

THÈSE

En vue de l'obtention du
DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par l'Université Toulouse 2 - Jean Jaurès

Présentée et soutenue par
Nicolas GONZALEZ

Le 16 octobre 2020

**Architectures protocolaires interopérables pour le réseau de
collecte de l'Internet des Objets**

Ecole doctorale : **EDMITT - Ecole Doctorale Mathématiques, Informatique et
Télécommunications de Toulouse**

Spécialité : **Informatique et Télécommunications**

Unité de recherche :

IRIT : Institut de Recherche en Informatique de Toulouse

Thèse dirigée par

Thierry VAL et Adrien VAN DEN BOSSCHE

Jury

M. Michel MISSON, Rapporteur

M. François SPIES, Rapporteur

Mme Nathalie MITTON, Examinatrice

M. Laurent TOUTAIN, Examinateur

Mme Réjane DALCE, Examinatrice

Mme Katia JAFFRES-RUNSER, Examinatrice

M. Thierry VAL, Directeur de thèse

M. Adrien VAN DEN BOSSCHE, Co-directeur de thèse

Remerciements

Je tiens à remercier Monsieur le Professeur émérite Michel Misson et Monsieur le Professeur François Spies pour l'intérêt qu'ils ont porté à mes travaux en acceptant de les rapporter.

Je remercie tout particulièrement Monsieur le Professeur *Thierry Val* ainsi que Monsieur le Docteur Habilité *Adrien Van Den Bossche* pour avoir accepté de prendre la direction de mes travaux. Je vous remercie pour votre confiance, votre écoute, vos conseils, et de façon générale votre bienveillance à l'égard de ma personne et de mes travaux. Je vous remercie de m'avoir accordé l'autonomie qui m'a permis de découvrir, d'apprendre et de m'essayer au métier d'enseignant et de chercheur.

Je remercie vivement *Benjamin Freeman* avec qui j'ai partagé un bout de cette thèse en tant qu'ami, co-bureau, collègue, binôme et maintenant en tant qu'associé dans l'entreprise OperaMetrix que nous avons co-fondée. Ta fidélité, ta patience et ta franchise m'enrichissent autant sur le plan personnel que professionnel. Je te remercie pour ta confiance et pour ton aide dans ces épreuves.

Je remercie du fond du coeur *Julien Bresciani* et *Laurent Guerby* qui ont fait beaucoup pour moi et principalement dans le cadre de cette thèse. Souvent dans l'ombre et sans en récolter les honneurs vous agissez pour rendre les choses possibles. Je tiens aujourd'hui à vous remercier pour tout cela et vous prie de croire en ma reconnaissance infinie à l'affection que vous m'avez apportée.

Je remercie l'ensemble du personnel de l'IUT de Blagnac pour son accueil chaleureux et sa sympathie le long de ce bout de chemin que l'on a fait ensemble. Votre confiance m'a permis de m'épanouir sur de nombreux aspects, dont l'enseignement. Je remercie tout particulièrement *Rémi Boulle* qui m'a permis de réaliser des enseignements et d'en créer de nouveaux sur mes thématiques de recherche. Ta confiance m'a profondément touché.

Je remercie mes amis et ma famille qui m'accompagnent dans tous les challenges que je tente de relever. L'indisponibilité physique et souvent cérébrale, la charge de travail, les phases de stress ou encore les nombreuses heures à travailler en dehors des périodes acceptables vous font supporter les côtés négatifs de mes choix. Vous êtes pourtant toujours là pour m'accompagner, me soutenir, m'encourager dans ces étapes de ma vie que je ne pourrais franchir sans vous. Et pour tout cela, je vous remercie tendrement de votre patience et de votre confiance dans la sincérité de mon amour pour vous.

Je tiens à remercier *Pauline* qui subit au quotidien ma charge de travail et mon implication forte dans de nombreux projets. Tu mériterais ton nom sur la première page de cette thèse tant ta contribution morale m'a été nécessaire. Ta gentillesse, ta patience et ta tolérance sont d'une rareté dont je prends conscience un peu plus chaque jour qui passe.

Table des matières

Remerciements	3
Introduction	9
1 L'Internet des Objets pour l'industrie et les enjeux de l'interopérabilité	11
1.1 Définition et origines	13
1.1.1 Définition	13
1.1.2 Origines	19
1.2 L'Internet des Objets pour l'industrie (IIoT)	22
1.2.1 Vers de nouvelles formes d'organisation des entreprises	22
1.2.2 Stratégie de valorisation des données	24
1.2.3 La problématique de l'interopérabilité	25
1.3 État de l'art scientifique sur l'interopérabilité	27
1.3.1 Sémantique et encodage des données	27
1.3.2 Architectures interopérables pour l'IoT	28
1.3.3 Protocoles de transport des données	28
1.3.4 Encapsulation des messages	29
1.3.5 Architecture IoT event-driven	29
1.4 Les concepts et composants de base de l'IoT	31
1.4.1 La couche d'acquisition des informations	32
1.4.2 La couche de collecte des informations	36
1.4.3 La couche de traitement des informations	40
1.5 Première piste d'architecture interopérable	42
1.5.1 Implémentation du modèle de graphe de flux	42
1.5.2 Limitation de Node-RED	43
1.5.3 Conclusion	43
1.6 Système de messagerie pour l'Internet des Objets	43

1.6.1	Le format et le type des messages	44
1.6.2	Représentation au format JSON	44
1.6.3	Limitations des capacités des objets	45
1.6.4	Nécessité d'un intermédiaire	45
1.7	Garantir l'interopérabilité à l'aide du protocole MQTT	46
1.7.1	Intérêts	46
1.7.2	Paradigme Publish/Subscribe	46
1.7.3	Implémentation des mots clés dans MQTT	47
1.7.4	Implémentation de la couche transport	48
1.7.5	Utilisation de MQTT dans le cadre de l'Internet des Objets	48
1.7.6	Couche de transport augmentée en espace utilisateur	49
1.7.7	Gestion du multicast par la QoS	49
1.8	Conclusion	51
2	Les réseaux LPWAN : originalités, architectures et déploiement	53
2.1	Origines de la démarche de recherche	55
2.1.1	Développement actif d'entreprises autour des LPWAN	55
2.1.2	Appropriation des LPWAN par le milieu académique	55
2.1.3	Les promesses des réseaux LPWAN	56
2.2	État de l'art scientifique sur les LPWAN	57
2.2.1	Période 2015/2016 : les premières publications sur les LPWAN et la couche physique LoRa	57
2.2.2	Période 2017/2018 : début de cette thèse et intensification des publications	57
2.2.3	Période 2019/2020 : déploiement des premiers réseaux de recherche	58
2.2.4	Réseaux maillés avec couche physique LoRa	58
2.3	Législation des bandes ISM	58
2.4	Évaluation des paramètres de la couche physique	60
2.4.1	Idée de départ : du multi-saut au multi-médium	60
2.4.2	Modulation LoRa	60
2.4.3	Codage de canal dans la couche physique LoRa	61
2.4.4	Temps d'émission et débit	62
2.5	Composants d'infrastructure	65
2.5.1	Passerelle LPWAN	65
2.5.2	Serveur de réseau	66
2.5.3	Serveurs applicatifs	66
2.6	Réseaux LPWAN privés et publics	68

2.6.1	Réseaux LPWAN publics	68
2.6.2	Réseaux LPWAN privés	69
2.7	Protocoles de transport des messages	69
2.7.1	Packet Forwarder	70
2.7.2	Transport des trames via MQTT	73
2.8	De la théorie à la pratique	73
3	Proposition d'une architecture de <i>testbed</i> urbain ouvert pour les LPWAN	75
3.1	Origines du projet	76
3.1.1	<i>Testbed</i> urbain	76
3.1.2	L'association Tetaneutral.net	76
3.1.3	Lancement du projet	76
3.2	Proposition d'architecture d'un <i>testbed</i> urbain ouvert	77
3.2.1	Passerelle LoRa	77
3.2.2	Réseau de collecte des trames LoRa	79
3.2.3	Concept de commutation au niveau du bus logiciel	81
3.3	Programmation de nouveaux réseaux LoRa	82
3.3.1	Programmation de nouveaux services réseau pour les LPWAN	82
3.3.2	Programmation des interactions avec les objets	84
3.4	Architecture globale du <i>testbed</i> urbain	86
3.4.1	Services réseau du <i>testbed</i> urbain	86
3.5	Cas d'usages et valorisation scientifique du <i>testbed</i>	87
3.5.1	Cas d'usages	89
3.5.2	Valorisation	89
4	Vers l'extensibilité d'un réseau LoRa	91
4.1	Évaluation des performances de propagation de la modulation LoRa	93
4.1.1	Travaux préalables de découverte de la couche physique	93
4.1.2	Bilan de liaison d'une transmission LoRa	95
4.1.3	Modèle de propagation	96
4.1.4	Campagne de mesures	101
4.1.5	Analyse des conditions de collision entre les trames LoRa	104
4.1.6	Modélisation des processus aléatoires de fading	105
4.2	Simulation d'un accès au médium sur une topologie mono-passerelle	106
4.2.1	Objectifs de la simulation	106
4.2.2	Hypothèses instinctives d'étude	108

4.2.3	Topologie d'étude	108
4.2.4	Simulateur à évènements discrets	108
4.2.5	Simulation d'un accès multiple à une gateway LoRa	110
4.2.6	Paramétrage de la simulation	111
4.2.7	Disposition aléatoire des nœuds	111
4.2.8	Choix statique de paramétrage du spreading factor	112
4.2.9	Analyse du taux de livraison des trames (FDR)	113
4.3	Optimisation du taux de livraison des trames (FDR)	115
4.3.1	Critère d'optimisation	115
4.3.2	Méthode d'optimisation CMA	116
4.3.3	Proposition d'une nouvelle méthode d'optimisation par bandes successives	116
4.4	Étude de l'extensibilité de l'accès au médium des réseaux LoRa	120
4.4.1	Limite d'extensibilité par canal	120
4.4.2	Limite d'extensibilité globale	120
4.5	Algorithme de réglage automatique des paramètres de la couche physique	122
4.5.1	Automatic Data Rate de LoRaWAN	122
4.5.2	Les trois zones comportementales	123
4.5.3	Proposition d'algorithme	125
4.5.4	Conclusion et perspectives de ces travaux	128
	Conclusion générale et perspectives de recherche	131
	Bibliographie	135
	Glossaire	141
	Table des figures	143
	Liste des tableaux	147
	Liste des publications	148

Introduction

Les années 2000 ont été marquées par la démocratisation massive de l'utilisation du réseau Internet. L'intégration de l'accès aux données mobiles dans les téléphones cellulaires permet à chaque citoyen d'avoir un accès permanent en tous lieux au réseau mondial. Cette connectivité devient omniprésente avec la création de nombreuses plateformes d'échanges, de messagerie, d'achats et de partage d'informations. Cet engouement nécessite le développement de technologies de communication toujours plus rapides et de serveurs performants. Cette montée en charge favorise le développement du concept de "Cloud Computing", couche d'abstraction des infrastructures qui deviennent provisionnables automatiquement en fonction de l'utilisation. Ces infrastructures permettent d'envisager la collecte, le stockage et la manipulation de volumes de données de plus en plus importants. Les techniques de Machine Learning et d'Intelligence Artificielles développées depuis les années 60 redeviennent au goût du jour avec l'espoir qu'un volume de données important ait des propriétés statistiques permettant de modéliser des tendances et de prendre des décisions.

Dans un même temps, la communauté scientifique étudie les réseaux de capteurs sans fil avec l'idée de rendre les objets électroniques, communicants avec l'informatique de façon à collecter des données issues du monde physique. Les applications industrielles sont nombreuses comme la surveillance des procédés, l'observation de l'environnement, la logistique ou encore la robotique. Les applications grand public sont moins nombreuses mais se développent autour du smartphone principalement avec les technologies Bluetooth et Wi-Fi. C'est l'émergence des capteurs sportifs, des casques audio, des capteurs de surveillance de la maison ou des plantes, ou encore des interactions via des QR codes ou des tags NFC. La plupart de ces objets communicants fournissent leurs services sur un réseau local, industriel ou personnel. Le développement du réseau Internet et des infrastructures de la donnée, comme évoqué au paragraphe précédent, font émerger le concept d'Internet des Objets (IoT), un réseau mondial où les objets, les humains et les applications informatiques peuvent communiquer ensemble. La traduction de ce concept a été faite par la création d'une nouvelle catégorie de réseaux sans-fil, les réseaux LPWAN pour "Low Power Wide Area Network" qui visent à être l'équivalent des réseaux mobiles cellulaires mais adaptés aux objets. En effet, les objets transmettent des données principalement de capteurs, avec une périodicité d'émission suffisamment faible pour être économes en énergie. Pour cela, les réseaux LPWAN sont définis afin d'envoyer des messages courts, sur une longue distance et avec un bilan énergétique très faible. Comme pour les réseaux mobiles, la couverture doit être globale de façon à rendre les applications de mobilités viables.

Les technologies radios doivent pour exister faire des compromis très importants au niveau de l'économie d'énergie, du débit, de la portée, de la sécurité ou encore de la fiabilité. Les applications sont souvent très exigeantes et nécessitent le choix d'une ou plusieurs technologies adaptées. Aucune technologie assez reconfigurable n'émerge de façon à unifier l'ensemble de ces compromis. Dès lors, les équipes de recherche s'orientent sur la piste de l'interopérabilité. Lorsque les applications requièrent de faire des compromis non compatibles, les objets deviennent plus complexes en intégrant plusieurs technologies radio et en changeant de mode de fonctionnement suivant l'usage.

Voici le contexte dans lequel les travaux issus de cette thèse s'intègrent. L'approche de l'Internet des Objets ne peut se faire sans cette vision globale des enjeux et de l'historique du développement des technologies. Nous avons fait le choix de garder cette approche globale, tout en implémentant nos contributions sur une technologie en particulier, les réseaux LoRa. La technologie LoRa permet un accès très aisé aux composants matériels et logiciels autant en terme financier qu'en terme technique. Le transmetteur radio est le seul composant de l'architecture qui est protégé ce qui a permis la création de briques libres pour chacun des autres composants.

La structure de ce mémoire est la suivante : nous détaillons dans le premier chapitre le contexte dans lequel ces travaux ont été effectués. Nous redéfinissons de manière académique le concept d'Internet des Objets en insistant sur les différents enjeux et sous-concepts qui le composent. Ce chapitre constitue les bases théoriques nécessaires à la compréhension de nos contributions.

Dans le second chapitre, nous focalisons l'étude sur les réseaux LPWAN et plus particulièrement les réseaux LoRa. Ce chapitre est plus technique et permet d'observer l'application à un cas concret des concepts et des enjeux définis. Nous détaillons les composants logiciels et matériels nécessaires au déploiement d'une infrastructure de gestion d'un réseau LPWAN.

Le chapitre trois présente la contribution majeure de cette thèse, le développement de l'architecture et le déploiement d'une plateforme d'enseignement et de recherche pour l'étude des réseaux LoRa. Cette plateforme est constituée de plusieurs dizaines de passerelles sur la ville de Toulouse, soit un terrain d'expérimentation de niveau métropolitain. Notre contribution scientifique consiste à proposer une méthode et une architecture pour l'interopérabilité de ces réseaux.

Le chapitre quatre montre comment grâce, à cette infrastructure, nous avons pu réaliser une étude sur la couverture et l'extensibilité des réseaux LPWAN. Le déploiement d'une infrastructure d'expérimentation dans un environnement réel permet de faire des campagnes d'expérimentations pour tester un nouveau protocole, un nouveau paramétrage de la couche physique ou encore une nouvelle architecture logicielle.

Chapitre 1

L'Internet des Objets pour l'industrie et les enjeux de l'interopérabilité

La récolte d'informations issues du terrain par un très grand nombre d'objets communicants, est un challenge que de nombreuses entreprises cherchent à relever afin d'optimiser leur productivité. Cette récolte est réalisée par des capteurs de différentes natures, avec des systèmes embarqués très différents en terme d'énergie et de capacités de calcul. Ces systèmes cherchent à converger vers l'Internet, le réseau IP, au travers d'autres réseaux qui ont tous des performances, des portées et des débits très inégaux. L'interopérabilité est la capacité que vont avoir ces objets à être capable de communiquer entre eux malgré la forte hétérogénéité des technologies de communication.

Nous allons dans ce chapitre présenter les origines de l'Internet puis son élargissement aux objets connectés. Après un tour d'horizon des différents enjeux et architectures actuelles, nous expliquerons les choix de notre étude dans le cadre de cette thèse.

Sommaire

1.1	Définition et origines	13
1.1.1	Définition	13
1.1.2	Origines	19
1.2	L'Internet des Objets pour l'industrie (IIoT)	22
1.2.1	Vers de nouvelles formes d'organisation des entreprises	22
1.2.2	Stratégie de valorisation des données	24
1.2.3	La problématique de l'interopérabilité	25
1.3	État de l'art scientifique sur l'interopérabilité	27
1.3.1	Sémantique et encodage des données	27
1.3.2	Architectures interopérables pour l'IoT	28
1.3.3	Protocoles de transport des données	28
1.3.4	Encapsulation des messages	29
1.3.5	Architecture IoT event-driven	29

1.4	Les concepts et composants de base de l'IoT	31
1.4.1	La couche d'acquisition des informations	32
1.4.2	La couche de collecte des informations	36
1.4.3	La couche de traitement des informations	40
1.5	Première piste d'architecture interopérable	42
1.5.1	Implémentation du modèle de graphe de flux	42
1.5.2	Limitation de Node-RED	43
1.5.3	Conclusion	43
1.6	Système de messagerie pour l'Internet des Objets	43
1.6.1	Le format et le type des messages	44
1.6.2	Représentation au format JSON	44
1.6.3	Limitations des capacités des objets	45
1.6.4	Nécessité d'un intermédiaire	45
1.7	Garantir l'interopérabilité à l'aide du protocole MQTT	46
1.7.1	Intérêts	46
1.7.2	Paradigme Publish/Subscribe	46
1.7.3	Implémentation des mots clés dans MQTT	47
1.7.4	Implémentation de la couche transport	48
1.7.5	Utilisation de MQTT dans le cadre de l'Internet des Objets	48
1.7.6	Couche de transport augmentée en espace utilisateur	49
1.7.7	Gestion du multicast par la QoS	49
1.8	Conclusion	51

1.1 Définition et origines

1.1.1 Définition

1.1.1.1 De l'Internet à l'Internet des objets

Le réseau Internet tel que nous le connaissons aujourd'hui en 2020 et qui fait partie de notre quotidien (smartphones, tablettes, ordinateurs etc.) est un concept assez complexe à appréhender dans son fonctionnement comme dans les enjeux sociaux associés.

Le réseau Internet puise ses racines dans les années 1960-1970 lorsque le développement de l'informatique dans les grandes universités et les grandes administrations a imposé la nécessité d'échanger et de partager des informations.

Les systèmes informatique à cette époque sont très loin de ce que nous connaissons aujourd'hui, les «mainframe» étaient des machines très volumineuses et très chères. Seules les plus grandes universités, centres de recherche et administrations étaient capables d'investir l'argent nécessaire dans l'achat d'un mainframe comme le CERN (figure 1.1). La complexité de construction de ces systèmes et leur faible nombre ont fait que le marché était assez restreint autour de quelques constructeurs (Bull, IBM, etc.).



FIGURE 1.1 – Le mainframe du CERN en octobre 1963 (<http://information-technology.web.cern.ch/about/computer-centre/computing-history>)

L'enjeu majeur à cette époque réside sur le matériel. Historiquement, les premiers développeurs infor-

matique qui concevaient des programmes pour ces systèmes échangeaient les programmes et les astuces associées afin de permettre de faire avancer la discipline. Les financiers et les juristes n'étant à l'époque pas inquiétés par ces échanges car leurs regards étaient fixés autour du matériel et de son financement.

Dès lors le besoin d'échanges entre ces systèmes informatique est apparu, principalement entre les sites militaires pour le contrôle-commande des sites de lancement d'ogives nucléaires. En 1961 l'US Air Force confie à l'ARPA (aujourd'hui DARPA) le développement d'un réseau entre les sites de commandement des bombardements nucléaires. Dans le cahier des charges du projet figure la notion de résilience du réseau afin de ne pas rendre l'ensemble des points de commandement hors d'usage en cas d'attaque nucléaire sur l'un de ces points. C'est pour cela que des travaux ont été menés pour concevoir le premier réseau à commutation de paquets afin de ne pas avoir d'architecture centralisée.

Ce réseau a été nommé ARPANET et le premier paquet envoyé est «lo», le 29 octobre 1969, car le système a planté avant de pouvoir finir d'écrire «login».

Très vite les universités américaines ont été connectées entre elles via ce réseau ARPANET. De 23 noeuds en 1971, le réseau est passé à 111 en 1977. La figure 1.2 représente la carte de ce réseau à l'époque.

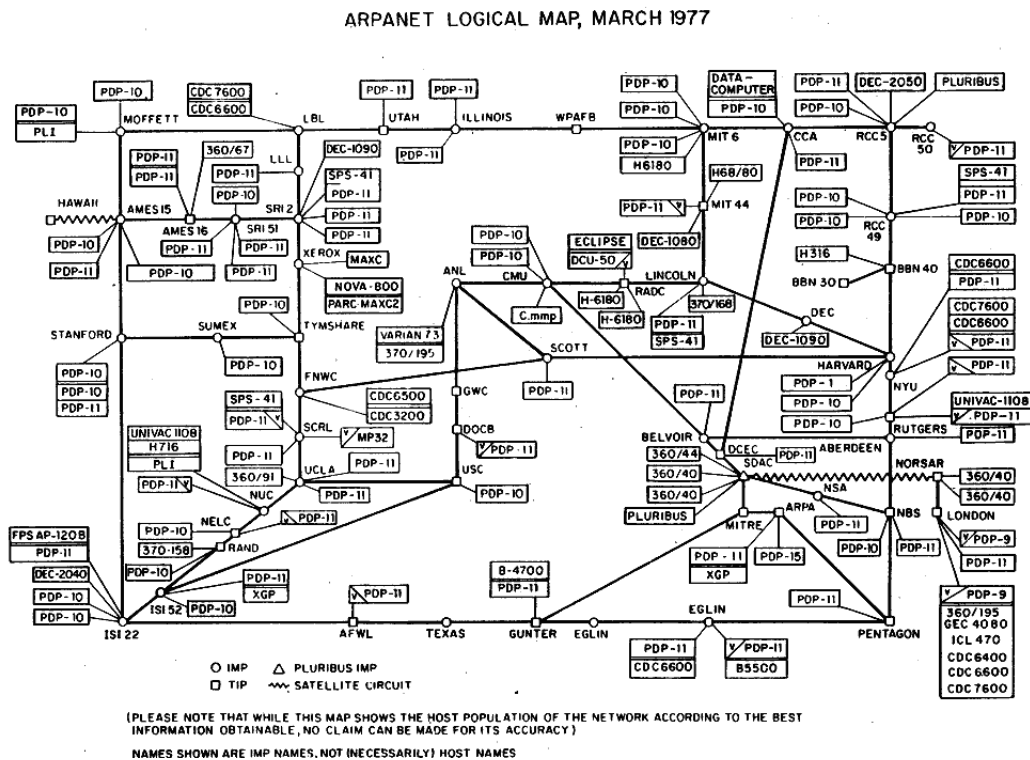


FIGURE 1.2 – La carte logique du réseau ARPANET en 1977 (<https://fr.wikipedia.org/wiki/ARPANET>)

Ce qui est important de comprendre à ce moment là de l'histoire d'Internet est que l'architecture décentralisée d'ARPANET qui a été pensée pour sa résilience est devenue une architecture intéressante pour les universités. En effet, l'architecture décentralisée permet de ne pas avoir de prise de pouvoir d'une université sur les autres d'un point de vue technique car si le réseau d'une université est en panne, le réseau reste fonctionnel pour les autres car le réseau est maintenu par l'état et donc considéré comme «équitable» face aux différentes universités.

Le deuxième point important est que les technologies (TCP, HTTP etc...) qui ont été développées dans le cadre de ce réseau n'ont pas été brevetées et sont restées ouvertes. Cette logique que l'on va retrouver dans l'Internet des Objets est importante car elle garantit la dissémination des techniques d'accès au réseau et permet ainsi d'être équitable.

Dans les années 1970 et 1980, ce réseau va grossir et devenir national aux États-Unis, puis mondial. C'est la naissance d'Internet, un réseau global, non centralisé, sans gouvernance et où les acteurs locaux peuvent participer au réseau et à son infrastructure.

La popularité d'Internet devient réelle à partir de la création du web et plus particulièrement du protocole HTTP au début des années 1990 par Tim Berners-Lee et Robert Cailliau au CERN. Ce protocole va permettre les communications avec des images et du texte dans un navigateur ce qui facilite l'accès au réseau et au partage d'informations aux personnes moins qualifiées en informatique ou en réseau.

La navigation sur Internet reste compliquée par l'absence de moteur de recherche afin de trouver l'URL des sites que l'on veut consulter et de rechercher des sites web en fonction de leur contenu.

Plusieurs moteurs de recherche voient le jour comme Yahoo! en 1994, Altavista en 1995 puis Google en 1998.

À ce moment de l'histoire, le développement des processeurs et du matériel informatique en général replace le logiciel au centre des attentions des financiers et des juristes. De nouvelles entreprises de logiciels sont apparues comme Microsoft en 1975 ou Apple en 1976. Ces entreprises développent des logiciels et les revendent sous la forme de licence d'exploitation. C'est un virage fort car c'est le début du capitalisme du logiciel par opposition à la vision universelle et basée sur les communs des années 1970. Les juristes ont travaillé de façon acharnée pour faire accepter le logiciel comme une œuvre intellectuelle qui appartient à son développeur ou à l'entreprise qui embauche le développeur par le biais du copyright. Ce copyright lui donne le droit de définir les conditions d'utilisation et de distribution de ce logiciel. En réponse à ce mouvement d'enfermement des logiciels, Richard Stallman, célèbre développeur informatique issu du MIT publie en 1989, la première version de la «GNU Public License». Cette première licence de logiciel libre se base habilement sur le concept de copyright non pas pour restreindre les droits sur le logiciel mais pour les libérer. L'auteur du logiciel cède ainsi ses droits à la communauté et autorise la distribution, la modification et l'exécution de ces logiciels à n'importe quelle personne.

À partir des années 2003, Internet prend un nouveau virage lorsque les grandes entreprises s'aperçoivent que ce ne sont plus les logiciels qui ont de la valeur mais les informations collectées par leur biais.

C'est l'arrivée des géants du web avec Myspace (2003), Facebook (2004), Youtube (2005) et Twitter (2006). Le *business model* de ces entreprises est principalement basé sur la collecte et la revente des données. On passe d'un modèle du web où l'on fournit un service souvent gratuit aux clients en échange de bandeaux publicitaires pour financer le service, à un modèle où les clients fournissent gratuitement les données au service qui les revend à d'autres entreprises. Les internautes sont devenus des fournisseurs de données et plus les clients principaux des acteurs majeurs du web.

1.1.1.2 Objets

La notion d'objet dans le contexte de l'Internet des Objets renvoie à la définition d'objet communicant car un objet physique devra être en mesure de communiquer pour exister sur le réseau Internet.

Definition 1.1.1. Objet communicant : Système physique constitué au strict minimum d'un moyen de communication afin d'échanger des informations avec d'autres objets ou des équipements réseaux. Un objet est généralement constitué de capteurs et/ou d'actionneurs qui permettent d'interagir avec l'environnement en fonction des informations traitées.

Definition 1.1.2. Capteur : Organe qui élabore, à partir d'une grandeur physique, une autre grandeur physique, souvent de nature électrique, utilisable à des fins de mesure ou de commande.

Definition 1.1.3. Actionneur : Appareil ou organe permettant d'agir sur une machine ou un processus en vue de modifier son comportement ou son état.

Les capteurs, les actionneurs et les interfaces de communication sont souvent des circuits standards qui sont assemblés ensemble via un circuit imprimé (PCB). Afin de pouvoir concevoir des applications spécifiques, il est nécessaire d'avoir un processeur entre ces éléments standards pour pouvoir écrire un scénario d'interaction suivant l'application.

Definition 1.1.4. Unité de traitement : Organe destiné, dans un ordinateur, à interpréter et exécuter des instructions.

Un processeur exécute un logiciel de façon séquentielle, en réalisant des opérations logiques et arithmétiques entre différents emplacements de la mémoire. Les périphériques d'entrées et de sorties étant placés en mémoire, il est possible de venir lire un capteur en entrée, réaliser un traitement et piloter un actionneur en sortie. Ce traitement peut aussi concerner l'utilisation de variables internes qui peuvent être issues de la fusion d'entrées ou dépendantes du temps d'exécution.

À partir de ces premières définitions, il est possible de faire une représentation générique d'un système embarqué comme le montre la figure 1.3.

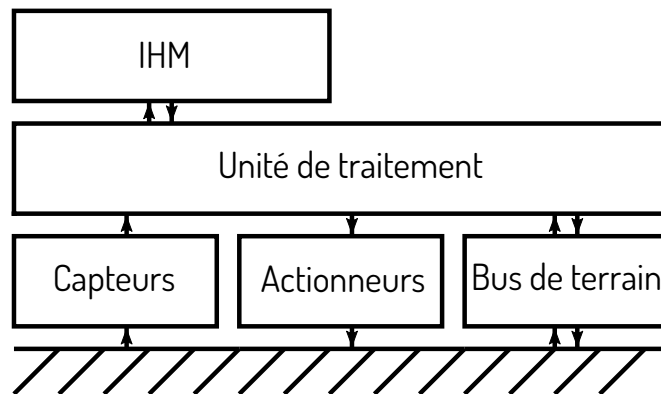


FIGURE 1.3 – Modélisation d'un système embarqué standard

Ce modèle est très générique et permet de représenter la plupart des systèmes embarqués. Les sections suivantes illustrent et donnent des exemples pour chacun des blocs.

1.1.1.2.1 Capteurs

Les capteurs sont à l'interface entre l'électronique et le monde physique, nous présentons ici les différents types de capteurs.

- Analogiques : ces capteurs sont capables de générer un signal qui est une fonction mathématique d'une grandeur physique. Ce signal va devoir être numérisé par un convertisseur analogique/numérique afin d'être traité

- Numériques (binaire ou TOR) : ces capteurs sont capables de générer un signal booléen qui indique la présence ou non d'une grandeur physique
- Intégrés : ces capteurs sont constitués d'une chaîne complète d'acquisition avec un capteur, le convertisseur analogique/numérique et une interface de communication numérique comme l'I2C ou le SPI

1.1.1.2.2 Actionneurs

Les actionneurs permettent à l'électronique d'agir sur l'environnement via différentes formes de transfert de puissance.

- Moteurs : la motorisation permet d'entraîner des mécanismes et/ou de déplacer un robot ou une machine
- Relais : la commande d'équipements de puissance est souvent réalisée via l'ouverture ou la fermeture d'un relais. L'avantage d'un relais est également de garantir l'isolation galvanique entre les équipements
- Électrovannes : la commande de fluides est souvent réalisée par des électrovannes afin de commuter l'écoulement dans les systèmes

1.1.1.2.3 Bus de terrain

Les communications entre différents systèmes ou sous-systèmes sont réalisées via des bus de communication ou bus de terrain.

- Bus sans couche physique Ethernet : ces bus sont historiques et permettent d'interconnecter des composants électroniques très simples. Ces bus sont souvent temporellement très stricts et présents dans les automates comme le bus CAN [1], Modbus[2], Profibus [3], le bus LIN [4]
- Bus avec couche physique Ethernet : avec l'évolution des composants électroniques, de plus en plus de composants intègrent une interface physique Ethernet. Ces nouveaux bus permettent une interopérabilité avec les systèmes informatique existant via une passerelle

1.1.1.2.4 Unité de traitement

L'unité de traitement est le bloc programmable qui permet de réaliser le traitement arithmétique et logique ainsi que la gestion de la mémoire des systèmes embarqués.

- Systèmes à base de microcontrôleur : ces systèmes sont parmi les plus petits systèmes numériques programmables. Ces systèmes sont assez limités en capacité de calculs, de mémoire et de périphériques. Ces systèmes sont assez complexes à mettre en oeuvre parce qu'ils n'ont généralement pas de système d'exploitation donc le logiciel est très dépendant du matériel
- Systèmes à base de microprocesseur : ces systèmes intègrent beaucoup plus de périphériques comme des interfaces Ethernet, des caméras, du stockage. Ils ont généralement des systèmes d'exploitation plus évolués comme Linux. Il est possible d'avoir des systèmes d'exploitation temps-réel (avec des garanties de temps de traitement) ou pas

- Systèmes informatique : ces systèmes sont constitués des architectures que l'on peut retrouver dans les ordinateurs ou les serveurs

1.1.1.2.5 Interface Homme-Machine

L'humain a besoin de communiquer au système des informations de commande, de configuration et d'avoir en retour des indications sur son état et la prise en compte des ordres. L'ensemble des composants qui réalisent ces fonctions s'appelle une Interface Homme-Machine.

- Périphériques d'entrée : un opérateur peut interagir avec le système via des boutons, des joysticks, un clavier, etc.
- Périphériques de sortie : la machine peut interagir avec l'opérateur via des leds, des imprimantes, des dispositifs sonores, des hauts-parleurs etc.
- Périphériques de visualisation : les écrans tactiles sont de plus en plus utilisés car ils permettent une interaction riche via des vidéos, des images, des menus

1.1.1.3 Internet des Objets

L'Internet des Objets est comme son nom l'indique l'art de connecter ces «objets» au réseau Internet mondial.

Les systèmes embarqués deviennent des systèmes embarqués communicants via l'ajout d'un module de communication. Le modèle de système embarqué précédent est ainsi modifié pour ajouter la capacité de communication.

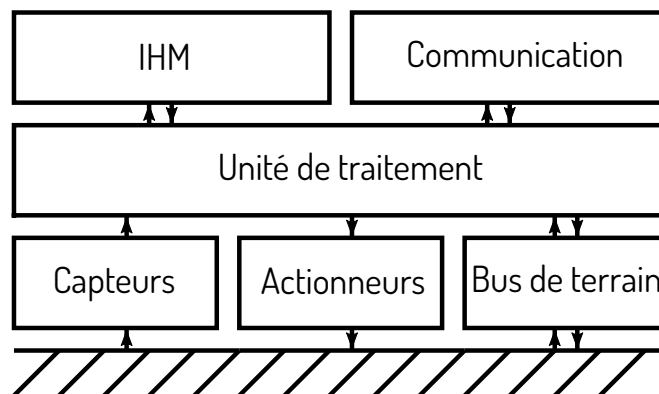


FIGURE 1.4 – Modélisation d'un système embarqué communicant

Ce module de communication va permettre l'échange de variables internes et d'évènements entre ce système et d'autres systèmes. L'objet peut aussi être commandé ou configuré à distance.

Le concept d'Internet des Objets permet de relier ces objets au réseau Internet de façon à ce qu'ils puissent communiquer avec des services hébergés sur des serveurs.

1.1.2 Origines

Nous avons parlé des origines d'Internet dans la section 1.1.1.1, il est évident que les origines de l'Internet des Objets sont très liées aux origines de l'Internet, qui lui sert de support.

1.1.2.1 Les origines académiques

Nous avons vu que les origines d'Internet sont très liées au monde académique via le réseau ARPANET et l'interconnexion des universités aux États-Unis. Les objets connectés ont été depuis longtemps très étudiés sous le nom de «réseaux de capteurs sans-fil» *Wireless Sensor Network*.

Et c'est une fois de plus la DARPA qui lance le projet «Smart Dust» [5] au milieu des années 1990. Ce projet militaire a pour but la création d'un réseau de capteurs communicants qui peuvent être dispersés, par avion par exemple, sur une zone assez éparse en vue de collecter des informations sur l'environnement. Ces capteurs doivent être petits pour être discrets et sont par nature autonomes en énergie.

Ces contraintes fortes ont amené les équipes de recherche à créer de nouveaux systèmes électroniques toujours plus sobres, de nouveaux systèmes d'exploitation temps-réel dédiés aux réseaux de capteurs, ainsi que des protocoles de communication spécifiques.

Les principaux leviers scientifiques qui ont été étudiés dans cette discipline sont les suivants :

- Économie d'énergie sur les noeuds alimentés sur batterie,
- Adaptation du réseau et de ses protocoles aux noeuds mobiles,
- Passage à l'échelle au niveau protocolaire des réseaux sans fil,
- Cross-layering au niveau des empilements protocolaires,
- Routage multi-saut,
- Auto reconfiguration du réseau.

Les deux principales technologies radio-fréquences qui sont sorties de ces travaux sont le Bluetooth [6] et le standard IEEE 802.15.4 avec différentes couches hautes dont la plus connue est ZigBee [7].

1.1.2.1.1 IEEE 802.15.4 / ZigBee

Le standard IEEE 802.15.4 est la technologie la plus proche de l'idée de départ des réseaux de capteurs sans fil.

Ce standard définit la couche physique et la couche de contrôle d'accès au médium pour des communications à faible portée, économes en énergie et avec les fonctions d'organisation du réseau.

