclusion .....</b>	<b>88</b>
<b>CHAPITRE 5 REALISATION D'UN SIMULATEUR DE WSN .....</b>	<b>89</b>
<b>5.1 Introduction.....</b>	<b>89</b>
<b>5.2 Le langage de programmation C# .....</b>	<b>89</b>
<b>5.2.1 Caractéristiques .....</b>	<b>89</b>
<b>5.2.2 La bibliothèque de classe .....</b>	<b>92</b>
5.2.2.1 Réutilisabilité .....	92
5.2.2.2 Architecture :.....	93
<b>5.2.3 Comparaison entre C# et Java .....</b>	<b>93</b>
5.2.3.1 Développement multi langage.....	94
5.2.3.2 Les principaux IDE Java .....	94
5.2.3.1 Les principaux IDE C#.....	94
5.2.3.2 Interfaces graphiques.....	95
5.2.3.3 Les inconvénients de Microsoft .Net.....	96
<b>5.3 Cadre de travail.....</b>	<b>97</b>
<b>5.3.1 Environnement matériel.....</b>	<b>97</b>
<b>5.3.2 Environnement logiciel.....</b>	<b>97</b>
<b>5.4 Présentation de Visual Studio 2013 .....</b>	<b>98</b>
<b>5.4.1 Créer une application simple.....</b>	<b>98</b>

5.4.2 Pour créer le projet WPF.....	98
<b>5.5 Le simulateur.....</b>	<b>99</b>
5.5.1 Critères de conception .....	100
5.5.2 Caractéristiques .....	100
5.5.2.2 Les spécifications des algorithmes de localisation .....	101
5.5.2.3 La bibliothèque du réseau.....	102
5.5.3 L'interface graphique.....	104
5.5.3.2 Les paramètres.....	104
<b>5.6 Fonctionnement réel du programme.....</b>	<b>108</b>
5.6.1 Présentation de l'interface finale.....	108
5.6.2 Mode d'utilisation et illustration.....	111
<b>5.7 Comparaison du résultat de l'analyse des algorithmes de localisation .....</b>	<b>114</b>
<b>5.8 Conclusion .....</b>	<b>116</b>
<b>BIBLIOGRAPHIE .....</b>	<b>124</b>
<b>FICHE DE RENSEIGNEMENT .....</b>	<b>126</b>
<b>RESUME .....</b>	<b>127</b>
<b>ABSTRACT .....</b>	<b>127</b>

## NOTATIONS

### 1. Minuscules latines

$b$	Bruits
$c$	Célérité de la lumière
$d$	Distance
$d_c$	Portée de communication
$d_{er}$	Distance entre émetteur et récepteur
$d_{Loc}$	Distance des relais les plus proches
$d_s$	Portée de capture
$f$	Fréquence d'onde
$f_c$	Fréquence du signal émis
$f_d$	Décalage Doppler
$f_{d_{max}}$	Fréquence Doppler maximale
$f_r$	Fréquence du signal reçu
$h(\tau, t)$	Réponse impulsionnelle du canal
$h_{mn}$	Canal entre la $n^{ième}$ antenne d'émission et la $m^{ième}$ antenne de réception
hopcount	Nombre de saut
hopsize	Distance moyenne par saut
$k$	Facteur d'atténuation
$r(t)$	Signaux reçu
$s(t)$	Signaux émis
$t_c$	Temps de cohérences
$\hat{t}_{vol}$	Estimation du temps de vol
$v$	Vitesse de déplacement
$v_p$	Fréquence de décalage Doppler
$x$	Vecteur de symbole
$y$	Vecteur reçu

### 2. Majuscules latines

$B_s$	Bande symbole
$C$	Capacité
$CM_i$	Métrieque de connectivité
$CSI_{GMAXS_{r-d}}$	CSI global maximum
$CSI_{s-r}$	Qualité des liens entre source et relais
$CSI_{r-d}$	Qualité des liens entre relais et destination
$D_t$	Diversité temporelle
$D_f$	Diversité fréquentielle
$E_{bt}$	Consommation totale d'énergie par bit

$E_{totale}$	Energie totale
$E_{Totale}^{MIMO-Coop}$	Energie globale du système MIMO-Coopératif
$E_{Rx}^{Coop}$	Energie au niveau des récepteurs coopératifs
$E_{Tx}^{Coop}$	Energie au niveau des émetteurs coopératifs
$G_c$	Gain de codage
$G_d$	Gain de diversité
$H$	Matrice du canal
$N_{recv}(i,t)$	Nombre de beacons émis par la balise i et reçus durant le temps t
$N_s$	Nombre de symbole
$N_{sb}$	Nombre de Bit/Symbole-STBC
$N_{sent}(i,t)$	Nombre de beacons émis par la balise i durant le temps t
$P_c$	Puissance dissipée dans le circuit
$P_e$	Puissance émise
$P_{Pa}$	Puissance de transmission
$P_r$	Puissance reçu
$P_s(E)$	SEP moyen
$P_t$	Puissance totale
$Q$	Processus de quantification
$R$	Rendement du code
$R_b$	Débit binaire réel
$T$	Durée
$T_s$	Durée symbole
$T_{trame}$	Durée d'une trame
$T_{vol}$	Temps de vol
$W$	Vecteur d'égalisation
$\hat{W}$	Egaliseur MMSE

### 3. Minuscules grecques

$\lambda$	Longueur d'onde
$\beta_p$	Amplitude de trajet
$\gamma$	SNR
$\theta$	Angle d'incidence
$\theta_p$	Phase du trajet
$\rho_f$	Fréquence de corrélation
$\sigma_\tau$	Dispersion moyenne des retards
$\tau_p$	Retard du trajet
$\tau_{max}$	Retard maximale
$\gamma_s P_{\gamma_s}(\gamma_s)$	Densité de probabilité du SNR instantanée



## LISTE DES ABREVIATIONS

<b>FCL</b>	Framework Class Library
<b>FDD</b>	Frequency Division Duplex
<b>FER</b>	Frame Error Rate
<b>GDI</b>	Graphical Device Interface
<b>GPS</b>	Global Positioning System
<b>Hirloc</b>	High Resolution Robust Localization Algorithm
<b>IDE</b>	Integrated Development Environment
<b>IES</b>	Interférence Entre Symbole
<b>IID</b>	Indépendant Identique Distribution
<b>IL</b>	Intermediate Language
<b>ITGPSLS</b>	Independent Time GPS Least Squares
<b>J2EE</b>	Java 2 Enterprise Edition
<b>JCP</b>	Java for Process Control
<b>JVM</b>	Java Virtual Machine
<b>LD</b>	Linear Dispersion
<b>LINQ</b>	Language Integrated Query
<b>LLS</b>	Linear Least Squares
<b>LOS</b>	Line Of Sight
<b>LST</b>	Layered Space-Time
<b>MAC</b>	Medium Access Control
<b>MIMO</b>	Multiple Input Multiple Output
<b>MISO</b>	Multiple Input Single Output
<b>ML</b>	Maximum Likelihood
<b>MMSE</b>	Minimum Mean Square Error
<b>MRC</b>	Maximum Ratio Combiner
<b>MVC</b>	Model - View - Controller
<b>NLLS</b>	Non Linear Least Squares
<b>NLOS</b>	Non Line Of Sight
<b>NN</b>	Nearest Neighbor
<b>NNSS</b>	Nearest Neighbor in Signal Space
<b>OFDM</b>	Orthogonal Frequency Division Multiplexing
<b>OS</b>	Operating System
<b>OSIC</b>	Ordered Successive Interference Cancellations
<b>PSK</b>	Phase Shift Keying
<b>QAM</b>	Quadrature Amplitude Modulation
<b>QoS</b>	Qualité de Service
<b>QPSK</b>	Quadrature Phase Shift Keying
<b>RAD</b>	Rapid Application Development
<b>RAL</b>	Restricted Area Based Localization Algorithm
<b>RAM</b>	Random Access Memory

<b>RF</b>	Radio Fréquence
<b>RSSI</b>	Received Signal Strength Indicator
<b>SD</b>	Sphere Decoder
<b>SDS</b>	Symmetric Double-Sided
<b>SDS/TWR</b>	Symmetric Double-Sided/Two-Way Ranging
<b>SEP</b>	Symbol Error Probability
<b>SeRLoc</b>	Secure Range Independent Localization Algorithm
<b>SIMO</b>	Single Input Multiple Output
<b>SISO</b>	Single Input Single Output
<b>SNR</b>	Signal Noise Ratio
<b>SS</b>	Signal Strength
<b>STB</b>	Spatio-Temporels en Bloc
<b>STBC</b>	Space Time Block Code
<b>STC</b>	Space Time Coding
<b>STF</b>	Spatio Tempo Fréquentiels
<b>STT</b>	Spatio-Temporels en Treillis
<b>STTC</b>	Space Time Treillis Coding
<b>SWT</b>	Standard Widget Toolkit
<b>TDD</b>	Time Division Duplex
<b>TDOA</b>	Time Difference Of Arrival
<b>TEB</b>	Taux d'Erreur Binaire
<b>TEB</b>	Taux d'Erreur Binaire
<b>TOA</b>	Time Of Arrival
<b>TWR</b>	Two-Way Ranging
<b>V-BLAST</b>	Vertical - Bell Labs Advanced Space Time
<b>WF</b>	Water-Filling
<b>WPF</b>	Windows Presentation Foundation
<b>WSN</b>	Wireless Sensor Network
<b>WSSUS</b>	Wide Sense Stationnary Uncorrelated Scattering
<b>XML</b>	eXtended Markup Language
<b>ZF</b>	Zero Forcing

## INTRODUCTION GENERALE

Aujourd'hui, les progrès réalisés dans les domaines de la micro-électronique, et des technologies de communications sans fil permettent de créer de petits systèmes communicants équipés de capteurs à un coût raisonnable. Ces microcomposants communicants appelés capteurs sont équipés d'une unité de mesure, d'une unité de calcul, de mémoires et d'une radio pour communiquer. Les micro-capteurs sont donc de véritables systèmes embarqués. Ces avancées technologiques rendent possible le déploiement de réseaux de capteurs sans fil. En effet, les domaines d'application des réseaux de capteurs sont très variés : applications militaires, domotique, surveillance industrielle ou de phénomènes naturels.

Un réseau de capteurs sans fil est un réseau ad hoc avec un grand nombre de nœuds qui sont des micro-capteurs. La position de ces nœuds n'est pas obligatoirement prédéterminée. Ils peuvent être aléatoirement dispersés dans une zone géographique, appelée « champ de captage » correspondant au terrain d'intérêt pour le phénomène capté. Ce champ de captage est souvent un environnement hostile auquel l'homme n'a pas toujours accès. C'est pourquoi on considère qu'une fois qu'ils sont déployés, les capteurs sont autonomes.

Cette autonomie dépend en grande partie de la durée de vie des capteurs qui est la durée de vie de leur batterie. Le facteur énergie est donc un des piliers fondamentaux pour les capteurs. De ce fait on cherche des protocoles de routage "économiques", une technologie sans fil adaptée, redondance des équipements. Il faut minimiser les dépenses énergétiques, car l'énergie est une contrainte clé dans les réseaux de capteurs.

Par ailleurs, un système MIMO (Multiple Input Multiple Output) est constitué de plusieurs antennes, ainsi que des traitements adaptatifs de signal, à la fois à l'émetteur et au récepteur. Ce système exploite les effets des trajets multiples au lieu de les atténuer. Ce sont des systèmes très économes en matière d'énergie surtout pour les transmissions longues distances.

Le CO-MIMO (Cooperative MIMO), réseau MIMO (Net-MIMO) ou MIMO ad hoc, utilise des antennes qui appartiennent à d'autres utilisateurs, tandis que le MIMO classique (mono-utilisateur MIMO) n'emploie que des antennes appartenant au terminal local distribué. CO-MIMO améliore les performances d'un réseau sans fil en introduisant de multiples avantages d'antenne, comme la diversité, le multiplexage et la formation de faisceau. Si l'intérêt principal repose sur le gain de la

diversité, il est connu que la diversité coopérative, il peut être décrit comme une forme de macro-diversité, utilisés par exemple dans le soft-handover.

Un noyau unique de haute performance est la première génération de l'évolution du CPU (Central Process Unit), progressant à quelques noyaux, puis de nombreux noyaux d'une manière centralisée. Il est prévu qu'il sera commun pour le travail coopératif de procéder à partir de plusieurs noyaux appartenant à différents utilisateurs, mis à l'utilisateur individuel en contrepartie de l'aide pour le traitement de l'information des autres. Ces slogans comprennent l'intelligence ambiante, l'informatique ubiquitaire sans fil, et le web sémantique.

### Problématique

La rentabilité des capteurs porte sur sa durée de vie qui est fortement diminuée par la quantité d'information qu'il doit transmettre. Puisque ces informations sont, dans certains cas, extrêmement sensibles ; les capteurs doivent s'assurer de transmettre ces données sans altération (si possible). Il faut aussi tenir à l'esprit que la position géographique du capteur est aussi une information capitale. Dans la mesure du possible il serait donc apprécié d'affiner des techniques qui ne comportent pas beaucoup de traitement pour réduire la consommation d'énergie et dans cette même optique d'assurer que la transmission soit sans erreur ou que les erreurs soient corrigibles au niveau des récepteurs finaux.

### Objectif générale

Pour permettre une amélioration de la durée de vie des capteurs ainsi que d'améliorer les communications utilisant des capteurs mobiles nous nous focaliserons sur la partie de la transmission.

Etant donné l'importance des techniques de traitement d'informations, nous analyserons donc tous les algorithmes de codage des informations ainsi que les algorithmes de localisations généralement utilisés dans les réseaux de CO-MIMO.

# CHAPITRE 1

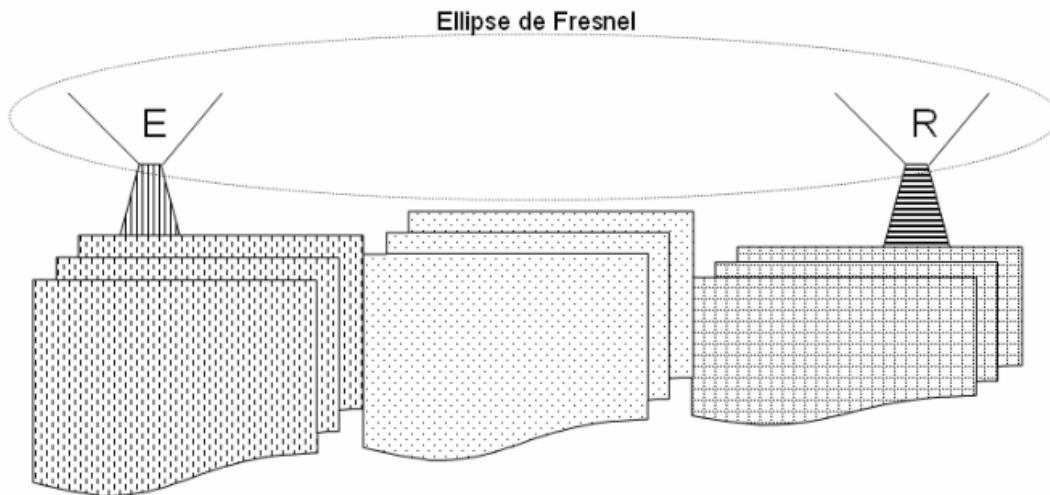
## ENVIRONNEMENT DE PROPAGATION ET GENERALITE SUR LES RESEAUX DE CAPTEURS SANS FILS

### 1.1 Environnement des systèmes de radiocommunication

Transmettre de l'information d'un point à un autre sans-fil nécessite un système de radiocommunication. Ce système utilise le canal hertzien comme support physique. Au cours de sa propagation, l'onde transmise traverse différents types de canaux jusqu'à atteindre sa cible. Ces canaux affectent l'onde émise. Cela rend nécessaire l'étude des canaux de propagation. Les performances de transmission dépendent considérablement des caractéristiques du canal. Selon les conditions d'observation, ces canaux peuvent être considérés comme statiques ou dynamiques.

#### 1.1.1 La propagation en espace libre

Selon l'environnement de propagation, les influences du canal diffèrent. Le modèle d'espace libre permet d'avoir une première approche. L'espace libre peut être modélisé comme l'illustre la Figure 1.01. Dans le cas d'une liaison en vue directe, connue aussi sous le terme LOS (Line Of Sight) entre l'émetteur et le récepteur, une ellipse appelée zone de Fresnel permet de cerner la validité du modèle espace libre dans un canal réel. Effectivement, si cette zone n'est pas dégagée, le modèle n'est plus valable.



**Figure 1.01 : Ellipse de Fresnel**

En espace libre, l'onde se propage depuis l'émetteur jusqu'à ce qu'une partie de la puissance transmise  $P_e$  excite l'antenne réceptrice. Cette puissance reçue  $P_r$  s'exprime en fonction du gain des deux antennes  $G_e$  et  $G_r$ , de la distance de propagation  $d$  et de la longueur d'onde  $\lambda$  comme suit:

$$P_r = P_e G_e G_r \left( \frac{\lambda}{4\pi d} \right)^2 \quad (1.01)$$

C'est une puissance de réception maximale. Cette valeur ne prend pas en compte la dépolarisation, la désadaptation et les variations spatiales du gain des antennes.

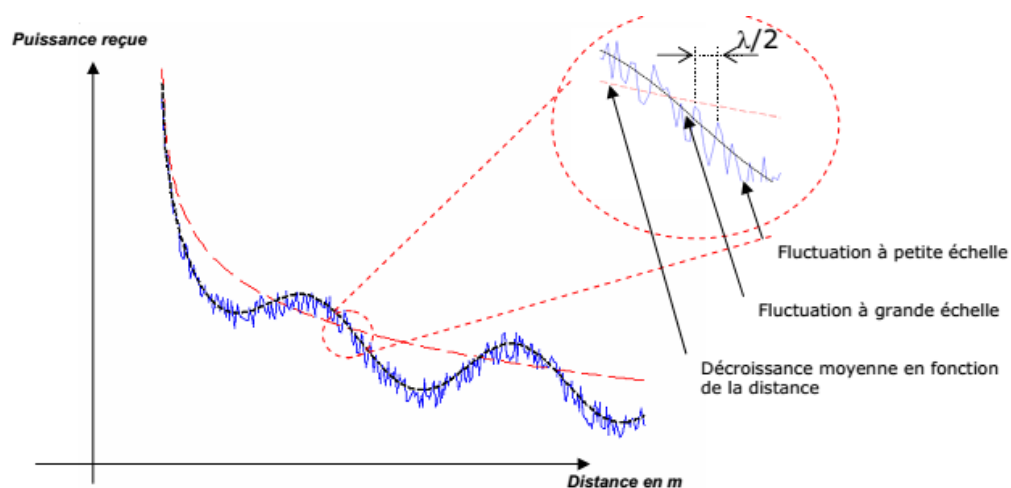
On en déduit l'affaiblissement de la liaison :

$$L = \frac{P_r}{P_e} = G_e G_r \left( \frac{\lambda}{4\pi d} \right)^2 \quad (1.02)$$

En considérant des antennes omnidirectionnelles ( $G_e = G_r = 1$ ), on en déduit de l'équation précédente que les pertes de propagation, à fréquence constante, sont inversement proportionnelles à  $d^2$ .

### 1.1.2 La propagation hors espace libre

Les situations les plus courantes de propagation des systèmes de radiocommunication sont loin d'être en espace libre. Les obstacles de différentes formes géométriques et de diverses caractéristiques physiques perturbent la propagation. Ces perturbations se traduisent par des fluctuations de la puissance du signal reçu en fonction de la distance comme l'illustre la figure 1.02. Les pertes en fonction de la distance et les effets de masquages [1] sont les phénomènes à l'origine de ces variations. La décroissance en  $1/d$  est la principale perte de puissance. La densité de puissance se réduit au fur à mesure que l'onde s'éloigne de sa source jusqu'à atteindre l'antenne réceptrice. L'influence des obstacles rencontrés par l'onde varie selon leurs configurations.



**Figure 1.02 : Puissance reçue en fonction de la distance**

En plus de la décroissance de la densité de puissance en fonction de la distance, entre l'antenne d'émission et l'antenne de réception, le signal subit deux types de pertes : pertes à petite et grande échelle. Comme le décrit la figure 1.02, les pertes à grande échelle définissent les fluctuations de la puissance moyenne mesurées sur un déplacement de plusieurs dizaines de longueurs d'onde. Les fluctuations à petite échelle sont observées sur un déplacement suffisamment petit (quelques longueurs d'onde). La présence d'obstacles dans l'environnement de propagation causant l'apparition de différentes répliques du signal émis au niveau du récepteur est la cause principale de ces variations. Selon leurs phases d'arrivée au niveau de l'antenne, la somme de ces répliques peut être constructive ou destructive. C'est la source du phénomène d'évanouissement. Ces obstacles naturels (sol, arbres, bâtiments, etc.) qui se trouvent sur le trajet de l'onde. Il en résulte une multitude de trajets et donc, une multitude d'ondes retardées, atténuées et déphasées au niveau du récepteur.

#### 1.1.2.1 Les phénomènes de base en propagation

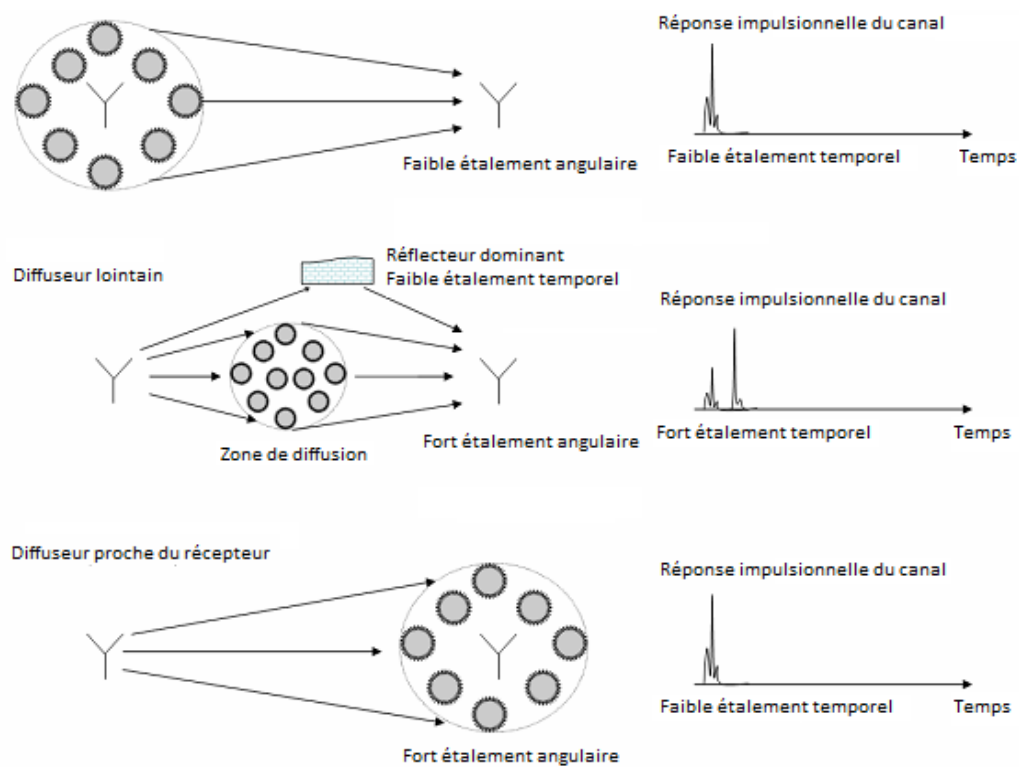
Les obstacles rencontrés par le signal lors de son trajet de l'antenne d'émission à l'antenne de réception agissent différemment sur le signal. En effet, différents phénomènes apparaissent selon la taille vis à vis de la longueur d'onde  $\lambda$ , de la nature et de la forme de ces obstacles [2].

La réflexion, la diffraction et la diffusion sont les trois principaux mécanismes qui perturbent le signal [3,4]. Leur description est la suivante :

- **Réflexion/Réfraction:** Les phénomènes de réflexion et de réfraction apparaissent lorsque l'obstacle rencontré par l'onde a une taille très supérieure et de très petites irrégularités devant la longueur d'onde du signal.
- **Diffraction:** Des phénomènes de diffraction apparaissent lorsque le chemin de propagation est obstrué par un obstacle imperméable aux ondes électromagnétiques. Les dimensions de cet obstacle doivent être faibles devant la longueur d'onde du signal ou l'obstacle possède des arêtes vives. L'énergie transmise par ces sources permet au signal de se propager dans les zones d'ombre, ce qui explique l'arrivée d'ondes radio au niveau du récepteur en l'absence de visibilité directe et d'interventions des autres types d'interactions. C'est une grande source de multitrajets.
- **Diffusion:** La diffusion apparaît s'il existe sur le trajet de l'onde une zone très dense d'objets de dimensions du même ordre de grandeur ou inférieures à la longueur d'onde. Le même phénomène est observé avec une surface rugueuse présentant des irrégularités suffisamment

petites. L'influence de la position spatiale de l'obstacle est illustrée sur la figure 1.03. On distingue deux principaux types de réflecteurs :

- **Diffuseurs locaux:** Les diffuseurs locaux sont les obstacles proches de l'émetteur ou du récepteur. Au niveau du récepteur, les diffuseurs placés dans son voisinage occasionnent un grand étalement angulaire des échos et un étalement temporel faible. Les diffuseurs proches de l'émetteur introduisent de faibles étalements, temporel et angulaire.
- **Diffuseurs lointains:** Les diffuseurs lointains désignent les obstacles éloignés simultanément de l'émetteur et du récepteur. Ils donnent lieu à des trajets multiples généralement caractérisés par un fort étalement temporel.



**Figure 1.03 :** *Influence de la localisation spatiale des diffuseurs sur la réponse du canal de propagation*

Selon l'application envisagée, ces obstacles peuvent être considérés comme un avantage ou un inconvénient. Par exemple, lorsque l'émetteur et le récepteur sont en vue directe (LOS), la réflexion perturbe la liaison. Cependant, dans le cas du canal NLOS (Non Line Of Sight), la diffraction et la diffusion assurent la continuité de la liaison.



### 1.1.2.2 La propagation par trajets multiples

Les phénomènes précédemment présentés modifient le signal transmis. De nombreuses répliques de ce signal sont ainsi créées. Celles-ci sont plus ou moins retardées selon les longueurs des trajets effectués. Dès lors, elles sont plus ou moins atténuées selon la distance parcourue et selon les phénomènes de base rencontrés. A la réception, ces répliques se combinent de façon constructive ou destructive donnant naissance à des évanouissements, représentés sur la figure 1.02. Dans le cas du canal radio mobile, en supposant que les diffuseurs sont uniformément répartis sur  $[0 ; 2\pi [$ , ces évanouissements apparaissent statistiquement en moyenne tous les  $\lambda/2$ .

Les systèmes de radiocommunications mobiles, que ce soit à l'intérieur ou l'extérieur des bâtiments, sont ainsi soumis aux distorsions induites par les trajets multiples. Cependant dans le cas NLOS, les trajets multiples évitent l'interruption de la liaison entre l'émetteur et le récepteur.

### 1.1.2.3 Les pertes moyennes hors espace libre

Déduite de l'équation des télécommunications, l'équation 1.02 permet d'estimer les pertes dans le canal. Sauf que cette relation n'est valable que dans le cas d'espace libre. Dans un cas réel, c'est-à-dire hors espace libre, cette relation n'est plus exploitable. Ceci est principalement dû aux phénomènes qui apparaissent en présence d'obstacle. Le dimensionnement théorique du canal est très complexe. Cependant, des campagnes de sondage du canal ont permis d'élaborer des relations empiriques permettant d'estimer les pertes moyennes de puissance dans des canaux réalistes [1]. L'équation 1.03 ci-dessous est une relation qui permet d'estimer les pertes dues au canal de propagation. Elle s'exprime en fonction de la distance entre l'émetteur et le récepteur, le gain des antennes, la fréquence porteuse de l'onde et le type de canal de propagation.

$$L' = 32.44 + 20\log_{10}(f) - 10\log_{10}(G_e G_r) + 10\alpha\log_{10}(d) \quad (1.03)$$

où  $f$  est la fréquence de l'onde émise en MHz et  $d$  la distance entre les deux antennes exprimée en km.

Comme on peut le déduire de cette relation, les pertes moyennes de puissance ne sont plus proportionnelles à  $d^2$  comme c'était le cas dans un espace libre, mais à  $d^\alpha$  où  $\alpha$  est une variable dépendante de l'environnement. Les différentes valeurs de  $\alpha$  en fonction du type d'environnement sont présentées dans le tableau suivant :

Environnement	Valeur de $\alpha$
Espace Libre	2
Urbain	2.7 à 3.5
Urbain dense	3 à 5
Intérieur de bâtiment de type bureau avec trajet direct	1.6 à 1.8
Intérieur de bâtiment de type bureau sans trajet direct	4 à 6
Intérieur de bâtiment de type hall sans trajet direct	2 à 3

**Tableau 1.01:** Valeur de la variable  $\alpha$  en fonction du type d'environnement

### 1.1.3 Le bruit radioélectrique

Un signal radioélectrique est dit bruit s'il ne transporte pas l'information utile et perturbe la liaison. Il est considéré comme aléatoire. Le milieu de propagation ainsi que les dispositifs électroniques de l'émetteur et du récepteur sont respectivement ces origines externes et internes [2,3]. Les sources de bruits externes peuvent être de nature extra-terrestre ou terrestre. Elles regroupent les bruits et les parasites atmosphériques, les rayonnements divers captés par l'antenne, les interférences éventuelles entre les utilisateurs du milieu de transmission ou encore les bruits d'origine industrielle. Le bruit interne a pour origine le mouvement brownien des électrons présents dans les composants électroniques du récepteur. Ces électrons évoluent indépendamment les uns des autres, tout en suivant une même loi. Le bruit interne peut alors être modélisé, d'après le théorème de la limite centrale [4], par un processus gaussien.

Le bruit interne est considéré comme blanc, ce qui veut dire que sa densité de puissance est la même quel que soit la fréquence.

### 1.1.4 Canal de propagation

#### 1.1.4.1 Modélisation du canal de propagation

La caractérisation d'un canal peut se faire en étudiant sa réponse impulsionnelle. En considérant l'émetteur et/ou le récepteur en mouvement, l'expression de la réponse impulsionnelle  $h$  s'exprime comme :

$$h(\tau, t) = \sum_{p=1}^{P-1} \beta_p(t) e^{i(2\pi\nu_p t + \theta_p(t))} \delta(\tau - \tau_k) \quad (1.04)$$

Dans cette expression,  $P$  correspond au nombre de trajets discernables par le récepteur. Le nombre de trajets discernables n'est pas obligatoirement égal au nombre de trajets réellement existant dans le canal. En effet, chacun des  $P$  signaux retardés résulte de la recombinaison de plusieurs trajets.

Chaque trajet discernable peut ainsi être modélisé par une amplitude  $\beta_p$  et un déphasage  $\theta_p$ , associés au retard  $\tau_p$ . Le terme  $v_p$ , désigne la fréquence de décalage Doppler, décalage provoqué par les déplacements relatifs de l'émetteur et/ou du récepteur et/ou des obstacles. Cette représentation de la réponse impulsionnelle du canal de propagation permet de lier les signaux reçus  $r$  et émis  $s$ , selon les relations 1.05 et 1.06, où  $n$  désigne le terme de bruit additif gaussien.

$$r(t) = (h * s)(t) + n(t) \quad (1.05)$$

$$= \int_{-\infty}^{+\infty} \sum_{p=1}^{P-1} \beta_p(t) e^{i(2\pi v_p t + \theta_p(t))} \delta(\tau - \tau_k) s(t - \tau) d\tau + n(t) \quad (1.06)$$

$$r(t) = \sum_{p=1}^{P-1} \beta_p(t) e^{i(2\pi v_p t + \theta_p(t))} s(t - \tau_k) + n(t) \quad (1.07)$$

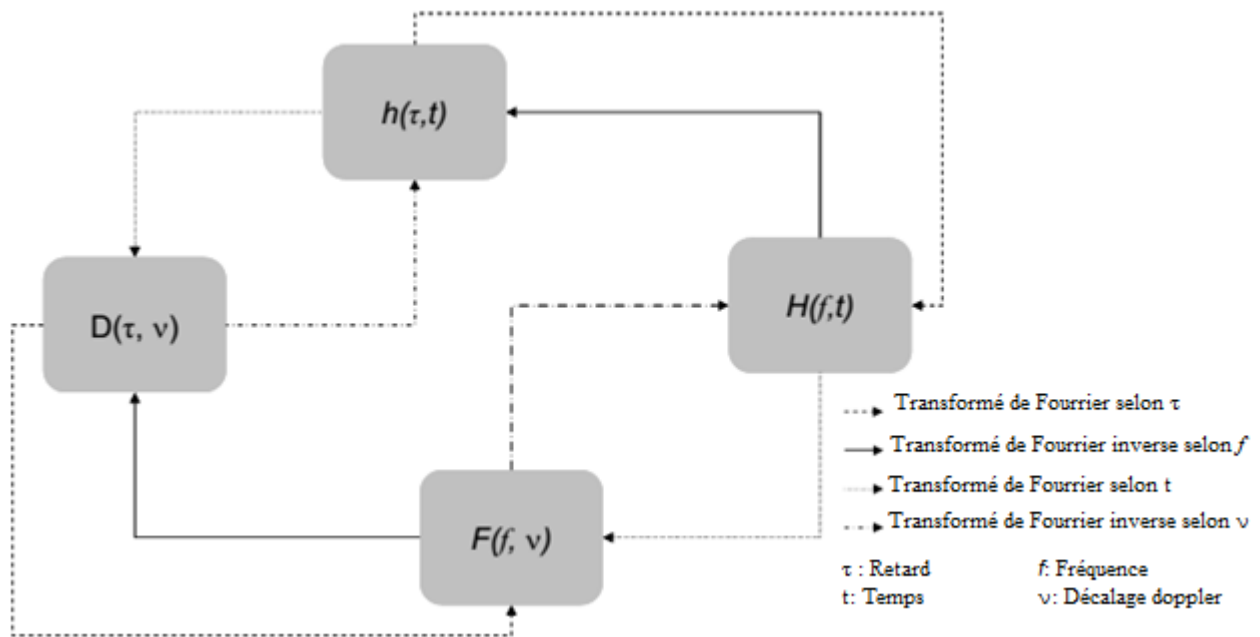
Il est ainsi possible de définir la réponse impulsionnelle  $h$  du canal dans l'espace-temps retard. Cette réponse désigne la fonction d'étalement des retards [7]. D'autres relations permettent la caractérisation du canal. Ces relations sont les espaces fréquence-temps, retard Doppler et fréquence-Doppler et sont définies dans les expressions suivantes :

$$H(f, t) = \int_{-\infty}^{+\infty} h(\tau, t) e^{-2i\pi f\tau} d\tau \quad \text{Fréquence-Temps} \quad (1.08)$$

$$D(\tau, v) = \int_{-\infty}^{+\infty} h(\tau, t) e^{-2i\pi v\tau} dt \quad \text{Retard-Doppler} \quad (1.09)$$

$$H(f, t) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} h(\tau, t) e^{-2i\pi(vt + f\tau)} d\tau dt \quad \text{Fréquence-Doppler} \quad (1.10)$$

Un diagramme défini dans [7] est appelé diagramme de Bello. Il permet d'établir une relation simple entre les fonctions introduites et la fonction de transfert, la fonction de diffusion ainsi que la réponse bi-fréquentielle du canal. Chaque relation est reliée à une autre par l'opération de transformée de Fourier directe ou inverse. La connaissance d'une de ces fonctions permet alors la connaissance totale du canal de propagation.



**Figure 1.04 : Diagramme de Bello**

Les paramètres de caractérisation du canal sont définis en étudiant les fonctions d'autocorrélation des équations 1.08, 1.09 et 1.10. En supposant que le canal est stationnaire en temps et fréquence et que les trajets multiples sont d'amplitude et de phase décorrélés, la fonction d'autocorrélation de la fonction de transfert du canal permet de définir les fonctions de corrélation temporelle et fréquentielle. Ces précédentes hypothèses connues sous le nom WSSUS (Wide Sense Stationary Uncorrelated Scattering) permettent de simplifier la caractérisation statistique du canal de propagation.

#### 1.1.4.2 Dispersions dans les canaux

La configuration spatiale du récepteur et de l'émetteur engendre des dispersions temporelles et fréquentielles. Ces dispersions sont déterminantes pour le canal de propagation. Elles définissent des paramètres d'exploitation du canal tel que la bande et le temps de cohérence. Ces paramètres sont essentiels au dimensionnement d'un système de radiocommunication.

#### 1.1.4.3 Les dispersions temporelles

La caractérisation impulsionnelle du canal consiste à observer en réception l'arrivée de l'impulsion émise. En présence de multi trajet, après l'arrivée de la réponse du trajet le plus court, arrivent les réponses des échos. Le temps écoulé entre l'impulsion reçue du premier trajet et celui du dernier

écho est l'étalement des retards maximal noté  $\tau_{\max}$ . Ainsi, l'étude statistique de la variable aléatoire  $\tau \in [0; \tau_{\max}]$  déterminera la dispersion moyenne des retards, notée  $\sigma_\tau$ . Cette dispersion temporelle des retards se caractérise dans le domaine fréquentiel par une corrélation sur une bande de fréquence. La bande de cohérence  $B_c$  permet de quantifier le degré de corrélation entre deux fréquences distinctes. Ainsi, si l'écart fréquentiel entre deux signaux émis est supérieur à la bande de cohérence, alors les signaux reçus seront considérés comme décorrélés.

La dispersion des retards et la bande de cohérence sont inversement proportionnelles. En effet, une dispersion élevée des retards provoque une bande de cohérence faible. De nombreuses relations empiriques [1], [5] dépendantes du type du canal, permettent alors d'exprimer  $B_c$  en fonction de  $\sigma_\tau$ . Voici quelques expressions empiriques [1] :

$$B_c \approx \frac{1}{5\sigma_\tau} \quad \text{pour } \rho_f = 0.5 \quad (1.11)$$

$$B_c \approx \frac{1}{50\sigma_\tau} \quad \text{pour } \rho_f = 0.9 \quad (1.12)$$

où  $\rho_f$  est le coefficient de corrélation.

#### 1.1.4.4 Les dispersions fréquentielles

Les dispersions fréquentielles sont liées au mouvement de l'émetteur et/ou du récepteur. Ces mouvements provoquent un décalage fréquentiel du spectre des signaux émis, appelé effet Doppler. Ce décalage dépend de la vitesse relative de déplacement. Ainsi pour un signal émis à la fréquence  $f_c$  le récepteur aura une fréquence de réception  $f_r$  qui sera la somme de la fréquence émise  $f_c$  et le décalage Doppler  $f_d$  :

$$f_r = f_c + f_d \quad (1.13)$$

$f_d$  est fonction de l'angle  $\theta$  d'incidence de l'onde reçue et de la vitesse  $v$  de déplacement relatif entre l'émetteur et le récepteur :

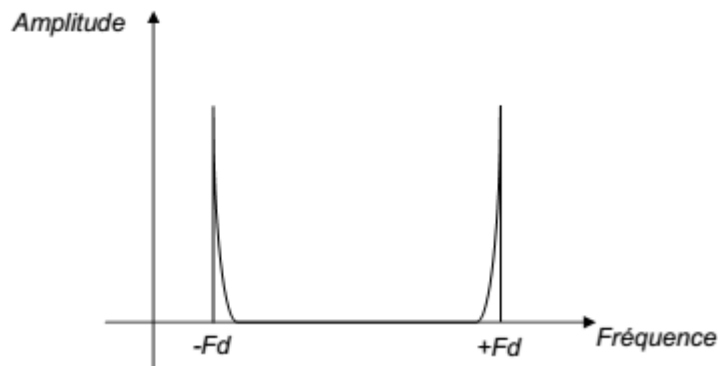
$$f_d = \frac{vf_c}{c} \cos\theta \quad (1.14)$$

où  $c$  désigne la célérité de la lumière. Selon cette expression, la bande fréquentielle sur laquelle évolue la fréquence de décalage Doppler appelée étalement Doppler ou bande Doppler est égale à :

$$B_d = 2f_{d_{\max}} \quad \text{avec } f_{d_{\max}} = \frac{vf_c}{c} \quad (1.15)$$

où  $f_{d_{\max}}$  désigne la fréquence Doppler maximale.

