



# **L'internet des objets pour les régions rurales et éloignées**

**Mémoire**

**Francis Lehoux**

**Maîtrise en génie électrique - avec mémoire**  
Maître ès sciences (M. Sc.)

Québec, Canada

# **L'internet des objets pour les régions rurales et éloignées**

**Mémoire**

**Francis Lehoux**

Sous la direction de

Paul Fortier, directeur de recherche  
Benoit Gosselin, codirecteur de recherche

# Résumé

Cet ouvrage vise l'étude d'une technologie visant à optimiser la communication sans fil entre des objets dans un environnement rural ou éloigné. On y fait une comparaison des technologies LPWAN pour ensuite sélectionner Symphony Link par Link Labs qui n'est visée par très peu d'études. Un prototype est conçu afin de mesurer plusieurs paramètres de transmission dont l'énergie consommée, le nombre de paquets retransmis et la puissance à la réception. Un modèle d'atténuation de l'environnement rural de l'expérience est établi. Il s'en suit l'élaboration d'un modèle de simulation illustrant le comportement d'un réseau comportant plusieurs de ces nœuds. On y analyse les résultats et approxime des équations visant entre autres le taux de réception total, l'efficacité spectrale et l'efficacité énergétique. Finalement, ces approximations sont utilisées dans des exemples de déploiement visant l'agriculture et le Grand Nord, représentant un milieu éloigné.

# Abstract

This work is on a technology that optimizes wireless communications between objects operating in a remote or rural environment. We compare different LPWAN technologies. Then we select Symphony Link by Link Labs that is the subject to very few studies. A prototype is made to monitor transmission parameters like the energy consumed, the number of packets that has to be resent and the power at reception. An attenuation model of the rural environment of the experiment is made. After, a simulation model is built to illustrate the behavior of a network made of many of these nodes. We analyze and work on equations that approximate, among others, the total reception ratio, the spectral efficiency and the energy efficiency. Finally, these approximations are used in deployment examples on agricultural and remote areas.

# Table des matières

Résumé	iii
Abstract	iv
Table des matières	v
Liste des tableaux	vi
Liste des figures	vii
Acronymes	ix
Remerciements	xi
Introduction	1
<b>1 Technologies et concepts</b>	<b>5</b>
1.1 LPWAN . . . . .	5
1.2 Symphony Link™ . . . . .	9
1.3 Paramètres d'évaluation . . . . .	13
<b>2 Expérimentation terrain</b>	<b>17</b>
2.1 Système pour expérimentation . . . . .	17
2.2 Prototype du module radio . . . . .	22
2.3 Résultats . . . . .	27
<b>3 Simulation</b>	<b>32</b>
3.1 Modèle . . . . .	32
3.2 Résultats . . . . .	36
<b>4 Exemples de déploiement</b>	<b>53</b>
4.1 Agriculture . . . . .	53
4.2 Grand Nord . . . . .	56
Conclusion	60
A Code de récupération de données	63
B Code du prototype	68



# Liste des tableaux

1.1	Tableau comparatif des différentes technologies LPWAN listées dans ce document [5], [9], [2]. . . . .	8
1.2	Proportion du débit de données et de temps de transmission selon SF. . . . .	11
2.1	Coefficient de puissance d'atténuation selon l'environnement. . . . .	31
3.1	SNR minimal pour réception selon le <i>spreading factor</i> . . . . .	41
3.2	Taux de collision de la section 1 et 2 selon le débit de données. . . . .	44
4.1	Paramètres de déploiement du scénario d'agriculture. . . . .	54
4.2	Taux de réception total selon le nombre de capteurs. . . . .	55
4.3	Densité des noeuds selon le rayon de déploiement. . . . .	57
4.4	Taux de réception estimé selon le rayon de déploiement. . . . .	58
4.5	Efficacité spectrale estimée selon le rayon de déploiement. . . . .	59
4.6	Efficacité énergétique estimée selon le rayon de déploiement. . . . .	59

# Liste des figures

1.1	Représentation d'un <i>chirp</i> dans le temps [6]. . . . .	10
1.2	Fenêtres de transmission et divisions. . . . .	12
1.3	Paquet divisé en sous-paquets illustré par le courant tiré par le module radio. . . . .	13
2.1	Structure du montage expérimental. . . . .	18
2.2	Position du noeud lors des prises de mesures. . . . .	19
2.3	Position de la station de base. . . . .	20
2.4	Schéma-bloc de la version 1 du module radio. . . . .	22
2.5	Courant mesuré par le système dans le prototype version 1. . . . .	23
2.6	Prototype version 1 avec les modules identifiés. . . . .	23
2.7	Schéma-bloc du prototype du module radio. . . . .	24
2.8	Photo du prototype version 2. . . . .	24
2.9	Courant mesuré par le système dans le prototype version 2. . . . .	25
2.10	Photo du prototype version 3. . . . .	25
2.11	Mesure de courant du prototype version 3 sans moyenne mobile. . . . .	26
2.12	Mesure de courant du prototype version 3 avec moyenne mobile. . . . .	26
2.13	Puissance de transmission selon le courant (module LL-RXR-27). . . . .	27
2.14	Puissance de transmission selon le courant (module LL-RLP-20). . . . .	28
2.15	Carte des emplacements des mesures. . . . .	29
2.16	Détermination du coefficient de puissance d'atténuation par la régression linéaire du <i>path loss</i> et du logarithme de la distance. . . . .	30
2.17	Variation du <i>path loss</i> par rapport à sa moyenne. . . . .	31
3.1	Fonctionnement du modèle de simulation. . . . .	33
3.2	Exemples de placements pour 500 noeuds. . . . .	35
3.3	Exemples de placements pour 5000 noeuds. . . . .	36
3.4	Calcul des paramètres de transmission. . . . .	36
3.5	Puissance de transmission moyenne selon le rayon de déploiement et le débit de données. . . . .	37
3.6	Distribution du <i>spreading factor</i> selon le rayon de déploiement. . . . .	38
3.7	Aire de chacune des valeurs de SF. . . . .	39
3.8	Temps de transmission selon le débit de données et le rayon de déploiement. . . . .	41
3.9	Approximation du temps de transmission avec un débit de 3200 octets/min selon le rayon de déploiement. . . . .	42
3.10	Pertes par atténuation selon le débit de données et le rayon de déploiement. . . . .	43
3.11	Taux de collisions selon le débit de données et le rayon de déploiement. . . . .	44



3.12	Approximations du taux de collisions selon le rayon de déploiement et le débit de données. . . . .	48
3.13	Taux de réception total selon le débit de données et le rayon de déploiement. . .	48
3.14	Taux de réception total selon le débit de données et le rayon de déploiement. . .	49
3.15	Efficacité énergétique selon le débit de données et le rayon de déploiement. . . .	49
3.16	Approximation de l'efficacité énergétique selon le rayon de déploiement. . . . .	50
3.17	Efficacité spectrale selon le débit de données et le rayon de déploiement. . . . .	51
3.18	Efficacité spectrale selon le débit de données et le rayon de déploiement. . . . .	51
4.1	Système EcoChip et ses lieux de déploiement. . . . .	57

# Acronymes

16-QAM	16-Quadrature Amplitude Modulation.
3GPP	Third Generation Partnership Project.
API	Application Programming Interface.
BPSK	Binary Phase Shift Keying.
CDMA	Code Division Multiple Access.
CSS	Chirp Spread Spectrum.
DBPSK	Differential Binary Phase Shift Keying.
DSSS	Direct Sequence Spread Spectrum.
eDRx	extended Discontinuous Reception.
eMTC	Enhancements for Machine Type Communications.
GMSK	Gaussian Minimum Shift Keying.
GSM	Global System for Mobile Communications.
IoT	Internet of Things.
ISM	Industrial, Scientific, Medical.
ITU	International Telecommunication Union.
LPWAN	Low-Power Wide-Area Network.
LTE	Long Term Evolution.
PSM	Power Saving Mode.
QPSK	Quadrature Phase Shift Keying.

RPMA	Random Phase Multiple Access.
RSSI	Relative Signal Strength Indication.
SF	Spreading Factor.
SNR	Signal-to-Noise Ratio.
TDOA	Time Difference Of Arrival.

# Remerciements

Ce travail n'aurait pas été possible sans la participation de Nutaq/Nuran et de l'organisme Mitacs.

# Introduction

L'Internet des objets (IoT) peut paraître comme un concept abstrait. Selon l'ITU (International Telecommunication Union), l'IoT est une infrastructure globale pour la société d'information, permettant des services spécialisés en connectant des choses ensemble, physiques ou virtuelles. Concrètement, il s'agit d'apporter une connectivité Internet aux différents objets de tous les jours afin de pouvoir les contrôler et les surveiller à distance. Le terme « objet » peut prendre une immense variété de formes, allant des véhicules, du grille-pain, du thermostat et autres pour l'usage domestique. Différents capteurs sont bien sûr inclus comme par exemple pour surveiller le taux d'humidité du sol en agriculture ou la disponibilité d'une espace de stationnement. Actuellement, un exemple domestique serait un thermostat intelligent qui est connecté à l'Internet, ce qui permet de le contrôler à distance avec un téléphone. Il s'adapte même à l'horaire de l'utilisateur par lui-même. Au Québec, la société d'état qui produit, transporte et distribue l'électricité, Hydro-Québec a annoncé en octobre 2019 qu'elle lançait sa filiale "Hilo" qui vise une gestion efficace de l'énergie. Cette gestion efficace passe par la connectivité de chaque appareil électrique afin qu'il puisse être monitoré et contrôlé individuellement, autant manuellement par l'utilisateur qu'automatiquement par un système centralisé. On pense par exemple à couper l'alimentation du chauffe-eau en période de pointe, réduire le chauffage et éteindre les lumières lorsqu'il n'y a personne à la maison. Pour les utilisateurs ayant des dispositifs pour produire et stocker de l'énergie comme des panneaux solaires avec des batteries, les services d'Hilo permettraient à Hydro-Québec d'acheter de l'électricité momentanément à ces utilisateurs afin de mieux répondre à la demande en période de haute consommation. Pour cet exemple et les autres objets mentionnés plus haut, ils sont reliés au réseau électrique.

En milieu rural ou éloigné, le réseau électrique n'est parfois pas accessible. Les objets mesurant et contrôlant leur environnement devraient être autonomes au niveau énergétique. Un premier exemple est en agriculture. En monitorant l'environnement précisément, il est possible d'augmenter considérablement l'efficacité de l'agriculture. Pour le sol, on peut surveiller l'humidité, l'acidité, la température. Certaines zones peuvent s'assécher plus rapidement, d'autres peuvent geler plus lentement. On peut ensuite agir aux endroits et aux moments précis afin de préserver les conditions optimales. D'autres paramètres comme la luminosité peuvent être monitorés mais on ne peut les influencer. Cela permet au moins de prédire le volume de récolte

pour mieux planifier la distribution par la suite.

Un deuxième exemple d'IoT en milieu éloigné se trouve dans [12] où on explique qu'une façon de mesurer les changements climatiques et l'impact de la présence humaine dans un environnement est de monitorer les microorganismes qui y vivent. Pour ce faire, un prototype a été développé prenant plusieurs mesures sur son environnement et aussi sur les microorganismes. Ces paramètres sont la température, l'humidité, la luminosité, l'acidité, le taux de dioxyde de carbone pour l'environnement et la bio-impédance pour les microorganismes. Les territoires nordiques sont visés puisque ce sont les premiers touchés par les changements climatiques et c'est à ces endroits que les effets sont les plus visibles. De plus, le microbiome de ces régions n'est pas bien connu. Lorsque le système est récupéré, il est possible d'en faire le séquençage afin d'identifier les microorganismes connus et d'étudier les inconnus. Pendant que le système est sur place, il communique de manière sans-fil avec une station de base permettant d'avoir des mesures régulièrement. Plus son autonomie est grande, plus longtemps le système peut fonctionner sans intervention humaine et plus de données peuvent être recueillies.

Il y a bien sûr plusieurs défis à relever dans le domaine [10]. Il existe présentement plusieurs protocoles de communication permettant aux objets de communiquer entre eux. On peut penser à Bluetooth, Wifi, ZigBee, etc. En ayant plusieurs langages différents, il peut devenir difficile de les faire tous communiquer ensemble. Il y a aussi l'enjeu de pouvoir stocker une quantité faramineuse de données, maintenant que chaque objet produit des données. La sécurité en préoccupe plus d'un. Comment peut-on sécuriser toutes ces connexions ? Quelqu'un de mal intentionné qui s'introduit dans votre réseau peut déclencher un incendie en démarrant le four ou mettre votre vie en danger en jouant avec votre stimulateur cardiaque ou votre pompe à insuline. Les données recueillies par tous ces capteurs appartiennent à l'utilisateur ou au fabricant de l'objet ? Le débat concernant la vie privée s'invite aussi.

Pour amener cette connectivité aux objets, il nous faut un lien de communication et ce lien dans la grande majorité des cas sera sans fil. Les protocoles de communication sans fil des dernières années sont pour la plupart optimisés au niveau du débit de données et de la latence au détriment de la distance et de la consommation d'énergie. Pensons par exemple aux réseaux cellulaires et WiFi. Pour plusieurs applications d'IoT, le débit de données n'a pas besoin d'être élevé et la latence n'est pas un enjeu. La consommation d'énergie oui, puisqu'on vise une autonomie de plusieurs années pour un capteur, ce qui peut paraître énorme en comparaison avec nos cellulaires que nous devons recharger quotidiennement. Cet aspect est d'autant plus important lorsqu'on aborde les régions rurales ou éloignées. La technologie LPWAN (*Low-Power Wide-Area Network*) vise l'optimisation de ces paramètres et sera étudiée dans cet ouvrage.

Dans le chapitre 1, la technologie LPWAN est décrite. Différents produits de sa catégorie sont comparés. Le candidat retenu est analysé afin d'expliquer son fonctionnement en profon-

leur, autant sa modulation de signal que son protocole de communication.

Au chapitre 2, on fait une expérimentation terrain de la technologie retenue dans un environnement rural. On établit la structure du système qui sera composée d'un serveur, d'une station de base et d'un nœud. Le nœud est un prototype développé pour cette expérimentation. Il permet de mesurer l'énergie consommée et le nombre de paquets devant être retransmis. Sa conception et ses itérations sont détaillées. Des mesures seront prises à différents endroits et une carte sera générée. Les données sont recueillies dans un serveur et ensuite analysées, ce qui permettra l'élaboration d'un modèle d'atténuation de ce milieu ainsi qu'un modèle de consommation d'énergie à la transmission de la technologie retenue.

Au chapitre 3, on simule un réseau à une station de base et plusieurs nœuds. On construit un modèle à partir d'un modèle existant [1] que l'on adapte à la technologie retenue et on établit ses paramètres en se basant sur les résultats obtenus dans notre expérimentation terrain. Les résultats sont ensuite présentés et analysés pour différents nombres de nœuds et différents rayons de déploiement. La puissance de transmission, la distribution du *spreading factor*, le temps de transmission, le taux de pertes par atténuation, le taux de pertes par collisions, le taux de réception total, l'efficacité spectrale et l'efficacité énergétique sont étudiés. Il en ressort des équations d'approximation afin de rapidement les évaluer selon le nombre de nœuds et le rayon de déploiement.

Au chapitre 4, on présente deux exemples de déploiement où on utilise les équations d'approximation du précédent chapitre. Le premier exemple est une situation en agriculture où il est requis de recueillir une quantité importante de données sur l'état du sol afin d'agir adéquatement pour optimiser les résultats de récolte. Le deuxième exemple est l'EcoChip [12] déployé dans le Grand Nord. L'emphase est alors mise sur la consommation d'énergie afin de minimiser l'intervention humaine dans ce milieu isolé et difficile d'accès.

Les contributions de ce mémoire se résument par :

- Une série de mesures qui :
  - détermine le coefficient de *pathloss* dans un environnement rural.
  - détermine la consommation d'énergie d'un appareil IoT commercial.
- Une série de simulations qui :
  - génère une série de géométries de déploiements aléatoires pour une surface de couverture fixe.
  - utilise un environnement de simulation publique en Python modifié pour calculer les mesures de performance paramétrées par les données recueillies sur le *pathloss* et la consommation d'énergie.
  - utilise MATLAB pour trouver une équation polynomiale de certaines mesures de performance.
- Une prédiction :

- pour deux scénarios simples de déploiement :
  - la superficie de couverture est fixée et il y a trois choix pour le nombre de noeuds.
  - le nombre de noeuds est fixé et quatre choix de superficie de couverture sont disponibles.
- de certains indicateurs de performance pour chaque environnement ciblé avec insertion des paramètres dans les équations polynomiales.



# Chapitre 1

## Technologies et concepts

Ce chapitre offre une description de la technologie utilisée dans ce travail et décrit ensuite les paramètres d'évaluation qui seront utilisés

### 1.1 LPWAN

LPWAN est l'acronyme de *Low-Power Wide-Area Network*. Cette technologie se caractérise par un faible débit de données sur une longue portée. Elle vise les applications nécessitant une longue durée de vie de batterie, qui sont peu coûteuses, qui ont une longue portée, mais dont la vitesse de transmission et la latence ne sont pas un enjeu [5].

La faible consommation d'énergie s'explique par le choix de topologie et la simplicité de la technologie, autant au niveau du matériel qu'au niveau du protocole.

La topologie en étoile est habituellement utilisée puisque chaque nœud a la portée nécessaire pour se connecter directement à la station de base. On évite ainsi le principal désavantage de la topologie en filet qui nécessite la retransmission du message par plusieurs nœuds, limitant la durée de vie du réseau à cause d'une plus grande consommation d'énergie.

Si certaines opérations complexes il y a, elles sont assurées par la station de base ou autres systèmes en amont, le but étant de minimiser et simplifier le plus possible les opérations effectuées par le nœud pour lequel la consommation d'énergie est critique. Du même coup, on diminue la nécessité de circuits complexes ce qui rend le nœud moins dispendieux.

Ci-dessous, on donne quelques exemples de protocoles s'inscrivant dans la catégorie Low-Power Wide-Area Network (LPWAN) [9].

### 1.1.1 Ingenu

Ingenu opère dans la bande de fréquences ISM 2.4 GHz. Elle utilise un protocole nommé RPMA avec une modulation DSSS qui est une variation du CDMA. Pour une fenêtre de transmission donnée, au lieu d'avoir tous les noeuds commençant à transmettre dès le départ, le début de la transmission de chaque noeud est déterminé aléatoirement à l'intérieur de la fenêtre, réduisant ainsi le chevauchement des transmissions. De plus, chaque transmetteur a son propre code de modulation, le rendant différentiable des autres même s'ils transmettent au même moment. La communication peut être bilatérale. Cependant, le lien descendant est en CDMA seulement.

### 1.1.2 Sigfox

SigFox opère dans la bande de fréquences libres ISM sub-GHz. Elle utilise une bande passante très étroite de l'ordre de 100 Hz pour un canal de communication et une modulation BPSK, ce qui a pour effet de minimiser le niveau de bruit et par conséquent d'augmenter la sensibilité du récepteur. Son débit maximal est de 100 bits/s, ce qui est très bas, même pour une technologie LPWAN [9]. Le lien communication est essentiellement montant, malgré qu'une quantité très limitée de données peut être transmise en descendant.

### 1.1.3 Weightless

On retrouve trois standards dans ce groupe. Chaque standard a ses caractéristiques changeant le rapport débit de donnée et consommation énergétique.

- Weightless-W : Utilise les fréquences libres dans les plages de télévision. Plusieurs modulations sont disponibles comme la modulation 16-QAM et DBPSK. L'accès aux fréquences de télévision dépend des réglementations en place. Ce standard possède le plus grand débit de données, mais aussi la plus grande consommation.
- Weightless-P : Utilise la bande ISM sub-GHz avec une modulation GMSK et QPSK. La communication est bilatérale.
- Weightless-N : Utilise la bande ISM sub-GHz avec une modulation DBPSK. Seulement la communication montante est possible. Il s'agit du standard le plus économique mais aussi le plus lent.

### 1.1.4 3GPP

3GPP est l'organisation qui établit les standards pour les réseaux cellulaires. Ces réseaux avec leurs courants standards ne conviennent pas pour les applications demandant une faible consommation et un faible coût. Ils sont optimisés afin de minimiser la latence et maximiser la fiabilité et le débit de données et cela amène une grande consommation d'énergie et

un coût monétaire plus important. Pour remédier à ce problème et entrer dans le LPWAN, l'organisation a développé de nouveaux standards.

- LTE eMTC ou CAT-M1 : Basé sur le LTE, ce standard a une largeur de bande moins grande (1,4 MHz) et son protocole est plus simple réduisant ses coût et sa consommation. Deux modes d'opérations sont aussi ajoutés afin de réduire la consommation d'énergie. PSM et eDRx font en sorte que le noeud peut être en veille sur une longue période de temps sans devoir échanger constamment avec la station de base afin de rester connecté.
- EC-GSM : Il s'agit d'une adaptation de la technologie GSM pour la communication entre objets. Comme il a été annoncé que le 2G ne sera plus supporté dans certaines régions, il sera possible de récupérer ces bandes de fréquences avec ce standard après une mise à jour logicielle.
- NB-IoT : Cette technologie peut être déployée avec le GSM, LTE ou seule. Elle utilise une largeur de bande de 180 kHz. Elle est aussi plus simple, le matériel nécessaire est moins complexe, mais le débit de données est plus bas que le LTE Cat-M1. Il est possible de l'intégrer à un réseau LTE avec une mise à jour logicielle.

### 1.1.5 LoRaWAN

LoRaWan opère dans la bande de fréquences libres ISM. La modulation utilisée est LoRa™, expliquée un peu plus bas dans la section 1.2.1. Le protocole utilisé est celui d'ALOHA. Les noeuds transmettent à n'importe quel moment en utilisant des canaux choisis aléatoirement et possiblement avec des codes orthogonaux entre eux. Une particularité de LoRaWAN est qu'un message peut être reçu par plusieurs station de bases, augmentant les chances de réception. Aussi, cela donne la possibilité de localiser le transmetteur par TDOA.

Pour la gestion de la communication descendante, on retrouve trois classes de noeuds.

- Classe A : Cette classe va être disponible pour recevoir des données seulement une courte période après avoir transmis.
- Classe B : En plus de la fenêtre décrite par la Classe A, le noeud sera disponible pour une réception à un intervalle donné.
- Classe C : Le noeud est toujours disponible pour recevoir de la station de base.

### 1.1.6 Tableau comparatif

Le tableau 1.1 compare les différentes technologies présentées précédemment.

TABLE 1.1 – Tableau comparatif des différentes technologies LPWAN listées dans ce document [5], [9], [2].

	Ingenu	Sigfox	Weightless			3GPP			LoRaWAN
			W	N	P	NB-IoT	LTE-M		
Modulation	RPMA-DSSS	UNB-DBPSK	16-QAM-BPSK-QPSK-DBPSK	UNB-DBPSK	GMSK-QPSK	UNB-GFSK-BPSK	OFDMA	CSS	
Débit	78 kbps	100 bps	1 kbps à 10 Mbps	30 kbps à 100 kbps	200 bps à 100 kbps	0.3 à 4000 kbps	200 kbps à 1 Mbps	290 bps à 50 kbps	
Largeur de bande	1 MHz	100 Hz	-	200 Hz	12.5 kHz	3.75 kHz à 20	1.08 MHz	125 ou 250 kHz	
Portée (link budget)	168 dB	146 dB	-	-	-	151 dB	155.7 dB	154 dB	
Bande de fréquences	ISM 2.4 GHz	Sub-GHz ISM	TV white spaces	Sub-GHz ISM	Sub-GHz ISM	LTE range	LTE range	Sub-GHz ISM	

## 1.2 Symphony Link™

Symphony Link™ est la technologie choisie afin de réaliser le projet. Il s'agit d'une technologie IoT de type LPWAN créée par Link Labs [7]. C'est une solution partant des modules radio sur le terrain jusqu'au serveur recueillant les données. Le développement nécessaire est donc minimal et plus de temps peut être consacré à l'étude du réseau. Cette technologie a été l'objet de peu ou pas d'étude jusqu'à présent.