Ce standard permet des communications synchronisées via l'émission de *beacons* réguliers ou un mode non synchronisé avec une méthode d'accès au médium de type CSMA/CA. La thèse "Proposition of a new deterministic medium access method for time-constrained wireless personal area networks" [8] d'Adrien Van Den Bossche décrit ces protocoles d'accès au médium et propose une nouvelle méthode déterministe.

Les noeuds du réseau peuvent avoir plusieurs niveaux d'implication dans la gestion du réseau :

- Coordinateur du réseau : c'est le coordinateur du réseau PAN qui définit la période de synchronisation

- *Full Function Device* : un nœud de type FFD se doit d'implémenter les fonctionnalités de base, ainsi que les fonctions d'agent actif du réseau, en participant au routage des paquets.
- *Reduced Function Device* : un nœud de type RFD est le nœud le plus simple possible qui implémente uniquement les fonctionnalités de base et qui ne participe pas de façon active à l'organisation du réseau

Le standard IEEE 802.15.4 n'est pas suffisant pour réaliser une application de réseau de capteurs, il a besoin d'une couche réseau qui organise principalement le routage des paquets dans le réseau.

Plusieurs protocoles sans-fil se basant sur les couches IEEE 802.15.4 ont vu le jour comme ZigBee ou 6LoWPAN [9].

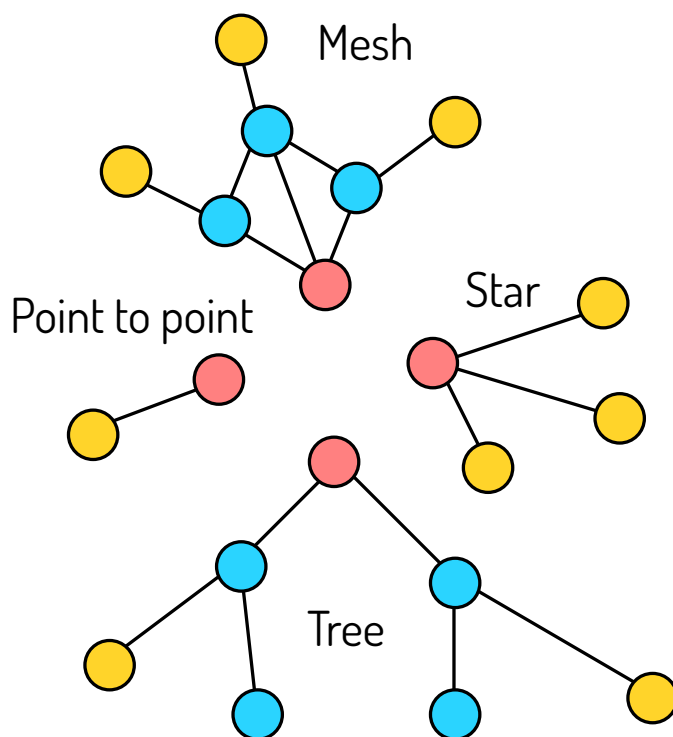


FIGURE 1.5 – Différentes topologies d'un réseau ZigBee

L'union d'une couche basse IEEE 802.15.4 avec une couche haute de type ZigBee ou 6LoWPAN permet de créer des réseaux de capteurs avec des topologies étendues comme sur la figure 1.5, de type réseau *mesh* ou réseau sous forme d'arbre.

Ces technologies permettent bien de répondre à la problématique des réseaux de capteurs sans fil et autonomes en énergie, mais quid de la connexion entre ces réseaux et Internet ?

ZigBee qui est la technologie historique n'intègre pas de solution standard pour les échanges de trames entre le monde IP et le réseau ZigBee. 6LoWPAN est basé sur l'idée de relier ces deux mondes par conception. Cette vision est une des voies qui a bâti la notion d'Internet des Objets, sur l'idée que les services web et les objets doivent être capables de dialoguer directement.

Il existe aujourd'hui une gamme de produit ZigBee relativement large avec des ampoules connectées,

des détecteurs et capteurs pour la domotique et quelques projets industriels.

1.1.2.1.2 Bluetooth

La norme Bluetooth a été créée par Ericson en 1994 en plein essor des téléphones mobiles. L'objectif d'Ericson était de favoriser la communication à courte distance entre deux téléphones mobiles ou entre un téléphone mobile et des accessoires.

Voici les principaux cas d'usages de la technologie Bluetooth :

- Communications audio entre un téléphone portable et une oreillette, un autoradio, une enceinte,
- Partage de données comme un carnet d'adresse entre plusieurs téléphones ou un téléphone et un ordinateur,
- Communication sans fil entre un ordinateur et un clavier, une souris ou un casque audio.

Bluetooth a longtemps été cantonné à ces applications d'accessoires autour du téléphone mobile et des ordinateurs. L'idée est de déployer des accessoires à proximité du maître via une liaison sans-fil pour favoriser la mobilité et les interactions dans un rayon proche de l'utilisateur. Ces types de réseaux sont appelés piconet ou scatternet dans la version à plusieurs sauts.

Au fil des versions du protocole Bluetooth et principalement à partir de la version 4.0 et Bluetooth Low Energy (BLE) en 2010, les cas d'usages ont été étendus à destination des objets connectés qui doivent être économes en énergie. Cette volonté est observable de plus par l'augmentation de la portée possible ainsi que la réduction de l'impact énergétique des protocoles de sécurité [10].

Ces cas d'usages plus récents peuvent être ajoutés aux anciens :

- Éclairage connecté,
- Matériel médical et sportif : thermomètre, oxymètre, glucomètre, podomètre,
- Capteurs d'environnement,
- Détection de proximité via des balises.

1.1.2.2 L'essor des smartphones

En 2006, la sortie de la technologie 3G permet d'avoir un débit assez élevé pour naviguer sur Internet. Les opérateurs commencent à proposer des forfaits avec une limite en volume de données par mois qui permet de pouvoir laisser son téléphone constamment connecté à Internet.

Le marché va totalement exploser à partir de 2007 avec la sortie du premier iPhone. Le téléphone mobile se transforme en smartphone par l'ajout d'un grand écran tactile multipoint, la présence d'un système d'exploitation qui s'approche des fonctionnalités que l'on peut avoir sur un ordinateur et le développement du concept d'applications.

En 2005 la guerre des systèmes d'exploitation fait rage pour faire face à Apple qui était en train de développer iOS. Microsoft s'allie avec HTC pour le développement de Windows Phone et Google rachète la startup Android.

Chaque système d'exploitation crée son SDK afin que chacun puisse développer une application mobile et la distribuer via une application particulière qui sert d'annuaire et de gestionnaire d'installation. Afin

de ne pas avoir à créer une application pour chaque marque de téléphone, Android a gagné des parts de marché en «offrant» le système d'exploitation aux fabricants de smartphones en échange de la présence des applications de Google par défaut.

La démocratisation entre 2006 et 2010 des smartphones et de leurs applications a permis de mettre dans la poche du très grand nombre, une passerelle entre l'Internet et le monde physique. C'est dans un premier temps l'humain qui est devenu connecté, en interagissant avec son environnement au travers d'applications et des capteurs embarqués dans les smartphones. Il a été possible d'envoyer via le réseau 2G/3G/4G ou par WiFi ces informations directement à des serveurs distants. C'est l'essor des applications de navigation, des réseaux sociaux, du partage de photos et de vidéos en direct etc.

1.1.2.3 Le smartphone comme première passerelle universelle

Avant l'arrivée des smartphones, les téléphones étaient de plus en plus compacts, ce qui rendait la tâche difficile pour l'intégration d'antennes et de transmetteurs radio. La tendance a été de limiter la connectivité selon deux axes : les technologies mobiles et WiFi pour la liaison de données et Bluetooth pour les accessoires.

Les smartphones qui nécessitent des grands écrans tactiles ont permis d'inverser la tendance et d'avoir plus de volume pour l'intégration de capteurs et de technologies sans fil.

Malheureusement, très peu de technologies ont été intégrées en complément des existantes sauf l'Ultra Wide Band qui a été intégré récemment dans l'iPhone 11 [11]. L'interaction avec les objets connectés se fait principalement via le Bluetooth ou le WiFi. Le Bluetooth n'étant pas pensé pour des réseaux de capteurs étendus, il est limité aux interactions proches de l'utilisateur et donc plutôt pour de la collecte de capteurs ambiants ou d'accessoires. Le WiFi étant plutôt consommateur d'énergie, il ne peut être qu'implémenté par des noeuds non économes en énergie.

Le problème de concevoir un smartphone comme une passerelle pour un réseau sans fil est limitant du fait que le smartphone est toujours porté par l'utilisateur et que l'utilisateur n'est pas toujours à portée du réseau de collecte et parfois même du réseau opérateur.

L'essor des smartphones a bousculé la création d'interfaces hommes-machine présentes sous forme d'écrans LCD, de boutons et de leds en les remplaçant par des applications sur smartphone. Le smartphone est utilisé dans ce cadre là comme une interface homme-machine connectée où l'on peut visualiser les informations du système et interagir avec lui lorsque l'on est proche de lui. La nouveauté est qu'il est possible également de transmettre ces données à un serveur distant, de récupérer des données d'autres systèmes et de cette façon unir, dans une première approche le monde physique et le monde du numérique. Il est important de souligner que cette interface permet aussi de gérer la mise à jour des objets, ce qui est un enjeu primordial autant pour la sécurité que pour la gestion d'un parc d'objets comme nous aurons l'occasion de le voir plus loin.

1.2 L'Internet des Objets pour l'industrie (IIoT)

1.2.1 Vers de nouvelles formes d'organisation des entreprises

Le développement des technologies de l'information et de la communication crée des modifications profondes dans nos sociétés.

Dans le passé, l'apparition de l'écriture, de l'imprimerie puis de la presse écrite a permis de partager toujours plus d'informations via des supports variés.

Ces technologies permettent au plus grand nombre d'avoir accès aux connaissances et de participer à créer, critiquer, modifier et partager ce savoir.

Internet est la dernière innovation majeure dans ce domaine. Il est intéressant de faire le parallèle entre l'encyclopédie de Diderot qui a été rendue possible grâce à l'invention de l'imprimerie et wikipedia, son pendant moderne, qui a été rendu possible grâce à Internet.

Les entreprises ont compris que l'apparition de ces nouvelles technologies allait modifier de façon significative leur façon de collaborer, de vendre, d'échanger, de gérer ... et même d'enseigner depuis la crise du Covid-19! La donnée devient pour les entreprises actuelles une nouvelle source de valeur ajoutée et une nouvelle matière première. Toutes les entreprises produisent des données qui peuvent servir à mieux comprendre les performances de l'entreprise, les besoins de ses clients, l'évolution du marché etc.

L'enjeu des entreprises est de prendre conscience de l'existence de ces données, du moyen de les collecter, de les protéger et d'être capable d'estimer leur valeur marchande.

Un nouveau modèle d'organisation des entreprises est en train d'émerger : le modèle d'*holocratie* [12]. La figure 1.6 présente une adaptation de ce modèle à la gestion des données de l'entreprise. Chaque service de l'entreprise rend compte de ses actions non plus par un reporting écrit ou oral mais par des données en continue. Les services rendent compte des variables de l'environnement externe à l'entreprise permettant ainsi de numériser de façon globale l'action de l'entreprise et du contexte de cette exécution.

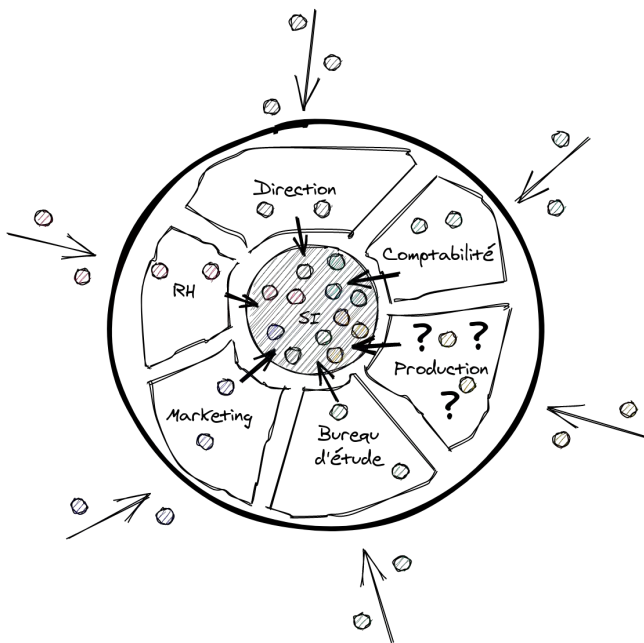


FIGURE 1.6 – Nouvelle forme d'organisation des entreprises autour de la donnée

L'Internet des Objets pour l'industrie s'intéresse plus particulièrement à la collecte des données et de l'environnement de production, représenté avec des points d'interrogation sur la figure 1.6. Quelles données sont pertinentes? Comment les collecter, les transmettre et les stocker en flux tendu?

L'idée de cette démarche est de pouvoir par la suite analyser ces données centralisées et normalisées à l'aide d'algorithmes afin d'en extraire des tendances, des indicateurs de performance et des décisions

éclairées. Cette vision est développée autour d'une stratégie de valorisation des données au sein de l'entreprise.

1.2.2 Stratégie de valorisation des données

Dans un monde toujours de plus en plus complexe, il devient obligatoire pour prendre des décisions éclairées, de s'aider d'algorithmes statistiques qui explorent les volumes de données métier que possède l'entreprise.

Imaginons un assistant personnel virtuel pour les entreprises à qui on puisse poser ce genre de questions :

- Où dois-je ouvrir une boutique pour minimiser le temps de trajet de mes clients ?
- Quel est l'impact sur les finances de l'entreprise si on décide de fermer tel site ?
- Quelle est la machine qui a le plus de chance de tomber en panne dans une semaine ?
- Faut-il mieux signer le contrat A ou le contrat B ?

Afin de poursuivre cet objectif, les entreprises investissent de plus en plus dans une stratégie en trois étapes comme décrites sur la figure 1.7.

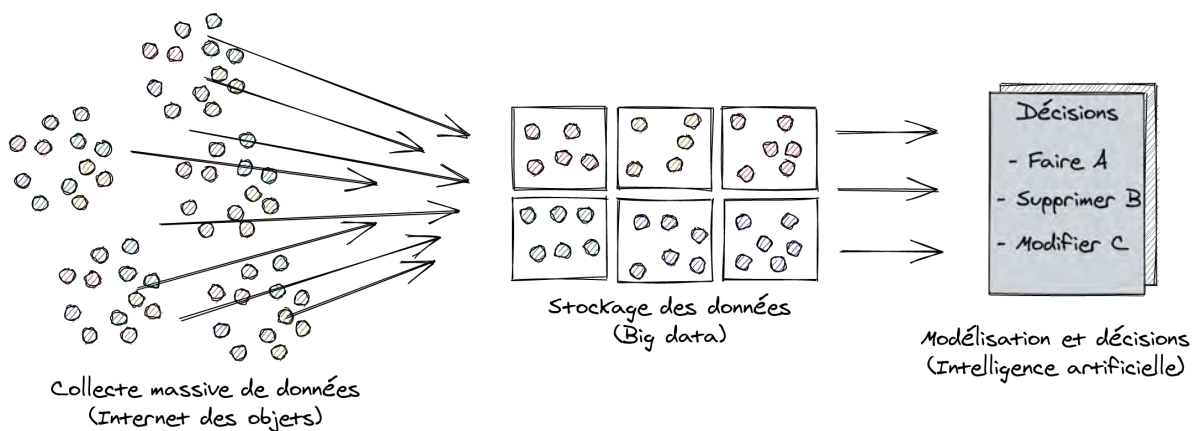


FIGURE 1.7 – De la collecte des informations à l'agrégation et la modélisation

Ces trois grandes étapes qui sont des domaines de recherche à part entière, représentent le cheminement complet de la collecte d'informations sur le terrain à leur interprétation sous forme logique. La figure 1.7 montre par des flèches les flux d'informations qui transitent entre les différentes fonctions. Les objets connectés sont généralement très nombreux et dispersés sur le territoire afin de collecter des données réparties dans le périmètre d'action de l'entreprise. Ces flux de données sont collectés et rassemblés afin d'être traités de façon unifiée. Il devient nécessaire de sauvegarder ces données de façon ordonnée afin de faciliter par la suite le processus d'extraction. Cette extraction permet par la suite de modéliser et de transformer ces données en informations logiques dans l'objectif de les combiner.

1.2.2.1 Collecte des informations

La collecte des informations dans l'environnement consiste à identifier les sources de données et à mettre en œuvre des techniques d'extraction et de transmission de ces données à travers différents réseaux.

Les données peuvent être collectées via un reporting numérique au travers d'applications, de formulaires ou de collecte de photos et de vidéos des équipes de terrain. Cette collecte manuelle est perçue comme une perte de temps et d'énergie alors que l'objectif initial doit rester l'optimisation et l'efficacité du travail.

L'Internet des Objets est perçu comme un moyen d'obtenir une partie de ces informations de façon automatique, en flux tendu de manière à alléger cette tâche fastidieuse et source d'erreurs. Pour cela, les équipements, les machines, les entrepôts, les usines vont être équipés de systèmes embarqués capables d'acquérir et de transmettre ces informations précieuses.

1.2.2.2 Agrégation des informations

Un fois la collecte réalisée, il est nécessaire de stocker ces informations et de les trier dans des immenses entrepôts de données.

La collecte massive d'informations entraîne une redondance de l'information. Il faut procéder à la suppression des doublons et à l'extraction de l'information utile. Par exemple, il est inutile de stocker une photo si l'information utile est par exemple, le nombre d'objets présents sur la photo.

Cette discipline est connue sous le nom de *Big Data* où l'art de stocker et d'agréger d'énormes volumes de données. [13]

1.2.2.3 Modélisation des informations

La dernière étape consiste à créer des modèles statistiques de ces données qui vont pouvoir être utilisés par des algorithmes afin de catégoriser et d'analyser ces données.

Le volume de donnée étant trop conséquent, le recours à des algorithmes d'apprentissage automatique est de plus en plus utilisé pour éviter une modélisation complexe.

Des réseaux de neurones artificiels permettent après une phase d'apprentissage sur des jeux de données de test, d'être en mesure de fournir des fonctions de tris, de décision, de traitement.

Cette discipline d'apprentissage automatisé par la machine est reconnue sous le nom de Machine Learning. [14]

1.2.3 La problématique de l'interopérabilité

Dans une volonté de développement de cette stratégie, les entreprises investissent dans différents projets d'optimisation de leurs métiers via la collecte de données. Les difficultés structurelles et humaines des entreprises et plus particulièrement des grands groupes, rendent difficile la mise en place d'une politique globale. Il est aisé de comprendre qu'une politique globale est nécessaire afin de mutualiser des services, des objets, des moyens humains entre les différentes applications de manière à ne pas multiplier les coûts.

Ces difficultés entraînent le développement d'applications très disjointes et concurrentes. Imaginez que, lors du développement d'une nouvelle application qui nécessite les mêmes données que l'application précédente, de nouveaux objets soient déployés à quelques centimètres des autres par échec d'entente entre les différents projets. Cette situation peut faire sourire mais est malheureusement réaliste vis-à-vis de la complexité des problèmes humains et structurels.

Les architectures résultantes de ce manque de collaboration sont décrites comme des silos applicatifs. Une architecture en silos applicatifs est un ensemble de composants logiciels et matériels qui sont mis côte à côte afin de répondre à un besoin spécifique sans se soucier de l'interopérabilité. La figure 1.8 représente ces silos applicatifs dans le cadre d'applications pour l'Internet des Objets. Cette figure montre un enchaînement de logiciels et de matériels du terrain en bas jusqu'aux applications serveurs en haut. Les données transitent à travers cet enchaînement via différents protocoles et technologies réseaux. Le concept de *silo applicatif* consiste à critiquer la mise en parallèle de ce genre d'architecture sans chercher à les unifier d'une façon ou d'une autre. Les architectures en silo proviennent de problèmes politiques, techniques ou organisationnels qui motivent cette volonté de cloisonner les choses.

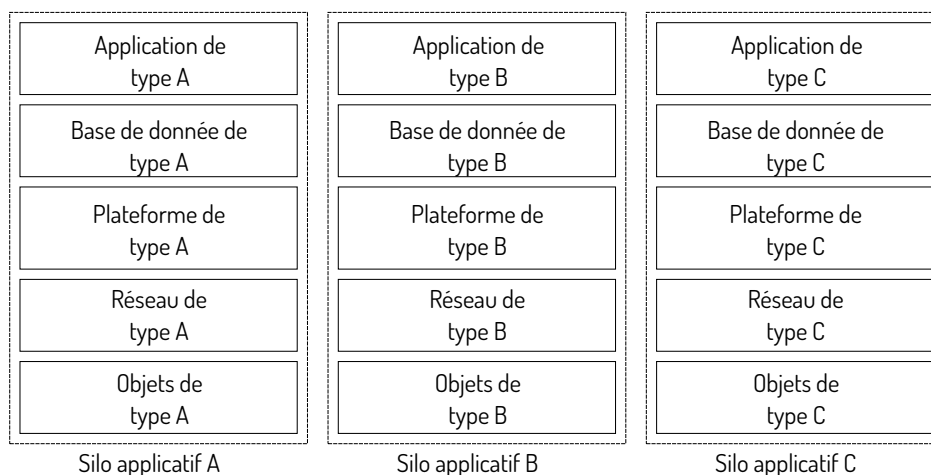


FIGURE 1.8 – Architecture en silos applicatifs

D'après ce constat, nous définissons le sens du mot *interopérabilité* dans le cadre de ces travaux :

Definition 1.2.1. Interopérabilité : Capacité intrinsèque que possède un système à pouvoir fonctionner avec d'autres systèmes via la définition de ses interfaces. Dans le cadre des télécommunications, c'est plus précisément la capacité des systèmes à pouvoir communiquer avec d'autres systèmes existants ou futurs par la définition d'interfaces de communication.

Par l'expérience acquise durant cette thèse, nous avons constaté que le fonctionnement des entreprises et particulièrement des grandes entreprises rend souvent difficile le fait d'avoir une politique globale sur des projets différents. Les projets industriels dans l'Internet des Objets auxquels nous avons été confrontés, ont été complexes à cause de ces mêmes raisons.

La principale raison est que les projets sont financés indépendamment et qu'ils sont en concurrence les uns avec les autres. Cette concurrence n'aide pas les différents projets à trouver un terrain d'entente sur des composants mutualisables ou sur des standards communs. Elle n'aide pas non plus à s'entendre sur un financement commun de ces ressources.

La deuxième raison est que pour avoir une politique globale, il faut une entité spécialisée sur les problématiques d'Internet des Objets et ayant comme mission de chercher à mutualiser les ressources, les protocoles, les plateformes, etc.

Ce type d'architecture entraîne bien souvent des inter-dépendances entre les maillons de la chaîne d'information. Ceci veut dire que le flux d'information transite de composant en composant et que chaque maillon est effectivement dépendant de ses voisins.

Ces architectures sont difficiles à maintenir car généralement une erreur est constatée dans l'interface de visualisation et il est difficile de trouver le maillon fautif dans la chaîne. Il est nécessaire d'avoir un accès à l'interface entre chaque maillon afin de diagnostiquer dans quel maillon se situe l'erreur ou le début d'erreur.

Ces architectures ne permettent pas de mutualiser les ressources, par exemple si une standardisation est faite au niveau de la base de donnée, il est possible que les applications puissent facilement être évolutives sans gros efforts lorsque de nouveaux objets vont être déployés.

Ce type d'architecture peut entraîner des coûts élevés s'il faut déployer plusieurs infrastructures de réseaux ou venir dupliquer des objets connectés pour en avoir un par application.

Afin de chercher une solution technique à cette problématique, il faut chercher des pistes d'interopérabilité des composants par design en proposant de nouvelles architectures favorisant cette interopérabilité. Comme nous l'avons décrit, ces problématiques ne sont pas uniquement d'ordre technique et nécessitent des changements profonds dans les organisations qui sortent du cadre de cette thèse.

1.3 État de l'art scientifique sur l'interopérabilité

Nous présentons dans cette section une liste de travaux qui ont été proposés au sujet de l'interopérabilité dans le contexte de l'Internet des Objets. Cette thématique est très importante dans le cadre de déploiements réels car l'idée derrière l'interopérabilité est de pérenniser dans le temps une application et d'éviter les solutions qui ne peuvent pas dialoguer ensemble. De cette façon, une couche d'interopérabilité peut être implémentée dans l'empilement de telle sorte que les objets puissent être remplaçables facilement et avec peu d'impact sur les logiciels applicatifs.

Cette couche d'interopérabilité peut être implémentée de différentes manières, nous avons observé dans la littérature ci-après trois principales manières, par codage, par design et par un protocole de transport.

1.3.1 Sémantique et encodage des données

Une première piste d'interopérabilité qui a été étudiée est la définition d'un encodage standard des données. L'idée est que les réseaux de systèmes embarqués communicants transportent la plupart du temps des informations issues de capteurs, d'actionneurs ou d'IHM et que c'est la manière de représenter ces informations qui peut être unifiée afin de créer de l'interopérabilité à la source.

OneM2M est une initiative de standardisation des architectures IoT. Dans le cadre de cette initiative, une démarche de définition d'une ontologie a été effectuée sous le nom de IoT-O. L'article "Toward semantic interoperability in oneM2M architecture" [15] publié en 2015 par "IEEE Communications Magazine" décrit cette ontologie ainsi que des cas d'usage.

Une seconde initiative issue de l'IETF cette fois, propose SenML [16], une sémantique pour l'encodage de données via CoAP ou HTTP. Cette sémantique peut être également utilisée avec d'autres protocoles de transport que ceux standardisés par l'IETF comme MQTT, dans la plateforme IoT open-source Mainflux[17]. Un ensemble de symboles sont définis pour représenter les unités ou les labels à donner aux variables.

Ce champ de recherche est très spécifique et relève plus de la modélisation que de la recherche en réseau, ainsi nous avons écarté cette piste dans le cadre de notre étude.

1.3.2 Architectures interopérables pour l’IoT

Le survey ”Middleware for Internet of Things : A Survey” [18] est particulièrement intéressant et original car il traite de l’interopérabilité par design en présentant différents modèles. Ce survey a été publié en novembre 2015 dans l’”IEEE Internet of Things Journal”. Il montre différentes architectures permettant d’offrir une interopérabilité :

- une gestion par évènement, avec la transmission d’évènements aux applications concernées,
- une gestion par service, lorsque les données traversent différents services logiciels qui traitent des fonctions spécifiques,
- une architecture par machine virtuelle (VM), avec des fonctions très séparées qui sont unifiées au plus proche des applications par une VM d’interopérabilité,
- une architecture multi-agents, avec une grande importance donnée aux objets qui implémentent des fonctions d’interopérabilité de très bas niveau,
- une gestion par base de données, lorsque les données sont toutes stockées dans des bases de données ou des tables séparées par fonction et où les applications viennent gérer l’interopérabilité en récupérant les données dans toutes les bases de données.

Cet article ne traite pas des architectures à base de conteneurs que nous allons étudier dans cette thèse.

1.3.3 Protocoles de transport des données

La problématique de l’interopérabilité peut être traitée au niveau réseau en normalisant un protocole de transport. En effet, l’Internet des Objets intègre forcément la notion de réseau où des messages transitent d’objets en objets puis à travers des passerelles rejoignent des serveurs en passant par le réseau Internet. De par cette nature de transport de données, la couche de transport est idéale pour prendre en charge cette convergence.

De nombreux protocoles existent dans la littérature et sont déjà mis en œuvre dans des applications IoT pour le transport des données. Le tableau 1.1 liste les protocoles les plus répandus actuellement.

De nombreux papiers ont apporté des éléments de comparaison entre ces protocoles comme ”Choice of effective messaging protocols for IoT systems : MQTT, CoAP, AMQP and HTTP” [19] publié en 2017 dans ”IEEE International Systems Engineering Symposium (ISSE)” duquel est extrait en partie le tableau 1.1.

Protocole	MQTT	AMQP	CoAP	HTTP
Année	1999	2003	2010	2003
Paradigme	Publish/Subscribe	Producer/Consumer	Request/Response	Request/Response
Transport	TCP	TCP	UDP	TCP
Sécurité	TLS	TLS	DTLS	TLS
Ports	1883/8883	5671/5672	5683/5684	80/443
Format	Binaire, texte	Binaire, texte	Binaire, texte	Binaire, texte
Licence	Open Source	Open Source	Open Source	Libre
Contraintes	Faible	Forte	Très faible	Moyenne

TABLE 1.1 – Principaux protocoles de transport pour l’IoT

Au niveau de la standardisation, HTTP [20] et CoAP [21] sont normalisés par l'IETF alors que MQTT [22] et AMQP [23] sont normalisés par l'OASIS un organisme de standardisation open source.

1.3.4 Encapsulation des messages

L'Internet des Objets consiste à faire le lien entre des réseaux de collecte, souvent sans fil et les technologies de l'Internet. La fonction d'Internet est d'identifier par une adresse unique au niveau global un agent et de permettre le transport des messages d'un agent à un autre. Cette fonction est principalement assurée par le protocole IP pour la couche réseau et les protocoles TCP et UDP pour la couche transport.

Deux stratégies s'affrontent pour connecter les objets à Internet. La première qui est portée par l'IETF, l'organisme qui standardise les technologies du web et d'Internet, cherche à donner une adresse IPv6 à chaque objet afin de garder les paradigmes existants. Pour les objets qui ne disposent pas d'interface TCP/IP, c'est typiquement une passerelle qui va faire la translation entre l'adresse locale sur le réseau de collecte et l'adresse globale sur Internet. C'est par exemple l'idée derrière la RFC 8724 [24], qui standardise le protocole SCHC (Static Context Header Compression and Fragmentation) qui vise à appliquer cette technique aux réseaux LPWAN.

La deuxième idée consiste à utiliser le concept de tunnel pour faire transiter des trames ou des paquets du réseau de collecte à destination de serveurs à travers un tunnel de couche 3 ou 4. Nous connaissons les notions de trames ou de paquets qui sont souvent utilisées pour représenter les blocs de données de la couche 2 ou 3 du modèle OSI. Nous situons le concept de message plutôt au niveau de la couche de transport.

Il est nécessaire lorsque l'on encapsule cette trame de joindre un ensemble de méta-données qui permettent de spécifier dans quel contexte a été reçue cette trame, et de quelle manière elle a transité jusqu'aux serveurs. Ces méta-données sont nombreuses et ont besoin d'être organisées, le modèle protocolaire typique semble difficile à mettre en œuvre et notre choix s'oriente sur des messages avec un format plus riche tel que le JSON qui permet d'organiser proprement les informations.

Voici un exemple de message qui peut être envoyé dans ce genre d'architecture :

```
1 {  
2   "payload": "052985f2f4552a2b4b5b6",  
3   "metadata": {  
4     "frequency": 868.0,  
5     "power": 20.0  
6   }  
7 }
```

FIGURE 1.9 – Exemple d'un format de données JSON

Le serveur qui va recevoir ce message va être capable de décoder la trame et d'avoir une liste d'informations sur les conditions dans lesquelles elle a été reçue.

Un standard est en train de se développer afin de compresser ce type de message lors du transport entre plusieurs logiciels, c'est le protocole de sérialisation et de désérialisation « Protocol Buffers » [25]. Protocol Buffers (protobuf) permet de définir une spécification d'interface et de générer du code pour sérialiser et désérialiser des données dans de nombreux langages.

1.3.5 Architecture IoT event-driven

Les architectures IoT sont le plus souvent modélisables sous forme de graphe de flux, c'est-à-dire un enchaînement de composants logiciels ou matériels qui sont reliés par des flux de messages.

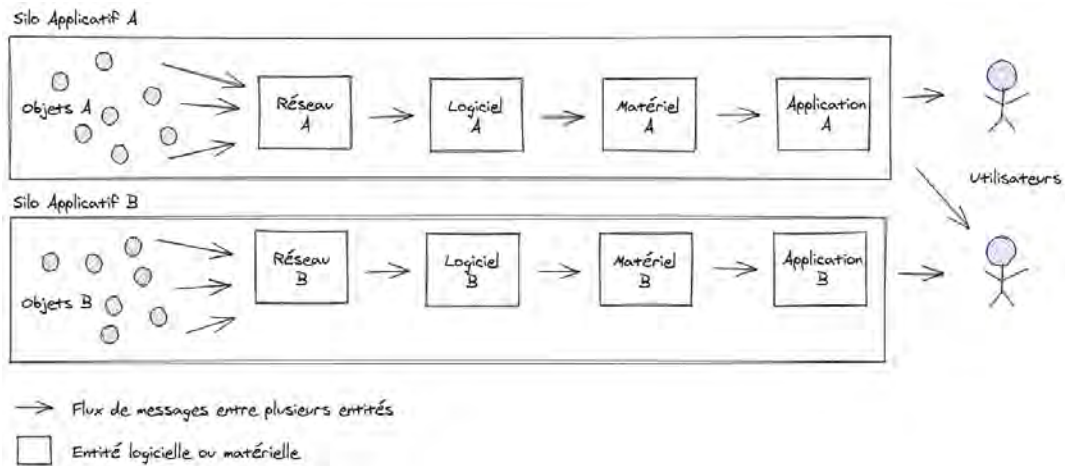


FIGURE 1.10 – Graphe de flux des silos applicatifs

La figure 1.10, présente un exemple d'architecture où des nœuds à gauche publient des messages à destination d'applications à droite par l'intermédiaire de différents composants. Ces composants peuvent fournir des services réseaux, des services de stockage, des services de modification des flux, des services d'agrégation des flux, etc. Deux silos applicatifs sont définis, ils sont totalement indépendants car aucun message ne peut transiter de l'un à l'autre. Nous pouvons affirmer qu'ils ne sont pas interopérables. Les échanges entre les composants sont réalisés par différents protocoles réseaux avec différents paradigmes.

Chercher une piste d'interopérabilité consiste à trouver un composant commun à beaucoup d'applications qui pourrait servir de point central dans l'architecture et permettre l'échange de messages entre différentes applications.

Cette stratégie est complexe parce que le composant choisi doit être commun et doit implémenter des mécanismes de notification de son activité auprès des connexions actives.

Dans la littérature scientifique, cette approche de logiciel intermédiaire se situant entre les objets et les utilisateurs est nommée *IoT Middleware*. L'article "Middleware for Internet of Things : A Survey" [18] propose un survey des différentes fonctions que peut proposer un logiciel intermédiaire pour l'Internet des Objets ainsi que la liste des implémentations existantes. L'approche *middleware* a plutôt tendance à enfermer les architectures logicielles dans une boîte noire qui se veut complète alors que l'approche réseau a intrinsèquement tendance à ouvrir vers l'extérieur. Il est complexe de pouvoir externaliser une petite fonctionnalité manquante que l'on souhaiterait rajouter au milieu de la chaîne de traitement. L'approche logiciel libre est une solution à privilégier dans ce cas, afin de laisser l'opportunité aux utilisateurs d'ajouter ou de modifier le logiciel en question selon leurs besoins.

L'approche *middleware* peut être également le choix d'un logiciel simple permettant d'unir les silos applicatifs. La figure 1.11 montre une architecture centrée autour d'une base de données qui a été sélectionnée pour servir de composant d'interopérabilité. Pour cela, elle doit implémenter un mécanisme afin de signaler à tous les clients que l'un d'eux vient de faire une opération. De cette façon on peut imaginer avoir des producteurs et des consommateurs des données qui sont stockées.

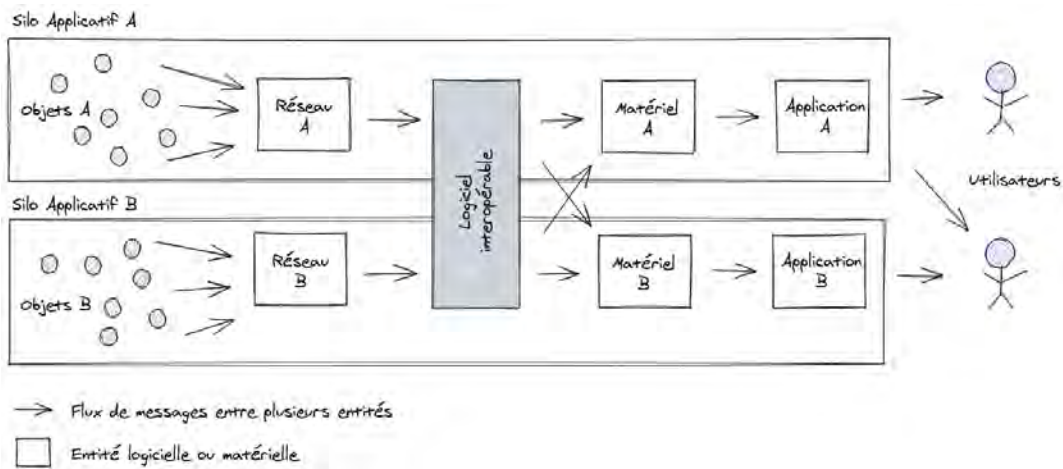


FIGURE 1.11 – Notion de composant interopérable

La stratégie que nous avons étudiée consiste à chercher non pas un agent commun mais à normaliser les échanges entre les agents. L'objectif est que les communications entre les agents soient unifiées et centralisées et qu'aucun agent ne soit plus important que les autres.