Le spectre Doppler représente la répartition de la puissance autour de la fréquence porteuse. Lorsque les signaux sont reçus uniformément sur toutes les directions et que l'antenne de réception est omnidirectionnelle, le spectre Doppler obtenu a une forme de « U » dite de Jakes.



**Figure 1.05 :** *Spectre Doppler par modèle de Jakes*

Le temps de cohérence  $t_c$  du canal et la fréquence Doppler maximale  $f_{dmax}$  évoluent de manière inversement proportionnelle. Le temps de cohérence du canal représente l'évolution temporelle de celui-ci, il est lié aux dispersions fréquentielles. Il est alors possible d'exprimer  $t_c$  en fonction de  $f_d$  [1] :

$$t_c \approx \frac{9}{16\pi f_{dmax}} \quad (1.16)$$

Ces dernières notions de bande et de temps de cohérence permettent de dégager certaines contraintes de développement pour les systèmes de radiocommunications vis-à-vis d'un canal de propagation donné.

#### 1.1.4.5 La sélectivité des canaux

La reconstruction des signaux transmis nécessite quelques suppositions pour un traitement numérique adéquat en aval. Les valeurs de la bande de cohérence et la fréquence de cohérence définissent la sélectivité du canal. Les multi-trajets ainsi que les déplacements de l'émetteur et/ou du récepteur sont les sources d'une sélectivité fréquentielle et temporelle.

Un signal transmis peut être caractérisé par sa durée symbole  $T_s$  ainsi que sa bande  $B_s$ . La robustesse de ce signal dépendra des rapports  $T_s/t_c$  et  $B_s/B_c$ .

Ainsi, quatre cas de figures se présentent :

- $B_s \ll B_c, T_s \gg \sigma_\tau$  : si la bande occupée par le signal est inférieure à la bande de cohérence du canal, ou la durée du symbole émis est largement supérieure à la dispersion des retards, alors, les fréquences du spectre sont corrélées. Ainsi, les fréquences du spectre du signal

subissent les mêmes amplifications ou atténuations. Dès lors, le canal est considéré comme non sélectif en fréquence et à «évanouissements plats».

- $B_s > B_c$ ,  $T_s < \sigma_\tau$  : si la bande occupée par le signal est supérieure à la bande de cohérence du canal, ou la durée du symbole émis est inférieure à la dispersion des retards, alors, les fréquences du spectre sont décorréliées. Lors d'un évanouissement dans la bande de cohérence, une partie du signal seulement sera perturbée. Le canal est dit sélectif en fréquence. En outre, la durée du symbole étant inférieure à la dispersion des retards, des interférences entre symboles apparaissent.
- $T_s < t_c$ ,  $B_s > B_d$  : si la durée du symbole émis est inférieure au temps de cohérence du signal, ou la bande occupée par le signal est supérieure à la bande Doppler, alors, le canal est dit à «évanouissements lents». Le canal est alors non-sélectif en temps. La réponse impulsionnelle du canal reste constante sur plusieurs symboles consécutifs.
- $T_s \gg t_c$ ,  $B_s \ll B_d$  : si la durée du symbole émis est largement supérieure au temps de cohérence du signal, ou la bande occupée par le signal est largement inférieure à la bande Doppler, alors, le canal est dit à «évanouissements plats». Dans ces conditions, la réponse impulsionnelle du canal varie de façon significative pendant la durée d'un symbole. Le canal est alors sélectif en temps.

#### 1.1.4.6 La diversité dans le canal

La diversité se définit par la présence en réception de plusieurs répliques indépendantes d'une même information. On distingue plusieurs formes de diversité, par exemple, les diversités spatiales, temporelles, fréquentielles, etc ... La notion de diversité est utile afin d'évaluer la robustesse d'un système de radiocommunications vis à vis d'un canal donné. Les grandeurs  $D_t$  et  $D_f$  représentent respectivement l'ordre de diversité temporelle et fréquentielle. Ces grandeurs correspondent à l'ordre de diversité utilisé lors de la transmission d'un signal d'une trame de durée  $T_{trame}$  de signaux de durée  $T_s$ , leur expression est la suivante:

$$D_t = \frac{T_{trame}}{t_c} \quad (1.17)$$

$$D_f = \frac{B_s}{B_c} \quad (1.18)$$

Recevoir plusieurs répliques décorréliées du même signal apporte de la diversité. Cette diversité permet non seulement de contourner les obstacles physiques du canal (évanouissement) mais aussi de garantir une bonne transmission de l'information.

La dimension spatiale permet aussi d'obtenir une forme de diversité. La diversité spatiale consiste à transmettre simultanément ou non, des informations sur plusieurs antennes. Ces dernières doivent être espacées d'au moins la distance minimale assurant l'indépendance des évanouissements qui est dite « distance de cohérence ». Alors que la cohérence fréquentielle est engendrée par les multi-

trajets arrivant à des instants différents, la cohérence spatiale à l'émission/réception est engendrée par les multi-trajets partant/arrivant de différentes directions spatiales.

L'expression de la diversité spatiale dépend du schéma de modulation (SIMO : Single Input Multiple Output, MISO : Multiple Input Single Output ou MIMO). Les travaux présentés dans [6] énumèrent les différentes expressions possibles de la diversité spatiale. Dans le cas d'un système MIMO comprenant  $N$  antennes à l'émission et  $M$  antennes à la réception, la diversité spatiale maximale sera égale à  $N \times M$ . Profiter des diversités temporelles, fréquentielles et spatiales du canal revient à maximiser ces valeurs tout en garantissant une bonne efficacité spectrale. L'efficacité spectrale est une autre forme de caractérisation numérique du canal. Elle est estimée en déterminant la capacité de ce dernier.

## **1.2 Réseaux de capteurs sans fil**

Durant ces dernières décennies, les réseaux de capteurs sans fil ont suscité un intérêt majeur. Ils se présentent comme un nouveau domaine en plein émergence au niveau des technologies de communication. Les WSNs (Wireless Sensor Network) connaissent aujourd'hui un nouvel essor dû au besoin d'observer et contrôler des phénomènes physiques tels que la température, la pression, ou la luminosité. Il faut noter qu'un réseau de capteurs est constitué d'équipements de tailles réduites appelés nœuds capteurs dont le nombre peut atteindre des centaines de milliers. Ces nœuds communiquent entre eux selon une topologie qui peut être fixe ou mobile, leur tâche est d'acheminer les informations vers une autre unité de commande qui est en dehors de la zone de mesure [7]. L'objectif principal de ce chapitre est de donner une vue générale sur les réseaux de capteurs sans fil, en montrant les différentes contraintes et les outils existants pour leurs conceptions afin de permettre une bonne simulation de ces réseaux.

### **1.2.1 Description d'un Capteur**

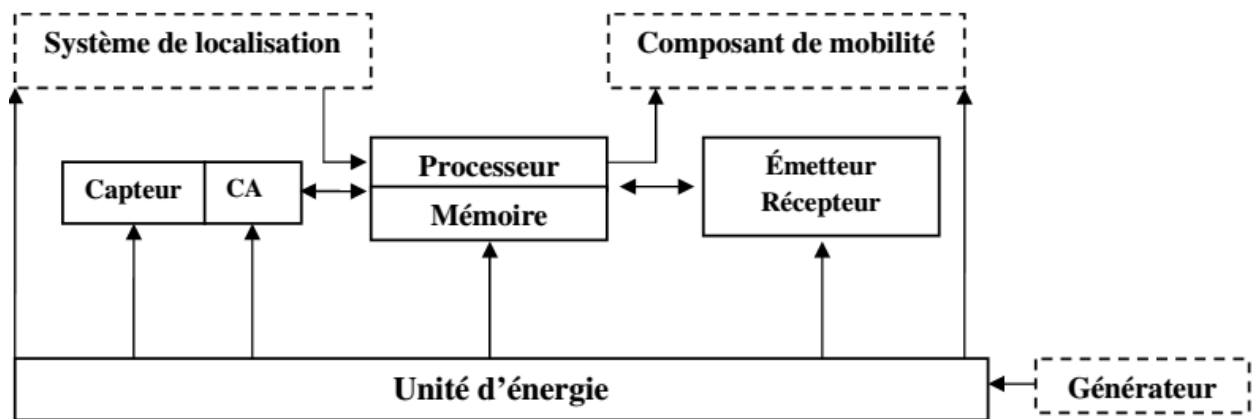
#### **1.2.1.1 Définition**

Un nœud capteur est une unité électronique mesurant une quantité physique (figure 2.01) [8]. Généralement, un capteur est constitué par :

- Une batterie (Unité d'énergie)
- Un processeur
- Une mémoire
- Un modem radio (Emetteur + Récepteur)



– Un système d'exploitation



**Figure 1.06 :** *Composantes d'un capteur*

#### 1.2.1.2 Propriétés

Un nœud capteur est un dispositif électronique qui dispose de trois propriétés essentielles :

- La capacité de capturer des données relatives à l'environnement où il est physiquement placé, c'est à dire qu'un capteur peut mesurer des valeurs relatives à son environnement et les convertir en signaux électriques à l'aide d'un CAN (Convertisseur Analogique / Numérique). Les données récupérées peuvent être de natures différentes et de même, la manière d'obtenir ces données est susceptible de varier.
- La capacité d'effectuer un traitement sur les données récupérées, soit par un système matériel (processeur) ou logiciel (système d'exploitation).
- La capacité d'échanger des données avec d'autres dispositifs, il s'agit des émissions et réceptions de données via un support de communication radio.

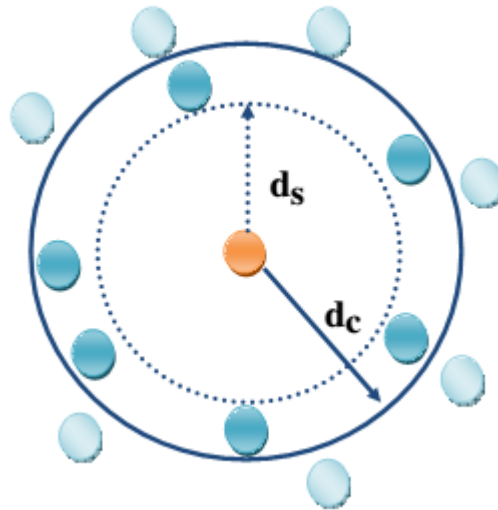
En d'autres termes, un nœud est composé de trois unités principales : l'unité d'acquisition, l'unité de traitement et l'unité de transmission. Ces trois unités sont alimentées par une seule batterie. Il y a d'autres composants pouvant être ajoutés à un nœud capteur, comme un système de localisation (GPS (Global Positioning System)), un composant de mobilité ou aussi un générateur d'énergie (batteries rechargeables), etc. On peut souligner aussi qu'il existe de nombreux types de capteurs : nous citons par exemple des capteurs thermiques, sonores, et des capteurs de mouvements. Un même capteur peut contenir plusieurs sous-systèmes pour récupérer différents types de données.

#### 1.2.1.3 Caractéristiques

Un capteur est associé à deux régions (figure 1.05) :

- **Région de capture (sensing region)** : est la zone géographique pour laquelle le capteur est capable de faire des mesures utiles. La distance maximale de mesure du capteur est la portée de capture (sensing range) que l'on notera  $d_s$ .

- **Région de communication (communication région)** : est la région où tous les dispositifs placés dedans peuvent recevoir les données envoyées par le capteur. La distance maximale entre le capteur et le dispositif de réception est la portée de communication (communication range) que l'on notera  $d_c$ .



**Figure 1.07 :** Région de capture  $d_c$  et de communication  $d_s$

La portée de capture  $d_s$  est inférieure à la portée de communication  $d_c$  :  $d_s < d_c$ . Il serait en effet inutile d'avoir une portée de communication inférieure à la portée de capture puisque dans le cas où le dispositif de réception est lui-même un capteur (dans le cas général). Ce capteur pourrait mesurer les mêmes valeurs (étant dans la région de capture) et les échanges de données seraient inutiles.

### **Contraintes**

Les capteurs sans fil permettent d'effectuer le monitoring de composants mécaniques, capter et transmettre des données sur la pression, la vibration, la force, l'acoustique, la température, la vitesse et la position des objets surveillés. La notion majeure qu'il est nécessaire d'évoquer lorsqu'on s'intéresse aux capteurs est celle des (nombreuses) contraintes imposées. Nous citons quatre contraintes:

- **Energie** : la contrainte majeure qui intervient dans la durée de vie du réseau concerne l'énergie. Les nœuds capteurs traditionnels sont des composants micro-électroniques équipés d'une source limitée en énergie. Ceci veut dire que dans certaines applications, ces nœuds ne peuvent pas être

dotés de mécanismes de rechargement d'énergie, par conséquent la durée de vie d'un nœud capteur dépend fortement de la durée de vie de la batterie associée. C'est pour ces raisons que le facteur de consommation d'énergie est d'une importance primordiale dans les réseaux de capteurs. La majorité des travaux de recherche menés actuellement se concentrent sur le contrôle de l'énergie afin de concevoir des algorithmes et protocoles spécifiques pour remédier à ce problème. Le coût énergétique nécessaire pour transmettre 1KB sur une portée de 100m est approximativement égal à celui nécessaire pour exécuter 3 millions d'instructions à une vitesse de 100 millions d'instructions par seconde.

- **Portée** : parallèlement à la contrainte énergétique, on trouve également une contrainte de la portée. La communication entre deux capteurs par exemple ne peut avoir lieu que si la distance qui les séparent n'est pas trop importante (quelques mètres en pratique). Plus cette distance est grande, plus le coût énergétique est élevé. De plus il faut être conscient du fait que s'il y a des obstacles (physiques) entre les deux capteurs, le coût énergétique augmente, et dans le pire des cas, la communication est impossible.

- **Bande passante** : une autre contrainte qu'il est important d'évoquer concerne la bande passante. Cette dernière est relativement réduite et les protocoles de communication qui sont développés doivent donc limiter au maximum les échanges sans "charge utile", c'est à dire limiter les données qui sont destinées au bon fonctionnement du protocole lui-même (overhead).

- **Sécurité** : d'un point de vue sécurité, la probabilité de détection d'une communication entre capteurs doit être réduite au minimum (la raison est évidente dans un contexte d'utilisation militaire) et le capteur doit être protégé contre les intrusions ou les perturbations affectant son comportement.

#### 1.2.1.4 Types de batterie

Une source d'énergie est nécessaire afin d'effectuer une tâche programmée. Cette source d'énergie est souvent constituée d'une batterie avec un budget (Limite / Illimite) en énergie. Par conséquent, on distingue deux types de batteries : des batteries rechargeables et des batteries non-rechargeables. Dans tous les cas, l'énergie est une ressource très importante qui doit être utilisée avec parcimonie.  
[9]

##### *a. Batteries non-rechargeables*

Traditionnellement, un capteur est principalement alimenté par une batterie non-rechargeable ayant une capacité de stockage d'énergie limitée. En outre, il pourrait être impossible de recharger la

batterie ou la remplacer, parce que les nœuds peuvent être déployés dans un environnement hostile ou peu pratique. Par conséquent, la batterie doit avoir une durée de vie assez longue pour répondre aux exigences de l'application. Donc, la maximisation de la durée de vie du réseau est indispensable pour assurer le fonctionnement des nœuds le plus longtemps possible. A cet égard, de nombreuses études ont été consacrées afin de prolonger la durée de vie du réseau en améliorant l'efficacité énergétique [9].

#### *b. Batteries rechargeables*

Récemment, et avec l'évolution que connaissent les WSNs, il est possible de récupérer l'énergie de l'environnement extérieur en utilisant des cellules solaires, par exemple, comme source d'énergie. Cependant, des sources d'alimentation externes présentent souvent un comportement non continu, de sorte qu'un tampon d'énergie (une batterie) est également nécessaire.

#### **1.2.2 Technologie de collecte d'énergie :**

Récemment, la notion de la récupération d'énergie a été proposée pour résoudre le problème de la durée de vie finie dans un WSN en permettant aux nœuds de capteurs de reconstituer l'énergie à partir de sources ambiantes, par exemple, en utilisant des panneaux solaires pour convertir la lumière solaire en électricité ou en utilisant des générateurs thermoélectriques [9].

#### **1.2.3 Méthodes de collecte d'énergie :**

Il existe principalement trois méthodes de récolte d'énergie, tel que mentionne ci- dessous :

##### **– Méthode photonique :**

Les cellules solaires en silicium exploitent l'effet photovoltaïque pour convertir la lumière solaire en électricité. Lorsque les photons de la lumière du soleil frappent la cellule de silicium, leurs énergies peuvent être absorbées et transférées aux électrons du silicium, qui sont alors en mesure de sortir de leurs positions normales dans le silicium pour devenir une partie du courant dans un circuit électrique. Ce phénomène est appelé l'effet photovoltaïque. L'énergie solaire est une source commode de récolte, beaucoup d'implémentations de nœuds de capteurs d'énergie solaire de récolte ont été effectué [9].

##### **– Méthode de vibration :**

Les vibrations peuvent générer de l'énergie électrique. Il existe principalement trois méthodes de récolte par vibrations à savoir les matériaux piézo-électriques, les systèmes inductifs et des systèmes capacitifs [9].

### – Méthode thermique :

L'effet thermoélectrique est la conversion directe des différences de température à tension électrique. Les dispositifs thermoélectriques utilisent cet effet et peuvent produire de l'électricité quand il existe un gradient de température à travers eux. Par rapport aux dispositifs à base de vibrations, les dispositifs thermoélectriques peuvent fonctionner pour une durée beaucoup plus longue en raison de l'absence de pièces mobiles.

#### 1.2.4 Architectures de récupération d'énergie :

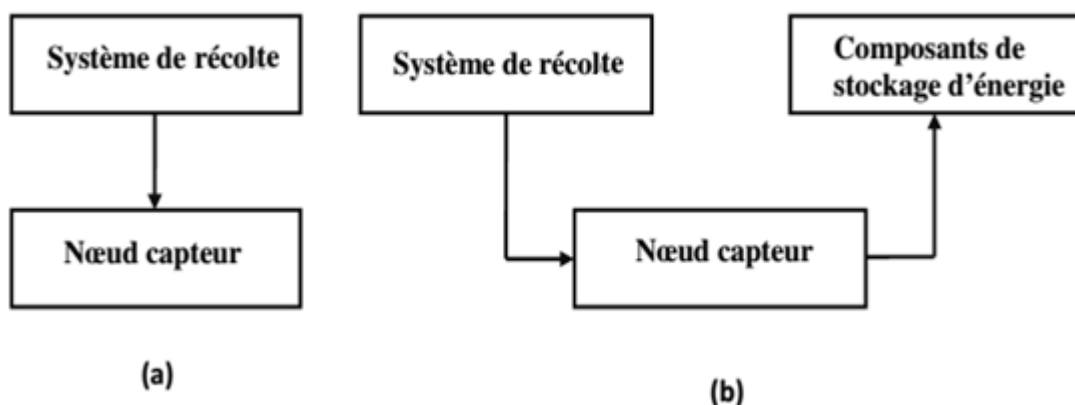
En général, les architectures de récupération d'énergie pour les nœuds de capteurs peuvent être divisées en deux architectures (figure 1.08), la récolte à usage "harvest use architecture" et Récolte-stockage-utilisation "harvest store use architecture" [9].

##### – Architecture de récolte à usage :

Comme le montre la figure 1.08 (a), le système de récupération d'énergie alimente le nœud de capteur directement. Par conséquent, afin de maintenir le capteur opérationnel, la sortie de la puissance du système de collecte doit être en permanence au-dessus du point de fonctionnement minimum. Sinon, le nœud de capteur est désactivé.

##### – Architecture Récolte-stockage-utilisation :

La figure 2.3 (b), représente l'architecture Récolte-stockage-utilisation, qui possède un composant de stockage d'énergie supplémentaire par rapport à l'architecture de récolte à usage. L'énergie est exploitée par le système d'exploitation et stockée dans l'élément de stockage d'énergie. Le stockage de l'énergie est très utile lorsque l'énergie récoltée est plus que le besoin actuel du capteur. L'énergie stockée peut être utilisée plus tard, quand il n'y a pas de possibilité de récolte.

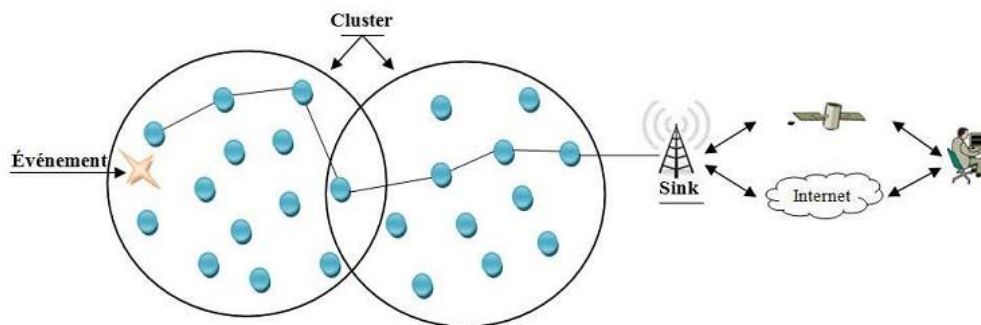


**Figure 1.08 :** Architectures de récupération d'énergie avec (b) et sans (a) la capacité de stockage de l'énergie.

### 1.2.5 Description d'un réseau de capteurs sans fil

#### 1.2.5.1 Définition

Les WSN sont considérés comme un type spécial de réseaux ad hoc. Les nœuds de ce type de réseaux consistent en un grand nombre de micro-capteurs capables de récolter et de transmettre des données environnementales d'une manière autonome [9]. La position de ces nœuds n'est pas obligatoirement prédéterminée. Ils sont dispersés aléatoirement à travers une zone géographique (Région de captage), qui définit le terrain d'intérêt pour le phénomène capté. Les données captées sont acheminées, la plupart du temps, grâce à un routage multi saut à un nœud considéré comme un "point de collecte", appelé sink ou centre de fusion (figure 1.07).



**Figure 1.09 : Exemple d'un réseau de capteurs sans fil**

Ce dernier peut être connecté à l'utilisateur du réseau via Internet ou un satellite. Ainsi, l'utilisateur peut adresser des requêtes aux autres nœuds du réseau, précisant le type de données requises et récolter les données environnementales captées par le biais du nœud puits.

Propriétés

#### - Densité « importante » des nœuds :

Les réseaux de capteurs sans fil sont des réseaux de plusieurs centaines (voire des milliers) de capteurs qui sont destinés à être déployés en pratique. On compte sur le nombre plutôt que la qualité des capteurs pour obtenir des résultats de bonne qualité. Nous visons donc à avoir des réseaux de capteurs denses. Le coût et le format des capteurs permettent de construire des réseaux à grand déploiement.

#### - Déploiement des nœuds :

La répartition géographique exacte des nœuds n'est pas prévisible (si l'on considère un modèle réaliste tout du moins) : un cas classique pour mettre en place un réseau de capteurs est de

simplement "jeter" les capteurs dans une zone géographique (par voie aérienne par exemple) en essayant de les répartir "au mieux" mais sans les déposer à des coordonnées précises. L'avantage des réseaux (denses) de capteurs étant justement que la probabilité de couvrir la zone d'intérêt est élevé même en déployant des capteurs avec une faible précision, du fait de leur nombre. Par ailleurs, un réseau peut être homogène (nœuds sont du même type) ou hétérogènes.

**- Protocoles robustes :**

Du fait d'une production "en masse" des capteurs, il devient tout à fait probable qu'un certain nombre d'entre eux soient défectueux ce qui implique que les protocoles doivent être robustes et tenir compte de cette contrainte.

**- Topologie dynamique :**

Un des aspects originaux des réseaux de capteurs réside dans la dynamique de leur topologie, c'est à dire que les nœuds peuvent être mobiles. C'est notamment le cas avec des réseaux formés par des robots, ou encore lorsque les capteurs eux-mêmes sont placés sur des entités mobiles. Il est donc clair, du fait de ces spécifications, que la réalisation des protocoles permettant l'exploitation des réseaux de capteurs n'est pas triviale.

**- Localité :**

Le principe de localité, notamment, est à privilégier, c'est à dire exploiter les données déjà possédées par un nœud pour réduire au maximum les communications.

**- Pannes fréquentes :**

La communication dans le réseau, entre capteurs, avec l'utilisateur ou avec un point de contrôle, est susceptible d'entraîner des erreurs, coûteuses et bruyantes. Puisque la topologie du réseau est dynamique, alors les capteurs peuvent tomber en panne ou être corrompus. Les ondes radio sont de courte portée pour empêcher la collision dans les transmissions. Les données doivent voyager à travers un grand nombre de capteurs pour arriver à destination. A chaque déplacement, il y a une probabilité d'avoir des erreurs ou un capteur en panne.

#### 1.2.5.2 Domaine d'application

Les propriétés des réseaux de capteurs sans fil leur permettent d'envahir plusieurs domaines d'applications [9]. En effet, ils se révèlent très utiles dans de nombreuses applications lorsqu'il s'agit de collecter et de traiter des informations provenant de l'environnement. Les WSNs peuvent offrir de meilleures contributions dans plusieurs domaines à savoir : Militaire, sécurité, environnemental,

domestique, santé, etc. Des exemples d'applications potentielles dans ces différents domaines sont exposés ci-dessous.

#### *a. Applications militaires*

Comme dans le cas de plusieurs technologies, le domaine militaire a été un moteur initial pour le développement des réseaux de capteurs. Les propriétés de WSN comme le déploiement rapide, le coût réduit, l'auto-organisation et la tolérance aux pannes rendent ce type de réseaux efficace dans le domaine militaire. Ils peuvent être utilisés pour surveiller toutes les activités des forces ennemies, ou d'analyser le terrain avant d'y envoyer des troupes (figure 1.08).



**Figure 1.10 :** *Exemple d'une application militaire.*

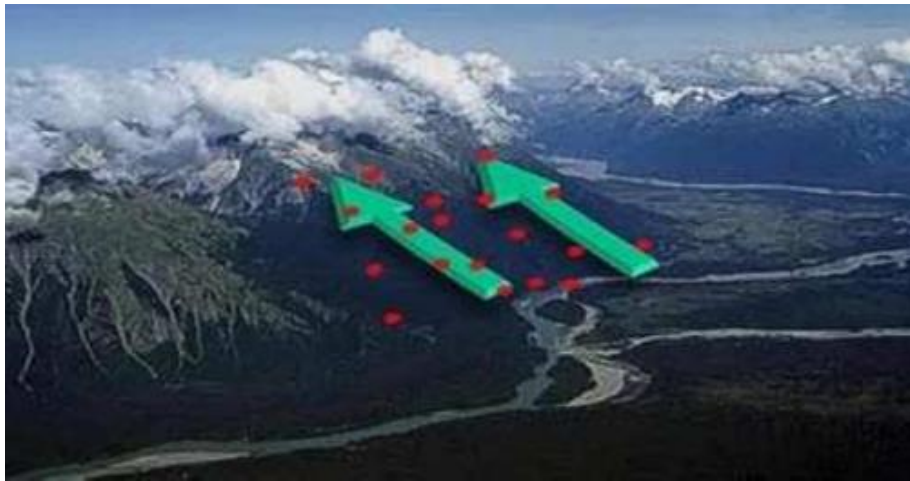
#### *b. Applications à la sécurité*

Au niveau de la sécurité des constructions, les capteurs peuvent être intégrés dans les murs ou le béton d'un bâtiment afin de détecter un séisme ou un vieillissement, sans alimentation électrique ou autres connexions filaires. Dans ce cas, les capteurs doivent fonctionner durant des années, voir des décennies. Ces capteurs peuvent être utilisés pour surveiller des voies ferrées afin de prévoir à l'avance des accidents. En outre, un réseau de capteurs de mouvements peut constituer un système d'alarme distribue qui servira à détecter les intrusions sur un large secteur. L'application des réseaux de capteurs dans le domaine de la sécurité peut largement réduire les coûts des lieux afin d'assurer la protection des êtres humains tout en garantissant de meilleurs résultats.



### *c. Applications environnementales*

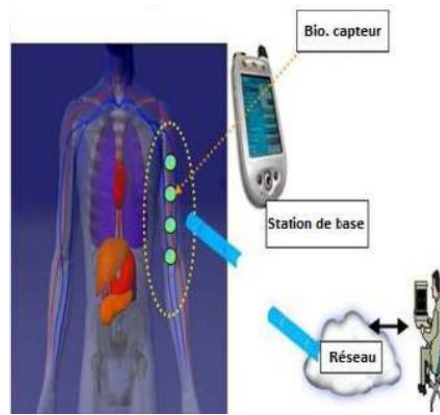
Ces réseaux peuvent être déployés dans des environnements hostiles comme les volcans, les sites de tremblements de terre, ou même dans les forêts, dans l'air, la mer etc. Ces capteurs ont pour but de recueillir diverses informations sur l'état du milieu naturel et sur les comportements de déplacement. Parmi les types de capteurs, il existe des thermo-capteurs qui peuvent être utilisés afin d'avertir un éventuel début d'incendie dans une forêt. Ceci peut permettre de lutter efficacement contre les feux de forêt. Dans les champs agricoles, il est possible d'identifier facilement les zones sèches et rendre l'irrigation efficace en utilisant la plantation des capteurs avec les graines. Sur les sites industriels, les centrales nucléaires ou dans les pétroliers, il est possible de déployer des capteurs pour détecter des fuites de produits toxiques (gaz, produits chimiques, éléments radioactifs, pétrole, etc.) (figure 1.11).



**Figure 1.11 :** *Exemple d'une application environnementale.*

### *d. Applications médicales*

Les réseaux de capteurs peuvent également être utiles dans le domaine de la médecine. Ils permettent de surveiller d'une manière permanente les patients et collecter des informations physiologiques afin de faciliter le diagnostic de quelques maladies. La surveillance des fonctions vitales de l'être humain est possible avec l'utilisation des micro-capteurs pouvant être avalés ou implantés sous la peau. Désormais, il est possible de transmettre des images de l'intérieur d'un corps humain avec une autonomie de 24 heures sans faire appel à la chirurgie. D'autres ambitieuses applications biomédicales sont aussi présentées, tel que la surveillance du niveau de glucose, le monitoring des organes vitaux ou la détection des cancers (figure 1.12).



**Figure 1.12 :** *Exemple d'une application médicale.*

#### *e. Applications commerciales*

L'intégration des nœuds capteurs au processus de stockage et de livraison peut permettre à un client de connaître la position actuelle de la livraison de son paquet à tout moment. Pour les entreprises manufacturières, les réseaux de capteurs permettent de suivre à distance la fabrication des produits du début jusqu'à la fin. Dans les immeubles, le système de climatisation peut être conçu en intégrant plusieurs micro-capteurs dans les tuiles du plancher et les meubles. Ainsi, La climatisation pourra être déclenchée seulement aux endroits où il y a des personnes présentes et seulement si c'est nécessaire.

Les réseaux de capteurs sans fil sont en train de vivre un développement continu guidé par l'intégration de l'Internet et des applications multimédias. Ceci implique non seulement une augmentation du débit mais aussi une garantie de la qualité de service. Les systèmes MIMO et l'amélioration de performances qu'ils assurent dans ce genre de problème sont unanimement reconnus. Cependant, ces petits capteurs ne peuvent pas tous bénéficier de ces technologies multi-antennaire vu leurs taille. D'où l'idée de la coopération [9].

#### **1.2.6 Communication coopérative**

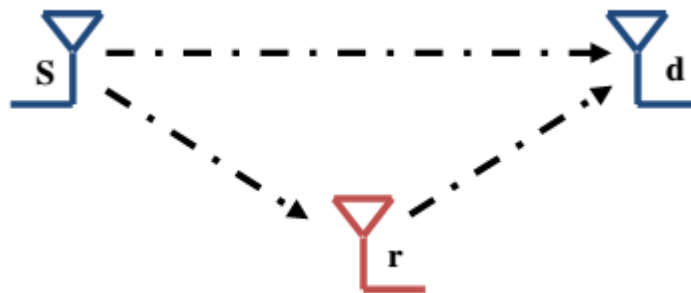
La communication coopérative est une technique où la source transmet ses propres données à la destination passant par un ou plusieurs relais au lieu de l'atteindre directement. Nous considérons une source "s" qui veut envoyer un signal à une destination "d". Il existe alors deux possibilités :

1. La source "s" envoie son information directement vers la destination "d" sur le lien direct (figure 1.13). Dans ce cas, on se trouve dans un système Single Input Single Output (SISO).



**Figure 1.13 :** *Communication Non coopérative*

2. Un autre nœud, dit relais  $r$ , aide la source  $s$  à envoyer son signal (figure 1.14). La transmission a lieu alors à la fois, sur le lien direct  $s \rightarrow d$  et sur le lien indirect  $s \rightarrow r \rightarrow d$ . On se trouve dans le cas d'un système multi-antennes virtuel dit système coopératif.



**Figure 1.14 :** *Communication coopérative*

Pour le trajet source-relais-destination, l'ajout du relais augmente la diversité définie comme le nombre de canaux indépendants et ça nous ramène en quelque sorte à un système, avec deux antennes en émission (source et relais) et une antenne en réception (destination). Cette diversité lisse les effets de fading et résout alors le problème des fluctuations à court terme dues au phénomène d'évanouissement, accompagnant la transmission sur le canal [9]. La coopération résout également le problème des fluctuations à moyen terme "shadowing". Ces derniers ont lieu dans le cas où deux nœuds communicant sont séparés par un certain obstacle, l'utilisation d'un troisième nœud est alors une solution probable [9].

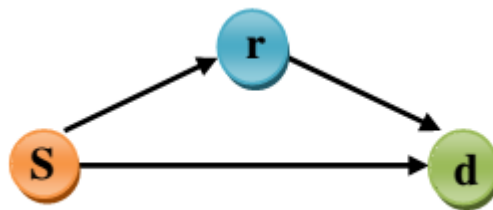
Les avantages de la coopération se traduisent par une amélioration des performances du système pour la même puissance à la transmission, ou éventuellement une réduction de cette puissance pour les mêmes performances.

Nous présentons, dans ce qui suit, les différentes architectures de relayage ainsi que quelques protocoles de coopération. Ces derniers prennent en compte certaines contraintes d'implémentation concernant les types de canaux d'une part (half ou full duplex) et la méthode d'accès au canal (orthogonal et non-orthogonal) d'autre part.

Au cours d'une communication coopérative on peut avoir plusieurs architectures de relayage. Dans cette partie on cite cinq cas de coopération :

#### 1.2.6.2 Coopération via un simple relais

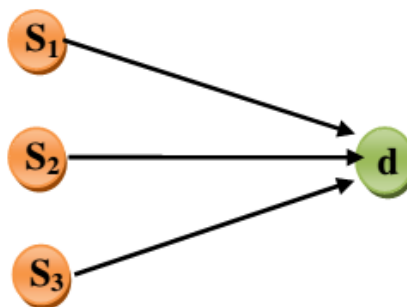
On commence par le 1<sup>er</sup> cas, c'est le plus classique, comme il est indiqué dans l'architecture de la figure 1.15, qui est aussi appelé "trois corps", avec "s" la source, "r" le relais et "d" la destination. La source diffuse le signal à la fois au relais et à la destination. Par la suite, le relais retransmet l'information vers la destination.



**Figure 1.15 :** *Coopération via un seul relais.*

#### 1.2.6.3 Coopération via une communication directe

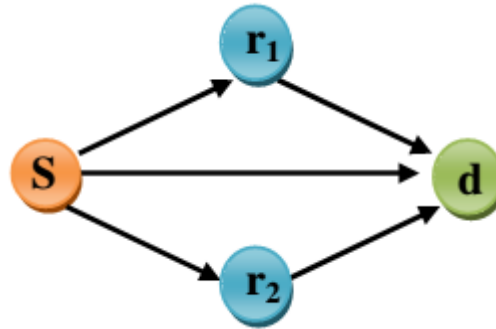
Dans le 2<sup>ème</sup> cas, les sources  $\{S_1, S_2, S_3\}$  transmettent l'information d'une manière simultanée à la destination d, ceci amené à une chaîne comme le montre la figure 1.16 où il y a une communication directe avec la destination.



**Figure 1.16 :** *Coopération avec une communication directe.*

#### 1.2.6.4 Coopération via des multi-Branches

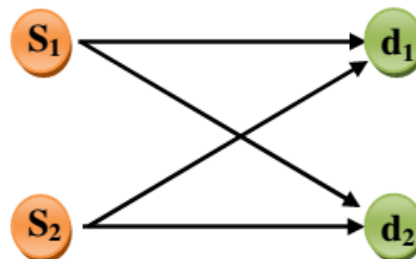
Le 4<sup>ème</sup> cas comme la figure 1.17 illustre la source qui transmet ses données au relais avant d'atteindre la destination tout en créant plusieurs branches.



**Figure 1.17 :** *Coopération via des multi-Branches.*

#### 1.2.6.5 Coopération via clustering

Le 5<sup>ème</sup> cas illustre une architecture où un relais près de la source et un autre près de la destination coopèrent. Ceci amène à un simple “cluster” à la communication (figure 1.18).



**Figure 1.18 :** *Coopération via clustering.*

### 1.2.7 Différents niveaux de coopération

Dans la littérature, plusieurs travaux de recherche ont été proposés dans le cadre des communications coopératives. Cette partie est dédiée à l'étude de l'impact de l'intégration de la communication coopérative dans les réseaux de capteurs, particulièrement au niveau de la couche physique, MAC (Medium Access Control) et réseau.

#### 1.2.7.1 Couche physique

Au niveau de la couche physique, la coopération se fait généralement en effectuant des constructions du signal par des transmissions simultanées du même signal via plusieurs nœuds. Cette technique est connue sous le nom de STC (Space Time Coding). Notons qu'il existe de nombreuses contributions qui ont été proposées dans ce cadre [9].

A ce niveau, on peut distinguer deux types de coopération, à savoir synchrone ou asynchrone.

##### *a. Synchrone*

Dans une communication coopérative synchrone, il est nécessaire d'avoir une synchronisation entre les relais, et sans cette dernière le récepteur ne peut pas décoder le signal reçu car la synchronisation aide à éviter les collisions entre les relais. La synchronisation peut être faite à l'aide de l'envoi périodique de micro-paquets appelés préfixe de synchronisation. Pour obtenir une synchronisation parfaite entre les relais, il est nécessaire d'avoir beaucoup de trafic de contrôle ainsi que des horloges à haute précision ce qui augmente la complexité ainsi que le coût de la communication [9].

##### *b. Asynchrone*

Une autre classe de communication coopérative est apparue, il s'agit de la coopération asynchrone. Afin d'éliminer les contraintes de la synchronisation des relais, plusieurs protocoles ont été proposés avec des coopérations asynchrones. On cite le travail où les auteurs ont proposés un schéma distribué et asynchrone de STC. Dans ce schéma, les relais n'ont pas besoin de décoder le paquet à relayer. Ils construisent une matrice de code qu'ils vont utiliser afin de relayer le paquet.

#### 1.2.7.2 Couche MAC

Au niveau de la couche MAC, les nœuds sont en compétition entre eux pour accéder au canal, on parle des méthodes d'accès au canal. Dans les protocoles MAC coopératifs, les nœuds s'aident mutuellement pour un meilleur usage du canal radio. Chaque nœud a un lien qui connaît des perturbations, il demande de l'aide à des relais de ses voisinages qui ont une meilleure qualité du canal; on parle d'un canal à relais.