### 1.2.1 Modulation

La modulation utilisée se nomme LoRa™, une modulation à large bande combinant la DSSS et la CSS [11].

#### Le théorème de Shannon-Hartley

Ce théorème définit la limite maximale théorique du débit de données selon une bande passante et un ratio signal à bruit donné. Il s'agit du principe du base derrière la modulation DSSS.

$$C = B * \log_2\left(1 + \frac{S}{N}\right) \quad (1.1)$$

$C$  : Capacité du lien de communication (bit/s)

$B$  : Bande passante (Hz)

$S$  : Puissance du signal reçu (W)

$N$  : Puissance du bruit ambiant et de l'interférence (W)

On constate que pour un même débit de données, si le rapport signal à bruit diminue, il suffit d'augmenter la bande passante pour compenser.

#### DSSS

La modulation DSSS consiste à multiplier les données par un code (*chip*) prédéterminé à une fréquence plus élevée. On obtient alors une largeur de bande plus élevée, mais pour un même débit de données. Du théorème de Shannon-Hartley on peut tirer :

$$G_p = 10 * \log_{10}\left(\frac{R_c}{R_b}\right)(\text{dB}) \quad (1.2)$$

$G_p$  : Gain de traitement (dB)

$R_c$  : Vitesse du *chip* (chips/seconde)

$R_b$  : Débit de données (bits/seconde)

## CSS

Cette technologie a été initialement breveté en 1944 pour être utilisé dans les radars. Un *chirp* est une variation constante d'une fréquence initiale vers une fréquence finale dans un intervalle de temps  $T$  (voir figure 1.1). La fréquence peut augmenter comme elle peut diminuer. La différence entre la fréquence de départ et la fréquence de fin est une bonne approximation de la largeur de bande. Cette technologie se caractérise par une grande robustesse due à

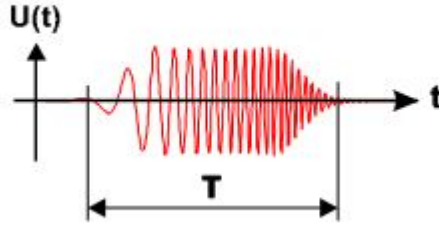


FIGURE 1.1 – Représentation d'un *chirp* dans le temps [6].

son haut produit largeur de bande et période. Elle présente aussi une excellente résistance à l'effet *multipath* grâce à sa large bande. Elle peut aussi être implémentée avec un circuit analogique, réduisant la consommation d'énergie. Elle est aussi résistante à l'effet Doppler et à l'interférence intra-bande [6].

## LoRa™

La modulation Lora™ combine les technologies DSSS et CSS, additionnant leurs avantages. Le débit données est défini par :

$$R_b = SF * \frac{1}{\frac{2^{SF}}{BW}} \quad (1.3)$$

$R_b$  : Débit de données (bits/sec)

$SF$  : *Spreading Factor* (7, 8, 9, 10, 11, 12)

$BW$  : Largeur de bande (Hz)

LoRa™ comprend aussi une correction d'erreurs variable qui améliore la robustesse du signal, mais réduit le débit de données.

$$CEV = \frac{4}{4 + CR} \quad (1.4)$$

$CEV$  : Facteur pour déterminer le débit de données effectif dû au facteur de correction.

$CR$  : Facteur de correction (1, 2, 3, 4).

On peut l'intégrer à l'équation 1.3.

$$R_b = SF * \frac{4}{\frac{4+CR}{2^{SF}} BW} \quad (1.5)$$

Dans le cas où LoRa™ est utilisé par Symphony Link™, le *spreading factor* est limité de 7 à 10, le facteur de correction est fixé à 1 et la largeur de bande est fixée à 125 kHz. On peut donc simplifier l'équation et la ramener en bytes par seconde puisque les quantités de données seront exprimées ainsi.

$$R_B = \frac{1}{8} * SF * \frac{4}{\frac{4+CR}{2^{SF} BW}} \quad (1.6)$$

$$R_B = \frac{1}{8} * SF * \frac{4}{\frac{4+1}{2^{SF} 125000}} \quad (1.7)$$

$$R_B = \frac{SF * 12500}{2^{SF}} \quad (1.8)$$

$R_B$  : Débit de données (bytes/sec)

Ce débit de données est celui à la couche physique de la communication, c'est-à-dire qu'il inclut autant le contrôle que l'information. Cette équation ne peut donc pas être utilisée directement pour calculer le débit de données final. Elle peut cependant servir à établir une proportion entre les débits de données selon le *spreading factor*. Le tableau 1.2 présente ces proportions avec comme référence un *spreading factor* de 7.

TABLE 1.2 – Proportion du débit de données et de temps de transmission selon SF.

	SF			
	7	8	9	10
$R_B$	$x$	$0.5714x$	$0.3214x$	$0.1786x$
$T_{Tx}$	$t$	$1.75t$	$3.11t$	$5.60t$

### 1.2.2 Protocole

Afin d'expliquer le protocole de Symphony Link™, un cycle complet de transmission sera donné, d'abord du point de vue de la station de base et ensuite du noeud.

#### Station de base

Lorsqu'une station de base est activée, elle effectue une analyse des plages de fréquences disponibles et sélectionne un canal de 500 kHz pour ses transmissions descendantes, c'est-à-dire vers les noeuds. Un choix de canal sera rejeté si la station de base y voit de l'interférence ou qu'une autre station de base l'utilise.

Une fois le canal de communication descendant sélectionné, la station de base transmet à toutes les deux secondes de l'information sur le réseau créant des fenêtres de transmission. On

y retrouve d’abord l’identification du réseau et l’identification de l’application. Ensuite comme LoRa™ ne permet pas simultanément une communication descendante et ascendante, la station de base va transmettre comment la prochaine fenêtre de transmission sera divisée, évitant les collisions ascendantes-descendantes. La figure 1.2 illustre les fenêtres de transmission et leurs divisions.

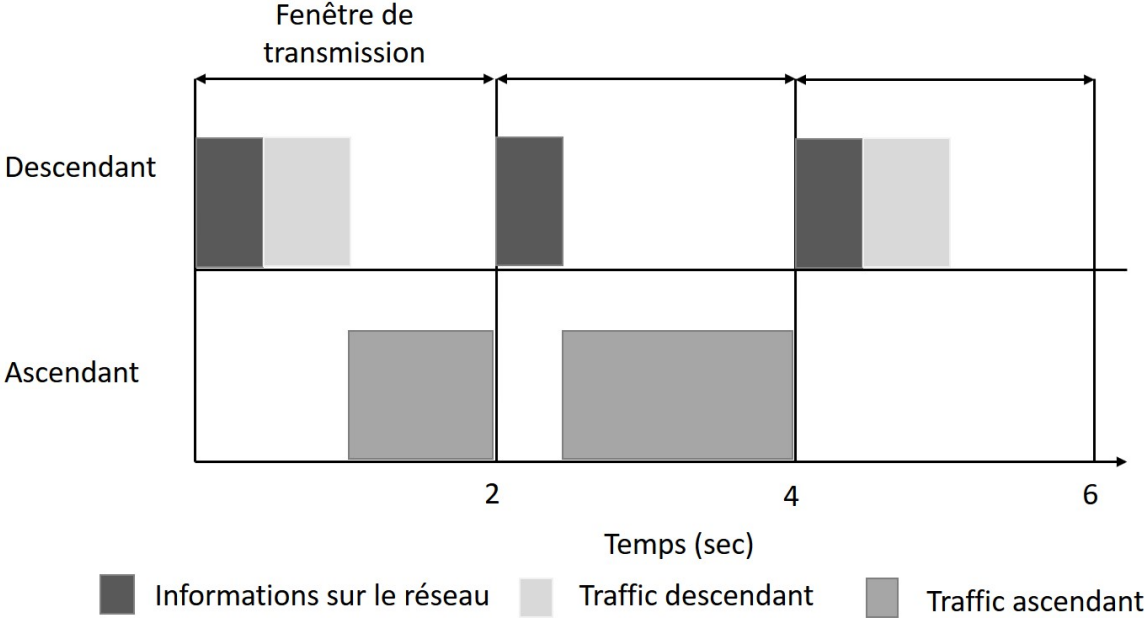


FIGURE 1.2 – Fenêtres de transmission et divisions.

Un autre morceau faisant partie des informations transmises au début de chaque fenêtre de transmission est la position des canaux qui seront disponibles pour des transmission ascendantes. Ces canaux changent aléatoirement entre chaque fenêtre de transmission. Les noeuds doivent savoir quels canaux utiliser pour transmettre leurs données. Dans notre cas, la station de base a huit de ces canaux. Ensuite, on retrouve le niveau de qualité de service qui donne priorité aux noeuds les plus importants s’il y a une congestion sur le réseau et les confirmations compressées des messages reçus lors de la dernière fenêtre de transmission. Finalement, après chaque série de huit fenêtres de transmission, des informations additionnelles sont transmises, dont la qualité du lien sans-fil avec les noeuds, permettant à ces derniers de s’adapter si besoin est.

**Noeuds**

Le noeud est d’abord en veille jusqu’à qu’il ait besoin de transmettre. À ce moment, il se réveille et détecte l’information envoyée par la station de base au début d’une fenêtre de transmission. Il va ajuster sa puissance de transmission pour minimiser sa consommation d’énergie et ajuster son *spreading factor* si la puissance maximale de transmission n’est pas



suffisante pour atteindre la station de base. Ces paramètres sont ajustés toutes les huit fenêtres de transmission avec l'information fournie par la station de base. Le noeud va aussi choisir une fréquence au hasard parmi celles données par la station de base.

Le paquet de données envoyé a une taille maximale de 256 octets. Le paquet est divisé en sous-paquets de 20 octets pour un SF < 10 et 10 octets pour un SF = 10. La réception de ces sous-paquets est confirmée individuellement, ce qui fait en sorte que si une partie du paquet n'est pas reçue, seulement le sous-paquet sera retransmis et non le paquet en entier. On peut observer ces sous-paquets dans la puissance consommée par le module radio. Dans la figure 1.3, on observe un paquet d'une taille de 256 octets transmis avec un SF de 7. On retrouve donc 13 sous-paquets de 20 octets.

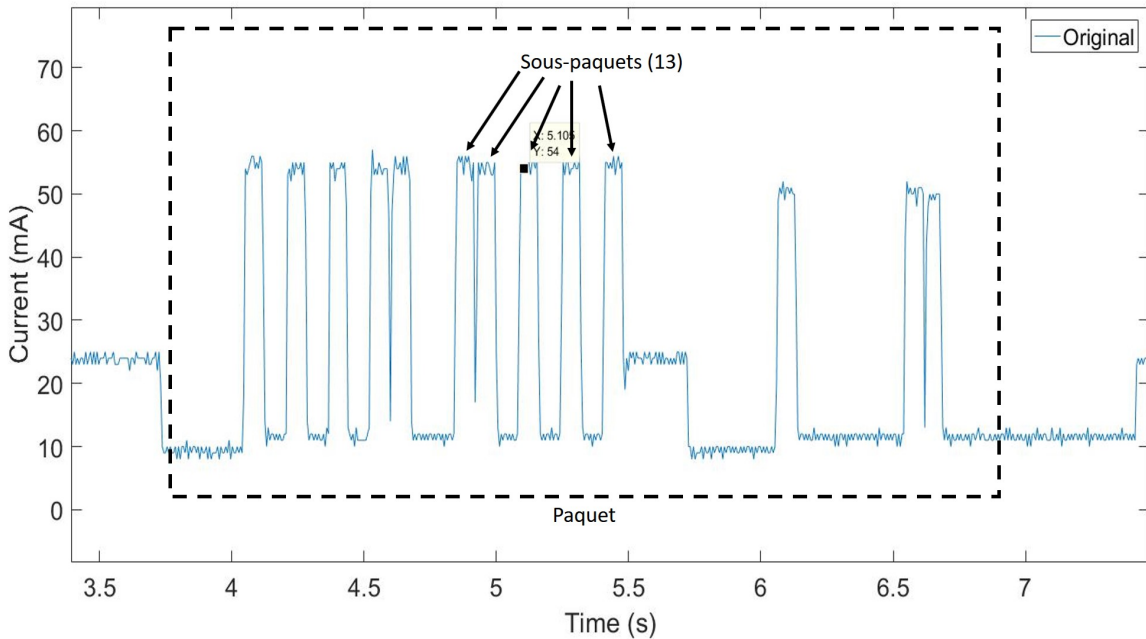


FIGURE 1.3 – Paquet divisé en sous-paquets illustré par le courant tiré par le module radio.

Suite à la transmission de son paquet, le noeud attend la confirmation de la station de base. Si un sous-paquet n'a pas été reçu, le noeud le retransmet lors de la prochaine fenêtre de transmission. Il retombe ensuite en veille jusqu'à sa prochaine transmission.

### 1.3 Paramètres d'évaluation

Afin de pouvoir optimiser le réseau, il faut avoir des paramètres afin de pouvoir évaluer ses performances. Comme mentionné en introduction, les aspects à optimiser sont la consommation d'énergie, la largeur de bande utilisée et le taux de réception à la station de base.

### 1.3.1 Efficacité énergétique

L'efficacité énergétique se traduit par la quantité d'énergie nécessaire pour transmettre une quantité de données. Ici, les unités utilisés seront les joules pour l'énergie et les bits pour les données. Ce sera donc le nombre de joules nécessaire pour envoyer un bit de donnée. Pour ce faire, on calculera la quantité totale d'énergie consommée par le réseau dans un intervalle de temps et on le divisera par le nombre de bit de données reçus à la station de base.

Pour calculer l'énergie consommée par le réseau, on calcule d'abord la consommation des noeuds et on les additionne ensuite. L'énergie consommée lors des périodes de veilles est négligée.

$$E_n = V * I * T_{TX} \quad (1.9)$$

$$E_{Total} = \sum_{i=1}^m E_n(i) \quad (1.10)$$

$E_n$  : Énergie consommée (J)

$E_{Total}$  : Énergie totale consommée (J)

$V$  : Tension électrique d'opération du noeud (V)

$I$  : Courant électrique du module radio lors de la transmission (A)

$T_{TX}$  : Temps de transmission total (s)

$m$  : Nombre de noeuds

Pour ce qui est de la quantité de données, il est important de prendre la quantité reçue par la station de base et non la quantité transmise par les noeuds puisqu'il faut prendre en compte les pertes d'énergie causées par les paquets perdus.

On obtient l'équation suivante pour déterminer l'efficacité énergétique.

$$E_b = \frac{Q_{rec}}{E_{Total}} \quad (1.11)$$

$E_b$  : Efficacité énergétique (bits/J)

$Q_{rec}$  : Quantité de données reçue (bits)

### 1.3.2 Efficacité spectrale

L'efficacité spectrale se traduit par le débit de données selon la largeur de bande utilisée. Les unités utilisées sont les bits par secondes (bits/sec) pour le débit de données et les Hz pour la largeur de bande utilisée. On retrouve donc une efficacité spectrale exprimée en bits/sec/Hz. Le débit de données sera calculé avec la quantité de données reçues par la station de base et le temps total de transmission des noeuds.

$$R_b = \frac{Q_{rec}}{T_{TX}} \quad (1.12)$$

$R_b$  : Débit de données (bits/sec)

$Q_{rec}$  : Quantité de données reçue (bits)

$T_{TX}$  : Temps de transmission total (s)

La largeur de bande utilisée dans l'application présente est fixe à 125 kHz.

On obtient l'équation suivante pour déterminer l'efficacité spectrale.

$$\eta_f = \frac{R_b}{B} \quad (1.13)$$

$\eta_f$  : Efficacité spectrale (bits/sec/Hz)

$B$  : Largeur de bande (Hz)

### 1.3.3 Taux de réception total

Le taux de réception total correspond au rapport de la quantité de données reçue à la station de base sur la quantité de données transmises par les noeuds.

$$RR = \frac{Q_{rec}}{Q_{TX}} \quad (1.14)$$

$RR$  : Taux de réception total (bits/bits)

$Q_{rec}$  : Quantité de données reçues (bits)

$Q_{TX}$  : Quantité de données transmises (bits)

La différence entre la quantité de données transmises et la quantité de données reçues est la quantité de données perdues. Ces données peuvent être perdues pour deux raisons, par collision ou par *path loss*.

#### Taux de collisions

Une raison pour laquelle un paquet peut être perdu est qu'il est entré en collision avec un autre paquet. Trois conditions doivent être comblées pour qu'il y ait collision entre deux paquets :

1. Les deux paquets, à un moment, sont transmis simultanément.
2. Les deux paquets sont transmis à la même fréquence.
3. Les deux paquets utilisent le même SF.

Il y a alors collision. Il reste à déterminer si les deux paquets ou un seul paquet seront perdus. Si la puissance de réception à la station de base des paquets est semblable, les deux paquets sont perdus. Cependant, si la différence de puissance dépasse un certain niveau, dans ce cas-ci 6 dB pour le module radio de Semtech [1], seulement le paquet avec la puissance la plus faible sera perdu et l'autre sera reçu correctement [11].

### **Taux de pertes par *path loss***

Une autre raison pour laquelle un paquet serait perdu est le *path loss*. Si la puissance de réception d'un paquet, même à une puissance maximale de transmission et avec un SF maximale, n'est pas supérieure à la sensibilité correspondant à son SF, le paquet ne sera pas reçu. Les pertes par *path loss* sont causées habituellement par une trop grande distance entre le noeud et la station de base.

## Chapitre 2

# Expérimentation terrain

Il a été décrit à la fin du chapitre 1 comment les performances du réseau sont évaluées et les paramètres nécessaires afin d’y arriver. Les mesures expérimentales doivent comprendre le courant consommé par un noeud et caractériser le temps de transmission. Il est possible de remplir ces deux conditions en monitorant le courant en temps réel, c’est-à-dire en mesurant le courant à une fréquence assez élevée afin d’obtenir un profil. On peut en déduire ensuite beaucoup d’information comme le taux de retransmission, l’énergie consommée, le temps de transmission, etc. Le système et la méthodologie d’expérimentation est décrit. Ensuite, les différentes versions du prototype et finalement les résultats sont présentés et commentés.

### 2.1 Système pour expérimentation

Une campagne de mesure a été réalisée afin d’obtenir suffisamment de données pour déterminer les paramètres nécessaires pour établir un modèle de simulation.

#### 2.1.1 Structure

La figure 2.1 illustre la structure du montage expérimental. Le noeud transmet les données à la station base via le protocole de Symphony Link™. La station de base est connectée à internet par WiFi et transfère les données sur le serveur fourni par Link Labs (Conductor). À l’aide d’un portable, les données sont récupérées du serveur avec un programme python et analysées sur MATLAB.

#### 2.1.2 Procédure

La prise de mesures à un point donné s’effectue comme suit. Le noeud est placé à un endroit arbitraire. Il est configuré pour envoyer des paquets un à la suite de l’autre. Les paquets sont reçus par la station de base et mis sur le serveur. Pendant ce temps, un programme de récupération de données est en marche sur un ordinateur détectant et recouvrant les paquets

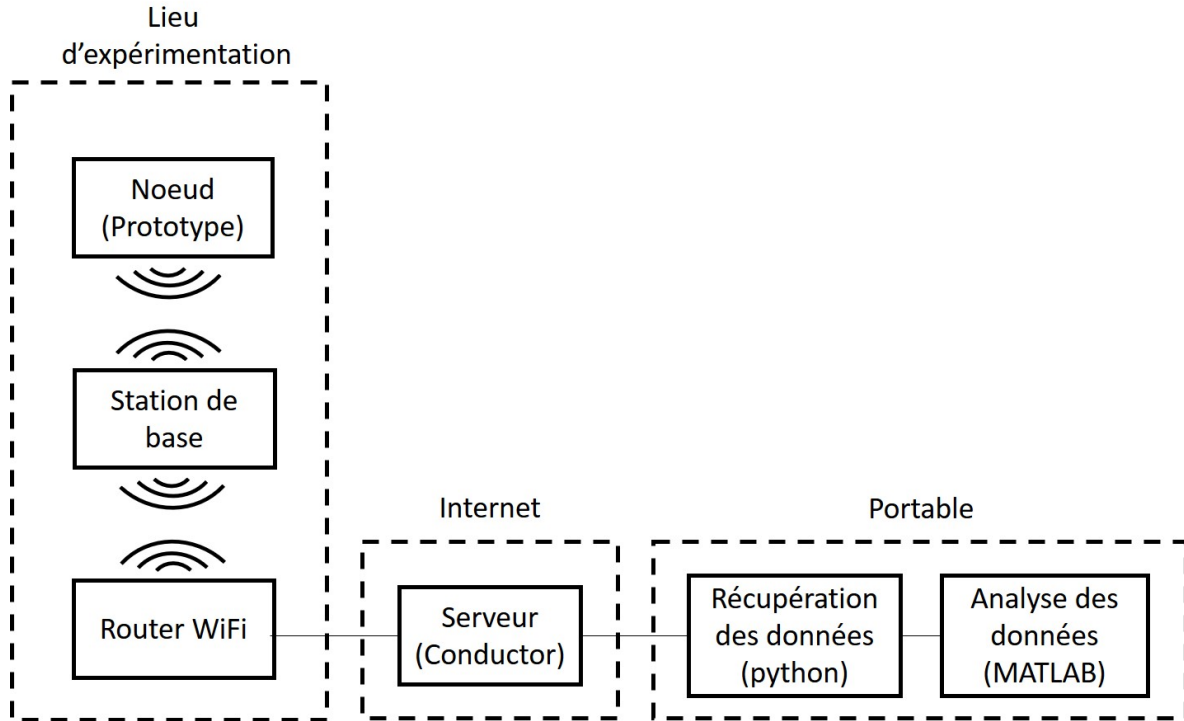


FIGURE 2.1 – Structure du montage expérimental.

entrant sur le serveur en temps réel. De cette manière, il est sûr que les données sont attribuées au bon point de mesure. Après qu'il y ait eu 20 paquets de récupérés, afin d'avoir un bon nombre d'échantillons, les données des 20 paquets sont sauvegardées dans un fichier MATLAB et ce fichier est identifié. Pendant que les données sont recueillies, un point GPS du noeud est pris avec la même identité que le fichier de données. Le noeud est ensuite mis à un autre endroit et le processus recommence.

### 2.1.3 Noeud et station de base

Le noeud est, pour la plupart des mesures prises, placé sur le toit d'une voiture ayant une hauteur de 1,3 mètre (voir figure 2.2).

Pour les mesures qui ont été prise à pied, le noeud a été placé approximativement à la même hauteur. On utilise le module LL-RLP-20 avec l'antenne imprimée puisque c'est cette configuration qui se prête le mieux à une application comprenant des batteries. Le module LL-RXR-27 consomme un courant trop élevé à puissance maximale pour être viable avec des batteries. Il est recommandé de l'utiliser lorsqu'une alimentation infinie est disponible.

Une autre raison est qu'il est possible de voir l'adaptation des paramètres de transmission de manière plus progressive. La portée du module LL-RXR-27 ou du module LL-RLP-20 avec l'antenne externe avec un SF minimal dépasse le rayon autour de la station de base où le terrain est plat. Passé ce rayon, la qualité du lien sans fil se dégrade drastiquement et ne suit plus



FIGURE 2.2 – Position du noeud lors des prises de mesures.

la courbe progressive. Ainsi en utilisant le module avec la puissance la plus petite et l'antenne avec le plus petit gain, il est possible d'observer un changement progressif de la valeur du SF. La station de base est positionnée sur le toit d'une maison à une hauteur approximative de 10 mètres (voir figure 2.3). Elle est connectée à l'internet avec un réseau WiFi.