Nous traitons donc l'enjeu de l'interopérabilité au niveau de la couche transport de l'empilement protocolaire OSI en cherchant un protocole commun et un paradigme adapté à l'Internet des Objets.

Cette stratégie est représentée sur la figure 1.12, les agents sont centralisés autour de leur fonction réseau car ils ont su s'entendre sur une façon d'échanger des messages de manière standard. Cette uniformisation de la fonction réseau permet de casser les interactions directes au niveau réseau entre les différents composants. Il n'y a plus de notion de connexion ou de client et de serveurs entre les composants car ils sont tous des clients du service de messagerie. De cette façon, si un client devient hors d'usage, le service associé ne serait plus disponible mais cela n'aurait aucun impact sur les mécanismes réseaux des autres composants. De la même façon, l'ajout de nouveaux composants est plus facile car il ne nécessite pas de modifier la configuration des autres.

Un réseau en graphe de flux dans des silos applicatifs est bien adapté au paradigme client/serveur car chaque composant est serveur pour le composant précédent et client pour le composant suivant.

Ce modèle de graphe de flux est utilisé dans différents logiciels de réseau comme GNURadio [26] ou comme Node-RED [27]. Nous étudierons l'implémentation de l'interopérabilité dans le logiciel Node-RED qui est une application web afin de créer des scénarios IoT.

1.4 Les concepts et composants de base de l'IoT

Dans un premier temps, afin de comprendre les enjeux de l'Internet des Objets Industriels, il est nécessaire de modéliser les différents concepts et composants spécifiques de cette discipline. Ces modèles vont nous permettre de représenter les architectures et de comprendre les interactions entre les différents éléments constitutifs.

L'architecture protocolaire principalement admise de l'Internet des Objets est une architecture dite en trois couches. Des agents physiques sur le terrain relèvent des informations et les transmettent à des

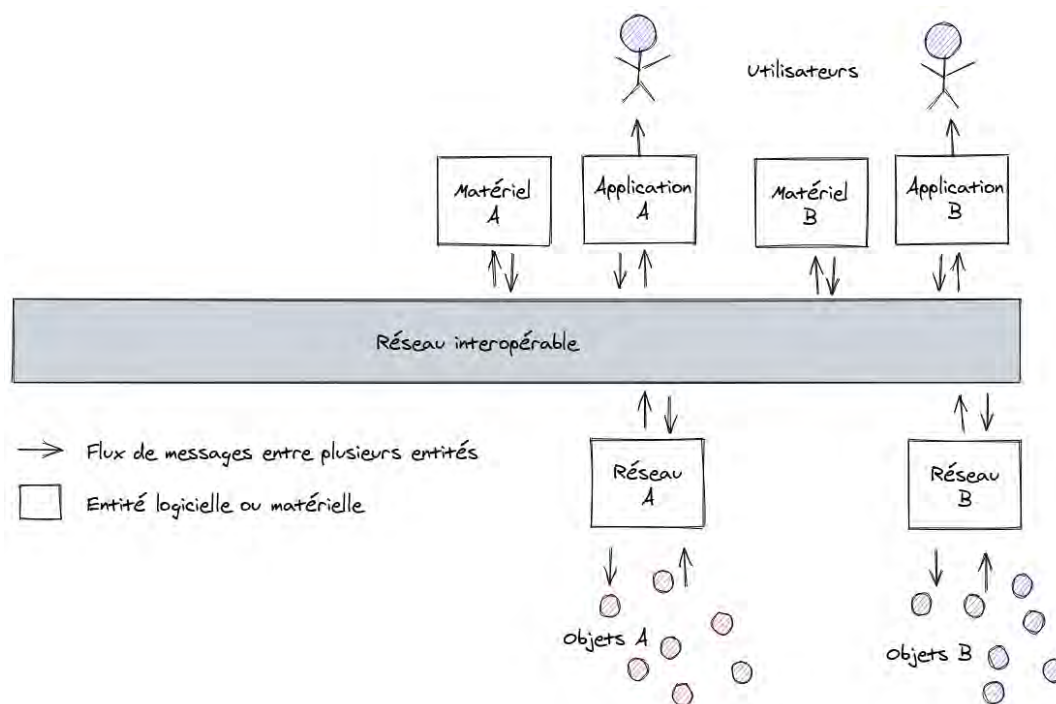


FIGURE 1.12 – Notion de réseau interopérable

passerelles via un réseau de collecte. Ces passerelles servent de relayeurs de messages à destination des serveurs répartis dans le cloud.

1.4.1 La couche d'acquisition des informations

1.4.1.1 Postulat

L'acquisition des informations consiste à capter des données physiques issues de l'environnement, de les numériser et de les transmettre.

L'acquisition d'informations est réalisée par des agents présents sur le terrain et en contact direct avec l'environnement.

Leur rôle principal est de faire l'interface entre le monde physique et le ou les réseaux de collecte.

Les agents d'acquisition sont très spécifiques car dédiés et dimensionnés pour un cas d'usage précis.

1.4.1.2 Modèle

1.4.1.2.1 Modèle complet d'objet connecté multi interfaces

L'étage d'acquisition est celui qui est le plus proche du terrain, c'est dans cet étage que l'on va retrouver les objets connectés comme nous l'avons définis plus haut.

Nous pouvons reprendre le modèle d'objet connecté présenté en figure 1.3 en lui ajoutant plusieurs interfaces de communication (figure 1.13).

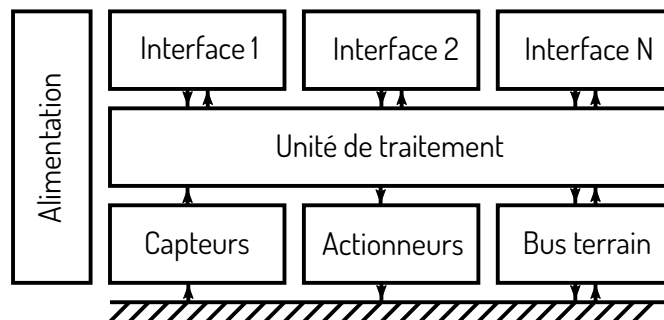


FIGURE 1.13 – Modèle d'objet connecté à plusieurs interfaces de communication

Ce modèle prend en compte la présence de plusieurs interfaces de communication. Ce cas d'usage est de plus en plus fréquent car les interfaces de communication et essentiellement les réseaux sans fil, sont de plus en plus nombreux et spécialisés sur des usages bien particuliers.

Un objet qui doit répondre à plusieurs cas d'usages ou des fusions de cas d'usages est amené à intégrer plusieurs technologies de communication.

L'autre élément du modèle qui fait son apparition est la notion d'alimentation de l'objet connecté. Cette alimentation est réalisée soit par le secteur soit par une batterie embarquée dans l'objet.

1.4.1.2.2 Modèle de capteur connecté

Un capteur connecté est un sous-ensemble du modèle d'objet connecté dans lequel il y a juste : un ou plusieurs capteurs, une unité de traitement et une interface de communication.

Les capteurs sont capables de générer, soit périodiquement, soit sur un évènement, un ensemble de valeurs numériques issues de l'environnement.

Dans le cas d'une acquisition périodique, la fréquence d'acquisition est généralement très grande devant la capacité de l'interface réseau. De plus, les valeurs brutes issues d'un capteur ont une entropie assez faible. Un étage de traitement va permettre de filtrer, de fusionner ou de faire des opérations mathématiques sur les valeurs brutes afin de réduire le volume d'informations à transmettre et d'augmenter l'entropie des informations.

Un étage de seuillage optionnel peut permettre de générer des évènements en fonction des valeurs brutes.

La transmission de ces informations via une interface peut se faire soit à l'initiative du capteur, soit par mise à disposition des données et c'est alors les autres agents qui viennent les collecter.

Nous avons distingué deux modes de transmission, le mode périodique et le mode sporadique.

Dans le mode sporadique, la transmission d'un message n'est pas prévisible et va être réalisée sur évènement. Dans le mode périodique, un message va être envoyé à une période déterminée afin de transmettre les valeurs.

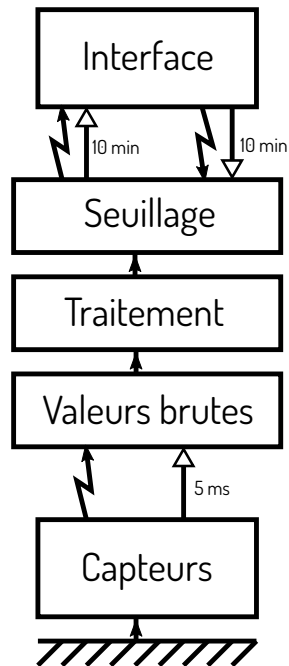


FIGURE 1.14 – Modèle des mécanismes réseaux d'un capteur connecté

1.4.1.2.3 Modèle d'actionneur connecté

Les actionneurs sont souvent des systèmes asservis qui ont des capteurs afin de réaliser la contre réaction. La plupart des systèmes vont donc être une hybridation des deux modèles.

Un message à destination d'un actionneur est un changement de consigne. Entre deux messages, la consigne reste constante et le système asservi continue à utiliser cette valeur pour la commande.

1.4.1.2.4 Modèle d'un objet connecté à un bus de terrain

Les bus de terrain ont généralement un débit plus élevé que les réseaux de collecte. Il est nécessaire de faire comme dans le cas d'un capteur connecté, de choisir des variables à observer et d'avoir un algorithme d'agrégation et/ou un algorithme de filtrage pour générer des événements ou un rafraîchissement périodique.

Il est possible d'envoyer un message sur le bus de terrain qui provient de l'interface de communication. Dans ce cas on se retrouve dans le cas d'un actionneur connecté.

1.4.1.3 Enjeux

Depuis le début de l'étude des réseaux de capteurs sans fil, l'économie d'énergie est une contrainte forte du cahier des charges qui influence les choix technologiques de l'ensemble du système.

Les objets connectés autonomes en énergie sont utiles car leur déploiement est facilité dans un envi-

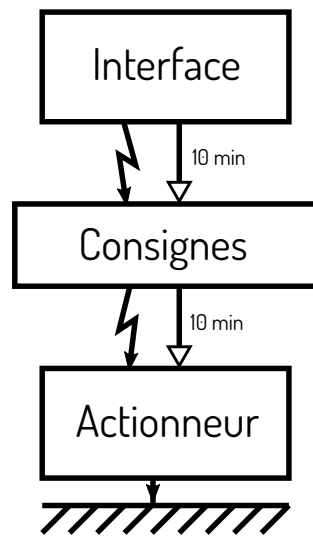


FIGURE 1.15 – Modèle des mécanismes réseaux d'un actionneur connecté

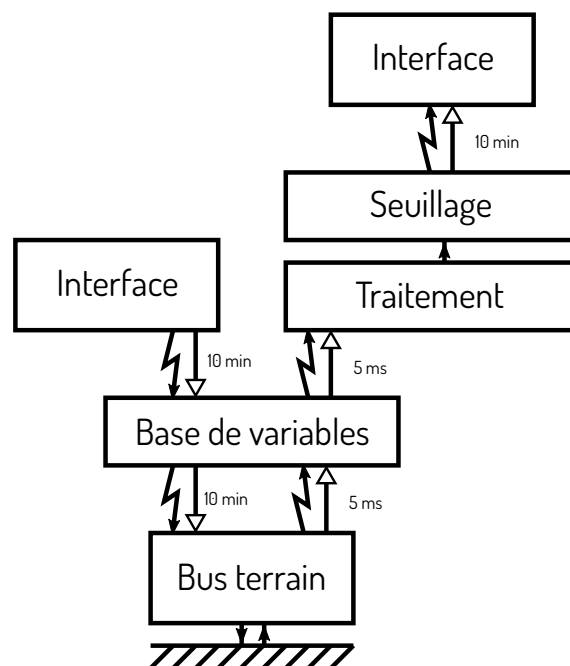


FIGURE 1.16 – Modèle des mécanismes réseaux d'un objet connecté avec un bus de terrain

ronnement qui n'a pas été conçu pour les recevoir. En effet, lors du déploiement de capteurs dans une usine par exemple, la présence des objets connectés n'a pas été pensée lors de la conception et donc les alimentations et les infrastructures réseaux sont inexistantes. Dans ce cas de figure, le déploiement de capteurs sans fil autonomes en énergie permet une intégration plus facile.

Comme ces agents sont les plus nombreux, il est obligatoire de bien les dimensionner pour ne pas avoir des coûts fixes et des coûts de maintenance qui pourraient remettre en cause la viabilité de l'application. Dans la plupart des projets IoT, le travail le plus important et le plus critique réside dans l'ingénierie des systèmes embarqués des agents de collecte.

Considérons par exemple un système de localisation en intérieur qui nécessite de déposer des objets fixes qui servent de référence pour le positionnement. Dans une usine où le déploiement de ce réseau d'« ancrés » n'a pas été prévu, aucune infrastructure d'alimentation n'a non plus été prévue. Comme le coût d'installation d'un chemin de câble pour l'alimentation est élevé, la tentation est forte de déployer les ancrés sur batterie. Cependant ce choix entraîne un changement des batteries très régulièrement qui représente un coût probablement plus important à long terme.

1.4.2 La couche de collecte des informations

1.4.2.1 Définition

L'étage de collecte des informations consiste à transporter les données des objets connectés qui sont dans l'environnement vers les serveurs placés dans un datacenter.

Les messages peuvent transiter par un réseau complet géré par une entreprise, dans ce cas c'est un réseau de collecte dit privé.

Les messages peuvent également transiter par d'autres types de réseaux comme les réseaux opérateurs.

Afin de supporter les communications avec des objets connectés multi-technologie, les réseaux de collecte se doivent d'être multi-technologie.

Les serveurs étant reliés par des réseaux IP, les réseaux de collecte sont forcément des réseaux qui convergent vers le réseau IP.

Les objets connectés n'étant pour la plupart pas capables de transmettre sur les réseaux IP, il est nécessaire de définir la fonction de passerelle, élément essentiel des réseaux de collecte.

1.4.2.2 Réseaux de collecte filaires

Les réseaux de collecte filaires sont principalement présents dans l'industrie pour convertir un bus de terrain vers le réseau IP.

Cette forme de collecte est réalisée par des passerelles physiques qui font la translation des mécanismes de communication dans l'environnement industriel. Ces passerelles peuvent être de deux types :

- Passives : les passerelles passives sont reliées au bus de terrain et ne participent pas aux communications, elles espionnent les communications entre différents équipements et sont capables de rapporter ces informations via le réseau IP.
- Actives : les passerelles actives sont capables d'émettre des messages sur le bus de terrain afin d'interroger un équipement et de renvoyer sa réponse sur le réseau IP.

Ces passerelles sont déployées depuis longtemps dans les installations industrielles, il n'y a pas beaucoup d'enjeux nouveaux dans ce secteur. Cependant nous allons détailler dans le cadre de notre étude comment faire évoluer les stratégies de communication de ces passerelles.

Les passerelles les plus répandues dans l'industrie effectuent la translation entre le protocole Modbus RTU sur un liaison RS-485 et Modbus TCP sur un réseau Ethernet.

1.4.2.3 Réseaux de collecte sans fil

Les réseaux de collecte sans fil sont en plein essor car ils permettent de venir collecter les données sans avoir un accès physique avec les objets connectés.

Exemple : Un opérateur peut venir collecter les informations des capteurs qui sont dans une usine sans avoir un équipement ou un lien filaire avec les équipements de l'usine.

Afin de pouvoir réaliser un réseau sans fil et plus particulièrement un réseau de collecte, il est nécessaire d'implémenter au minimum les trois couches les plus basses du modèle OSI : la couche physique, la couche de contrôle d'accès au médium et la couche réseau.

1.4.2.3.1 Couche physique

La couche physique pour les réseaux sans fil, consiste à avoir un ensemble de composants électroniques (des convertisseurs numérique/analogique et analogique/numérique, des amplificateurs, des filtres et des antennes) afin de pouvoir rayonner de l'énergie dans une zone physique.

L'émission de cette énergie est modulée afin de transmettre des informations binaires correspondantes au message que l'on souhaite transmettre.

Cette fonction est souvent remplie par un composant spécialisé que l'on appelle transmetteur ou transceiver radiofréquence. Ce composant intègre des modulations spécifiques, il faut ainsi choisir un ou plusieurs transmetteurs suivant les technologies sans fil.

Cette couche physique peut aussi être réalisée en partie sous forme de radio logicielle (Software Defined Radio) [28].

1.4.2.3.2 Couche d'accès au médium

La couche d'accès au médium permet de cadencer les émissions d'énergie sur le médium via la couche physique. Il ne faut pas oublier que plusieurs émetteurs vont être présents simultanément sur le médium. Cette couche permet d'éviter que plusieurs noeuds émettent simultanément sur le médium des énergies qui seraient destructives, c'est-à-dire qu'un récepteur ne pourrait pas être en mesure de décoder un message parmi les différents messages.

Cette couche peut être implémentée de façon matérielle dans un composant dédiée comme celui implémentant la couche physique ou de façon logicielle dans un système programmable.

Cette couche doit respecter des contraintes temporelles fortes sur l'émission et la réception des messages afin de respecter les temps de transmission sur le médium.

1.4.2.3.3 Couche réseau

La couche réseau permet de gérer les mécanismes réseaux qui transmettent des paquets à un nœud qui n'est pas un voisin direct à un saut.

Afin de réaliser cela, la couche réseau implémente des algorithmes de routage et d'adressage.

Cette couche est généralement implémentée de façon logicielle et est beaucoup moins stricte temporellement que la couche d'accès au médium.

1.4.2.4 Modèle

Afin de caractériser ces différents étages de collecte des informations, nous proposons différents modèles usuels.

1.4.2.4.1 Réseau de collecte avec passerelle intégrée

Dans ce contexte, une passerelle est un équipement informatique qui intègre des interfaces de communication avec les objets connectés et qui a une interface IP pour communiquer avec des serveurs hébergés dans un datacenter.

Ce type de passerelles a une interface simple pour envoyer un tableau d'octets en indiquant les paramètres de l'émission (modulation, puissance, adresse etc.). Cette interface est de haut niveau parce que nous sommes au niveau applicatif, c'est-à-dire que nous ne pouvons pas utiliser cette interface pour lier des réseaux entre eux comme on le ferait avec un routeur par exemple.

La figure 1.17 représente un schéma classique de réseau de collecte avec passerelle intégrée. La passerelle à gauche est modélisée avec ses deux interfaces de communication, l'interface de collecte à gauche et l'interface IP du backbone à droite. L'interface de collecte peut être une interface radio comme un bus de terrain etc. Les flèches représentent le trajet des messages qui transitent entre les objets connectés situés dans le réseau de collecte et les applications situées côté serveur. La passerelle comme son nom l'indique fait *passer* les messages d'une interface à l'autre.

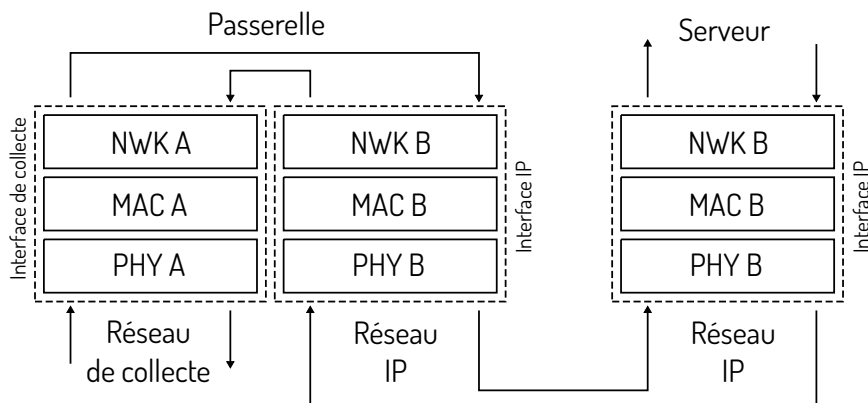


FIGURE 1.17 – Modèle d'une passerelle avec empilement protocolaire intégré

1.4.2.4.2 Réseau de collecte à couche réseau déportée

Les passerelles à couche réseau déportée permettent d'avoir des réseaux de passerelles. Comme on peut le voir sur la figure 1.18 l'empilement protocolaire est scindé en deux étages. Un étage à fortes contraintes temporelles (MAC+physique) qui est dans l'équipement d'infrastructure et un étage réseau qui est déporté et souvent centralisé sur un serveur.

De cette façon, l'interface qui est vue par les serveurs est un ensemble de passerelles à qui on peut envoyer et recevoir des trames qui vont être émises avec des paramètres spécifiques (modulation, puissance,

fréquence ...) et également un marqueur temporel afin d'indiquer à quel moment dans le futur on aimerait que le message soit envoyé.

La couche MAC qui est embarquée sur la passerelle va traiter une file d'attente de messages et essayer de les envoyer au moment désiré.

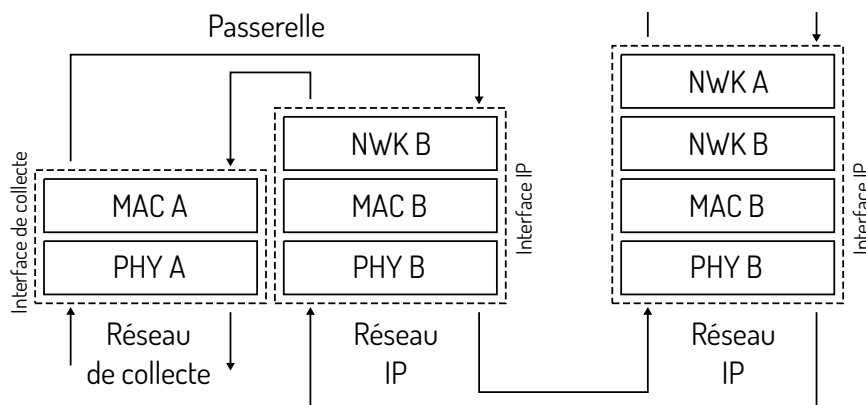


FIGURE 1.18 – Modèle d'une passerelle avec couche réseau déportée

1.4.2.4.3 Réseau de collecte à couche MAC déportée

Les passerelles à couches MAC déportées ne sont de nos jours que très rarement mises en œuvre à cause des performances des réseaux IP (essentiellement 3G/4G) pour les passerelles.

Le concept de ces passerelles est de venir numériser une bande de fréquence, de diffuser ce flux d'échantillons et de concevoir les algorithmes de traitement du signal au niveau des serveurs. Aujourd'hui ces algorithmes sont effectués en local sur la passerelle, mais le débit nécessaire sur le coeur de réseau est relativement faible pour pouvoir imaginer les déporter sur des serveurs. De cette façon l'électronique qui est présente dans les passerelles est générique et on peut utiliser la puissance de calcul des serveurs pour exécuter des algorithmes de traitement du signal beaucoup plus gourmands et même de corrélérer les numérisations des différentes passerelles. N'ayant pas les compétences requises en radio logicielle, nous n'avons pas cherché à implémenter cette idée originale mais nous la proposons en tant que perspective de recherche.

La figure 1.19 représente une passerelle à couche MAC déportée. Il ne lui reste plus que la couche physique de son réseau de collecte et une interface IP afin de transmettre le flot d'échantillons numérisés qui proviennent du front-end radio. Côté serveur, c'est des logiciels qui traitent ces flots d'échantillons afin d'en extraire des messages cohérents.

1.4.2.5 Enjeux

Les enjeux au niveau de l'étage de collecte sont très complexes car c'est un étage qui doit faire converger des réseaux très variés vers le réseau IP.

La topologie de ces réseaux n'est pas un arbre car un noeud peut avoir plusieurs interfaces de communication et donc être connecté sur plusieurs réseaux simultanément.

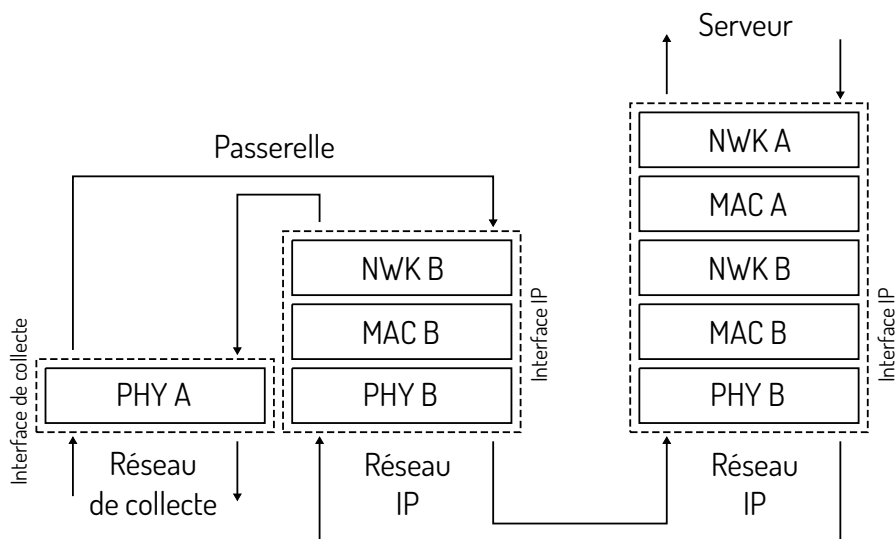


FIGURE 1.19 – Modèle d’une passerelle avec couche MAC déportée

L’évolution des réseaux IP et les caractéristiques de certains réseaux de collecte, vont permettre de créer de nouveaux étages de collecte où l’on va pouvoir déporter et virtualiser de plus en plus de services des passerelles dans des serveurs.

1.4.3 La couche de traitement des informations

1.4.3.1 Définition

L’étage de traitement des informations concerne l’ensemble des processus qui sont exécutés par des serveurs et qui prennent en entrée des flux de messages provenant des objets connectés et/ou des passerelles et sont capables, via différentes opérations successives, d’en extraire des informations à destination des applications métiers.

1.4.3.2 Caractéristiques

Les messages arrivent en entrée en flux tendu, c’est-à-dire qu’il n’y a pas d’élément de stockage dans les étages inférieurs. Un message qui arrive en entrée de l’étage de traitement est un message qui a été reçu il y a peu de temps, typiquement moins de 2 secondes. Un message qui sort de cet étage va être transmis dans un futur proche, typiquement 5 secondes. Les valeurs de 2 et 5 secondes ne sont pas réelles et servent uniquement d’exemple.

A partir de ce moment-là, nous nous retrouvons dans une architecture typique du big data : l’architecture lambda [29]. En effet, les messages qui arrivent par flux peuvent être assimilés à des flux de notifications ou d’événements comme on le retrouve de plus en plus dans les architectures web de traitement des notifications mobiles.

1.4.3.3 Modèle

L'architecture lambda est constituée de deux étages de traitement des données :

- Un étage rapide en flux tendu de traitement par flux,
- Un étage lent de traitement par lots

La figure 1.20 représente l'étage rapide à droite et l'étage lent à gauche. Le flux temps-réel qui arrive des objets par le bas, est dupliqué à destination de chaque étage.

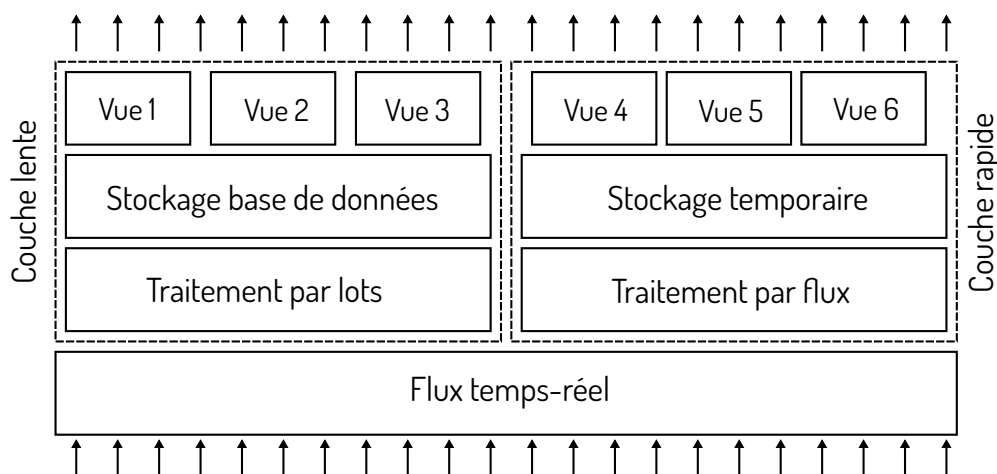


FIGURE 1.20 – Architecture lambda pour le traitement des données

L'étage rapide est un étage où il n'y a quasiment pas de stockage persistant, c'est-à-dire que les messages circulent entre les processus et ne sont stockés que temporairement dans des files d'attente le long du chemin.

L'étage lent est un étage de stockage d'un énorme volume de données et dans lequel des algorithmes vont essayer de digérer les données afin d'en extraire les informations essentielles.

Ces deux étages génèrent des vues qui sont accessibles par les applications pour venir récupérer les données.

1.4.3.4 Enjeux

Les flux provenant des flottes d'objets arrivent généralement au fil de l'eau et la plateforme doit pouvoir passer à l'échelle afin de supporter la charge.

La sécurité au sens cryptographique du terme est un enjeu important de ces plateformes, car l'identification, l'authentification des objets et le chiffrement des messages sont difficiles à mettre en oeuvre avec des objets aussi contraints.

L'infrastructure nécessaire pour ce genre d'architecture peut être très imposante et est aujourd'hui favorisée par l'émergence des plateformes de cloud computing où l'infrastructure peut passer à l'échelle en fonction de l'évolution de l'application.

1.5 Première piste d'architecture interopérable

Node-RED est un éditeur visuel open-source conçu par IBM, dédié principalement à l'Internet des objets. Ce logiciel implémente un mécanisme d'interopérabilité intéressant pour notre étude.

La force de Node-RED est d'avoir utilisé nodejs, un interpréteur javascript qui est exécuté côté serveur. De cette façon le code qui est exécuté côté client (navigateur web) et celui exécuté côté serveur sont très similaire et les interactions sont aisées.

L'interface web permet d'éditer de façon graphique et textuelle, des suites de blocs d'acquisition et transmission de données, ainsi que des traitements intermédiaires.

Lorsque l'édition est terminée, la configuration est envoyée au serveur pour qu'il exécute le programme, tout en envoyant les informations au navigateur web pour les visualiser.

Node-RED étant exclusivement écrit en javascript, les messages qui sont transmis à entre les composants internes sont des objets javascript.

Les objets javascript sont très facilement sérialisables en format JSON donc on peut dire que les messages à l'intérieur de Node-RED sont au format JSON.

Le format minimal d'un message est sous cette forme :

```
1  {
2      "_msgid": "12345",
3      "payload": "...",
4  }
5
```

FIGURE 1.21 – Exemple de format de message au sein de Node-RED

Le champ « `_msgid` » est l'identifiant unique du message et le champ « `payload` » est le champ qui contient la charge utile de message. Le type de cette charge utile n'est pas défini, il est possible de transporter une liste, un dictionnaire, un entier, une chaîne de caractères, etc.

1.5.1 Implémentation du modèle de graphe de flux

Node-RED se base sur un paradigme de graphe de flux orienté objets. Les objets sont des entités qui ont une ou plusieurs entrées, une sortie, une fonction d'exécution et des variables internes.

L'idée est que la fonction interne soit exécutée à chaque fois qu'un message arrive sur une entrée, ce qui permet alors de choisir d'émettre ou non un message en sortie.

Ce mode de programmation est essentiellement événementiel et dirigé par les messages.

Ces blocs génériques peuvent être instanciés et reliés entre eux par des connexions. Ces connexions signifient que lorsque qu'un bloc émet un message en sortie tous les blocs qui sont connectés à cette sortie vont recevoir le message.

L'architecture finale représente un pipeline de traitement dans lequel passe les flux de messages. Il y a une ou plusieurs entrées du graphe de flux et une ou plusieurs sorties comme nous pouvons le voir sur la figure 1.22.

Nous avons expérimenté de nombreuses architectures à l'aide de Node-RED et ce logiciel répond très justement à notre vision de l'Internet des Objets d'une part via le modèle de graphe de flux et d'autre

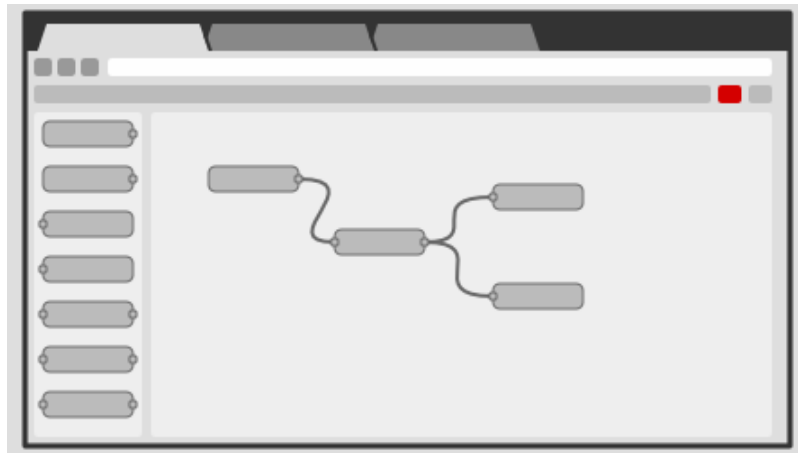


FIGURE 1.22 – Principe de l'interface de Node-RED

part via l'interopérabilité en unifiant le format des messages.

1.5.2 Limitation de Node-RED

La limitation principale de Node-RED est qu'il n'a pas été pensé pour être intégré dans des architectures haute-disponibilité ou complexes.

C'est un logiciel qui contient des variables internes et qui ne propose aucune solution pour les partager entre plusieurs entités (clustering).

De plus même si de base l'idée de Node-RED est d'être interopérable avec toutes sortes d'équipements et de logiciels, le choix d'avoir standardisé le coeur de Node-RED en javascript rend difficile des extensions avec d'autres langages.

1.5.3 Conclusion

Node-RED est un logiciel qui est très proche de notre idée de l'Internet des Objets. Une architecture serveur pour l'IoT doit être capable d'ingérer des flots de données et de communiquer ces données à de nombreux services comme des bases de données, des équipements physiques, des réseaux, des interfaces hommes-machine, etc.

Node-RED réalise très bien tout cela, mais pour nous, le problème a été résolu dans des couches trop élevées du modèle OSI. La normalisation des messages entre les agents se doit d'être le plus bas possible afin d'être indifférent des langages, des logiciels et des mécanismes les plus basiques afin de ne pas contraindre les usages.

1.6 Système de messagerie pour l'Internet des Objets

Pourquoi ne pas se satisfaire de quelque chose qui est si proche du but ? Quels sont les éléments présents dans Node-RED qu'il faut remplacer et par quoi ?

1.6.1 Le format et le type des messages

Le format des messages dans Node-RED est problématique parce que les trames et les paquets tels qu'ils circulent dans les réseaux sont transformés pour être interprétés comme des objets javascript.

La conséquence est que nous perdons les apports importants de garder une approche réseau qui est de s'appuyer sur les protocoles TCP/IP afin de garantir l'interopérabilité.

Il est important de garder à l'esprit que ce qui transite entre les différents agents de l'architecture sont des messages qui sont au format des protocoles qui les composent.

Ainsi un agent n'est qu'un logiciel qui sait faire la transcription d'un protocole vers un autre qui peut être le même, en réalisant un traitement au milieu.

Nous proposons de standardiser ces flots de messages au plus proche de la couche transport afin de garder cette vision réseau de la messagerie.

1.6.2 Représentation au format JSON

Le format JSON est un format de données textuelles, qui est organisé sous la forme d'arbre mais qui est moins volumineux que le XML par exemple.

Voici un exemple de message au format JSON :

```
1  {
2    "Prenom": "Nicolas",
3    "Nom": "Gonzalez",
4    "Age": 26,
5    "Parents": ["Lora", "Victorino"],
6    "Diplomes": [
7      {"Master": "Ingenieur"},
8      {"BAC+2": "DUT"}
9    ]
10 }
11
```

FIGURE 1.23 – Exemple de message avec la représentation JSON

Cet exemple permet de voir les différents types de données que l'on peut représenter dans un message JSON ainsi que la structuration des informations.

Il est possible de représenter les types suivants :

- Chaîne de caractères
- Nombre
- Booléen
- Type null
- Tableau
- Dictionnaire

Il est possible de structurer les données en utilisant les types suivants :

- Tableau
- Dictionnaire

Standardiser ce format au niveau de la couche transport permet de garder l'idée et les fonctionnalités de Node-RED mais à un niveau dans l'empilement protocolaire qui est mieux standardisé.