A l'aide de la coopération, le signal est relayé par un ou plusieurs voisins. Le problème qui se pose maintenant est au niveau de la sélection et du nombre de relais qui vont coopérer.

Concernant le choix des relais [9], il est possible d'utiliser :

- Tous les relais : dans ce cas, tous les voisins communs de la source et de la destination interviennent dans le relayage du paquet.
- Les M meilleurs relais : dans ce cas la destination supervise de façon continue l'état des canaux avec ses relais et choisit par la suite les M relais ayant les meilleurs canaux.
- Un seul relais : on parle d'un canal à un seul relais.

#### 1.2.7.3 Couche réseau

Au niveau de la couche réseau, deux objectifs de communication doivent être atteints : établir une route stable et contrôler la topologie du réseau [9].

#### 1.2.7.4 Etablir une route stable

Un chemin de routage peut être rompu soit par un canal perturbé ou une énergie non disponible. Puisque la coopération à cette échelle est souvent représentée par la qualité des liens, notre problème est donc comment avoir une route ayant de très bons liens mais qui ne nécessite pas trop d'énergie. Autrement dit, l'objectif est de maintenir un chemin avec le moins de trafic de contrôle possible. Dans le but d'inclure et de bénéficier de la coopération au niveau du routage, certains travaux ont opté pour des protocoles de routage non coopératifs déjà existants et ils ont essayé de les modifier et d'y inclure la coopération (c'est le cas pour de AODV (Ad-hoc On-Demand Distance Vector)) [9]. D'autres chercheurs ont décidé de concevoir de nouveaux protocoles coopératifs. Dans le même but, en 2009 des auteurs ont introduit la théorie des jeux pour trouver des routes économes en énergie en utilisant la coopération. Chaque nœud se base sur un historique contenant les communications et les coopérations précédentes afin de prendre des décisions de coopérations [9].

### 1.2.8 *Contrôler la topologie du réseau*

Afin de pouvoir améliorer un lien donné, un capteur doit émettre avec plus de puissance et donc plus de consommation d'énergie. Le routage fondé sur la coopération propose de faire coopérer les nœuds du réseau afin de minimiser l'énergie requise à l'émission et d'augmenter leur portée. Généralement, le contrôle de topologie a pour objectif de minimiser la puissance de transmission tout en gardant un réseau connecté. L'objectif c'est de garantir la connectivité des nœuds dans le réseau tout en assurant une consommation d'énergie économe. Le problème de contrôle de topologie à l'aide de la communication coopérative est classé comme un problème NP-complet. Les nœuds utilisent les informations concernant la qualité des liens de leurs voisins à deux sauts pour optimiser

la consommation d'énergie en faisant de la coopération. La communication coopérative permet aussi d'étendre la portée des antennes en faisant coopérer des nœuds voisins. On peut même grouper les nœuds en créant des clusters pour faciliter la coopération. Les nœuds d'un cluster organisent leurs transmissions afin d'atteindre des nœuds plus lointains et de renforcer les liens radios.

### **1.3 Conclusion**

Dans ce chapitre nous avons vu dans ce chapitre la présentation de l'environnement de propagation des ondes radio ainsi qu'une vue d'ensemble sur les réseaux de capteurs. Nous avons ainsi pu constater les principaux paramètres qui entreront en jeu dans la composition de ce mémoire. Nous avons aussi exposé le fonctionnement basique des transmissions d'informations dans un réseau de capteurs sans fils.



## CHAPITRE 2

### EVALUATION DES SYSTEMES MIMO

#### 2.1 Introduction

Multiplier le nombre d'antenne en émission et/ou en réception peut permettre d'améliorer les performances d'un système de communication sans-fil. Parmi ces solutions on cite la diversité de réception permettant de minimiser l'effet de l'évanouissement. De plus, on démontre qu'il est possible de multiplexer des signaux dans l'espace. Grâce à la décorrélation spatiale apportée par des différents trajets, il est possible d'émettre des signaux utilisant la même bande spectrale. C'est ce que la technologie MIMO propose.

Le besoin de transmettre de l'information dans des environnements complexes tout en augmentant le débit a donné lieu à une solution originale qui fait appel non seulement à plusieurs antennes en réception mais aussi en émission. Cette technique connue sous l'appellation MIMO permet en utilisant la même bande spectrale de transmettre plus de débit ou d'améliorer la qualité de liaison. Plusieurs techniques sont proposées dans la littérature. Nous présentons ici les principales techniques qui exploitent les systèmes MIMO.

#### 2.2 Modèle théorique d'un système MIMO

Un système MIMO comprend  $N$  antennes d'émission et  $M$  antennes de réception et est représenté sur la Figure 2.03, avec  $x$  un vecteur de  $N$  symboles.

$$x = [x_1 \dots x_t \dots x_N]^T \quad (2.01)$$

correspondant aux symboles à transmettre sur chacune des  $N$  antennes d'émission pendant une durée symbole  $T_s$ ,  $H$  est la matrice du canal de dimension  $N \times M$ .

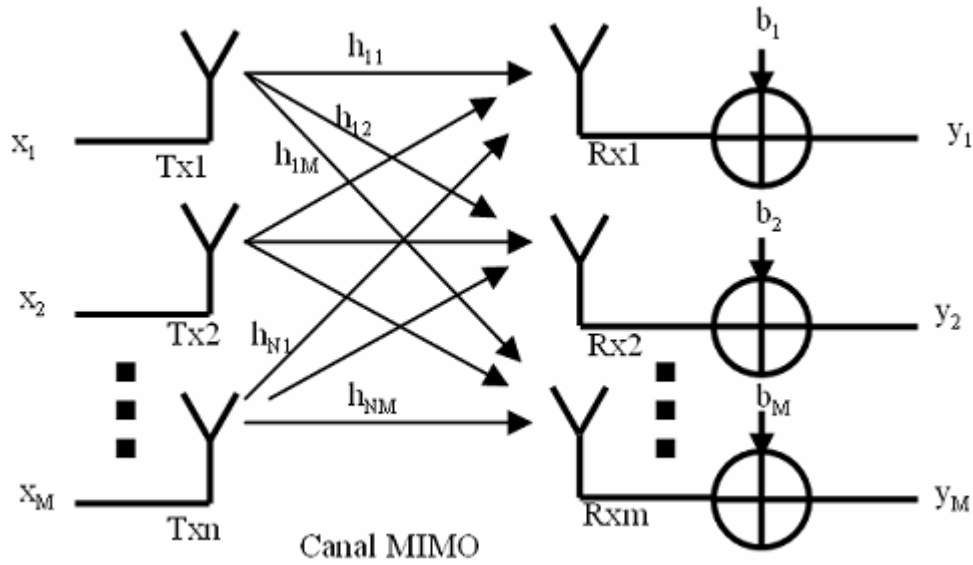
$$H_k = \begin{bmatrix} h_{11} & \dots & h_{1N} \\ \vdots & \ddots & \vdots \\ h_{M1} & \dots & h_{MN} \end{bmatrix} \quad (2.02)$$

correspondant au canal multi-antennes avec  $h_{mn}$  le canal entre la  $n^{\text{ième}}$  antenne d'émission et la  $m^{\text{ième}}$  antenne de réception, et en considérant  $b$  le vecteur de bruit des  $M$  antennes de réception tel que :

$$b = [b_1 \dots b_t \dots b_N]^T \quad (2.03)$$

alors  $y$  le vecteur reçu s'exprime de la manière suivante :

$$y = Hx + b \quad (2.04)$$



**Figure 2.01 :** *Système MIMO  $N \times M$*

On considère généralement que le bruit est négligeable pour pouvoir reconstruire le signal transmis. Ainsi l'expression 2.04 peut s'écrire :

$$y = Hx \quad (2.05)$$

Connaissant y et H la relation qui permet de reconstruire x est la suivante :

$$x = H^{-1}y \quad (2.06)$$

Mathématiquement, résoudre cette dernière équation requiert que H soit inversible ce qui veut dire que H est de rang plein (sa dimension est égale à son rang). Physiquement, ceci se traduit par des atténuations et des déphasages différents subits par chaque signal transmis. C'est pour cela qu'il est essentiel d'avoir un environnement riche en multi-trajets pour bénéficier pleinement des apports d'un système MIMO [10].

### 2.3 Les techniques MIMO

Les systèmes MIMO nous permettent d'exploiter deux dimensions distinctes d'une liaison radio. La première étant la diversité et la seconde étant la capacité. Nous allons présenter les différentes techniques et conclure sur l'apport de chacune. Selon la connaissance ou non de l'état du canal (CSI (Channel State Information)) en émission ou/et en réception, différentes solutions sont présentées pour exploiter au mieux le système MIMO. Ces solutions sont plus ou moins optimales du point de vue de la diversité ou de la capacité [10].

### 2.3.1 Systèmes multi-antennes avec CSI à la réception

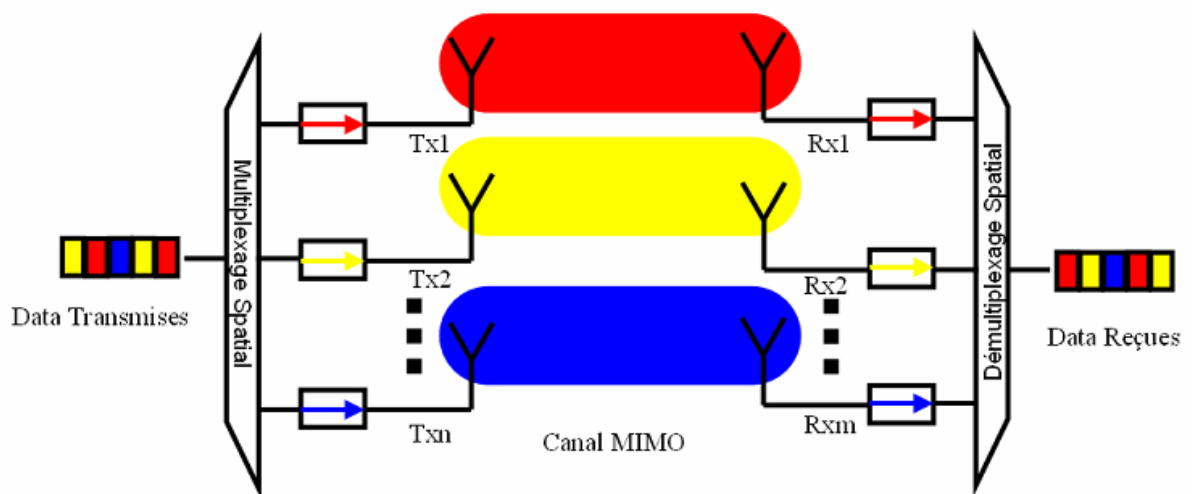
Deux principales techniques multi-antennes avec CSI à la réception existent : la technique MIMO avec multiplexage spatial et la technique MIMO avec codage espace-temps.

Dans les deux paragraphes suivants, la technique du multiplexage spatial où des données sont démultiplexées avant d'être émises sur différentes antennes est détaillée. Ensuite la technique du codage espace-temps est décrite. La seule différence entre ces deux techniques provient de la redondance introduite par l'utilisation ou non d'un code espace-temps. Les familles de codes espace-temps aboutissent à plusieurs rendements de code  $R$ , le rendement de code étant défini comme le rapport entre le nombre de symboles  $N_s$  et la durée  $T$  pendant lequel ces symboles ont été émis soit  $R = N_s/T$ .

Les systèmes à multiplexage spatial exploitent la capacité optimale du système MIMO puisque  $R = N$  alors que les codes espace-temps ne l'exploitent pas puisque  $R \leq 1$ . Dans les deux cas, l'expression de la capacité de chacune des techniques est donnée, cela permet d'identifier l'augmentation de l'efficacité spectrale, l'une des caractéristiques principales des systèmes MIMO.

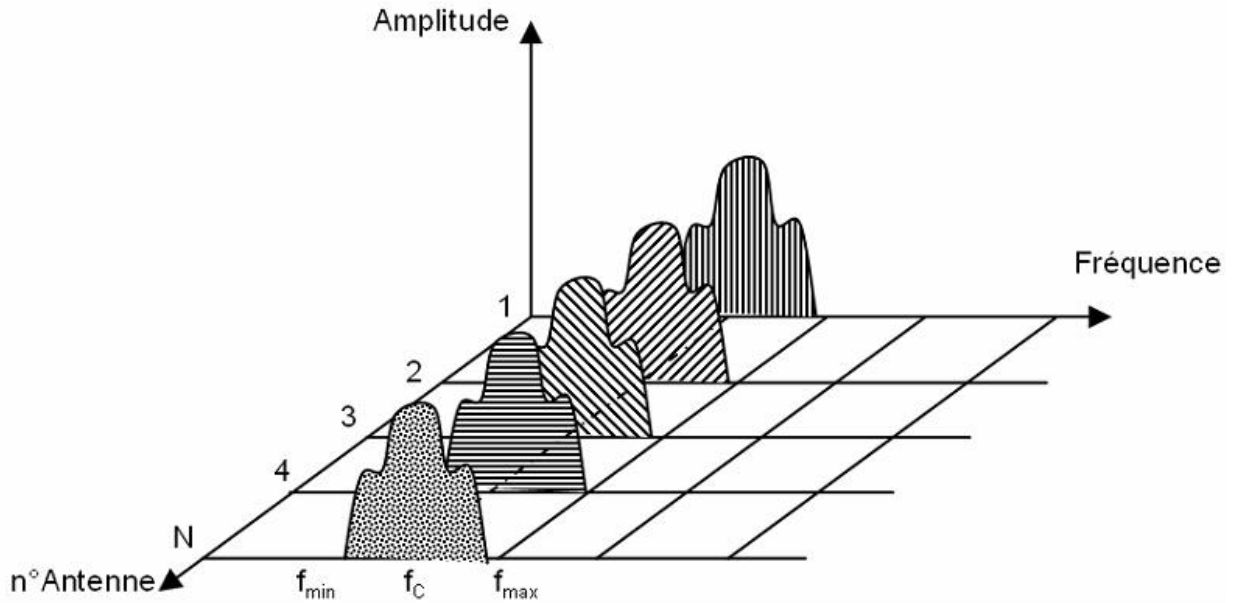
#### 2.3.1.1 Systèmes MIMO à multiplexage spatial

Le système MIMO à multiplexage spatial exploite la diversité spatiale pour transmettre plusieurs flux d'information dans la même bande spectrale attribuée. L'information à transmettre est divisée selon le nombre d'antennes émettrices avec un codage canal. Ce codage permettra par la suite d'estimer la matrice  $H$  du canal en réception. Ici les antennes rayonnent d'une manière aveugle sans aucune préférence de direction particulière.



**Figure 2.02 :** *Système de transmission MIMO à multiplexage spatial*

La puissance totale transmise est équi-répartie sur les différentes antennes (figure 2.02). Le nombre de chaînes radio est égal au nombre d'antennes. Ainsi, nous pouvons représenter au niveau de l'émission, la répartition du signal en fonction de l'antenne et du spectre fréquentiel comme le montre la figure 2.03.



**Figure 2.03 :** Répartition spectrale et spatiale du signal RF dans un système MIMO à multiplexage spatial

L'objectif de ce type de technique est de réaliser  $N$  liaisons SISO. Techniquement, dans un contexte SISO pour séparer les différents systèmes, on évite d'utiliser le même spectre fréquentiel pour ne pas créer de l'interférence co-canal. Grâce à la décorrélation des signaux apportés en réception par le canal à multi-trajets, les différentes sources pourront être séparées. On crée ainsi des sous canaux opérant en parallèle.

La capacité est un outil mathématique permettant de caractériser au mieux une liaison Radio. Elle permet aussi de modéliser la liaison. Les études concernant la capacité d'un canal MIMO de Rayleigh ont été menées par Telatar dans [10]. La capacité ergodique en bps/Hz d'un système MIMO à multiplexage spatial est:

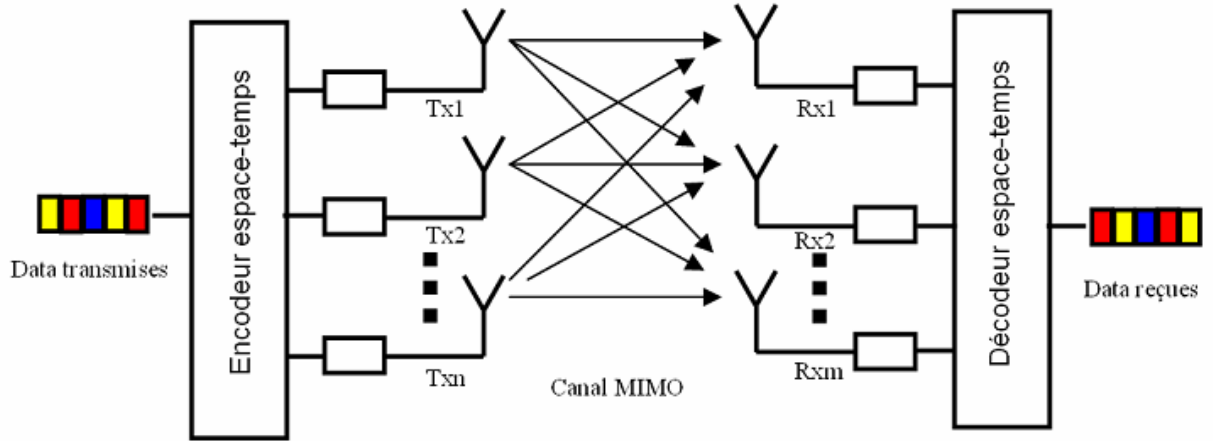
$$C = E \left\{ n \log_2 \left[ 1 + \frac{\gamma}{N} \sum_{l=1}^m |h_l|^2 \right] \right\} \quad (2.07)$$

avec  $n = \min(N, M)$  et  $m = \max(N, M)$ . Cette capacité représente  $n$  systèmes SISO indépendants exploitant une diversité de  $m$  canaux  $h_l$  de la Figure 2.02 et  $\gamma$  est le rapport signal à

bruit au niveau d'une antenne élémentaire de réception. Plusieurs publications traitent la capacité des systèmes multi-antennes selon différents modèles de propagations [10].

### 2.3.1.2 Systèmes MIMO à codage espace-temps

Le codage espace-temps peut être vu comme un code à répétition. Après avoir dérivé la capacité d'un code à répétition, nous donnons la capacité de n'importe quel code espace-temps.



**Figure 2.04 :** *Système MIMO à codage espace-temps*

Considérons une transmission de mêmes symboles de puissance  $P/N$  sur chaque antenne (figure 2.04). Cette capacité correspond à l'utilisation de codes espace-temps apportant la diversité totale en émission et en réception. La capacité en bps/Hz de ce système utilisant un code espace-temps de rendement  $R = \min(N, M)/T$  à gain de diversité  $N \times M$  est :

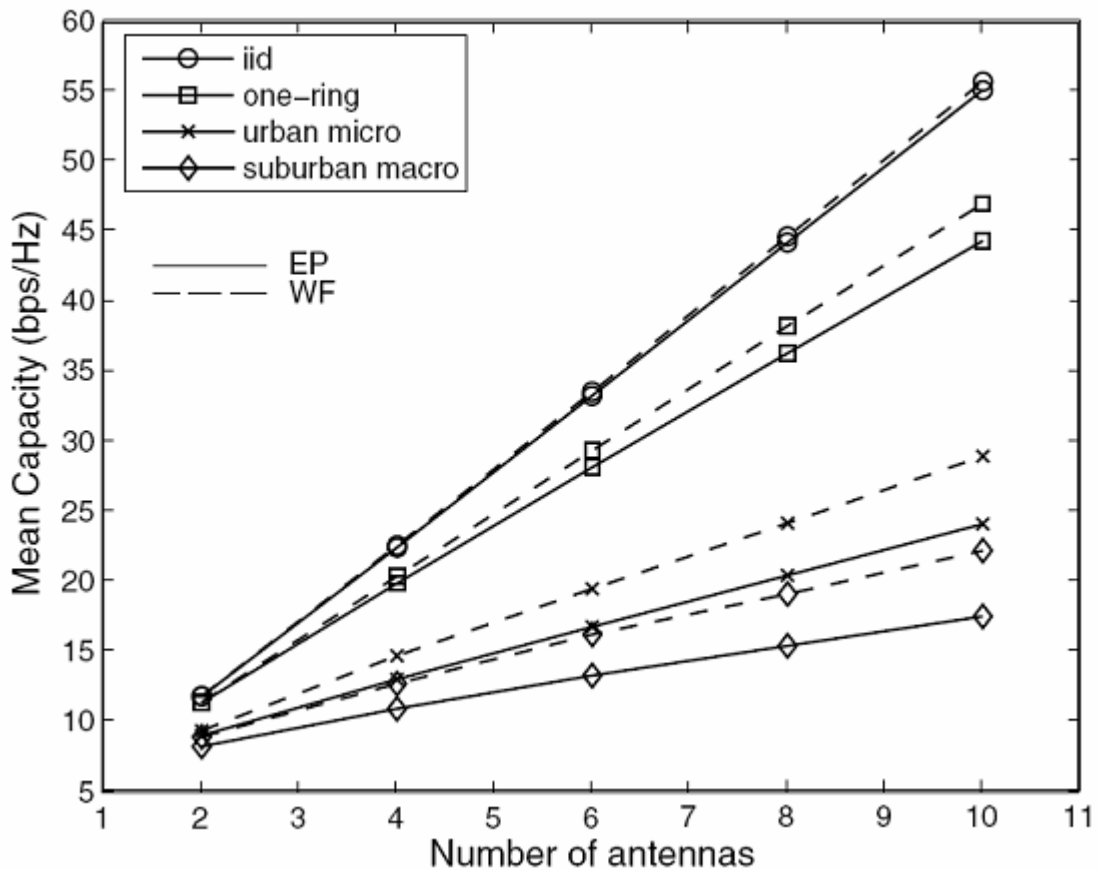
$$C = E \left\{ n \log_2 \left[ 1 + \frac{\gamma}{N} \sum_{l=1}^{N \times M} |h_l|^2 \right] \right\} \quad (2.08)$$

Comme la puissance est normalisée en émission, c'est à dire la même puissance  $P$  est émise quel que soit le nombre d'antennes d'émission, la capacité d'un code espace-temps de rendement 1 ne dépend que du nombre d'antennes de réception. S'il n'existe qu'une seule antenne de réception, la capacité moyenne de tout code espace-temps à rendement unitaire est identique à la capacité moyenne d'un système SISO. Par contre, si la puissance en émission n'est pas normalisée, c'est à dire  $P \times N$  est émis sur le réseau d'antennes, la capacité d'un code espace-temps à rendement 1 est fonction à la fois du nombre d'antennes de réception et d'émission.

La figure 2.05 représente la capacité d'un canal MIMO simulée en fonction du nombre d'antenne pour deux cas différents de connaissance de canal [10]. Dans le cas EP (Equal Power) le canal de Rayleigh est connu uniquement par le récepteur. Dans le cas WF (Water-filling) c'est au niveau de

l'émetteur et du récepteur que le canal est connu; dans ce cas, on utilise la technique d'allocation de puissance, appelée technique du water-filling.

Le terme EP désigne que le système MIMO émet au niveau des antennes une puissance identique par contre dans le second cas, WF, la puissance est répartie selon l'état du canal.



**Figure 2.05 :** Capacité MIMO (EP, WF) en fonction du nombre d'antenne pour différents modèles de canaux de propagations (IID, one-ring, urbain et suburbain) pour un SNR=20dB

Ces simulations ont été faites pour différents modèles de canal de propagation dont le IID (Indépendant Identique Distribution), le one-ring, le urbain et le suburbain. Le nombre d'antennes utilisées pour cette simulation est identique au niveau du récepteur et de l'émetteur. Le rapport du signal à bruit est de 20dB et la distance entre les éléments est de  $\lambda/2$ . On peut observer une augmentation linéaire en fonction du  $\min(N,M)$  dans le cas d'un multiplexage spatial. Cela reste vrai selon les auteurs [10] uniquement pour des valeurs élevées de rapport signal à bruit. En effet, d'après les résultats qu'ils obtiennent pour un SNR=3dB en doublant le nombre d'antenne (de 2 à 4) en émission et en réception une augmentation de 40% seulement est observée pour le cas du canal suburbain et une allocation EP.

### 2.3.2 *Systèmes multi-antennes avec CSI à l'émission et à la réception*

Afin de reconstituer le signal émis et perturbé par le canal de transmission au niveau du récepteur, il est également possible de renvoyer la connaissance des perturbations du canal de transmission du récepteur vers l'émetteur. L'émetteur peut alors modifier le signal à émettre en fonction des caractéristiques du canal de transmission afin de traiter de façon optimale le signal reçu au niveau du récepteur.

Dans cette partie, deux techniques multi-antennes avec CSI à l'émission et à la réception sont traitées, c'est à dire le water-filling ou la formation de faisceaux. Cependant, il est vrai que la formation de faisceaux peut être mise en œuvre séparément à l'émission ou à la réception. Il est un canal quelconque connu par le transmetteur et par le récepteur.

Dans les deux paragraphes suivants, la capacité de la technique du water-filling où des données différentes sont démultiplexées avant d'être émises sur différentes antennes est décrite. Ensuite la capacité de la technique de la formation de faisceaux est décrite.

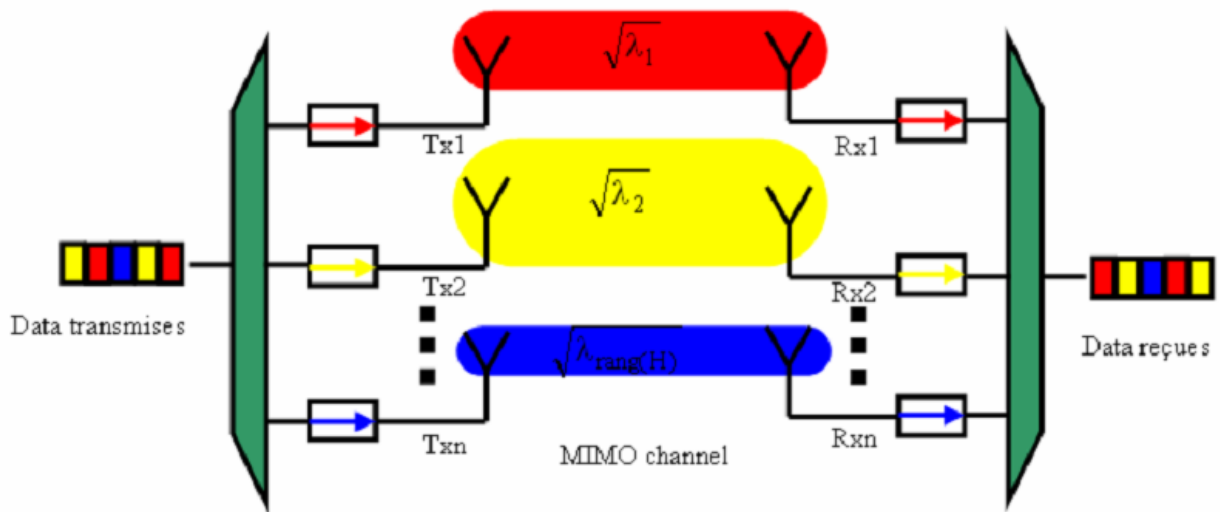
Dans le cas du water-filling, l'émetteur va pouvoir sélectionner les valeurs propres du canal multi-antennes afin d'optimiser le lien et la capacité grâce à une allocation de puissance. Avec la formation de faisceaux, l'émetteur optimise le lien avec le récepteur en dirigeant le faisceau du réseau d'antennes vers les directions d'arrivées des ondes émises. La seule différence entre ces deux techniques provient de la redondance introduite dans le cas de la formation de faisceaux puisque le même symbole est émis sur chaque antenne alors que des symboles différents sont émis sur les différentes antennes dans le cas du water-filling. Il est aussi possible d'associer des codes espace-temps aux deux techniques. Par conséquent, les systèmes utilisant des techniques de water-filling exploitent la capacité optimale du système MIMO puisque  $R = N$  alors que les systèmes utilisant des techniques de formation de faisceaux ne l'exploitent pas car  $R = 1$ . [10]

#### 2.3.2.1 Systèmes MIMO utilisant des techniques de water-filling

Lorsque l'émetteur et le récepteur connaissent le canal, la solution optimale au sens de la capacité est une solution de water-filling. Les puissances sont réparties de manière différente sur les antennes afin d'exploiter au maximum la capacité en revalorisant les faibles valeurs propres du canal ( $H$ ) multi-antennes.

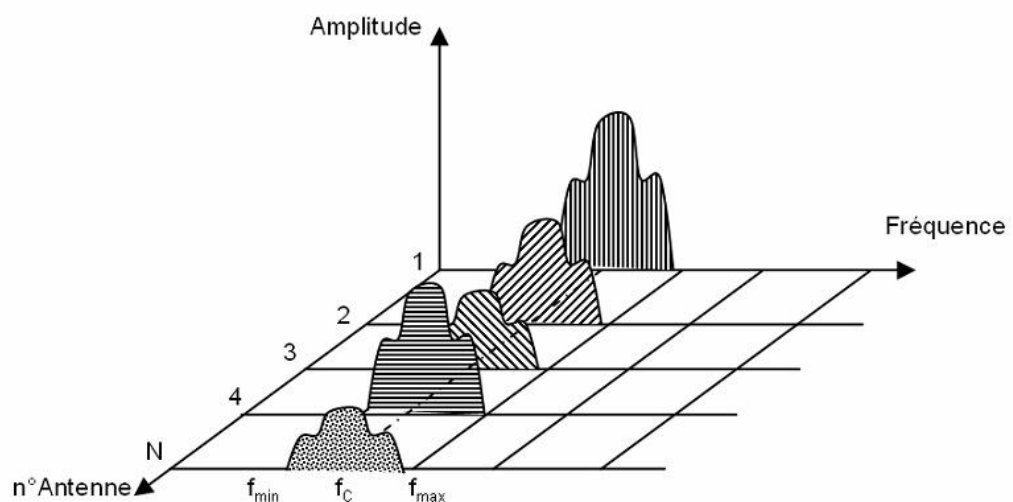
Dans le cas du water-filling (allocation de puissance), on cherche à optimiser la liaison émission-réception en répartissant les puissances sur les antennes d'émission de manière à obtenir la capacité optimale. Pour cela, différents symboles sont émis sur les différentes antennes d'émission. En

pratique, la puissance d'émission est limitée donc on ne parvient pas à atteindre totalement la capacité optimale mais une bonne fraction de celle-ci.



**Figure 2.06 :** *Systèmes MIMO utilisant des techniques de water-filling*

Ce système à water-filling transmet les données sur les différentes antennes dont les puissances dépendent des différentes valeurs propres et permet d'obtenir de meilleurs résultats tels que le décrit la figure 2.06. Connaissant la matrice du canal  $H$ , une première étape consiste à déterminer les vecteurs propres de cette matrice. Ces vecteurs propres sont directement liés à l'atténuation apportée par le canal à chaque onde émise. Cela permettra par la suite d'estimer la répartition de puissance adéquate pour atteindre la capacité optimale. [10]



**Figure 2.07 :** *Répartition spectrale et spatiale du signal RF dans un système MIMO utilisant la*



*technique du water-filling.*

La figure 2.07 représente le niveau de puissance transmis par chaque antenne centrée autour de la fréquence porteuse. La répartition de puissance diffère selon l'état du canal mais la puissance totale est toujours la même.

L'expression de la capacité d'un système utilisant l'allocation de puissance ou le water-filling :

$$C = E \left\{ \sum_{k=1}^n \log_2 \left[ 1 + \frac{\gamma}{N} \lambda_k \right] \right\} \quad (2.09)$$

Cette technique a l'avantage d'être optimale au niveau de la capacité. Cependant elle requiert la CSI au niveau du récepteur et de l'émetteur, ce qui peut paraître réaliste dans un environnement intérieur en mode TDD (Time Division Duplex) ou en mode FDD (Frequency Division Duplex) avec retour de voie, mais difficile dans un environnement à grande mobilité à cause de la nécessité d'un canal stationnaire pendant la durée de transmission pour la voie de retour.

### **2.3.3 Systèmes multi-antennes sans CSI ni à l'émission ni à la réception**

Les techniques sans CSI ni à l'émission ni à la réception sont les techniques de codage espace-temps différentiel. Elles peuvent être une solution pour éviter les dégradations dues à l'estimation de canal MIMO mais provoquent une perte intrinsèque de 3dB à cause de la non cohérence. En effet, dans des environnements typiques de communications sans fil, les coefficients du canal sont constants seulement pour des périodes de temps limitées. Pour un nombre important d'antennes d'émission, la quantité d'information introduite par une estimation de canal par des pilotes ou une égalisation par séquences d'apprentissage diminue le débit. Pour contourner ce problème, Marzetta et Hochwald ont étudié le scénario où le récepteur n'utilise pas la CSI. Dans [11], Zheng et Tse ont trouvé la capacité de ce canal pour un rapport signal sur bruit très élevé :

$$C = n \left( 1 - \frac{n}{T} \right) \log_2(\gamma) \quad (2.10)$$

avec  $n = \min(N, M, T/2)$  où  $T$  est le nombre de durées symboles du code durant lequel le canal est constant.

Cette technique d'émission peut être adaptée lorsqu'il est difficile d'effectuer une estimation de canal précise, c'est à dire quand le canal varie temporellement. Cependant, à cause de la perte intrinsèque de 3 dB due à la non-cohérence, les codes espace-temps différentiels obtiennent de moins bonnes performances. Il est intéressant de noter que lorsque le temps de cohérence du code espace-temps différentiel augmente, la capacité varie linéairement avec le minimum du nombre d'antennes d'émission et de réception  $\min(N, M)$ .

La détermination de la capacité théorique d'un canal MIMO permet de formaliser ces performances attendues. Cependant en pratique il est important de trouver le récepteur numérique adéquat permettant d'atteindre l'optimalité en termes de capacité et/ou de diversité. Selon les conditions d'utilisations un récepteur peut être mieux adapté qu'un autre et aussi plus ou moins complexe que l'autre.

## 2.4 Critères d'analyses de performance des systèmes

### 2.4.1 SNR moyen dans un canal à évanouissements rapides

Le SNR est probablement l'indicateur le plus connu pour mesurer les performances d'un système de communication numérique, notamment des systèmes à diversité d'espace à la réception. Le plus souvent évalué en sortie de traitement, il est directement relié au processus de détection des données et constitue donc un bon indice sur la performance court terme d'un lien radio. Le SNR moyen est le moment d'ordre 1 de la variable aléatoire "SNR instantané" qui dépend de la statistique de l'évanouissement du canal et s'écrit simplement :

$$\bar{\gamma}_s = \int_{-\infty}^{+\infty} \gamma_s P_{\gamma_s}(\gamma_s) d\gamma_s \quad (2.11)$$

où  $\gamma_s P_{\gamma_s}(\gamma_s)$  est la densité de probabilité du SNR instantané introduite dans les sections ci-dessus. Cependant en présence d'un canal sélectif, la bonne détection des symboles reçus ne dépend plus uniquement de la valeur de  $\bar{\gamma}_s$ , mais également de l'interférence entre symbole présente au moment de la décision. Ce n'est donc plus un critère suffisant pour caractériser la qualité d'une communication. [10]

### 2.4.2 Probabilité d'erreur moyenne

#### 2.4.2.1 Evanouissements rapides

Il est assez naturel de caractériser les variations court termes par la probabilité d'erreur symbole moyenne ou "Symbol Error Probability" (SEP). C'est indéniablement le critère le plus pertinent, et donc le plus largement étudié pour l'analyse de performances des systèmes de communications numériques. Dans ce manuscrit, le SEP moyen est mathématiquement désigné par  $P_s(E)$  et renvoie à la probabilité que l'événement E "une erreur sur la détection du symbole est commise" soit réalisé. Son expression sans codage, est :

$$P_s(E|\bar{\gamma}_s) = \int_0^{\infty} P_s(E|\gamma_s, \bar{\gamma}_s) P_{\gamma_s}(\gamma_s) d\gamma_s \quad (2.12)$$

C'est aussi le plus délicat à obtenir analytiquement. Cela tient principalement au fait que la probabilité d'erreur conditionnelle  $P_s(E|\gamma_s, \bar{\gamma}_s)$  est généralement une fonction non linéaire du SNR instantané et dont la forme dépend de la modulation et du type de détection effectuée par le récepteur. L'équation (2.12) est l'expression rigoureuse de la probabilité d'erreur symbole. Elle traduit le fait que le SEP moyen dépend du SNR moyen. Notons qu'en présence de shadowing,  $\bar{\gamma}_s$  est une variable aléatoire, donc  $P_s(E|\bar{\gamma}_s)$  également.

#### 2.4.2.2 Evanouissements lents

Lorsque le canal varie très lentement, les deux effets évanouissements et shadowing se superposent et sont considérés comme des phénomènes long termes (au regard de la durée symbole). Comme le canal à court terme est vu comme un canal AWGN (Additive White Gaussian Noise) (et non plus comme un canal à évanouissement), la formulation en (1.12) doit être adaptée. Puisque le SNR est stationnaire à court terme  $\bar{\gamma}_s = \gamma_s = \frac{E_s}{N_0}$ , alors  $P_s\left(E|\frac{E_s}{N_0}\right)$  est la probabilité d'erreur en canal AWGN. Pour les modulations que nous considérons dans ce manuscrit (M-PSK, M-QAM), elle s'exprime en fonction de la fonction gaussienne d'erreur ou une combinaison de celle-ci.

### 2.5 Evaluation de système MIMO-coop

Les systèmes multi-antennes peuvent être exploitées pour réduire la consommation énergétique. Cependant, l'application directe de ces systèmes aux WSNs est difficile étant donnée la taille physique des nœuds qui ne peuvent supporter qu'une seule antenne. Toutefois, le gain des systèmes multi-antenne peuvent encore être atteints par une nouvelle technique dites «MIMO-Coopératif». Cette technique est basée sur le principe de la coopération, en profitant de l'existence de différents nœuds situés dans le réseau afin de transmettre le message d'une source vers une destination précise constituant ainsi un système MIMO-virtuel.

Dans cette section, notre objectif est de montrer qu'effectivement grâce à l'utilisation de «MIMO-Coopératif», selon notre définition, on peut conserver de l'énergie. Pour ce faire, nous devons tout d'abord définir l'architecture proposée ainsi que le scénario suivie pour la construction, ensuite le modèle énergétique adapté à notre architecture.

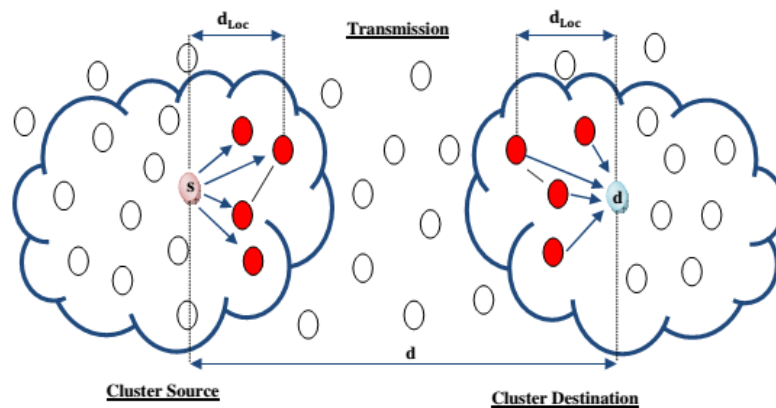
#### 2.5.1 Architecture

Après avoir défini les différents éléments constitutifs de nos travaux, il est temps de les rassembler afin de synthétiser, d'exploiter et d'argumenter notre système MIMO-Coopératif. Dans ce sens,

notre travail se base sur le principe de la coopération entre plusieurs nœuds, avec l'utilisation du principe du système de communication MIMO afin d'exploiter la diversité dans un et profiter ainsi de l'utilisation des différents nœuds pour l'amélioration de la QoS (Qualité de Service).