#### 2.1.4 Données

Chaque paquet envoyé avec le protocole Symphony Link™ est accompagné automatiquement de certaines statistiques de base dont le SF, le SNR, le RSSI et la fréquence de transmission. Les paquets envoyés contiennent les mesures prises lors de la dernière transmission. Ces données sont le courant moyen de transmission, le nombre de sous-paquets transmis et le nombre d'échantillons pendant la transmission.

En observant la figure 2.12, on peut voir que le courant lors de la transmission est plus élevé (environ 55 mA par rapport à 25 mA et 10mA). Ainsi, pour calculer la consommation de courant lors d'une transmission, on considère seulement les échantillons supérieurs à un seuil.



FIGURE 2.3 – Position de la station de base.

Un seuil  $I_{TH}$  de 40 mA est utilisé dans ce cas. De cette manière, seulement les échantillons durant une transmission sont gardés et les autres sont rejetés.

Un sous-paquet est détecté par la présence d'une montée suivie d'une descente dans les échantillon de courant. Une montée est détectée si au moins cinq échantillons sous le seuil  $I_{TH}$  sont suivis de dix échantillons au-dessus de  $I_{TH}$ . Une descente est détectée si ces conditions sont rencontrées en sens inverse.

Le nombre d'échantillons pendant la transmission correspond aux nombre de mesures ayant une valeur supérieure à  $I_{TH}$ .

### 2.1.5 Serveur

Les données recueillies par la station de base sont envoyées sur le serveur nommé *Conductor* fourni par Link Labs avec Symphony Link™ via internet. Un API en langage python est aussi fourni par la compagnie, permettant de facilement récupérer les données. Le programme



python récupérant les données en temps réel et les formatant afin qu'elles puissent être utilisées sur MATLAB est présenté à l'annexe A.

### 2.1.6 Analyse des données

L'analyse des données est faite avec le logiciel MATLAB. À partir des données brutes entrantes (présentées à la section 2.1.4), on obtient ces paramètres pour chaque paquets reçus :

1.  $I_{TX}$  : Courant moyen de transmission (mA)
2.  $BW$  : Largeur de bande (Hz)
3. SF
4. SNR
5. RSSI
6.  $RR_{lost}$  : Ratio de sous-paquets perdus
7.  $T_{TX}$  : Temps de transmission (ms)
8.  $R_B$  : Débit de données effectif (Bytes/s)
9.  $E_{TX}$  : Énergie consommée (J)
10.  $P_{TX}$  : Puissance de transmission (dBm)
11.  $PL$  : *Path loss* (dB)

Certains paramètres sont des données entrantes directement (1, 2, 3, 4, 5).

Le ratio de sous-paquets perdus est déduit par le rapport du nombre de sous-paquets attendus sur le nombre de sous-paquets transmis. Il est possible de déterminer le nombre de sous-paquets qui sera transmis s'il n'y a aucune perte (voir section 1.2.2), ce qui correspond au nombre de sous-paquets attendus. Le nombre de sous-paquets transmis est dans les données entrantes.

Le temps de transmission est déterminé à l'aide du nombre d'échantillons recueillis lors de la transmission, qui est une donnée entrante et la fréquence d'échantillonnage (10 kHz).

$$T_{TX} = \frac{N_{samples}}{10\text{kHz}} \quad (2.1)$$

Le nombre de données transmises avec succès qui est connu (250 Bytes). On peut déduire le débit de données effectif en le divisant par le temps de transmission.

$$R_B = \frac{250}{T_{TX}} \quad (2.2)$$

L'énergie consommé lors de la transmission est calculée avec le produit du courant moyen, de la tension et du temps de transmission.

$$E_{TX} = 3.3V * I_{TX} * T_{TX} \quad (2.3)$$

La puissance de transmission est déterminée à partir du courant moyen de transmission et de la courbe caractérisant la relation courant-puissance de transmission présentée à la section

### 2.3.1.

La valeur du *path loss* est trouvée par la différence entre la puissance de transmission et le RSSI.

$$PL = P_{TX} - \text{RSSI} \quad (2.4)$$

Il s'agit des paramètres déterminés pour chaque point de mesure. Une analyse globale de l'entièreté des mesures est décrite à la section 2.3.

## 2.2 Prototype du module radio

### 2.2.1 Version 1

La première version du module radio était constituée de plusieurs cartes d'évaluation connectées ensemble par câble. À la figure 2.4, on présente le schéma-blocs du prototype. Le Launch-

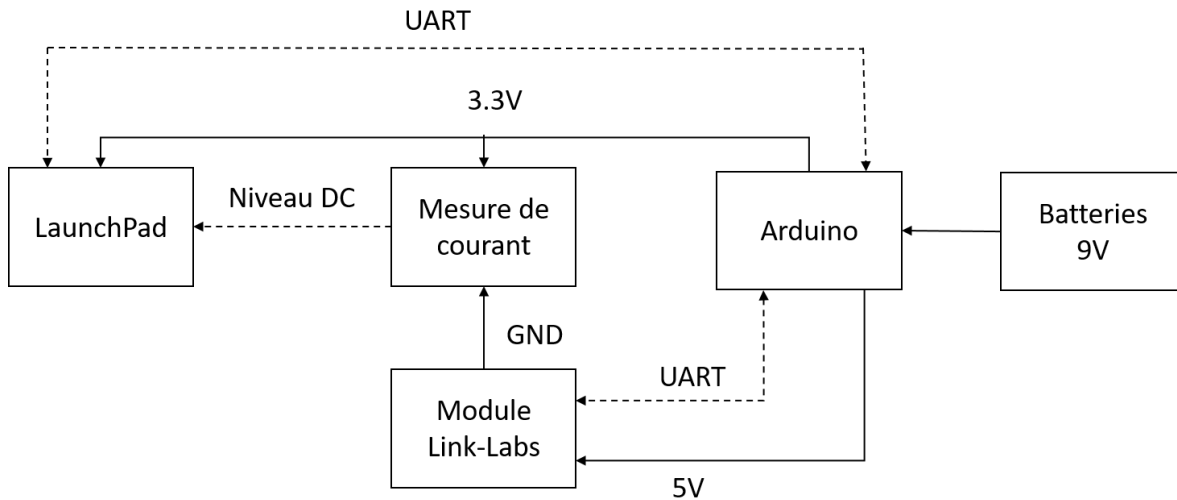


FIGURE 2.4 – Schéma-bloc de la version 1 du module radio.

Pad est un module de Texas Instruments. Il s'agit d'un module de développement du MSP430F5529 permettant de facilement s'y interfacer. Le module de courant est une carte faite maison. Il s'agit d'un circuit de transimpédance *Low side*. Le module Link Labs est une carte d'évaluation du LL-RLP-20 ou du LL-RXR-27.

Le module Arduino contrôle le module Link Labs via le UART et récupère les données du courant consommé via le UART du Launch Pad. Il alimente aussi le système en 5 V et 3.3 V à partir d'un bloc de batteries 9 V (6 AA x 1.5 V). Le Launchpad échantillonne à une fréquence de 10 kHz le niveau de tension à son ADC correspondant au courant consommé par le module de Link Labs. Il transmet les informations recueillies au Arduino lorsqu'il est interrogé.

À la figure 2.5, on peut voir un exemple de mesure de courant effectuée par le système. Il est possible de bien discerner les différentes *time slots*. Cependant, les connexions de cette version

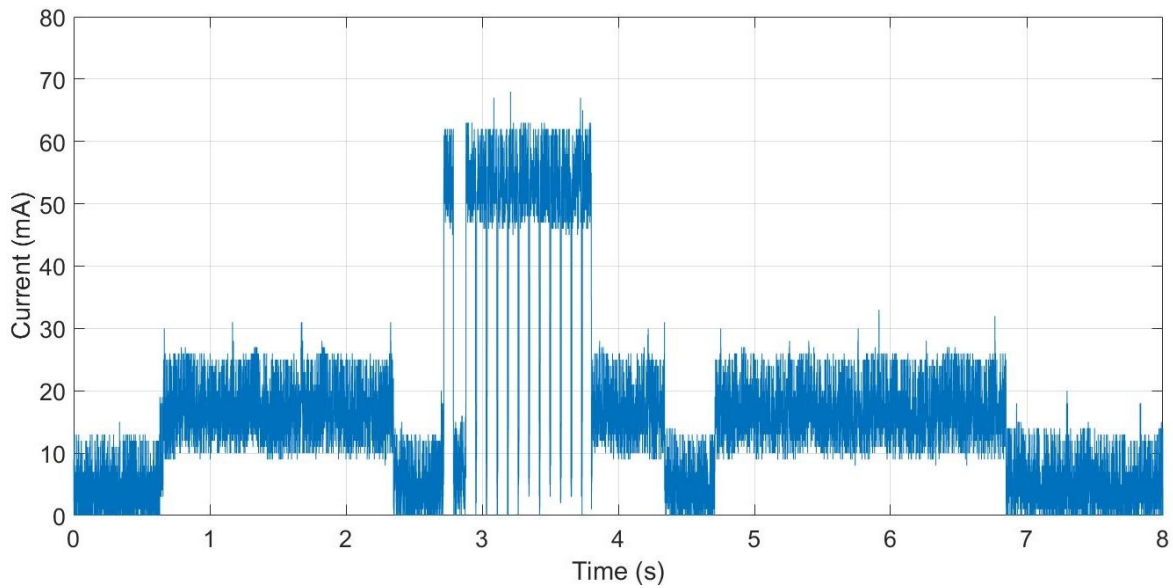


FIGURE 2.5 – Courant mesuré par le système dans le prototype version 1.

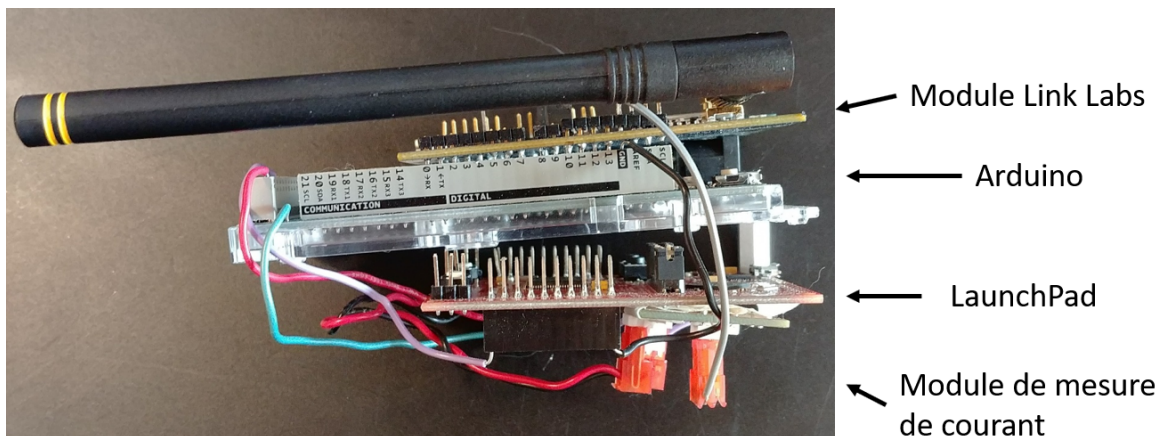


FIGURE 2.6 – Prototype version 1 avec les modules identifiés.

sont fragiles. Les données sont influencées lorsqu'on manipule le prototype. De plus, il y a plus de pièces que nécessaire. C'est pourquoi une seconde version a été réalisée.

La figure 2.6 montre le prototype version 1 avec les différents modules identifiés.

### 2.2.2 Version 2

La deuxième version du prototype est ramenée sur une seule carte électronique.

On peut voir le schéma-blocs du prototype version 2 à la figure 2.7. On constate que l'Arduino est éliminé et qu'un module d'alimentation a été ajouté. Les fonction de l'Arduino sont assu-

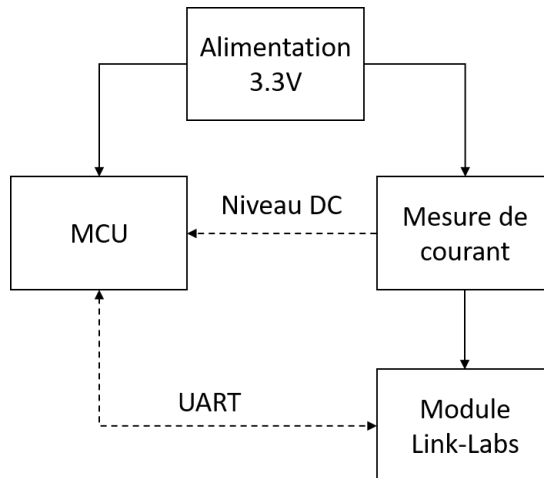


FIGURE 2.7 – Schéma-bloc du prototype du module radio.

mées par le MSP430 (MCU). Le bloc d'alimentation est un *buck-boost* permettant une grande plage de tensions d'entrée (2 V-17 V).

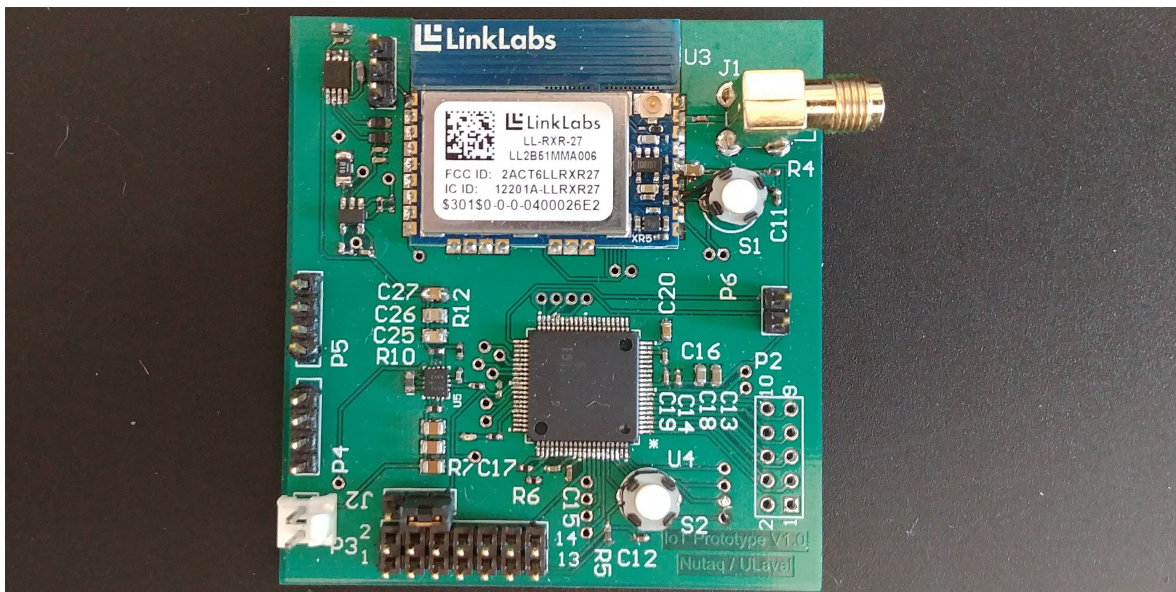


FIGURE 2.8 – Photo du prototype version 2.

On peut voir une photo de la version 2 à la figure 2.8. La taille du prototype est beaucoup moindre par rapport à la version 1.

À la figure 2.9, on peut voir un exemple de mesure de courant par le prototype version 2. On remarque que le signal est très bruité. Il est impossible de bien discerner les *time slots* et même de déduire un courant moyen. Une version 3 est donc nécessaire afin de clarifier la mesure du courant.

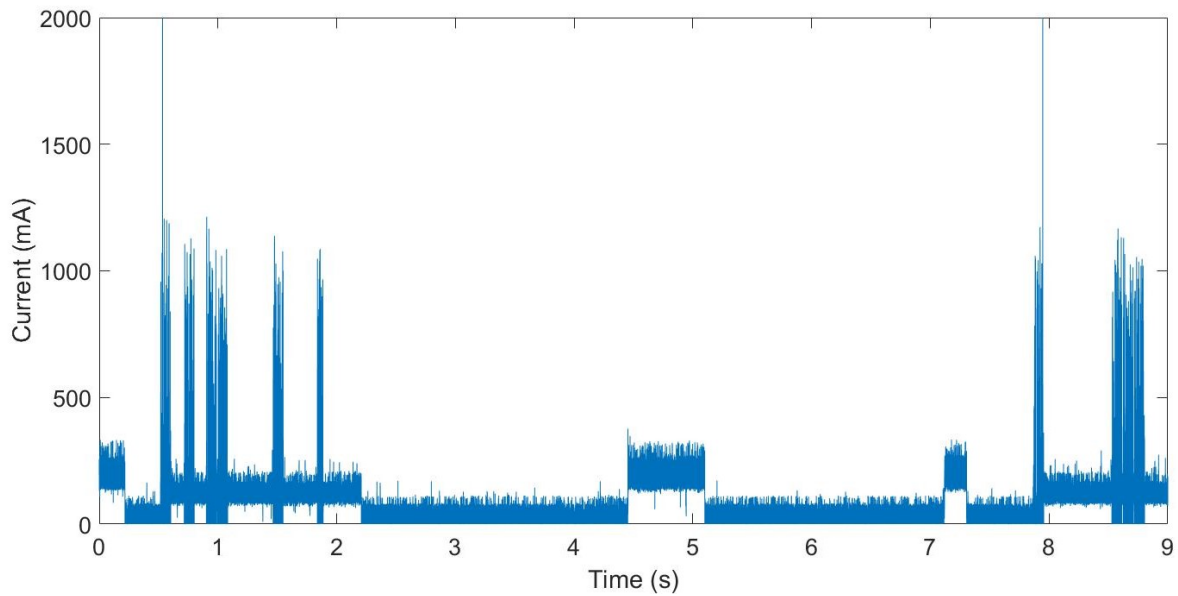


FIGURE 2.9 – Courant mesuré par le système dans le prototype version 2.

### 2.2.3 Version 3

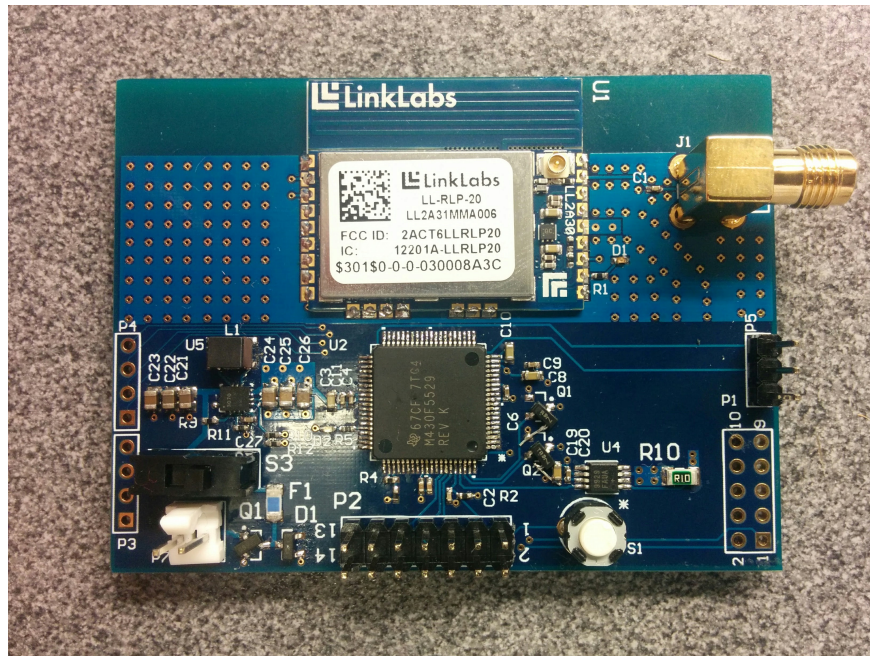


FIGURE 2.10 – Photo du prototype version 3.

Le concept de la version 3 est identique à celui de la version 2, à l'exception de la mesure de courant qui se fait *High side* au lieu de *Low side*. La figure 2.10 montre une photo de cette version. Les principales modifications sont dans le design de la carte de circuits imprimés. Les

traces de signal analogique ont été isolées et raccourcies le plus possible. Le tout dans le but d'améliorer la qualité de la mesure de courant.

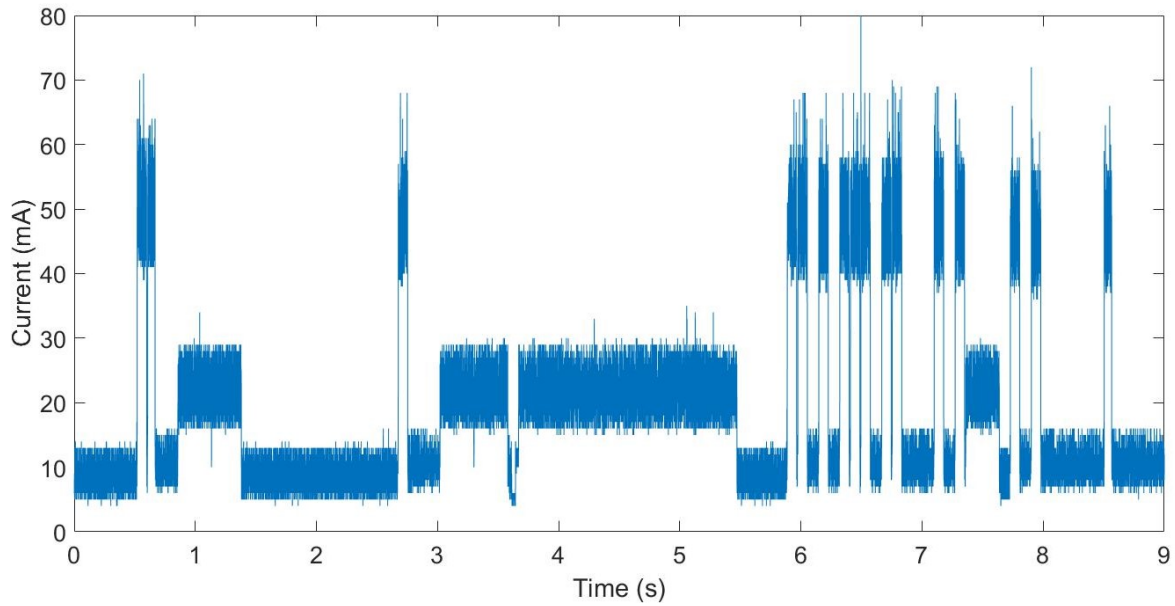


FIGURE 2.11 – Mesure de courant du prototype version 3 sans moyenne mobile.