Le format JSON étant au format ASCII, il est peu rentable de l'utiliser pour transporter des données binaires. Cependant, dans le contexte de l'Internet des Objets, la taille des trames est très inférieure à la taille des paquets IP. C'est pour cela qu'il est très facile d'envisager de les convertir en chaîne de caractères hexadécimaux et de les transmettre sous un format JSON.

1.6.3 Limitations des capacités des objets

Nous avons expliqué que les architectures typiques sont formées d'un ensemble d'agents qui communiquent entre eux via des protocoles standardisés. Ces architectures sont essentiellement virtualisées et par nature évolutives.

Les objets communicants sont très contraints au niveau processeur, énergie, capacités réseaux, sécurité, etc. Ils embarquent des logiciels qui sont beaucoup plus stables et moins évolutifs que ceux que l'on peut déployer côté serveur. Les objets vont difficilement pouvoir s'adapter à la dynamique d'intégration côté serveur.

Par exemple, un capteur de température qui communique en unicast avec une application, peut devoir gérer de nouvelles connexions unicasts lors du déploiement de nouvelles applications. Ceci veut dire gérer les multiples mécanismes de connexion, d'acquiescement, de retransmission qui sont assez complexes pour des objets contraints.

1.6.4 Nécessité d'un intermédiaire

Les contraintes fortes des objets communicants face à l'évolutivité rapide des architectures serveurs, vont dans le sens de la création d'un service réseau intermédiaire.

En effet, nous pensons qu'un service intermédiaire de messagerie permet de casser les liens logiques entre les agents et de simplifier les paradigmes réseaux.

Faisons l'analogie avec le service postal, il est très rare que l'on effectue un acquiescement de bout en bout lorsque l'on envoie une lettre. Ce qui veut dire que l'on attend une information comme quoi notre destinataire a bien reçu la lettre. Pourquoi? C'est parce que nous avons confiance en le facteur qui est assez fiable et surtout qui est capable de gérer dans une certaine mesure une qualité de service en cas de problème.

Cet intermédiaire permet de régler de nombreux problèmes, comme le transport des lettres, la résolution de l'adresse, les tentatives de livraisons, la redirection, etc. Ces problèmes sont trop complexes pour les personnes qui veulent envoyer du courrier, le service postal permet via le modèle boîte noire de cacher cette complexité et de permettre d'envoyer du courrier simplement.

Les objets eux aussi sont contraints et ces mécanismes réseaux sont trop complexes pour eux, il est nécessaire de créer un service de messagerie intermédiaire.

La difficulté supplémentaire des objets communicants est que les données qui sont transmises (les

lettres par analogie) sont souvent destinées à une ou plusieurs personnes et cela peut changer dans le temps.

Notre service de messagerie va devoir être capable de gérer cet aspect multicast à la demande. Notre analogie du service postal n'est plus tout à fait juste car il faudrait que la poste soit capable de dupliquer nos lettres afin d'envoyer une copie à chaque destinataire !

Nos besoins se rapprochent plutôt d'une nouvelle analogie qui est celle d'un système d'abonnement à une revue. Suivant différents centres d'intérêts, nous souscrivons à différentes revues car nous avons analysé que ces revues proposent des articles sur un ensemble de nos centres d'intérêts. En souscrivant à ces revues, nous indiquons à un éditeur que nous voulons à partir de maintenant recevoir les revues qui sont sur cette thématique très précise.

Maintenant mettons-nous à la place d'une personne qui rédige un article sur un thème très précis. Au lieu de chercher à rentrer en contact avec toutes les personnes qui sont intéressées par cette thématique, il va rencontrer un éditeur qui possède des revues sur le sujet et lui proposer son article. De cette façon, c'est l'éditeur qui est l'intermédiaire de messagerie, mais ici le paradigme est différent du service postal car on est dans des communications qui sont plutôt en diffusion et non plus sur des communication de pair à pair.

Alors quel est le paradigme à privilégier pour l'Internet des Objets ? Nous pensons que c'est la méthode de diffusion car il est facile de venir ajouter de nouveaux agents tout en étant transparent pour les autres. Cette méthodologie permet de venir inspecter les flux en rajoutant un observateur ou des sondes de métrologie et de supervision.

1.7 Garantir l'interopérabilité à l'aide du protocole MQTT

1.7.1 Intérêts

Le paradigme de diffusion dans le domaine des logiciels de messagerie a été très étudié et il existe de nombreux protocoles. Celui qui a retenu notre attention est MQTT (Message Queuing Telemetry Transport) qui est un protocole assez ancien mais qui est redevenu très actuel depuis les problématiques liées à l'Internet des Objets.

Ce protocole nous intéresse car il intègre le paradigme d'éditeur/abonné (publish/subscribe) et qu'il permet ces fonctionnalités très bas dans l'empilement protocolaire.

1.7.2 Paradigme Publish/Subscribe

L'analogie de l'éditeur de revue permet de bien comprendre le paradigme de publish/subscribe.

C'est une manière de casser le lien entre les agents via l'ajout d'un agent intermédiaire. Chaque agent est connecté à l'agent intermédiaire et gère avec lui des mécanismes de communication pour fiabiliser les échanges. Et via cette unique connexion, il est possible de transmettre des messages à publier, d'en recevoir, de vérifier que l'intermédiaire a bien traité votre message etc.

De cette manière on a un agent de messagerie, comme le service postal, qui gère la complexité du transport de messages et qui permet à chacun d'envoyer des messages aux autres agents.

Le mécanisme de publish/subscribe repense également la notion de message. Lorsque l'on veut envoyer une lettre à quelqu'un en particulier, il est nécessaire de rédiger cette lettre, la mettre dans une enveloppe et écrire l'adresse exacte du destinataire. Le service postal ajoute un cachet sur l'enveloppe qui permet

d'identifier de façon unique le courrier, dès lors le service postal connaît le destinataire du courrier, parfois l'émetteur et il a identifié le courrier de façon unique : il peut réaliser son service de transport du message sans avoir à connaître le contenu du message.

Avec le paradigme de publish/subscribe, comme nous ne connaissons pas les destinataires, nous devons indiquer sur l'enveloppe des mots clés qui vont donner des informations au service intermédiaire afin de distribuer le message. De cette façon le service de messagerie va être capable de reconnaître les agents qui souhaitent recevoir les messages car ils se sont au préalable abonnés à ces différents mots clés.

1.7.3 Implémentation des mots clés dans MQTT

Dans MQTT, un message est un ensemble binaire de 268435456 octets (256Mio) maximum qui constitue la charge utile et d'un *topic* qui constitue l'ensemble des mots clés.

Un *topic* est une chaîne de caractères qui représente une arborescence de mots-clés. Par exemple :

```
sport
  /football
    /masculin
    /feminin
  /rugby
  /golf
  /equitation
musique
  /rock
    /psychedelique
    /rockandroll
    /hardrock
  /rap
  /classique
  /jazz
people
  /cinema
  /musique
  /theatre
```

FIGURE 1.24 – Exemple d'arborescence de topics MQTT

En utilisant cette arborescence il est possible de créer des topics. Par exemple `musique/rock/hardrock` signifie que l'on souhaite s'abonner à tous les messages concernant le hardrock. Un topic est une chaîne de caractères avec des '/' qui signifie les différents étages de l'arborescence, comme une arborescence de fichiers ou un URL par exemple.

Il est possible d'utiliser différents jokers comme le '+' qui signifie qu'importe ce qui est défini à cet étage. Par exemple, `sport/+feminin` signifie que l'on s'intéresse à tous les sports féminins.

Le deuxième joker est le '#' qui permet de s'abonner à tous les fils de l'arborescence. Par exemple, `people/#` signifie que l'on s'abonne à tous les magazines peuples.

1.7.4 Implémentation de la couche transport

MQTT est un protocole à faible overhead qui est basé sur TCP. Chaque client établit une connexion TCP avec un serveur central qui est appelé « broker ». Une fois la connexion établie, le protocole MQTT va permettre de réaliser tous les mécanismes que l'on a évoqué :

- CONNECT : connexion de niveau MQTT entre le client et le broker
- CONNACK : acquittement de connexion du broker
- PUBLISH : publication d'un message du client vers le broker
- PUBACK : acquittement de prise en charge d'un message par le broker
- PUBREC : acquittement de transmission d'un message par le broker (si QoS 2)
- PUBREL : acquittement de PUBREC par un client
- PUBCOMP : dernier message d'une transaction PUBREC
- SUBSCRIBE : abonnement à un topic par un client
- SUBACK : acquittement de l'abonnement à un topic par le serveur
- UNSUBSCRIBE : désabonnement d'un topic par un client
- UNSUBACK : acquittement du désabonnement par un serveur
- PINGREQ : requête de ping afin de garder la connexion par le client
- PINGRESP : réponse de ping par le serveur
- DISCONNECT : message de déconnexion par le client

Il est intéressant de noter deux types de messages :

- Les messages d'établissement, d'arrêt et de maintien de la connexion
- Des messages pour la gestion du paradigme publish/subscribe

Ce protocole très simple permet d'avoir une couche de transport TCP avec l'ajout du mécanisme publish/subscribe.

1.7.5 Utilisation de MQTT dans le cadre de l'Internet des Objets

MQTT semble très approprié pour notre système de messagerie car il est assez bas dans le modèle OSI, c'est-à-dire juste au dessus ou dans une sous couche haute de la couche transport. Cette couche est considérée comme basse car c'est la première qui est disponible sur un système d'exploitation depuis l'espace utilisateur.

1.7.6 Couche de transport augmentée en espace utilisateur

L'Internet des Objets est pour nous une expression qui n'est pas très précise. En effet, il est difficile d'expliquer à un industriel que ses machines vont être connectées sur Internet. Nous pensons qu'il s'agit plutôt de l'Intranet des objets, c'est-à-dire l'art de connecter des systèmes embarqués qui interagissent avec l'environnement physique et avec des systèmes informatique au travers de la norme principale de communication qui sont les protocoles TCP/IP.

Casser cette volonté revient à dire qu'il faudrait inventer de nouveaux réseaux ou de nouveaux protocoles pour faire communiquer les objets et les systèmes informatique. Au niveau des couches basses, les protocoles comme IP ou UDP/TCP sont devenus la norme d'interopérabilité entre les systèmes informatique, les ordinateurs personnels, les smartphones, les tablettes etc. De nombreux équipements réseaux sont également très liés à ces protocoles au niveau matériel pour faire fonctionner ce que l'on nomme Internet.

Étudier l'interopérabilité entre les objets et ces systèmes informatique revient donc bien à analyser ces protocoles existants, analyser les contraintes spécifiques des objets et de trouver des moyens intelligents de les faire coexister.

Le protocole IP est un socle solide pour la transmission de messages au niveau global, il gère notamment l'adressage et permet à chaque agent du réseau d'envoyer un message à un autre agent du réseau. L'Internet des Objets pose la question de la scalabilité d'Internet, du nombre maximal d'adresses disponibles mais ces questions ont déjà été traitées au niveau IP via l'élaboration de IPv6.

Les enjeux se situent dans la couche immédiatement supérieure : la couche transport. En effet, comme nous l'avons expliqué, les mécanismes réseaux utilisés dans l'IoT sont principalement multicast. Cette approche multicast est toujours complexe en réseau car lorsque l'on veut envoyer un message à beaucoup de destinataires, il est difficile de réaliser un acquittement avec chacun des destinataires. Cela entraîne de nombreux messages d'acquittement qui sont transmis à l'émetteur à la suite de chaque message diffusé. Un objet contraint en énergie, en mémoire et en performances ne peut pas gérer ce genre de mécanismes réseau. Même sur des systèmes informatique plus performants, il est difficile de tenir la cadence et les problématiques de passage à l'échelle sont très critiques.

Les enjeux se situent donc là, c'est-à-dire dans la couche transport, pour répondre à la question de savoir comment être capable d'avoir des mécanismes de diffusion de messages qui soient robustes et qui puissent passer à l'échelle.

Il est intéressant de souligner que dans les systèmes d'exploitation actuels, la couche de transport (TCP et UDP par exemple) est un service fourni en environnement noyau c'est-à-dire par le système d'exploitation lui-même. Les difficultés à faire évoluer ces couches critiques des systèmes d'exploitation font qu'aujourd'hui l'innovation sur ces couches transports améliorées, se fait en espace utilisateur.

1.7.7 Gestion du multicast par la QoS

Lorsque qu'un message est publié par un émetteur et que plusieurs destinataires sont abonnés à ce topic MQTT, le broker va envoyer une copie de ce message à chacun. C'est à ce moment là que le broker gère l'acquittement avec tous les destinataires.

Comme nous l'avons expliqué, l'acquittement de messages en multicast est assez lourd au niveau overhead et nombre de messages transmis. MQTT prévoit une notion de QoS afin de choisir les garanties de gestion de l'acquittement.

- QoS 0 : Le niveau de service le plus bas est le plus rapide car un seul message est transmis, par

contre il n'est pas acquitté donc le broker ne peut pas avoir de garantie sur le fait que le message ait été reçu par les objets qui ont souscrit (figure 1.25).

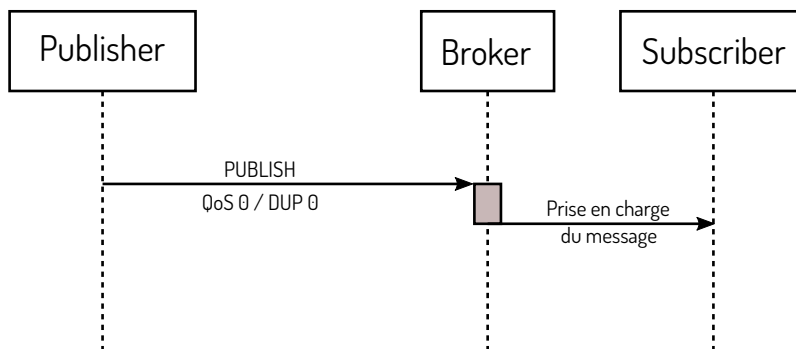


FIGURE 1.25 – MQTT avec une QoS de 0

- QoS 1 : Dans le niveau 1, le destinataire va émettre un acquittement et le broker va attendre cet acquittement (figure 1.26). Si le broker ne reçoit pas d'acquittement, il va rémettre le message plusieurs fois. Si l'acquittement et le message réémis se croisent, le destinataire va recevoir plusieurs fois le message (figure 1.27).

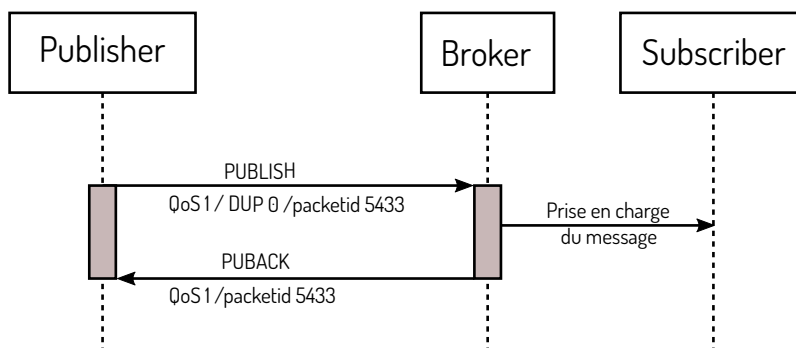


FIGURE 1.26 – MQTT avec une QoS de 1

- QoS 2 : Le niveau le plus élevé de QoS permet d'avoir des garanties sur le fait que chaque destinataire ait reçu le message et sans duplication.

Dans ce mode, un premier échange permet de transmettre le message et de l'acquitter et un deuxième échange permet de se mettre d'accord sur la libération de l'identifiant du message. De cette façon, les deux interlocuteurs peuvent avoir la garantie qu'une fois cette libération effectuée, aucun message ne va être retransmis avec le même identifiant et qui serait un duplicata du premier. La figure 1.28 montre le diagramme de séquence des différents échanges.

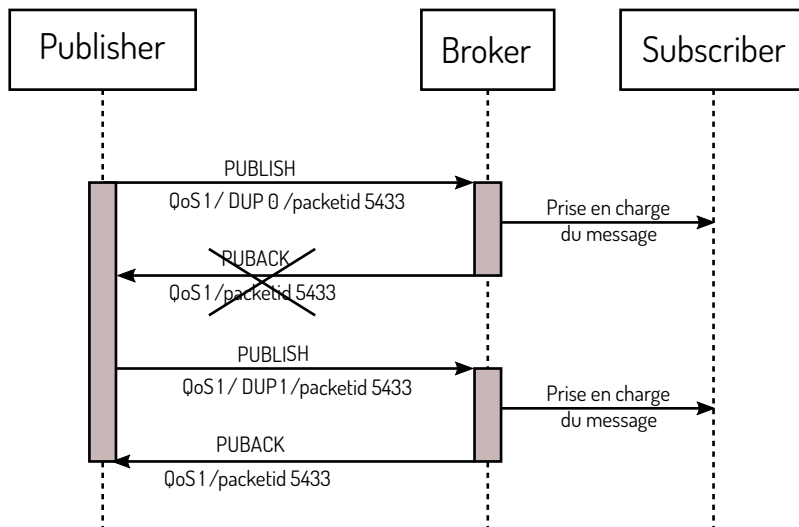


FIGURE 1.27 – Problématique de retransmission avec une QoS de 1

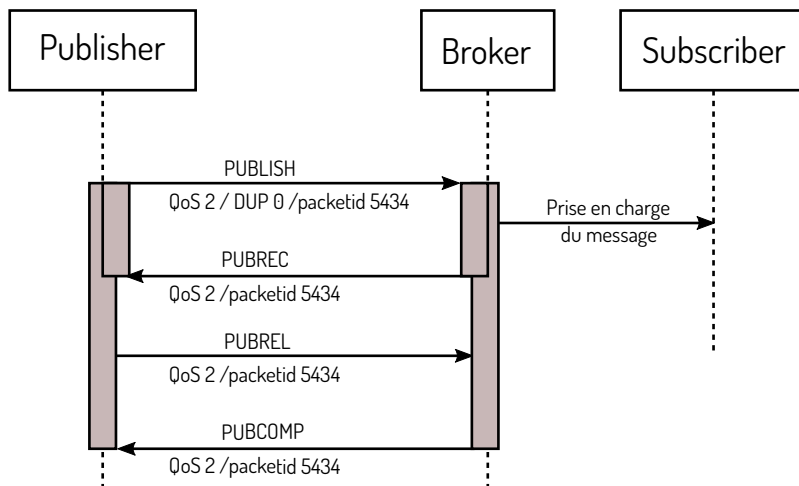


FIGURE 1.28 – MQTT avec une QoS de 2

1.8 Conclusion

Dans ce premier chapitre, nous avons tout d'abord retracé les origines de l'Internet et rappelé les concepts généraux de l'Internet des Objets, puis fait un focus sur l'Internet des Objets pour l'industrie. Nous avons précisé la problématique de l'interopérabilité et l'intérêt du multicast pour garantir l'ouverture et l'évolutivité des architectures logicielles. Nous avons enfin présenté le protocole MQTT et souligné à quel point il constitue une base très intéressante pour notre problématique.

Afin d'éprouver et d'explorer la piste de l'interopérabilité à l'aide du protocole MQTT, nous décidons d'étudier les réseaux LPWAN LoRa. Ces réseaux bas débits devraient permettre d'illustrer les concepts

liés à l'encapsulation des trames à travers Internet.

Chapitre 2

Les réseaux LPWAN : originalités, architectures et déploiement

Cette thèse a été guidée par l'émergence de nouveaux réseaux : les "Low Power Wide Area Network". Ces réseaux longue distance sont promis à supporter énormément d'objets connectés et actuellement à partager des bandes libres ou non licenciées. Ces deux aspects nécessitent de penser en amont à la problématique d'interopérabilité.

Sommaire

2.1	Origines de la démarche de recherche	55
2.1.1	Développement actif d'entreprises autour des LPWAN	55
2.1.2	Appropriation des LPWAN par le milieu académique	55
2.1.3	Les promesses des réseaux LPWAN	56
2.2	État de l'art scientifique sur les LPWAN	57
2.2.1	Période 2015/2016 : les premières publications sur les LPWAN et la couche physique LoRa	57
2.2.2	Période 2017/2018 : début de cette thèse et intensification des publications	57
2.2.3	Période 2019/2020 : déploiement des premiers réseaux de recherche	58
2.2.4	Réseaux maillés avec couche physique LoRa	58
2.3	Législation des bandes ISM	58
2.4	Évaluation des paramètres de la couche physique	60
2.4.1	Idée de départ : du multi-saut au multi-médium	60
2.4.2	Modulation LoRa	60
2.4.3	Codage de canal dans la couche physique LoRa	61
2.4.4	Temps d'émission et débit	62
2.5	Composants d'infrastructure	65
2.5.1	Passerelle LPWAN	65
2.5.2	Serveur de réseau	66
2.5.3	Serveurs applicatifs	66
2.6	Réseaux LPWAN privés et publics	68

2.6.1 Réseaux LPWAN publics	68
2.6.2 Réseaux LPWAN privés	69
2.7 Protocoles de transport des messages	69
2.7.1 Packet Forwarder	70
2.7.2 Transport des trames via MQTT	73
2.8 De la théorie à la pratique	73

2.1 Origines de la démarche de recherche

2.1.1 Développement actif d'entreprises autour des LPWAN

L'origine de l'engouement pour les réseaux LPWAN se situe en 2009 avec la création des entreprises Sigfox à Toulouse et Cycleo à Grenoble. Ces deux entreprises veulent développer une technologie radio afin de transmettre des messages sur de longues distances de façon économe en énergie.

Ils se lancent dans la course afin de spécifier les différentes couches réseaux et de développer les premiers transmetteurs matériels.

Cette phase de lancement a duré 3 à 4 ans, en effet en 2012 Cycleo a été racheté par l'américain Semtech pour 5 millions d'euros. Les premiers déploiements ont lieu entre 2013 et 2015 pour les deux technologies.

L'entreprise Snootlab sort en 2013 Akeru, une première carte Arduino de prototypage avec un module radio Sigfox de l'entreprise Telecom Design.

Sigfox est une entreprise qui a depuis le départ la stratégie de maîtriser de bout en bout ce qui lui a permis de lancer directement le déploiement d'un réseau et annonce en 2014 couvrir entièrement la France.

Pendant ce temps, en 2015 la LoRa Alliance est créée afin de rassembler les industriels intéressés par la modulation LoRa. Cette même année, les premières spécifications de LoRaWAN [30] (v1.0) sont publiées. Ces spécifications définissent les bandes de fréquences utilisées, la couche d'accès au médium et le transport des trames à travers le réseau IP.

Le premier pays à être couvert est les Pays-Bas en 2016 puis d'autres pays comme la France en 2017 et 2018 avec les réseaux d'opérateurs commerciaux comme Orange et Objenious.

Les technologies LPWAN ont pu être aussi vite développées par des entreprises de petite taille et sans historique car malgré les innovations et l'originalité de la démarche, l'ensemble des techniques utilisées se base sur un vivier technique actif depuis plusieurs dizaines d'années.

L'origine des LPWAN est de fait principalement commerciale, portée par l'émergence de l'idée d'Internet des objets, et des prédictions d'un nouveau marché aux milliards d'objets à connecter.

2.1.2 Appropriation des LPWAN par le milieu académique

Ces domaines d'étude de l'Internet des Objets et des réseaux LPWAN étant un agglomérat d'idées, de concepts et de techniques issus de différentes communautés scientifiques, il n'existe pas au démarrage de cette thèse une communauté IoT bien identifiée. Malgré cela, de nombreuses communautés n'ont pas de mal à redéfinir leurs travaux passés et présents comme appartenant aux thématiques de l'IoT.

C'est le cas de notre équipe de recherche RMESS-IRIT qui a publié de nombreux travaux de recherche principalement dans la communauté des réseaux de capteurs sans-fil (WSN).

Avant le début de cette thèse, l'équipe du laboratoire a développé et fabriqué plusieurs modules électroniques constitués d'un microcontrôleur et d'un composant radio permettant d'émettre des trames via la modulation LoRa.

La piste de recherche consistait à évaluer les apports de la couche physique LoRa, qui était très récente à l'époque, dans un contexte de réseaux maillés. L'équipe de recherche a historiquement publié de nombreux travaux sur les réseaux maillés [31], la synchronisation temporelle [32] dans ces réseaux ainsi que sur le développement de différentes couches MAC [33] et ceci depuis de très nombreuses années depuis la thèse de Thierry Val à l'UBP de Clermont Ferrand. Ces travaux ont été menés principalement depuis la thèse

d'Adrien Van Den Bossche sur des technologies issues de la norme MAC IEEE802.15.4.

L'étude de la couche physique LoRa constituait une piste de recherche attrayante car les caractéristiques annoncées de liaisons très longues distances pour un bilan de puissance très faible offrait des perspectives exaltantes pour les réseaux de capteurs sans fil.

Dans les réseaux de capteurs répartis en topologie maillée la conclusion des nombreux travaux qui ont été menés, notamment dans le cadre du développement des technologies IEEE802.15.4 et des couches supérieures comme celui de Zigbee, montre un passage à l'échelle délicat et complexe.

2.1.3 Les promesses des réseaux LPWAN

La promesse des réseaux LPWAN consiste à être capable de transmettre de petites trames (10 à 100 octets) sur de longues distances (1 à 3 km) et de façon économe en énergie.

La notion d'économie d'énergie est assez vague et nécessite d'être décrite plus clairement. Le marché ciblé est celui des réseaux de capteurs sans fil autonomes. Par exemple, les capteurs d'alarme, les capteurs de fumée, les capteurs ambiants (luminosité, humidité et température) etc. Ces noeuds sont constitués de microcontrôleurs et de capteurs, l'ensemble étant alimenté par des piles ou des petites batteries.

La promesse des réseaux LPWAN est que ces systèmes puissent fonctionner sans être rechargés pendant la durée de vie des batteries, c'est à dire une dizaine d'année. Cette durée de vie est bien entendu très dépendante des conditions de mise en veille des composants mais aussi du coût énergétique d'un message et de la périodicité des messages. L'ordre de grandeur est l'émission d'une petite trame (10 à 100 octets) toutes les dizaines de minutes.

En dehors des capteurs fixes dans les maisons, il y a également de nombreuses applications de géolocalisation d'objets comme des containers, des palettes, des colis et autres équipements dans l'industrie. Au niveau applicatif, il est nécessaire d'avoir un réseau ambiant et global. C'est pour cela que Sigfox a investi via plusieurs levées de fonds successives entre 2013 et 2016 pour déployer son réseau au niveau mondial. La stratégie a été de promouvoir le fait que la couverture était mondiale et que ce genre d'applications était possible. LoRaWAN a eu du mal à fournir la même promesse au niveau mondial car elle nécessite le déploiement de roaming entre les différents opérateurs locaux.

La dernière promesse est le coût de la connectivité qui doit venir concurrencer les réseaux mobiles d'un côté et le prix faible des objets où il faut ajouter de la connectivité de l'autre côté. L'idée est que le modèle économique des opérateurs est assez lourd et que les objets ne doivent pas avoir d'abonnement pour les particuliers, ni de carte SIM mais que le coût de la connectivité soit inclus dans l'achat de l'objet et supporté par le fabricant des objets.

Voici les différentes promesses des réseaux LPWAN lorsque nous avons débuté cette thèse, le premier axe d'étude a été d'éprouver les performances de ces technologies pour vérifier si ces promesses étaient tenues.

2.2 État de l'art scientifique sur les LPWAN

2.2.1 Période 2015/2016 : les premières publications sur les LPWAN et la couche physique LoRa

Les premières publications scientifiques au sujet de LoRa apparaissent en 2014 avec l'article "Free space range measurements with Semtech LoRa™ technology" [34] qui est une première campagne d'évaluation de la couche physique LoRa. Cette étude, publiée dans "IEEE Wireless Communications" a été menée en extérieur et en ligne de vue directe avec pour objectif, une mesure du RSSI et du PER (Packet Error Rate). Elle permet de montrer un premier aperçu des performances réelles de la modulation LoRa.

L'article "Long-range communications in unlicensed bands : the rising stars in the IoT and smart city scenarios" [35] publié en novembre 2016 dans "IEEE Wireless Communications" est un des premiers surveys de référence sur la comparaison entre le paradigme des réseaux cellulaires et des réseaux LPWAN. Cet article décrit les avantages et les inconvénients de la technologie dans un contexte Smart City.

La publication "On the coverage of LPWANs : range evaluation and channel attenuation model for LoRa™ technology" [36] en 2016 dans la conférence internationale "ITS Telecommunications" complète les premiers travaux d'estimation de la portée, du bilan de liaison et de l'estimation du Packet Error Rate. L'originalité de cette étude est de réaliser les mesures à la surface de l'eau en Finlande.

2.2.2 Période 2017/2018 : début de cette thèse et intensification des publications

Les publications sur les LPWAN et plus particulièrement sur LoRa ont commencé à s'intensifier sur la période 2017/2018. Cette période coïncide avec le début de cette thèse en janvier 2017. L'année 2017 marque la fin du développement de la technologie LoRa qui s'est déroulé de 2008 à 2017. Il est intéressant de remarquer ici que les premiers travaux académiques débutent lors du lancement sur le marché de la technologie et que ces travaux sont tous liés à des évaluations de performances.

Les études précédentes sur la couche physique ont mis en évidence que la performance de ces réseaux est fortement liée à la capacité de réjection des interférences entre nœuds et de multiplexage du médium au sein de multiples canaux orthogonaux. Le papier "Interference Measurements in the European 868 MHz ISM Band with Focus on LoRa and SigFox" [37] montre que cette étude doit être menée pour une modulation donnée, mais également entre les différentes modulations car elles doivent cohabiter dans les mêmes bandes de fréquences. Notre première publication scientifique dans le cadre de cette thèse "Specificities of the LoRa physical layer for the development of new ad hoc MAC layers" [38] traite le sujet de l'orthogonalité des trames LoRa. Nous avons réalisé une série d'expériences afin d'évaluer l'orthogonalité des trames entre elles lorsqu'elles sont émises avec un spreading factor différent.

A la suite de l'étude de la couche physique, différentes publications concernant l'extensibilité et la capacité des réseaux LoRa sont apparues. Lorsque de nombreux nœuds LoRa sont déployés sur une zone donnée, il devient important d'utiliser les mécanismes de réjection des interférents et d'orthogonalité des trames. Comme nous l'avons évoqué, ces technologies doivent permettre d'émettre des messages à longue distance et être déployées en très grand nombre. L'étude de l'extensibilité du réseau en nombre de nœuds devient, dans ces conditions, primordiale. L'article "Analysis of Capacity and Scalability of the LoRa Low Power Wide Area Network Technology" [39] et "Low Power Wide Area Network Analysis : Can LoRa Scale?" [40] deviennent des articles de référence sur cette étude de l'extensibilité avec une approche très théorique.

Il apparaît sur cette période un champ d'étude des réseaux LPWAN pour un déploiement d'infra-

structure via une constellation de satellites. L'article "LEO Satellite Constellation for Internet of Things" [41] est très complet sur cette thématique et montre comment le déploiement de satellites en orbite basse permettrait d'établir une couverture mondiale.

Ainsi l'article "Low Power Wide Area Networks : An Overview" [42] publié dans "IEEE Communications Surveys and Tutorials" complète la premier survey de 2016 en y intégrant de nouvelles technologies comme 802.15.4g, WEIGHTLESS-P ou encore DASH7.

2.2.3 Période 2019/2020 : déploiement des premiers réseaux de recherche

L'orthogonalité des trames LoRa de *spreading factor* différents n'est pas parfaite, du bruit est ajouté sur le canal par les interférents et vient entacher le rapport signal sur bruit. Le terme employé pour désigner cette propriété devient "quasi-orthogonalité" des canaux afin de préciser cette hypothèse. Les articles "Scalability Analysis of a LoRa Network Under Imperfect Orthogonality" [43] et "Impact of LoRa Imperfect Orthogonality : Analysis of Link-Level Performance" [44] traitent spécifiquement de cette imperfection. L'article "Joint Allocation Strategies of Power and Spreading Factors with Imperfect Orthogonality in LoRa Networks" [45] publié en février 2020 dans "IEEE Transactions on Communications" propose une étude théorique de l'allocation de spreading factors et puissance d'émission vis à vis de ce problème de "quasi-orthogonalité".

Cette période correspond également au lancement de plusieurs projets de réseaux et de testbeds pour la recherche. Par exemple, l'équipe DRAKKAR du Laboratoire d'Informatique de Grenoble (LIG) a développé plusieurs projets comme WiSH-WaitT [46], un framework afin de créer des testbeds LoRa reproductibles et contrôlables à distance. Un autre déploiement intéressant de réseau LoRa pour la recherche est le projet LORAFABIAN [47] porté par l'IMT Atlantique dans le centre ville de Rennes. Ce dernier a pour objectif de permettre d'évaluer les performances des réseaux LoRa et des techniques de déploiement.

2.2.4 Réseaux maillés avec couche physique LoRa

Un champ de recherche important consiste à étudier les caractéristiques de la couche physique LoRa dans le cadre de réseaux maillés. Le papier "Evaluating LoRa energy efficiency for adaptive networks : From star to mesh topologies" [48] publié en 2017 dans la conférence WiMob montre l'intérêt de basculer d'un réseau LPWAN classique en étoile vers une topologie hybride. De cette manière, un nœud très éloigné de la passerelle sera capable de choisir entre une communication directe avec un spreading factor élevé ou avec un ou plusieurs sauts via un spreading factor plus faible.

Le papier "Monitoring of Large-Area IoT Sensors Using a LoRa Wireless Mesh Network System : Design and Evaluation" [49] propose un cas d'usage d'un réseau maillé LoRa de 19 nœuds répartis sur un campus ainsi qu'une unique passerelle. Ce papier montre également le gain obtenu au niveau du taux de réception des trames via le passage des nœuds distants en multi-sauts.

Alors que les deux contributions précédents traitent des réseaux maillés via un simple relayage des trames au niveau MAC, le papier "A Routing Protocol for LoRa Mesh Networks" [50] présente une stratégie de routage pour les réseaux LoRa maillés.

2.3 Législation des bandes ISM

Les bandes ISM (instrumentation, scientifique et médical) sont des bandes de fréquence qui ont une législation particulière qui vise à garantir un accès à la communication sans besoin d'autorisation préalable.

Ces bandes de fréquence et les conditions associées sont définies par des normes différentes suivant les continents.

La figure 2.1 montre les différentes bandes qui sont définies pour le continent européen.

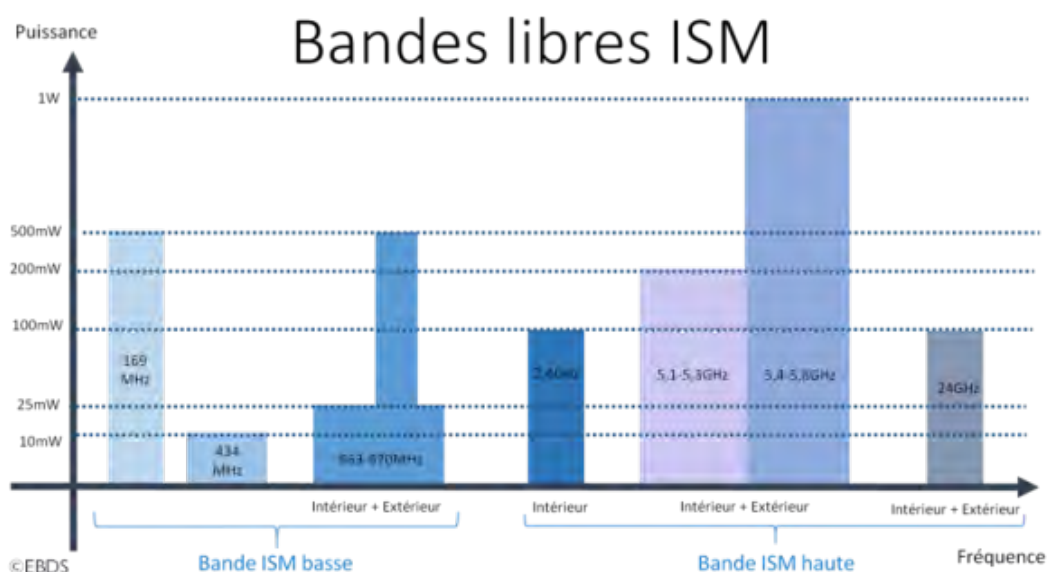


FIGURE 2.1 – Liste des bandes ISM en Europe (crédit ebds.eu)

Les concepts liés aux réseaux LPWAN ne sont pas dépendants d'une bande de fréquence particulière, cependant les technologies actuelles utilisent en majorité la bande ISM 868 MHz.

Les contraintes d'utilisation définies pour cette bande de fréquence sont les suivantes :

- Limite de la puissance d'émission de 10mW à 500mW
- Limite de la largeur de canal de 100kHz à 600kHz
- Limite de rapport cyclique de 0,1% à 100%

Les canaux fréquentiels utilisés par les technologies LPWAN comme LoRa ou Sigfox sont les canaux limités à 25mW de puissance d'émission, à 1% ou 0.1% de rapport cyclique et des canaux de 600 ou 500 kHz sous-divisibles.