Une transmission MIMO coopérative utilise principalement les Codes de Blocs d'Espace-Temps (Space Time Block Code (STBC)) vu sa simplicité de codage et de décodage qui est très utile et aussi très pratique aux WSNs. L'application du système MIMO-Coopératif suit un scénario précis afin de garantir un tel niveau de qualité de service.

Le schéma coopératif qu'on représente dans la figure 2.07 est constitué par une source  $s$ , un destinataire  $d$ ,  $N$  relais au niveau de l'émission (Cluster source) et  $M$  relais au niveau de la réception (Cluster destination), munis chacun d'une seule antenne. Ces relais ( $N+M$ ) aident la source à envoyer ses données, sous un schéma équivalent à un système MIMO.



**Figure 2.08 : MIMO Coopératif.**

La communication MIMO-Coopérative se déroule en trois phases qu'on présente comme suit :

*a. Echange local de données (SIMO) :*

Le cluster source va contenir la source et les relais les plus proches d'elle avec une distance  $d_{Loc}$  qui ne dépasse pas un certain  $d_{max} = 5$  m. A la première phase, le nœud source  $s$  doit coopérer avec les  $N$  nœuds et échanger ses données constituant ainsi un système SIMO entre  $s$  et les  $N$  relais.

*b. Transmission Coopérative (MIMO traditionnel) :*

Après la réception des données venues de  $s$  par les  $N$  relais au niveau de cluster source, ces données vont être modulées et codées aux symboles QPSK-STBC et ensuite transmettent simultanément au

M relais de cluster destination en constituant un système MIMO traditionnel (chaque nœud coopératif joue le rôle d'une antenne du système MIMO).

*c. Réception Coopérative (MISO) :*

Au niveau du deuxième cluster destination, les M nœuds voisins du nœud de d vont d'une part recevoir les symboles modules, puis les retransmettre au d pour qu'il les décode par la suite. La distance entre les M nœuds et le nœud destinataire est aussi  $d_{Loc}$ . Une 4<sup>ème</sup> phase supplémentaire est ajoutée après un cas spécial décrit au-dessous.

*d. Clusters intermédiaires :*

Après avoir construit les deux dites clusters et choisie les relais candidats dans les deux côtes, une vérification au niveau de la transmission coopérative est prévue, afin de tester si ces relais candidats, dans les deux clusters, peuvent se communiquer entre eux (un test de la portée). Dans le cas où la vérification est négative, on optera donc pour la construction d'un cluster intermédiaire ou plusieurs selon le besoin.

### **2.5.2 Description du modelé**

#### **2.5.2.1 Modèle**

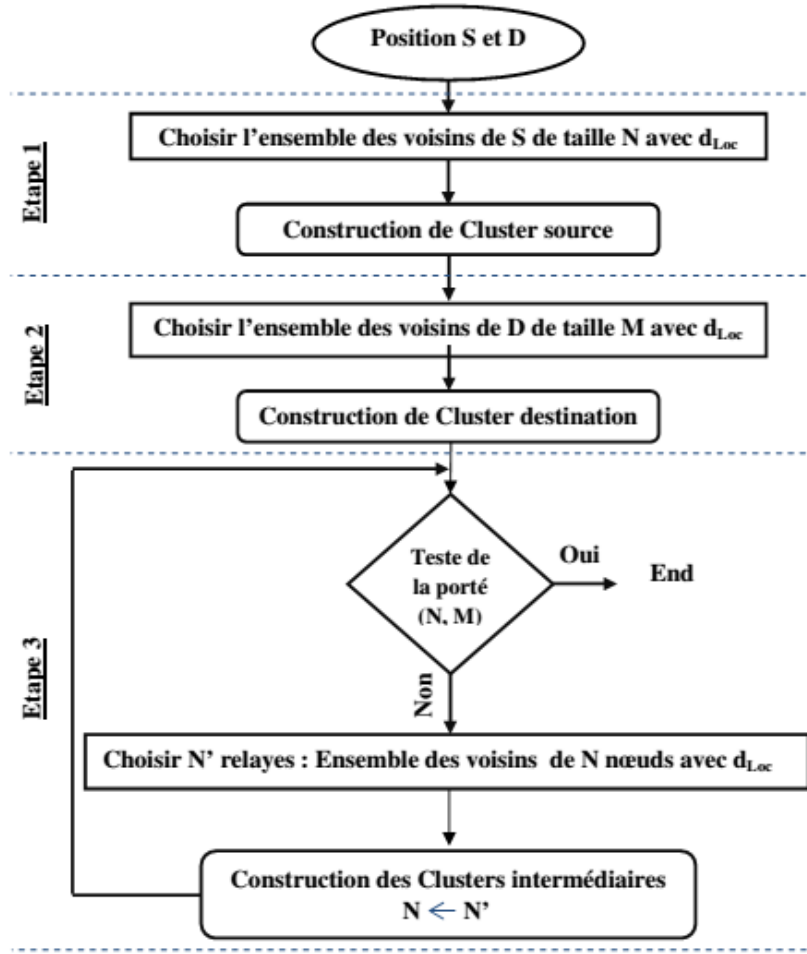
Le modèle MIMO-Coopératif est adaptatif à n'importe qu'elle architecture du réseau.

#### **2.5.2.2 Sélection des relais coopèreront**

La détermination des relais aidant à la coopération est une opération pas du tout triviale, elle nécessite des traitements physiques compliqués. Afin de garantir une meilleure performance, les relais aidant à la coopération doivent être choisis minutieusement. On peut distinguer les différents stratégies de sélection du relais en trois catégories selon le critère choisie à savoir ; la qualité du canal, la distance et la qualité de service requis.

*a. Qualité du canal*

La première méthode se base sur la qualité du canal qui lie le relais à la source et la destination. Les relais sélectionnés doivent disposer les meilleurs canaux vers la source et vers la destination (RTC/CRT).



**Figure 2.09 :** *Scenario de la construction du système MIMO-Coopérative.*

- Global max-min :

On pourra choisir un relais ayant le CSI global maximum parmi les autres nœuds en utilisant la formule :

$$CSI_{GMAX_{S-r-d}} = Argmax[Max(Min(CSI_{S-r}, CSI_{r-d}))] \quad (2.13)$$

ou  $CSI_{S-r}$  et  $CSI_{r-d}$  la qualité des liens respectivement entre la source s et le relais r et la destination d.

- Moyenne harmonique

Il est possible de choisir les relais ayant la meilleure moyenne harmonique des qualités de ses liens en utilisant la formule :

$$CSI_{GH_{S-R-D}} = Argmax \frac{2|CSI_{S-r}|^2 |CSI_{r-d}|^2}{|CSI_{S-r}|^2 + |CSI_{r-d}|^2} \quad (2.14)$$

Ces deux formules garantissent une qualité minimale de la coopération, mais la première stratégie est plus performante et elle garantit un lien plus stable.

En revanche, vu le changement fréquent de l'état du canal, les capteurs n'ont pas une connaissance au préalable du CSI de chaque canal. Un échange de messages doit se faire afin de déterminer les  $CSI_s$  instantanées des différents canaux et choisir les relais convenables.

- Global max

Dans les deux cas précédents, nous supposons une coopération proactive. Dans le cas d'une coopération réactive, le relais est choisi après l'envoi des données par la source, la sélection du relais se fait en utilisant la qualité du canal R – D. Pour cela, la formule utilisée est la suivante :

$$CSI_{GMAX_{r-d}} = Argmax[Max(CSI_{r-d})] \quad (2.15)$$

Chaque relais calcule par la suite un temporisateur inversement proportionnel à  $CSI_{GMAX_{r-d}}$  et entre en compétition avec les autres relais. Celui ayant le temporisateur le plus faible gagne le canal et envoie sa demande d'élection.

#### *b. Distance*

Le nombre des relais qui vont participer à la coopération est déterminé selon la distance entre la source et la destination. On peut utiliser donc une méthode qui se base sur la distance de telle sorte que ces relais soient au bord du cluster et proche du cluster destination, afin de conserver l'énergie, ou l'utilisateur a la possibilité de choisir le nombre de relais utilisés.

#### *c. Qualité de service requise*

Les relais coopérants peuvent être sélectionnés et même quantifiés en termes de qualité de service exigée à la destination. Cette qualité de service dépend entièrement de l'application utilisée. En d'autres termes, on cherche les relais qui satisfont l'exigence d'application.

### **2.5.3 Analyse de la consommation énergétique pour les WSNs**

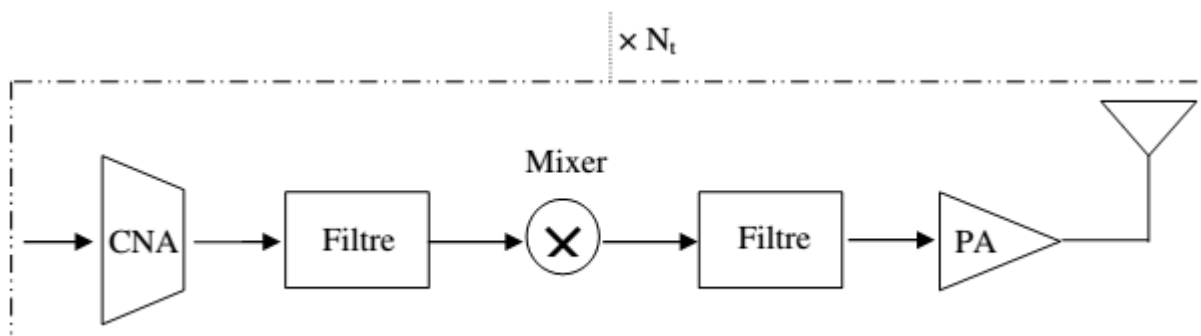
L'énergie nécessaire pour le transfert de données est un facteur clé pour la conception de l'efficacité énergétique dans les WSNs. Pour ce faire, on doit tout d'abord savoir où cette énergie est consommée ce qui est le but justement de cette partie.

Dans cette section, l'efficacité énergétique de la technique MIMO-Coopérative est proposée pour les WSNs. Le critère de la sélection des relais qui vont coopérer, afin de former un MIMO, est en fonction des conditions du canal et de la distance. Nos résultats expérimentaux montrent que le

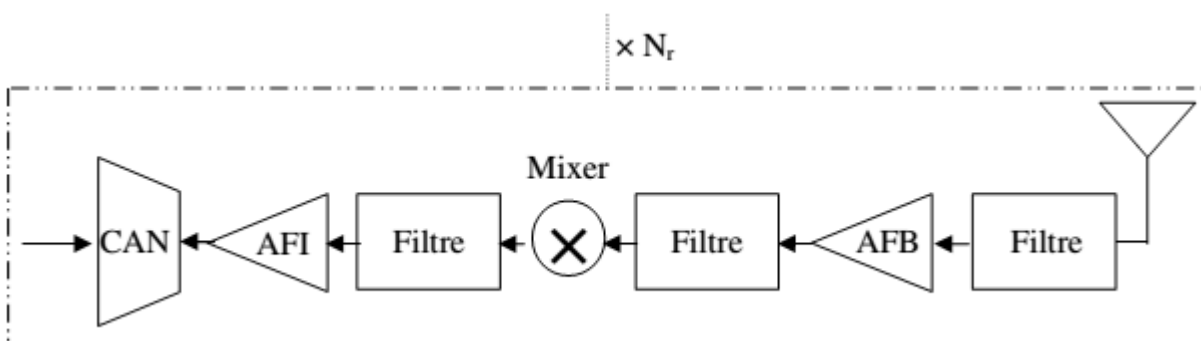
MIMO-Coopérative est le meilleur en termes de consommation de l'énergie totale en se basant sur le travail de Nguyen en 2007. Le modèle de l'énergie est évalué pour différentes combinaisons de tailles de cluster, de nombre de capteurs sélectionnés et constellations utilisées.

### 2.5.3.1 Modèle énergétique

Afin d'évaluer l'énergie consommée pour les MIMO-Coopératifs, on suppose une chaîne de transmission RF :



**Figure 2.10 :** Chaines d'émission RF classique.



**Figure 2.11 :** Chaines de réception RF classique.

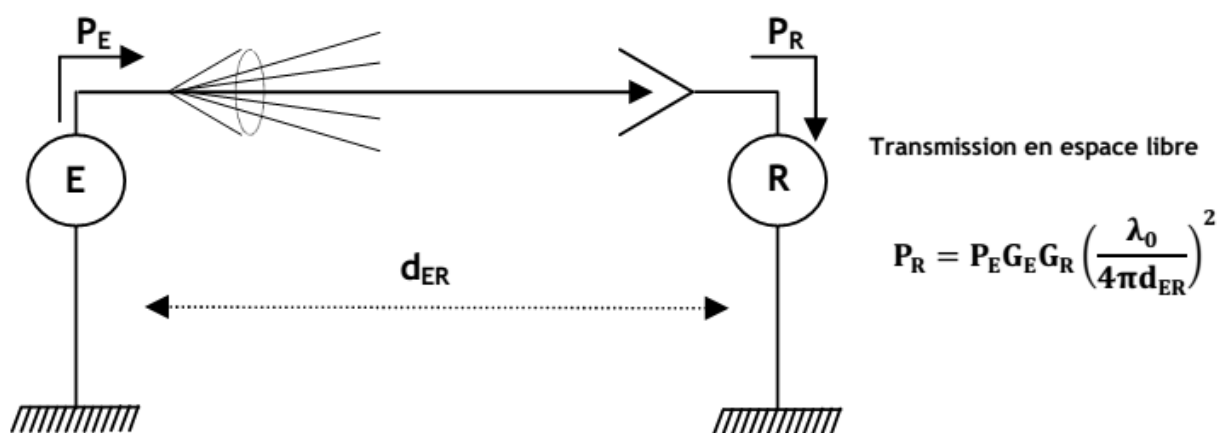
où ses différentes composantes qui le constitue (figure 2.10) sont présentées dans le tableau 2.01.



Blocks	Description	Puissance consommée
CNA	Convertisseur Numérique / Analogique	$P_{CAN}$
Mixer	Mélangeur	$P_{mix}$
AF B	Amplificateur faible bruit	$P_{AFB}$
AF I	Amplificateur de fréquence intermédiaire	$P_{AFI}$
Filtre	Filtres actifs à l'émission	$P_{Filtre -T}$
Filtre	Filtres actifs à la réception	$P_{Filtre -R}$
CAN	Convertisseur Analogique / Numérique	$P_{CAN}$
Syn	Synthétiseur de fréquence	$P_{Syn}$
$N_t$	Nombre d'émetteurs	-
$N_r$	Nombre de récepteurs	-

**Tableau 2.01:** Les composantes du système RF.

On propose des antennes omnidirectionnelles très simples garantissant un bilan de liaison plus favorable afin de gagner en couverture et / ou en qualité de transmission. L'équation du bilan de liaison est donnée par le schéma et la relation associée suivante (figure 2.12) où  $P_E$  est la puissance d'émission,  $G_E$  et  $G_R$  respectivement les gains d'antenne à l'émission et à la réception,  $d_{ER}$  la distance (émetteur / récepteur) et  $\lambda_0$  la longueur d'onde dans le vide.



**Figure 2.12 :** Liaison d'une chaîne Emission / Réception.

### 2.5.3.2 Consommation énergétique (SISO et MIMO)

L'objectif principal de cette partie est de montrer que grâce à l'utilisation de MIMO-Coopératif on peut conserver l'énergie pour une certaine distance. Afin de l'évaluer, on considère le système RF (figure 2.12) qu'on suppose en premier lieu non-coopératif et en second coopératif.

#### d. Système non-coopératif

Afin d'évaluer l'énergie du système MIMO-Coopératif, on commence par le cas simple d'un système SISO et après une extension pour le cas d'un système MIMO. Nous utilisons le modèle de l'énergie présenté précédemment où nous avons omis les blocs de traitement numérique du signal (codage, modulation, détection ...) afin de simplifier les calculs. Tout au long de cette section, nous supposons des canaux à évanouissements de Rayleigh, et parfaitement synchronisés de transmission / réception entre nœuds de capteurs sans fil.

Le signal reçu à temps discret est atténué par une matrice de canal  $H$ . On suppose que chaque élément de  $H$  est une variable aléatoire gaussienne symétrique complexe avec une moyenne nulle et une variance unité.

La consommation totale de la puissance peut être classée en deux grandes parties, à savoir, la consommation de la puissance de transmission de l'amplificateur de puissance  $P_{P_a}$  et la puissance dissipée dans le circuit par tous les blocs RF notée par  $P_c$ .

$$P_t = P_{P_a} + P_c \quad (2.16)$$

$P_{P_a}$  : Puissance de transmission

$P_c$  : Puissance dissipée dans le circuit

#### **Consommation totale de l'énergie**

La consommation totale de l'énergie par bit pour un système non-Coopératif  $E_{bt}$  est formulée comme suit :

$$E_{bt} = \frac{P_{P_a} + P_c}{R_b} \quad (2.17)$$

Où  $R_b$  est le débit binaire réel, mais si on considère que les nœuds n'ont pas d'informations sur l'état du canal (CSI), le  $R_b$  doit s'écrire autrement. Pour intégrer l'énergie supplémentaire consommée par ses symboles de formation (apprentissage), nous supposons que la taille de bloc est égale à  $F$  symboles et dans chaque bloc, nous incluons les  $pN_t$  symboles d'apprentissage. Ces  $p$  symboles sont utilisés pour former chaque paire émetteur / récepteur.

En générale, la consommation de l'énergie totale peut être écrite ainsi :

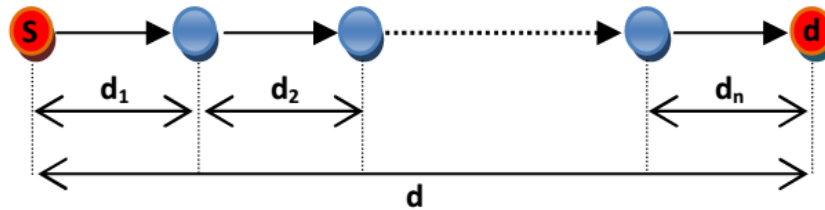
$$E_{totale} = E_{bt} \times N_b \quad (2.18)$$

### e. Système coopératif

Afin d'évaluer l'énergie du système MIMO-Coopératif, on commence par le cas d'un système SISO-Coopératif et puis une extension pour le cas d'un système MIMO.

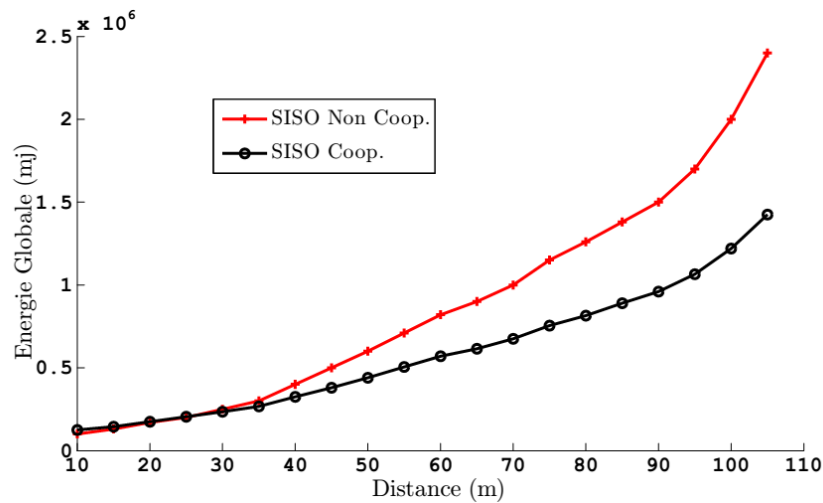
#### SISO-Coopératif :

Afin de réduire la puissance de transmission nécessaire  $P_{Pa}$ , la technique SISO multi-saut peut être utilisée. Si le canal de transmission est divisé en fonction d'une transmission à multi-sauts, la consommation totale de transmission est la somme de chaque transmission de saut unique. Cette consommation augmente par conséquent de façon linéaire avec la distance de transmission.



**Figure 2.13 :** *SISO Coopératif.*

Dans le meilleur des cas où tous les nœuds sont alignés et où la distance entre les deux nœuds  $r_1$  et  $r_2$  est  $d_2$ . La consommation énergétique totale est la somme de la consommation de l'énergie de chaque saut.



**Figure 2.14 :** *SISO Non- Coopératif vs SISO Coopérative.*

La courbe de consommation de l'énergie du système SISO, on note  $d_i^* = 25 \text{ m}$ . La figure 2.14 présente une comparaison entre la consommation énergétique du système SISO Non-Coopératif et SISO-Coopératif avec les paramètres cités au tableau 2.02. On peut constater qu'avec une distance

de transmission  $d = 100$  m, la technique SISO-Coopératif peut économiser 61% de la consommation totale de l'énergie en la comparant avec système SISO Non-Coopératif et par conséquent on peut augmenter considérablement la durée de vie du réseau.

Paramètres	Description
$10^7$	La quantité des bits à transmettre ( $N_b$ )
$10^{-3}$	Taux d'Erreur Binaire (TEB)
5m	La distance locale entre les nœuds coopérants ( $d_{loc}$ )
10	Nombre de Bit/Symbole-STBC ( $N_{sb}$ )
2	Facteur d'atténuation ( $k$ )

**Tableau 2.02:** Les paramètres de simulation pour SISO Non-Coopératif vs SISO Coopératif.

### **MIMO-Coopératif :**

On suppose qu'on souhaite transmettre  $N_b$  bits à partir d'un nœud source  $s$  vers une destination nœud  $d$  (séparés par une distance  $d$ ) où  $N$  nœuds coopèrent au cluster source et  $M$  nœuds coopèrent au cluster destination.

– Cluster Source : Du côté de la transmission, le nœud "s" doit tout d'abord diffuser ses  $N_b$  bits au  $N$  relais qui sont éloignés de la source par  $d_i^*$  au maximum. Le choix de ces relais est aussi en fonction de la configuration géographique du réseau. On considère un canal à AWGN avec un chemin de facteur de perte  $k = \{2, 3\}$ . On suppose qu'il existe une transmission SISO à un seul saut entre deux nœuds coopératifs ou une modulation 16QAM est utilisée (16QAM permet de diminuer la consommation de circuit en réduisant le temps de transmission).

– Inter Cluster (MIMO traditionnel) : Après la réception de bits  $N_b$ , les  $N$  nœuds modulent et codent l'information aux symboles QPSK-STBC puis transmettent simultanément au nœud de destination (ou multi-destination nœuds) sur canal de Rayleigh MIMO décoloration.

– Cluster Destination : Au niveau du côté réception, tout d'abord les  $M$  nœuds coopératifs reçoivent les symboles codés MIMO, quantifient chaque symbole STBC aux  $N_{sb}$  bits puis retransmettent ces bits quantifiés, respectivement, au nœud de destination  $d$  en utilisant le décodage de modulation 16QAM.

Finalement, la consommation de l'énergie globale du système MIMO-Coopératif est :

$$E_{Totale}^{MIMO-Coop} = E_{pa} + E_c + E_{Rx}^{Coop} + E_{Tx}^{Coop} \quad (2.19)$$

## 2.6 Conclusion

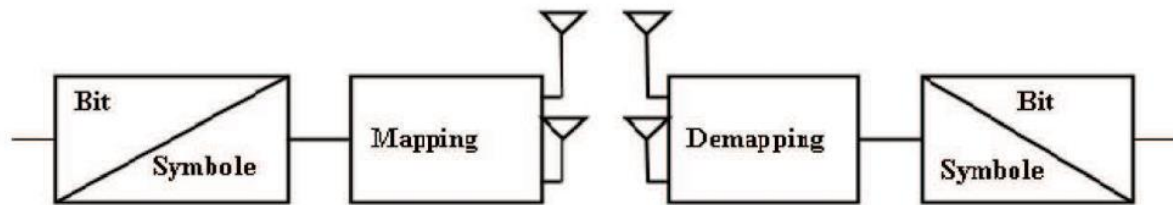
Dans ce chapitre, une étude analytique est faite pour MIMO-Coopératif. Les systèmes MIMO-Coopératifs peuvent exploiter l'efficacité énergétique de la transmission MIMO dans des réseaux de capteurs sans fil. Nous avons montré que les systèmes MISO et MIMO coopératifs sont plus économes en termes d'énergie que les systèmes SISO (Coop. / Non-Coop.) et SIMO pour les transmissions à moyenne et longue distance dans le cas des WSNs. Une sélection optimale concernant la configuration des antennes  $[N, M]$  est effectuée selon la distance de transmission.

## CHAPITRE 3

### LE CODAGE SPATIO-TEMPOREL

#### 3.1 Introduction

Lors de l'implémentation des systèmes à antennes multiples, afin d'améliorer l'efficacité spectrale, plusieurs approches ont été définies. L'approche la plus classique, consiste à utiliser des antennes multiples à la réception et d'appliquer un combineur par ratio maximal (Maximum Ratio Combiner, MRC) aux signaux reçus. Mais, cette approche présente l'inconvénient d'augmenter la complexité du récepteur. Dans le cas où l'émetteur est muni de  $M$  antennes, nous devons définir quel est le traitement « Mapping » nécessaire à appliquer aux signaux avant de les émettre. Ce traitement supplémentaire de part et d'autre du canal radio, mappeur/demappeur, est spécifique aux systèmes MISO et plus généralement MIMO en comparaison au système SISO comme le montre la figure 3.01, est appelé codage spatio-temporel ou codage espace- temps. Ces codes permettent d'introduire de la corrélation spatiale et temporelle entre les signaux émis d'une manière intelligente, afin qu'à la réception le signal reçu soit bien décodé.



**Figure 3.01 :** Introduction du mappeur/demappeur spatio-temporel à l'émission/réception

#### 3.2 Construction des codes

Les critères de performances qui sont généralement considérés dans la construction du codage spatio-temporel et le décodeur qui lui est associé afin d'optimiser les performances de la liaison radio sont:

##### 3.2.1 L'efficacité spectrale :

Elle s'exprime en bit par seconde par Hertz. Notre but est d'avoir le débit le plus élevé possible. La limite ultime, où le débit est le plus élevé possible est définie par la capacité.

### 3.2.2 La robustesse :

La robustesse de la transmission est inversement proportionnelle au débit ; on peut la mesurer par le taux d'erreur binaire (BER).

Lors de l'étude des codes spatio-temporels, le BER est le critère de performance considéré. C'est pour cela que l'un des objectifs principaux de tels codes est d'engendrer un maximum de diversité.

### 3.2.3 La complexité :

Il est possible dans les réseaux sans fil qu'une des deux extrémités de la liaison radio soit alimentée par une batterie. Donc, il est important que la complexité du mappeur/demappeur spatio-temporel soit la plus faible possible. Ainsi, il faut avoir une conception des schémas de transmission et de réception de telle façon que la complexité soit non symétrique, avec la complexité la plus faible du côté alimenter par des batteries (le terminal mobile dans le cas des réseaux cellulaires).

Ces critères de performances sont contradictoires. D'où vient le rôle des ingénieurs qui doivent faire des choix et des compromis selon les applications (mobilité, conditions du canal physique, etc.) et les aspects économiques afin d'avoir des systèmes de transmission MIMO les mieux optimisés. Les techniques de codage spatio-temporel peuvent être classées en deux catégories : les codes Spatio-Temporels en Treillis (STT) et les codes Spatio-Temporels en Bloc (STB).

### 3.2.4 Hypothèses

- Le canal spatio-temporel est composé de  $N \times M$  sous canaux variant temporellement lentement.
- Chaque sous canal est un canal de Rayleigh.
- Les évanouissements des sous canaux sont indépendants.
- Les coefficients du canal sont parfaitement estimés.

### 3.2.5 Critères de construction des codes espace-temps

Différents types de critères sont utilisés pour évaluer et construire les codes espace-temps.

Nous allons introduire ces critères dans ce paragraphe. Nous reprenons ici le modèle du système MIMO utilisant  $N_t$  antennes émettrices et  $N_r$  antennes réceptrices. Nous considérons  $K$  symboles  $s$  à transmettre :

$S = [S_1 \dots S_N]$  Ces symboles sont encodés par un code MIMO qui associe à  $s$  la matrice code  $C$  de dimension  $N_t \times T$  :

$$C = \begin{pmatrix} c_{11} & \cdots & c_{1T} \\ \vdots & \ddots & \vdots \\ c_{N_t 1} & \cdots & c_{N_t T} \end{pmatrix} \quad (3.01)$$

Le rendement du code MIMO est égal à  $R = K/T$ . Lorsque ces symboles codés sont transmis sur un canal non sélectif, nous obtenons les signaux reçus  $Y$  :

$$Y = HC + B \quad (3.02)$$

Avec  $B$  le bruit additif de dimension  $N_r \times T$ .

La probabilité, pour une réalisation de canal  $H$  fixée, que le récepteur décode le bloc  $C'$  alors que le bloc  $C$  a été transmis est appelée probabilité d'erreurs par paire et est Notée :  $P[C \rightarrow C'(H)]$

On appelle  $D$  la matrice représentant la différence entre les blocs  $C$  et  $C'$  :  $D = C - C'$ . Soit  $A$  la matrice Hermitienne telle que  $A = DD^H = V^H \Gamma V$  où  $V$  est une matrice unitaire et  $\Gamma = \text{diag}[\lambda_1 \dots \dots \lambda_N]$  une matrice diagonale.

La probabilité  $P[C \rightarrow C'(H)]$  est alors bornée de la manière suivante :

$$P[C \rightarrow C'|H] \leq e^{-\frac{E_s}{4N_0} d^2(c, c')} \quad (3.03)$$

Où  $\frac{E_s}{N_0}$  représente le rapport signal à bruit par antenne et

$$d^2(c, c') = \sum_{n=1}^{N_r} h_n D D^H h_n^H \quad (3.04)$$

$$= \sum_{n=1}^{N_r} \sum_{m=1}^{N_t} \lambda_n |\beta_{mn}|^2 \quad (3.05)$$

Avec  $h_m$  la  $m^e$  colonne de la matrice  $H$ ,  $v_n$  la  $n^e$  colonne de la matrice  $V$ ,  $\beta_{mn} = h_m v_n$ . La borne supérieure de la probabilité par paire  $P[C \rightarrow C']$  est obtenue par l'espérance de la probabilité  $P[C \rightarrow C'(H)]$ .

$$[C \rightarrow C'] \leq \left( \frac{E_s}{N_0} \right)^{-r_d N_r} \left( \prod_{k=1}^{r_d} \lambda_k \right) \quad (3.06)$$

Où  $r_d$  est le rang de la matrice  $A$  et  $\lambda_k$  sont les coefficients non nuls de la matrice  $\Gamma$ .

Les codes espace-temps sont construits de manière à minimiser la probabilité d'erreur par paire  $P[C \rightarrow C']$  pour toutes les paires possibles [12,13]. Deux critères sont alors définis :

- **Le critère du rang** : la matrice  $A$  doit avoir un rang  $r_d$  le plus élevé possible afin de maximiser le gain de diversité égal à  $r_d N_r$ . Lorsque  $A$  est de rang plein, la diversité est maximale et vaut  $N_t N_r$ .
- **Le critère du déterminant** : le terme  $\prod_{k=1}^{r_d} \lambda_k$  est appelé gain de codage. Celui-ci doit être maximisé pour l'ensemble de toutes les paires de matrices codes.

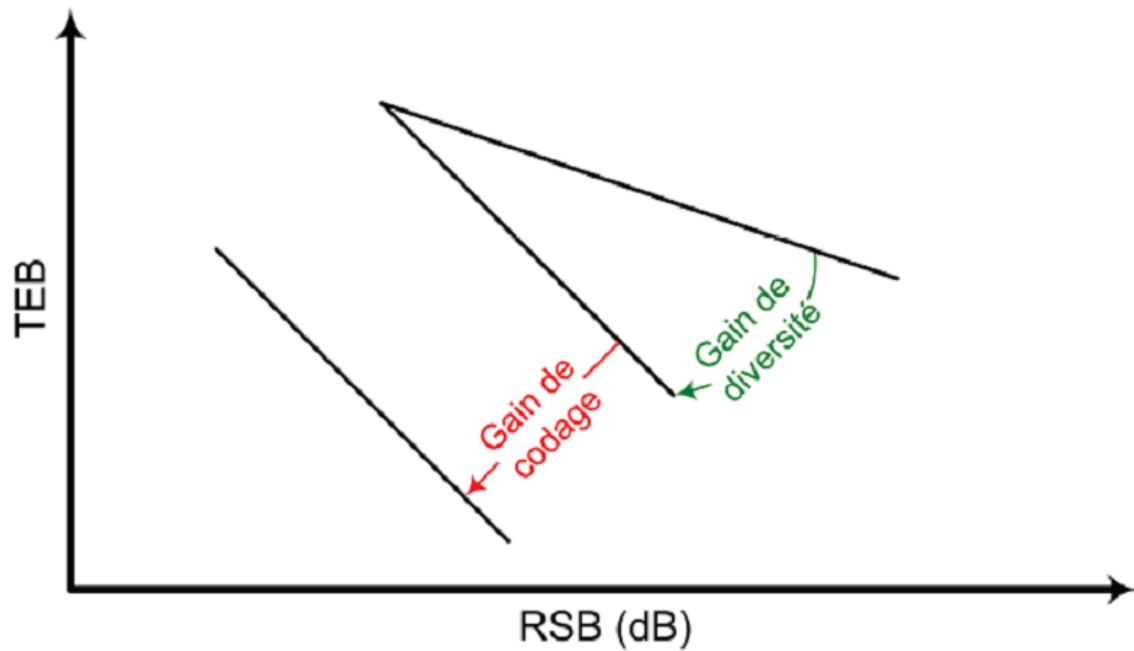
Nous avons donc vu que les performances d'un code espace-temps sont mesurées à l'aide de trois critères :

- **Le gain de diversité** noté  $G_d$
- **Le rendement du code** noté  $R$
- **Le gain de codage** noté  $G_c$



La probabilité d'erreurs moyenne d'une transmission de données dépend du gain de codage, du gain de diversité et du rapport signal à bruit de la manière suivante :

$$P_e \propto G_c(SNR)^{-G_d} \quad (3.07)$$

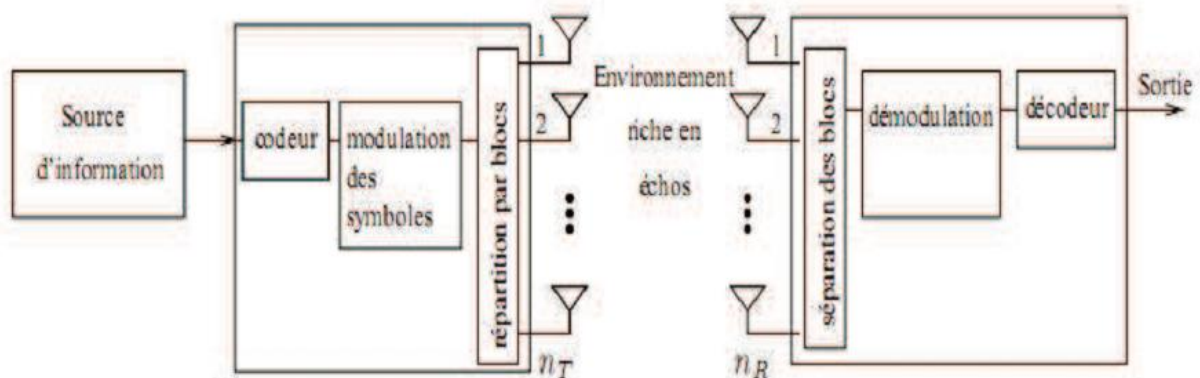


**Figure 3.02 :** Effets du gain de diversité et de codage sur la courbe de probabilité d'erreur

### 3.2.6 Architecture générale d'un système de codage spatio-temporel

L'architecture générale du codage spatio-temporel (CST) est représentée sur la figure 3.03.

Le codage ajoute de la redondance aux données binaires émises afin d'augmenter la diversité spatiale et éviter les évanouissements propre au canal radio-mobile.



**Figure 3.03 :** Schéma du codage spatio-temporel

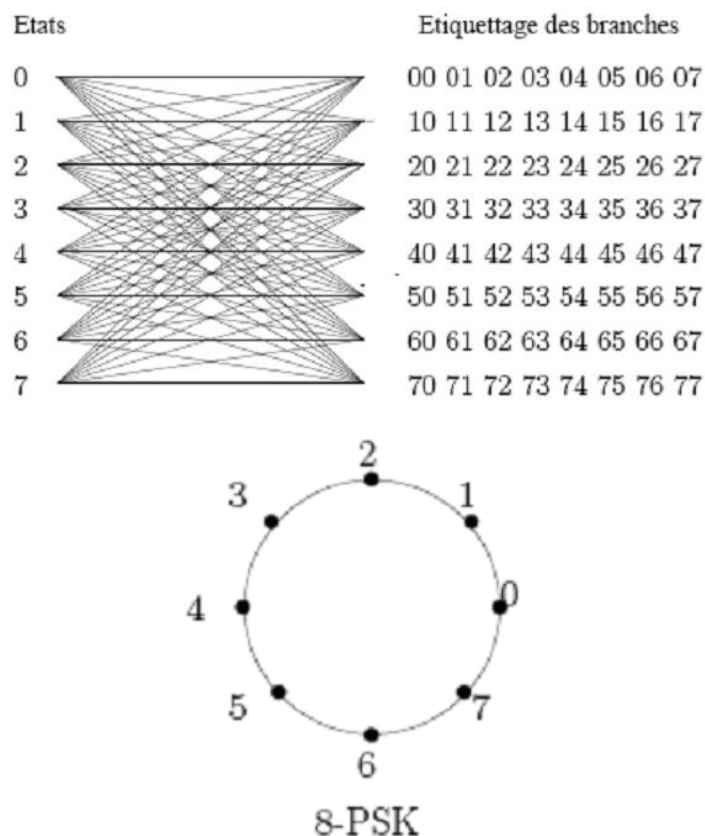
### 3.2.7 Différentes Classes des codes spatio-temporels

Une multitude de techniques de codage ont vu le jour dans le but d'une meilleure utilisation de la diversité de transmission. Ces techniques de codages spatio-temporels peuvent être classées en deux catégories : les STT et les STB.

#### 3.2.7.1 Codes ST en Treillis

Les premiers codes ST en Treillis ont été construits par Tarokh et al. Dans [14]. C'est une généralisation des codes en Treillis classiques (canal Gaussien) pour les systèmes MIMO.

Dans un code ST en Treillis les symboles à transmettre à chaque instant, par toutes les antennes émettrices, représentent une transition dans le Treillis. Au début du codage d'une trame le codeur doit être à l'état 0. En fonction de l'état du codeur et des bits entrants, une transition est choisie à chaque instant  $t$ . Si l'étiquette de cette branche est  $s_1^t, s_2^t, \dots, s_n^t$  cela veut dire qu'à l'instant  $t$ , l'antenne  $j$  va transmettre le symbole  $s_j^t$ . La figure 3.04 illustre un exemple de code en Treillis, pour  $n_t = 2$ , utilisant une modulation 8-PSK.



**Figure 3.04 :** Exemple de code ST en Treillis à 8 états

Soit la trame  $\{0, 1, 5, 7, 6, 4\}$  de symboles 8-PSK à transmettre, le mot de code s'écrit alors :

$$X = \begin{bmatrix} 0 & 0 & 1 & 5 & 7 & 6 \\ 0 & 1 & 5 & 7 & 6 & 4 \end{bmatrix} \quad (3.07)$$

Le décodage d'un code en Treillis se fait par l'algorithme de Viterbi. Cet algorithme minimise une métrique additive sur tous les chemins dans le Treillis. La complexité de cet algorithme est exponentielle en fonction du nombre d'états du codeur, ce qui rend les codes en Treillis peu pratiques.