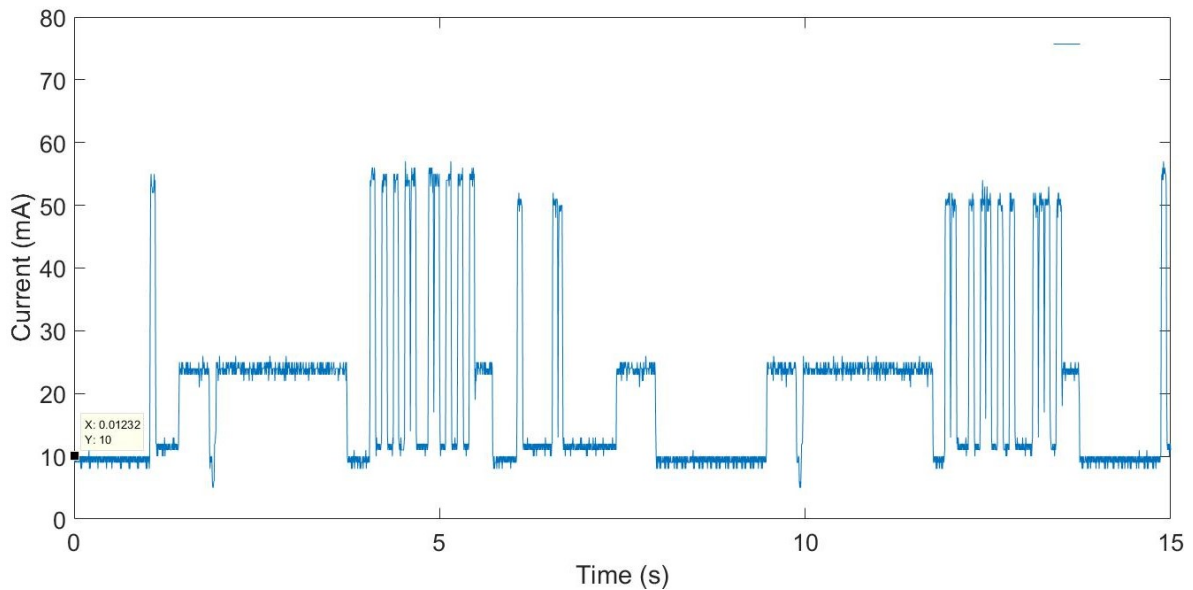


FIGURE 2.12 – Mesure de courant du prototype version 3 avec moyenne mobile.

La figure 2.11 montre la mesure originale de courant du prototype version 3. On constate qu'elle est beaucoup plus claire, mais tout de même bruitée. Il s'avère que la fréquence de mesure de courant est plus élevée que nécessaire. Ce qui nous permet de faire une moyenne

mobile qui vient lisser la courbe de courant (voir la figure 2.12). Cette version du prototype permet de bien discerner les *time slots*, de déterminer de manière stable le courant moyen et est assez robuste pour être sortie sur le terrain. C’est avec cette version que les mesures nécessaires au modèles seront prises.

## 2.3 Résultats

Nous présentons ici les résultats obtenus et leur interprétation, d’abord, la relation courant et puissance de transmission du module radio, ensuite la campagne de mesures en campagne et le calcul des paramètres nécessaires au modèle de simulation.

### 2.3.1 Relation courant et puissance de transmission

Il est nécessaire de connaître la puissance de transmission pour déduire le *path loss* et ensuite établir le facteur d’atténuation de la variable aléatoire (*shadowing*). Pour ce faire, on doit établir la relation entre une mesures qui est accessible (le courant) et le paramètre voulu.

On configure le module radio pour qu’il émette une tonalité continue à une puissance connue. On utilise le prototype version 3 incluant la circuiterie nécessaire afin de monitorer le courant du module radio. On obtient alors plusieurs points reliant courant et puissance de transmission permettant de tracer la courbe les reliant (figure 2.13 pour le LL-RXR-27 et figure 2.14 pour le LL-RLP-20). Ces figures illustrent aussi la précision des équations déterminées à l’aide de ces mesures.

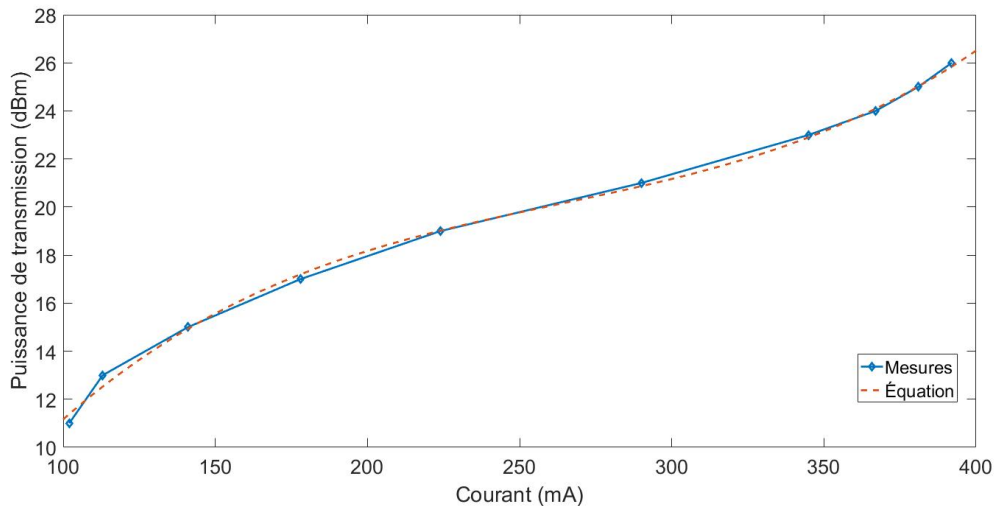


FIGURE 2.13 – Puissance de transmission selon le courant (module LL-RXR-27).

Pour le module LL-RXR-27, avec un coefficient de détermination de 0.9981, on a ce polynôme

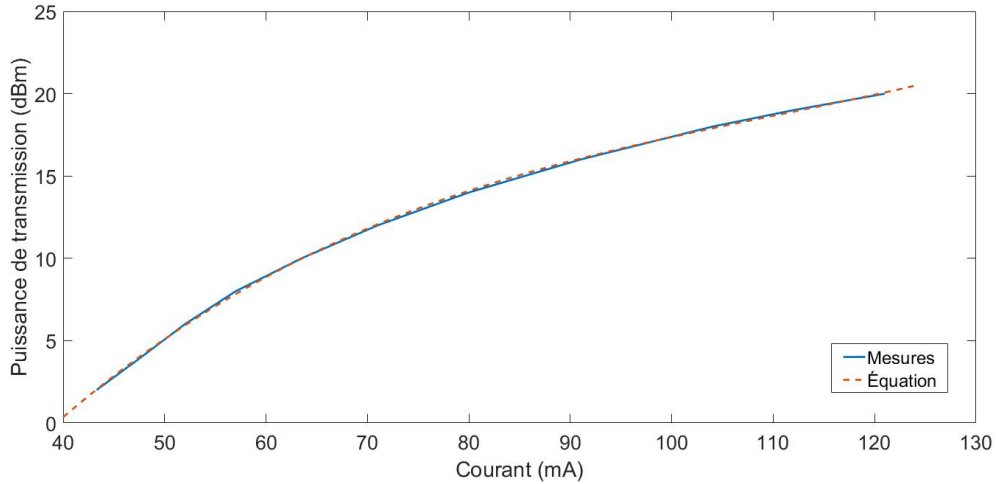


FIGURE 2.14 – Puissance de transmission selon le courant (module LL-RLP-20).

d'ordre 3

$$P_{RXR} = (1 \times 10^{-6})i^3 - 0.0008i^2 + 0.2469i - 6.2225 \quad (2.5)$$

valide pour une puissance de 11 à 27 dBm correspondant à l'intervalle de valeurs de puissance de transmission possible pour ce module. Pour le module LL-RLP-20, avec un coefficient de détermination de 0.9998, on a ce polynôme d'ordre 3

$$P_{RLP} = (3 \times 10^{-5})i^3 - 0.0089i^2 + 1.114x - 31.642 \quad (2.6)$$

valide pour une puissance de 2 à 20 dBm correspondant à l'intervalle de valeurs de puissance de transmission possible pour ce module.

### 2.3.2 Carte des mesures

La figure 2.15 est une carte montrant où les mesures ont été prises. Les mesures suivent les différentes routes autour de la station de base puisque la plupart des mesures ont été faites avec une voiture. Toutes les directions ont été prises en compte à l'exception du nord où l'accès n'était pas possible.





FIGURE 2.15 – Carte des emplacements des mesures.

### 2.3.3 Path Loss

On utilisera un modèle empirique afin d'établir la valeur du *path loss* selon la distance. Le modèle par ajustement de coefficient de puissance [4] ou *Log-distance path loss model* est populaire pour estimer une valeur du *path loss* [1].

$$L(d) = L(d_0) + 10\gamma_p \log_{10} \left( \frac{d}{d_0} \right) + X_p \quad (2.7)$$

$L(d)$  : *Path loss* en dB à une distance  $d$  (m)

$L(d_0)$  : *Path loss* en dB à distance arbitraire  $d_0$  (m)

$\gamma_p$  : coefficient de puissance d'atténuation

$X_p$  : Variable aléatoire Gaussienne centrée à 0 représentant le *shadowing*

#### Coefficient de puissance d'atténuation

En regardant (2.7), on peut voir que le coefficient de puissance d'atténuation correspond au rapport suivant :

$$\gamma_p = \frac{L}{10 \log(d)} \quad (2.8)$$

On trace le graphique du *path loss* selon  $10 \log(d)$  et on y calcule la régression linéaire. La pente de la droite résultante est le coefficient de puissance d'atténuation (voir figure 2.16).

On obtient alors un coefficient  $\gamma_p = 2.40$ . En comparant cette valeur aux valeurs de référence

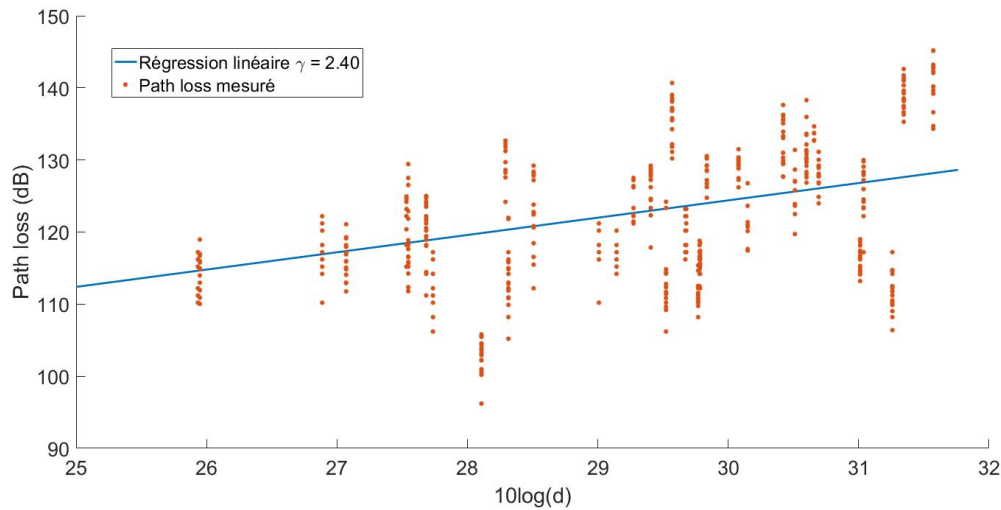


FIGURE 2.16 – Détermination du coefficient de puissance d'atténuation par la régression linéaire du *path loss* et du logarithme de la distance.

montrées dans le tableau 2.1 [8], on peut voir qu'elle se trouve entre l'espace libre et une zone urbaine. Ce qui a du sens, considérant que l'environnement de mesure est en campagne, un environnement se rapprochant de l'espace libre, mais comprenant des obstacles disparates tels que des bâtiments et des forêts.

TABLE 2.1 – Coefficient de puissance d’atténuation selon l’environnement.

<b>Environnement</b>	$\gamma_p$
Espace libre	2
Zone urbaine	2.7 à 3.5
Zone urbain avec ombrage	3 à 5
Intérieur d’un bâtiment dégagé	1.6 à 1.8
Intérieur d’un bâtiment obstrué	4 à 6
Intérieur d’une usine obstruée	2 à 3
<b>Expérimentation</b>	<b>2.40</b>

### Variable aléatoire pour l’effet d’ombrage

L’effet d’ombrage est une variation aléatoire de la valeur d’atténuation du signal qui peut être causé par plusieurs facteurs. Cette variable aléatoire est déterminée en combinant tous les paquets reçus. Il est possible d’établir une distribution de la valeur d’ombrage par la variation de la valeur du *path loss*. La figure 2.17 montre cette distribution. Elle peut être représentée par une distribution normale [4] qui a pour paramètres :  $\mu = 0$  et  $\sigma = 2.972$ .

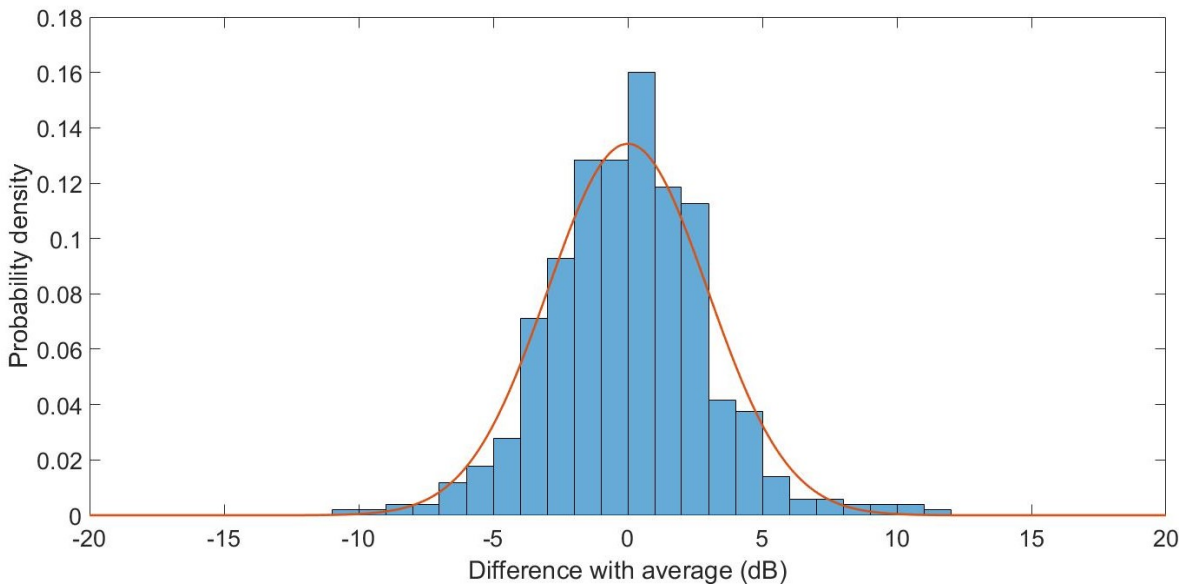


FIGURE 2.17 – Variation du *path loss* par rapport à sa moyenne.

### Distance de référence

La distance de référence est fixée à 394 mètres avec sa valeur de *path loss* moyenne de 114.7 dB. Cette valeur a été choisie parce qu’il s’agit d’une des moins grande distance, que la variance de la valeur de *path loss* est moins élevée que les autres (2.66 vs 2.97) et que sa moyenne suit la courbe moyenne de *path loss* de cette expérimentation.

## Chapitre 3

# Simulation

La simulation est réalisée à l'aide du langage de programmation Python 2.7, plus précisément avec le module SimPy. Il est basé sur un simulateur de LoRaWAN<sup>TM</sup> nommé LoraSim [1] qui a été adapté afin de modéliser le protocole de Symphony Link<sup>TM</sup>.

SimPy est une structure de simulation discrète basée sur des processus. Ces processus existent à l'intérieur d'un environnement et interagissent entre eux et avec l'environnement avec des événements. Lorsqu'un processus produit un événement, il est suspendu jusqu'à ce que la ou les conditions pour cet événement soit remplie. Ces conditions peuvent être une durée d'attente, un signal d'un autre processus, etc.

Pour cette application, il n'y a qu'un seul environnement dans lequel chaque nœud a un processus qui est de transmettre. Le processus est suspendu pendant l'intervalle de temps entre deux transmissions.

Grâce aux données de l'expérimentation terrain, on utilise l'équation (2.7) déterminée à la section 2.3.3 et les équations de consommation de courant (2.5) et (2.6) déterminées à la section 2.3.1.

### 3.1 Modèle

Les nœuds transmettent des paquets à la station de base. L'action de transmettre est un processus qui est répété tout au long de la simulation à un intervalle de temps. Les paquets reçus par la station de base sont sauvegardés dans une liste. Deux conditions peuvent faire en sorte que le paquet ne soit pas reçu par la station de base : une collision avec un autre paquet ou la puissance du signal reçu est trop faible. Si le paquet n'est pas reçu, le processus de transmission est recommencé sans attendre l'intervalle de temps.

Le fonctionnement du modèle est résumé dans la figure 3.1.

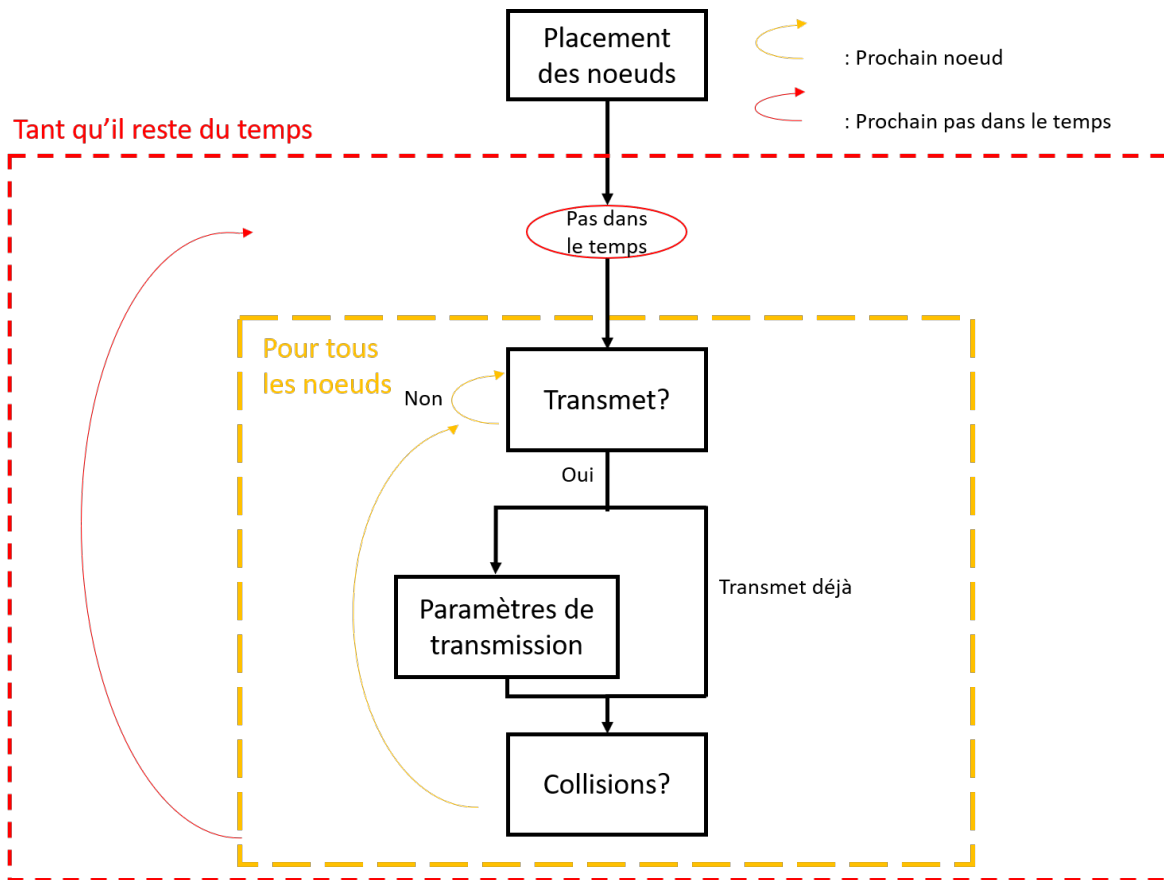


FIGURE 3.1 – Fonctionnement du modèle de simulation.

### 3.1.1 Structure

Le modèle est composé de trois types d'objet :

- Une station de base.
- Des paquets.
- Des noeuds.

#### Station de base

Une station de base est un objet ayant ces attributs :

- *id* : numéro d'identification (nombre entier).
- *x* : position en x dans l'espace (m).
- *y* : position en y dans l'espace (m).

#### Paquets

Un paquet est un objet ayant ces attributs :

- *bs* : station de base à laquelle le paquet est transmis (objet station de base).
- *nodeid* : le numéro d'identification du noeud à partir duquel le paquet est transmis (nombre entier).
- *lost* : si à 1, le paquet n'est pas reçu à la station de base dû à une trop grande atténuation {0,1}.
- *end* : temps après le début de la simulation où la transmission du paquet se termine (ms).
- *totalAirTime* : temps total durant lequel le paquet est transmis, incluant les retransmissions si applicable (ms).
- *Ptx* : puissance du paquet transmis du noeud (dBm).
- *sf* : *spreading factor* auquel le paquet est transmis {7, 8, 9, 10}.
- *snr* : rapport signal à bruit du paquet reçu à la station de base (dB).
- *cr* : facteur de correction du paquet transmis {1, 2, 3, 4}.
- *bw* : largeur de bande utilisée pour transmettre le paquet (Hz).
- *Itx* : courant moyen consommé pour transmettre le paquet (mA).
- *pl* : longueur du paquet (octets).
- *rssi* : puissance à laquelle le paquet est reçue à la station de base (dBm).
- *freq* : fréquence à laquelle le paquet est transmis (Hz).
- *rectime* : temps durant lequel le paquet est transmis sans retransmission (ms).
- *collided* : si à 1, le paquet est entré en collision avec un autre paquet à la station de base {0,1}.
- *reTx* : si à 1, une retransmission est nécessaire pour au moins une partie du paquet {0,1}.
- *reTxRectime* : temps de retransmission nécessaire (ms).

## Nœuds

Un noeud est un objet ayant ces attributs :

- *bs* : station de base avec laquelle le noeud communique (objet station de base).
- *id* : numéro d'identification du noeud (nombre entier).
- *period* : intervalle de temps entre la transmission de deux paquets vers la station de base (ms).
- *mode* : modèle de radio utilisée (LL-RXR-27 ou LL-RLP-20).
- *x* : position en x dans l'espace (m).
- *y* : position en y dans l'espace (m).
- *dist* : distance entre le noeud et la station de base (m).
- *packet* : paquet qui sera transmis par le noeud (objet paquet).

### 3.1.2 Placement des nœuds

La stratégie pour placer les nœuds dans l'espace est un placement aléatoire dans une région déterminée et en respectant une distance minimale entre chaque nœud.

Les paramètres à fixer sont la distance maximale d'un nœud à la *gateway*, la distance minimale, le nombre de nœuds à placer et la distance minimale entre les nœuds.

Les coordonnées sur l'axe des  $x$  et l'axe des  $y$  sont générées aléatoirement entre 0 et la distance maximale. On vérifie que le nœud en  $(x, y)$  respecte ces conditions :

1. Distance minimale et maximale de la *gateway* ;
2. Distance minimale avec tous les autres nœuds.

Si l'une de ces conditions n'est pas respectée, le nœud est rejeté. On répète l'opération jusqu'à ce que nombre de nœuds à générer soit atteint. Les figures 3.2 et 3.3 sont des exemple de placement de nœuds.