La limitation du rapport cyclique est un point essentiel de notre étude car c'est la principale limitation des réseaux LPWAN aujourd'hui. Un rapport cyclique radio permet de représenter le ratio entre le temps d'émission autorisé sur un canal et le temps à attendre entre deux émissions.

Il est fréquent d'entendre parler de débit instantané en bit par seconde mais il ne faut pas oublier qu'il est autorisé d'émettre que 1% ou 0.1% du temps à ce débit là.

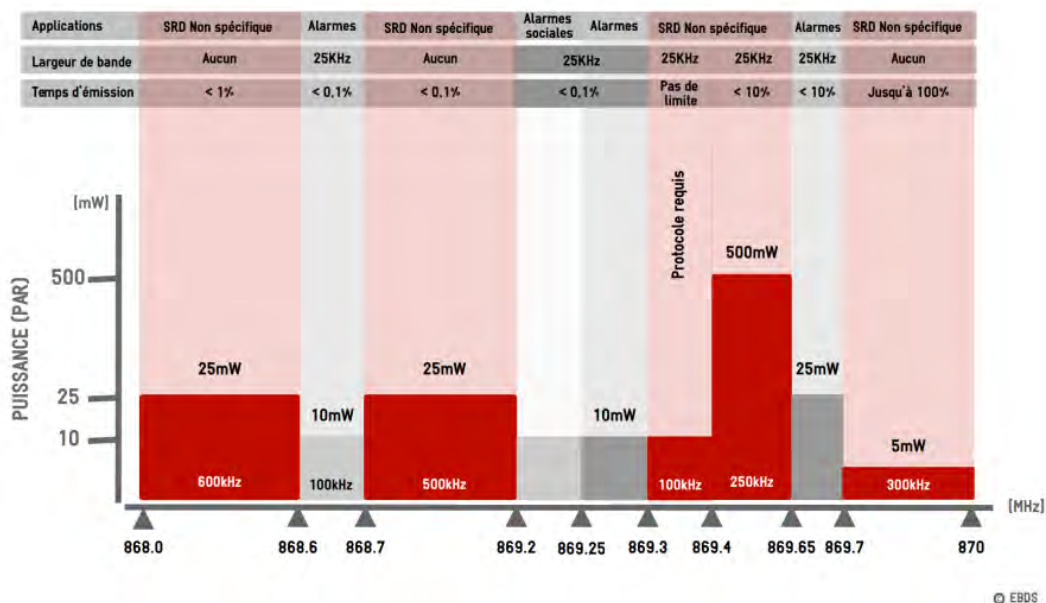


FIGURE 2.2 – Détails de la bande ISM 868 MHz (crédit ebds.eu)

Il faut considérer les réseaux LPWAN actuels, plutôt comme des réseaux de messagerie et non comme des réseaux avec une bande passante constante dans le temps.

2.4 Évaluation des paramètres de la couche physique

Notre équipe au laboratoire est spécialisée dans l'ingénierie protocolaire des couches MAC. Lors de la phase d'exploration d'une nouvelle couche physique, nous avons une approche de caractérisation des leviers disponibles en vue de concevoir une couche MAC optimisée. Historiquement, l'équipe du laboratoire a travaillé principalement sur les empilements protocolaires pour les réseaux maillés avec par exemple des contributions sur Bluetooth, Zigbee/IEEE802.15.4 ou encore une couche MAC dédiée au réseau OCARI.

2.4.1 Idée de départ : du multi-saut au multi-médium

Nous avons été très vite intéressés par la couche physique LoRa avec une approche originale dans la communauté scientifique. Les nombreuses recherches qui ont eu lieu sur les réseaux maillés sont unanimes sur les difficultés de passage à l'échelle des algorithmes multi-saut. Les promesses annoncées par la couche physique LoRa permettent d'envisager par exemple que tous les noeuds d'une maison standard partagent le même médium. De cette façon, les enjeux algorithmiques au niveau de la couche de routage dans le contexte multi-saut se transforment en enjeux algorithmiques au niveau de la couche MAC dans le contexte multi-médium.

2.4.2 Modulation LoRa

La modulation LoRa est une modulation propriétaire et il y a très peu d'information sur les mécanismes précis de cette modulation. Cependant il y a tout de même assez de documentation pour comprendre le principe de cette modulation et des concepts associés.

La modulation LoRa est de type étalement de spectre, et plus précisément CSS (Chirp Spread Spectrum). La philosophie de l'étalement de spectre est de répartir le codage de l'information sur une large bande passante par rapport à un interférent. L'ajout de beaucoup de redondance dans le codage du message permet de limiter l'impact de cet interférent sur une sous bande fréquentielle et temporelle.

En observant l'émission d'une trame LoRa à l'aide d'un analyseur de spectre de type SDR, il est possible d'observer la modulation comme le montre la figure 2.3

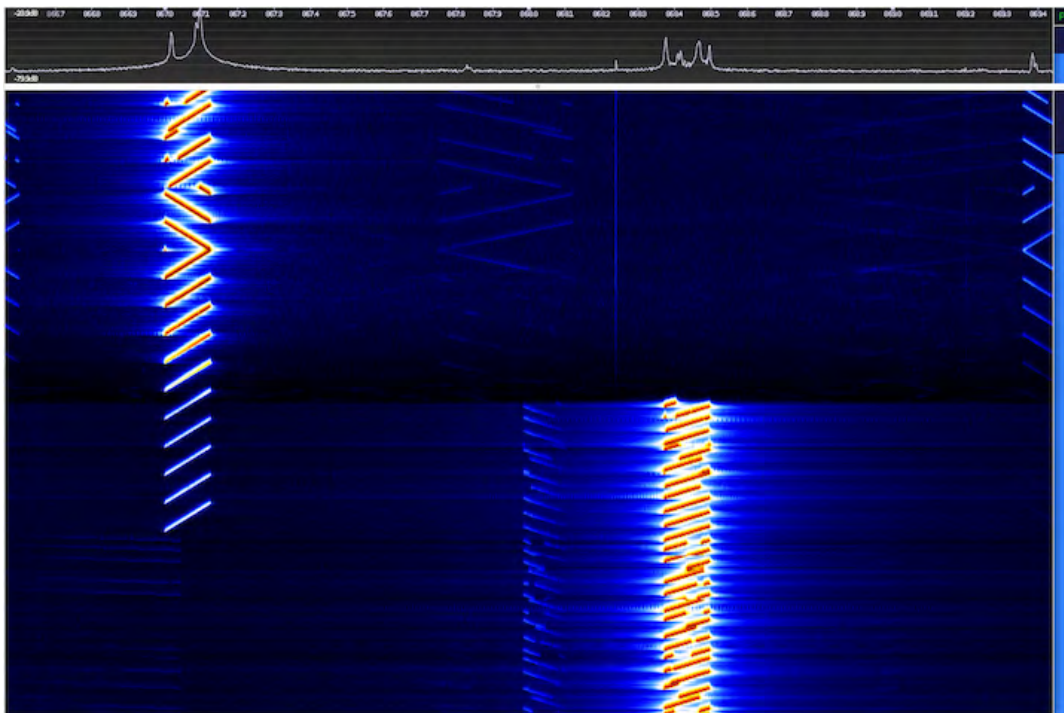


FIGURE 2.3 – Visualisation de la modulation LoRa observée à l'analyseur de spectre (SDR waterfall)

Cette trace est typique de la modulation CSS car on voit clairement que la porteuse évolue à vitesse angulaire constante sur une bande passante fixée avec des changements de phase réguliers. Le fait de garder une vitesse angulaire constante permet de limiter l'impact des phénomènes de multi-trajets ce qui est très important en milieu urbain par exemple.

Nous observons que les choix qui ont été fait au niveau de la modulation possèdent des caractéristiques intéressantes mais le codage des symboles va devoir être important pour être vraiment efficace.

2.4.3 Codage de canal dans la couche physique LoRa

Au niveau de l'étage de codage des trames LoRa, il y a plusieurs fonctions de codage qui servent à ajouter de la redondance, de la détection d'erreurs et des séquences de symboles importantes pour augmenter la différenciation des symboles '0' et '1'.

2.4.3.1 Spreading Factor

Le spreading factor ou facteur d'étalement en français est le nombre de sous-symboles (chirp) utilisés pour représenter un symbole '0' ou '1'. Ces sous-séquences sont des multiples de 2, ainsi le spreading factor est représenté de la façon suivante :

$$\text{Nombre de chirps par symbole} = 2^{SF}$$

Sur les trancieveurs LoRa, le spreading factor peut varier de 6 à 12 c'est à dire de 64 à 4096 chirps par symbole. Lorsque l'on augmente le spreading factor de 1, il faut deux fois plus de chirps pour coder un symbole, donc le temps d'émission est deux fois plus long pour une taille de charge utile constante. Mais comme les séquences de chirps sont plus longues, il est plus difficile de confondre la sous-séquence du symbole '0' et celle de '1' car la distance entre eux deux est plus grande. Ce gain de codage s'exprime en décibel et va être intéressant lors du bilan de liaison.

Les sous-séquences entre les spreading factors n'ont pas été choisies au hasard, elle ont une très faible intercorrélacion ce qui permet d'avoir un fort taux de réjection entre deux trames émises simultanément sur le même canal.

2.4.3.2 Codage de canal

Différents traitements sont réalisés au niveau codage de canal afin d'améliorer sa robustesse : whitening, codage de Hamming, interleaving, codage de Gray et un CRC sur l'ensemble de la trame [51]. Ces différentes méthodes permettent de limiter certaines faiblesses de trames avec peu d'entropie et de chercher à détecter et corriger les erreurs lors de la transmission du signal. Nous décrirons ici uniquement le CRC et le codage de Hamming qui sont réglables au niveau des composants radio.

Code correcteur cyclique La modulation LoRa utilise un codage cyclique pour détecter et corriger les erreurs, c'est le classique "coding rate". Le code cyclique utilisé est un code de Hamming avec les paramétrages suivants : 5,4, 6,4, 7,4 et 8,4. Pour 4 bits de données transmises, le tranciever ajoute entre 1 et 4 bits de redondance afin de pouvoir détecter et/ou corriger des erreurs. Le paramétrage standard du code de Hamming est le 7,4 qui permet de corriger au plus un bit altéré. La version 5,4 est simplement une contrôle de parité qui permet de détecter au plus un bit altéré.

Contrôle de redondance cyclique Le tranciever peut optionnellement ajouter et vérifier à la fin de la trame un CRC sur 16 bits de type CCITT-16 2.4 [52]. Ce code de contrôle ajoute une sécurité supplémentaire pour la détection d'erreurs au sein de la trame. Cette sécurité est utile au niveau du récepteur pour décider de valider la trame ou non.

2.4.4 Temps d'émission et débit

Le temps d'émission d'une trame sur le médium en fonction des différents paramètres est très important pour estimer un débit utile et des profils applicatifs de la technologie. Ce temps est quasi proportionnel au nombre d'octets utiles mais attention le coefficient multiplicateur varie énormément d'un paramétrage à l'autre.

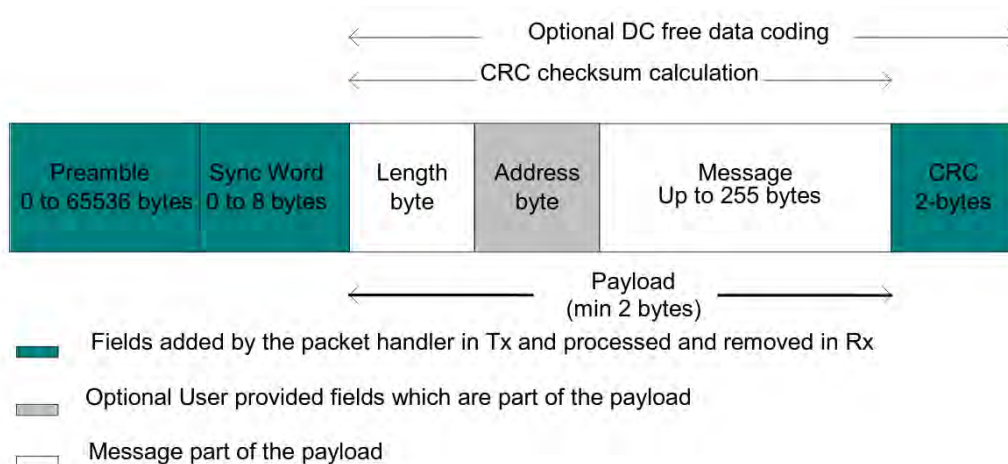


FIGURE 2.4 – Format de trame LoRa à longueur variable (crédit Semtech)

2.4.4.1 Temps d'émission sur le canal

Les calculs du temps d'émission sur le médium d'une trame, sont détaillés dans la documentation technique du composant SX1276 [53]. Nous proposons une implémentation de cette formule en Python de façon à pouvoir l'utiliser dans nos simulations :

```

1 def timeOnAir(self):
2     symbolRate = self.bandwidth / math.pow(2, self.spreadingFactor)
3     tSymbol = 1 / symbolRate
4
5     if self.codingRate == 1.25: CR = 1
6     elif self.codingRate == 1.5: CR = 2
7     elif self.codingRate == 1.75: CR = 3
8     elif self.codingRate == 2: CR = 4
9
10    nPayload = 8 + max(math.ceil(float((8*self.payloadLength
11    - 4*self.spreadingFactor + 28 + 16
12    - 20*self.explicitMode))/float((4*(self.spreadingFactor
13    - 2*self.lowDataRateOptimization)))) * (CR + 4), 0)
14
15    tPreamble = (self.preambleLength + 4.25) * tSymbol
16    tPayload = nPayload * tSymbol
17    tPacket = tPreamble + tPayload
18    return tPacket

```

FIGURE 2.5 – Fonction python qui calcule le temps d'émission sur le canal

Afin d'expliciter les différents ordres de grandeur mis en jeu, nous avons réalisé un tableau des différents temps d'émission selon le paramétrage.

	20 octets	40 octets	60 octets	80 octets	100 octets
Spreading Factor 7	0.0566	0.0822	0.1129	0.1436	0.1743
Spreading Factor 8	0.1029	0.1541	0.2053	0.2565	0.3077
Spreading Factor 9	0.1853	0.2877	0.3697	0.4516	0.5540
Spreading Factor 10	0.3707	0.5345	0.6984	0.8622	1.0260
Spreading Factor 11	0.6595	0.9871	1.2329	1.5606	1.8883
Spreading Factor 12	1.3189	1.8104	2.3020	2.9573	3.4488

FIGURE 2.6 – Temps de transmission des trames LoRa en seconde

2.4.4.2 Notion de débit dans un réseau LoRa

La notion de débit dans un réseau LoRa doit être soulignée et interprétée avec précautions. Une trame est transmise avec un débit de symboles fixé, chaque symbole reste présent sur le médium pendant un temps que l'on appelle "temps symbole". La première remarque consiste à faire la différence entre un symbole et un bit utile, en effet suivant le paramétrage un bit peut être codé sur plusieurs symboles. Ce débit de symboles peut être décrit comme le débit de transmission instantané en symbole par seconde.

La législation des bandes ISM impose un rapport cyclique à l'émission qui ne permet pas de transmettre en continu avec le débit instantané. Il est alors requis d'alterner entre une émission et un temps d'attente proportionnel au temps d'émission pour respecter le rapport cyclique. La notion de débit doit être exprimée de ce fait au niveau d'un cycle liée au temps d'émission. Nous définissons ce débit comme débit applicatif global.

Les tableaux 2.7 et 2.8 représentent cette périodicité d'émission pour les différentes trames du tableau 2.4 pour respectivement un canal à 1% et un canal à 0,1%.

	20 octets	40 octets	60 octets	80 octets	100 octets
Spreading Factor 7	5.6576	8.2176	11.2896	14.3616	17.4336
Spreading Factor 8	10.2912	15.4112	20.5312	25.6512	30.7712
Spreading Factor 9	18.5344	28.7744	36.9664	45.1584	55.3984
Spreading Factor 10	37.0688	53.4528	69.8368	86.2208	102.6048
Spreading Factor 11	65.9456	98.7136	123.2896	156.0576	188.8256
Spreading Factor 12	131.8912	181.0432	230.1952	295.7312	344.8832

FIGURE 2.7 – Longueur du rapport cyclique en seconde des trames LoRa sur un canal à 1%

	20 octets	40 octets	60 octets	80 octets	100 octets
Spreading Factor 7	56.5760	82.1760	112.8960	143.6160	174.3360
Spreading Factor 8	102.9120	154.1120	205.3120	256.5120	307.7120
Spreading Factor 9	185.3440	287.7440	369.6640	451.5840	553.9840
Spreading Factor 10	370.6880	534.5280	698.3680	862.2080	1026.0480
Spreading Factor 11	659.4560	987.1360	1232.8960	1560.5760	1888.2560
Spreading Factor 12	1318.9120	1810.4320	2301.9520	2957.3120	3448.8320

FIGURE 2.8 – Longueur du rapport cyclique en seconde des trames LoRa sur un canal à 0,1%

Analysons le cas le plus favorable et le cas le plus défavorable. Avec un spreading factor de 7 et une trame de 20 octets, il est possible d'émettre ce message toutes les 5,6 secondes sur un canal à 1% et toutes les 56 secondes sur un canal à 0,1%. Avec un spreading factor de 12 et une trame de 100 octets, il est possible d'émettre ce message toutes les 5,73 minutes sur un canal à 1% et toutes les 57,3 minutes sur un canal à 0,1%.

Les réseaux LPWAN qui transmettent sur les bandes ISM doivent être considérés comme des réseaux de messagerie. Leur débit doit être exprimé en nombre de messages par heure ou par jour suivant un paramétrage donné. Une application qui doit pouvoir émettre deux messages quasi simultanément doit considérer cette problématique avec une grande attention.

2.5 Composants d'infrastructure

Une fois la couche physique LoRa et ses paramètres étudiés, nous allons lister les différents composants d'infrastructure logiciels et matériels nécessaires au déploiement d'un réseau.

2.5.1 Passerelle LPWAN

Dans les réseaux sans fil, il est commun de vouloir faire la jonction entre un réseau filaire et ces réseaux locaux sans fil. Ce composant est appelé point d'accès ou passerelle (gateway en anglais). Au niveau conceptuel, ce composant est capable de communiquer via deux technologies physiques différentes et est capable de transformer des trames d'un réseau vers un autre. Il ne modifie pas les informations mais translate les données d'un format à un autre pour gérer l'interopérabilité entre les deux réseaux.

2.5.1.1 Particularités des passerelles LPWAN

Il est possible d'imaginer qu'une passerelle LPWAN puisse fonctionner comme un point d'accès Wi-Fi en cherchant à transformer directement une trame radio en trame IP. La trame radio devrait pour cela contenir les informations nécessaires à la construction de ce paquet à moins que la passerelle soit capable de stocker une table de données pour réaliser cette translation d'adresse.

Cependant dans les réseaux LPWAN communément déployés par les opérateurs, les trames reçues via le réseau sans fil LPWAN ne vont pas être transformées directement en paquets IP qui pourraient atteindre directement leur cible. Les trames sont encapsulées vers un composant central qui va se charger de collecter et de traiter tous les messages : le serveur de réseau.

2.5.1.2 Protocole de transmission

Avant de décrire un serveur de réseau, nous allons nous arrêter un instant sur le protocole de transmission utilisé pour le dialogue entre ces deux composants.

Comme nous l'avons expliqué, la translation des trames ne se fait pas au niveau de la passerelle mais est déportée dans un composant central. Afin d'acheminer les trames vers ce composant, il faut utiliser un protocole de transport qui va encapsuler les trames et les livrer au serveur de réseau. Le serveur de réseau et les passerelles communiquent via un cœur de réseau IP, donc le protocole utilisé peut être choisi parmi les protocoles de transport existant, en y ajoutant une couche protocolaire pour formater les trames et y ajouter des méta-données.

Au niveau des réseaux LoRa, Semtech l'entreprise propriétaire de la modulation et des composants physiques, a proposé un protocole qui utilise UDP qui s'appelle le "Packet Forwarder". Nous décrivons ce protocole dans une section dédiée.

2.5.1.3 Zone de couverture

Lorsqu'une zone géographique doit être couverte par un réseau LPWAN, il devient nécessaire de déployer plusieurs passerelles afin d'obtenir une couverture contiguë. Ces passerelles sont affiliées aux stations de base dans les technologies cellulaires.

Chose intéressante, les réseaux LPWAN fonctionnent dans un mode très faiblement connecté. Dans un réseau cellulaire, les mobiles sont attachés à une passerelle et lors d'un déplacement, vont s'accrocher aux passerelles voisines (handover). Les nœuds mobiles de LPWAN ne sont eux pas rattachés à une passerelle et sont même très faiblement rattachés au serveur de réseau. Ainsi lorsqu'une trame va être émise par un mobile, l'ensemble des passerelles à portée radio vont recevoir la trame, l'encapsuler et la transmettre au serveur de réseau.

La stratégie optimale dans les réseaux cellulaires était de former des cellules avec le moins de recouvrement possible et de colorier le réseau en allouant des fréquences différentes à deux stations de base voisines pour limiter les interférences. Dans un réseau LPWAN la stratégie est inverse, on va chercher à avoir un maximum de recouvrement car cela va augmenter les chances de succès de réception d'une trame.

2.5.1.4 Spécificités Technologiques

Les passerelles pour les LPWAN et plus particulièrement les passerelles LoRa, sont plus évoluées que les modems radio des objets car elles sont capables de pouvoir communiquer simultanément avec un grand nombre d'objets sur des canaux différents. En effet, l'autonomie énergétique n'est plus un problème car les passerelles sont alimentées en permanence. Le composant radio des passerelles doit permettre de recevoir un maximum de messages simultanément sur différents canaux fréquentiels et à différents *spreading factors*.

2.5.2 Serveur de réseau

Un serveur de réseau pour les LPWAN est un service informatique qui sert de planificateur des transmissions au niveau des passerelles. Il peut être seul et agir en tant que chef d'orchestre dans une architecture centralisée où il est possible d'avoir plusieurs serveurs de réseau qui agissent de façon concurrente ou collaborative.

Ce service est à l'interface entre les passerelles et les applications qui désirent communiquer sur le réseau. C'est lui qui en fonction des messages qui sont reçus ou qui désirent être envoyés, effectue la planification des échanges réseaux.

Le serveur de réseau est responsable de maintenir une table avec les nœuds autorisés à communiquer sur le réseau, d'effectuer les échanges protocolaires pour l'association et la dé-association.

Pour communiquer avec les nœuds, le serveur de réseau fait des requêtes aux passerelles afin d'émettre les messages dans la zone de portée radio des nœuds. Afin de faire cela, le serveur de réseau maintient dans sa table une colonne avec la passerelle la plus pertinente pour émettre à destination de chaque nœud. La détermination de cette passerelle peut se faire par la mémorisation de la passerelle qui a reçu le dernier message dans les meilleures conditions. Le problème est que si le nœud est très mobile et que les conditions ont fortement changé depuis la dernière émission, que la passerelle mémorisée ne soit plus pertinente. Afin

d'éviter ce problème il est nécessaire d'attendre que le noeud émette pour lui répondre, d'avoir un réseau synchronisé ou que le noeud se déclare immobile.

2.5.3 Serveurs applicatifs

Les composants d'infrastructure précédents fournissent uniquement un service de collecte et de transport de la donnée, nous pouvons faire une analogie avec le service postal. Dans une infrastructure LPWAN, les serveurs applicatifs sont les destinataires des messages qui proviennent des objets communicants.

Les serveurs applicatifs sont à l'interface entre les objets et les utilisateurs. Des applications web fournissent des API, des interfaces homme-machine, ou encore des points d'accès pour les clients mobiles.

2.5.3.1 Stockage

L'ensemble des composants réseaux LPWAN n'effectuent pas de stockage, c'est uniquement au niveau du serveur applicatif que les données commencent à être stockées. Il y a deux types de données à stocker : les données temporelles et les méta-données. Les méta-données sont importantes afin de donner leur contexte aux données temporelles.

Exemple : Une application de suivi de véhicules réfrigérés relève en temps-réel les différentes informations de température et d'hygrométrie de la chambre froide. Sans définir de méta-données, notre application recevrait un flux de températures sans pouvoir la mettre en lien avec les véhicules, les marchandises ou encore la localisation géographique.

Voici un format de donnée utilisable en exemple :

```
1 {
2   "time": "2020-05-20T13:44:11.159Z",
3   "tags": {
4     "vehicule-id": "veh-fr-015315",
5     "localization": {
6       "latitude": 1,45646,
7       "longitude": 44,54654
8     },
9     "path-id": "3186486346"
10  },
11  "values": {
12    "temperature": 21.5,
13    "humidity": 62.4
14  }
15 }
```

FIGURE 2.9 – Exemple de format de donnée utilisable

Dans cet exemple, nous voyons que les données sont rangées en deux catégories, les méta-données (tags) et les données temporelles (values). Ce type de données est communément appelé "séries temporelles" car ces valeurs n'ont de sens qu'à un instant du temps donné et dans un environnement donné. D'où la nécessité de bien ajouter ce contexte lors du stockage.

Les séries temporelles sont étudiées depuis les débuts de l'informatique mais la percée du *cloud* a fait émerger de nombreuses bases de données de séries temporelles. En effet, le suivi en temps-réel de nombreux serveurs et de nombreuses applications nécessite de collecter et de stocker ces données de monitoring et de supervision. Pour ne citer que quelques acteurs du marché : InfluxDB, Prometheus, MongoDB et

TimescaleDB.

2.5.3.2 Traitement des données

La plateforme applicative est également responsable du traitement des données collectées. Ce traitement s'effectue de deux façons différentes : traitements en temps réel des flux de données et traitement par lots pour les données stockées. Cette architecture typique du Big Data s'appelle une architecture Lambda comme présenté sur la figure 2.10.

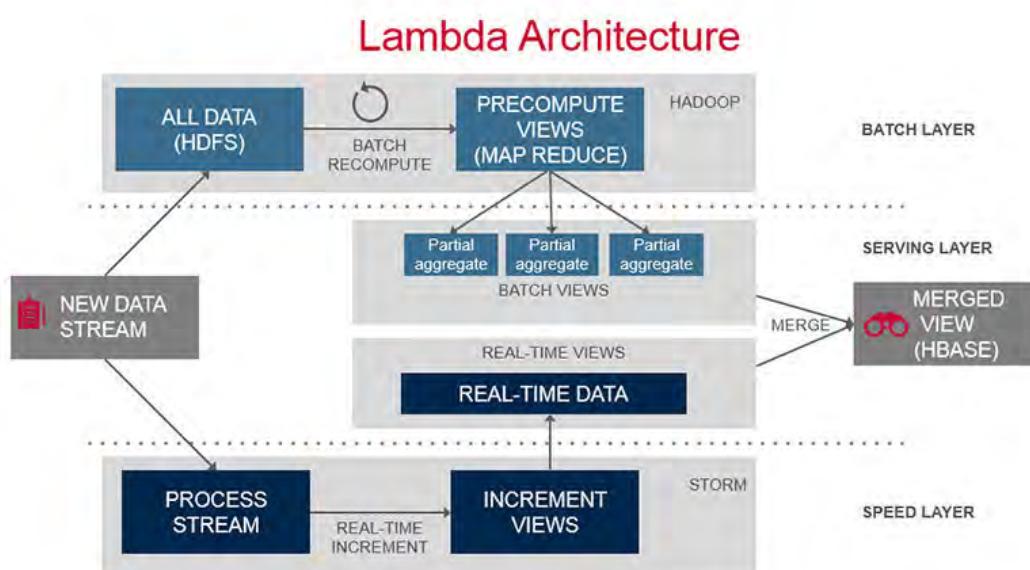


FIGURE 2.10 – Architecture lambda pour le traitement des données

Le traitement de ces données permet de générer des alertes, de faire de la prédiction ou de calculer des tendances et des statistiques.

2.6 Réseaux LPWAN privés et publics

Les architectures actuelles des réseaux LPWAN sont constituées des éléments que nous avons présentés.

Chaque réseau dispose de ses passerelles qui sont dispersées dans l'environnement afin de garantir une zone de couverture adéquate et sont reliées ensemble via un coeur de réseau IP vers un serveur de réseau centralisé.

L'originalité des réseaux LoRa est de proposer un déploiement par un opérateur (public) ou un déploiement en propre par une organisation (privée).

2.6.1 Réseaux LPWAN publics

Le modèle économique et technique des réseaux LPWAN s'apparente fortement aux réseaux cellulaires existants. Les opérateurs historiques ont de ce fait rapidement développés une offre dédiée pour les réseaux LoRaWAN. La promesse applicative de réseaux omniprésents sur un territoire comme la France, l'Europe ou le monde nécessite le déploiement de nombreuses passerelles réparties sur le territoire. Ce déploiement est complexe lorsque l'opérateur du réseau ne possède pas de points de présence sur le territoire. C'est comme cela que les opérateurs historiques se sont ainsi emparés de ce marché, en réutilisant les tours télécoms utilisées pour les réseaux cellulaires car l'installation d'une passerelle LoRa est assez légère pour ne pas nécessiter de refaire des études de sol.

Les réseaux LoRa massivement déployés par les opérateurs se basent sur les spécifications de LoRaWAN qui décrivent un protocole d'encodage des trames qui transitent via la modulation LoRa et également le protocole de transport de ces messages sur le backbone IP à destination du serveur de réseau.

Dans les spécifications LoRaWAN, la distinction entre le service réseau et le service applicatif est faite via un chiffrement de la couche applicative de façon à ce que l'opérateur de réseau n'ait pas accès aux données. Cependant pour des soucis de simplicité d'utilisation, les opérateurs proposent de base un déchiffrement des données afin de fournir un premier niveau de services de gestion d'alertes, de stockage et de visualisation.

2.6.2 Réseaux LPWAN privés

Le faible coût des matériels et la disposition de logiciels libres qui implémentent les différents agents de l'infrastructure permettent à une organisation de pouvoir déployer son propre réseau. Bien entendu, ces réseaux privés sont adaptés à une zone géographique limitée qui est généralement sous la responsabilité de l'organisation.

Lorsqu'une application nécessite de la mobilité à l'échelle du territoire, il est nécessaire de migrer sur des réseaux publics. Il faut noter que les spécifications LoRaWAN permettent de provisionner plusieurs identifiants chez plusieurs opérateurs afin de gérer un basculement entre opérateurs ou un basculement d'un réseau privé vers un réseau public.

2.6.2.1 Réseaux privés fixes

Les réseaux privés fixes sont déployés par des grands groupes qui bénéficient de nombreux bâtiments répartis sur un complexe industriel permettant d'envisager le déploiement en propre d'un réseau. C'est le cas par exemple d'Airbus à Toulouse qui avec l'aide d'un opérateur public, a déployé un réseau privé.¹

Un réseau privé fixe peut être pensé au niveau métropolitain pour fournir un service aux citoyens qui soit maîtrisé par la collectivité. Cette démarche peut s'inscrire dans une politique "smart city" ou "ville intelligente" avec un ensemble de services de collecte et de traitement des données pour optimiser les services publics.

2.6.2.2 Réseaux privés temporaires

Un autre cas d'usage de ces réseaux privés est à destination d'entreprises qui ont une zone à couvrir de façon temporaire. C'est le cas des entreprises de travaux publics qui peuvent déployer un réseau LoRa

1. <https://www.usine-digitale.fr/article/airbus-utilise-le-reseau-dedie-d-objenious-pour-exploiter-ses-objets-connectes.N808510>

de façon à couvrir un chantier et collecter en temps réel les données issues des engins de chantier.

Un réseau privé temporaire peut permettre de couvrir une zone d'intervention dans le cadre d'opérations de sauvetage pour des pompiers ou d'assauts pour des forces spéciales.

2.7 Protocoles de transport des messages

L'architecture d'un réseau LPWAN est distribuée entre des passerelles sur le terrain et un ou plusieurs serveurs de réseau côté cœur de réseau. L'ensemble de ces agents doivent communiquer via TCP/IP à travers plusieurs réseaux comme la collecte en 4G, la sécurisation via des VPN ou le transport au travers d'Internet. Afin de réaliser cette fonction de transport des messages entre les passerelles LPWAN et le cœur de réseau, il est nécessaire d'utiliser un protocole spécifique.

2.7.1 Packet Forwarder

Dans le cadre des réseaux LoRaWAN, Semtech a défini un protocole de transport qui se nomme « Packet Forwarder » [54]. Ce protocole est implémenté dans le driver open source qui permet de gérer un sx1301 via un système d'exploitation Linux.

Comme il est écrit dans la spécification du protocole, celui-ci est très basique et n'intègre aucune fonction de sécurité. Il est recommandé de ne l'utiliser que sur un réseau privé et fiable. Cette problématique vient du fait qu'il est implémenté sur UDP et qu'il ne spécifie pas de couche de chiffrement à utiliser. Sur la figure 2.11, le protocole "Packet Forwarder" est utilisé sur le lien de "backhaul".

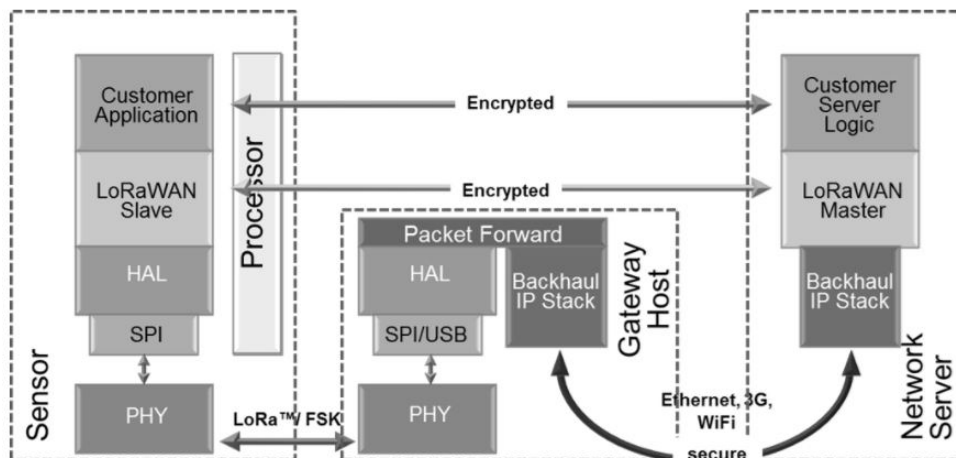


FIGURE 2.11 – Architecture LoRaWAN via le Packet Forwarder telle que définie par Semtech (crédit Semtech)

2.7.1.1 Réception d'une trame LoRa

La figure 2.12 montre le diagramme de séquence pour un relaiage d'une trame reçue par la radio sx1301 et transmise au cœur de réseau via la commande PUSH_DATA. Cette figure est issue de la spécification du Packet Forwarder.



FIGURE 2.12 – Diagramme de séquence d’un relayage de trame

Lorsqu’un message est reçu par la radio, la passerelle encode la trame en base64 et l’encapsule dans un JSON avec toutes les informations sur le contexte de réception :

- Timestamp UTC de réception
- Timestamp GPS de réception en milliseconde depuis le 06 janvier 1980
- Timestamp interne du sx1301
- Fréquence centrale de réception
- Référence hardware du décodeur et de la sous-porteuse
- Résultat de la vérification du CRC
- Configuration du datarate LoRa
- RSSI (force du signal reçu)
- Rapport signal/bruit
- Taille de la trame
- Trame encodée en base64

Le header d’une commande PUSH_DATA est constitué de :

- Numéro de version du protocole
- Token aléatoire sur 16 bits
- Type de commande (ici PUSH_DATA)
- Identifiant unique de la gateway (adresse MAC)

- Charge utile du paquet (le JSON défini ci-dessus)

La procédure de transmission du paquet est très basique avec l'identification par un nombre aléatoire sur 16 bits et l'acquittement de la réception du paquet par le serveur via un paquet PUSH_ACK qui contient le même jeton aléatoire.

2.7.1.2 Émission d'une trame LoRa

Lorsqu'un serveur de réseau décide d'émettre une trame à travers une passerelle, il effectue le même type de transaction que précédemment. Il utilise cette fois une commande de type PULL_RESP pour transmettre la trame dans le sens descendant. La figure 2.13 issue de la spécification illustre le diagramme de séquence de l'émission d'une trame par une passerelle.