Afin de construire des codes ST en Treillis optimaux, les auteurs proposent de vérifier les critères du rang, du déterminant et de la trace. La construction proposée des codes ST en Treillis optimaux consiste à choisir un code en Treillis ayant une distance euclidienne optimale et un multiplexage spatio-temporel des symboles d'information maximisant le gain de codage (afin d'assurer une diversité maximale).

### 3.2.7.2 Codes ST en blocs

Plusieurs constructions de codes ST en blocs existent dans la littérature, dont quelques-unes sont des généralisations d'autres.

Le premier code ST, le fameux code d'Alamouti [15], a trouvé un grand succès grâce à ses propriétés : rendement 1 symbole/uc, diversité pleine et capacité maximale atteinte, pour  $n_t = 2$  et  $n_r = 1$ . Un autre avantage du code d'Alamouti est son décodage linéaire qui est une conséquence de sa structure orthogonale (les colonnes de la matrice mot de code sont orthogonales). Toutes les bonnes propriétés du code d'Alamouti ont été un facteur motivant pour sa généralisation pour des dimensions plus élevées. Malheureusement, ces constructions ont été pénalisées par leurs rendements strictement inférieurs à 1 symbole/uc. En relâchant la contrainte d'orthogonalité, des codes ST de rendement compris entre 1 et  $N_t$  ont été construits.

La famille des codes ST en couches (Layered Space-Time codes - LST), ont un rendement qui augmente linéairement en fonction du nombre d'antennes, pour des systèmes symétriques ( $n_t = n_r$ ). Un code LST est entièrement défini par le choix du nombre de couches et du codage associé à chacune des couches. Il existe trois types de codes LST : le codage D-BLAST (Diagonal - Bell Labs Advanced Space Time), sa version simplifiée V-BLAST (Vertical-Bell Labs Advanced Space Time) et le code ST "Wrapped".

Une autre famille de codes ST en blocs existe qui est la famille des codes à dispersion linéaire (Linear Dispersion code - LD). Un code LD est entièrement défini par le choix du nombre de sous-trames et des matrices de dispersions.

Dans la suite nous détaillerons toutes les constructions de code ST citées, en donnant la structure, le rendement et l'ordre de diversité pour chacun des codes.

### 3.2.7.3 Codes ST orthogonaux

Les codes ST orthogonaux sont des codes ST en blocs tels que les mots de code sont des matrices orthogonales ou unitaires [16]. Cette propriété permet un décodage linéaire de ces codes, c'est pourquoi elle est considérée comme leur point fort. Leur faiblesse est un rendement qui s'écroule lorsque le nombre d'antennes est grand. Le code ST orthogonal ayant le plus grand rendement est le code d'Alamouti.

### 3.2.7.4 Code d'Alamouti

S. Alamouti a construit un code ST remarquable, qui atteint la diversité maximale et a un rendement égal à 1 symbole/uc. Ce code est optimal pour deux antennes à l'émission et une antenne à la réception,  $n_t = 2$ ,  $n_r = 1$ . Un mot de code s'écrit :

$$X = \begin{bmatrix} S_1 & -S_2^* \\ S_2 & S_1^* \end{bmatrix} \quad (3.08)$$

Avec  $S_1$  et  $S_2$  deux symboles d'information. Le vecteur reçu s'écrit :

$$\begin{bmatrix} y_1 & y_2 \end{bmatrix} = \begin{bmatrix} h_1 & h_2 \end{bmatrix} \cdot \begin{bmatrix} S_1 & -S_2^* \\ S_2 & S_1^* \end{bmatrix} + \begin{bmatrix} w_1 & w_2 \end{bmatrix} \quad (3.09)$$

Les signaux reçus aux instants  $t$  et  $t + T$  sont donc

$$\begin{cases} y_1 = h_1 s_1 + h_2 s_2 + w_1 \\ y_2 = h_1 s_2^* + h_2 s_1^* + w_2 \end{cases} \quad (3.10)$$

Soient

$$z = \begin{bmatrix} y_1 & y_2^* \end{bmatrix}^t, \text{ et } T = \begin{bmatrix} h_1 & h_2 \\ h_2^* & -h_1^* \end{bmatrix} \quad (3.11)$$

On a alors

$$z = \begin{bmatrix} h_1 & h_2 \\ h_2^* & -h_1^* \end{bmatrix} \cdot \begin{bmatrix} S_1 \\ S_2 \end{bmatrix} + \begin{bmatrix} w_1 \\ w_2^* \end{bmatrix} \quad (3.12)$$

$$T^H \cdot z = (|h_1|^2 + |h_2|^2) \begin{bmatrix} S_1 \\ S_2 \end{bmatrix} + \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} \quad (3.13)$$

Étant donné que  $h_1$  et  $h_2$  sont décorrélés alors  $v_1$  et  $v_2$  le sont aussi. Une simple détection à seuil permet de retrouver les symboles  $S_1$  et  $S_2$ . Le détecteur ML se réduit donc à une détection à seuil. La diversité maximale est atteinte, et est égale à 2.

### a. Généralisation du code d'Alamouti

La généralisation du code d'Alamouti a été faite par Tarokh et al. dans [16] et Tirkkonen et Hottinen dans [17].

Dans [16], il a été démontré que, pour le cas réel, il existe des constructions orthogonales de code ST seulement pour  $n_t = 2, 4$  et  $8$ . Ce problème d'existence de matrice orthogonales réelles est connu en mathématique comme le problème de "Hurwitz-Radon". Par exemple le code pour  $n_t = T = 4$  peut être trouvé à partir du code d'Alamouti en passant à la représentation réelle des quaternions de Hamilton. En posant :

$$i = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -1 & 0 \end{bmatrix} \quad j = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \end{bmatrix} \quad (3.14)$$

XR s'écrit :

$$X_R = a + bi + cj + dk = \begin{bmatrix} a & -b & c & -d \\ b & a & d & c \\ -c & -d & a & b \\ d & -c & -b & a \end{bmatrix} \quad (3.15)$$

Ce dernier résultat a été généralisé aux systèmes dont les symboles appartiennent à des constellations complexes. Considérant un système ayant  $n_t$  antennes à l'émission et un nombre quelconque d'antennes à la réception. Durant  $T$  temps symbole, le système va transmettre  $K$  symboles d'information si. Un mot de code sera donc une matrice  $n_t \times T$  dont les éléments sont des combinaisons linéaires des symboles d'information. Un mot de code orthogonal  $C$  vérifie la relation suivante :

$$C^H C = (\sum_{i=1}^K |s_i|^2) I_{n_t} \quad (3.16)$$

Le code d'Alamouti est l'unique code de rendement 1 symbole/uc, de rang plein sur un ensemble fini de  $C$ .

Avec  $n_r = 1$ , le code d'Alamouti atteint la capacité maximale d'un système MIMO. Mais pour  $n_r > 1$ , le code n'exploite plus tous les degrés de liberté du système, il a une capacité équivalente à un système MIMO  $n_t = 2n_r$  et  $n_r = 1$ . Le code d'Alamouti n'est optimal que pour  $n_t = 2$  et  $n_r = 1$ .

Avec  $I_{n_t}$  la matrice identité de dimension  $n_t$ . La relation ci-dessus garantie :

– La saturation du critère du rang : soient  $C_1$  et  $C_2$  deux mots de code,  $(C_1 - C_2)$  est une matrice unitaire de rang  $n_t$ . Ainsi la diversité maximale est atteinte.

- La saturation du critère du déterminant :  $\min_{c_1 c_2 \in C} (\det (C_1 - C_2)) = a$  , avec a une constante, qui est donc maximisée.
- L'existence d'un décodage linéaire ML.

Un mot de code s'écrit sous la forme suivante :

$$c = \sum_{l=1}^k (a_l \beta_{2l-1} + b_l \beta_{2l-1}) \quad (3.17)$$

Avec  $s_l = a_l + ib_l, l = 1, \dots, k$ , et  $\beta_l, l = 0, \dots, 2k-1$ , et des matrices constantes complexes de dimension  $T \times n_t$ . Afin de vérifier la relation (3.16), les matrices  $\beta_l$  doivent vérifier la condition suivante :

$$\beta_l^H \beta_k + \beta_k^H \beta_l = 2\delta_{lk} I_{n_t} \quad (3.18)$$

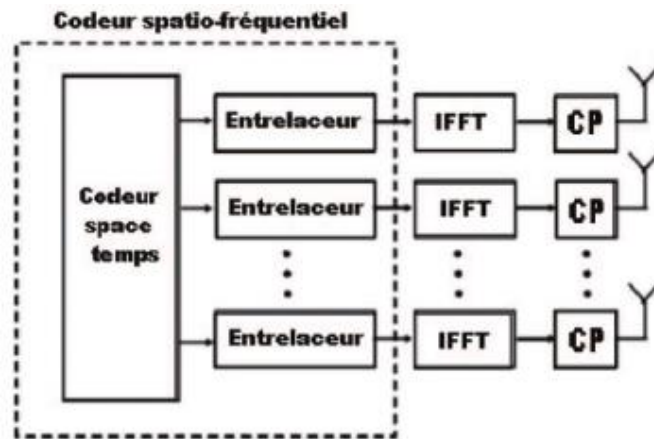
Un code ST C, défini à partir de l'équation II.18 et vérifiant II.19 a une diversité pleine, et un rendement égal à  $K/T$  symboles/uc.

### 3.2.7.5 Codes Spatio Tempo Fréquentiels (STF)

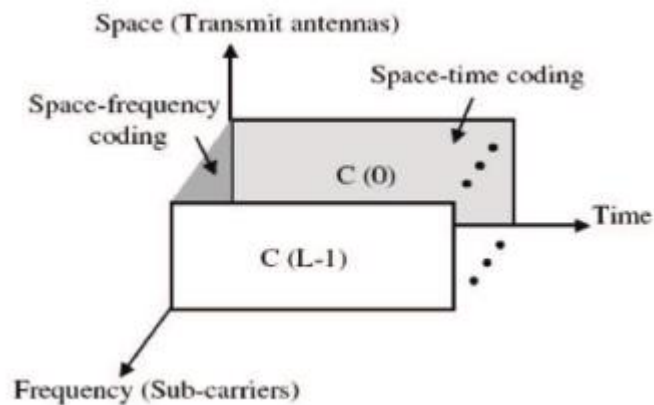
Le codage spatio-temporel fourni un gain de diversité en codant sur les dimensions spatiale, traduit par les antennes multiples, et temporelles. Une transmission de données sur les systèmes MIMO-OFDM (Orthogonal Frequency Division Multiplexing) est possible en appliquant un code spatio-temporel à chaque sous porteuse. Dans de tels systèmes, on ne réalise pas le gain de diversité maximal possible même si les codes spatio-temporels sont utilisés. En effet, la diversité fréquentielle et la corrélation entre différents sous porteuses sont ignorées dans ces systèmes.

$$\hat{s} = \operatorname{argmin} \|y - Hs\|^2 \quad (3.19)$$

Une autre approche pour la transmission sur les canaux MIMO utilisant l'OFDM est de remplacer la dimension temporelle par la dimension fréquentielle. En d'autres mots, différents sous porteuses de l'OFDM peuvent être utilisées pour remplacer la dimension temporelle d'un code spatio-temporel. Ceci résulte sur le codeur spatio-fréquentiel dont la figure 3.05 donne le schéma.



**Figure 3.05 :** Schémas bloc d'un codeur spatio-fréquentiel.



**Figure 3.06 :** Codage spatio-tempo fréquentiel dans l'OFDM

Pour atteindre véritablement le gain de diversité maximal possible, on pourrait coder sur les trois dimensions espace - temps – fréquence comme le montre la figure 3.06. Il est clair que la transmission d'un mot code sur différentes sous porteuses peut procurer une diversité additionnelle fréquentielle. Une approche pour une telle transmission consistera en l'usage d'un code correcteur d'erreur et de l'entrelacement. Le rôle de l'entrelacement est de s'assurer que le codage se fait sur les trois dimensions espace-temps-fréquence. On peut donc utiliser divers codes correcteurs d'erreurs entre autre les codes convolutionnels et les turbo codes.

Ouachani en étudiant les schémas de codage spatio-tempo-fréquentiels basés sur le code d'Alamouti arrive à la conclusion qu'un intérêt de ces codes STF peut résider dans la possibilité de les utiliser dans les réseaux locaux afin d'augmenter les zones de couverture, dans des modes de transmission à bas débit.

### 3.3 Algorithmes de décodage associés aux techniques MIMO

Les paragraphes suivants présentent les différentes techniques de décodage en réception. Le canal  $H$  est supposé avoir été estimé au préalable à l'aide de techniques d'estimation de canal. Dans un premier temps les techniques les moins complexes, c'est à dire les récepteurs linéaires tels que l'égaliseur à forçage à zéro (ZF : Zero Forcing) et à minimisation de l'erreur quadratique moyenne (MMSE : Minimum Mean Square Error), seront décrites. Puis une technique non linéaire appelée élimination successive d'interférences (OSIC : Ordered Successive Interference Cancellations) sera détaillée. Enfin nous aborderons le détecteur optimal basé sur le critère de maximum de vraisemblance (ML : Maximum Likelihood) ainsi que le décodeur par sphère (SD : Sphere Decoder).

#### 3.3.1 L'égaliseur de forçage à zéro (ZF)

L'égaliseur de forçage à zéro est une technique de détection par inversion de matrice. Cette technique consiste à appliquer au vecteur reçu une matrice d'égalisation  $W$ . Le critère ZF a été proposé afin de garantir une élimination de l'IES (Interférence Entre Symbole) en sortie de l'égaliseur. La matrice  $W$  est alors égale au pseudo inverse de la matrice de canal :

$$W = (H^H H)^{-1} H^H \quad (3.20)$$

En présence d'évanouissements et de bruit, le critère ZF a pour effet d'augmenter le niveau de bruit ce qui dégrade les performances.

#### 3.3.2 L'égaliseur minimisant l'erreur quadratique moyenne (MMSE)

L'égaliseur MMSE est basé sur la minimisation de l'erreur quadratique moyenne, donnée par

$$\hat{W} = \arg \min_W E[\|W^H y - s\|^2] \quad (3.21)$$

La matrice d'égalisation optimale est alors :

$$W = H^H \left( \frac{\sigma_b^2}{\sigma_s^2} I_{N_r} + H H^H \right)^{-1} \quad (3.22)$$

Avec

$$E[bb^H] = \sigma_b^2 I_{N_r} \text{ et } E[ss^H] = \sigma_s^2 I_{N_r} \quad (3.23)$$

En présence de bruit, la matrice à inverser est toujours définie positive et donc inversible.

Ce récepteur est donc moins sensible au bruit que l'égaliseur ZF mais ces deux égaliseurs deviennent équivalents en l'absence de bruit.



### 3.3.3 L'égaliseur à annulations successives d'interférences ordonnées

Les égaliseurs ZF et MMSE n'étant pas toujours satisfaisants : le premier est sensible au bruit et le second n'élimine pas toute l'ISI. Ainsi, Golden et Foschini ont proposé dans un algorithme basé sur un critère ZF ou MMSE à retour de décision afin de décoder les codes BLAST.

Le principe de cet algorithme appelé OSIC est le suivant : la contribution du symbole  $s$  qui vient d'être détecté est soustraite du vecteur reçu, ce qui donne un vecteur contenant moins d'interférences. L'ordre dans lequel sont détectés les symboles joue un rôle important sur les performances du système.

Le symbole transmis sur la voie de données ayant la plus forte puissance est décodé en premier. Une fois décodé, sa contribution est annulée sur le vecteur reçu, l'opération est ensuite répétée jusqu'à l'obtention de tous les symboles transmis.

Les notations utilisées dans l'algorithme sont les suivantes :

- $(G_i)_j$  est la  $j$ ème ligne de  $G_i$ .
- les  $k_i$  représentent l'ordre de détection des symboles.
- $Q$  symbolise le processus de quantification.
- $H_{\overline{k_i}}$  est la matrice de canal  $H$  annulée des contributions des premiers émetteurs, c'est à dire que les colonnes  $k_1, \dots, k_i$  de  $H$  ont été remplacées par des zéros.

L'algorithme OSIC est décrit comme suite :

Initialisation
$i = 1$ $G_1 = \mathbf{H}^\dagger$ $k_1 = \arg \min_j \left\  (\mathbf{G}_1)_j \right\ ^2$
Boucle itérative
$\mathbf{W}_{k_i} = (\mathbf{G}_i)_{k_i}$ $r_{k_i} = \mathbf{w}_{k_i}^T \mathbf{r}_i$ $\hat{s}_{k_i} = Q(y_{k_i})$ $\mathbf{y}_{i+1} = \mathbf{y}_i - \hat{s}_{k_i} (\mathbf{H})_{k_i}$ $\mathbf{G}_{i+1} = \left( \mathbf{H}_{\overline{k_i}} \right)^\dagger$ $k_{i+1} = \arg \min_{j \notin \{k_1, \dots, k_i\}} \left\  (\mathbf{G}_{i+1})_j \right\ ^2$ $i = i + 1$

**Figure 3.07 :** Algorithme de détection OSIC

Comme tous les égaliseurs à retour de décision, l'OSIC a l'inconvénient de propager les erreurs. Ainsi de nombreux algorithmes d'amélioration ont été proposés.

### 3.3.4 L'égaliseur à Maximum de Vraisemblance

L'égaliseur ML est optimal en termes de BER. Cette méthode consiste à comparer tous les signaux pouvant être reçus aux signaux réellement reçus, afin de sélectionner le plus vraisemblable d'entre eux :

$$\hat{s} = \arg \min_s \|y - Hs\| \quad (3.24)$$

La complexité de cet algorithme croît exponentiellement avec le nombre d'antennes utilisées en émission et le nombre d'états M de la modulation. En effet, le récepteur doit comparer  $M^{N_t}$  solutions possibles avec le vecteur de signaux reçus.

### 3.3.5 Le décodage par sphère

Le décodeur par sphère est un algorithme qui permet d'approcher les performances du décodeur ML tout en réduisant la complexité. Le principe de cet algorithme est de se placer au niveau du signal reçu y et de chercher le point le plus proche parmi un réseau de points à l'intérieur d'une sphère de rayon  $\sqrt{R}$ . Cela permet de restreindre le domaine de recherche par maximum de vraisemblance et donc la complexité.

En effet, seuls les points du réseau situés à moins de la distance  $\sqrt{R}$  du signal reçu sont considérés. Le choix de la valeur de R est donc crucial.

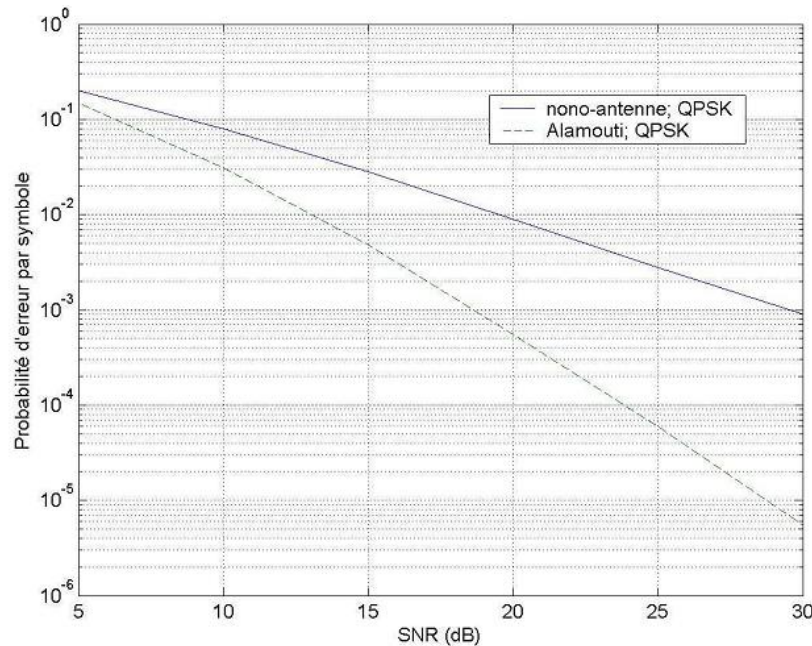
## 3.4 La performance des codages

### 3.4.1 Performance du codage d'Alamouti

La performance du code d'Alamouti, sur un canal de Rayleigh quasi-statique, à évanouissement est illustrée sur la figure 3.08. Le système étudié est constitué d'une seule antenne de réception et utilise une modulation QPSK. On trace sur la figure 3.08, la probabilité d'erreur par symbole, en fonction du SNR. Comme nous pouvons le constater, la performance du code d'Alamouti avec deux antennes d'émission est bien meilleure que celle d'un système à une antenne d'émission. Pour un taux d'erreur par symbole de  $10^{-3}$  le code d'Alamouti offre un gain de 11 dB. Ce qui est plus important de constater encore, c'est que plus le SNR augmente, plus l'écart de performance entre les deux systèmes augmente. En fait le taux d'erreur diminue de façon inversement proportionnel au carré du SNR.

Pour conclure nous dirons que le code d'Alamouti fourni deux propriétés importantes :

- Décodage simple
- Diversité maximale



**Figure 3.08 :** Performance du code d'Alamouti

### 3.4.2 Performance du codage STTC (Space Time Treillis Coding)

#### 3.4.2.1 Améliorations apporté par le STTC

A la sortie d'un codeur en bloc, le code STBC est uniquement fonction des bits en entrée utilisés pour constituer ce bloc. Un code convolutif ou en treillis est différent, parce qu'il dispose d'une mémoire. Cela permet d'avoir en sortie de l'encodeur à un instant  $t$ , un code qui n'est pas uniquement fonction des bits présents aux même instant, en entrée du codeur, car il dépend également des bits utilisés précédemment. Cette dépendance, est liée à la profondeur de la mémoire du treillis.

Pour une constellation qui dispose de  $L = 2^b$  points, représentant  $b$  bits, un STBC code un bloc de  $Kb$  bits pendant  $T$  intervalles de temps. La sortie du codeur, pour les  $T$  intervalles de temps suivants sera uniquement une fonction des  $Kb$  nouveaux bits en entrée. En d'autres termes, comme le processus de codage en bloc n'a pas d'effet mémoire, les  $Kb$  bits en entrée du codeur ne participent qu'une seule fois à ce processus.

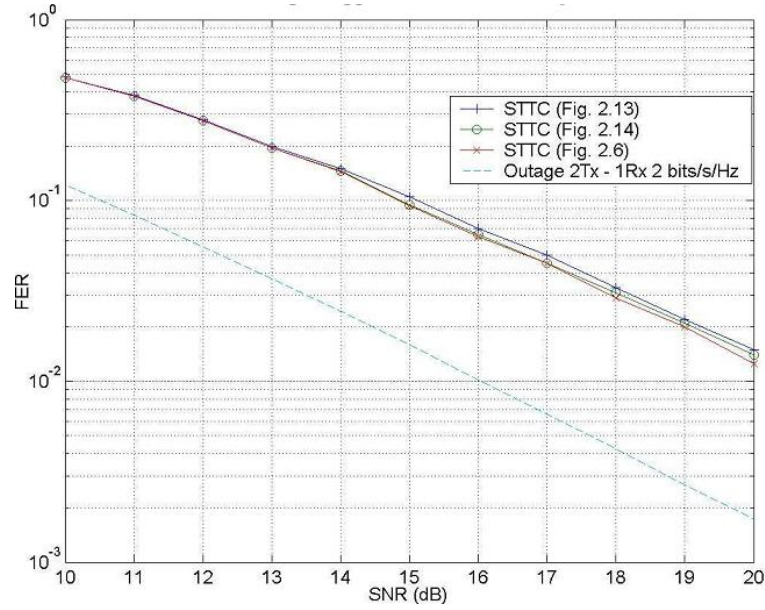
D'un autre côté, dans le cas des STTC, à cause de l'effet mémoire, tous les  $b$  bits en entrée participent au processus de codage  $Q+1$  fois. Une fois en entrée et  $Q$  fois au travers des différents états du treillis. Notons alors que les  $(Q+1)b$  bits en entrée, qui génèrent la sortie du codeur à un instant  $t$ , se chevauchent avec les  $(Q+1)b$  bits qui génèrent les symboles de sortie pendant un certain temps. Il n'y a aucun phénomène de chevauchement en STBC.

#### 3.4.2.2 Performance des codages STTC

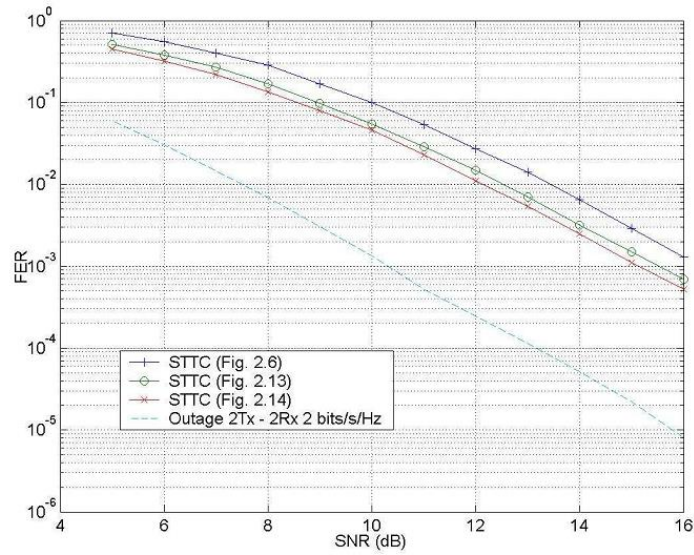
Dans ce paragraphe nous donnons les résultats de simulation des performances des différents exemples de STTC présentés précédemment. Nous supposons un canal de Rayleigh, quasi-statique, à évanouissement lent. Ainsi, les différents coefficients du canal sont des variables aléatoires Gaussiennes complexes, fixes durant l'émission de toute une trame de données. Nous considérons des paquets de longueur 150. Sur les courbes de résultats que nous présentons, figure le taux d'erreur par trame (FER : Frame Error Rate) en fonction du SNR.

La figure 3.09 montre les résultats de simulation, pour un système utilisant deux antennes d'émission et une antenne de réception, et pour les codes représentés par les fig 2.6, 2.13 et 2.14. Tous ces codes ont des treillis constitués de quatre états et émettent 2 bits/s/Hz en utilisant une modulation QPSK. Ces codes STTC présentent les mêmes performances, mais ils ne possèdent pas le même CGD (Cumulative Gain Distance) minimum. En fait, le CGD n'est pas le seul paramètre qui affecte les performances d'un code.

Par contre, lorsque le nombre d'antennes de réception devient égal ou supérieur à deux, le critère du CGD a de l'influence sur les performances des codes. En effet, comme le montre la figure 2.18, c'est le code qui présente le CGD minimum le plus élevé qui a les meilleures performances. Ainsi, comme nous pouvons le voir, les STTC des figures 2.13 et 2.14 apportent une amélioration de 1 dB par rapport au STTC de la figure 2.6.



**Figure 3.09 :** *FER en fonction du SNR, pour des STTC à 4 états, 2 Tx – 1 Rx,  $r = 2$  bits/s/Hz*



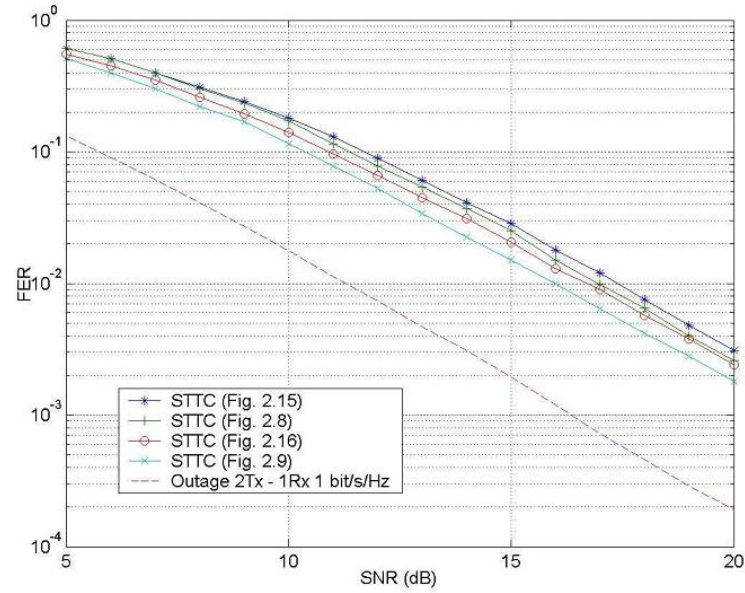
**Figure 3.10 :** *Figure 2.18 – FER en fonction du SNR, pour des STTC à 4 états, 2 Tx – 2 Rx,  $r = 2$  bits/s/Hz*

Nous retrouvons ici, l'argument en [18], qui dit que le paramètre du CGD a une réelle influence uniquement lorsque le nombre d'antennes de réception est supérieur à un.

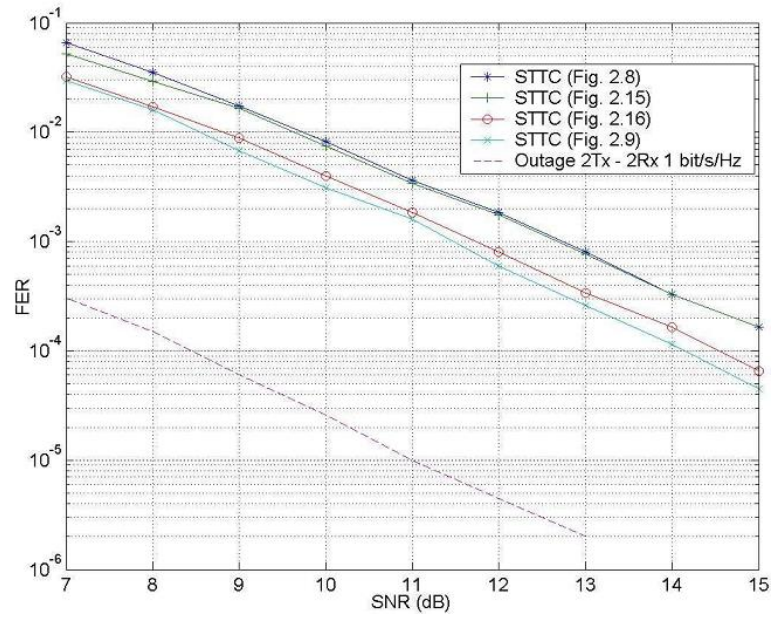
La figure 3.11 montre les résultats d'un système où le débit vaut 1 bit/s/Hz et où le nombre d'antennes de réception est de un. Les performances présentées concernent des treillis de deux et quatre états des fig 2.8, 2.15 et 2.16. Les comportements des codes à 1 bit/s/Hz et ceux des codes à 2 bits/s/Hz sont vraiment similaires. On constate également que lorsque le nombre d'états dans le



treillis augmente, les performances des codes deviennent meilleures, mais en général, il y a un compromis à trouver entre, complexité du décodage (liée au nombre d'états du treillis) et performance du code. Les résultats de la figure 3.12 sont obtenus avec les mêmes paramètres que précédemment à la différence que le nombre d'antennes de réception est cette fois-ci égal à deux. On retrouve alors les mêmes conclusions que pour le cas 2 bits/s/Hz, à savoir que lorsque  $N_r > 1$ , le critère du CGD est une bonne indication pour les performances du code.



**Figure 3.11 :** *FER en fonction du SNR, pour des STTC, 2 Tx – 1 Rx,  $r = 1$  bits/s/Hz*



**Figure 3.12 :** *FER en fonction du SNR, pour des STTC, 2 Tx – 2 Rx,  $r = 1$  bits/s/Hz*

Dans cette partie nous nous sommes intéressés à la construction des codes STTC pour lesquels, si le nombre d'antennes d'émission du système est de  $N_t$ , on doit assigner  $N_t$  symboles de la constellation sur toutes les transitions du treillis. Certaines contraintes existent pour la réalisation de ces codes, comme celle qui stipule que si l'on souhaite concevoir un code de débit  $b$  et de diversité  $r$  le treillis doit disposer impérativement d'un certain nombre d'états minimum. Il est possible de construire un STTC de façon manuelle, par l'intermédiaire de simulations ou alors en se servant d'une matrice génératrice  $G$ . Ces diverses méthodes de conception ont permis de montrer que certaines d'entre elles, pour une même complexité, étaient plus performantes. Le critère que nous avons privilégié ici dans les exemples de STTC synthétisés est basé sur le critère du déterminant et porte le nom de critère du  $\min_{CGD}$ . Il consiste à maximiser la valeur du déterminant du produit de la matrice des différences de code par sa transposée conjuguée pour chaque paire de mots de code émis depuis un état donné du treillis. Il est également possible d'augmenter les performances d'un code en treillis, en augmentant le nombre d'états, mais cela ne va pas sans diminuer la complexité du décodage. Il faut donc trouver pour ce genre de code, un compromis entre la complexité du décodage et les performances souhaitées.

### 3.5 Conclusion

Une transmission MIMO coopérative utilise principalement les STBC vu sa simplicité de décodage et de décodage qui est très utile et aussi très pratique aux WSNs. L'application du système MIMO-Coopératif suit un scénario précis afin de garantir un tel niveau de qualité de service.

## CHAPITRE 4

### LES METHODES DE LOCALISATION DANS UN RESEAU DE CAPTEURS SANS FILS

#### 4.1 Introduction

La localisation peut résulter de mesures des caractéristiques ou des méthodes liées à des hypothèses relatives à l'environnement. Les méthodes fondées sur les premières seront regroupées au sein de la famille de solutions range-based car dépendant de mesures pouvant principalement être converties en distance. Une seconde famille dite range-free utilisera des données telles que la connectivité ou le nombre de sauts.

Selon l'information retenue, un protocole de mesure spécifique doit piloter le matériel en vue de récupérer des échantillons. Nous considérons comme faisant partie du protocole tous les échanges de messages nécessaires à l'exécution de l'algorithme de localisation.

#### 4.2 Protocoles de collecte dans les méthodes *range-based*

Selon la caractéristique du signal à récupérer, le processus de collecte change. Nous présentons dans ce qui suit les deux principales caractéristiques physiques utilisables, la puissance du signal et le temps.

##### 4.2.1 Indicateur de puissance de signal reçu

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

*Mobile centric* : dans cette configuration, le mobile émet périodiquement des messages que les ancres recueillent. Une fois le *RSSI* (Received Signal Strength Indicator) mesuré, les divers échantillons doivent être renvoyés à un serveur central qui les traitera en vue de la localisation. Le protocole de collecte va donc opérer principalement durant la phase d'acheminement des données vers le serveur. Le travail de [19] se situe dans cette lignée.

*Anchor centric* : à l'inverse du cas précédent, le mobile reste à l'écoute du médium pour identifier ses voisins directs et le *RSSI* associé à leurs transmissions [20]. Dépendamment du type d'algorithme utilisé dans la solution, le mobile transfèrera ou non les informations récupérées à un serveur : si l'objectif est de transformer directement les mesures en distances et d'exécuter un



algorithme léger de calcul de position, aucun autre échange n'est nécessaire. Au contraire, quand des moyens plus importants sont nécessaires, le mobile doit faire parvenir au serveur les informations qu'il détient.

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

#### 4.2.2 Temps de vol du signal

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

##### 4.2.2.1 Le mono-signal

Typiquement, le signal de communication est ici mis à contribution. On suppose que les nœuds sont capables de récupérer avec une précision suffisante les instants de départ et d'arrivée des messages. Si le réseau est complètement synchronisé, le TOA (Time Of Arrival) peut être appliqué : ce protocole repose sur l'échange de deux messages : le premier réalisant la mesure et le second le rapatriement des résultats. La figure 4.01 illustre cet échange : les nœuds A et B partageant une horloge commune, l'estampille collectée par B s'exprime dans la même base temporelle que celle obtenue par A. Les deux valeurs peuvent donc être combinées par l'équation (4.01).

$$T_{vol} = T_1 - T_0 \quad (4.01)$$

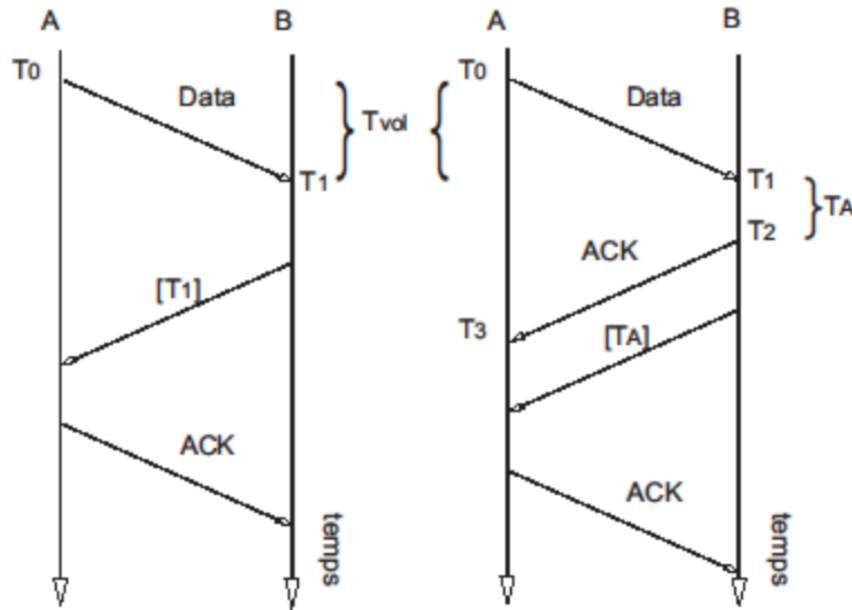
La figure 4.01 présente aussi le protocole TWR [54]. Ce dernier lève la contrainte de synchronisation sur le réseau. Au lieu de combiner des estampilles, la différence de durées est utilisée pour calculer le temps de vol (équation 4.02). Ce type de protocole suppose que la variation de la distance entre les deux partenaires est nulle durant l'exécution de la mesure : ainsi, le temps de propagation peut être considéré comme symétrique. Le travail de [21] présente l'exploitation de ce protocole pour une couche physique 802.11. Il est bon de noter que pour obtenir des résultats stables, 300 échantillons sont nécessaires. [22], quant à lui, décrit l'exploitation d'un signal modulé

par un PN code ou ranging signal pour la mesure du temps de vol. Dans ce cas, il ne s'agit plus d'une paire de trames, données et acquittement, mais d'un code qui, une fois reçu par la cible, est retransmis et corrélé au signal originel pour identifier le temps de vol.

$$T_{vol} = \frac{(T_3 - T_0) - (T_2 - T_1)}{2} \quad (4.02)$$

Les nœuds A et B sont donc chargés de récupérer les instants de départ et d'arrivée des messages. Une seconde trame de données est requise pour le transfert des informations à A. TWR ne résout pourtant pas tous les problèmes. Il peut arriver que les horloges de A et B soient différentes : ceci signifie qu'une microseconde mesurée par A n'a pas la même durée qu'une microseconde mesurée par B. Pour lutter contre ce phénomène, une troisième procédure de collecte a été proposée, le Symmetric Double-Sided/Two-Way Ranging (SDS/TWR). Ce protocole repose sur deux exécutions symétriques de TWR (figure 4.02). Les huit estampilles sont rassemblées après réception de la trame DATA\_REPLY et sont introduites dans l'équation 4.03.

$$T_{vol} = \frac{T_4 - T_1 + T_8 - T_5 - T_3 + T_2 - T_7 - T_6}{8} \quad (4.03)$$



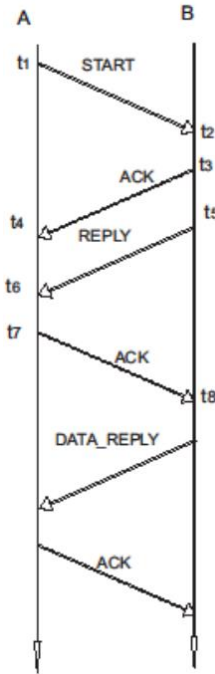
**Figure 4.01 : Time Of Arrival et Two Way Ranging**

Le SDS-TWR réduit l'écart entre la valeur estimée et la vraie valeur du temps de propagation. Si on appelle  $t_{reply1}$  la différence  $t_4 - t_3$ ,  $t_{reply2}$  la différence  $t_2 - t_1$ ,  $\Delta_{reply}$  l'écart entre  $t_{reply2}$  et  $t_{reply1}$ ,  $t_{vol}$  la vraie valeur du temps de vol et  $\hat{t}$  son estimation, la précision passe de l'expression donnée par l'équation 4.04 à celle de l'équation 4.05.

$$\hat{t}_{vol} - t_{vol} \approx \frac{1}{2} \times t_{reply2} \times (e_1 - e_2) \quad (4.04)$$

$$\hat{t}_{vol} - t_{vol} \approx \frac{1}{2} \times \Delta_{reply} \times (e_1 - e_2) \quad (4.05)$$

[64] utilise le protocole SDS-TWR et ajoute au processus de communication une phase de réservation d'un slot temporel dédié à l'exécution du ranging. L'affectation du slot est réalisée par un nœud coordinateur : ce dernier publie la liste des slots disponibles puis, une fois les requêtes des mobiles obtenues, signale à chaque mobile qu'il peut entamer sa mesure de distance. Avant de pouvoir contacter le mobile suivant, le coordinateur doit recevoir une trame de libération provenant du mobile courant. Afin d'assurer le passage à l'échelle, le coordinateur est défini sur une cellule et non sur l'ensemble du réseau. Pendant la mesure d'un mobile, les autres mobiles entrent en mode sleep.