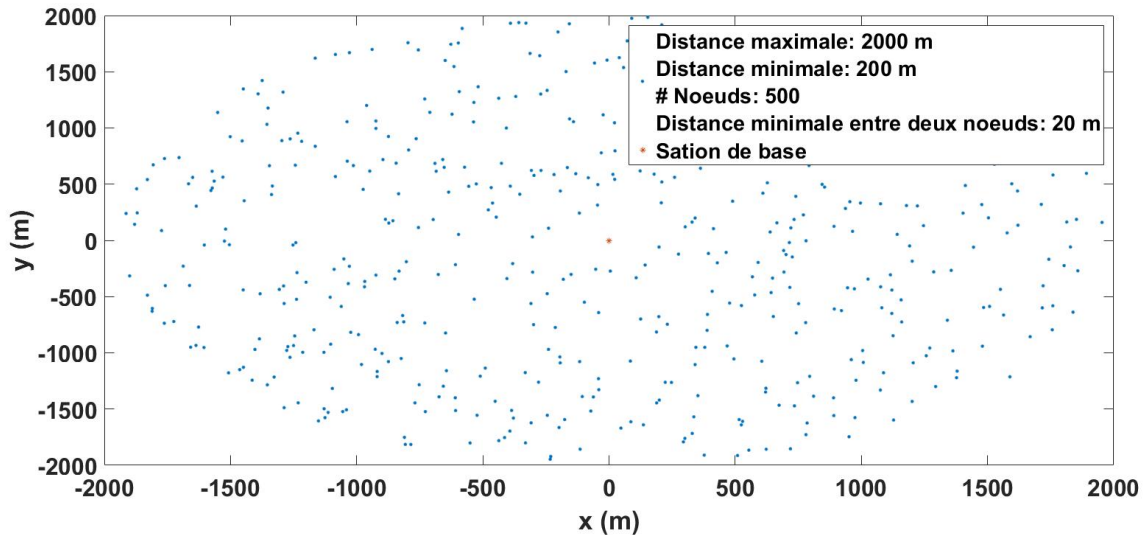


FIGURE 3.2 – Exemples de placements pour 500 noeuds.

### 3.1.3 Paramètres de transmission

Lorsqu'un noeud doit transmettre, il doit d'abord établir ses paramètres de transmission. Ces paramètres sont calculés selon les équations présentées à la section 1.2. La figure 3.4 montre l'ordre et la dépendance des différents paramètres de transmission.

### 3.1.4 Évaluation des collisions

Pour qu'une collision survienne entre deux paquets, trois conditions doivent être réunies :

1. La transmission des paquets se fait au même moment.

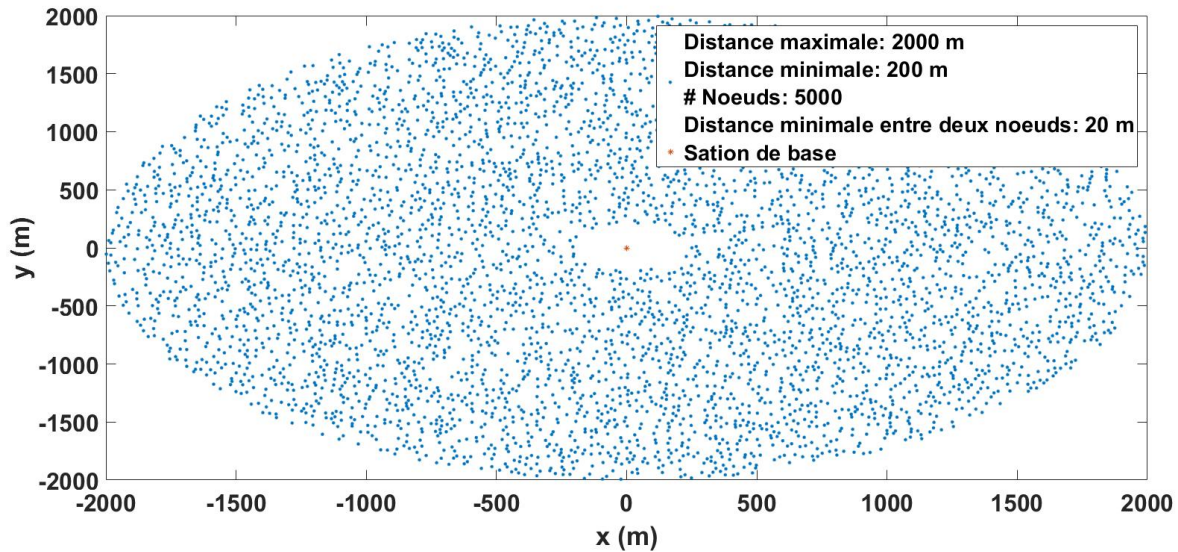


FIGURE 3.3 – Exemples de placements pour 5000 noeuds.

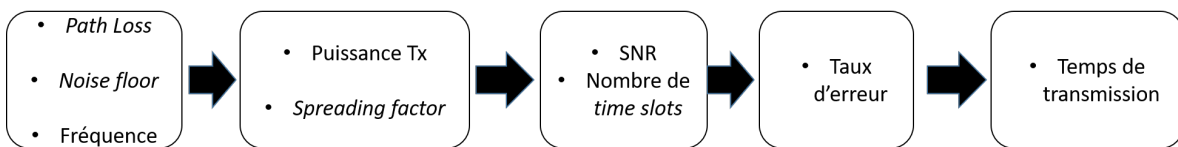


FIGURE 3.4 – Calcul des paramètres de transmission.

2. La fréquence de transmission doit être la même.
3. Le *spreading factor* des deux paquets doit être le même.

Une fois ces trois conditions réunies, il reste à savoir si les deux paquets ou un seul seront perdus. Si la puissance à laquelle un paquet est reçu est supérieure à celle de l'autre paquet, celui avec la puissance la plus élevée est reçu et l'autre est perdu. Tandis que si les puissances de réception des deux paquets est semblable, les deux paquets sont perdus.

## 3.2 Résultats

Les résultats obtenus lors des simulations sont présentés dans cette section. Les paramètres étudiés seront :

- la puissance de transmission ;
- la distribution du *spreading factor* ;
- le temps de transmission par paquet ;
- le taux de collisions ;
- le taux de pertes par *path loss* ;



- le taux de réception total ;
- l'efficacité énergétique ;
- l'efficacité spectrale.

Ces paramètres sont présentés en relation avec le rayon maximal de déploiement ainsi que le nombre d'octets transmis à la station de base par minute. Chaque noeud transmet 256 octets par intervalles de 20 min. Ce qui fait 12.8 octets à la minute par noeud. Les résultats présentés sont la moyenne de 100 itérations de simulation. Le rayon de déploiement varie de 10 à 12000 mètres par saut de 50 mètres. Le débit de données varie de 640 à 5760 octets par minute. Les résultats seront présentés selon le rayon de déploiement maximal et le débit de données transmises. Au départ, il avait été pensé de les mettre en relation avec le nombre de noeuds. Cependant, les résultats sont influencés selon la fréquence de transmission de chaque noeud ; il fait donc plus de sens d'utiliser le débit de données qui prend en compte à la fois le nombre de noeuds et la fréquence de transmission.

### 3.2.1 Puissance de transmission

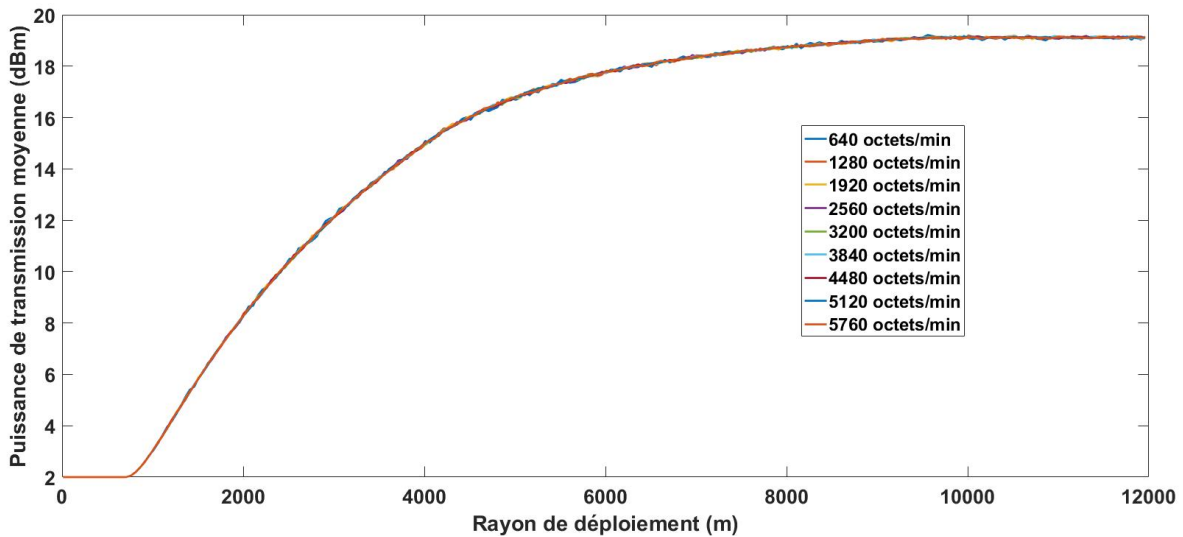


FIGURE 3.5 – Puissance de transmission moyenne selon le rayon de déploiement et le débit de données.

À la figure 3.5, on peut constater que le nombre de noeuds n'a pas d'effet significatif sur les résultats. En regardant de plus près, on remarque que plus le nombre de noeuds est élevé, plus la courbe est lisse puisqu'il y a plus d'éléments contribuant à la moyenne.

Pour la relation avec la distance maximale des noeuds, on remarque un plateau de 10 à 660 mètres. La puissance minimale de transmission est fixée à 2 dBm. Passé 660 mètres, la puissance de transmission doit être élevée graduellement afin d'avoir le niveau minimal à la réception par la station de base. Avec l'équation (3.7) et le SNR minimale d'un SF de 7 (-6 dBm/dBm) [3], on peut avoir l'équation donnant la distance maximale d'un noeud

transmettant à puissance minimale de 2 dBm.

$$d = 394 * 10^{\frac{-0.075 - SNR}{24}} = 696 \quad (3.1)$$

La différence entre le 660 mètre des résultats s'explique par la résolution de la simulation puisque il a des pas de 50 mètres entre chacune des distances.

À 4000 mètres, le taux de croissance commence à diminuer. C'est à ce point que certains noeuds atteignent la puissance de transmission maximale. À 9000 mètres, la puissance moyenne atteint un plateau à 19.1 dBm. Le plateau débute lorsque certains noeuds commencent à être trop loin pour être reçus et ne sont donc plus pris en compte dans le calcul de la puissance moyenne de transmission. Sa distribution à travers les noeuds à portée est alors constante même si le rayon de déploiement augmente.

En utilisant l'outil *Curve fitting tool* de MATLAB, on trouve cette relation avec un facteur R de 0.9991.

$$P_{Tx} = \begin{cases} 2 & 0 \leq d \leq 660 \\ \frac{(6.221e5)d - 3.520e8}{d^2 + (1.308e4)d + 7.603e7} & 660 < d \leq 12000 \end{cases} \quad (3.2)$$

### 3.2.2 Distribution du *spreading factor*

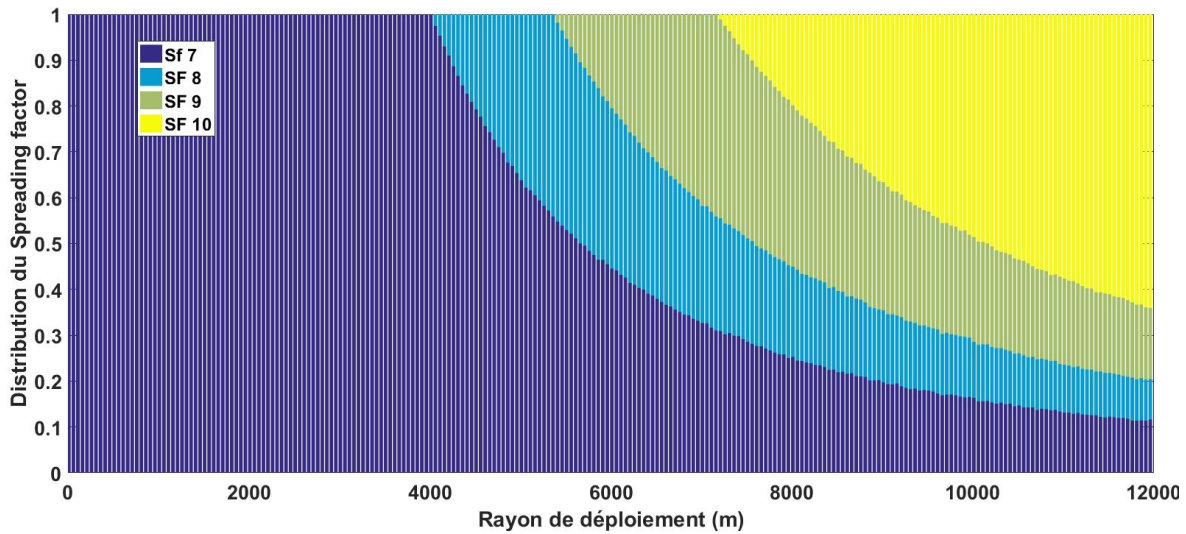


FIGURE 3.6 – Distribution du *spreading factor* selon le rayon de déploiement.

La figure 3.6 représente la distribution du *spreading factor* selon le rayon de déploiement. Ce paramètre ne dépend d'aucune façon du nombre de noeuds en opération, donc seulement la simulation à 450 noeuds est illustrée. Les proportions restent les mêmes, peu importe le nombre de noeuds.

On remarque que la distribution est entièrement SF = 7 de 10 mètres à 3912 mètres. Comme vu à la section précédente, les noeuds augmentent d'abord leur puissance de transmission pour palier à l'atténuation due à la distance avec la station de base. Ce n'est que lorsque la puissance

maximale est atteinte et que ce n'est plus suffisant que le *spreading factor* augmente.

À 5216 mètres, SF = 8 apparaît et encore une fois à 6956 mètres, SF = 9.

On constate qu'à chaque nouveau palier, la proportion de la dernière valeur augmente jusqu'à ce qu'un nouveau palier soit atteint. Après quoi, elle se met à descendre. Par exemple, de 3912 à 5216 mètres, la proportion de SF 7 augmente, mais au-delà de 5216 mètres, sa proportion diminue. La figure 3.7 illustre l'idée que chaque valeur de SF possède son aire du rayon de

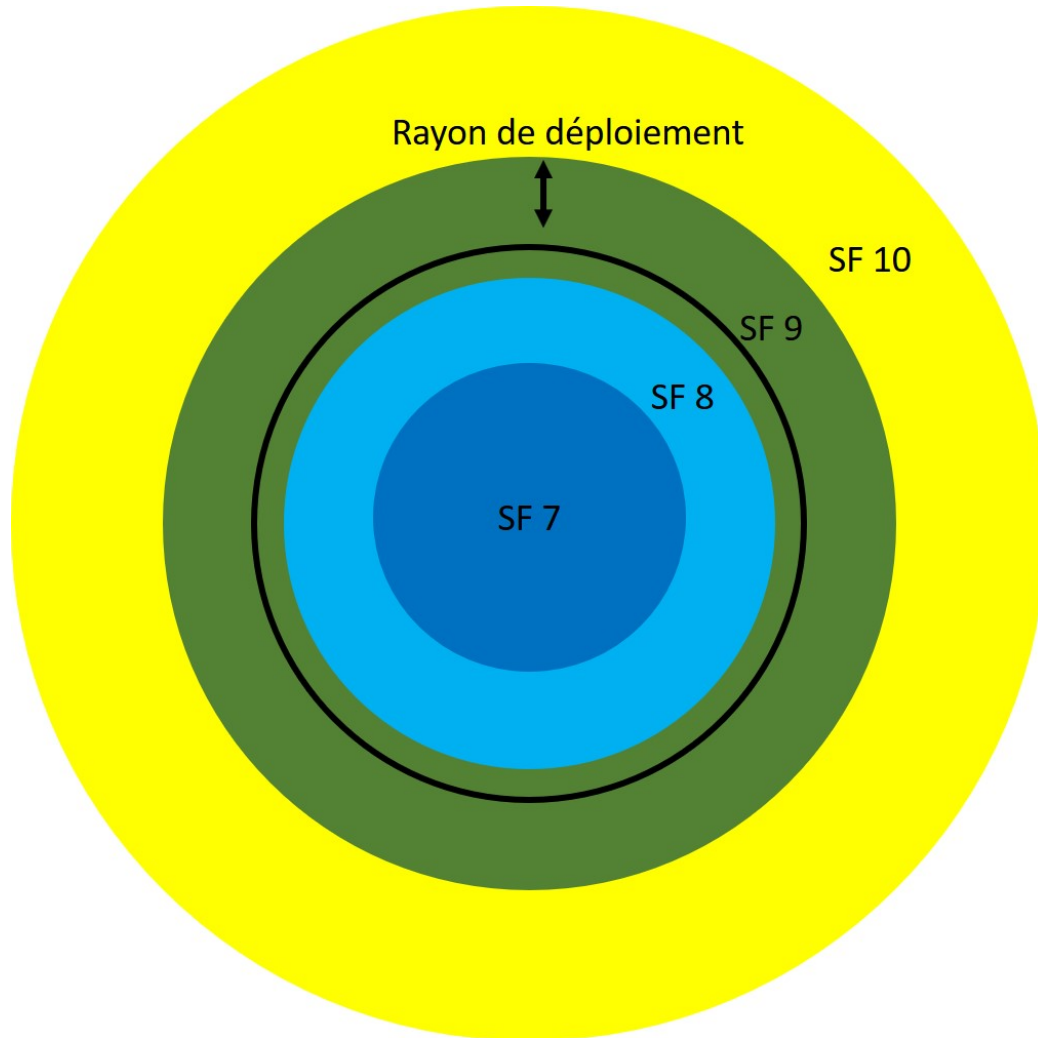


FIGURE 3.7 – Aire de chacune des valeurs de SF.

déploiement (le dessin n'est pas à l'échelle). Imaginons que le rayon de déploiement commence petit et augmente de plus en plus comme pour la simulation. Lorsqu'il atteint la frontière entre SF 7 et SF 8, la proportion de SF 8 augmente jusqu'à ce qu'il atteigne SF 9 puisque l'aire ajoutée à la zone de déploiement est l'aire ayant un SF de 8. À la frontière de SF 9, toute l'aire de SF 8 est incluse et sa proportion est maximale. Elle va ensuite diminuer puisque l'aire totale augmente mais pas l'aire de SF 8.

On peut calculer les proportions de *spreading factor* selon leur rayon et le rayon de déploiement.

Définissons  $r_7$ ,  $r_8$  et  $r_9$  comme étant les rayons de SF et  $r_d$  comme rayon de déploiement. Les proportions des valeurs de SF sont  $\alpha_{SF7}$ ,  $\alpha_{SF8}$ ,  $\alpha_{SF9}$ ,  $\alpha_{SF10}$ .

Pour  $r_d < r_7$  :

$$\begin{cases} \alpha_{SF7} = 1 \\ \alpha_{SF8} = 0 \\ \alpha_{SF9} = 0 \\ \alpha_{SF10} = 0 \end{cases} \quad (3.3)$$

Pour  $r_7 < r_d < r_8$  :

$$\begin{cases} \alpha_{SF7} = \frac{\pi r_7^2}{\pi r_d^2} \\ \alpha_{SF8} = 1 - \alpha_{SF7} \\ \alpha_{SF9} = 0 \\ \alpha_{SF10} = 0 \end{cases} \quad (3.4)$$

Pour  $r_8 < r_d < r_9$  :

$$\begin{cases} \alpha_{SF7} = \frac{\pi r_7^2}{\pi r_d^2} \\ \alpha_{SF8} = \frac{\pi(r_8^2 - r_7^2)}{\pi r_d^2} \\ \alpha_{SF9} = 1 - (\alpha_{SF7} + \alpha_{SF8}) \\ \alpha_{SF10} = 0 \end{cases} \quad (3.5)$$

Pour  $r_d > r_9$  :

$$\begin{cases} \alpha_{SF7} = \frac{\pi r_7^2}{\pi r_d^2} \\ \alpha_{SF8} = \frac{\pi(r_8^2 - r_7^2)}{\pi r_d^2} \\ \alpha_{SF9} = \frac{\pi(r_9^2 - r_8^2)}{\pi r_d^2} \\ \alpha_{SF10} = 1 - (\alpha_{SF7} + \alpha_{SF8} + \alpha_{SF9}) \end{cases} \quad (3.6)$$

On peut aussi estimer les rayons des zones des différents SF. Nous avons les SNR minimaux de chaque SF pour la réception d'un paquet [3]. À partir de 3.7, on isole la distance.

$$SNR = P_{TX} - L_{PL} - N \quad (3.7)$$

- $SNR$  : la valeur est donnée par le tableau 3.1.
- $P_{TX}$  : puissance maximale de transmission 20 dBm.
- $L_{PL}$  : 2.7.
- $N$  : niveau de bruit évalué à -112.6254 dBm lors de l'expérimentation.

En substituant les valeurs et en isolant la distance du noeud, on obtient :

$$d = 394 * 10^{\frac{17,925 - SNR}{24}} \quad (3.8)$$

En y insérant les valeurs minimales de SNR pour chaque *spreading factor*, on obtient le rayon maximal de chacun.

TABLE 3.1 – SNR minimal pour réception selon le *spreading factor*.

SF	SNR min [3] (dBm/dBm)	Rayon max (m)
7	-6	3912
8	-9	5216
9	-12	6956
10	-15	9276

On constate que les rayons maximaux du tableau 3.1 correspondent au rayon de déploiement où l'on voit une nouvelle valeur de SF dans les résultats de simulation.

### 3.2.3 Temps de transmission

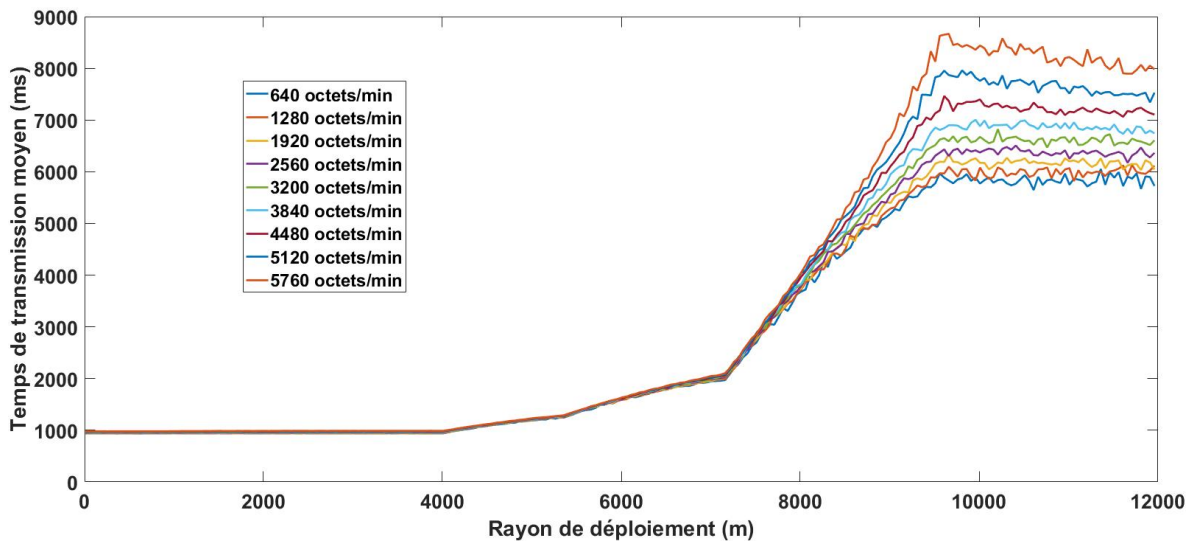


FIGURE 3.8 – Temps de transmission selon le débit de données et le rayon de déploiement.