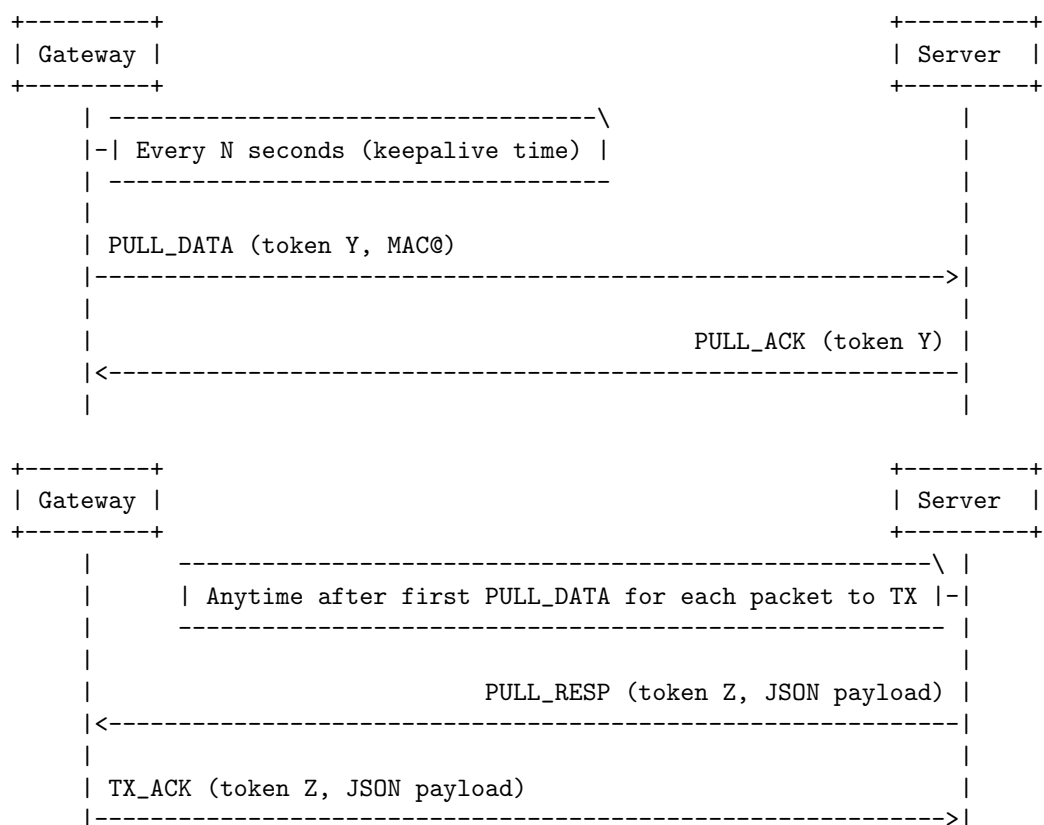


FIGURE 2.13 – Diagramme de séquence d'une transmission de trame

Le diagramme de séquence montre que c'est toujours le logiciel de la passerelle qui initie les échanges de part sa position de client UDP. Dans le cas d'un paquet descendant, c'est la passerelle qui va interroger périodiquement le serveur via une commande PULL_DATA afin de garder le chemin ouvert lorsque les trames UDP passent à travers un NAT par exemple. Une fois que ce chemin est ouvert, le serveur de réseau peut quand il le souhaite répondre via une commande PULL_RESP lui permettant de transmettre une trame LoRa à émettre via la passerelle.

Les paramètres sont très similaires à ceux de réception d'une trame, cependant la gestion du temps est très différente. Un objet LoRaWAN émet une trame et se rendort, puis se réveille pendant deux fenêtres afin de se mettre en attente de réception d'une réponse de la part du serveur de réseau. Le temps entre la fin de l'émission de la trame et la première fenêtre de réception, permet l'échange protocolaire entre la passerelle et le serveur de réseau. Le serveur de réseau peut lors de l'émission d'un message descendant, demander à la passerelle d'émettre le message dès la réception du datagramme ou à un instant donné dans le futur. Dans le cas d'une émission décalée, la passerelle va stocker la trame en mémoire en attendant le moment d'émission. Ce mécanisme permet d'être précis temporellement, tout en permettant un échange avec le serveur de réseau au travers de différents réseaux avec des latences très diverses.

2.7.2 Transport des trames via MQTT

Durant cette thèse, nous avons très tôt utilisé le protocole MQTT afin de transmettre les trames radio entre différents agents. Ce protocole nous a permis via sa faible empreinte protocolaire et la possibilité de duplication des messages entre différents agents, de gérer le service de transport de façon élégante et extensible.

En même temps que nos expérimentations, les développeurs de Chirpstack (loraserver à l'époque) ont rendu disponible le logiciel « Chirpstack Gateway Bridge » [55] permettant de faire la translation entre le protocole packet forwarder et MQTT.

L'objectif de la translation est de garder le packet forwarder en interne au sein de la gateway pour ne pas avoir à modifier le driver qui gère le sx1301 qui utilise uniquement le packet forwarder pour communiquer les trames. De cette façon, le protocole sur UDP reste en local et c'est MQTT qui est utilisé pour transmettre les trames à destination du serveur.

L'avantage de ce nouveau paradigme est de casser le modèle client/serveur qui obligeait à avoir un serveur de réseau. Maintenant le problème est séparé en deux parties, la collecte des trames via MQTT et la valorisation du contenu des trames par un service réseau LoRaWAN qui gère l'analyse protocolaire.

2.8 De la théorie à la pratique

Dans ce second chapitre, nous avons présenté les points qui font l'originalité des réseaux LPWAN. Nous avons plus particulièrement identifié l'état de l'art scientifique et technologique des réseaux LoRa qui vont nous servir à la mise en pratique des concepts théoriques.

Les réseaux LPWAN ont la particularité d'émettre sur des longues distances. Afin d'étudier ces réseaux dans un environnement réel, les murs du laboratoire ne suffisent plus et il devient nécessaire d'imaginer plus grand... Un réseau d'expérimentation de la taille d'un campus? D'un quartier? D'une ville? Il était nécessaire de doter Toulouse d'un réseau d'expérimentation métropolitain afin d'étudier, de former, d'intégrer et de promouvoir ces nouvelles technologies : c'est l'objectif du chapitre suivant.

Chapitre 3

Proposition d'une architecture de *testbed* urbain ouvert pour les LPWAN

Une contribution majeure de la thèse a été d'imaginer et de gérer le déploiement d'un réseau LPWAN expérimental sur la ville de Toulouse. Cette contribution a été présentée via différentes présentations et publications. Vous pouvez retrouver ces contributions dans la liste des publications. Le déploiement d'un réseau métropolitain ouvert permet de développer de nouveaux protocoles et de réaliser l'analyse de leurs performances dans un environnement réel. Nous avons mené plusieurs travaux d'expérimentation à l'aide du testbed urbain pendant et après son déploiement. De cette façon, nous montrons différentes voies de recherche en protocoles et réseaux grâce à cet outil.

Sommaire

3.1	Origines du projet	76
3.1.1	<i>Testbed</i> urbain	76
3.1.2	L'association Tetaneutral.net	76
3.1.3	Lancement du projet	76
3.2	Proposition d'architecture d'un <i>testbed</i> urbain ouvert	77
3.2.1	Passerelle LoRa	77
3.2.2	Réseau de collecte des trames LoRa	79
3.2.3	Concept de commutation au niveau du bus logiciel	81
3.3	Programmation de nouveaux réseaux LoRa	82
3.3.1	Programmation de nouveaux services réseau pour les LPWAN	82
3.3.2	Programmation des interactions avec les objets	84
3.4	Architecture globale du <i>testbed</i> urbain	86
3.4.1	Services réseau du <i>testbed</i> urbain	86
3.5	Cas d'usages et valorisation scientifique du <i>testbed</i>	87
3.5.1	Cas d'usages	89
3.5.2	Valorisation	89

3.1 Origines du projet

3.1.1 *Testbed* urbain

Notre équipe de recherche RMESS du laboratoire IRIT (auparavant nommée IRT, SCSF du LATTIS, ICARE...) est spécialisée dans les infrastructures matérielles et logicielles à but expérimental, aussi appelées *testbed*. Depuis plusieurs années, notre équipe développe des objets communicants ainsi que les infrastructures associées permettant de concevoir et de tester des protocoles pour les réseaux sans fil. Cette approche de la recherche consiste à évaluer les réseaux sans fil dans un environnement réel, c'est à dire avec les problématiques de couverture, de liens asymétriques, de limites physiques non prises en compte dans les simulateurs etc.

L'équipe a une expertise dans les réseaux maillés faible portée comme les réseaux à base de la couche MAC 802.15.4. Ces réseaux à faible portée sont plus facilement déployables dans le bâtiment d'un laboratoire de part leur nature courte portée, ce qui n'est pas le cas des réseaux émergents LPWAN. Nous avons dans un premier temps commencé à étudier la couche physique LoRa à l'échelle du laboratoire mais nous avons vite compris que les enjeux étaient à une échelle plus importante de l'ordre du kilomètre.

À partir de ce constat, nous avons dû imaginer une infrastructure de *testbed* qui dépasse les murs du laboratoire. Ce *testbed* devait être plus représentatif du contexte LPWAN, c'est à dire avec des noeuds de type passerelle, avec des distances de plusieurs kilomètres et avec un environnement urbain prédominant. De là il deviendrait possible d'étudier la propagation de la couche physique LoRa, de comprendre son influence et les enjeux sur un service MAC associé et finalement de pouvoir implémenter et tester des nouveaux protocoles pour les LPWAN. Les services réseaux associés sont également pris en compte dans les plateformes de l'équipe (remontée de capteurs, surveillance de personnes, localisation...).

3.1.2 L'association Tetaneutral.net

L'association Tetaneutral.net est un opérateur associatif toulousain qui héberge des infrastructures numériques et est fournisseur d'accès à Internet via un réseau de ponts hertziens 802.11ac entre les adhérents. Ces ponts hertziens sont situés sur les toits de multiples immeubles permettant d'avoir dans le cadre de ce projet des points hauts répartis dans la ville. Tetaneutral.net est une association qui a un lien historique assez privilégié avec le laboratoire car plusieurs membres du laboratoire sont membres depuis sa création.

Tetaneutral.net est une association qui milite pour la neutralité d'Internet et l'éducation populaire. Notre projet de *testbed* urbain s'inscrit également dans cette démarche de fournir une infrastructure de test aux citoyens, pour la pédagogie et la recherche, de manière à démystifier les réseaux LPWAN et fournir une plateforme ouverte pour tous.

3.1.3 Lancement du projet

Nous avons décidé dans le cadre de cette thèse, de créer ce projet de *testbed* urbain libre pour la pédagogie et la recherche avec l'association Tetaneutral.net via le partage d'intérêts communs. Notre contribution a été d'imaginer, de déployer et de faire vivre ce *testbed* au dessus de l'infrastructure existante de l'association Tetaneutral.net.

L'idée principale est de couvrir la ville de Toulouse et particulièrement les établissements d'enseignement et les laboratoires intéressés via des passerelles LoRa. Notre architecture doit permettre d'émettre et de recevoir des trames LoRa via ces passerelles de façon simple et automatisable. L'architecture doit

être à «cœur ouvert», pour que des étudiants puisse avoir accès librement à tous les composants de ce système de façon à l'étudier globalement ou par partie.

3.2 Proposition d'architecture d'un *testbed* urbain ouvert

3.2.1 Passerelle LoRa

Afin d'émettre et de pouvoir transmettre des trames via la modulation LoRa, il est nécessaire de déployer des émetteurs accessibles via un backbone IP. Il existe deux types d'émetteurs, les émetteurs à destination des objets ou les émetteurs à destination des passerelles. Après avoir vérifié que les émetteurs des passerelles permettent bien d'utiliser directement la couche physique sans imposer une couche MAC, nous avons décidé de déployer des passerelles LoRa à base d'émetteurs sx1301.

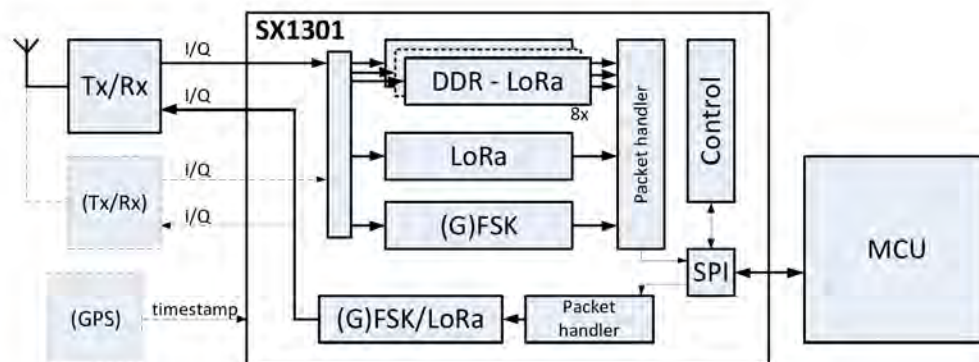


FIGURE 3.1 – Architecture de la puce sx1301 (Semtech)

La figure 3.1 permet de mieux comprendre l'architecture interne de cet émetteur. Il est constitué d'un étage de numérisation de la bande de fréquence ISM 868MHz sous la forme d'échantillons complexes I/Q. Ces échantillons sont ensuite séparés par sous-canaux comme on peut le voir sur la figure 3.2.

Les canaux IF8 et IF9 sont particuliers car ils sont fixes et servent pour faire des liens montants ou des liens entre les gateways. Les 8 canaux IF0 à IF7 sont eux plus complexes car ils intègrent 8 décodeurs LoRa mutualisables c'est à dire qu'un décodeur peut être raccordé à un canal à n'importe quel moment. Des détecteurs de préambule scrutent les canaux et raccordent un décodeur disponible dès qu'un préambule est détecté. De cette façon, un composant sx1301 peut au maximum recevoir 8 messages simultanément sur des canaux différents et quel que soit le spreading factor.

Nous n'utiliserons pas le canal spécifique pour la FSK dans le cadre de notre *testbed* et le canal spécifique LoRa sera fixé comme décrit dans la spécification LoRaWAN pour la région européenne.

3.2.1.1 Packet Forwarder

L'entreprise Semtech qui commercialise le sx1301, fournit une bibliothèque logicielle de gestion des communications avec le sx1301 par l'intermédiaire d'un bus SPI. Cette bibliothèque permet de charger le

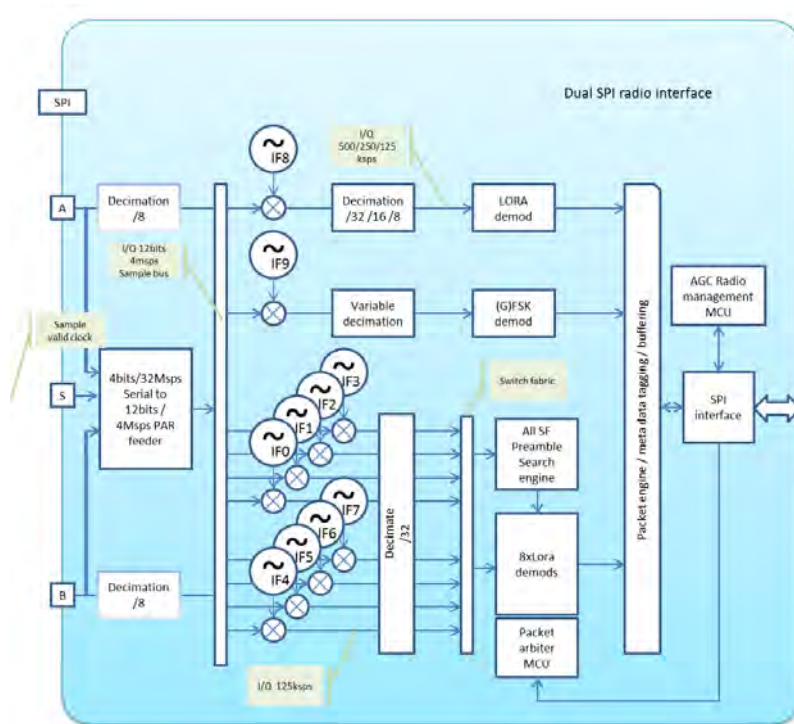


FIGURE 3.2 – Architecture interne de la puce sx1301 (Semtech)

firmware, de faire la gestion du démarrage ainsi que d'émettre et de recevoir des trames via le *frontend* radio.

Une bibliothèque logicielle permettant de faire le transport de ces trames à travers un réseau IP est fournie : le «Packet Forwarder». Cette bibliothèque se base sur la couche de transport UDP pour communiquer avec un serveur de réseau centralisé.

Semtech décrit dans la documentation du protocole [54], qu'il est exclusivement destiné à un usage basique pour faire des démonstrations sur un réseau privé et fiable. En effet, ce protocole se base sur UDP et ne rajoute pas de mécanisme pour chiffrer les messages ou pour gérer la retransmission des paquets en cas d'erreurs. Ce protocole utilise le paradigme de client/serveur qui ne nous satisfait pas car il n'est pas possible de faire simplement du multicast ou du broadcast sur plusieurs passerelles par exemple.

Nous décidons d'utiliser ce protocole uniquement en local sur la *gateway* et de faire la communication vers notre serveur de réseau via une autre couche de transport.

3.2.1.2 Intégration matérielle des gateways

Le déploiement réel d'un réseau de *gateways* nécessite par exemple de sélectionner le matériel, de l'intégrer puis de le tester. Un réseau LPWAN urbain nécessite de déployer des *gateways* sur les toits des immeubles les plus haut de la ville afin que l'antenne soit bien exposée en terme de couverture radio. Cette contrainte nécessite de sélectionner des *gateways* d'extérieur étanches aux intempéries.

Sur l'image 3.3, il est possible de voir deux adhérents de l'association en train de déployer notre antenne

LoRa contenue dans une boîte étanche blanche. Et sur l'image 3.4, il est possible de voir l'envergure de l'antenne droite par rapport à la boîte étanche. Nous utilisons trois antennes différentes suivant la taille et le gain désiré entre 5 dBi et 10 dBi.

L'alimentation des antennes s'effectue par PoE, ce qui permet de n'avoir qu'un câble ethernet à installer dans lequel circulent les données et l'alimentation.



FIGURE 3.3 – Exemple d'installation d'une antenne LoRa sur un toit de Toulouse

3.2.2 Réseau de collecte des trames LoRa

3.2.2.1 Mise en réseau des gateways

Après le déploiement de plusieurs passerelles indépendantes, il devient intéressant de chercher à les mailler afin de réaliser un réseau global.

Nous avons cherché différentes couches de transport afin de remplacer le «Packet Forwarder» et gagner en interopérabilité. Nous avons défini plusieurs critères d'évaluation pour cette couche de transport :

- Temps-réel : la couche de transport ne doit pas casser l'aspect temps-réel des communications via



FIGURE 3.4 – Aperçu global d’une antenne ainsi que la gateway associée

l’ajout de temporisations ou de transmissions par lots

- Multicast et broadcast : la couche de transport doit permettre d’ajouter des mécanismes de diffusion
- Fiabilité : l’utilisation de mécanismes de retransmission des trames est nécessaire pour ne pas avoir de perte de trames pendant le transport
- Sécurité : la couche de transport doit implémenter des standards en terme de chiffrement des communications

Comme entrevu au chapitre 2, notre choix s’est porté sur le protocole MQTT car il intègre très bien ces différents aspects.

Le protocole MQTT se base sur TCP ce qui lui permet de bénéficier des garanties de retransmission et de gestion de la connexion. L’inconvénient de TCP est qu’il crée une dépendance des trames les unes avec les autres. Si une trame doit être ré-émise, elle met en attente toutes les trames suivantes. Ce point pourra être corrigé en utilisant le nouveau protocole QUIC [56] en remplacement de TCP afin de bénéficier de la parallélisation de la gestion du flux. Nous reviendrons sur cet aspect dans les perspectives.

Le paradigme publish/subscribe permet de créer virtuellement une plateforme d'échange, où les différents agents peuvent publier et recevoir des messages. Cet aspect est très fort dans notre démarche d'interopérabilité car la notion d'émetteur et de destinataire est rompue. Cette mise à disposition des informations permet de connecter de nouveaux agents sur la plateforme sans avoir à modifier les autres et sans avoir d'impact sur l'existant. La plateforme est un lieu d'échange et de mise en commun des informations. Il est cependant possible de gérer les accès à cette plateforme et de gérer les droits en lecture et en écriture sur les flux d'informations de façon à garantir la sécurité entre les agents.

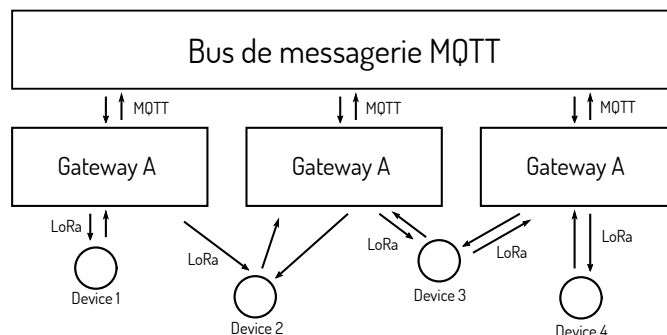


FIGURE 3.5 – Architecture de la collecte des trames LoRa

3.2.3 Concept de commutation au niveau du bus logiciel

Une gateway qui est connectée à un broker MQTT, expose une API permettant à un autre agent de lire les messages reçus par cette gateway et de lui transmettre des messages à envoyer. Pour ce faire, la passerelle s'abonne à un topic sur lequel circule les messages à lui envoyer et publie les messages qu'elle reçoit sur un autre topic. Ces topics contiennent l'adresse MAC de la gateway ce qui permet de pouvoir sélectionner la gateway à laquelle envoyer un message. Dans le topic, il est nécessaire d'indiquer l'adresse MAC de la gateway et *rx* ou *tx* afin de créer un topic d'émission et un topic de réception. Par exemple, voici le topic de réception de la gateway dont l'adresse MAC est *315000000000003* : *gateway/315000000000003/rx*

Lorsque l'on connecte une gateway qui implémente ce mode de fonctionnement sur un broker MQTT, elle se met à publier tous les messages qu'elle décode sur le médium sans-fil dans son topic de réception. Elle se met en attente de messages à envoyer en souscrivant à son topic d'émission. Il est dorénavant possible de programmer un agent logiciel qui va utiliser ces fonctions pour envoyer et recevoir des messages sur le réseau ainsi constitué.

Il est intéressant de faire le parallèle avec un réseau filaire de type Ethernet. Nous venons d'interconnecter logiquement des gateways qui sont identifiées par une adresse MAC unique, avec un mécanisme permettant d'envoyer et de recevoir des trames à ces gateways en les sélectionnant via leur adresse MAC. Ce comportement est très proche d'un commutateur réseau dans les réseaux Ethernet par exemple. Il est important de bien en comprendre les différences :

- Dans un contexte de réseau sans-fil, les passerelles ne partagent pas le même médium, ici l'air. Dans l'analogie avec un commutateur filaire, il est nécessaire de remarquer que les passerelles ne sont pas sur le même médium physique mais partagent les trames qui transitent sur leur médium.
- Dans l'hypothèse d'un déploiement de gateways bien supérieur au nombre minimal nécessaire afin d'obtenir une couverture minimale, chaque point géographique est couvert par une ou plusieurs

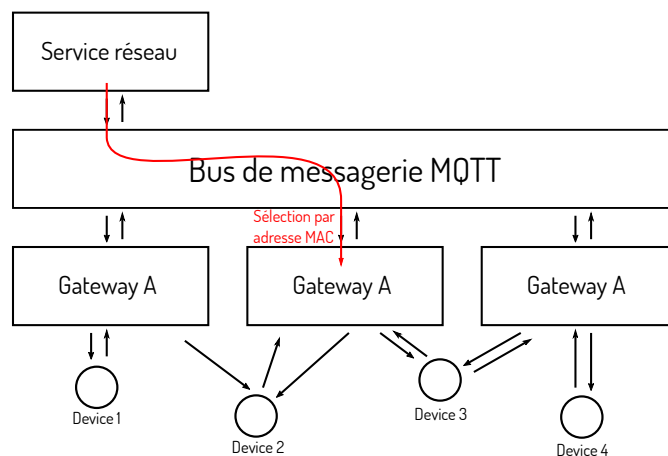


FIGURE 3.6 – Concept de commutation de trames LoRa via MQTT

gateways. Ce qui signifie que plusieurs passerelles peuvent tout de même partager des morceaux de médium grâce au cadencement de réseau par le serveur.

3.3 Programmation de nouveaux réseaux LoRa

La création de ce réseau métropolitain pour la pédagogie et la recherche, ouvre la voie à de nouvelles façons d’expérimenter et de développer de nouveaux protocoles pour les réseaux sans-fil. Nous décrivons dans cette partie, cette méthodologie que nous avons imaginée à la suite de ce déploiement.

L’avantage du déploiement d’un réseau métropolitain d’expérimentation est de pouvoir le modifier, le casser, le tester sans crainte de gênes au niveau des utilisateurs.

3.3.1 Programmation de nouveaux services réseau pour les LPWAN

3.3.1.1 Service réseau pour les LPWAN

Il est nécessaire de définir ce qu’est un service réseau LPWAN comme on peut l’apercevoir sur la figure 3.6.

Definition 3.3.1. Service réseau LPWAN : Composant logiciel connecté au service de collecte et qui est capable via des échanges protocolaires au sein de ce service de collecte de fournir une ou plusieurs fonctions réseau.

On peut considérer que le logiciel intégré dans une gateway est un service réseau LPWAN car il permet d’émettre et de recevoir des trames dans une zone géographique donnée en fonction des messages qui transitent sur la plateforme de collecte. Il est possible d’imaginer tout un ensemble d’autres services, d’identification, de chiffrement, de localisation, de suivi etc. Ces services peuvent être mis en relation de façon séquentielle, c’est-à-dire que les messages vont transiter d’un service à l’autre pour fournir un ensemble de services. Il est possible de visualiser un exemple d’ensemble de services sur la figure 3.7.

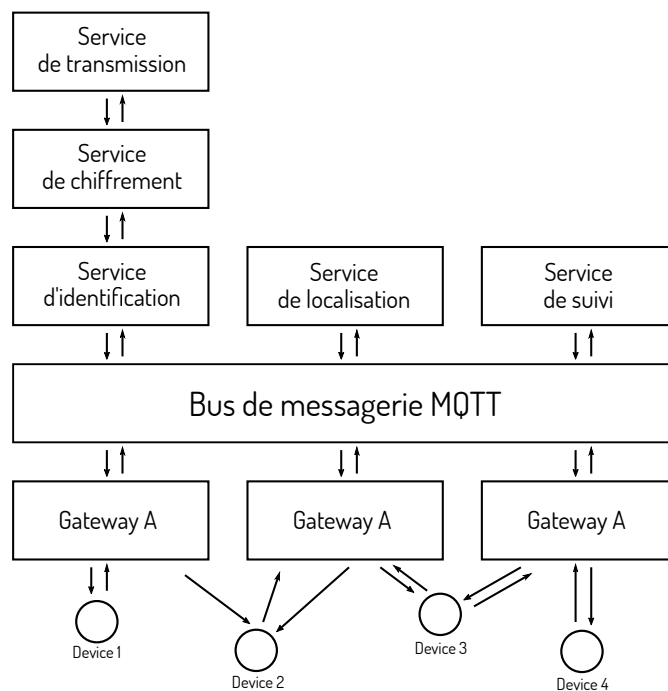


FIGURE 3.7 – Ensemble de services réseau LPWAN

Historiquement en réseau, un ensemble de services mis en relation de façon séquentielle est appelé un empilement protocolaire selon la théorie du modèle OSI. Dans les réseaux LPWAN que nous considérons comme des réseaux de messagerie, le modèle OSI reste un concept applicable mais il est désormais possible de faire des architectures beaucoup plus complexes et pas forcément modélisables via ce modèle OSI.

3.3.1.2 Programmation d'un service réseau pour les LPWAN

D'un point de vue programmation, un service réseau est un logiciel qui tourne en tâche de fond sur un système informatique et qui communique avec d'autres logiciels via différents protocoles de communication. Dans le cadre de l'architecture que nous proposons, un service doit gérer au minimum une connexion à un broker MQTT et implémenter la gestion de ce protocole. Un service réseau peut également avoir une interface web de gestion ou implémenter une API REST par exemple pour dialoguer avec une interface web mutualisée.

3.3.1.3 Déploiement d'un ensemble de services réseau pour les LPWAN

Les services réseaux étant des logiciels que l'on peut considérer comme des logiciels de *backend*, ils est possible d'utiliser les outils de déploiements qui sont en pleine émergence : les conteneurs informatiques.

Les mécanismes de "conteneurisation" informatique sont présents au sein des systèmes d'exploitation depuis de nombreuses années. Ils permettent de créer au sein d'un système d'exploitation l'isolation d'un processus en terme de réseau, de système de fichiers ainsi qu'en terme de privilèges et de gestion de l'accès aux périphériques. Cette technologie est redevenue d'actualité avec le développement de Docker [57] qui

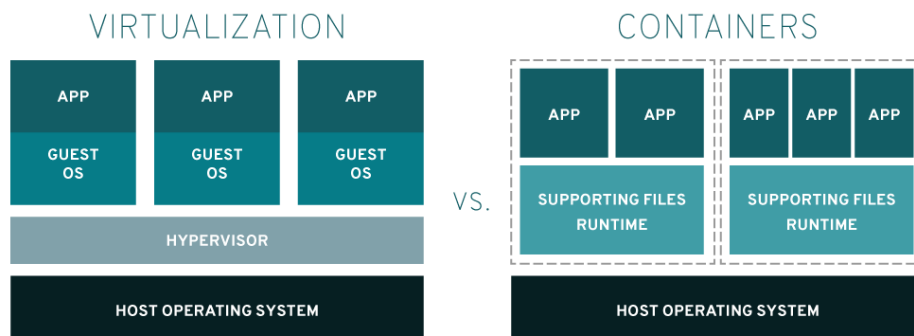


FIGURE 3.8 – Comparaison des concepts de conteneurisation et de virtualisation

est un logiciel de gestion de conteneurs. L'enthousiasme autour de cette technologie est également dû au développement du *cloud* et de ses offres SaaS. La conteneurisation permet une abstraction du système d'exploitation et de déployer des applications réseaux de façon transparente sur un ensemble de machines.

Un conteneur est livré sous forme d'image et contient en son sein l'ensemble d'un système de fichiers UNIX ainsi que les bibliothèques utilisées comme nous pouvons le voir sur la figure 3.8

3.3.2 Programmation des interactions avec les objets

3.3.2.1 *testbed* pour les réseaux LPWAN

Afin de développer de nouveaux réseaux LPWAN et de nouveaux services réseau, il est nécessaire de pouvoir simuler des noeuds mais aussi de les émuler.

La notion d'émulation consiste à faire du «Hardware In the Loop» c'est à dire avoir dans nos scénarios d'expérimentation des noeuds réels qui envoient des trames sur un médium réel. Via cette méthodologie, il est ainsi possible de tester le comportement d'un protocole vis à vis des perturbations et des performances réelles du médium, des interférences, de la dissymétrie des liens et autres phénomènes d'un environnement non simulé. Cette méthodologie est une technique de recherche très utilisée par notre équipe depuis de nombreuses années sur des technologies radio diverses (infrarouge, ultrasons, 433 MHz, Bluetooth, BLE, WiFi, UWB...).

L'originalité de nos travaux dans le cadre de la technologie LoRa est d'intégrer les passerelles qui sont des agents clés des réseaux LPWAN.

3.3.2.2 Passage du *testbed* sous MQTT

Le transport des trames sur MQTT est très intéressant au niveau des passerelles et nous avons proposé de prolonger ce concept au niveau des noeuds afin de pouvoir les assimiler à des passerelles mono-canal.

Pour ce faire nous avons déployé des noeuds LoRa à base de processeur ARM Cortex-M0+ que l'on peut reprogrammer via le port USB. Il est également possible de communiquer avec le processeur via l'émulation d'une liaison série via l'USB. Ces noeuds sont branchés à un contrôleur de type Raspberry Pi. Le Raspberry Pi permet d'alimenter les noeuds et de fournir le service, via ethernet, de reprogrammation et de transmission des messages qui circulent sur l'USB. Nous avons fédéré l'ensemble de nos contrôleurs

sur un VPN qui permet d'unifier les échanges malgré des configurations très diverses (pare-feu, filtrage, NAT ...).

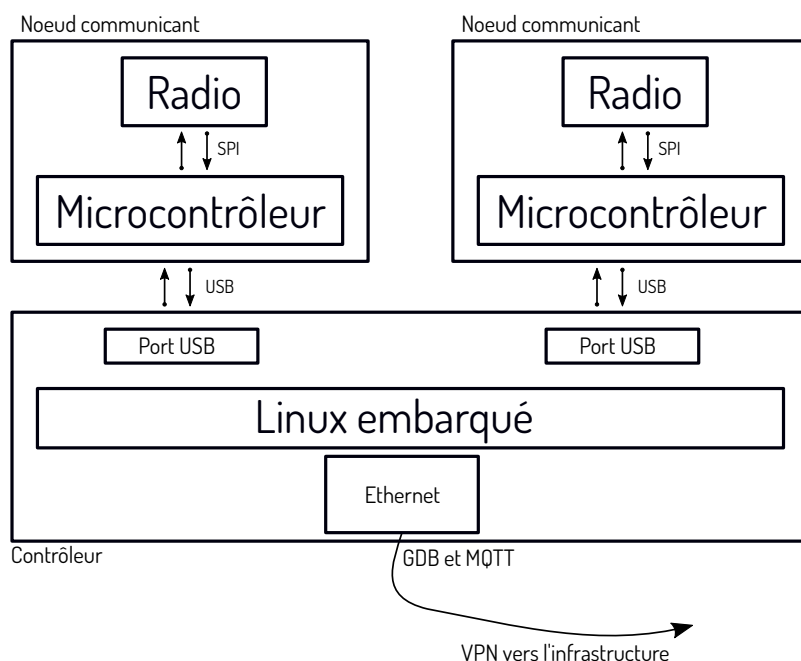


FIGURE 3.9 – Programmation et réception des messages par l'infrastructure *testbed*

Comme chaque nœud est relié en USB à un contrôleur, il partage une liaison série virtuelle qui permet une communication entre les deux. À partir de là, le contrôleur fait le pont entre la transmission série et une encapsulation via MQTT pour permettre l'émission et la réception des lignes envoyées sur le port série à distance via un agent connecté au broker. Une fois le déploiement de plusieurs contrôleurs et nœuds associés ainsi qu'un firmware spécifique qui gère la liaison série au niveau des nœuds, il est possible d'émettre et de recevoir des trames via la radio à distance, dans un langage de haut niveau et de façon reproductible.

3.3.2.3 Mise à jour des nœuds à distance

Afin de mettre à jour les nœuds à distance, nous avons choisi d'utiliser le standard de programmation GDB [58] avec une version client/serveur où le démon GDB qui est exécuté sur le contrôleur expose un service de programmation du nœud sur un port réseau du contrôleur. Ainsi il est possible via un client GDB de reprogrammer tous les nœuds du réseau qui sont connectés au serveur qui exécute les scripts d'expérience.

Le logiciel utilisé pour la gestion des debuggers par USB est OpenOCD [59], un logiciel open source capable de communiquer avec de nombreux debugger intégrés sur les cartes de développement pour un support très large de cibles matérielles.

3.4 Architecture globale du *testbed* urbain

La figure 3.10 présente l'architecture finale du *testbed* urbain. Il est possible de voir le parallèle entre les passerelles LoRaWAN et les contrôleurs des nœuds. Nous montrons ici l'implémentation d'un service réseau comme décrit dans la section 3.3.

En ayant rendu générique le concept de contrôleurs du réseau, nous pouvons utiliser sur le *testbed* différents nœuds communicants avec des technologies radio différentes. Notre équipe de recherche a développé plusieurs matériels que l'on peut retrouver sur le site <https://wino.cc/hardware/>. Ces matériels embarquent de l'UWB, du LoRa, de la FSK, du Bluetooth, et différents capteurs comme des IMU et des récepteurs GNSS.