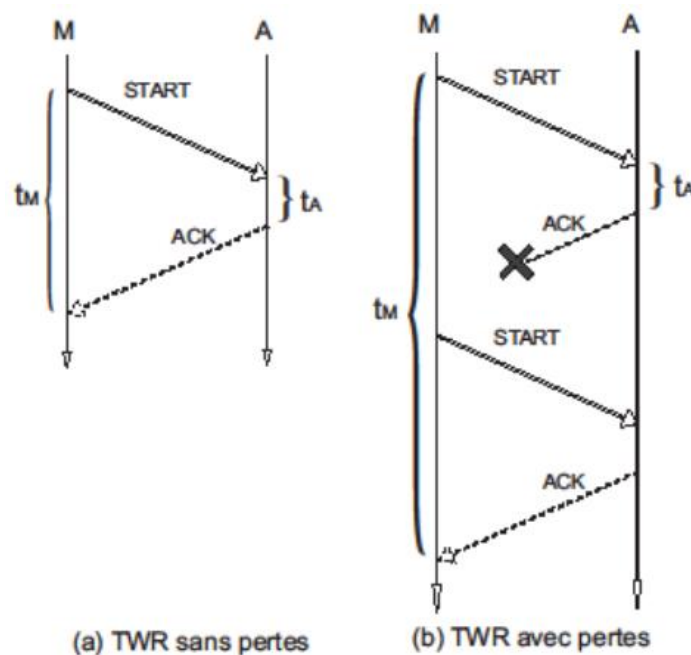
**Figure 4.02 :** *Le protocole Symmetric Double-Sided Two-Way Ranging*

Que l'on utilise TWR ou SDS-TWR, il existe une possibilité de corruption des données liées aux échecs de la communication sur le médium. La figure 4.03 illustre notre propos. Soient M et A les deux nœuds impliqués dans le ranging. Le temps de vol peut être calculé comme la moitié de la différence entre  $t_M$  et  $t_A$ . La mesure de  $t_M$  débute lorsque M soumet sa trame pour émission à la couche physique et prend fin à la réception de l'ACK (Acknowledgement). Dans le cas idéal, A commencerait la mesure de  $t_A$  à la réception de la trame START et arrêterait le processus à l'émission de l'ACK. Dans la réalité, la capture de l'instant d'émission étant très rarement

supportée, il est critique de déléguer la génération et l'émission de l'ACK à la couche PHY de manière à assurer un  $t_A$  constant. D'autre part, impliquer les couches supérieures dans la génération de l'ACK introduit un autre problème. Le calcul final devra tenir compte du temps de traitement associé. Certes, ce dernier peut être mesuré grâce aux horloges du microcontrôleur mais il s'agira d'une estimation grossière, affectée par la granularité des instruments.

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

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



**Figure 4.03 :** Corruption de la mesure dans TWR

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

par le mobile puis assure la mise en commun des différents échantillons au niveau de l'ancre de référence.

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

#### 4.2.2.2 La combinaison de signaux

Les capacités d'un nœud communicant peuvent être étendues à l'aide de divers modules. Certains de ces sous-systèmes sont capables de supporter la communication sans fil. Si la vitesse de propagation de ce signal additionnel est distincte de celle du signal radio, il est possible de calculer la distance en se basant uniquement sur la différence entre les instants d'arrivées. Connaissant la différence entre les instants d'émission  $t_0$ , la relation entre la distance et la différence des instants  $\tau$  est donnée par l'équation 4.06.

$$distance = (\tau - e_2) * \frac{(v_1 * v_2)}{(v_1 - v_2)} \quad (4.06)$$

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

### 4.3 Protocoles de collecte pour les méthodes range-free

Les méthodes range-free se dispensent de mesurer la distance séparant le mobile des balises. Elles exploitent des hypothèses sur la connectivité en vue d'estimer les écarts avant d'appliquer la trilatération ou une méthode géométrique équivalente. La famille DV-Hop (DV-Distance, CDV-Hop, Selective 3-Anchor DV-Hop etc) se base sur l'inondation pour collecter les informations concernant le réseau comme l'adresse des balises et le nombre de sauts séparant le mobile de chacune des balises. Ce genre de protocole génère un nombre important de collisions : en plus des trames perdues, les chemins identifiés ne sont pas nécessairement les plus courts. Dans [23], l'inondation est implémentée sous la forme d'un Enhanced CSMA/CA (Carrier Sense Multiple Access/Collision Avoidance) ou eCSMA/CA permettant de réduire les collisions et limitée dans le temps par des timers adaptés.

La majeure partie des propositions range-free ne spécifie pas clairement le protocole de communication utilisé mais l'étude de leur fonctionnement fournit des indications quant aux

prérequis. Par exemple, un réseau de capteurs peut être mis en place au-dessus d'un réseau 802.15.4 dans lequel les balises émettent périodiquement des beacons à un saut.

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

#### **4.4 Algorithmes de localisation**

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

##### **4.4.1 Algorithmes généraux**

Tous les algorithmes supposent un ensemble de données à l'entrée qui doivent être transformées en vue d'obtenir une estimation de la position. Ils ne se valent pourtant pas tous en termes de robustesse. Certains, dans leur forme standard, attendent des données exactes et échouent lorsque les informations de la réalité leur sont fournies car ces dernières comportent typiquement une erreur. Dans cette section, nous verrons d'abord les outils qui font l'hypothèse de données exactes puis ceux qui tiennent compte de la présence d'erreurs dans les mesures.

###### **4.4.1.1 Trilatération**

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

Soient trois balises  $B_1$ ,  $B_2$  et  $B_3$ ,  $M$  le nœud mobile que l'on désire localiser. L'exécution du protocole de mesure de distance a produit le triplet  $(d_1, d_2, d_3)$  et les échanges de données ont permis au mobile de connaître les positions des balises. Les équations 4.07 et 4.08 décrivent la relation entre  $B_1$ ,  $B_2$  et  $M$ .

$$(x - x_1)^2 + (y - y_1)^2 = d_1^2 \quad (4.07)$$

$$(x - x_2)^2 + (y - y_2)^2 = d_2^2 \quad (4.08)$$

The diagram shows three circles representing the possible positions of a mobile station (M) based on distances from three anchors (B1, B2, B3). The intersection of the first two circles is labeled M, and the intersection of the second and third is labeled M'. The distances are labeled d1, d2, and d3.

- $B_1(x_1, y_1)$ : Anchor 1 (blue dot)
- $B_2(x_2, y_2)$ : Anchor 2 (blue dot)
- $B_3(x_3, y_3)$ : Anchor 3 (blue dot)
- $M$ : Position probable du mobile (white circle)
- $M'$ : Position probable du mobile (white circle)
- $d_1$ : Distance from  $B_1$  to  $M$
- $d_2$ : Distance from  $B_2$  to  $M$
- $d_3$ : Distance from  $B_3$  to  $M$

$$M\left(\frac{y_2}{\tan\alpha+\tan\beta}; \frac{y_2\tan\alpha}{\tan\alpha+\tan\beta}\right) \quad (4.09)$$

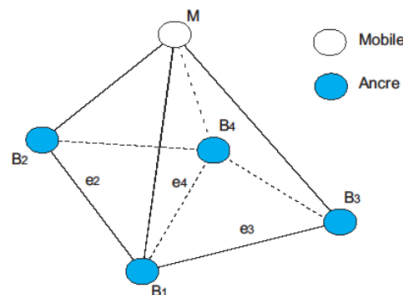
#### 4.4.1.3 Reconnaissance d'empreinte

Dans ce contexte, nous définirons une empreinte comme une collection d'échantillons d'une caractéristique susceptible de varier avec la position. Nous distinguerons les empreintes de référence des online fingerprints. Dans le premier cas, il s'agit de données capturées avant la mise en service du système de localisation (phase offline) : elles constituent un modèle de l'évolution des composantes de l'empreinte dans l'espace. Une fois la phase active déclenchée, de nouvelles mesures, dites online, seront collectées. Ces dernières seront comparées au modèle disponible en mémoire : la position correspondant à l'empreinte de référence la plus « proche » du cliché récent sera sélectionnée comme solution.

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

#### 4.4.1.4 Multilatération

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



**Figure 4.06 : Multilatération**



Le mobile est ici en contact radio avec des balises  $B_i (x_i, y_i)$ ,  $i=1 \dots 4$ . Le mobile émet un message et l'instant d'arrivée d'un message à la station  $i$  est noté  $t_i$ . Les ancres étant synchronisées, la différence entre ces instants peut être utilisée pour créer le système d'équations 4.10, 4.11, 4.12 où  $c$ 'est la vitesse de propagation,  $x_i$  et  $y_i$  les positions des nœuds  $B_i$  et  $e_i$  la distance entre le nœud  $B_i$  et une station choisie comme référence temporelle. Sur la figure 4.06,  $B_1$  joue ce rôle.

$$0 = c(t_2 - t_4) - \frac{e_2^2}{ct_2} + \frac{e_4^2}{ct_4} + 2x \left( \frac{x_2}{ct_2} - \frac{x_4}{ct_4} \right) + 2y \left( \frac{y_2}{ct_2} - \frac{y_4}{ct_4} \right) + 2z \left( \frac{z_2}{ct_2} - \frac{z_4}{ct_4} \right) \quad (4.10)$$

$$0 = c(t_2 - t_3) - \frac{e_2^2}{ct_2} + \frac{e_3^2}{ct_3} + 2x \left( \frac{x_2}{ct_2} - \frac{x_3}{ct_3} \right) + 2y \left( \frac{y_2}{ct_2} - \frac{y_3}{ct_3} \right) + 2z \left( \frac{z_2}{ct_2} - \frac{z_3}{ct_3} \right) \quad (4.11)$$

$$0 = c(t_3 - t_4) - \frac{e_3^2}{ct_3} + \frac{e_4^2}{ct_4} + 2x \left( \frac{x_3}{ct_3} - \frac{x_4}{ct_4} \right) + 2y \left( \frac{y_3}{ct_3} - \frac{y_4}{ct_4} \right) + 2z \left( \frac{z_3}{ct_3} - \frac{z_4}{ct_4} \right) \quad (4.12)$$

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

#### 4.5 Adaptations récentes des algorithmes à la localisation par WSN

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

##### 4.5.1 Méthodes range-based

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

###### 4.5.1.1 Temps de vol

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

connues et les coordonnées recherchées (équations 4.13 et 4.14). L'exécution itérative de cet algorithme demande une implémentation centralisée mais aussi un point de départ du calcul ou seed proche de la vraie solution.

$$F(x) = \sum_{i=1}^N \alpha_i^2 f_i^2(x) \quad (4.13)$$

$$f_i(x, y) = c\Delta_{t_i} - \sqrt{(x - x_i)^2 + (y - y_i)^2} \quad (4.14)$$

Le travail décrit dans [21] comporte deux parties : la mesure de distance et le positionnement. La seconde phase exploite, à travers MATLAB, les algorithmes Linear Least Squares (LLS), Non Linear Least Squares (NLLS) et Independent Time GPS Least Squares (ITGPSLS) pour calculer la position. Les estimations de distance sont générées à partir de résultats d'expérience utilisant la technologie Wifi. La répartition gaussienne des mesures a été introduite dans les simulations afin de reconstituer cet aspect de l'expérience réelle.

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

#### 4.5.1.2 RSSI et puissance reçue

L'étude de RADAR utilise la puissance du signal et le SNR pour déterminer la position. Les nœuds du réseau sont soit des balises, soit des mobiles, tous synchronisés.

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

Durant l'étape en ligne (online), les balises conservent une valeur moyenne de la puissance du signal et choisissent, pour retrouver la position, entre une combinaison des valeurs empiriques et un modèle de propagation. La recherche de la position se fait par la technique du Nearest Neighbor in Signal Space (NNSS) : cette méthode élit l'enregistrement pour lequel la distance entre le vecteur des Signal Strength (SS) stocké et celui mesuré est minimale. Une amélioration est considérée par le choix des  $k$  meilleurs NNSS et le calcul de leur centre de gravité comme la position du mobile. Cette

amélioration est encore plus nette si les  $k$  plus proches voisins sont choisis en ignorant les orientations : les points sont alors physiquement distincts : les triplets  $\langle \text{Position}_i, \text{Puissance}_i, \text{Nord} \rangle$  et  $\langle \text{Position}_i, \text{Puissance}_i, \text{Sud} \rangle$  ne sont plus différenciés.

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

Etablir ces différences n'étant pas une localisation en soi, les auteurs ont intégré leur proposition à deux méthodes connues : Nearest Neighbor (NN) et Bayesian Inference. NN repose sur une base de données de mesures qui sert de référence durant la phase online. La position juste est celle correspondant au point de l'espace des RSSI dont la distance euclidienne avec les valeurs mesurées est minimale. Un ajustement est apporté pour calculer la distance entre les différences. BI permet de choisir la position la plus probable connaissant les valeurs actuelles. L'adaptation de cette méthode à la proposition demande l'utilisation d'un vecteur d'observation  $o$  lié à une position  $l$ . Le vecteur  $o_k$  aura la forme (identifiant de paire d'AP, différence de RSSI pour la paire d'AP). La mise en relation de ces valeurs et de la position occupée lors de la mesure durant la phase offline permettra d'aboutir à l'équation 4.14. Partant de là, la probabilité de se trouver à une position  $l_k$  sachant la capture courante est donnée par l'équation 4.16. Retrouver la vraie position revient alors à maximiser cette probabilité en fonction de  $l_k$ .

$$\Pr(0|l_k) = \prod_{1 \leq i \leq j \leq m} \Pr(d(AP_i, AP_j) | AP_i \times AP_j, l_k) \quad (4.15)$$

$$P_r(l_k | o^*) = \frac{P_r(o^* | l_k) P_r(l_k)}{\sum_{k=1}^n P_r(o^* | l_k) P_r(l_k)} \quad (4.16)$$

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

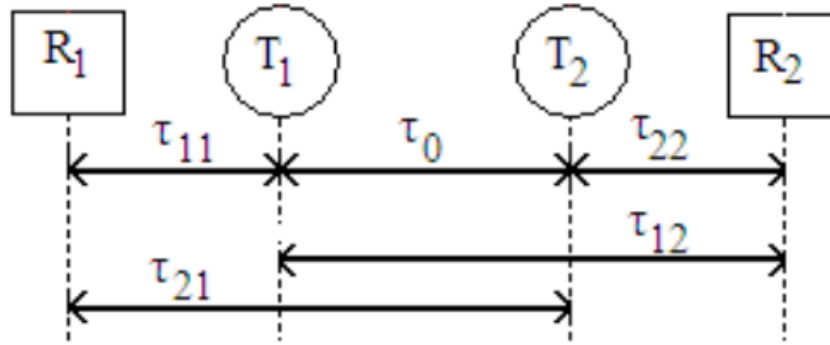
Le travail de [24] utilise un filtre particulière pour combiner fingerprinting et navigation inertielle. Un nouvel état est prédit par la loi du mouvement. La disponibilité d'un plan de l'environnement

permet d'assurer des déplacements réalistes. La reconnaissance d'empreinte fournit une base de données liant puissance reçue et position. Ces informations sont exploitées pour la transformation du vecteur de mesures, un n-uplet de puissances reçues, en une position. Finalement, le poids de chacune des particules évoluera en fonction de sa proximité avec la position issue des mesures et de la probabilité d'avoir traversé un mur.

#### 4.5.1.3 TDOA et phase du signal

Une solution proposée exploite le TDOA et la mesure de phase dans un réseau totalement synchronisé.

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



**Figure 4.07 :** *Disposition des nœuds et mesures temporelles*

Les différences de temps d'arrivée au niveau de chaque récepteur sont calculées et référencées par  $\Delta t_1$  et  $\Delta t_2$  (équations 4.17 et 4.18).

$$\Delta t_1 = \tau_{21} - \tau_{11} + \tau_{t_0} = \tau_0 + \tau_{t_0} \quad (4.17)$$

$$\Delta t_2 = \tau_{22} - \tau_{12} + \tau_{t_0} = -\tau_0 + \tau_{t_0} \quad (4.18)$$

Le terme  $\tau_{t_0}$  représente ici la différence entre les instants d'émission des signaux et  $\tau_0$  correspond à l'écart temporel lié à la distance séparant les émetteurs. La combinaison  $\Delta \tau$  de ces temps est proportionnelle à la distance séparant les émetteurs,  $d_0$ , d'où l'équation 4.19.

$$d_0 = c_0 \tau_0 = \frac{c_0}{2} \Delta \tau \quad (4.19)$$

Avec :

$$\Delta_{\tau} = \Delta t_1 - \Delta t_2 \quad (4.20)$$

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

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

#### 4.5.2 Méthodes Range-Free

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

##### 4.5.2.1 Famille DV-Hop

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

##### **DV-Hop originel**

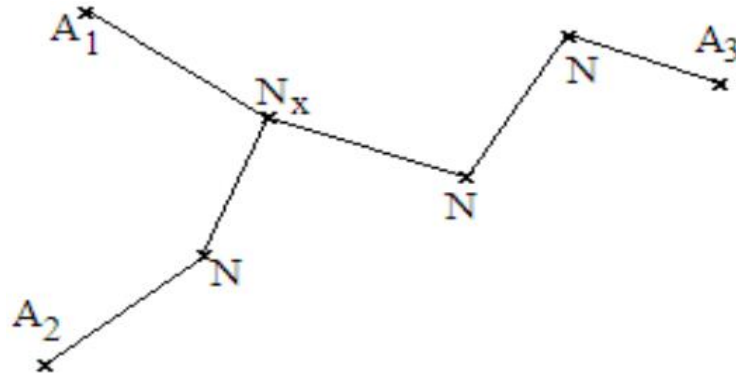
La méthode de localisation DV-Hop s'inspire des protocoles de routage à vecteur de distance pour proposer une estimation de la position des mobiles basée sur le nombre de sauts. Chaque ancre est ici chargée d'émettre une trame contenant sa position et un champ dédié au comptage de sauts. A la réception de ce message, tous les nœuds décideront de le retransmettre ou non en comparant le nombre de sauts vers l'ancre contenu dans la trame à l'information stockée en mémoire. Si le hopcount est inférieur au contenu de la mémoire, les données sont mises à jour, le champ de la trame est incrémenté et la trame retransmise. Une fois que toutes les balises ont récupéré la position et le nombre de sauts les séparant de leurs homologues, elles calculent une distance moyenne par saut (hopsize). Par exemple, la balise  $A_1$  (figure 4.08) peut calculer les distances géométriques entre elle et les balises  $A_2$  et  $A_3$ . Elle connaît également le nombre de sauts entre elle et ses voisines. L'équation 4.21 permet d'estimer la taille de saut. Ici,  $d_{i,j}$  est la distance réelle entre les balises  $i$  et  $j$  et  $\text{hop}_{i,j}$  correspond au nombre de sauts entre les balises  $i$  et  $j$ .

$$hopsize_1 = \frac{d_{1,2} + d_{1,3}}{hop_{1,2} + hop_{1,3}} \quad (4.21)$$

Cette taille de saut est rediffusée à l'ensemble du réseau. Elle sera utilisée par un nœud mobile quelconque pour estimer sa distance par rapport à une ancre. Un nœud la détermine par l'équation 4.22 où  $hopcount_{i,x}$  est le nombre de sauts entre le mobile  $x$  et la balise  $i$ . Une fois les distances à trois balises connues, le mobile peut appliquer la trilatération pour trouver sa position.

$$d_{i,x} = hopsize_i \times hopcount_{i,x} \quad (4.22)$$

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



**Figure 4.08 :** Principe topologique de DV-Hop

### Constraint DV-Hop

Une amélioration a été proposée avec pour point de départ la relation implicite existant entre la portée de communication  $D$  des nœuds et la distance calculée par DV-Hop. En effet, si un nœud se trouve à un saut d'une ancre, la distance entre les deux est inférieure ou égale à  $D$ . De même, si l'écart est de deux sauts, la borne devient alors le double de la portée. Partant de cette remarque, Constraint DV-Hop propose de déterminer les coordonnées du mobile en observant les contraintes liées à la portée. Le système s'écrit alors à l'aide des équations 4.23 à 4.25.

$$\min \sum_{i=1}^M ((x_i - x_u)^2 + (y_i - y_u)^2 - d_i^2)^2 \quad (4.23)$$

$$\sqrt{(x_{v_j} - x_u)^2 + (y_{v_j} - y_u)^2} \leq D, L = 1 \quad (4.24)$$

$$D \leq \sqrt{(x_{v_j} - x_u)^2 + (y_{v_j} - y_u)^2} \leq 2D, L \neq 1, j = 1, 2 \dots M_1 \quad (4.25)$$

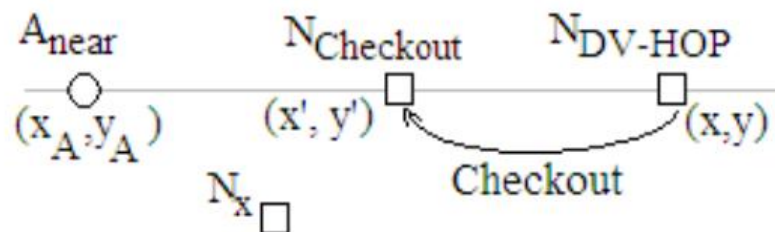
Cette proposition a été validée par une simulation de 100 nœuds, dispersés sur une surface de 50 m x 50 m. Les différentes expérimentations, avec un ratio d’ancres variables ou des modifications de la portée radio, ont indiqué la supériorité de CDV-Hop par rapport à DV-Hop.

### Checkout DV-Hop

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

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

Le travail réalisé représente également une amélioration de DV-Hop lorsque sont considérés les nœuds de niveau 2. Dans ce cas, Checkout DV-Hop, ajoute une étape de vérification (checkout). L’ancre la plus proche en termes de nombre de sauts, notée  $A_{near}$ , est celle pour laquelle l’accumulation d’erreurs lors du calcul de la distance entre le mobile et cette ancre est la plus faible. L’écart entre  $A_{near}$  et le mobile servira alors de référence pour corriger la position estimée par DV-Hop (figure 4.09).



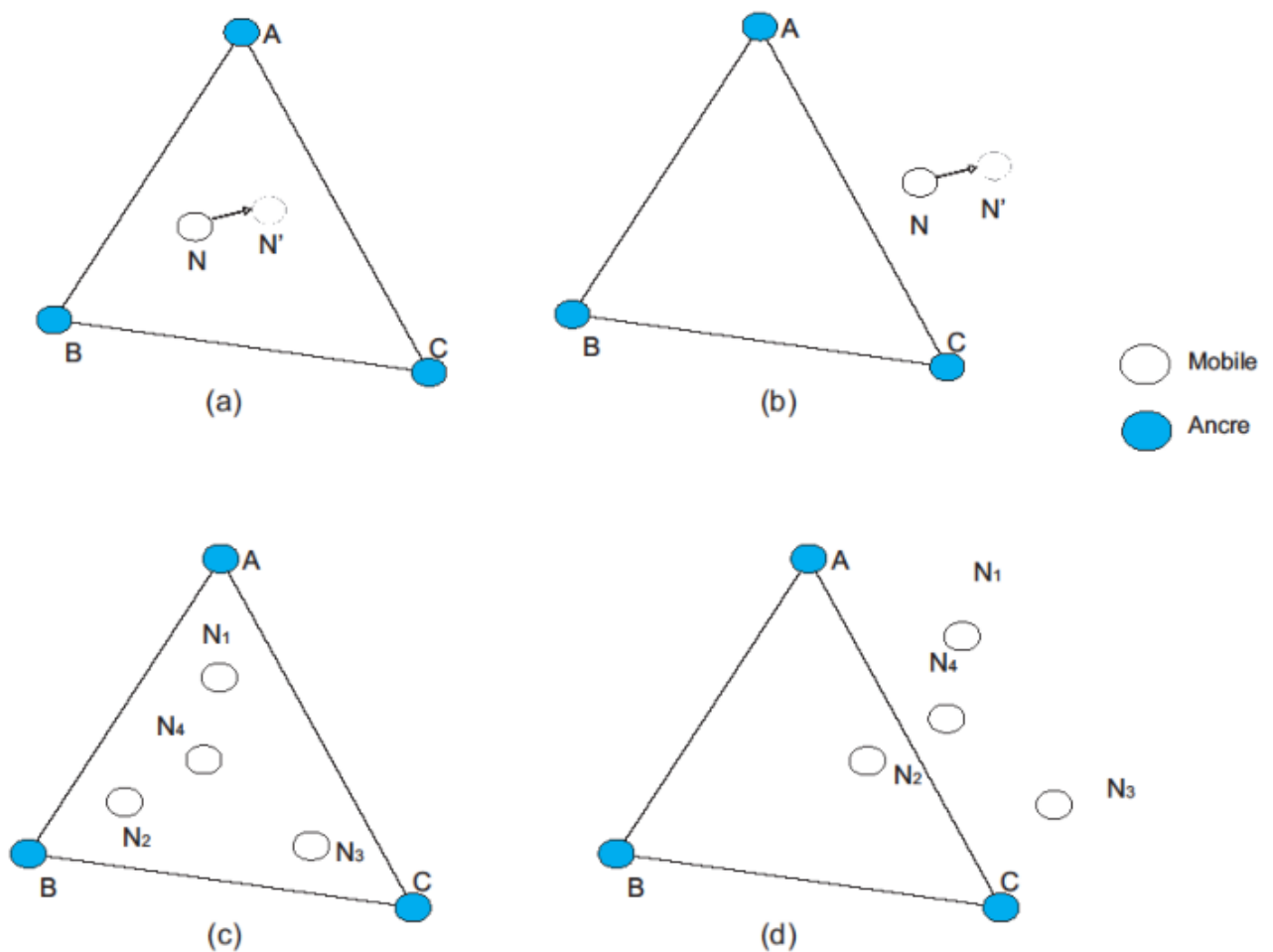
**Figure 4.09 :** *Checkout DV-Hop*

La figure 4.09 illustre l’effet de Checkout DV-Hop. La vraie position du nœud est  $N_x$  mais l’algorithme DV-Hop avait indiqué  $N_{DV-Hop}$  comme position probable. Connaissant la distance entre  $A_{near}$  et  $N_x$ , un nouveau point a été choisi par Checkout DV-Hop. Ce point,  $N_{checkout}$ , se trouve à la même distance d’ $A_{near}$  que  $N_x$  mais fait partie de la droite passant par  $A_{near}$  et  $N_{DV-Hop}$ .

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

#### 4.5.2.2 APIT

La méthode APIT (Area Point-In-Triangulation) est basée sur le Point-In-Triangulation test. Ce test permet à un nœud de déterminer s'il se trouve ou non à l'intérieur d'un certain triangle. Dans sa version mathématique, le test stipule que tout déplacement d'un mobile situé à l'intérieur du triangle correspond à son rapprochement/éloignement d'au moins un des sommets du triangle (figure 4.10(a)). S'il se rapproche/s'éloigne de tous les sommets à la fois, le mobile est à l'extérieur du triangle (figure 4.10(b)).



**Figure 4.10 :** Formulations géométrique et réseau de l'algorithme APIT

Ce test est adapté à un réseau en utilisant les relations entre les mobiles voisins du nœud d'intérêt et les ancres. Le mouvement du nœud n'étant pas forcément un paramètre contrôlé par l'application de localisation, les distances entre les voisins et les sommets sont utilisées pour émuler les déplacements. Ainsi, si tous les voisins d'un nœud N sont plus proches que N d'au moins un des



sommets, N est à l'intérieur du triangle (figure 4.10(c)). Cette version approximative du PIT est sensible au placement des nœuds. Dans les cas où ce dernier est irrégulier et pour celui où au moins un des voisins se trouve à l'extérieur du triangle, une mauvaise décision est prise (figure 4.10(d)). APIT est dit area-based car son objectif est l'obtention d'une surface contenant probablement le point recherché.

Les nœuds mobiles collectent les trames balises émises (beacons) par les ancres. Une fois la liste d'ancres à portée construite, le mobile en sélectionne trois et détermine s'il se trouve à l'intérieur du triangle ainsi formé. Le test se répète jusqu'à l'épuisement des combinaisons ou jusqu'à ce que l'intersection des triangles satisfasse le critère de taille de la surface de localisation obtenue. La surface obtenue correspond à la zone où se trouve très probablement le nœud. La position de ce dernier est alors associée au centre de gravité de cette surface.

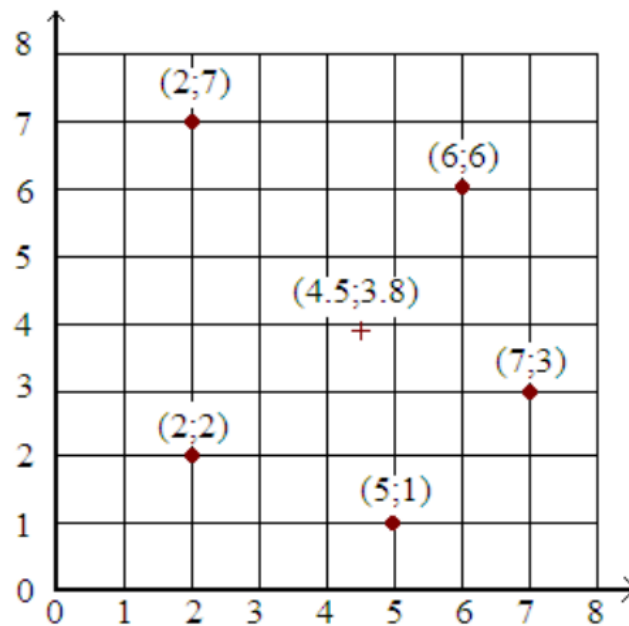
#### 4.5.2.3 Centroid

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

La méthode se base sur un quadrillage de nœuds de référence émettant des beacons à un intervalle T sans collision. Selon une métrique de connectivité calculée localement (équation 4.26) et un seuil Tresh, le mobile décide de tenir compte d'un sous-ensemble de balises détectées. Les facteurs de cette caractéristique sont :  $N_{recv}(i,t)$  : le nombre de beacons émis par la balise i et reçus durant le temps t et  $N_{sent}(i,t)$  : le nombre de beacons émis par la balise i durant le temps t.

$$CM_i = \frac{N_{recv}(i,t)}{N_{sent}(i,t)} \times 100 \quad (4.26)$$

Les coordonnées des balises retenues sont utilisées pour calculer le centre de gravité de la zone, position que le nœud considèrera comme sienne. Sur la figure 4.11, une réalisation de Centroid est illustrée avec uniquement les balises retenues représentées par un point. La position estimée est représentée par une croix.



**Figure 4.11 :** *Exemple de localisation avec Centroid*

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

#### 4.6 Conclusion

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

## **CHAPITRE 5**

### **REALISATION D'UN SIMULATEUR DE WSN**

#### **5.1 Introduction**

Afin de pouvoir évaluer l'efficacité des algorithmes de localisations dans un réseau de capteurs sans fils, nous proposons de développer une petite application qui permettrait de simuler un réseau de capteurs sans fil. Ce simulateur serait à la fois en charge de la modélisation des entités en jeux et des calculs de positions effectués en utilisant des algorithmes définies.

Nous effectuerons au préalable de brèves présentation du langage de programmation utilisé ainsi que les caractéristiques de l'environnement de développement sans oublier de définir les prérequis à l'utilisation du simulateur.

#### **5.2 Le langage de programmation C#**

C# est un langage orienté objet de type sécurisé et élégant qui permet aux développeurs de générer diverses applications sécurisées et fiables qui s'exécutent sur le .NET Framework. Vous pouvez utiliser le langage C# pour créer entre autres des applications clientes Windows, des services Web XML, des composants distribués, des applications client-serveur et des applications de base de données. Visual C# fournit un éditeur de code avancé, des concepteurs d'interfaces utilisateur pratiques, un débogueur intégré et de nombreux autres outils pour faciliter le développement d'applications basées sur le langage C# et .NET Framework. Le c# est dérivé du c++, on y retrouve aussi plusieurs caractéristiques des langages relativement récent à savoir par exemple le java. Il participe aussi pleinement à la création des pages Web dynamiques côté serveur et des services Web.

##### **5.2.1 Caractéristiques**

La syntaxe C# est très expressive, mais elle est également simple et facile à apprendre. La syntaxe de C# est facile à reconnaître à ses accolades si vous connaissez déjà les langages C, C++ ou Java. Si vous connaissez déjà l'un de ces langages, vous pouvez devenir très vite productif en C#. La syntaxe C# permet de répondre à de nombreuses complexités de C++ en fournissant des fonctionnalités puissantes telles que des types valeur Nullable, des énumérations, des délégués, des expressions lambda et des accès directs à la mémoire qui n'existent pas en Java. C# prend en charge des méthodes et types génériques qui améliorent la cohérence et les performances des types, ainsi

que des itérateurs, qui permettent aux implémenteurs de classes de collection de définir des comportements d'itération personnalisés simples à utiliser par le code client. Les expressions LINQ (Language Integrated Query) transforment les requêtes fortement typées en construction de langage de premier ordre.

En tant que langage orienté objet, C# prend en charge les concepts d'encapsulation, d'héritage et de polymorphisme. Toutes les variables et méthodes, y compris la méthode Main, point d'entrée de l'application, sont encapsulées dans des définitions de classe. Une classe peut hériter directement d'une classe parente, mais elle peut implémenter un nombre quelconque d'interfaces. Les méthodes qui substituent des méthodes virtuelles dans une classe parente requièrent le mot clé override, ce qui évite toute redéfinition accidentelle. En C#, le type struct ressemble à une classe légère ; alloué par pile, il peut implémenter des interfaces mais ne prend pas l'héritage en charge.

Outre ces principes orientés objet de base, C# permet de développer facilement des composants logiciel à travers plusieurs constructions de langage innovatrices, y compris les éléments suivants :

- Les signatures de méthodes encapsulées, appelées délégués, qui activent les notifications d'événement de type sécurisé.
- Les propriétés, utilisées comme accesseurs pour les variables membres privés.
- Les attributs, qui fournissent des métadonnées déclaratives à propos des types au moment de l'exécution.

Les commentaires de la documentation XML (eXtended Markup Language) inline.

- LINQ (Language Integrated Query) qui fournit des fonctions de requête intégrées à travers diverses sources de données.

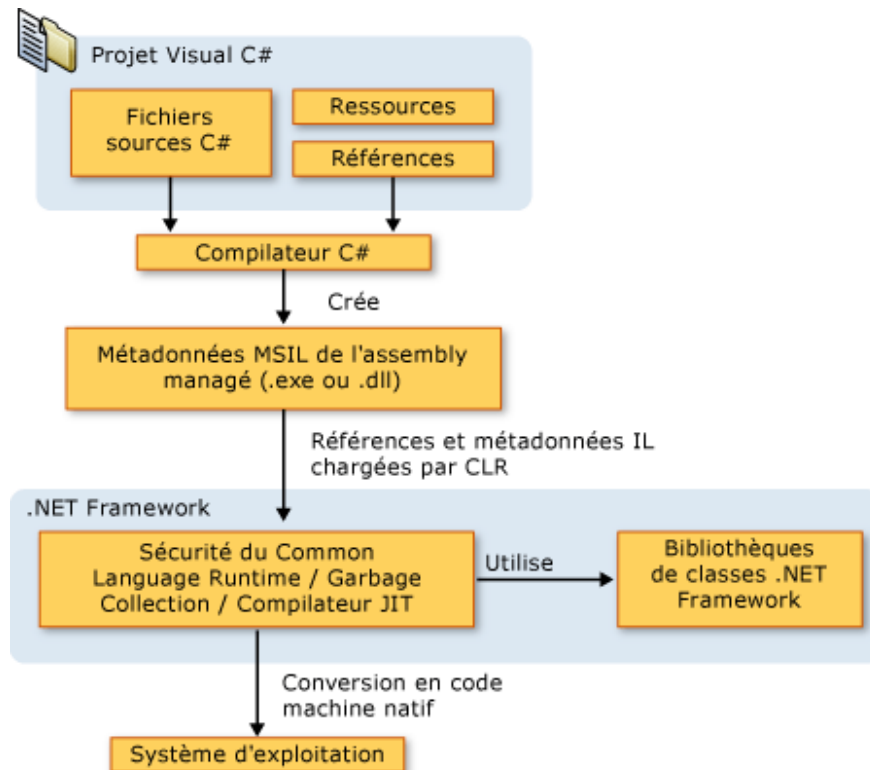
Si vous devez interagir avec d'autres logiciels Windows tels que des objets COM (Component Object Model) ou des DLL (Dynamic Link Library) Win32 natives, vous pouvez le faire en C# à l'aide d'un processus appelé « interopérabilité ». L'interopérabilité offre aux programmes C# autant de possibilités qu'une application C++ native. C# prend en charge les pointeurs et le concept de code « unsafe » lorsque l'accès direct à la mémoire est absolument essentiel.

En C#, le processus de génération est plus simple qu'en C et C++ et plus souple qu'en Java. Il n'y a pas de fichiers d'en-tête séparés et les méthodes et types peuvent être déclarés dans n'importe quel ordre. Un fichier source C# peut définir tout nombre de classes, structures, interfaces et événements. Les programmes en C# s'exécutent sur le .NET Framework, composant intégral de Windows qui inclut un système d'exécution virtuel appelé Common Language Runtime (CLR) et un ensemble unifié de bibliothèques de classes. Le CLR est l'implémentation commerciale de l'infrastructure du

langage commun (CLI, Common Language Infrastructure) de Microsoft, norme internationale constituant la base de toute création d'environnements d'exécution et de développement et assurant le fonctionnement homogène des langages et des bibliothèques.

Le code source écrit en C# est compilé dans un langage intermédiaire conforme à la spécification CLI. Le code IL de ce langage intermédiaire, ainsi que les ressources telles que les bitmaps et les chaînes, sont stockés sur le disque dans un fichier exécutable appelé assembly, dont l'extension est généralement .exe ou .dll. Un assembly contient un manifeste qui fournit des spécifications sur les types, la version, la culture et les conditions de sécurité de l'assembly.