La figure 3.8 illustre le temps de transmission nécessaire afin de transmettre avec succès un paquet à la station de base. Ce temps comprend les retransmissions dans les cas de collisions. En relation avec le rayon de déploiement, on constate que le temps de transmission varie avec les proportions de *spreading factor*. C'est-à-dire que plus la proportion des SF plus élevés est grande, plus le temps de transmission est grand aussi. On constate que les points où le taux de croissance du temps de transmission augmente correspondent aux points où un nouveau SF apparaît, soient 3912, 5216 et 6956 mètres. À 9276 mètres, on observe un plateau puisque les noeuds au-delà de 9216 mètres sont trop loin pour être reçus (voir figure 3.10). Le temps

de transmission des paquets perdus par atténuation n'est pas comptabilisé.

En relation avec le débit de données, on constate que le temps de transmission est semblable jusqu'à 6956 mètres. Après quoi, il se distingue bien d'un débit à l'autre. Cela est dû au taux de collisions. Comme un paquet perdu par collision est retransmis, son temps de transmission s'en trouve augmenté. On voit que pour un rayon de déploiement inférieur à 3912 mètres, on obtient une constante. Le pourcentage d'écart de temps de transmission entre le plus haut débit (985 ms) et le plus bas (940 ms) est d'environ 5% tout comme l'écart de leur taux de collisions (voir figure 3.11).

Le temps de transmission moyen est défini principalement par le rayon de déploiement et du même coup, la distribution de SF. L'écart entre les débits de donnée est causé par le taux de collisions qui augmente avec le débit. L'équation 3.9 est un exemple d'approximation linéaire de la courbe de débit 3200 octets/min. La figure 3.9 montre la trace de cette équation avec les données de simulation.

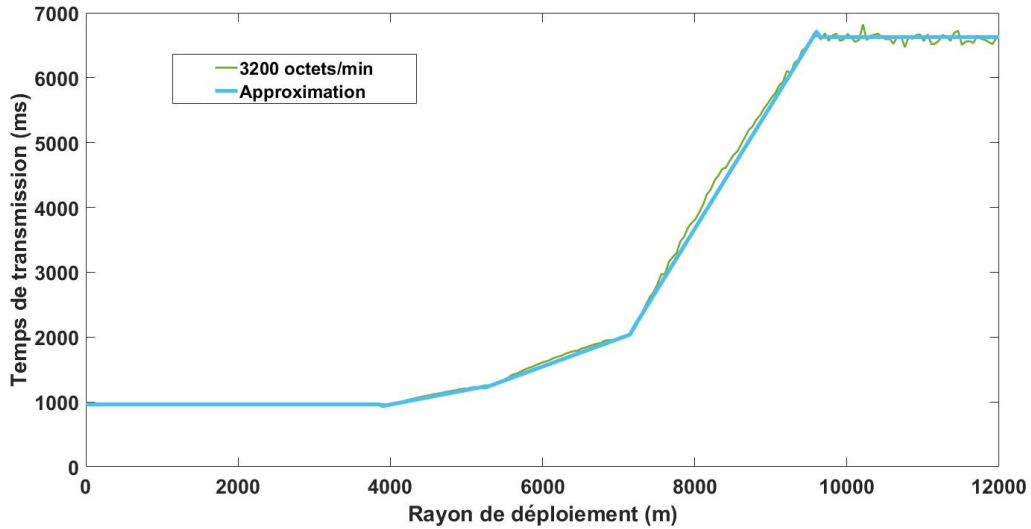


FIGURE 3.9 – Approximation du temps de transmission avec un débit de 3200 octets/min selon le rayon de déploiement.

$$T_{approx} = \begin{cases} 962 & d < 3900 \\ 0.225d + 59 & 3900 \leq d \leq 5200 \\ 0.42944d - 1032 & 5200 < d \leq 7150 \\ 1.9006d - 11536 & 7150 < d \leq 9600 \\ 6626 & d > 9600 \end{cases} \quad (3.9)$$

où :

$d$  : Rayon de déploiement (m).

$T_{tx}$  : Temps moyen de transmission (ms).

### 3.2.4 Taux de pertes par *path loss*

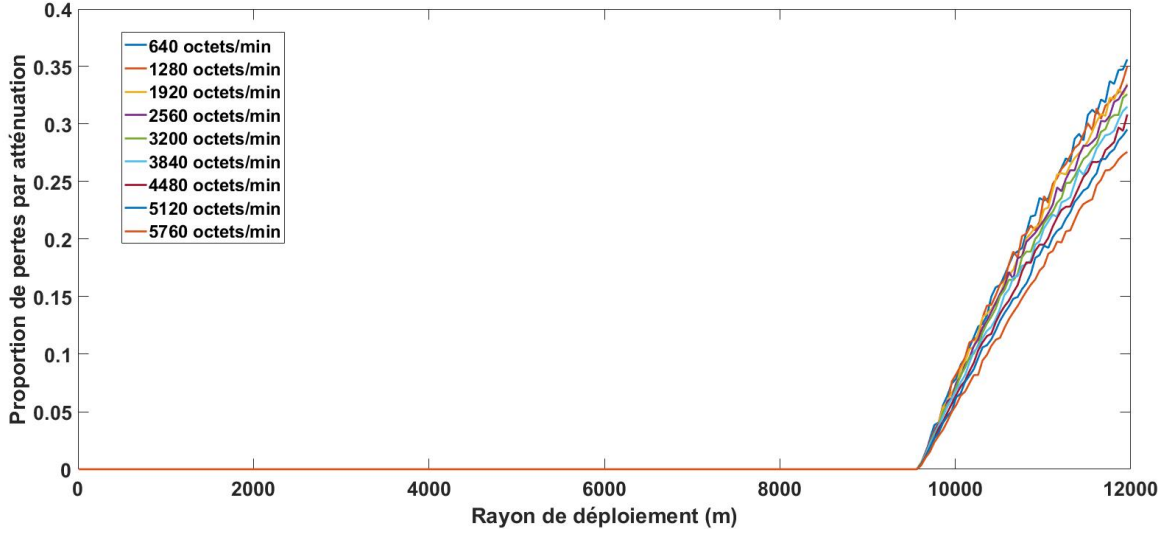


FIGURE 3.10 – Pertes par atténuation selon le débit de données et le rayon de déploiement.

La figure 3.10 illustre les résultats du taux de pertes par *path loss* ou par atténuation selon le nombre de noeuds et le rayon de déploiement. On constate qu’il y a des pertes par atténuation à partir de 9276 mètres et que le taux de croissance de ces pertes est plus important pour les plus petits nombres de noeuds. Le taux de croissance de pertes par atténuation varie de 1.48%/100 m à 1.15%/100 m.

La relation entre le taux croissance de pertes par atténuation et le débit de données est la suivante :

$$\tau_{PL} = -65.51e^{-6} \frac{\%}{100m * \text{octet}/\text{min}} \quad (3.10)$$

### 3.2.5 Taux de pertes par collision

La figure 3.11 présente les résultats de simulation pour le taux de pertes par collision selon le nombre de noeuds et le rayon de déploiement. On peut d’abord constater qu’il augmente avec le nombres de noeuds en opérations. On peut diviser le graphique en six sections où les courbes présentent des comportements différents.

#### Section 1 $0 \text{ m} < d < 900 \text{ m}$

Dans la section 1, les noeuds transmettent à puissance minimale et le *spreading factor* est minimal. Les taux de collisions sont constants et présentés dans le tableau 3.2. On observe une augmentation d’environ 0.5% par saut de 640 oct/min en débit.

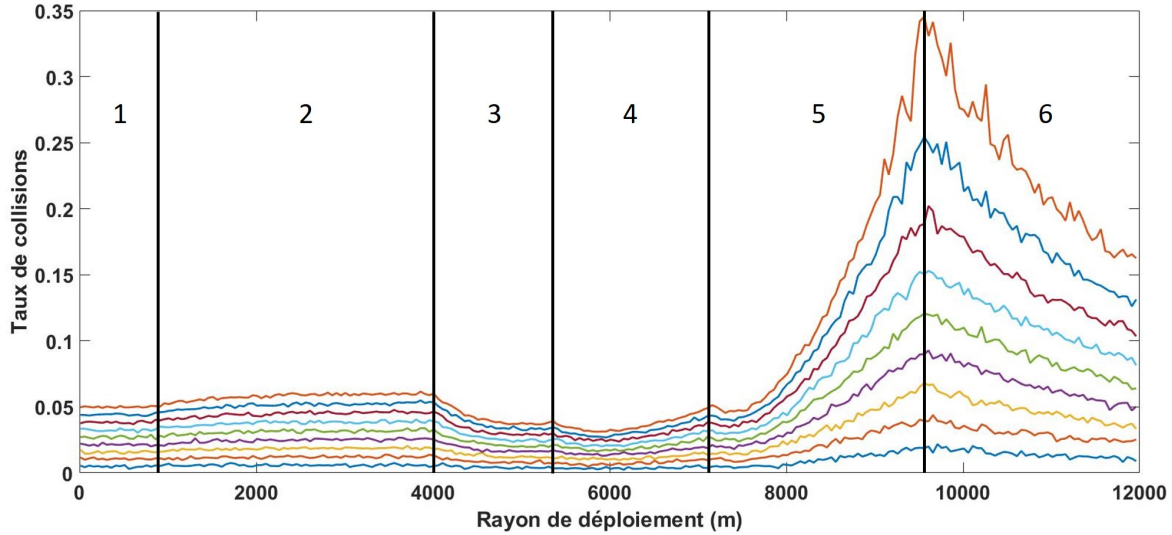


FIGURE 3.11 – Taux de collisions selon le débit de données et le rayon de déploiement.

TABLE 3.2 – Taux de collision de la section 1 et 2 selon le débit de données.

Débit (octets/min)	Taux de collision (%)	
	Section 1	Section 2
640	0.5	0.51
1280	1.0	1.32
1920	1.5	1.90
2560	2.1	2.55
3200	2.7	3.29
3840	3.2	3.90
4480	3.8	4.65
5120	4.5	5.36
5760	5.0	6.00

Nous obtenons cette équation pour approximer le taux de collision selon le débit de données pour la section 1 à partir des données du tableau 3.2.

$$c_{sec1} = 853.6e^{-6}R \quad (3.11)$$

où :

$c_{sec1}$  : Taux de collision section 1 (%).

$R$  : Débit de données (octet/min).

### Section 2 (900 m < $d$ < 4000 m)

À la section 2, on voit que le taux de collisions augmente avec le rayon de déploiement. Les noeuds qui sont situés au-delà de 660 mètres augmentent leur puissance de transmission afin



d'être reçu par la station de base. Cependant, la puissance de transmission est augmentée juste assez afin d'être reçue à la puissance minimale. Ce qui fait en sorte qu'il y a de plus en plus de paquets reçus à la même puissance à la station de base. Comme mentionné à la section 3.1.4, lorsqu'il y a collision entre deux paquets et ces paquets sont reçus à des puissances semblables, les deux paquets sont perdus. Ce qui explique l'augmentation du taux de pertes par collision à la section 2. Nous obtenons une nouvelle approximation linéaire du taux de collision dans la section 2 à partir des données du tableau 3.2.

$$c_{sec2} = 1.0325e^{-3}R \quad (3.12)$$

où :

$c_{sec2}$  : Taux de collision section 2 (%).

$R$  : Débit de données (octet/min).

### Section 3 (4000 m < $d$ < 5300 m)

Dans la section 3, on observe une baisse. Elle s'explique par la diversité du *spreading factor*. En effet, entre 3912 et 5216 mètres, le SF de 8 apparaît. Deux paquets reçus au même moment, mais avec des SF différents ne sont pas perdus. Nous pouvons donc traiter indépendamment les collisions selon leur SF et déterminer le taux de collisions global par la suite.

Nous y arrivons en additionnant les taux de collisions de chaque SF en les pondérant selon leurs proportions déterminées par (3.4).

$$c_{sec3} = c_{SF7} * \alpha_{SF7} + c_{SF8} * \alpha_{SF8} \quad (3.13)$$

où :

$c_{sec3}$  : Taux de collision section 3 (%).

$c_{SF7}$  : Taux de collision section SF7 (%).

$c_{SF8}$  : Taux de collision section SF8 (%).

$\alpha_{SF7}$  : Proportion des noeuds dans la section SF7.

$\alpha_{SF8}$  : Proportion des noeuds dans la section SF8.

Pour déterminer le taux de collision de la zone SF7, on utilise  $c_{sec2}$  et on ajuste le débit de données selon la proportion du SF. On fait la même chose du côté SF8 et on ajuste  $a_{sec2}$  avec le facteur de temps de transmission que nous obtenons du tableau 1.2.

$$c_{sec3} = c_{sec2} * \alpha_{SF7}^2 + c_{sec2} * T_{SF8} * \alpha_{SF8}^2 \quad (3.14)$$

En remplaçant  $c_{sec2}$  :

$$c_{sec3} = 1.0325R * \alpha_{SF7}^2 + 1.0325R * T_{SF8} * \alpha_{SF8}^2 \quad (3.15)$$

En simplifiant :

$$c_{sec3} = (\alpha_{SF7}^2 + \alpha_{SF8}^2 * T_{SF8})1.0325e^{-3}R \quad (3.16)$$

où :

$c_{sec3}$  : Taux de collision section 3 (%).

$\alpha_{SF7}$  : Proportion des noeuds dans la section SF7.

$\alpha_{SF8}$  : Proportion des noeuds dans la section SF8.

$R$  : Débit de données (octet/min).

$T_{SF8}$  : Facteur de temps de transmission de SF8.

#### Section 4 (5300 m < $d$ < 7100 m)

À la section 4, nous avons un phénomène semblable à celui de la section 3 avec l'apparition du SF de 9. On observe la décroissance du taux de collision en début de sections et une croissance en deuxième moitié. On procède de la même façon qu'à la section 3 pour déterminer l'équation du taux de collision.

$$c_{sec4} = (\alpha_{SF7}^2 + \alpha_{SF8}^2 * T_{SF8} + \alpha_{SF9}^2 * T_{SF9}) 1.0325e^{-3} R \quad (3.17)$$

où :

$c_{sec4}$  : Taux de collision section 4 (%).

$\alpha_{SF7}$  : Proportion des noeuds dans la section SF7.

$\alpha_{SF8}$  : Proportion des noeuds dans la section SF8.

$\alpha_{SF9}$  : Proportion des noeuds dans la section SF9.

$R$  : Débit de données (octet/min).

$T_{SF8}$  : Facteur de temps de transmission de SF8.

$T_{SF9}$  : Facteur de temps de transmission de SF9.

On trouve que le point minimum du taux de collision selon l'équation est à 5750 mètres.

#### Section 5 (7100 m < $d$ < 9550 m)

À la section 5, l'effet du temps de transmission sur les collisions surpasse l'effet de la diversité du *spreading factor*. On voit alors le taux de pertes par collision augmenter. L'apparition du SF 10 a pour effet de rapidement augmenter le temps de transmission, comme le montre la figure 3.8. On ne peut procéder comme aux sections 3 et 4 puisque le taux de collision est sous évalué. On doit ajuster le facteur relatif au débit afin d'obtenir de bons résultats.

$$c_{sec5} = (\alpha_{SF7}^2 + \alpha_{SF8}^2 * T_{SF8} + \alpha_{SF9}^2 * T_{SF9} + \alpha_{SF10}^2 * T_{SF10}) F * R \quad (3.18)$$

où :

$c_{sec5}$  : Taux de collision section 5 (%).

$\alpha_{SF7}$  : Proportion des noeuds dans la section SF7.

$\alpha_{SF8}$  : Proportion des noeuds dans la section SF8.

$\alpha_{SF9}$  : Proportion des noeuds dans la section SF9.

$\alpha_{SF10}$  : Proportion des noeuds dans la section SF10.

$F$  : Facteur débit vs collisions.

$R$  : Débit de données (octet/min)

$T_{SF8}$  : Facteur de temps de transmission de SF8.

$T_{SF9}$  : Facteur de temps de transmission de SF9.

$T_{SF10}$  : Facteur de temps de transmission de SF10.

La figure 3.12 illustre les approximations faites du taux de collision de la sections 1 à 5. On constate que les courbes se suivent assez bien et que ces équations pourront être utilisées afin de déterminer le taux de collision selon le débit de données et le rayon de déploiement. L'approximation de la section 6 ne sera pas faite puisqu'elle ne sera pas utile.

### Section 6 (9550 m < d)

À la section 6, des paquets commencent à être perdus par atténuation. Ces paquets ne peuvent contribuer aux collisions, ce qui explique la décroissance du taux. L'approximation de la section 6 ne sera pas faite puisqu'elle ne sera pas utile.

### Résumé

La figure 3.12 illustre les approximations faites du taux de collision de la sections 1 à 5. On constate que les courbes se suivent assez bien et que ces équations pourront être utilisées afin de déterminer le taux de collision selon le débit de données et le rayon de déploiement. Elles seront utiles aussi afin de déterminer le taux de réception total pour les premiers 9550 mètres.

$$\tau_{collision} = \begin{cases} c_{sec1} & 0 < d < 900 \text{ m} \\ c_{sec2} & 900 < d < 4000 \text{ m} \\ c_{sec3} & 4000 < d < 5300 \text{ m} \\ c_{sec4} & 5300 < d < 7100 \text{ m} \\ c_{sec5} & 7100 < d < 9550 \text{ m} \end{cases} \quad (3.19)$$

### 3.2.6 Taux de réception total

La figure 3.13 illustre le taux de réception total (voir 1.3.3) selon le débit de données et le rayon de déploiement. Le taux de réception est affecté par les pertes par collision et par atténuation. On constate que jusqu'à 9550 mètres, la courbe suit la courbe des pertes par collision. À 9550 mètres et plus, le taux de réception devient aussi affecté par les pertes par atténuation.

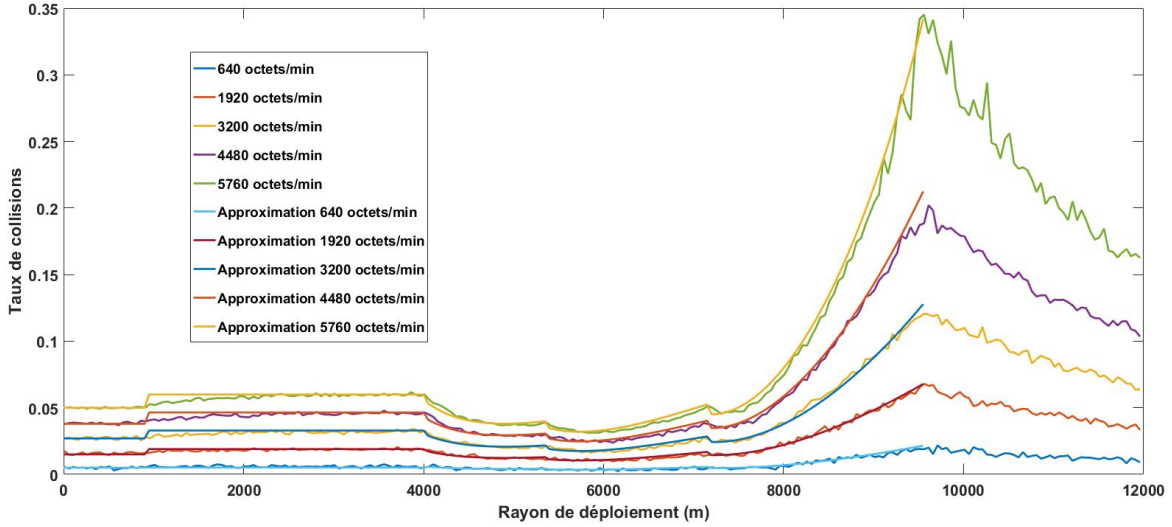


FIGURE 3.12 – Approximations du taux de collisions selon le rayon de déploiement et le débit de données.

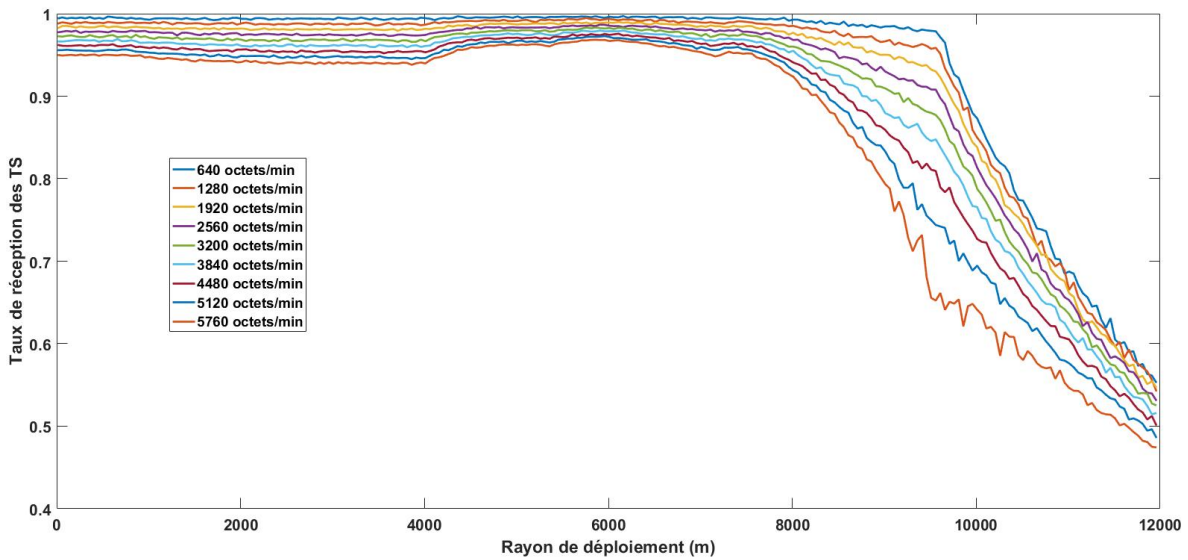


FIGURE 3.13 – Taux de réception total selon le débit de données et le rayon de déploiement.

On observe un point optimal de réception à 5750 mètres allant de 96,8% pour 5760 octets/min à 99,7% à 640 octets/min. À ce rayon de déploiement, on retrouve un minimum de collisions et il n'y a pas de pertes par atténuation.

Afin d'en faire une approximation, il suffit d'utiliser celle du taux de collisions jusqu'à 9550 mètres. Ensuite on ajoute l'approximation des pertes par atténuation. La figure 3.14 montre ce qui en résulte.

$$\tau_{reception} = 1 - \tau_{collision} \quad (3.20)$$

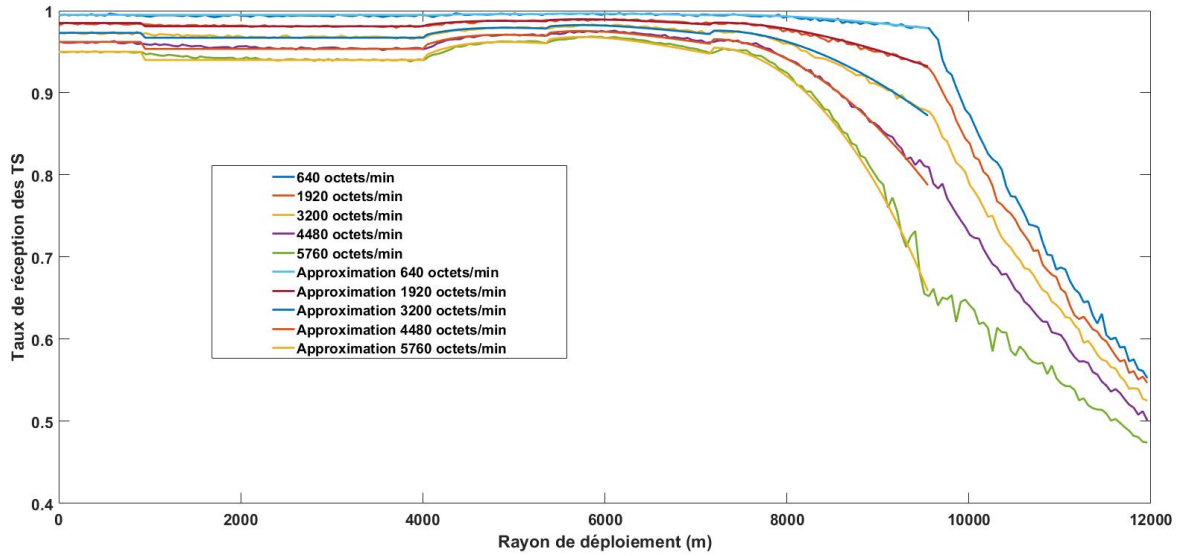


FIGURE 3.14 – Taux de réception total selon le débit de données et le rayon de déploiement.