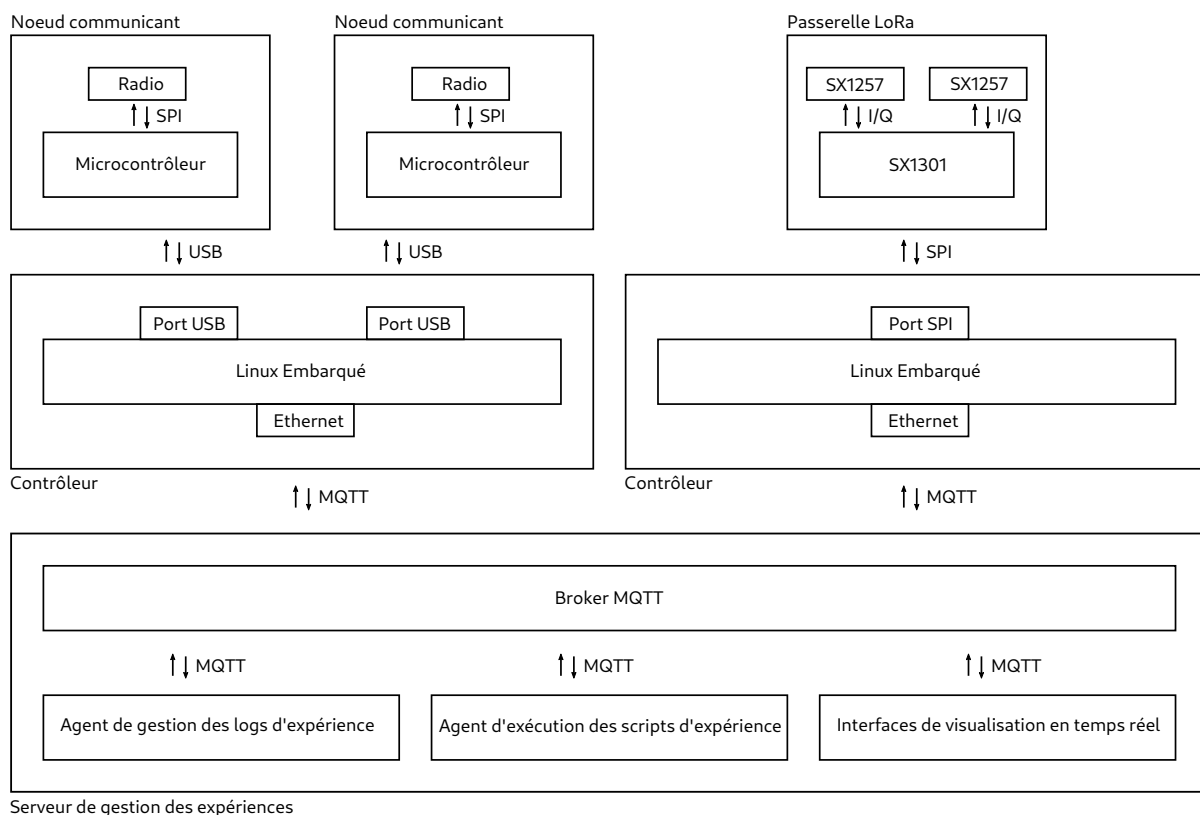


FIGURE 3.10 – Architecture logicielle et matérielle du *testbed* urbain

3.4.1 Services réseau du *testbed* urbain

3.4.1.1 Agent d'exécution des scripts d'expérience

La conception et les tests de nouveaux protocoles réseaux nécessitent d'exécuter des scénarios de tests sur un ensemble de systèmes communicants. Ces scénarios sont typiquement de l'envoi massif de trames pour tester les performances d'un réseau ou l'exécution de séquences d'échanges très spécifiques afin de tester le comportement d'un protocole. L'idée du *testbed* est de fournir l'infrastructure nécessaire à ces

expérimentations en conditions réelles. Afin de réaliser cela, un *testbed* doit contenir un moteur d'expérience qui gère l'ordonnancement des différentes expériences et leur bon déroulement.

L'ajout d'un bus temps-réel de messagerie dans l'architecture *testbed* permet au contrôleur de scénario d'avoir un accès en temps réel au déroulement de l'expérience. Chaque message reçu ou transmis sur le réseau est publié sur MQTT. De cette manière, le moteur d'expérience peut lancer des salves de trames et en analyser directement comment se comporte le réseau de façon globale.

Notre agent d'exécution des scripts d'expérience prend en paramètre un script d'expérience et le ou les binaires à programmer sur les différents nœuds. À partir de là, il programme les nœuds à distance via l'interface GDB et démarre son service réseau qui fournit via sa connexion au bus de messagerie, un service de programmation d'un réseau d'objets connectés.

3.4.1.2 Agent de gestion des logs d'expérience

Lors de l'exécution automatisée d'un scénario d'expérience, il est nécessaire de collecter l'ensemble des messages transmis sur le réseau de façon à pouvoir traiter par la suite les traces des données et les interpréter.

Nous avons implémenté un nouveau service réseau qui s'occupe de la gestion des logs et qui en s'abonnant à tous les topics de messages, les collectent et les enregistrent dans un fichier de log d'exécution.

Ce principe permet de comprendre un type de service que l'on peut appeler "observateur" qui ne fournit pas de service réseau en tant que tel mais qui s'abonne au bus de messagerie pour observer et extraire des informations pour un système externe. Il est également possible d'imaginer des services de sécurité ou de traçabilité dans d'autres cas d'usages.

3.4.1.3 Interfaces de visualisation en temps réel

Un dernier type de service réseau peut être la génération d'Interfaces Homme-Machine. Si un logiciel est connecté au bus de messagerie il peut y collecter des informations pour les représenter de façon visuelle à un humain. Ce service est également capable de transmettre des informations sur le bus en fonction de ces interactions.

Exemple : Interface de visualisation de la position géographique et des liens dans un contexte LPWAN

La figure 3.11 est un exemple d'interface web que nous avons développé afin de visualiser lors d'expériences dans la ville de Toulouse, les positions géographiques des nœuds équipés de puce GPS et des positions des passerelles avec qui ils communiquent. Nous avons représenté ces liens par un trait de couleur entre l'émetteur et le récepteur. La couleur du trait correspond à la puissance du signal reçu par le récepteur.

Nous pouvons observer les positions géographiques où les nœuds sont reçus par plusieurs antennes ainsi que les zones non couvertes.

3.5 Cas d'usages et valorisation scientifique du *testbed*

Le succès du déploiement de ce *testbed* métropolitain permet d'envisager les différentes voies de recherche qui s'ouvrent à la communauté dans l'étude des réseaux LPWAN. Il est possible de classer ces

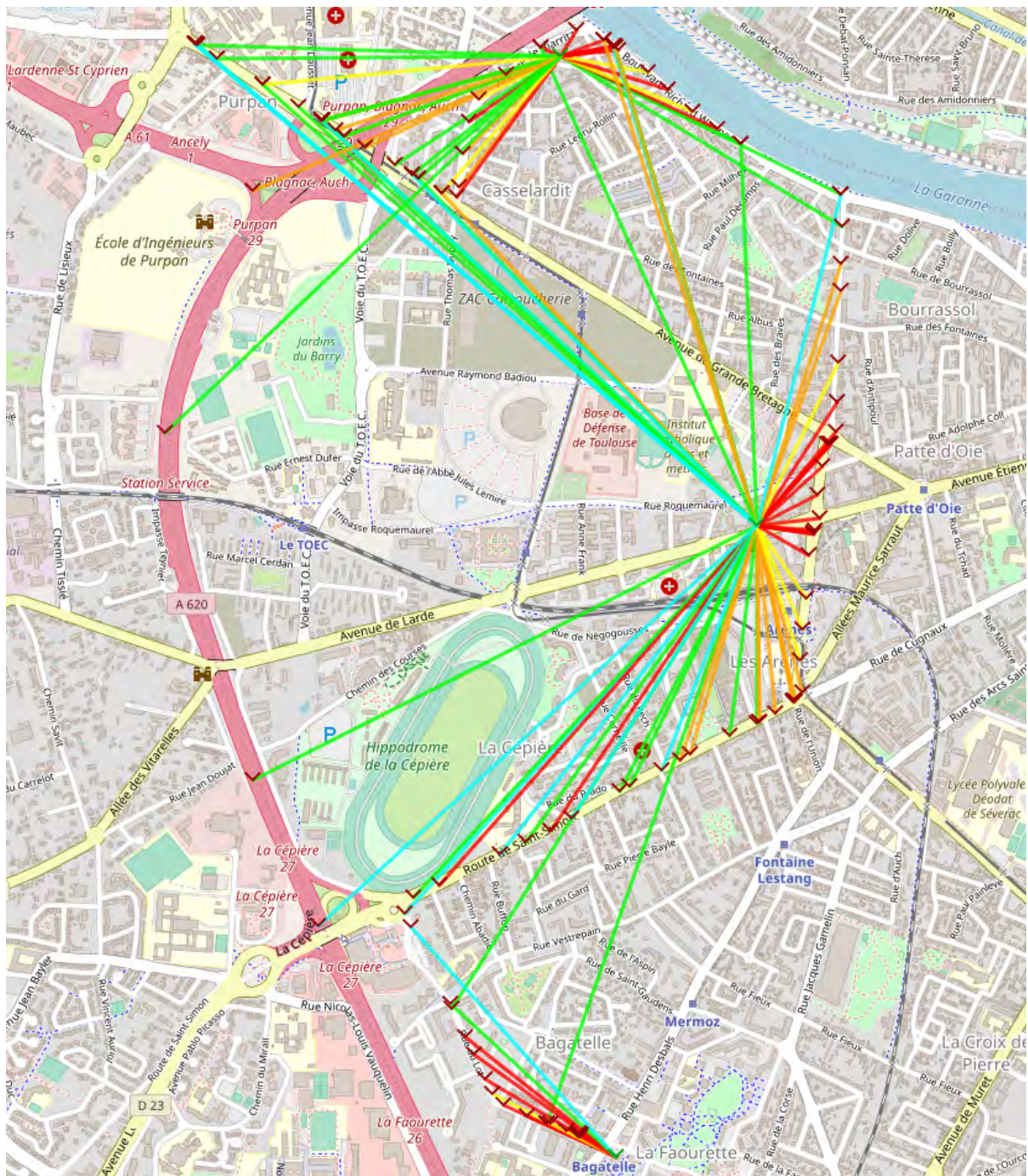


FIGURE 3.11 – Service réseau de visualisation des liens radio par géolocalisation

thématiques de recherche en deux thèmes : l'étude des cas d'usages et l'optimisation de la gestion et du déploiement des réseaux.

3.5.1 Cas d'usages

Les réseaux LPWAN occupent une place importante dans le concept de Smart City. En effet, les collectivités étudient avec intérêt les moyens technologiques permettant d'optimiser les services internes de la collectivité ainsi que de fournir de nouveaux services aux citoyens. C'est dans ce cadre que Toulouse Métropole lance le projet d'expérimentation «Vol de Nuit» afin de réaliser un état de l'art des technologies dans l'objectif d'équiper la ville de réseaux et d'applications.

Grâce aux travaux issus de cette thèse, nous avons pu intégrer ce projet afin de déployer un réseau privé pour la Métropole et présenter notre démarche testbed. L'idée d'intégrer le concept de testbed dans un réseau maintenu par la collectivité, est de pouvoir intégrer les volets de la pédagogie et de la recherche de part leur dimension collective.

Imaginer et développer de nouveaux cas d'usages dans la ville de ces réseaux est un champ complet de recherche. Nous n'avons pas cherché à l'explorer durant cette thèse mais plutôt à promouvoir ce réseau auprès d'équipes de recherche, de projets citoyens, d'associations ou encore d'entreprises.

Nous avons voulu que ce réseau soit porté et soutenu par l'association Tetaneutral.net afin qu'elle puisse gérer son exploitation en conservant les valeurs d'ouverture et de transparence que nous avons imaginés lors de sa création.

Nous avons utilisé ce réseau tout au long de la thèse dans le cadre d'enseignements en IUT, en licence et aussi en école d'ingénieur. Grâce à cette notion de réseau à cœur ouvert, il rend possible l'étude et la découverte de tous les concepts propres à ces nouveaux réseaux.

Enfin, les concepts ont également permis d'améliorer le testbed short-range et indoor de l'équipe « LocURa4IoT », construit initialement autour de la plateforme « WiNo ». Grâce aux apports du bus MQTT, de multiples traitements de données à la volée deviennent possibles (par exemple, dans le champs de la localisation indoor : calcul de l'erreur de distance, alimentation d'un algorithme de localisation, ...). La représentation des données de l'expérience en temps réel est également facilitée, comme l'illustre cette vidéo¹. Ces fonctionnalités sont très intéressantes et devraient permettre à la communauté des utilisateurs de testbeds d'intégrer plus facilement les travaux de recherche en réseau et protocoles dans des cas d'usages réels.

3.5.2 Valorisation

La valorisation scientifique du *testbed* métropolitain consiste à utiliser l'existant pour imaginer le futur. Elle permet la mise en œuvre et l'évaluation de nouvelles architectures, de nouveaux protocoles dans un environnement réel. Les spécifications de LoRaWAN sont vouées à évoluer et ce testbed peut participer à l'élaboration de ces changements.

Il rend possible des études statistiques visant à comprendre l'utilisation des ressources actuelles, d'en trouver les limites et de proposer des remèdes à ces manquements.

Afin d'illustrer les apports de ce *testbed* et d'ouvrir la voie de son exploitation, nous avons réalisé une étude visant à valider les modèles courants de simulation par la pratique. Cette étude est nécessaire lorsque l'on termine de déployer un réseau et que vient la question : quelles sont ses limites vis à vis du passage à l'échelle ?

1. <https://wino.cc/blog/automatic-ultra-wide-band-ranging-protocol-evaluation-a-motorised-testbed-for-decawino/>

Chapitre 4

Vers l'extensibilité d'un réseau LoRa

Le déploiement de notre réseau métropolitain nous a obligé à effectuer différentes campagnes de mesures dans le but d'étudier le placement, les performances et les limites de notre déploiement. À l'aide de ces mesures en environnement réel, il est possible d'établir une étude statistique de la propagation de signaux transmis via la modulation LoRa et de dresser un modèle du bilan de liaison. Une fois ce modèle validé, nous avons effectué une simulation d'un réseau LoRa pour évaluer les limites d'extensibilité en nombre de nœuds, qui sont évidemment très difficile à aborder de façon pratique.

Sommaire

4.1	Évaluation des performances de propagation de la modulation LoRa . . .	93
4.1.1	Travaux préalables de découverte de la couche physique	93
4.1.2	Bilan de liaison d'une transmission LoRa	95
4.1.3	Modèle de propagation	96
4.1.4	Campagne de mesures	101
4.1.5	Analyse des conditions de collision entre les trames LoRa	104
4.1.6	Modélisation des processus aléatoires de fading	105
4.2	Simulation d'un accès au médium sur une topologie mono-passerelle . . .	106
4.2.1	Objectifs de la simulation	106
4.2.2	Hypothèses instinctives d'étude	108
4.2.3	Topologie d'étude	108
4.2.4	Simulateur à évènements discrets	108
4.2.5	Simulation d'un accès multiple à une gateway LoRa	110
4.2.6	Paramétrage de la simulation	111
4.2.7	Disposition aléatoire des nœuds	111
4.2.8	Choix statique de paramétrage du spreading factor	112
4.2.9	Analyse du taux de livraison des trames (FDR)	113
4.3	Optimisation du taux de livraison des trames (FDR)	115
4.3.1	Critère d'optimisation	115
4.3.2	Méthode d'optimisation CMA	116
4.3.3	Proposition d'une nouvelle méthode d'optimisation par bandes successives	116

4.4	Étude de l’extensibilité de l’accès au médium des réseaux LoRa	120
4.4.1	Limite d’extensibilité par canal	120
4.4.2	Limite d’extensibilité globale	120
4.5	Algorithme de réglage automatique des paramètres de la couche physique	122
4.5.1	Automatic Data Rate de LoRaWAN	122
4.5.2	Les trois zones comportementales	123
4.5.3	Proposition d’algorithme	125
4.5.4	Conclusion et perspectives de ces travaux	128

4.1 Évaluation des performances de propagation de la modulation LoRa

4.1.1 Travaux préalables de découverte de la couche physique

Les travaux de notre équipe de recherche sont historiquement ancrés dans les domaines de l'ingénierie protocolaire et des réseaux de capteurs sans-fil. En utilisant les méthodologies et l'expertise acquise, nous avons appliqué cette même démarche exploratoire afin d'aborder, la thématique des LPWAN et plus particulièrement des réseaux LoRa.

4.1.1.1 Essais en laboratoire avec des émetteurs/récepteurs LoRa

La couche physique LoRa est une modulation propriétaire, développée par la société Cycleo qui a été rachetée par Semtech en 2012. Au niveau des systèmes embarqués, le composant physique utilisé est le sx1276 commercialisé par Semtech. Notre première démarche a été d'éprouver les différents drivers logiciels disponibles pour ce composant.

Notre choix s'est arrêté sur la bibliothèque Radiohead [60]. Cette bibliothèque est disponible sous licence GPLv2 et utilise les API de l'environnement de développement Arduino. Cette bibliothèque intègre au minimum les méthodes suivantes nécessaires à l'utilisation de la couche physique :

- *bool recv(uint8_t * buf, uint8_t * len)* : réception d'une trame
- *bool send(const uint8_t * data, uint8_t len)* : émission d'une trame
- *bool setFrequency(float centre)* : réglage de la fréquence d'émission et de réception
- *void setTxPower(int8_t power)* : réglage de la puissance d'émission
- *void setSpreadingFactor(uint8_t sf)* : réglage du spreading factor

Cette bibliothèque n'intègre pas de couche protocolaire comme LoRaWAN mais uniquement les méthodes de pilotage du transmetteur et de réglage de ses différents paramètres.

Au niveau des noeuds communicants, nous avons choisi d'utiliser le Feather M0 (Adafruit) que l'on peut voir sur la figure 4.1. Ce noeud est équipé d'un microcontrôleur SAMD21 de Microchip ainsi que d'un transmetteur sx1276.

4.1.1.2 Contribution à l'analyse expérimentale des collisions en LoRa

Le spreading factor est un des paramètres les plus importants de la couche physique LoRa. Comme nous l'avons présenté dans la sous-section 2.4.3.1, le spreading factor permet de régler le nombre de symboles par bit [61].

La propriété intéressante de cette modulation vient du fait que deux trames émises à la même fréquence et au même instant sont quasi orthogonales entre elles. Par conception au niveau signal, un démodulateur qui s'est fixé sur un des deux signaux va bénéficier d'une atténuation forte du signal considéré comme interférant. Il est possible de représenter ce taux de réjection inter spreading factor par une matrice. La matrice 4.2 est issue de l'article "Dedicated networks for IoT : PHY / MAC state of the art and challenges" [62] par Claire Goursaud et Jean-Marie Gorce de l'équipe Citi de l'INSA de Lyon. Cette matrice montre

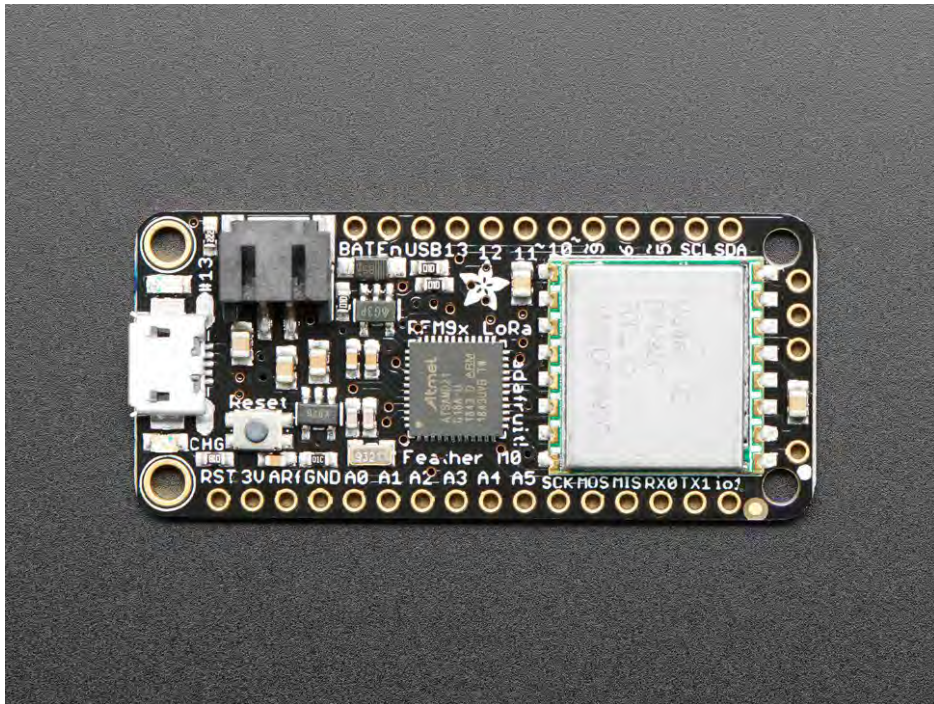


FIGURE 4.1 – Noeud Feather LoRa M0+

clairement que deux signaux qui ont le même spreading factor vont interférer alors que deux signaux avec un spreading factor différent, vont s’atténuer l’un et l’autre.

interferer desired SF	7	8	9	10	11	12
7	-6	16	18	19	19	20
8	24	-6	20	22	22	22
9	27	27	-6	23	25	25
10	30	30	30	-6	26	28
11	33	33	33	33	-6	29
12	36	36	36	36	36	-6

FIGURE 4.2 – Taux de réjection (dB) entre deux trames émises simultanément

Partant de cette propriété théorique, nous avons cherché à l’évaluer en pratique avant de bâtir de nouvelles couches MAC sur cette propriété physique. Nous avons pour cela réalisé une expérience en laboratoire avec deux noeuds Feather, synchronisés via des fils branchés sur des GPIO. Comme le représente la figure 4.3, la synchronisation est maître/esclave, le maître signale qu’il est prêt à envoyer un message via le changement d’état logique d’un fil et l’esclave valide qu’il est prêt en changeant de la même façon l’état d’un autre fil. Dès lors que l’esclave se signale prêt, les deux noeuds émettent simultanément une trame sur le front montant. Notre expérience consiste à utiliser ce mécanisme de synchronisation matérielle pour tester l’ensemble des couples de spreading factor.

Les résultats de cette expérience ont montré que la propriété d'isolation des canaux entre spreading factor est réelle et qu'elle ne dépend pas de la synchronisation temporelle entre les trames.

Le tableau 4.1 montre le taux d'erreur trame obtenu lors de l'émission de 500 trames avec toutes les combinaisons possibles entre deux émetteurs et deux récepteurs.

Dans le cadre de cette expérience, différents cas sont possibles :

- ok : chaque récepteur a reçu la trame de l'émetteur associé,
- 1 ok : un seul récepteur a reçu la trame de l'émetteur associé,
- nok : aucun des récepteurs n'a reçu de trame,
- error : les deux récepteurs ont reçu la même trame,

Il est important de regarder les colonnes où $sf1 = sf2$ et remarquer que dans ce cas, des trames sont perdues où arrivent au mauvais destinataire et donc qu'il n'y a pas d'étanchéité des canaux par spreading factor.

Ces travaux ont fait l'objet d'une publication à la conférence internationale Adhocnow 2018 [63].

Grâce à cette propriété, il est possible de développer de nouvelles couches MAC qui répartissent les trames équitablement entre les canaux de sorte à augmenter la capacité effective des réseaux LoRa.

TABLE 4.1 – Spreading Factor matrix for 10 ms delay

sf1	7	7	7	7	8	8	8	8	9	9	9	9	10	10	10	10
sf2	7	8	9	10	7	8	9	10	7	8	9	10	7	8	9	10
ok	0	500	488	497	497	0	484	500	489	489	37	489	487	491	495	8
1 ok	49	0	12	3	3	18	16	0	11	11	6	11	13	9	5	16
error	205	0	0	0	0	449	0	0	0	0	456	0	0	0	0	474
nok	246	0	0	0	0	33	0	0	0	0	1	0	0	0	0	2

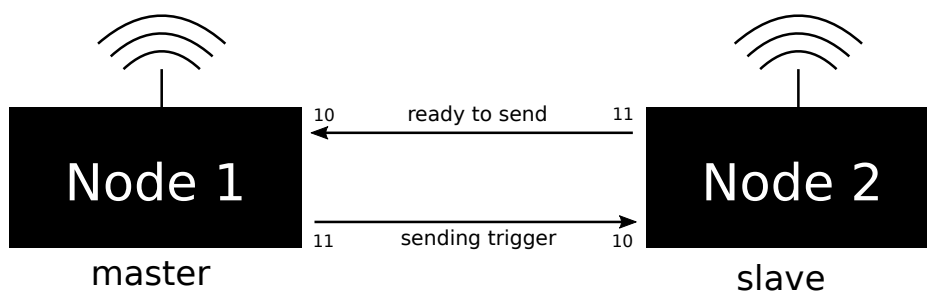


FIGURE 4.3 – Synchronisation entre les deux noeuds de l'expérience

4.1.2 Bilan de liaison d'une transmission LoRa

La méthodologie classique en radiocommunication pour faire une analyse des transmissions radiofréquences est d'établir un modèle du bilan de liaison. Ce bilan de liaison permet de caractériser tous les éléments logiciels, matériels, environnementaux et statistiques intervenant dans la réception des trames. Dans le

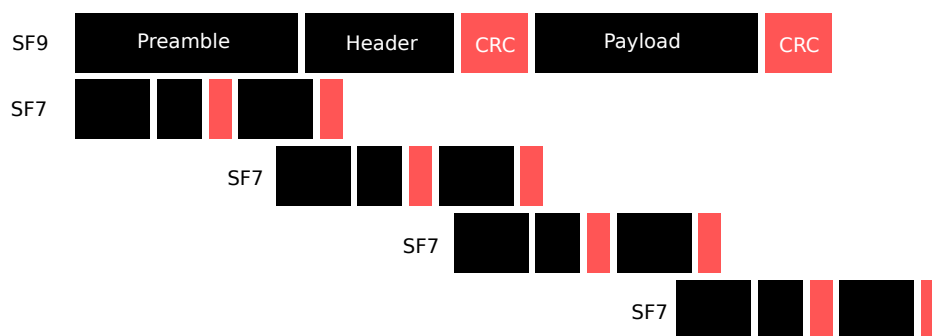


FIGURE 4.4 – Visualisation de la longueur des trames selon le spreading factor utilisé (l’abscisse est le temps)

cas de télécommunications numériques les messages sont transmis par bloc et chaque bloc doit être évalué afin de décider si il est valide ou non.

Le bilan de liaison est établi de l’émetteur jusqu’au récepteur en se basant sur une étude de puissance du signal. Dans la chaîne de transmission, il existe des éléments qui élèvent la puissance du signal (amplificateurs de puissance ou antennes) et des éléments qui l’atténuent (pertes par propagation, pertes dans les composants, atténuation des câbles, etc.). Un bilan de liaison est établi en décibel (dB) de manière à ce que les gains soient exprimés par des additions et des soustractions et non par des multiplications et des divisions. Nous avons représenté sur la figure 4.5 l’évolution de la puissance du signal en fonction de son trajet dans la chaîne de transmission. Un récepteur radio est défini par sa sensibilité, c’est à dire le niveau de puissance minimal nécessaire en entrée pour démoduler correctement un signal. L’étude de couverture d’un réseau consiste à identifier les conditions géographiques pour lesquelles un récepteur est en mesure de recevoir des trames par rapport à un émetteur.

Dans les documentations de Semtech sur la modulation LoRa, il est évident de remarquer que son avantage est d’avoir des composants peu onéreux avec une très bonne sensibilité par rapport aux autres technologies. La sensibilité des récepteurs est de -138 dBm en comparaison à celles des récepteurs WiFi qui sont en moyenne à -95 dBm. Ce gain significatif de sensibilité est principalement dû au gain de codage très élevé de LoRaWAN qui ajoute beaucoup de redondance par l’utilisation des spreading factors et du coding rate. Cette sensibilité importante va permettre de réaliser des transmissions à très longue distance. Le parti pris est un débit très faible pour deux raisons : la bande passante doit être faible et le gain de codage très élevé.

Le marché des réseaux LPWAN cible les applications où les noeuds ont besoin d’émettre des trames très courtes, à longue distance, de façon économe en énergie. Il ne faut pas oublier que comme expliqué dans la section 2.3, la législation en Europe dans la bande ISM 868 MHz, impose un rapport cyclique de 0,1% à 1% ce qui oblige d’avoir un temps minimal entre deux émissions. Il est important de ne pas parler de débit binaire pour une transmission LoRa mais de débit en messages par heures par exemple, en définissant une taille en octets pour un message et des paramètres de transmission.

4.1.3 Modèle de propagation

Les paramètres technologiques comme la puissance d’émission, le gain des antennes, la sensibilité du récepteur ou encore l’atténuation des câbles sont fixés, mesurables ou disponibles dans les documentations techniques. La difficulté majeure de la radiocommunication consiste à modéliser le canal radio et

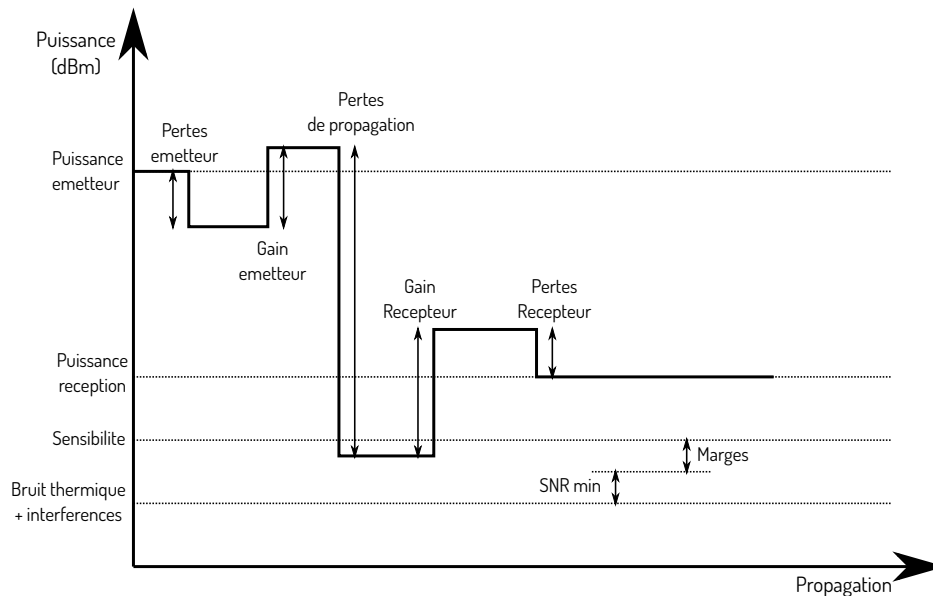


FIGURE 4.5 – Schématisation d'un bilan de liaison

les atténuations associées. Il est important de rappeler que lorsque qu'une onde électromagnétique se propage dans un milieu, elle va subir une atténuation qui est fonction de la distance de propagation, des obstacles rencontrés sur le chemin, de perturbations électromagnétiques ou encore des interférences entre les émetteurs. Ces paramètres parfois aléatoires sont complexes à définir précisément, il est nécessaire de les modéliser.

Il existe plusieurs approches afin d'établir un modèle de propagation :

- Modèles stochastiques : les modèles stochastiques consistent à modéliser le canal de propagation par des lois issues de la théorie des probabilités. Cette approche cherche à rendre compte des multi-trajets et de la diversité des canaux utilisés
- Modèles déterministes : les modèles déterministes utilisent la modélisation géométrique très précise d'un environnement afin de modéliser par des simulations à éléments finis ou par lancé de rayons la propagation des signaux
- Modèles statistiques : les modèles statistiques se basent sur des campagnes de mesures effectuées dans des environnements définis sur des bandes de fréquences fixées. De cette façon, des modèles mathématiques de l'atténuation en fonction de la distance sont données pour des types d'environnement (urbain, campagne, périphérie)
- Modèles semi-empiriques : les modèles semi-empiriques utilisent des bases de données géographiques et des modèles statistiques pour estimer l'atténuation du canal et optimiser le déploiement de stations de base. Ces modèles prennent en considération l'obstruction du cône de Fresnel qui n'est pas pris en compte dans les modèles statistiques purs

Nous avons décidé de nous orienter sur un modèle statistique pour essayer de confronter ce modèle avec une campagne de mesures réalisée en profitant de notre *testbed* métropolitain.

L'article "Low Power Wide Area Network Analysis : Can LoRa Scale" [64] de Orestis Georgiou et Usman Raza traite de la problématique de l'extensibilité en nombre de noeuds par antenne. Cet article se place en situation mono-gateway et cherche à estimer le nombre maximal d'émetteurs pour une QoS donnée. Cette étude se base sur un modèle stochastique afin de simuler les pertes de propagation.

Nous avons assisté aux journées non-thématiques du GDR RESCOM les jeudi 17 et vendredi 18 janvier 2019 à Grenoble, où Martin Heusse a présenté ses travaux à la suite de sa lecture de l'article qui vient d'être cité. Son approche a été de remettre en cause l'approche stochastique et de refaire la même étude avec des outils plus classiques de la radiocommunication. Parmi les sujets, il a proposé d'utiliser un modèle statistique plus classique : le modèle d'Okumura Hata. Il a utilisé ce modèle statistique sans prouver qu'il pouvait bien s'appliquer à la modulation LoRa. Comme notre *testbed* métropolitain est un environnement réel, il est possible de l'utiliser pour faire la preuve de la cohérence de ce modèle avec la réalité par une campagne de mesures.

4.1.3.1 Modèle d'Okumura

Le modèle d'Okumura [65] est un modèle de propagation du signal radio qui se base sur une campagne de tests effectués en 1968 dans la ville de Tokyo. Ce modèle se décline en trois environnements : urbain, semi-urbain et campagne.

Ce modèle est applicable dans une bande de fréquence donnée de 150 à 1920 MHz qui correspond aux conditions dans lesquelles les mesures ont été effectuées. Le modèle d'Okumura se définit comme ceci :

$$L = L_{FSL} + A_{MU} - H_{MG} - H_{BG} - \sum K_{correction} \quad (4.1)$$

- L = évaluation de l'atténuation moyenne par propagation
- L_{FSL} = perte de propagation en espace libre
- A_{MU} = atténuation médiane
- H_{MG} = gain d'exposition en hauteur du mobile
- H_{BG} = gain d'exposition en hauteur de la station de base
- $K_{correction}$ = Facteur de correction de l'environnement

4.1.3.2 Modèle d'Okumura-Hata

Le modèle de Hata [66] se base sur les travaux et sur le modèle de Okumura. Ces travaux ont lieu dans les années 1980, lors du développement de la téléphonie cellulaire (GSM). C'est également un modèle statistique car il se base toujours sur la campagne de mesures effectuée par Okumura mais au lieu de proposer des abaques, Hata propose un modèle mathématique avec des paramètres plus faciles à utiliser dans le contexte de la téléphonie cellulaire. Ce modèle est défini sur un sous-ensemble du modèle de Okumura mais reste disponible dans les versions urbaine, semi-urbaine et campagne.

Notre étude se situe en milieu de périphérie de ville (suburban en anglais) donc nous utilisons la version suivante du modèle de Okumura-Hata :

$$L_U = 69.55 + 26.16 \log_{10} f - 13.82 \log_{10} h_B - C_H + [44.9 - 6.55 \log_{10} h_B] \log_{10} d \quad (4.2)$$

- L_U = perte de propagation en zone urbaine (dB)
- h_B = hauteur de la station de base par rapport au sol (m)
- h_M = hauteur de la station mobile par rapport au sol (m)
- f = fréquence de transmission (MHz)
- C_H = Facteur de correction de hauteur d'antenne
- d = distance entre la station de base et le mobile (km)

Avec un paramétrage de C_H pour une zone urbaine peu dense :

$$C_H = 0.8 + (1.1 \log_{10} f - 0.7) h_M - 1.56 \log_{10} f \quad (4.3)$$

L'atténuation en zone périurbaine est finalement définie comme ceci :

$$L_{SU} = L_U - 2(\log_{10} \frac{f}{28})^2 - 5.4 \quad (4.4)$$

- L_{SU} = perte de propagation en environnement périurbain (dB)
- h_B = perte de propagation en environnement urbain (dB)
- h_M = fréquence de transmission (MHz)

4.1.3.3 Paramétrage du modèle pour notre environnement

L'environnement pour notre campagne de mesures est l'IUT de Blagnac et la ville de Blagnac environnante. Nous avons installé sur le toit du bâtiment (20 mètres), une antenne omnidirectionnelle reliée à une gateway LoRa permettant de couvrir l'IUT et le quartier environnant.

L'objet utilisé pour les tests est un Feather LoRa M0+ (figure 4.1) placé sur un tuyau en PVC de 2 mètres de hauteur. Le Feather est équipé d'une antenne filaire 1/4 d'onde réalisée en fil de cuivre rigide, positionné verticalement.

Avec ces informations il est désormais possible de paramétrer le modèle de propagation :

- Hauteur de l'antenne de la gateway par rapport au sol : 20 mètres
- Hauteur de l'antenne du mobile par rapport au sol : 2 mètres
- Fréquence centrale de la modulation d'étude : 868.1 MHz
- Distance entre la gateway et le mobile : entre 0 et 9 km



FIGURE 4.6 – Gateway LoRa sur le toit de l'IUT de Blagnac

La figure 4.7 représente le résultat de la courbe d'atténuation sur le médium en fonction de la distance selon le modèle d'Okumura-Hata. Nous remarquons que l'atténuation chute assez brutalement (-20 à -140 dB) jusqu'à 2 kilomètres. A partir de 2 kilomètres, l'atténuation diminue plus lentement (-3,33 dB par kilomètre). Les lignes bleues horizontales représentent les seuils de sensibilité. Plus le spreading factor est élevé, plus la sensibilité est faible, ainsi la ligne du haut représente la sensibilité pour un spreading factor 7 et celle du bas pour un spreading factor 12.