À l'exécution du programme C#, l'assembly est chargé dans le CLR, qui peut prendre différentes mesures sur la base des informations du manifeste. Ensuite, si les conditions de sécurité sont respectées, le CLR effectue une compilation juste-à-temps pour convertir le code du langage intermédiaire en instructions machine natives. Le CLR fournit également d'autres services en rapport avec les opérations automatiques de garbage collection et la gestion des exceptions et des ressources. Le code exécuté par le CLR est quelquefois appelé « code managé », par contraste avec le « code non managé », compilé dans le langage machine natif ciblant un système spécifique. Le diagramme suivant illustre les relations entre les temps de compilation et d'exécution des fichiers en code source C#, des bibliothèques de classes .NET Framework, des assemblies et du CLR.



**Figure 5.01 :** *Structure de fonctionnement d'un projet Visual C#*

L'interopérabilité des langages est une fonctionnalité clé du .NET Framework. Étant donné que le code IL (Intermediate Language) produit par le compilateur C# respecte la spécification de type commun (CTS : Common Type System), il peut, lorsqu'il est généré à partir de C#, interagir avec le code généré à partir des versions .NET de Visual Basic, Visual C++, Visual J# ou d'un des vingt autres langages respectant la norme CTS. Un même assembly peut contenir plusieurs modules écrits dans différents langages .NET, et les types peuvent se référencer l'un l'autre exactement comme s'ils avaient été écrits dans le même langage.

Outre les services d'exécution, le .NET Framework inclut également une bibliothèque étendue de plus de 4 000 classes organisées en espaces de noms, qui fournissent une large gamme de fonctionnalités couvrant de nombreuses utilisations, de l'entrée/sortie de fichiers aux contrôles Windows Forms, en passant par la manipulation de chaînes et l'analyse XML. Une application C# standard utilise largement la bibliothèque de classes .NET Framework pour assurer les tâches de maintenance courantes les plus fastidieuses.

### **5.2.2 La bibliothèque de classe**

Si .NET Framework n'a pas une classe que vous recherchez, vous pouvez en créer une et pouvoir l'utiliser à plusieurs reprises dans différents programmes. Vous pouvez même créer une classe commerciale et pouvoir la distribuer ou vendre. Pour rendre ceci possible, vous pouvez « organiser » une ou plusieurs classes dans une bibliothèque. Une bibliothèque est un programme qui contient les classes et/ou d'autres ressources que d'autres programmes peuvent utiliser. Un tel programme est créé avec la même approche que les programmes exécutables. Puisqu'une bibliothèque n'est pas un exécutable, elle n'a pas besoin de la fonction Main(). Une bibliothèque a habituellement l'extension .dll.

Une bibliothèque peut être constituée d'un simple fichier ou autant de fichiers selon les besoins. Un fichier qui fait partie d'une bibliothèque peut contenir une ou plusieurs classes. Chaque classe devrait mettre en œuvre un comportement qui peut par la suite être utile et accessible à d'autres classes.

Les raisons pour lesquels nous utilisons des bibliothèques de classes :

- Réutilisabilité
- Architecture

#### **5.2.2.1 Réutilisabilité**

Le projet de type bibliothèque de classes permet d'obtenir des assemblies avec l'extension .dll.

Nous pouvons y mettre tout le code C# que nous voulons, notamment des classes qui auront un intérêt à être utilisées à plusieurs endroits ou partagées par plusieurs applications. C'est le cas des assemblys du framework .NET. Elles possèdent plein de code très utile que nous aurons avantage à utiliser pour créer nos applications.

#### 5.2.2.2 Architecture :

L'autre avantage dans la création de bibliothèques de classes est de pouvoir architecturer son application de manière à faciliter sa mise en place, sa maintenabilité et l'évolutivité du code. Les raisons de l'usage d'une bonne architecture de programme.

Architecturer son application est important surtout si l'application est grosse. Par exemple, une bonne pratique dans une application informatique est la décomposition en couches. Nous n'allons pas faire ici un cours d'architecture, mais le but est de séparer les composantes de l'application. Un grand nombre d'applications de gestions est composée d'une couche de présentation, d'une couche métier et d'une couche d'accès aux données, les couches communiquant entre-elles.

Le but est de limiter les modifications d'une couche lors de la modification d'une autre. Si toutes les couches étaient mélangées alors chaque modification impliquerait une série de modifications en chaine.

Il est donc intéressant d'avoir une bibliothèque de classes qui gère l'accès aux données, une autre qui gère les règles métier et une autre qui s'occupe d'afficher le tout.

#### 5.2.3 *Comparaison entre C# et Java*

Java et C# sont deux langages qui se ressemblent beaucoup.

Java a l'avantage d'être plus ancien donc plus éprouvé et il est très attractif car il dispose de beaucoup d'outils de développement, il a une constante évolution (qui est gérée par la JCP) et a des codes sources disponibles librement.

Mais Java n'a pas de véritable plateforme dédiée en tant que poste de travail, contrairement à C# (et .Net en général) qui a la plateforme Windows.

Les solutions connexes à C# (base de donnée, messagerie...) sont développées par le même éditeur, cela offre plus de facilite et moins de compétences pour développer des solutions intégrées.

#### 5.2.3.1 Développement multi langage

L'infrastructure .NET s'est approprié la primeur de l'aspect multi langage. Le multi langage occupe une place prédominante dans .Net et elle intègre plus d'une vingtaine de langages différents. Par exemple lorsqu'on utilise des classes fournies par le Framework elles peuvent très bien avoir été développées en C#, VB.NET ou en J#.

Pour que le code managé soit accessible aux développeurs utilisant un autre langage de programmation, le .NET Framework propose la spécification CLS (Common Language Specification) qui décrit un ensemble fondamental de fonctionnalités de langage et définit les règles d'utilisation de ces fonctionnalités.

L'architecture .NET a créé un nouveau concept, l'interopérabilité des langages, plus ou moins marketing destinés à séduire la communauté des développeurs, dans ce domaine .Net à largement l'avantage par rapport à la plateforme J2EE qui possède l'inconvénient majeur d'avoir centré son architecture autour d'un seul et même langage : Java.

#### 5.2.3.2 Les principaux IDE Java

- Eclipse : Eclipse est une plateforme de développement écrite en Java, fruit du travail d'un consortium de grandes entreprises (Borland, IBM, ...). Il en résulte un IDE (Integrated Development Environment) performant et OpenSource qui a su trouver sa place.
- NetBeans : NetBeans crée à l'initiative de Sun Microsystem, présentes toutes les caractéristiques indispensables à un RAD Java de qualité. De licence OpenSource, NetBeans permet de développer et déployer rapidement et gratuitement des applications graphiques Swing, des Applets, des JSP/Servlets, des architectures J2EE, dans un environnement fortement customisable.

#### 5.2.3.1 Les principaux IDE C#

- Eclipse : Il existe un plugin permettant de programmer en C# avec le célèbre logiciel d'IBM.
- Visual Studio : Visual Studio .Net est le célèbre logiciel de Microsoft, très complet, très intuitif, pas grand-chose à lui reprocher sauf le prix peut être...



### 5.2.3.2 Interfaces graphiques

Avec Java il existe 3 APIs (Application Program Interface) pour la création d'interfaces graphiques :

- AWT utilise les librairies native de chaque OS, donc les fonctions se réduisent au plus petit dénominateur commun... c'est à dire, pas grand-chose de fonctionnel. Le look est dépendant de la plateforme.
  - Swing 100% java, est plus lente que des librairies natives. Le look est indépendant de l'OS (Operating System).
- SWT le look est à moitié dépendant de l'OS, mais on perd le garbage collector -> retour des fuites de mémoire.

Tous les composants de AWT ont leur équivalent dans Swing en plus joli et avec plus de fonctionnalités en plus d'offrir de nombreux composant qui n'existent pas dans AWT. Bref les utilisateurs de Java utilisent Swing plutôt que AWT. Mais celle-ci est réputée pour sa lenteur... (par rapport entre autre à une application créée en Windows forms).

Effectivement, la lenteur de Swing vient surtout du fait que l'interface sera identique sur toutes les architectures, ce qui est bien plus que de la simple portabilité! Il aurait peut-être était intéressant que Sun soit moins radical sur cela ... (que le checkbox soit rond, carré ou autre n'a pas une grande importance, surtout entre différent poste client!).

La question est donc :

Vaut-il mieux avoir un programme avec une interface graphique plus rapide et pas portable (GDI+ (Graphical Device Interface) de .NET(C#)) ou une interface graphique plus lente et entièrement portable (swing)?

Le sujet porte ici plutôt sur la portabilité, mais il ne faut pas se voiler la face, il y a beaucoup plus de monde qui travaille sous Windows que sous Linux et les utilisateurs Windows auront du mal à se mettre à Linux (il faut changer ses habitudes et ce n'est pas facile).

Mais optimisations du programme ou non, une interface graphique en swing sera toujours beaucoup fois plus lente qu'une interface native, c'est lié à l'architecture de swing, et au choix de Sun (que l'on peut trouver judicieux) de ne (presque) pas utiliser de composants natifs.

Si on part donc du fait que les utilisateurs finaux sont sur Windows, il est vrai qu'il est préférable d'utiliser C# qui s'intègre parfaitement dans l'ensemble des produits microsoft.

On a beau la critiquer, swing reste une excellent librairie destinée aux interfaces graphiques, aux performances tout a fait honnête, avec un 'look and feel' pluggable, une très bonne architecture (MVC : Model - View – Controller ; entre autres), et 100% portable avec ça.

Ce qui est sûr, c'est que la portabilité n'est pas gênante en soit, d'autant qu'il paraît que SWT marche plutôt bien. C'est plus simple que Swing et plus rapide. L'application aura un look dépendant de la machine, mais ça reste portable.

#### 5.2.3.3 Les inconvénients de Microsoft .Net

- Microsoft ne fournit le framework .NET que pour ses systèmes d'exploitations tel que Windows, Windows CE et la Xbox 360. Le portage sur d'autres systèmes d'exploitations est possible car les spécifications de la CLI sont disponibles (mais pas des frameworks). Les classes de bases Base Class Library (BCL), sont une partie de la bibliothèque de classes du framework (Framework Class Library ou FCL). La BCL fournit des classes qui encapsulent un certain nombre de fonctions courantes, comme la lecture et l'écriture de fichiers, le rendu graphique, l'interaction avec les bases de données, la manipulation de documents XML, etc. Le code source de la BCL a été rendu disponible pour faciliter le débogage des sessions dans Visual Studio 2008.
- La totalité du code source du framework n'est pas encore disponible. Il est actuellement possible de télécharger une bonne partie de la source du framework.
- .NET ne fonctionnant pleinement que sous Windows, il est très difficile de changer de système d'exploitation ce qui crée une relation de dépendance au seul fournisseur Microsoft. Cependant, le projet Mono essaie de pallier ce problème.
- Les applications fonctionnant avec du code managé en utilisant les environnements tels que le CLR du Framework Microsoft ou la JVM (Java Virtual Machine) Java ont tendance à nécessiter plus de ressources systèmes que des applications fournissant les mêmes fonctionnalités mais qui accèdent plus directement aux ressources. Cependant, certaines applications ont montré de meilleures performances en utilisant le .NET Framework qu'en utilisant leur version en code natif. Ceci peut être attribué aux optimisations du runtime rendu possible par un tel environnement, à l'utilisation de fonctions performantes au sein du framework, à la compilation à la volée du « managed code » ou encore à certains aspects de la CLR.
- Les langages compilés à la volée produisent du code qui peut être plus facilement rétro-analysé que s'ils étaient écrits en code natif, par conséquent il y a un risque en ce qui concerne la perte de secrets et le risque de passer outre les mécanismes de contrôle de licence. Plusieurs techniques

pour rendre le code impénétrable ont déjà été développées pour l'empêcher. Visual Studio 2005 inclut un tel outil (dotfuscator).

- Puisque le framework n'est pas porté sur les anciennes versions de Windows, une application utilisant un framework doit vérifier qu'il est présent, et s'il n'est pas présent, l'application doit guider l'utilisateur pour l'installer. Cette contrainte peut dissuader certains utilisateurs d'utiliser l'application.
- Les versions récentes du framework (3.5 et supérieures) ne sont pas préinstallées quelle que soit la version de Windows. Certains développeurs sont dérangés par la taille importante du framework (environ 54 Mo pour le .NET 3.0 et 197 Mo pour la version 3.5) et par le manque de fiabilité des installateurs. Ce problème est en partie résolu par l'introduction d'une version allégée .Net Client Profile, et qui ne pèse que 26,5 Mo.

### **5.3 Cadre de travail**

#### **5.3.1 Environnement matériel**

Le programme en soit est très léger une machine :

Pentium IV

Ram: DDR2 1Go

L'espace mémoire requis par le logiciel 10Mo.

Toutes fois lors du développement nous avons utilisé un ordinateur :

Processor: Intel(R) Core(TM) i5-3230M

RAM: 4Go

La raison de la différence de performance entre la machine utilisé pour le développement du programme et la machine de l'utilisateur vient de l'exigence de l'environnement de programmation que nous allons présenter dans la suite de ce chapitre après avoir détaillé les exigences logiciels.

#### **5.3.2 Environnement logiciel**

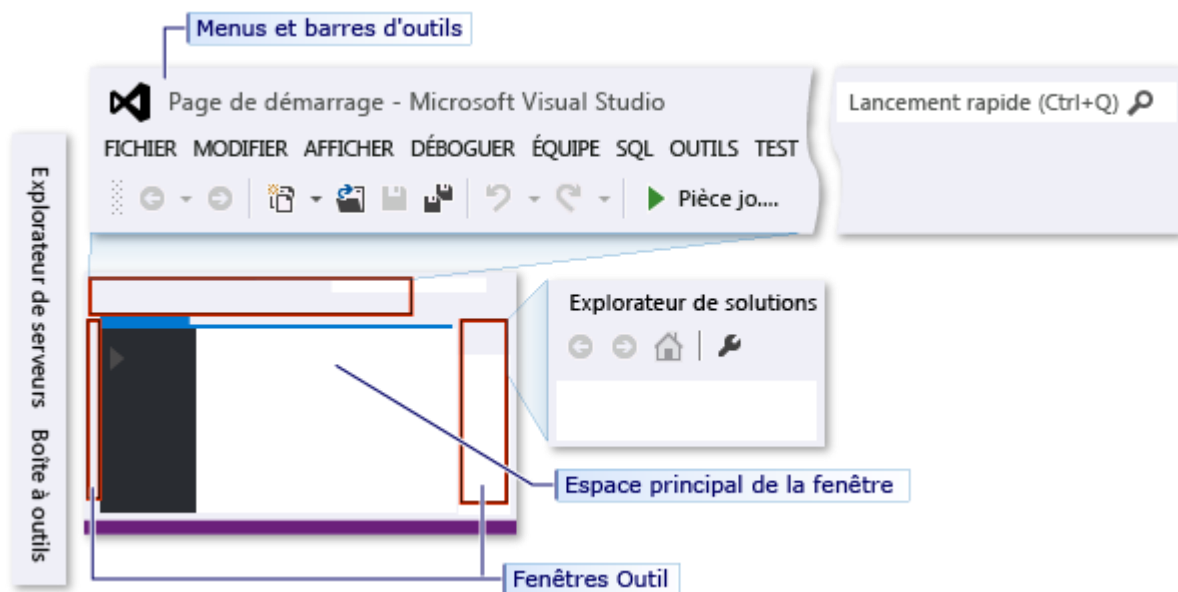
L'OS n'est pas réellement un problème pour le fonctionnement du programme par contre pour pouvoir installer les outils de développement la configuration suivante est nécessaire.

OS : Windows 8.1

Dans notre précédente présentation de C#, ce langage dépend des bibliothèques .Net dans notre cas nous avons utilisé : **.Net Framework 4.5**

## 5.4 Présentation de Visual Studio 2013

Après avoir ouvert Visual Studio, vous pouvez identifier les fenêtres Outil, les menus et barres d'outils et l'espace de la fenêtre principale. Les fenêtres Outil sont ancrées sur les côtés gauche et droit de la fenêtre d'application. Lancement rapide, la barre de menus et la barre d'outils standard sont situés en haut. La page de démarrage est située au centre de la fenêtre d'application. Lorsqu'une solution ou un projet est chargé, les éditeurs et les concepteurs apparaissent dans cet espace. Lorsque vous développez une application, vous passez la majeure partie de votre temps dans cette zone centrale.



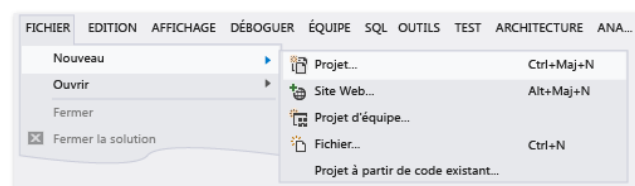
**Figure 5.02 :** Fenêtre principale de Visual Studio 2013

### 5.4.1 Créer une application simple

Lorsque vous créez une application dans Visual Studio, vous créez d'abord un projet et une solution. Pour cet exemple, vous allez créer une solution Windows Presentation Foundation.

### 5.4.2 Pour créer le projet WPF

1. Créez un nouveau projet. Dans la barre de menus, sélectionnez Fichier, Nouveau, Projet.



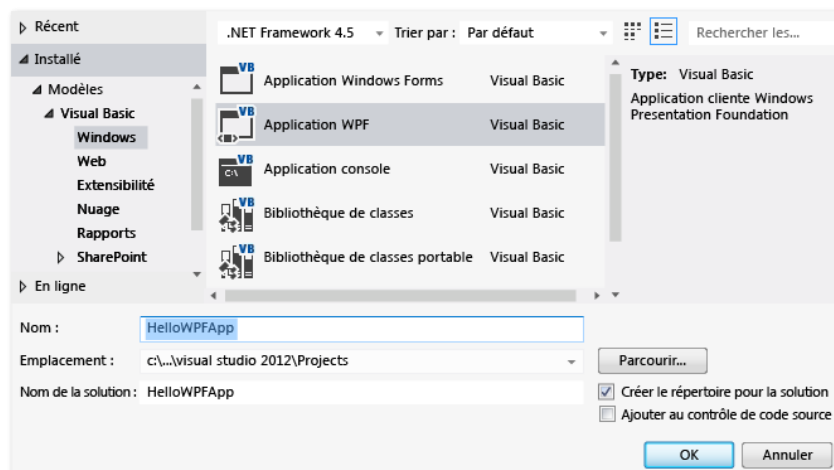
**Figure 5.03 :** Menu pour la création de projet

Vous pouvez également taper Nouveau projet dans la zone Lancement rapide pour effectuer la même opération.



**Figure 5.04 : Détail du menu**

2. Choisissez le modèle d'application WPF Visual Basic ou Visual C#, puis nommez le projet HelloWPFApp.



**Figure 5.05 : Choix du type de solution à créer.**

## 5.5 Le simulateur

La plupart des simulateurs WSN ne soutiennent pas un ou deux des critères suivants: faciles, extensibles et évolutives, de sorte que nous concevons et mettons en place un cadre intégré utilisé pour la localisation dans les réseaux de capteurs et disponibles pour l'analyse des différents types de systèmes de localisation (Range-libre, basée Range et algorithmes de localisation hybrides). Le cadre obtenu est simple et facile à utiliser pour démontrer l'évaluation de la performance et la comparaison des différents types de régimes de localisation. Il peut également être étendu pour inclure de nombreux algorithmes de localisation par les futurs chercheurs.

### 5.5.1 *Critères de conception*

- Supporte complètement réseaux à grande échelle, par exemple 400 x 400 noeuds des noeuds qui sont entièrement interconnectés qui implique 160000 connexions réseau. Ainsi, notre cadre proposé est évolutive.
- Le réseau peut être déployé sur la base d'une large gamme de paramètres: la taille du réseau, nœuds de capteurs à distance de communication, rayon de localisation (de portée de communication), l'angle du secteur (angle de rotation) et de la largeur du faisceau de localisation.
- Prise en charge multithread, où l'algorithme de simulation est exécuté sur un fil séparé à partir du fil à commander l'interface de révélateur.
- Emballage toutes les classes de simulateurs dans un fichier DLL référencé pour une meilleure conception de la modularité.
- Facilité de mise en œuvre de divers algorithmes de localisation par encapsulation chacun dans un fichier DLL séparée. Les développeurs peuvent écrire leur propre algorithme de localisation dans un fichier DLL et le référencer dans notre cadre.
- Peut être étendu à mettre en œuvre de nombreux algorithmes de localisation avec les différentes catégories (sans plage, base-gamme et un hybride des deux).
- Interface révélateur approprié pour permettre le déploiement et le déplacement nœuds de capteurs et de localisation et de modifier différents paramètres de simulation.

### 5.5.2 *Caractéristiques*

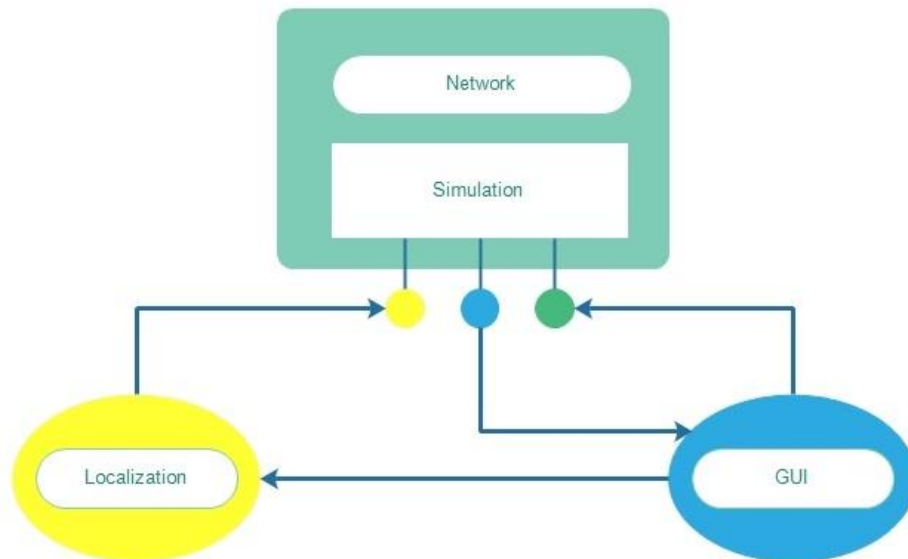
Le programme que nous allons développer sera divisé en 3 parties :

- Le réseau : les nœuds, les locateurs, les paquets transférer, ...
- Les algorithmes de localisation : chaque algorithme est stocké dans une bibliothèque indépendante.

Ces deux parties du simulateur sont des bibliothèques de classes pour éviter la confusion et de permettre une meilleure portabilité du programme.

- Interface de contrôle : cette partie est la partie manipulée par l'utilisateur.

Cette partie du programme ne sert qu'à interagir entre les 2 parties citées ci-dessus.



**Figure 5.06 :** Représentation de la structure générale du programme

De part ce diagramme nous pouvons voir que l’affichage sur l’interface utilisateur ne se fait qu’aux travers du simulateur. La bibliothèque de localisation n’interfère pas directement sur l’interface pour éviter l’encombrement et les problèmes de priorisation d’affichage.

#### 5.5.2.2 Les spécifications des algorithmes de localisation

Nous avons établi une certaine architecture de construction de la bibliothèque de l’algorithme de localisation pour permettre d’uniformiser l’utilisation et ainsi pouvoir fonctionner en collaboration avec les autres composants du programme.

```

public void set_Parameters (int locatorRange, int beamWidth ...)
{
    // Récupération des paramètres par l'interface utilisateur
}
public void connect (<Locator> localtorList, <Sensor> sensorList ...)
{
    // Sonnections
}
public void send (<Sensor> sensorList)
{
    // Transmission des paquets
}
public void localize (<Sensor> sensorList, int locatorRange ...)
{
    // Localisation du noeud
}
  
```

**Figure 5.07 :** Spécifications indispensables à chaque algorithme de localisation

Ces méthodes sont indispensables et doivent aussi être utilisées dans l'ordre dont elles sont présentées que la figure 5.06. Tout d'abord il faut récupérer les valeurs d'initialisations entrées par l'utilisateur sur l'interface principale. Cette fonction est « set\_Parameters ». Une fois les paramètres complets, la simulation commence par vérifier que la connexion entre les nœuds et les locuteurs fonctionnent en utilisant la méthode « connect ». Une fois la connexion effective, on simule la transmission de paquet par « send ». Et finalement, on effectue l'estimation de la position du nœud par la fonction « localize ».

Mis à part le constructeur de la classe « *AlgorithmeDeLocalisation* », il est possible de rajouter des méthodes supplémentaires. On peut aussi considérer que chaque algorithme ne dépend pas des mêmes paramètres. Nous avons pris soin de prendre en compte la majorité des paramètres communs à la majorité des techniques de localisation mais il est toutefois possible de rajouter des paramètres supplémentaires à l'interface graphique ; grâce à la structure que nous avons adoptée rajouter des éléments à l'interface n'affecterait pas le fonctionnement des éléments déjà présents dans le simulateur.

### 5.5.2.3 La bibliothèque du réseau

Nous avons rassemblés les informations et le fonctionnement du réseau. Cette librairie peut être considérée comme le cœur du programme. Elle contient les méthodes de construction et de lancement d'une simulation.

```

public WirelessSensorNetwork ( ... )
{
    // Constructeur du réseau
}
public void buildNetwork ()
{
    // Construction de la représentation graphique
}
public void runSimulation ()
{
    // Lancement de la simulation
}
public bool process (bool bSleep)
{
    // Gestion du processus
}

```

**Figure 5.08 :** Le constructeur et les méthodes de la classe réseau



Les paramètres utiles pour la construction du réseau sont intégrés dans le constructeur de la classe « WirelessSensorNetwork ». Comme pour les algorithmes de localisation les paramètres sont initialisés par l'utilisateur au travers de l'interface utilisateurs. Cette partie du simulateur ne comporte que trois méthodes.

La première étant « buildNetwork ». Cette méthode sert à mettre en place les éléments du réseau sur l'interface graphique. Cette fonction est aussi appelé déploiement du réseau.

Cette méthode à comme paramètres indispensables :

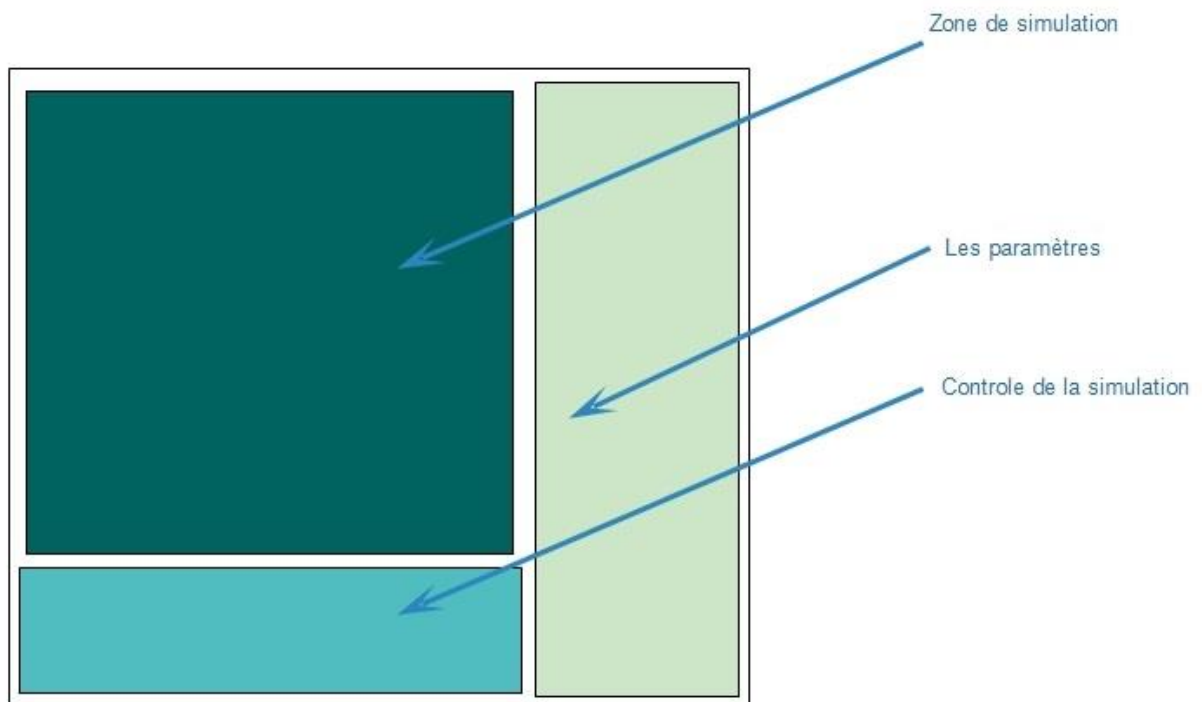
- La liste des balises
- La liste des nœuds
- La liste des localisateurs

« runSimulation » cette fonction démarre la simulation. Concrètement c'est au niveau de cette méthode que commence la simulation proprement dite. Cette fonction active la liaison effective entre les instances des algorithmes de localisation choisie et l'instance de réseau en cours de fonctionnement. En effet c'est à partir du moment où la simulation commence que les tests de localisation peuvent commencer.

« process », cette partie sert surtout à mettre à jours les déplacement des nœuds sur l'affichage. Cette fonction est surtout active lors du mode où les nœuds sont mobiles.

Notons toutes fois qu'il existe plusieurs autres mécanismes à l'intérieur du cœur du programme comme la génération d'instance de chaque nœud, chaque localisateur et chaque balise. Les classes que nous avons expliquées précédemment sont celles qui sont directement accessible par l'utilisateur, les autres fonctions travaillent en arrière-plan.

### 5.5.3 L'interface graphique



**Figure 5.09 :** *Présentation générale de l'interface du simulateur*

Comme nous pouvons le constater, l'interface se compose de 3 parties différentes.

- L'interface permettant de visualiser le déroulement de la simulation.
- Le paramétrage du réseau et de la localisation
- Les contrôles de la simulation

#### 5.5.3.2 Les paramètres

Dans ce paragraphe nous allons voir les paramètres que nous avons pris en compte pour la simulation. Nous avons la possibilité de manipuler des paramètres de localisation et des paramètres du réseau.

Sur l'interface ces 2 parties de paramétrages sont bien distinctes.

## a. Localisation

The image shows a software interface for configuring localization parameters. It includes various checkboxes, sliders, dropdown menus, and a grid of locators. Numbered callouts (1-10) highlight specific features:

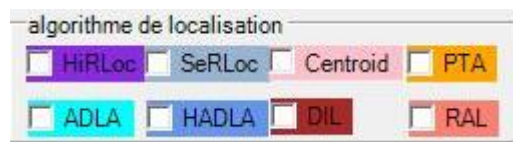
- 1**: ☐ Position estimé
- 2**: Locator Radio Range slider
- 3**: Mode de déploiement des localisateurs dropdown menu (Selectionner un élément)
- 4**: Locators' Beam Start Angle sliders for Locator 0 through Locator 7
- 5**: Type d'antenne de localisation: ☐ Directed, ☐ Omni-directional
- 6**: algorithmes de localisation: ☐ HiRLoc, ☐ Self, ☐ Centroid, ☐ PTA, ☐ ADLA, ☐ HADLA, ☐ DIL, ☐ RAL
- 7**: Nombre de localisateur: ☐ 2, ☐ 3, ☐ 4, ☐ 5, ☐ 6, ☐ 7, ☐ 8
- 8**: Niveau de puissance de ( pour PTA) dropdown menu (Choose Item (m))
- 9**: Sensors Mobility Model dropdown menu (Choose Item)
- 10**: Output file: Choose Output File button

**Figure 5.10 :** Le paramétrage de la localisation

- ①. Case à cocher pour décider du résultat de la simulation. L'estimation de la position permet d'avoir comme résultat de la simulation la position effective des nœuds ainsi que la position

estimé des nœuds obtenu par calcul de localisation par les algorithmes de localisation choisis.

- ②. A ce niveau on définit la couverture des antennes de localisation et des balises.
- ③. On a prédéfini la manière de dispersion des localisateurs bien que nous avons inclus une option où ces derniers sont dispersés aléatoirement.
- ④. Cette option ne concerne que les balises. Ce sont les orientations des balises. Dans notre cas, nous avons limité le réseau à 8 balises.
- ⑤. C'est le type d'antenne de localisation dont dispose le réseau
  - Omnidirectionnelle : pour une zone de couverture semblable à un disque.
  - Directionnelle : pour une zone de couverture semblable à une portion de disque.
- ⑥. Choix de l'algorithme de localisation.
  - Il suffit de cocher l'algorithme de votre choix, les nœuds dont la position peuvent être estimés se coloreront par la couleur qui souligne le nom de l'algorithme.



- La simulation est en mesure de combiner de différents algorithmes de localisation afin d'améliorer la performance de l'estimation de la position des nœuds.
- ⑦. Ce paramètre est le nombre d'intersection entre les zones de couverture des localisateurs afin d'obtenir une localisation. Notons que plus il y a d'intersection plus la localisation est précise, toutes fois cela suggère aussi que la disposition des localisateurs doivent être de manière à permettre plusieurs chevauchement entre leurs zones de couverture.
  - ⑧. Ce paramètre n'est utile que si vous employez l'algorithme PTA. Ce paramètre représente la catégorisation des niveaux de signal reçu. L'algorithme PTA, divise la zone de couverture du localisateur en plusieurs disques concentriques.
  - ⑨. Cette option permet de gérer la mobilité des nœuds. Nous avons mis à disposition 4 mouvements possibles mais il est suggéré d'utiliser la première proposition pour plus de réalisme dans la simulation. Les paramètres X et Y sont les vitesses de déplacement des nœuds.
  - ⑩. Cette option est pour le fichier de destination des données recueilli que nous allons présenter après la phase de simulation.

Comme nous avons pu le constater, il n'est pas nécessaire d'effectuer la modification de tous les paramètres mais cela dépend surtout du résultat que l'on attend et des algorithmes de localisation.

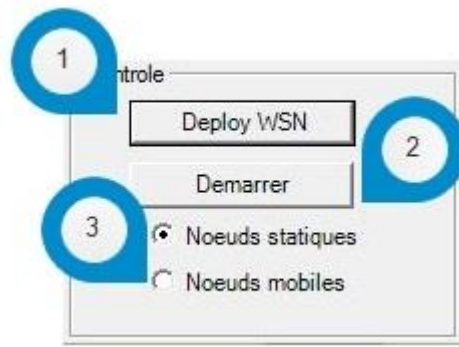
#### *b.Paramétrage du réseau*

The image shows a software window titled 'Paramètres du réseau' (Network Parameters). It is divided into two main sections. The top section, labeled 'Paramètres du réseau', contains three input fields: 'Number of Random Locators' with the value 8, 'Number of Static Sensor Nodes' with the value 200, and 'Number of Mobile sensors' with the value 20. A blue callout circle with the number 1 points to this section. The bottom section, labeled 'Bruit gaussien' (Gaussian noise), contains a radio button group with 'Non' selected and 'Oui' unselected. Below this are three input fields: 'Path Loss' with the value 2, 'Variance' with the value 1, and 'Mean' with the value 0. A blue callout circle with the number 2 points to this section.

**Figure 5.11 :** *Les paramètres du réseau*

- ①. Cette partie est la configuration matérielle du réseau c'est-à-dire le nombre de localisateurs si l'on choisit la disposition aléatoire, le nombre de nœuds dans le réseau, le nombre de capteur mobiles et le nombre de balises.
- ②. Cette option est faite pour tester l'efficacité des algorithmes dans un environnement bruité. L'utilisateur peut configurer le niveau de bruit par les valeurs des effets que le bruit a sur la transmission des paquets dans le réseau.

### *c.Paramétrages de la simulation*



**Figure 5.12 : Contrôle de la simulation.**

- ①. Le déploiement : ce bouton permet le déploiement du réseau une fois que tous les paramètres sont entrés.
- ②. Démarrage : c'est le bouton de démarrage de la simulation.
- ③. C'est le choix qui permet de simuler des nœuds mobiles ou des nœuds statiques. Notons que pour les nœuds statiques le calcul de localisation se fait une seule fois mais que pour les nœuds mobiles le calcul se fait à chaque déplacement.

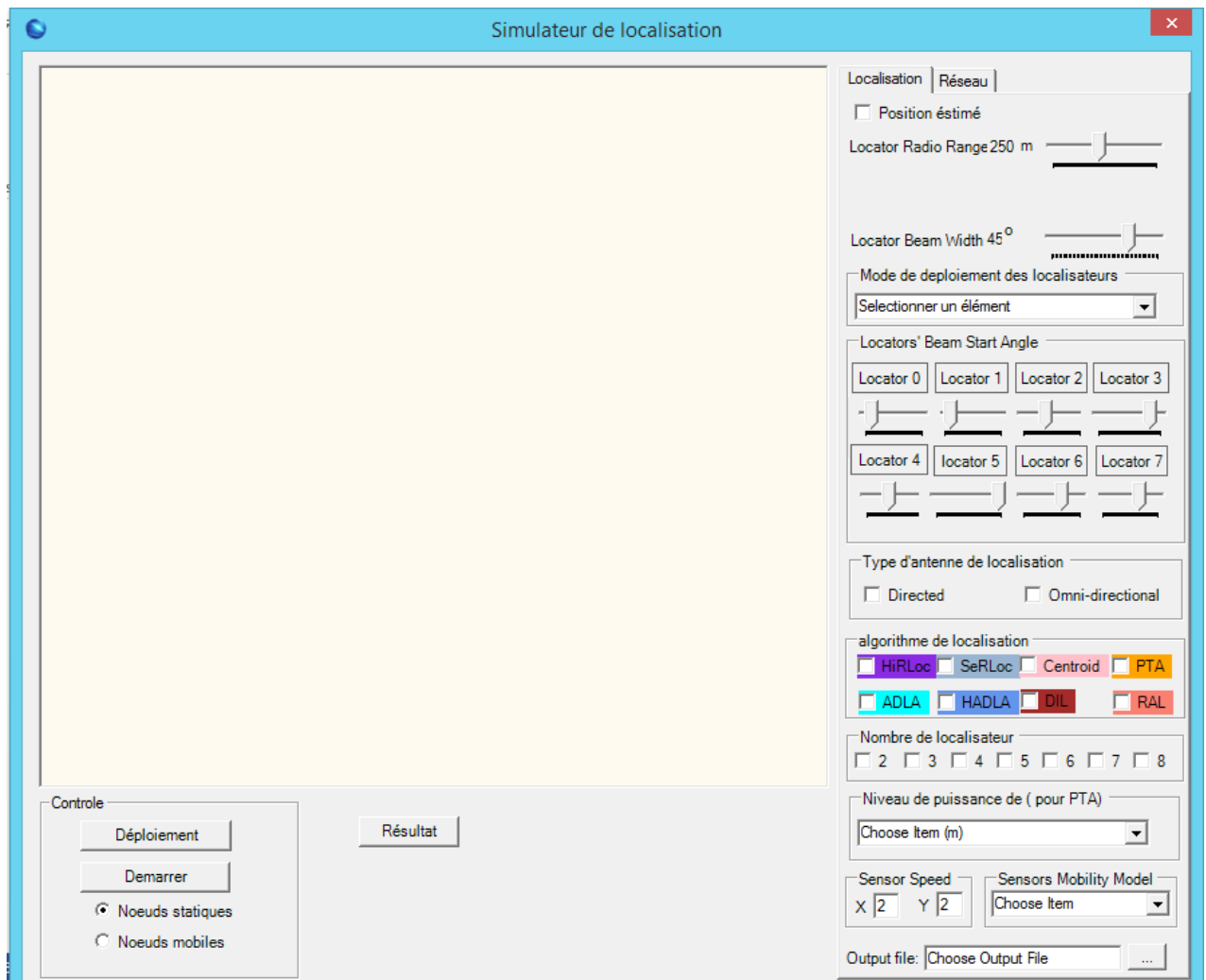
Le mode où les nœuds sont mobiles permettent de mettre en évidence sur l'écran la vitesse de calcul car on peut constater dans certaines conditions un léger décalage entre le déplacement du nœud et la position indiquée par l'estimation de la position.