### 3.2.7 Efficacité énergétique

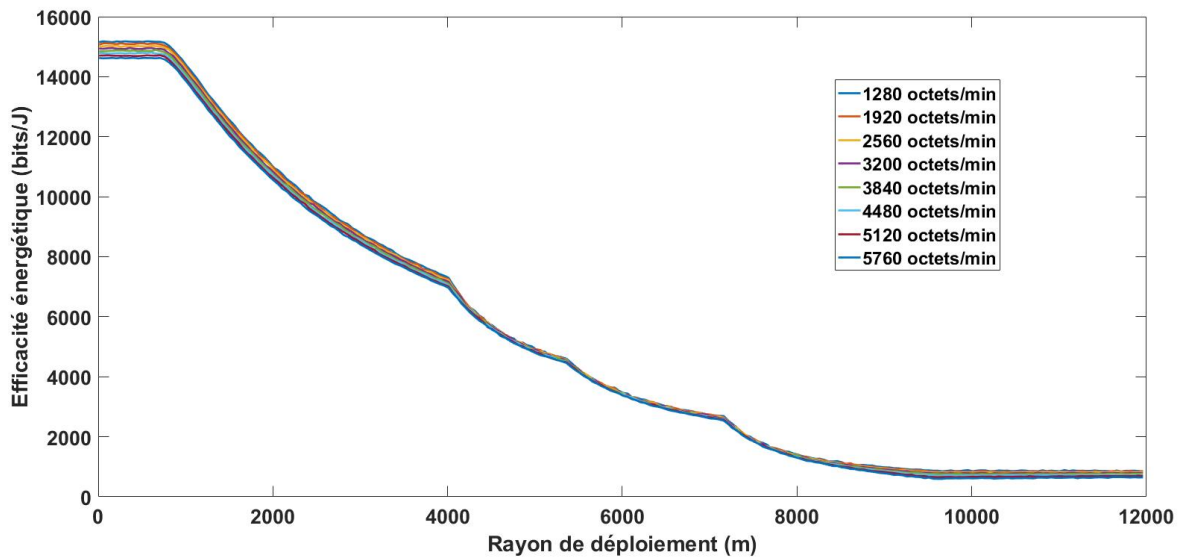


FIGURE 3.15 – Efficacité énergétique selon le débit de données et le rayon de déploiement.

La figure 3.15 illustre l'efficacité énergétique selon le débit de données et le rayon de déploiement. Selon l'équation (1.11) et l'allure des courbes, on constate l'influence du temps de transmission et de la puissance de transmission. On retrouve le comportement exponentiel de la puissance de transmission et les changements de gradient du temps de transmission. Le nombre de bits transmis, lui, est constant. Les courbes sont stables autour de 15 kbit/J jusqu'à 700 mètres où la puissance moyenne de transmission se met à augmenter. De 700 à 4000 mètres, l'efficacité diminue environ de moitié (48%). Avec l'apparition d'une nouvelle

valeur de SF à 4000 m, 5300 m et 7100 m, on voit que le gradient s'accroît tout comme celui du temps de transmission. À 9500 m, les courbes se stabilisent comme pour la puissance et le temps de transmission, autour de 750 bits/J.

L'équation (3.21) est une bonne approximation de la relation entre l'efficacité énergétique et le rayon de déploiement pour les différents débits de données de la simulation. On peut le constater à la figure 3.16.

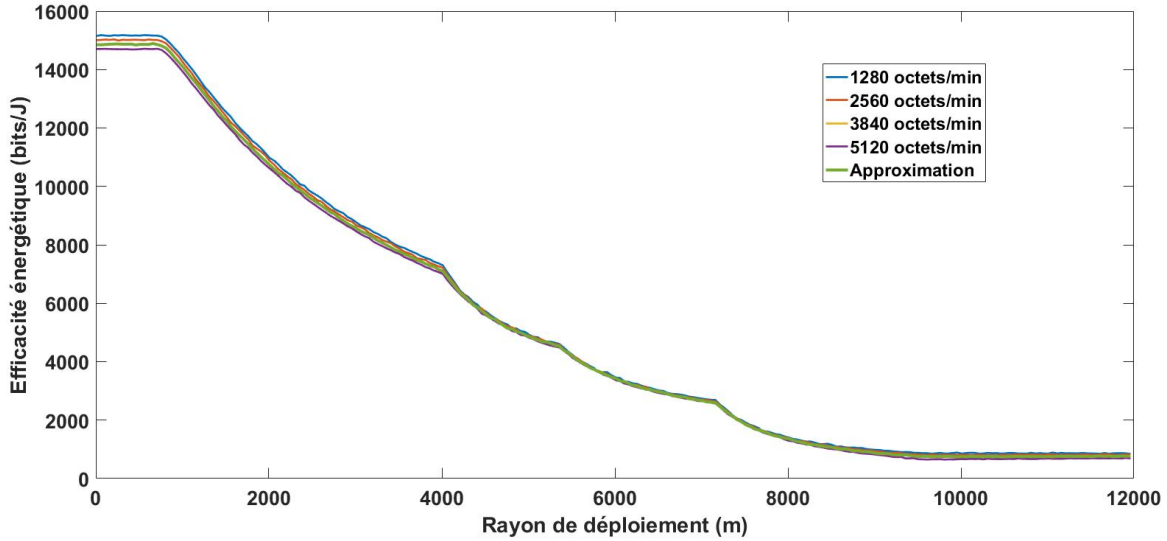


FIGURE 3.16 – Approximation de l'efficacité énergétique selon le rayon de déploiement.

$$Eb_{approx} = \begin{cases} 14870 & d < 910 \text{ m} \\ 0.0004824d^2 - 4.686d + 18250 & 910 \text{ m} \leq d \leq 3960 \text{ m} \\ 0.001111d^2 - 12.22d + 38150 & 3960 \text{ m} < d \leq 5360 \text{ m} \\ 0.0004827d^2 - 7.046d + 28340 & 5360 \text{ m} < d \leq 7160 \text{ m} \\ -1.968e^{-7}d^3 + 0.00529d^2 - 47.66d + 1.448e^5 & 7160 \text{ m} < d \leq 9510 \text{ m} \\ 753 & d > 9510 \text{ m} \end{cases} \quad (3.21)$$

### 3.2.8 Efficacité spectrale

La figure 3.17 illustre l'efficacité spectrale selon le débit de données et le rayon de déploiement. Au départ, on constate que l'efficacité spectrale est légèrement meilleure avec un bas débit. Elle reste constante jusqu'à 4000 mètres autour de 0.017 bits/s/Hz. De 4000 à 9600 mètres, elle diminue quasi linéairement pour atteindre un plateau autour de 0.0063 bits/s/Hz. En regardant de près, on reconnaît l'aspect exponentiel qui ressort à chaque ajout de SF, mais il ne sera pas pris en compte dans l'approximation puisqu'il apparaît négligeable.

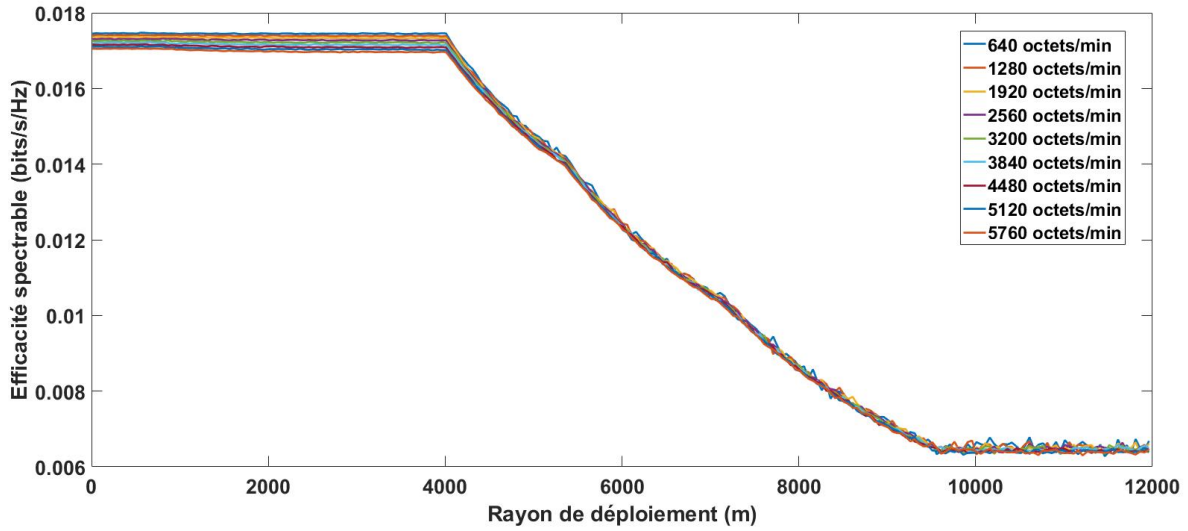


FIGURE 3.17 – Efficacité spectrale selon le débit de données et le rayon de déploiement.

L'équation (3.22) traduit la relation entre l'efficacité spectrale et le rayon de déploiement. La figure 3.18 montre cette approximation avec les données de la simulation.

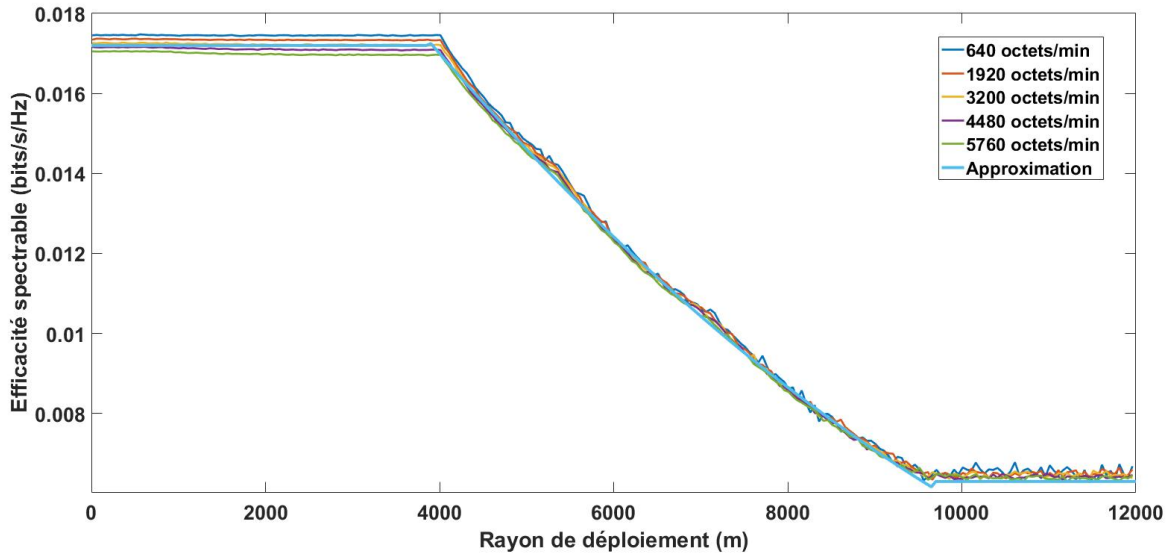


FIGURE 3.18 – Efficacité spectrale selon le débit de données et le rayon de déploiement.

$$\eta_{approx} = \begin{cases} 0.017 & d < 4000 \text{ m} \\ 1.022e^{-10}d^2 - 3.312e^{-6}d + 0.02861 & 4000 \text{ m} \leq d \leq 9550 \text{ m} \\ 0.0063 & d > 9510 \text{ m} \end{cases} \quad (3.22)$$

Nous avons maintenant tout ce qu'il nous faut afin de traiter des exemples de déploiement

dans le prochain chapitre.



# Chapitre 4

## Exemples de déploiement

Dans ce chapitre, on regarde deux exemples de déploiement du système étudié dans les chapitres précédents. On définit le contexte, l'environnement et les besoins puis on estime les performances à l'aide des approximations trouvées au chapitre 3. Les deux exemples traités sont la surveillance des sols en agriculture et celle du microbiome dans les territoires nordiques. Une seule station de base sera déployée dans les deux cas puisque le comportement à plusieurs stations de bases n'est pas abordé dans cet ouvrage. De plus, les paramètres de propagations resteront les mêmes que dans l'expérimentation terrain.

### 4.1 Agriculture

L'agriculture est un domaine où la mesure de l'environnement devient très importante. En y intégrant l'IoT, on crée l'agriculture intelligente qui, en réduisant les pertes et en augmentant la productivité, est plus efficace.

#### 4.1.1 Contexte

Dans ce scénario, on déploie une importante matrice de capteurs à travers des champs de culture. En monitorant l'environnement précisément, il est possible d'augmenter considérablement l'efficacité de l'agriculture. Pour le sol, on peut surveiller l'humidité, l'acidité et la température. Certaines zones peuvent s'assécher plus rapidement, d'autres peuvent geler plus lentement. On peut ensuite agir aux endroits et aux moments précis afin de préserver les conditions optimales. D'autres paramètres comme la luminosité peuvent être monitorés mais on ne peut les influencer. Cela permet au moins de prédire le volume de récolte pour mieux planifier la distribution par la suite.

### 4.1.2 Environnement

L'environnement est celle d'une ferme typique du Québec traversant quatre saisons. Le terrain est facile d'accès et le climat tempéré. Étant l'environnement dans lequel l'expérimentation terrain a pris place, notre modèle de propagation s'applique bien. L'aire des champs à couvrir est de  $100 \text{ km}^2$ , ce qui correspond environ à un rayon de déploiement de 5.6 km.

#### Besoins

Chaque système aura besoin de transmettre une fois par 15 minutes et aura un *payload* de 256 octets. Il faut avoir souvent des mesures afin de pouvoir agir rapidement au besoin. Il nous faut aussi avoir une bonne densité de capteurs afin de pouvoir agir de manière ciblée, par exemple, la température du sol. Pour une cannebergière à l'automne, il est important de protéger la culture du gel. Lorsque la température du sol descend trop, on arrose. Cependant, on ne veut pas arroser pour rien puisque l'eau doit être préservée pour la récolte et est en circuit fermé pour protéger l'environnement. L'intervention humaine n'est pas un problème, mais on souhaite tout de même la minimiser.

### 4.1.3 Performance estimée

Le tableau 4.1 présente le nombre de capteurs déployés, la densité de capteur ainsi que le débit de données associé selon la densité établie. L'équation suivante est un exemple de calcul de débit.

$$R_{350} = \frac{pa * n_{noeuds}}{t_{tx}} = \frac{256 * 350}{15} = 5973 \quad (4.1)$$

où :

$R_{350}$  : Débit de données à 350 nœuds (octet/min).

$pa$  : *Payload* (octet).

$n_{noeuds}$  : Nombre de nœuds.

$t_{tx}$  : Temps entre chaque transmission (min).

TABLE 4.1 – Paramètres de déploiement du scénario d'agriculture.

Nombre de capteurs	Densité (capteur/ $\text{km}^2$ )	Débit de données (octets/min)
350	3.5	5973
200	2.0	3413
50	0.5	853

#### Taux de réception total

Afin d'estimer le taux de réception total pour chacun des nombres de capteurs déployés, on utilise (3.20). Ayant un rayon de déploiement, on utilise (3.17) où seul le paramètre du débit

de données variera.

$$\tau_{agri} = 1 - c_{agri} \quad (4.2)$$

On définit le taux de collision selon (3.17).

$$c_{agri} = (\alpha_{SF7}^2 + \alpha_{SF8}^2 * T_{SF8} + \alpha_{SF9}^2 * T_{SF9})1.0325e^{-3}R \quad (4.3)$$

On calcule les proportions de capteurs de chaque section de SF avec (3.5) et on utilise les valeurs du tableau 1.2.

$$c_{agri} = (0.488^2 + 0.380^2 * 1.75 + 0.132^2 * 3.11)1.0325e^{-3}R \quad (4.4)$$

Le tableau 4.2 présente le taux de réception total selon les trois différents nombres de capteurs. On constate une différence de 3% entre le plus grand et le plus petit nombre. 96.6% est accep-

TABLE 4.2 – Taux de réception total selon le nombre de capteurs.

Nombre de capteurs	Taux de réception (%)
350	96.6
200	98.1
50	99.5

table tenant compte que le 3.4% perdu est retransmis jusqu'à ce qu'il soit reçu. On observe alors une latence plus importante, ce qui peut être sensible dans certaines circonstances, mais pas dans celle-ci. Le plus grand nombre de capteurs serait alors à privilégier.

### Efficacité spectrale

Notre approximation de l'efficacité spectrale donnée par (3.22) dépend seulement du rayon de déploiement. Nous avons donc une valeur unique.

$$\eta_{agri} = 1.022e^{-10}d^2 - 3.312e^{-6}d + 0.02861 \quad (4.5)$$

On a une efficacité spectrale de 0.01327bit/s/Hz.

### Efficacité énergétique

Comme pour l'efficacité spectrale, notre approximation donnée par (3.21) de l'efficacité énergétique dépend seulement du rayon de déploiement. Nous avons donc une valeur unique.

$$Eb_{agri} = 0.0004827d^2 - 7.046d + 28340 \quad (4.6)$$

On a une efficacité énergétique de 4020 bits/J.

## Conclusion

Les valeurs de l'efficacité spectrale et énergétique étant uniques, il nous reste à considérer le taux de réception total en opposition à la densité de capteurs. Nous avons observé un taux de réception plus bas avec 350 capteurs, mais les paquets perdus sont retransmis et la latence n'est pas un enjeu. On privilégie alors le plus grand nombre de capteurs pour le déploiement.

## 4.2 Grand Nord

Un autre exemple de déploiement traité dans cet ouvrage sera celui de l'EcoChip en région nordique. Il s'agit d'un système nécessitant de transmettre des données régulièrement et qui opère dans des régions difficiles d'accès.

### 4.2.1 Contexte

Dans [12], on explique qu'une façon de mesurer les changements climatiques et l'impact de la présence humaine dans un environnement est de monitorer les microorganismes qui y vivent. Pour ce faire, un prototype a été développé prenant plusieurs mesures sur son environnement et aussi sur les microorganismes. Ces paramètres sont la température, l'humidité, la luminosité, l'acidité, le taux de dioxyde de carbone pour l'environnement et la bio-impédance pour les microorganismes.

Les territoires nordiques sont visés puisque ce sont les premiers touchés par les changements climatiques et c'est à ces endroits que les effets sont les plus visibles. De plus, le microbiome de ces régions n'est pas bien connu. Lorsque le système est récupéré, il est possible d'en faire le séquençage afin d'identifier les microorganismes connus et d'étudier les inconnus.

Pendant que le système est sur place, il communique de manière sans-fil avec une station de base permettant d'avoir des mesures régulièrement. La figure 4.1, tirée de [12], présente un aperçu de l'EcoChip et de ses lieux de déploiement.

### 4.2.2 Environnement

L'environnement nordique est difficile, le froid étant le pire ennemi des batteries qui réduit considérablement leur capacité. De plus, les sites en étude sont difficiles d'accès et le personnel ne sera pas appelé à y aller régulièrement. En fait, le moins ils ont à s'y rendre, le mieux c'est. Il est donc primordial d'optimiser la consommation énergétique afin que le système puisse fonctionner le plus longtemps possible sans intervention humaine pour remplacer la batterie. Cependant, afin d'avoir des mesures représentatives, il faut couvrir une certaine superficie de territoire. Il y aura donc un compromis à faire.

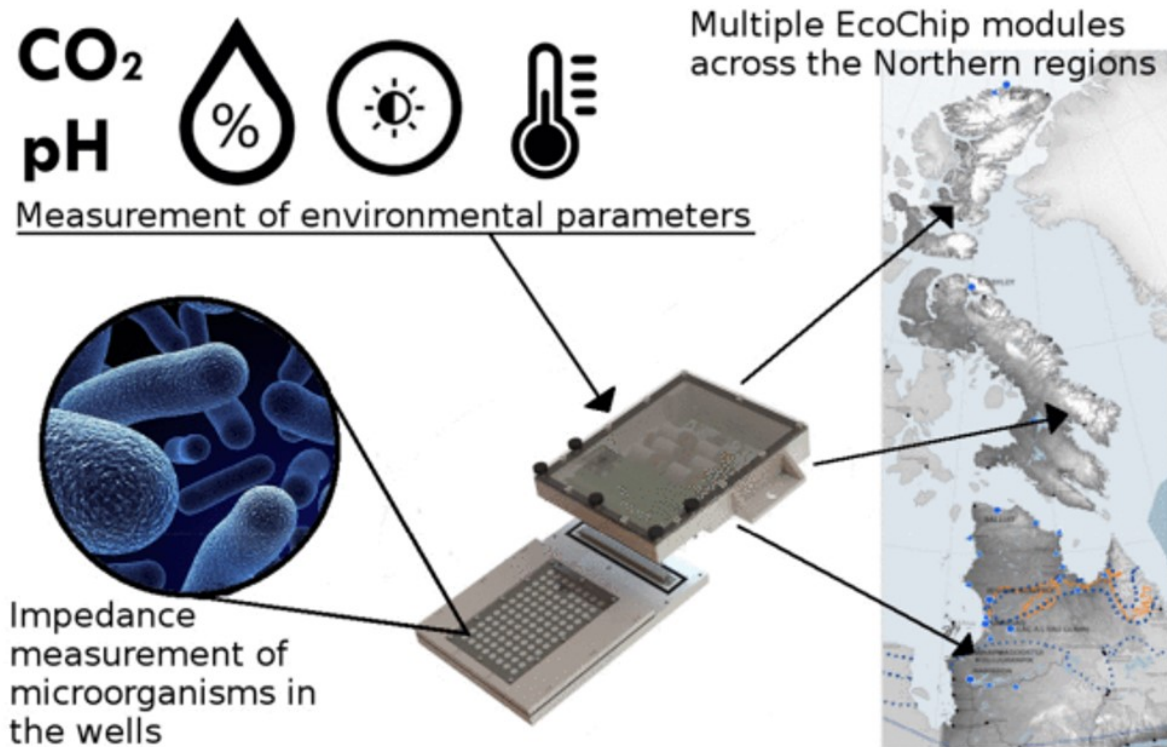


FIGURE 4.1 – Système EcoChip et ses lieux de déploiement.

TABLE 4.3 – Densité des noeuds selon le rayon de déploiement.

Rayon (km)	Aire (km <sup>2</sup> )	Densité (unités/km <sup>2</sup> )
4	50.3	9.9
5	78.5	6.4
7	153.9	3.3
9	254.4	2.0

### 4.2.3 Besoins

Chaque système aura besoin de transmettre une fois par heure et aura un *payload* de 256 octets. Supposons que nous avons 500 unités de ce système à déployer. Pour le rayon, nous verrons à le maximiser tout en considérant la consommation énergétique. Nous compareront quatre différents rayons, soient 4, 5, 7 et 9 km. Ces quatre rayons correspondent aux limites des différentes valeurs de SF.