## **5.6 Fonctionnement réel du programme**

Dans cette partie nous allons présenter le fonctionnement du programme selon les configurations de bases possibles sur le simulateur.

### **5.6.1 Présentation de l'interface finale.**

L'interface finale du programme n'est pas très différente de celle présentée dans les critères de conception.



**Figure 5.13 : Interface finale du simulateur**

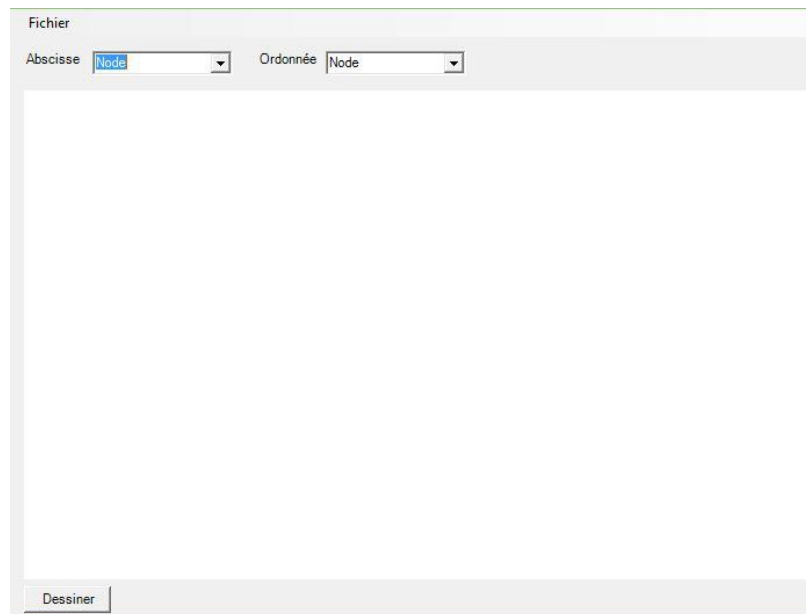
Cette interface est exactement la même que celle que nous avons présenté dans le paragraphe 5.4.3. Les seules différences sont :

- ①. La présence d'onglet pour séparer les configurations du réseau et de la localisation
- ②. Le bouton qui permet d'accéder à une interface qui effectue une représentation graphique du résultat.

L'interface a été conçue pour faciliter la manipulation du programme. Malgré le fait que la présence de plusieurs boutons et de fonctionnalité sur le premier plan, l'utilisateur final n'aura pas à se soucier de rechercher dans des menus les modifications de paramètres car elles sont toutes présentes sur cette interface.

Du fait que l'on a placé tous les options disponibles sur l'interface principale avec la zone d'affichage de la simulation proprement dite, cette fenêtre est dimensionnée **892x740** ; cette taille est fixe pour éviter que certains boutons ne soit pas visible à l'écran.

Notons toutes fois que l'interface qui affiche les résultats graphiques est indépendante de cette fenêtre.



**Figure 5.14 :** *Représentation graphique du résultat*

Ce module complémentaire extrait les données contenu dans le fichier de sortie et effectue une ou des représentations graphiques selon le choix de paramètres de l'utilisateur.

Pour permettre un meilleur affichage de résultat nous conseillons d'utiliser un fichier de sortie par algorithme de localisation.

Mais si vous souhaitez consulter les données brutes du résultat ils sont contenu dans le fichier de sortie et sous la forme suivante.



```

Node,X Estimated,Y Estimated,X Actual,Y Actual,Ex,Ey>Error
2,520,145,511,188,9,43,43
5,395,520,365,511,30,9,31
6,520,145,507,253,13,108,108
7,145,145,253,268,108,123,163
10,520,395,567,511,47,116,125
11,395,395,316,282,79,113,137
12,395,520,332,472,63,48,79
14,395,145,332,260,63,115,131
16,520,395,576,411,56,16,58
18,145,520,182,511,37,9,38
19,395,20,332,9,63,11,63
20,145,520,253,438,108,82,135
23,520,395,547,411,27,16,31
27,395,520,332,489,63,31,70
28,395,520,426,511,31,9,32

```

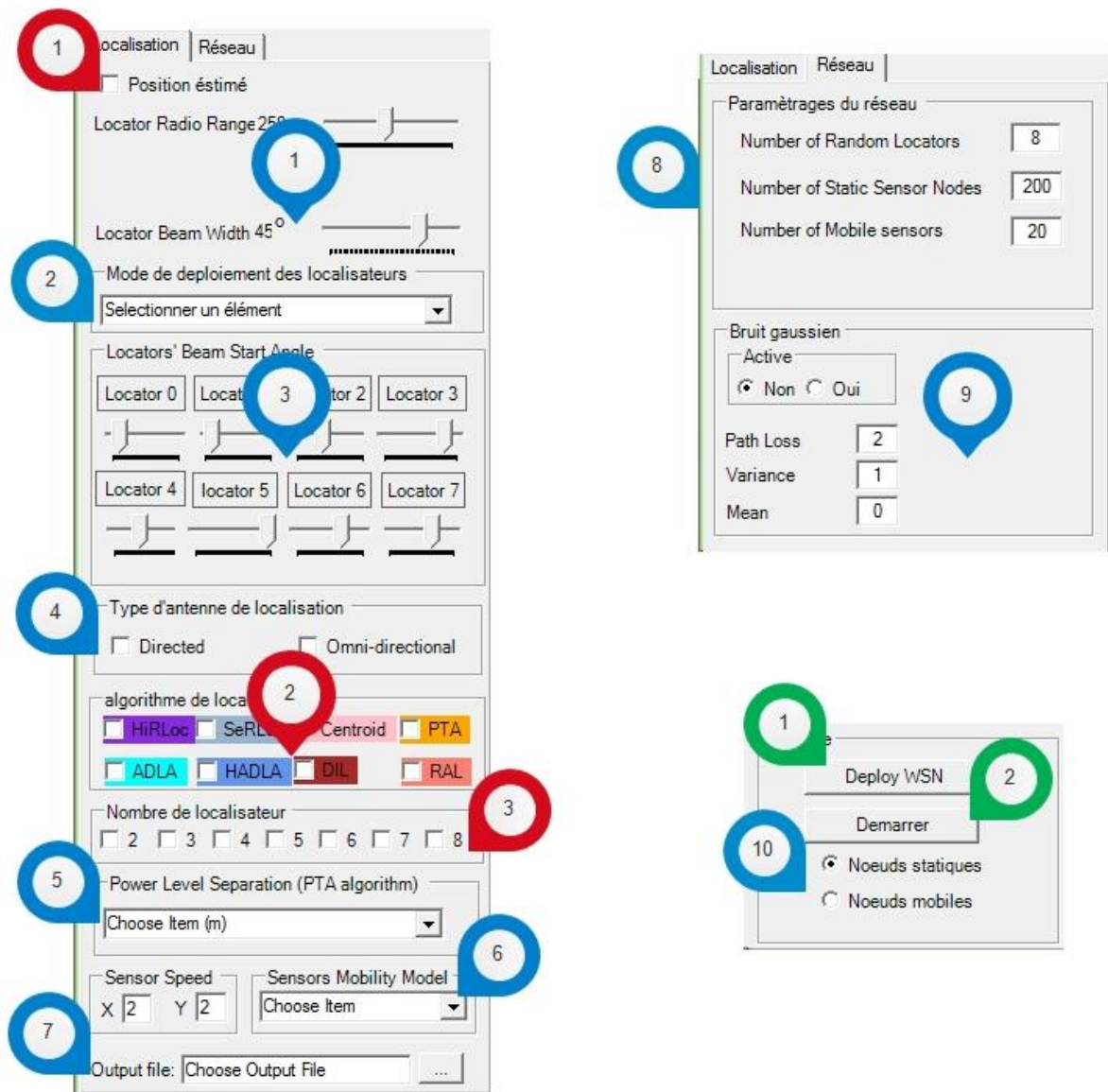
**Figure 5.15 :** *Contenu du fichier de sortie*

L'avantage du fait de consulter le fichier de sortie est le fait de pouvoir accéder aux résumé de l'analyse car dans ce fichier on peut voir le nombre de nœuds que le système a pu effectuer l'estimation de la localisation, on peut aussi voir l'erreur moyenne ainsi que le temps d'exécution du système de localisation.

### **5.6.2 Mode d'utilisation et illustration**

Nous allons effectuer un test rapide et facile concernant la manipulation du programme. Notons que dans ce test nous utiliserons les paramètres par défaut qz notre action n'est pas indispensable.

Une fois que le simulateur soit lancé. L'ordre d'initialisation des paramètres est la suivante.



**Figure 5.16 : Mode d'utilisation du simulateur**

Pour paramétrer correctement le programme veuillez remplir les valeurs dans l'ordre :

Pour commencer il faut paramétrer les éléments matériels, ce sont les éléments marqué d'une balise bleue.

Ensuite utiliser les balises vertes pour déployer et ensuite lancer la simulation.

Enfin manipuler les éléments marqués par des balises rouges.

### Remarque

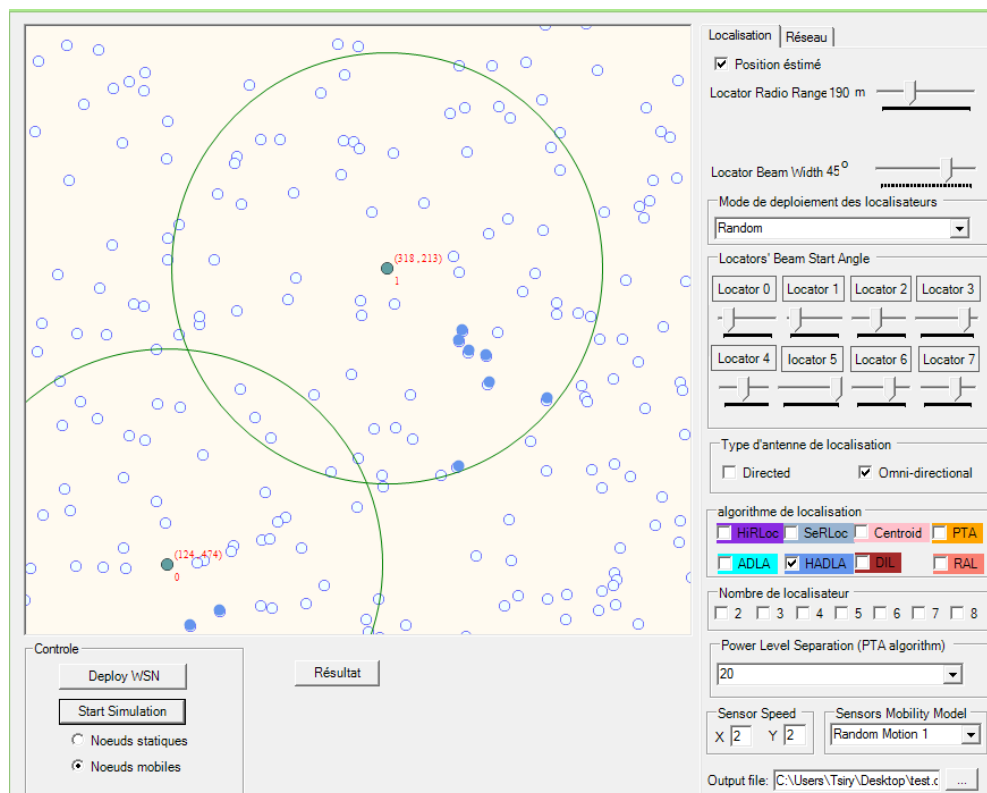
Les éléments marqués par des balises bleues peuvent être modifié avant le démarrage de la simulation même si l'on a déjà appuyé sur le déploiement du réseau. Une fois les nouveaux

paramètres entrés, il faut à nouveau appuyer sur le bouton de déploiement. On peut aussi admettre que le remplissage des balises bleu ne suit pas l'ordre indiqué.

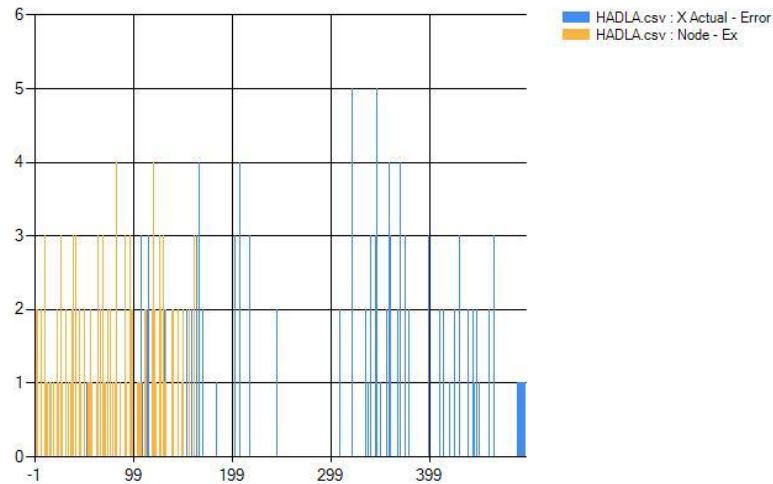
Une fois que la simulation est lancée, il est impossible de modifier les paramètres matériels sans à avoir à redémarrer entièrement le programme ; alors veuillez vérifier consciencieusement les paramètres physiques avant de lancer la simulation.

Il est possible de stopper la simulation et de la reprendre par la suite, dans cette condition seule les paramètres en balises rouges peuvent être modifiés.

L'unique exception aux règles est la localisation du fichier de sortie. Il est vrai que le programme refuse de déployer le réseau si ce paramètre n'est pas correctement rempli. Toutefois il est possible de modifier ce paramètre même après le lancement de la simulation. Pour des raisons de sécurité, ne modifier ce paramètre que lorsque vous avez stoppé la simulation. Vous pouvez en suite reprendre la simulation avec un fichier de sortie alternatif.



**Figure 5.17 :** *Affichage d'une simulation en cours*



**Figure 5.18 :** *Représentation graphique du résultat*

## 5.7 Comparaison du résultat de l'analyse des algorithmes de localisation

Notons que dans le cadre de cette analyse chaque algorithme est analysé indépendamment.

Sur la figure 5.18 on peut constater l'intégralité des résultats de chaque algorithme de localisation.

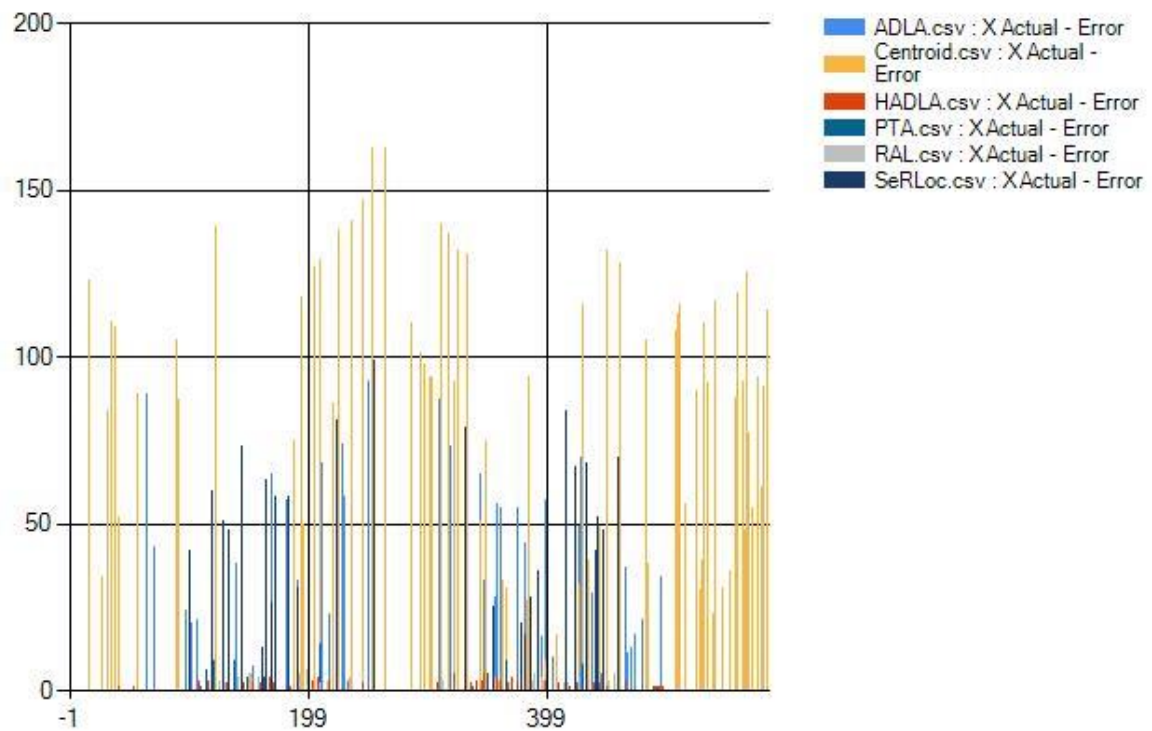
Chaque barre représente un nœud du réseau. On a constaté que plus le nombre de nœud dont la position est estimé est élevé plus le taux d'erreur d'estimation de position augmente.

Nous avons pris comme critère de choix le taux d'erreur minimal. Dans cette optique nous avons pu isoler 2 algorithmes qui ont un taux d'erreur inférieur à 10.

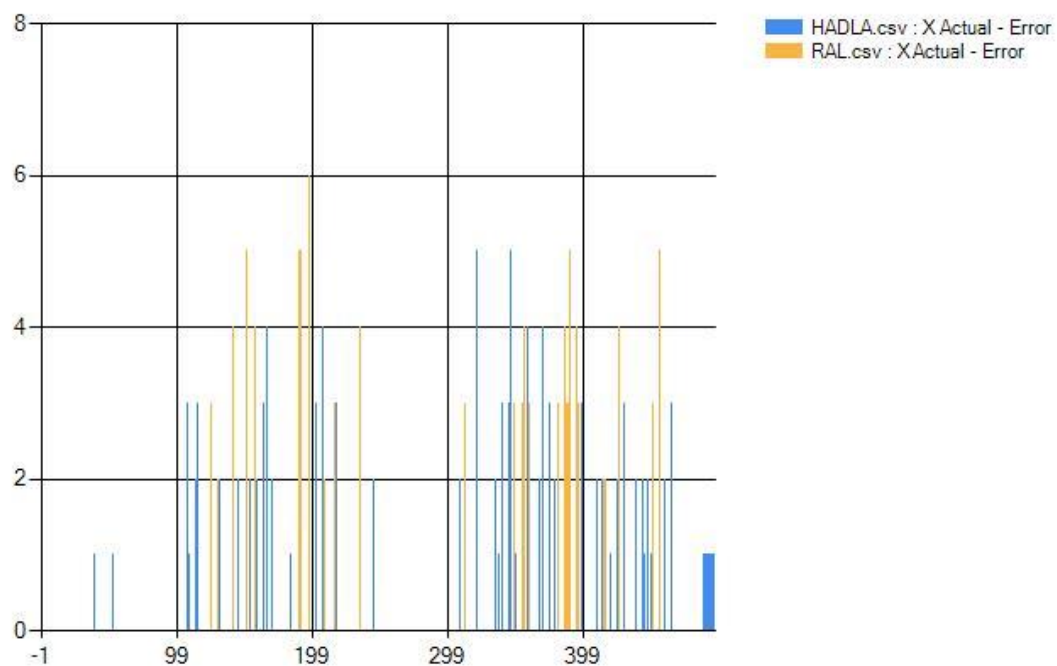
Après avoir consulté les fichiers de sorties voilà le résumé de chaque algorithme :

- **RAL**
  - Nombre de nœuds : 49
  - Erreur moyenne de localisation : 3
- **HADLA**
  - Nombre de nœuds : 82
  - Erreur moyenne de localisation : 1

En voyant ces chiffres nous pouvons en déduire que l'algorithme de localisation le plus efficace est le HDLA.

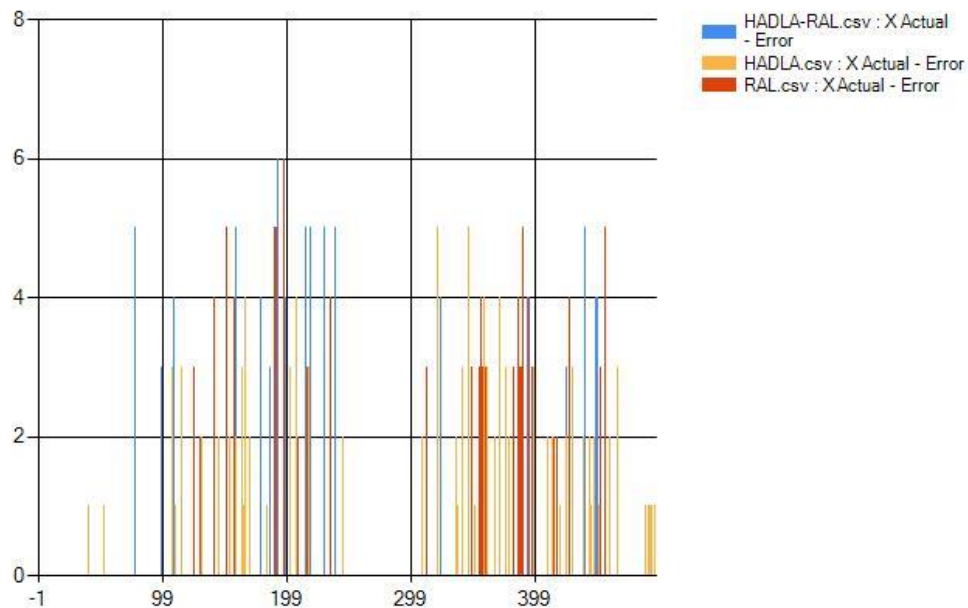


**Figure 5.19 :** *Représentation des résultats de chaque algorithmes de localisations*



**Figure 5.20 :** *Représentation des résultats des 2 algorithmes les plus efficaces*

Pour voir l'efficacité de la combinaison des algorithmes de localisation nous avons combiné les 2 algorithmes aux meilleurs résultats que nous avons pu observer.



**Figure 5.21 :** Représentation des résultats des 2 algorithmes les plus efficaces et le résultat de l'algorithme hybride HADLA-RAL

Visuellement le résultat semble être identique au résultat de l'algorithme de RAL.

### **RAL - HADLA**

- Nombre de nœuds : 37
- Erreur moyenne de localisation : 3

Nous pouvons ainsi voir que cette combinaison a pris les défauts des 2 algorithmes. Ce résultat montre que le nombre de nœuds pris en charge a fortement diminué même par rapport au nombre de nœuds pris en charge par l'algorithme HDLA et il a aussi gardé le même taux d'erreur que l'algorithme RAL.

## **5.8 Conclusion**

D'une part, le simulateur que nous avons développé est en mesure d'effectuer une simulation de localisation de nœuds dans un réseau de capteurs sans fils. Dans ce cadre, nous avons choisi de mettre en évidence les fonctionnalités des éléments électroniques en y ajoutant un facteur environnemental pour rendre le résultat le plus proche du réel possible.

D'autre part, le choix de l'algorithme de localisation se fait en se basant sur les critères que vous choisirez, soit selon sa précision soit selon le nombre de nœuds pouvant être localisé. Dans le chapitre 3 nous avons présumé qu'en générale la combinaison des algorithmes de localisation tend à améliorer leurs performances, mais nous avons constaté que cela n'est pas toujours le cas.

## CONCLUSION

A différentes échelles, les réseaux se sont intégrés à la vie humaine. Hier, le LAN, le MAN et le WAN s'ingéniaient à nous interconnecter. Aujourd'hui, les PAN s'intéressent au confort, à la protection et au service à la personne. D'un côté, par les capacités de mesures de paramètres environnementaux embarquées, les nœuds de ces infrastructures sont capables de suivre l'évolution tant de la température d'une pièce que l'état de santé d'un malade ou encore l'évolution d'un incident de type « marée noire ». Un complément à la richesse de ces offres est la localisation. Bien qu'il s'agisse d'une problématique aussi ancienne que le concept de mobilité, la détermination de la position d'un élément donné reste un sujet d'étude intéressant. Spécifiquement, lorsque ce service doit être rendu par un réseau de capteurs sans fil, l'investigation devient passionnante.

L'étude réalisée au cours de ce mémoire porte sur deux volets principaux. La première comporte une analyse théorique du fonctionnement des codages dans un système de MIMO. Il est montré que les codages STTC sont les plus fiables mais nécessitent des capacités de calcul élevées donc consommant plus d'énergie; tandis que les codages STBC sont simples à décoder tout en gardant un certain niveau de qualité de service. Le second porte sur l'analyse des techniques de localisation dans un réseau et tout particulièrement des réseaux composés de plusieurs nœuds.

Les algorithmes exploités pour la localisation vont de la simple application de règles de géométrie au filtrage du signal pour l'élimination des erreurs. Ces concepts sont résumés ainsi que les exigences liées à leur utilisation. La diversité des pistes, tant sur le signal à retenir que dans la manière de manipuler les informations, nous a porté à redéfinir notre problématique comme la recherche d'une solution d'auto-localisation range-based pour réseau de capteurs sans fil. A partir de cette formulation, nous avons proposé différentes stratégies dont les principes mais également les possibles débouchés et perspectives sont introduits dans les paragraphes suivants.



## ANNEXE 1. INFORMATIONS DISPONIBLES POUR LA LOCALISATION

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

### A1.01. Puissance du signal radio

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

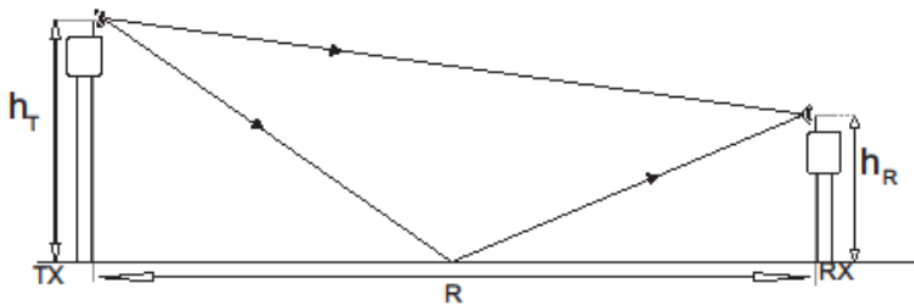
#### a. Equation de Friis

$$\frac{P_r}{P_t} = G_t \times G_r \times \frac{1}{L} \times \left( \frac{\lambda}{4\pi R} \right)^2 \quad (\text{A1.01})$$

Les bilans de liaison utilisent généralement le modèle de l'équation A.01 pour prédire le niveau de puissance reçue  $P_r$  à une certaine distance  $R$  (de l'émetteur au récepteur) connaissant la puissance émise  $P_t$ . Le milieu traversé est caractérisé par les pertes  $L$ . Ces pertes sont liées à la fréquence du signal dont la longueur d'onde  $\lambda$  intervient dans la formule. Finalement, les gains des antennes d'émission et de réception utilisées,  $G_t$  et  $G_r$ , entrent en ligne de compte. Ce modèle fait donc l'hypothèse d'une répartition uniforme de l'énergie sur des sphères concentriques.

#### b. Propagation avec réflexion sur le sol ou Two Ray Ground reflection model

Si l'équation de Friis est efficace dans le cadre des liaisons satellitaires, d'autres modèles sont nécessaires lorsqu'on s'intéresse à des communications proches du sol. Le Two Ray Ground Reflection model considère que le signal suit deux chemins principaux (figure A1.01) pour arriver au récepteur. Le premier est direct et le second réfléchi par le sol.



**Figure A1.01.** Propagation avec une réflexion

Le passage du modèle Free Space au modèle Two Ray ground reflection dépend de l'espacement  $R$  entre les antennes : entre 0 et la distance critique  $D_{Com}$ , le modèle de propagation en espace libre s'applique. Au-delà, le modèle avec réflexion au sol s'utilise.

### *c. Modèle de Shadowing*

Ce modèle comporte deux parties :

- Le modèle d'atténuation : une distance de référence  $d_0$  est définie et un niveau de puissance reçue lui est associé. La puissance reçue à une distance  $d$  quelconque est liée à ces informations.
- La variation de la puissance à une distance donnée : en un point donné, la puissance reçue varie au cours du temps. Une loi log-normale XdB a été retenue pour représenter cette variation.

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

### *d. Niveau de puissance radio reçue ou RSSI*

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

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

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

D'un autre côté, cette information est très sensible à l'environnement : le corps humain, du fait de sa grande teneur en eau, atténue les signaux HF qui le traversent. Ceci aura un impact en déploiement réel sur les solutions proposées pour des technologies exploitant la bande ISM 2.4GHz. Les positions relatives des antennes ont également une influence sur les niveaux de puissance détectés. Les informations brutes ne sont par conséquent pas directement exploitables : une étape de profilage,

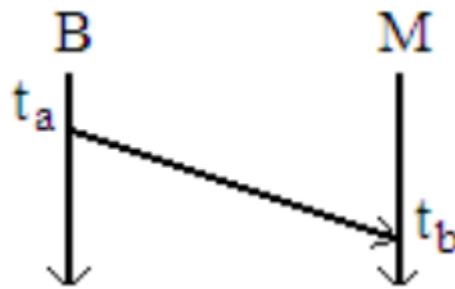
c'est-à-dire de constitution d'un cadre de référence à partir de données collectées dans l'environnement visé, doit donc être ajoutée et constitue un coût supplémentaire.

### A1.02. Temps de vol

Les informations temporelles ont la réputation d'être plus fiables que le niveau de puissance reçue : la vitesse de propagation du signal dans l'air étant connue, il suffit d'appliquer la formule (équation A1.02) pour obtenir la distance.

$$distance = vitesse \times temps \quad (A1.02)$$

Dans un réseau, l'échange de messages entre les équipements est un moyen d'estimer ce temps de vol. Dans le cas d'un réseau synchronisé, un seul message suffit. La figure A1.02 illustre cette situation : un message de B (comme Balise) vers M (comme Mobile) voit ses instants d'émission et de réception,  $t_a$  et  $t_b$ , capturés par les deux nœuds. Une fois les informations réunies au niveau de B, par une autre transmission de M vers B, le temps de vol ou TOF peut être estimé (équation A1.03). D'autres protocoles permettant la mesure du temps de vol seront présentés dans la suite de ce document.



**Figure A1.02.**Temps de vol

$$t_{vol} = t_b + t_a \quad (A1.03)$$

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

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

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

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

#### **A1.03. Angle d'incidence**

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

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

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

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

#### **A1.04. Phase du signal**

La mesure de la phase du signal permet une évaluation très précise de la distance. Supposons que le signal de longueur d'onde  $\lambda$  soit une simple sinusoïde : si le dispositif de mesure renvoie un déphasage  $\theta$ , cela indique qu'il se trouve à une distance  $\lambda^*$  de la source, écart donné par l'équation A1.04. Ainsi, pour un signal de 1GHz, la longueur d'onde  $\lambda$  sera de 30cm :  $\theta$  prenant ses valeurs dans l'intervalle  $[0 ; 2\pi]$ ,  $\lambda^*$  sera compris entre 0 et 30cm.

$$\lambda^* = \frac{\lambda\theta}{2\pi} \quad (\text{A1.04})$$

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

Il faut noter que cette technique ne peut s'utiliser seule à des distances supérieures à  $\lambda$ . En raison des caractéristiques de la mesure des angles, il est impossible de différencier  $\theta$  et  $\theta+k2\pi$ ,  $k \in \mathbb{N}$ . Deux pistes existent : associer une autre technologie permettant de se placer en un point de l'espace

pour ensuite affiner la mesure, ou travailler à basse fréquence. Dans ce dernier cas de figure, la dimension de la longueur d'onde en basse fréquence est exploitée. Dans la bande LF (Low Frequency) allant de 30kHz à 300kHz, la longueur d'onde sera comprise entre 1 km et 10 km, ce qui est largement suffisant pour se localiser dans un bâtiment ordinaire.

#### **A1.05. Nombre de sauts**

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

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

#### **A1.06. Autres paramètres**

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

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

## BIBLIOGRAPHIE

- [1] A. F. Molisch et M. Z. Win, "*MIMO systems with Antenna Selection*", IEEE microwave magazine, Mars 2004
- [2] J. Li, G. Bosisio, K. Wu, "*Computer and measurement simulation of a new digital receiver operating directly at millimeter-wave frequencies*", IEEE Trans. on MTT, Vol. 43, No. 12, December 1995
- [3] J. Hesselbarth, F. Wiedmann, B. Huyart, "*Two new Six-port reflectometers covering very large bandwidths*", IEEE Trans. On Instrumentation and Measurement, Vol. 46, No. 4, August 1997
- [4] M. F. Siyau, "*Channel estimation for layered space-time systems*", 4th IEEE Workshop on SPAWC. 2003
- [5] A. J; Paulraj, "*Space-time modems for wireless personal communications*", IEEE Personal Communications, February 1998
- [6] B. Widrow, P. E. Mantey, L. J. Griffiths, P. B. Goode, "*Adaptive Antennas Systems*", Proc. IEEE, vol. 55, 1967.
- [7] M. Heusse, "*Réseaux locaux sans fil*", 2009
- [8] T. Beluch, D. Dragomirescu, F. Perget et Plana, "*Cross-layered synchronization protocol for wireless sensor networks. In Ninth International Conference on Networks (ICN)*", 2010
- [9] S. El Abdellaoui, "*Etude et analyse de la performance du système coopératif «MIMO-Virtuel» pour les réseaux de capteurs sans fil*", thèse doctorale, LRIT-CNRST (URAC-29), septembre 2013
- [10] K. Mabrouk, "*Conception et réalisation d'un système de Télécommunications MIMO avec Formation Numérique de Faisceaux en réception ; Calibrage aveugle du Démodulateur triphasé Zéro-IF et comparaison au démodulateur classique à 2 voies I et Q*", thèse doctorale, ENST Paris, 2008
- [11] A. Perrier, "*Développement Et Analyse De Performance D'algorithmes De Type Gradient Stochastique*", thèse doctorale à l'ENST Paris 1995
- [12] D. Le Ruyet et B. Ozbek, "*Systèmes MIMO et codage spatio-temporel*", Revue de l'électricité et de l'électronique, (4), 2005.
- [13] V. Tarokh, H. Jafarkhani, et A. R. Calderbank, "*Space-time block coding for wireless communications: performance results*", IEEE Journal on Selected Areas in Communications, Mars 1999.
- [14] V. Tarokh, N. Seshadri, et A. R. Calderbank, "*Space-Time Codes for High Data Rate Wireless Communication: Performance Criterion and Code Construction*", IEEE Transactions On Information Theory, vol. 44, Mars 1998.
- [15] S. Alamouti, "*Space-Time block coding: A simple transmitter diversity technique for wireless communications*", IEEE Journal On Select Areas In Communications, vol. 16, Octobre 1998.

- [16] V. Tarokh, H. Jafarkhani, et R. A. Calderbank, "*Space-Time block codes from orthogonal designs*", IEEE Transactions on Information Theory, vol. 45, pp. 1456–1467, July 1999.
- [17] O. Tirkkonen et A. Hottinen, "*Square-matrix embeddable space-time block codes for complex signal constellations*", IEEE Transactions on Information Theory, vol. 48, Février 2002.
- [18] R. Piechocki, P. Fletcher, A. Nix, N. Canagarajah, et J. McGeehan, "*A measurement based feasibility study of space-frequency MIMO detection and decoding techniques for next generation wireless LANs*", IEEE Transactions on Consumer Electronics, Aout 2002.
- [19] E. Cassano, F. Florio, F. de Rango, S. Marano, "*A Performance Comparison between ROC-RSSI and Trilateration Localization Techniques for WPAN Sensor Networks in Real Outdoor Testbed*", Wireless Telecommunications Symposium, juin 2009
- [20] Y. Gwon, R. Jain, T. Kawahara, "*Robust Indoor Location Estimation of Stationary and Mobile Users*", Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies, mars 2004
- [21] F. Izquierdo, M. Ciurana, F. Barcelo, J. Paradells, E. Zola, "*Performance evaluation of a TOA-based trilateration method to locate terminals in WLAN*", Paper presented at the 1st International Symposium on Wireless Pervasive Computing. 16-18 Janvier 2006
- [22] S. Lanzisera, D. T. Lin, K. S. Pister, "*RF Time of Flight Ranging for Wireless Sensor Network Localization*", Workshop on Intelligent Solutions in Embedded Systems (WISES), Juin 2006
- [23] L. Gui, "*Amélioration de la Localisation dans les Réseaux de Capteurs sans Fil par Méthodes "Range-free"*", Thèse de doctorat, INSA Toulouse - EDSYS - Laboratoires LATTIS puis IRIT, février 2013.
- [24] F. Evennou, "*Techniques et technologies de localisation avancées pour terminaux mobiles dans les environnements indoor*", Ph.D. Dissertation, Université Joseph Fourier, Grenoble, 2007

## **FICHE DE RENSEIGNEMENT**

**Nom :** RADIASON

**Prénoms :** Tsiry Andriamampianina

**Adresse :** Lot IPN 48 bis Ambanilalana Itaosy

Antananarivo 102

**Téléphone :** +261 33 25 432 41

**E-mail :** rtsiry@live.fr



**Titre du mémoire:**

***SIMULATION D'UN RESEAU DE CAPTEURS SANS FILS ET EVALUATION DES  
ALGORITHMES DE LOCALISATIONS***

**Nombre de pages :** 114

**Nombre de tableaux :** 4

**Nombres de figures :** 75

**Directeur de mémoire :** M. ANDRIAMANALINA Ando Nirina

**Grade :** Maitre de conférences

**Téléphone :** +261 34 03 876 01

**E-mail :** andonirinabe@gmail.com



## RESUME

L'objectif étant de permettre de sélectionner la configuration optimale pour effectuer des localisations dans un réseau de capteurs sans fils. Nous avons opté d'utiliser des algorithmes de calculs entre les nœuds au lieu d'utiliser directement un dispositif de GPS. Nous avons divisé le travail en plusieurs étapes. La première est celle d'une vue d'ensemble des environnements de transmission et des réseaux de capteurs sans fils. Par la suite nous avons effectué une comparaison des techniques de transmission pour mettre en évidence l'utilité des systèmes MIMO-Coopératif. Nous avons aussi effectué une évaluation des codages STTC et STBC utilisés pour la transmission. Puis nous avons présenté les théories selon lesquelles on effectue le calcul de position dans un système d'émetteur récepteur. Et nous avons terminé notre étude en réalisant un simulateur combinant les éléments appris dans les chapitres précédant afin de faire une évaluation des techniques de localisation généralement utilisées. Nous avons pu constater dans cet ouvrage que la combinaison des algorithmes de localisation n'améliore pas forcément les résultats des calculs.

**Mots clés :** Réseau, Capteurs sans fils, Localisation, Codage STTC - STBC.

## ABSTRACT

The goal is to allow you to select the optimal configuration for performing locations in a wireless sensor network. We choose to use algorithms calculated between nodes, instead of directly using a GPS device. We divided the work in stages. The first is an overview of transmission environments and without wireless sensor networks. Subsequently we conducted a comparison of transmission techniques to highlight the usefulness of Cooperative MIMO systems. We also conducted an assessment of STBC and STTC codes used for transmission. Then we presented the theories which are carried out to calculate the position in a transceiver system. And we finished our study by conducting a simulator combining elements learned in the preceding chapters to make an assessment of localization techniques generally used. We have seen in this book that the combination of location algorithms does not necessarily improve the results of calculations.

**Keywords :** Network, Wireless sensor, Localization, STTC – STBC coding.