### 4.2.4 Performances estimées

Le tableau 4.3 présente la densité de systèmes selon les quatre rayons de déploiement. Ayant 500 unités transmettant 256 octets chaque heure, on obtient un débit de données de 2130 octe/min.

## Taux de réception total

Afin d'estimer le taux de réception total pour chacun des rayon de déploiement, on utilise (3.20). Découlant de l'approximation du taux de collisions, le résultat dépendra de la section dans laquelle chaque rayon se situe.

Pour le rayon de 4 km on regarde la section 2 de la figure 3.11. En utilisant (3.20) et (3.12), on obtient :

$$\tau_{r1} = 1 - \frac{1.0325R}{100} = 1 - \frac{1.0325 * 2.13}{100} = 0.978 = 97.8\% \quad (4.7)$$

Pour le rayon de 5 km on regarde la section 3 de la figure 3.11. En utilisant (3.20) et (3.16) on obtient :

$$\tau_{r2} = 1 - \frac{(\alpha_{SF7}^2 + \alpha_{SF8}^2 * T_{SF8})1,0325R}{100} = 0.986 = 98.6\% \quad (4.8)$$

Pour le rayon de 7 km on regarde la section 4 de la figure 3.11. En utilisant (3.20) et (3.17) on obtient :

$$\tau_{r3} = 1 - \frac{(\alpha_{SF7}^2 + \alpha_{SF8}^2 * T_{SF8} + \alpha_{SF9}^2 * T_{SF9})1,0325R}{100} = 0.982 = 98.2\% \quad (4.9)$$

Pour le rayon de 9 km on regarde la section 5 de la figure 3.11. En utilisant (3.20) et (3.18) on obtient :

$$\tau_{r4} = 1 - \frac{(\alpha_{SF7}^2 + \alpha_{SF8}^2 * T_{SF8} + \alpha_{SF9}^2 * T_{SF9} + \alpha_{SF10}^2 * T_{SF10})FR}{100} = 0.941 = 94.1\% \quad (4.10)$$

On résume les résultats dans le tableau 4.4.

TABLE 4.4 – Taux de réception estimé selon le rayon de déploiement.

Rayon (km)	$\tau$ (%)
4	97.8
5	98.6
7	98.2
9	94.1

On voit que le taux de réception total reste à l'intérieur de 1% pour les trois premiers rayons de déploiement et chute ensuite de 4%.

## Efficacité spectrale

Pour estimer l'efficacité spectrale de chaque déploiement, on utilise (3.22). On constate un facteur d'environ 2.5 entre le rayon de 4 et 9 km. Ces régions n'étant pas densément peuplé, les canaux n'ont pas tendance à saturer.

TABLE 4.5 – Efficacité spectrale estimée selon le rayon de déploiement.

Rayon (km)	$\eta$ (bit/s/Hz)
4	0.0170
5	0.0146
7	0.0104
9	0.0070

### Efficacité énergétique

Pour estimer l'efficacité énergétique de chaque déploiement, on utilise (3.21). On observe que

TABLE 4.6 – Efficacité énergétique estimée selon le rayon de déploiement.

Rayon (km)	$E_b$ (bit/J)
4	7046
5	4825
7	2670
9	1466

l'énergie consommée pour envoyer un bit est 5 fois plus importante avec un rayon de 9 km comparativement à 4 km. L'autonomie de batterie des nœuds avec un rayon de 9 km s'en trouverait grandement réduite. Pour optimiser ce paramètre, le rayon de 4 km est à privilégier.

### Conclusion

Pour le taux de réception total, la latence n'étant pas un enjeu, les résultats sont acceptables pour tous les rayons de déploiement. Pour l'efficacité spectrale, le spectre n'est pas saturé ce qui diminue l'importance de ce paramètre. Pour l'efficacité énergétique on constate une différence majeure selon le rayon de déploiement. Le rayon de 4 km est à privilégier afin de maximiser l'autonomie des batteries et minimiser l'intervention humaine dans ces régions difficiles d'accès.

# Conclusion

L'Internet des objets permet de connecter toute sortes d'objets et d'en connaître l'état à tout moment, permettant des actions précises et ciblées et créant d'importantes possibilités d'optimisation dans une panoplie de domaines.

Cette connectivité doit être dans la plupart des cas sans fil et les protocoles de communication dans les dernières années ont été optimisés afin de maximiser le débit de données et de minimiser la latence. En IoT, pour les régions rurales et éloignées, nous visons plutôt à optimiser la consommation énergétique afin de maximiser l'autonomie des objets. Le débit de données et la latence ne sont pas des enjeux.

Deux exemples ont été utilisés pour illustrer l'IoT en milieu rural et éloigné. Un premier est en agriculture. En monitorant l'environnement précisément, il est possible d'augmenter considérablement l'efficacité de l'agriculture. Pour le sol, on peut surveiller l'humidité, l'acidité, la température. Certaines zones peuvent s'assécher plus rapidement, d'autres peuvent geler plus lentement. On peut ensuite agir aux endroits et aux moments précis afin de préserver les conditions optimales. D'autres paramètres comme la luminosité peuvent être monitorés mais on ne peut les influencer. Cela permet au moins de prédire le volume de récolte pour mieux planifier la distribution par la suite.

Le deuxième exemple d'IoT en milieu éloigné se trouve dans [12] où on explique qu'une façon de mesurer les changements climatiques et l'impact de la présence humaine dans un environnement est de monitorer les microorganismes qui y vivent. Pour ce faire, un prototype a été développé prenant plusieurs mesures sur son environnement et aussi sur les microorganismes. Ces paramètres sont la température, l'humidité, la luminosité, l'acidité, le taux de dioxyde de carbone pour l'environnement et la bio-impédance pour les microorganismes. Les territoires nordiques sont visés puisque ce sont les premiers touchés par les changements climatiques et c'est à ces endroits que les effets sont les plus visibles. De plus, le microbiome de ces régions n'est pas bien connu. Lorsque le système est récupéré, il est possible d'en faire le séquençage afin d'identifier les microorganismes connus et d'étudier les inconnus. Pendant que le système est sur place, il communique de manière sans-fil avec une station de base permettant d'avoir des mesures régulièrement. Plus son autonomie est grande, plus longtemps le système peut fonctionner sans intervention humaine et plus de données peuvent être recueillies.



Au chapitre 1, nous avons décrit la technologie LPWAN. Nous avons ensuite fait un tableau comparatif de différentes technologies de cette catégorie. Symphony Link de Link Labs a été retenu pour sa simplicité de déploiement et puisque que peu ou pas de travaux ont été fait à son sujet. Nous avons expliqué son fonctionnement en profondeur, autant sa modulation de signal que son protocole de communications. Nous avons aussi établi trois paramètres importants pour notre réseau : le taux de réception total, l'efficacité spectrale et l'efficacité énergétique.

Au chapitre 2, une expérimentation terrain a été réalisée en environnement rural. La structure du système a été expliquée, soit le serveur, la station de base et le nœud. Le nœud est un prototype qui a été conçu spécialement pour l'expérimentation permettant la mesure embarquée du courant consommé lors de la transmission et du nombre de paquets retransmis. Sa conception et ses itérations ont été détaillées. Plusieurs mesures ont été prises à différents endroits ce qui a permis l'élaboration d'une carte. Après l'analyse des données recueillies, un modèle d'atténuation de l'environnement de l'expérimentation et un modèle de consommation d'énergie de transmission ont été établis.

Au chapitre 3, des simulations de réseaux à une station de base et à plusieurs nœuds ont été réalisées. Un modèle basé sur une autre technologie [1] a été adapté, pour représenter notre technologie. Nous avons utilisé nos résultats de l'expérimentation terrain pour déterminer les paramètres de simulation. Plusieurs résultats ont été présentés et analysés pour différents nombres de nœuds et différents rayons de déploiement : la puissance de transmission, la distribution du *spreading factor*, le temps de transmission, le taux de pertes par atténuation, le taux de pertes par collisions, le taux de réception total, l'efficacité spectrale et l'efficacité énergétique. Nous avons pu apprécier l'effet que le débit de données peut avoir sur les différents paramètres. L'effet de la diversité du *spreading factor* vient améliorer le taux de réception total pour certaines plages de rayon de déploiement, mais y nuit pour d'autres à cause de l'augmentation du temps de transmission. Des équations d'approximation ont été établies pour chacun des paramètres permettant une évaluation rapide selon le rayon de déploiement et le nombre de nœuds du réseau.

Au chapitre 4, deux exemples de déploiement sont présentés dans lesquels nous avons utilisé les équations d'approximation. Le premier vise une application en agriculture où l'on doit recueillir une importante quantité de données. Il a été déterminé que l'on peut augmenter la densité de nœuds afin d'optimiser la résolution des mesures. Le deuxième exemple est l'Ecochip dans le Grand Nord. Dans ce cas-ci, la consommation d'énergie a dû être optimisée afin de minimiser les interventions humaines dans ce milieu isolé et difficile d'accès. Il a été déterminé que le rayon de déploiement doit être petit.

On peut résumer les contributions de cet ouvrage par une série de mesures qui permet de déterminer le coefficient de *pathloss* dans un environnement rural et la consommation d'énergie d'un appareil IoT commercial. On retrouve aussi une série de simulation qui génère une

série de géométries de déploiements aléatoires pour une surface de couverture fixe. Elle utilise un environnement de simulation publique en Python modifié pour calculer les mesures de performance paramétrées par les données recueillies expérimentalement. MATLAB est utilisé pour trouver une équation polynomiale de certaines mesures de performance. On retrouve ensuite une prédiction de deux scénarios simples de déploiement. Pour l'un, la superficie de couverture est fixée et il y a trois choix pour le nombre de noeuds. Pour l'autre, le nombre de noeuds est fixé et quatre choix de superficie de couverture sont disponibles. Certains indicateurs de performance sont présentés pour les deux scénarios en utilisant les équations polynomiales.

Il pourrait être pertinent pour la suite d'intégrer le comportement des batteries afin de bien évaluer le potentiel d'autonomie énergétique que pourrait avoir ces noeuds dans le temps et selon l'environnement dans lequel ils se trouvent. La simulation d'un plus grand réseau de cette technologie avec plusieurs stations de base et un système de contrôle central serait aussi intéressante.

## Annexe A

### Code de récupération de données

```
import conductor
import serial
import re
import numpy as np, scipy.io
import future
import sys

account = conductor.ConductorAccount('username', 'password')

appToken = 'd250de211dd48b7b1fb1'
#msgToRetrieve = 5
msgToRetrieve = int(sys.argv[1])

app_token = account.get_application_token(appToken)
```

```

#ser = serial.Serial(port='COM8', baudrate=9600)
#ser.isOpen()

print('Will retrieve the next '+ str(msgToRetrieve) + ' messages received on App token ' + appToken
)
print('0% \r'),
Data = np.array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0])
#print(Data)

cnt = 0;

for msg in app_token.subscribe_iter():
    if (cnt >= msgToRetrieve):
        print('\n')
        break
    strMsg = str(msg)
    #print(strMsg)
    m = re.search('payload_hex=(.+?)0a', strMsg)
    r = re.search('4a(.+?)0a', strMsg)
    s = re.search('48(.+?)0a', strMsg)
    t = re.search('4b(.+?)0a', strMsg)
    k = re.search('4c(.+?)0a', strMsg)
    n = re.search('spreading_factor=(.+?)', strMsg)
    o = re.search('snr=(.+?)', strMsg)
    p = re.search('rssi=(.+?)', strMsg)
    q = re.search('frequency=(.+)', strMsg)

```

```

if m:
    hexPayload = str(m.group(1))[1:]
    #print(hexPayload + '\r\n')
    #decPayload = str(int(hexPayload, 16))
    #print(decPayload + '\r\n')
    strPayload = hexPayload.decode("hex")
    #ser.write('Payload = ' + strPayload)

if r:
    hexCnt = str(r.group(1))
    #print('HexCnt = ' + hexCnt)
    #decCnt = str(int(hexCnt, 16))
    strCnt = hexCnt.decode("hex")
    #print('Cnt = ' + strCnt)

if s:
    hexPulseCnt = str(s.group(1))
    #print('HexCnt = ' + hexCnt)
    #decCnt = str(int(hexCnt, 16))
    strPulseCnt = hexPulseCnt.decode("hex")
    #print('Cnt = ' + strCnt)

if t:
    hexTimePulse = str(t.group(1))
    #print('HexCnt = ' + hexCnt)
    #decCnt = str(int(hexCnt, 16))
    strTimePulse = hexTimePulse.decode("hex")
    #print('Cnt = ' + strCnt)

if k:
    hexInterCnt = str(k.group(1))

```

```

#print('HexCnt = ' + hexCnt)
#decCnt = str(int(hexCnt, 16))
strInterCnt = hexInterCnt.decode("hex")
#print('Cnt = ' + strCnt)

if n:
    SF = str(n.group(1))
    #ser.write('SF = ' + SF + '\r\n')

if o:
    snr = str(o.group(1))
    #ser.write('SNR = ' + snr + '\r\n')

if p:
    RSSI = str(p.group(1))
    #ser.write('RSSI = ' + RSSI + '\r\n')

if q:
    freq = str(q.group(1))[:-2]
    #ser.write('Frequency = ' + freq + '\r\n\r\n')
    fPayload = float(strPayload)
    iCnt = int(float(strCnt))
    iPulseCnt = int(float(strPulseCnt))
    iTimePulse = int(float(strTimePulse))
    iInterCnt = int(strInterCnt)
    iSF = int(SF)
    fSnr = float(snr)
    fRSSI = float(RSSI)
    iFreq = int(freq)
    Temp = np.array([fPayload, iCnt, iPulseCnt, iTimePulse, iInterCnt, iSF, fSnr, fRSSI, iFreq])
    Data = np.vstack((Data, Temp))

```

```

cnt = cnt + 1

print(str(int(cnt/float(msgToRetrieve) * 100)) + '% \r'),

filename = raw_input('Enter name of file where to save the data. \r\n
Path and extension will be added auto: \r\n')
filename = 'C:\\users\\Francis\\Google Drive\\Master\\IoT\\tests\\7-06-2017\\' + filename + '.mat'

scipy.io.savemat(filename, mdict={'Data': Data})
print('Done')
print('Data saved in ' + filename)
#print(fPayload + ',' + iSF + ',' + fSnr + ',' + fRSSI + ',' + iFreq)

```

## Annexe B

### Code du prototype

```
#include <msp430.h>
#include <stdio.h>
#include <string.h>
#include "SymphonyLink.h"
#include "UART.h"

/*
 * main.c
 */

#define TRESHOLD 35

unsigned int ii;
char mes[300];
unsigned int adc_value;
```



```

float fAvg = 0;
float fTemp = 0;
int cnt = 1;
int validCurrent = 0;
char sep[2] = {'J', '\0'};
char sep2[2] = {'H', '\0'};
char sep3[2] = {'K', '\0'};
char sep4[2] = {'L', '\0'};
int pulseCnt = 0;
int pulseTmpCnt = 0;
int pEnable = 0;
int idleCnt = 0;
int timePulse = 0;
int tmpTimePulse = 0;
int interCnt = 0;

int main(void) {
    WDCTL = WDIPW | WDIHOLD; // Stop watchdog timer
    /*****Setting Master clock to 8 MHz*****/
    UCSCTL3 = SELREF__REFOCLK; // FLL clock source set to REFOCLK 32768 Hz
    UCSCTL4 |= SELA__REFOCLK; // ACLK clock source set to REFOCLK 32768 Hz
    UCSCTL0 = 0; // Clear UCS

    do
    {
        UCSCTL7 &= ~( XT2OFFG + XT1LFOFFG + DCOFFG ); // Clear XT2 oscillator fault flag of the
        HF mode

```

```

// Clear XT1 oscillator fault flag of the
// LF mode
// Clear the digitally controlled
// oscillator fault flag
// Clear the oscillator fault interrupt flag
// Test the oscillator fault flag

SFRIFG1 &= ~(OFIFG);
}while (SFRIFG1&OFIFG);

UCSCTL1 = DCORSEL_6;
UCSCTL2 |= 243;

// Set the DCO frequency range 4.6 to 10.7 MHz
// (Value + 1) * 32768 = 8 MHz (7.9953MHz)

P7DIR |= BIT7;
P7SEL |= BIT7;

//P7.7 set as output
// Peripheral module function selected (8 MHz to P7.7)

/***** Init ADC module*****/
unsigned int i = 0;

P6DIR &= ~BIT0;
P6SEL |= BIT0;

//Settin P6.0 as input
// Setting peripheral function active

REFCTL0 &= ~REFMSTR;
ADC12CTL0 = (ADC12ON + ADC12SHT02 + ADC12REFON + ADC12REF2_5V);
ADC12CTL1 = ADC12SHP;
ADC12MCTL0 = ADC12SREF0;

for ( i = 0x30 - 1; i == 0; i-- );

```

```

ADC12CTL0 |= ADC12ENC;

/*****Timer configuration *****/
TA0CTL0 = CCIE;
TA0CCR0 = 1600; //Sampling at 8MHz / 1600 =
TA0CTL = (TASSEL1 + MC0 + TACLR);

__bis_SR_register(GIE);

/*****Symphony link *****/
__delay_cycles(100000);

SymphonyLink symLink;

char txData[256]; //TX data buffer, Max 256 bytes
uint8_t uTxData[256];
//uint8_t rxData[128]; //RX data buffer, Max 128 bytes

sym_module_state_t currentSymphonyState;
sym_module_state_t lastSymphonyState;

/*****Setup*****/

uart_init(UART_DEBUG); //Initialize UART port for Debug

//Set desired network token
uint32_t netToken = 0x4f50454e;

```

```

//Set desired application token
uint8_t appToken[APP_TOKEN_LEN] = {0xd2,0x50,0xde,0x21,0x1d,0xd4,0x8b,0x7b,0x1f,0xb1};

//Initialize the symphony link object and open UART communication with module.
symlink.begin(netToken, appToken, LL_DL_MAILBOX,15);

//Set antenna
symlink.setAntenna(1); //1: External antenna, 2:PCB antenna.

//Initialize the txData.
for(ii = (sizeof(txData)); ii == 0; ii --)
{
    txData[ii] = 0;
}

//Update the state of the SymphonyLink module (aka Modem)
lastSymphonyState = symlink.updateModemState();

/*****Main loop*****/
while(1)
{
    fTemp = 0.1509*adc_value - 0.0171;

    if (fTemp > TRESHOLD)
    {
        pulseTmpCnt++;
    }
}

```

```

if (pulseTmpCnt >= 10)
{
    tmpTimePulse = pulseTmpCnt;
    if (pEnable)
    {
        idleCnt = 0;
        pulseCnt++;
        pEnable = 0;
    }
}
fAvg += fTemp;
cnt++;
fTemp = 0;
}
else
{
    idleCnt++;
    if (idleCnt >= 2)
    {
        timePulse += tmpTimePulse;
        tmpTimePulse = 0;
        pulseTmpCnt = 0;
        pEnable = 1;
    }
}

validCurrent = 0;

```

```

//Update the state of the SymphonyLink Module
currentSymphonyState = symlink.updateModemState();
switch (currentSymphonyState)
{
case SYMPHONY_READY:
    if (SYMPHONY_TRANSMITTING != lastSymphonyState)
    {
        strcpy(txData, float2str(fAvg/cnt));
        strcat(txData, sep);
        strcat(txData, int2str(cnt));
        strcat(txData, sep2);
        strcat(txData, int2str(pulseCnt));
        strcat(txData, sep3);
        strcat(txData, int2str(timePulse));
        strcat(txData, sep4);
        strcat(txData, int2str(interCnt));
        fAvg = 0;
        cnt = 1;
        pulseCnt = 0;
        timePulse = 0;
        interCnt = 0;

        for (i = sizeof(txData) - 1 ; i ==0 ; i--)
        {
            uTxData[i] = (uint8_t) txData[i];
        }
        symlink.write(uTxData, sizeof(txData), TRUE);
    }
}

```

```

    }
    break;
default: break;
}
lastSymphonyState = currentSymphonyState;
}
}

#pragma vector=TIMER0_A0_VECTOR
__interrupt void TIMER0_A0_ISR(void)
{
    ADC12CTL0 |= ADC12SC;
    while (!(ADC12IFG & BIT0));
    adc_value = ADC12MEM0;
    validCurrent = 1;
    interCnt++;
}

```

# Bibliographie

- [1] Bor, M. U. Roedig T. Voigt et J. Alonso. 2016, «Do LoRa low-power wide-area networks scale?», dans *The 19th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems*.
- [2] de Carvalho Silva, J., J. Rodrigues, A. M. Alberti, P. Solic et A. L. L. Aquino. 2017, «LoRaWAN x2014; A low power WAN protocol for Internet of Things : A review and opportunities», dans *2017 2nd International Multidisciplinary Conference on Computer and Energy Science (SpliTech)*, p. 1–6.
- [3] Georgiou, O. et U. Raza. 2017, «Low Power Wide Area Network Analysis : Can LoRa Scale?», *IEEE Wireless Communications Letters*, vol. 6, n° 2, doi :10.1109/LWC.2016.2647247, p. 162–165, ISSN 2162-2337.
- [4] Grenier, D. 2016, *Antennes et propagation radio*, Université Laval.
- [5] Knyazev, N. S., V. A. Chechetkin et D. A. Letavin. 2017, «Comparative analysis of standards for Low-power Wide-area Network», dans *2017 Systems of Signal Synchronization, Generating and Processing in Telecommunications (SINKHROINFO)*, p. 1–4, doi :10.1109/SINKHROINFO.2017.7997528.
- [6] Lampe, J. et Z. Ianneli. 2003, «Introduction to Chirp Spread Spectrum (CSS) Technology», IEEE 802.15-03-0460-00-0000.
- [7] Link-Labs. 2016, *Symphony Link™ vs. LoRaWAN™*. URL [http://info.link-labs.com/hubfs/LPWAN\\_Technology\\_Explained.pdf?hsCtaTracking=f2c3a059-1ffd-47c2-8a0c-8df6b587bd14%7Cd9ae18ba-9d7c-4d4c-8cf5-ec7196cea97b](http://info.link-labs.com/hubfs/LPWAN_Technology_Explained.pdf?hsCtaTracking=f2c3a059-1ffd-47c2-8a0c-8df6b587bd14%7Cd9ae18ba-9d7c-4d4c-8cf5-ec7196cea97b).
- [8] Rappaport, T. S. et collab.. 1996, *Wireless communications : principles and practice*, vol. 2, Prentice Hall PTR New Jersey.
- [9] Raza, U., P. Kulkarni et M. Sooriyabandara. 2017, «Low Power Wide Area Networks : An Overview», *IEEE Communications Surveys Tutorials*, vol. 19, n° 2, doi :10.1109/COMST.2017.2652320, p. 855–873, ISSN 1553-877X.



- [10] Scott, A. 2015, *8 ways the Internet of things will change the way we live and work*. URL <https://www.theglobeandmail.com/report-on-business/rob-magazine/the-future-is-smart/article24586994/>.
- [11] Semtech. 2015, *AN1200.22 LoRa™ Modulation Basics*. URL <http://www.semtech.com/images/datasheet/an1200.22.pdf>.
- [12] Sylvain, M., F. Lehoux, S. Morency, F. Faucher, E. Bharucha, D. M. Tremblay, F. Raymond, D. Sarrazin, S. Moineau, M. Allard, J. Corbeil, Y. Messaddeq et B. Gosselin. 2018, «The EcoChip : A Wireless Multi-Sensor Platform for Comprehensive Environmental Monitoring», *IEEE Transactions on Biomedical Circuits and Systems*, vol. 12, n° 6, doi :10.1109/TBCAS.2018.2878404, p. 1289–1300.