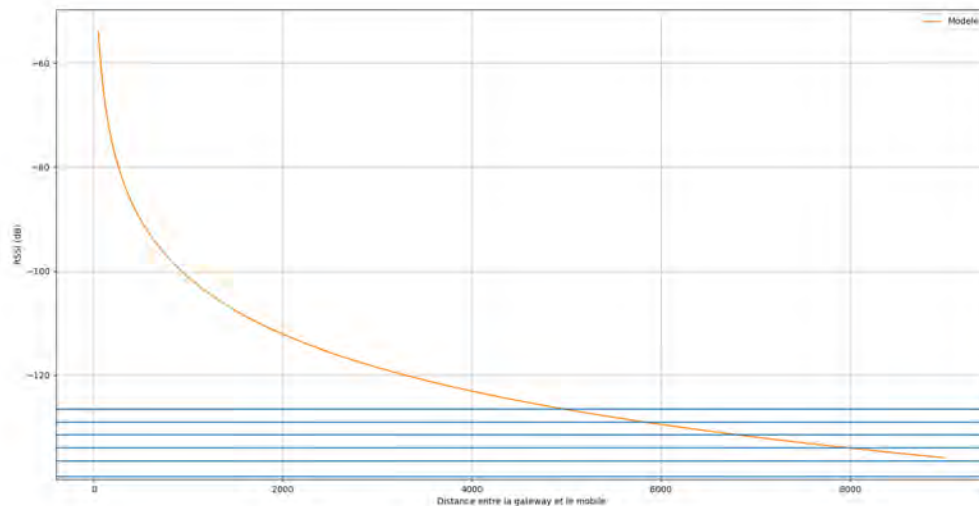


FIGURE 4.7 – Bilan de liaison entre la gateway d'étude et un nœud mobile

4.1.4 Campagne de mesures

Afin de confronter la courbe théorique avec des points en environnement réel, nous avons mis en œuvre une campagne de mesures.

Notre protocole expérimental est le suivant :

1. Définition d'un ensemble de points répartis selon 4 axes (nord, sud, est et ouest) entre 0 et 3 km (cartographie 4.8) du centre de la croix qui représente la gateway de l'IUT de Blagnac.
2. Écriture d'un script python qui s'abonne aux messages reçus de la gateway LoRa et qui enregistre dans un fichier les messages et les valeurs de RSSI et de rapport signal/bruit.
3. Déplacement selon les différents axes et prise de mesure sur les points pré-sélectionnés
4. Émission de 100 trames avec un délai de 1 seconde entre chaque trame.

Les trames ont été émises avec le paramétrage suivant de la couche physique :

- Spreading Factor : 10
- Fréquence d'émission : 868.1 MHz
- Coding Rate : 4/5
- Puissance d'émission : 23 dBm
- Bandwidth : 125 kHz



FIGURE 4.8 – Cartographie des différents points de mesures autour de l'IUT de Blagnac

Les résultats obtenus sont très proches pour les 4 axes, donc nous allons étudier uniquement les résultats pour l'axe nord. La figure 4.9 représente les mesures réalisées en regard avec la courbe théorique issue du modèle de Okumura-Hata.

4.1.4.1 Comparaison du modèle et des mesures réelles

Nous observons que la courbe théorique est globalement plus favorable que les résultats obtenus. Cependant la tendance du modèle se rapproche fortement de celle des mesures réelles. Cette variation va dépendre des obstacles et de l'environnement réel dans lequel les mesures sont effectuées.

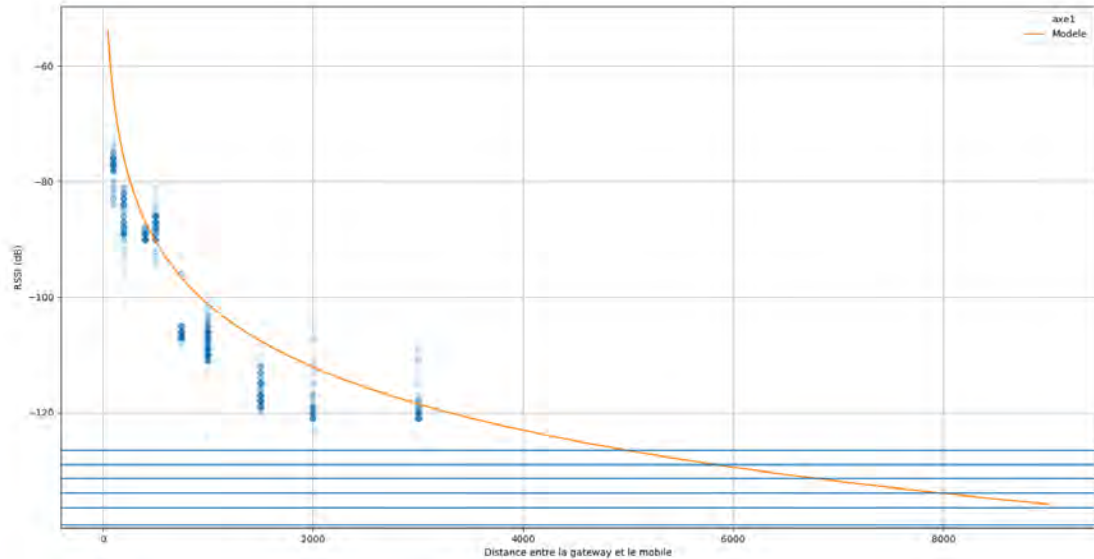


FIGURE 4.9 – Résultats issus de la campagne de mesure sur l'axe nord

4.1.4.2 Variance du RSSI pour une distance donnée

L'analyse des mesures permet de voir les variations de RSSI à une position géographique donnée. Le modèle de Okumura Hata donne une valeur unique pour une distance donnée alors que l'on observe des variations assez fortes du RSSI. Ces variations sont principalement dues à l'environnement qui varie entre le mobile et la station de base ainsi qu'aux phénomènes statistiques de rebonds et de multi-chemins. Lors de la modélisation du canal radio LoRa, nous proposons de combiner le modèle de Okumura Hata à une distribution statistique, typiquement en radiocommunication une distribution de Rayleigh [67].

4.1.4.3 Deux modes principaux de probabilités

Comme le laissait supposer l'analyse du modèle théorique, nous observons sur la courbe 4.10, deux modes principaux : un premier mode pour les nœuds qui sont à moins de 1 km de l'antenne et un deuxième mode pour ceux qui sont au delà de 1 km.

Bande favorable : Le premier mode correspond au premier disque de portée de l'antenne, zone dans laquelle les objets ont un RSSI bien au dessus des limites de sensibilité du récepteur. Dans cette zone les distributions des RSSI sont plus diffuses car comme la courbe de tendance est très pentue dans cette zone, un obstacle ou l'impact des multi-chemins entraîne une forte variation du RSSI. C'est une bande plutôt favorable aux transmissions entre un nœud et la gateway.

Bande de jeux équitables : Nous avons défini la seconde bande comme une bande de jeux équitables, c'est une bande très étendue dans laquelle les RSSI vont être assez similaires. Cette bande s'étend de 1 km de la gateway jusqu'à la limite de portée. Nous observons via les mesures que la courbe de tendance représente la moyenne des mesures et que la plage réelle se situe entre -100 et -120 dBm. Les nœuds ont

ainsi une probabilité forte d'émettre une trame qui est reçue avec un RSSI similaire sur une large plage de distance. Bien entendu plus le nœud est loin de la gateway moins il va avoir de chance d'émettre une trame qui sera reçue avec un RSSI élevé.

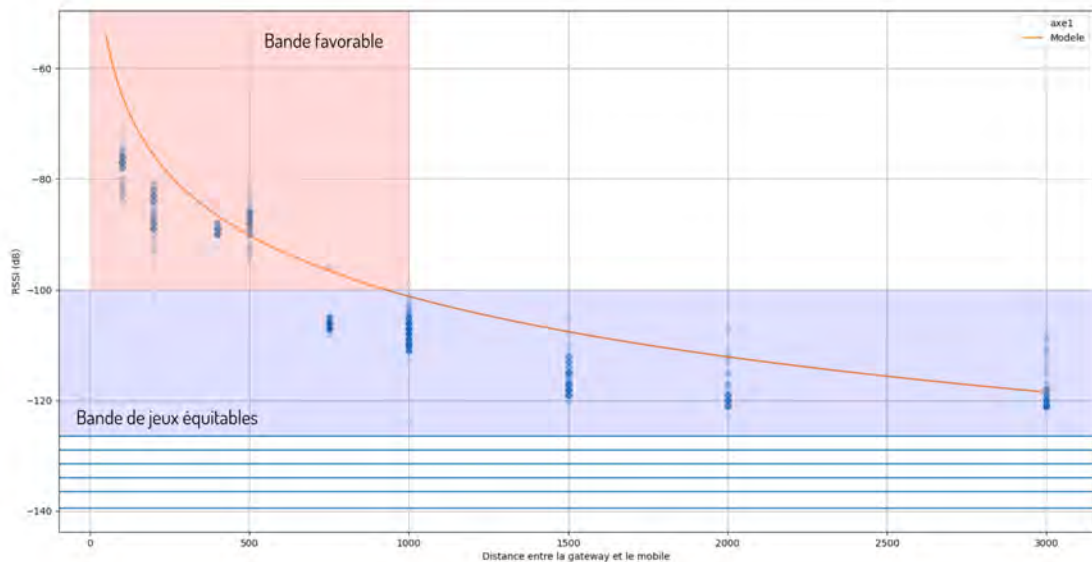


FIGURE 4.10 – Définition de la bande favorable et de la bande de jeux équitables

4.1.5 Analyse des conditions de collision entre les trames LoRa

Afin d'étendre notre étude à un cas multi-nœuds donc multi-émetteurs, il faut comprendre comment des trames envoyées simultanément par plusieurs nœuds, vont entrer en collision au niveau du récepteur. De façon plus précise, une ou plusieurs trames qui étaient dans de bonnes conditions pour être reçues, ne vont pas l'être car elles se perturbent entre elles.

A la suite de notre expérimentation sur les collisions entre les spreading factors, nous avons compris et validé le tableau 4.2 qui montre les taux de réjection de trames entre elles. Cette analyse met en évidence que dans l'étude des collisions entre différentes trames, il faut séparer d'une part les cas de collision entre plusieurs trames de spreading factor différents et d'autre part les cas de collision par trames de même spreading factor.

4.1.5.1 Collision entre plusieurs trames de même spreading factor

Une collision entre plusieurs trames de même spreading factor s'étudie en observant la diagonale du tableau 4.2. Il est très net que des trames qui se présentent simultanément au niveau du récepteur avec un même spreading factor vont devoir avoir une différence de niveau de puissance d'au moins 6 dB. C'est l'effet de "capture" qui est également applicable pour la modulation LoRa : lorsque deux signaux

électromagnétiques arrivent au niveau d'un récepteur, seul le signal plus fort d'au moins 6dB sera reçu correctement (phénomène de capture).

4.1.5.2 Collision entre plusieurs trames de spreading factor différents

Lorsque plusieurs trames de spreading factors différents arrivent simultanément au niveau d'un récepteur LoRa, elles se rejettent les unes les autres par construction physique de la modulation. Ces taux de réjection sont représentés dans le tableau 4.2. Il est important de comprendre que ce taux de réjection inter-spreading factor permet de considérer que plusieurs trames émises à des spreading factors différents sont dans des canaux logiques disjoints. Cependant cette hypothèse n'est pas parfaite car un signal ajoute quand même une puissance de bruit sur les autres signaux qui fait chuter son rapport signal/bruit et peut entraîner dans des conditions limites des pertes de messages par détérioration du canal.

4.1.5.3 Modélisation des collisions dans notre étude

Bien qu'une trame altère le rapport signal/bruit des autres signaux reçus simultanément à des spreading factors différents, le taux de réjection est assez élevé (entre 16 et 36 dB) pour pouvoir faire l'hypothèse que l'ajout de ce bruit est négligeable et donc non pris en compte dans cette étude.

Nous considérons qu'un message est reçu lorsque aucune autre trame n'est émise au même moment avec le même spreading factor ou alors que la trame interférante est reçue avec un niveau de puissance inférieur d'au moins 6 dB. Attention, il est nécessaire de vérifier cette propriété pendant toute la période d'émission de la trame car les trames ne sont pas synchronisées mais peuvent tout de même rentrer en collision partielles temporellement.

4.1.6 Modélisation des processus aléatoires de fading

L'étude de la figure 4.9 issue de la campagne de mesures montre la variance du RSSI lorsqu'un nœud est fixe. Ce phénomène aléatoire représente une variation conséquente du RSSI et doit être modélisé dans notre modèle de propagation du canal radio.

Ce type de phénomène aléatoire est très connu en radiocommunication, c'est l'effet de fading. Lorsqu'un signal électromagnétique est émis par un émetteur, il va lors de sa propagation se réfléchir sur les différents obstacles de l'environnement. Au niveau d'un récepteur, ces différents signaux sont décalés dans le temps car ils ont parcouru à la même vitesse des distances différentes. Suivant la modulation, ces signaux peuvent être constructifs ou destructifs. Malheureusement, en pratique, ces signaux ont souvent tendance à être plutôt destructifs.

La modélisation de ce phénomène se fait par ajout d'une loi de probabilité autour de la valeur moyenne des pertes de propagation. Le choix de cette distribution est fonction de la présence de ligne de vue entre l'émetteur et le récepteur ou non. Si une ligne de vue est présente c'est un canal dit de "Rice" si il n'y a pas de ligne de vue c'est un canal dit de "Rayleigh".

La figure 4.11 représente les différentes distributions de probabilités du RSSI en fonction de la distance entre le nœud et la gateway lors de notre campagne de mesure. Nous remarquons pour les distances inférieures à 2 000 mètres, un comportement plutôt proche d'un canal de Rice et pour les distances supérieures à 2 000 mètres un comportement plutôt proche d'un canal de Rayleigh.

La figure 4.13 représente les deux distributions paramétrées de façon à ce que 90% de la densité de probabilité soit en dessous de la valeur théorique obtenue via le modèle de Okumura Hata et avec un écart type représentatif des mesures, ici 4 dB.

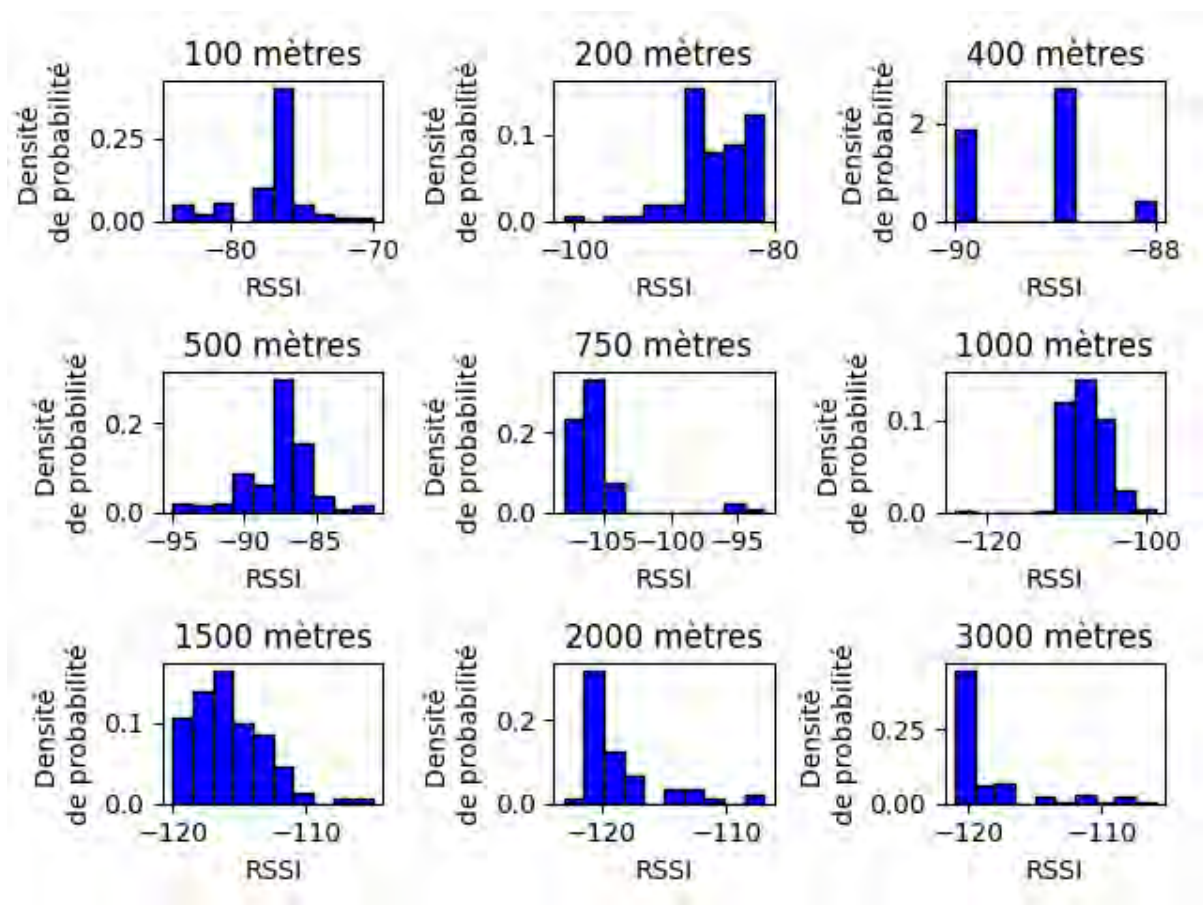


FIGURE 4.11 – Distributions de probabilités pour les différents distances

Nous utiliserons dans notre simulateur le paramétrage suivant au niveau du modèle statistique :

- Modèle de Rice si la distance est inférieure à 2 000 mètres avec un écart type de 4 dB et la limite de 90% de la distribution en dessous de la valeur théorique
- Modèle de Rayleigh si la distance est supérieure à 2 000 mètres avec un écart type de 4 dB et la limite de 90% de la distribution en dessous de la valeur théorique

4.2 Simulation d'un accès au médium sur une topologie mono-passerelle

4.2.1 Objectifs de la simulation

Afin de mieux appréhender la problématique d'accès au médium et de paramétrage optimal de la couche physique, nous avons cherché à modéliser et simuler les échanges de trames au niveau physique et MAC entre plusieurs nœuds et une gateway LoRa.

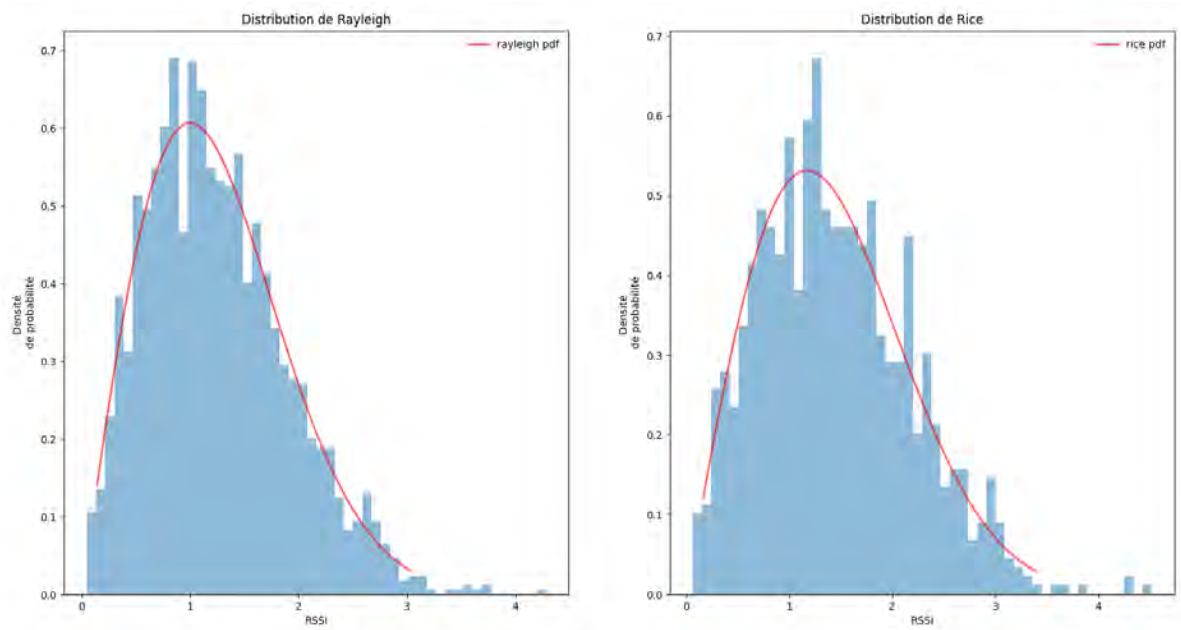


FIGURE 4.12 – Distributions de probabilités de Rice et de Rayleigh

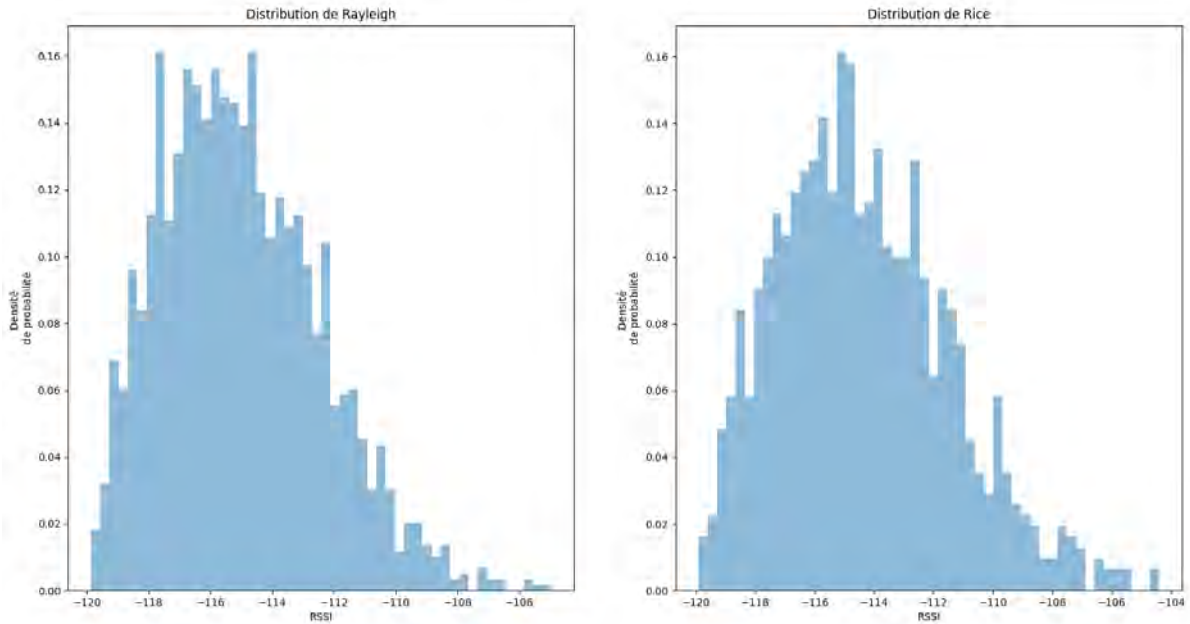


FIGURE 4.13 – Modélisation de la distribution des RSSI pour une distance de 1500 mètres

4.2.2 Hypothèses instinctives d'étude

Lorsque l'on relève les seuils de sensibilité par spreading factor, il est intéressant de remarquer qu'un spreading factor plus élevé permet de gagner une marge de sensibilité dans le bilan de liaison, au détriment d'un temps d'émission plus long. Cette augmentation des seuils de sensibilité est observable sur la figure 4.10 où les traits horizontaux en bas indiquent les seuils de sensibilité des récepteurs pour un spreading factor de 7 à 12. Plus le seuil est bas et plus le spreading factor est élevé.

Dans notre démarche d'optimisation du médium, nous cherchons à répartir au maximum les nœuds dans les canaux physiques et de paramétrer ces nœuds de façon à ne pas gaspiller de ressource inutilement. L'émission d'une trame dans un spreading factor plus élevé va prendre plus de temps sur le médium et donc gaspiller de la bande passante. Il est ainsi important de configurer avec un spreading factor élevé les nœuds qui nécessitent réellement ce gain de marge de sensibilité.

Un nœud peut avoir besoin d'une marge de sensibilité lorsqu'il est trop éloigné de l'antenne et en limite de sensibilité ou alors lorsqu'il est plus proche de l'antenne mais dans des conditions moins favorables comme dans un sous-sol ou sous une plaque d'égout.

4.2.3 Topologie d'étude

La topologie d'étude que nous avons sélectionnée est représentée sur la figure 4.14. Cette topologie est constituée d'une station de base au centre, notre antenne LoRa ainsi qu'une multitude de nœuds communicants répartis aléatoirement dans sa zone de portée. Nous considérons l'environnement homogène c'est à dire que nous ne considérons pas d'obstacles réels mais une atténuation de propagation qui suit le modèle de propagation statistique de Okumura Hata dans sa version ville moyenne.

Concrètement, notre étude cherche à montrer comment configurer l'ensemble des couches physiques de ces nœuds afin d'optimiser globalement l'accès au médium. Cette étude permet également de quantifier l'extensibilité des réseaux LoRa en terme de nombre d'objets par antenne.

Les disques concentriques délimitent des zones géographiques dans lesquelles un ensemble de nœuds est configuré avec le même spreading factor. Nous cherchons les limites de portée à partir desquelles il est optimal de changer de spreading factor.

Nous sommes conscients que cette approche n'est pas réaliste car ces limites sont théoriques mais elle apportent une référence optimale à laquelle confronter les résultats d'une campagne de mesure en environnement réel.

4.2.4 Simulateur à évènements discrets

Afin de simuler l'accès au médium pour un réseau LoRa, nous avons décidé de concevoir un simulateur à évènements discrets. Le principe de ce type de simulateur est de discrétiser le temps et d'évaluer un ensemble d'états et de conditions à chaque pas de simulation.

4.2.4.1 Framework de simulation Simpy

Dans notre équipe nous travaillons avec les outils issus du framework scientifique SciPy [68] de la communauté Python. Dans cet écosystème, il existe une bibliothèque de simulation à évènements discrets nommée SimPy [69]. Cette bibliothèque fournit les outils logiciels pour définir un scénario et un environnement de simulation en python et l'exécuter.

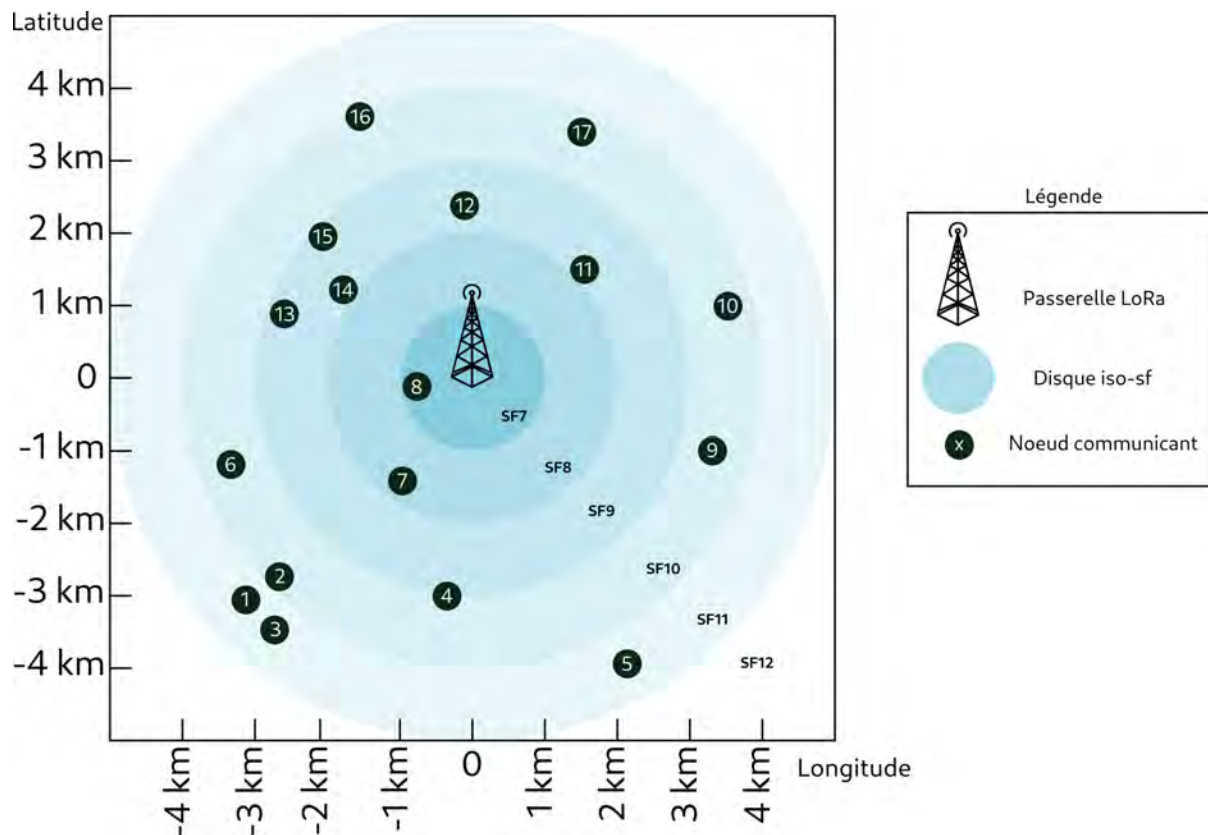


FIGURE 4.14 – Topologie utilisée dans notre simulateur

Cette bibliothèque de simulation fournit les outils suivants pour la simulation :

Environnement : Un environnement est un objet global dans lequel il faut définir via des méthodes l'ensemble des objets qui vont intervenir dans la simulation. C'est cet objet qui gère et exécute la simulation à l'aide de la méthode "run".

Évènement : Un évènement est un objet qui est généré sous certaines conditions et qui sert à modifier un ou plusieurs états d'un processus ou d'une ressource partagées.

Processus : Un processus est un objet qui contient un ensemble d'états et est relié à un ensemble de ressources partagées. Ses états évoluent dans le temps lors de la simulation par l'action d'évènements et/ou de modification des ressources partagées avec d'autres processus.

Ressource partagée : Une ressource partagée est un objet qui contient un ensemble de valeurs modifiables par plusieurs processus.

Gestion du temps : SimPy fournit les méthodes et objets pour gérer la notion du temps discret. Il est possible de réaliser une simulation en temps réel, c'est à dire que le temps est discrétisé mais la simulation est exécutée à une échelle de temps synchronisée dans un référentiel de temps humain.

4.2.4.2 Simulation d'une stratégie d'accès au médium

La problématique d'accès au médium consiste à observer un ou plusieurs médiums sur une période de temps fixé et d'analyser l'ordonnancement et les collisions entre les différentes trames d'un réseau.

Afin de réaliser cette simulation, il est nécessaire de modéliser les mécanismes de la couche physique : conditions de collision entre les trames, modélisation des mécanismes physiques du canal (atténuation, fading, etc.), conditions de réception d'une trame au niveau d'un récepteur, modélisation réaliste de la durée d'une trame.

Médium de transmission : Nous définissons un médium comme une ressource partagée entre plusieurs émetteurs. C'est une ressource qui a un état actif ou inactif suivant qu'une trame est en train d'être transmise sur ce support ou pas. Afin de suivre les transmissions sur ce support, la ressource contient une liste des trames et des émetteurs qui sont actifs sur le médium.

Un émetteur : Un émetteur est un processus qui à certains instants du temps génère un évènement afin de s'emparer d'une ressource partagée inactive dans l'objectif d'émettre une trame. Cet émetteur va attendre le temps de la transmission de la trame puis va émettre un nouvel évènement afin de relâcher le médium. Entre deux émissions, il est dans un état d'attente dans lequel il ne génère pas d'évènements.

Un récepteur : Un récepteur est un processus qui écoute un ensemble de médiums donc de ressources partagées dans la simulation. Il cherche à récupérer les messages qui transitent sur ces ressources partagées. A chaque instant du temps, il évalue si il y a une collision sur les médiums qu'il surveille. Si tout au long de la transmission d'une trame il n'observe pas de collision, il valide la réception de la trame à la fin de la transmission et met à jour les statistiques de la simulation.

Les étapes pour définir un scénario de simulation sont les suivantes :

1. Instanciation d'un ensemble d'émetteurs et de récepteurs
2. Répartition géographique de ces agents de façon réaliste
3. Définition d'un ensemble de médiums partagés entre ces agents
4. Définition d'une stratégie d'émission de trames (périodicité, longueur, ré-émission etc.)
5. Définition d'un temps de simulation qui doit être suffisamment grand pour que les statistiques soient recevables.

Une fois la simulation exécutée, nous analysons les statistiques suivantes :

1. Taux de réception invalide des trames (Frame Error Rate) c'est à dire le nombre de trames non reçues divisé par le nombre total de trames émises.
2. Pourcentage d'occupation de chaque canal
3. Taux d'émission réussi d'une trame c'est à dire le nombre de trames correctement transmises divisé par le nombre total de trames émises. C'est l'équivalent de (1-FER) mais au niveau de chaque émetteur.

4.2.5 Simulation d'un accès multiple à une gateway LoRa

Nous appliquons l'ensemble des outils et concepts définis dans les sections précédentes pour mettre en œuvre la simulation discrète d'un accès au médium LoRa par un ensemble de nœuds sans-fil de façon à évaluer l'extensibilité et les performances de ce type de réseau.

4.2.6 Paramétrage de la simulation

Les paramètres suivants ont été utilisés pour configurer le simulateur.

Définition des paramètres de simulation

Paramètres et hypothèses utilisés dans le cadre de cette simulation

Hauteur de la gateway par rapport au sol	30 mètres
Gain de l'antenne de la gateway	6 dBi
Hauteur des mobiles par rapport au sol	2 mètres
Gain de l'antenne des mobiles	0 dBi
Puissance d'émission des mobiles	20 dBm
Distance maximale entre la gateway et un mobile	9 kilomètres
Temps de simulation	1 000 000 ms
Nombre de mobiles	500
Modèle de propagation	Suburban Okumura Hata
Liste des spreading factors recevables par la gateway	7, 8, 9, 10, 11, 12
Nombre de canaux fréquentiels de la gateway	8 canaux
Taille des trames	20 octets
Fréquence d'émission des trames	Rapport cyclique max

TABLE 4.2 – Définition des paramètres de simulation

4.2.7 Disposition aléatoire des nœuds

La disposition des nœuds sur la zone de couverture est aléatoire. C'est un choix qui n'est pas forcément représentatif de la réalité où les nœuds sont souvent regroupés en clusters géographiques pour des applications données. Cependant nous cherchons à trouver des limites géographiques théoriques qui nécessitent d'avoir une répartition homogène. Nous chercherons dans les études suivantes à traiter le cas des regroupements de nœuds.

Voici la fonction Python utilisée donc notre scénario de simulation afin de répartir les nœuds en deux dimensions (x;y) autour de la station de base (0;0).

```
1 # Distribution uniforme sur un disque
2 rho = np.random.uniform(0, DISTANCE_MAX)
3 phi = np.random.uniform(0, 2*np.pi)
4 x = int(rho * np.cos(phi))
5 y = int(rho * np.sin(phi))
```

FIGURE 4.15 – Fonction de répartition des noeuds

L'inconvénient de cette fonction de répartition est qu'elle est homogène sur la répartition selon un axe et selon un angle, cela entraîne une densité plus élevée de nœuds autour de la gateway. Cette propriété

n'est pas gênante parce que la gateway reçoit très bien ceux qui sont proches d'elle et plus on s'éloigne de la gateway, plus elle ne va recevoir que les nœuds bien placés donc une densité de plus en plus faible.

La figure 4.16 montre un exemple de répartition géographique des nœuds issue de notre simulateur.

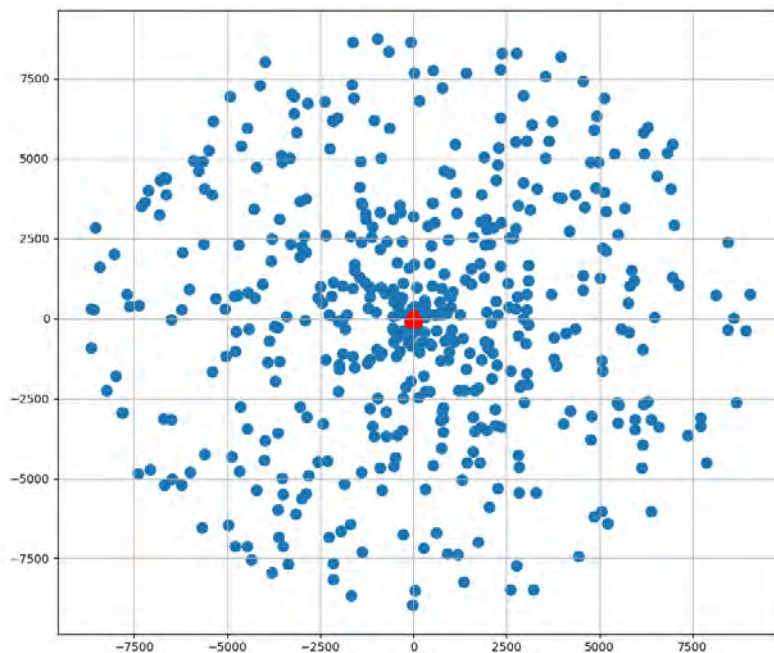


FIGURE 4.16 – Répartition des nœuds en mètres autour de la gateway

4.2.8 Choix statique de paramétrage du spreading factor

L'objectif de notre étude est de trouver les limites concentriques optimales pour le choix de répartition des nœuds en terme de spreading factor. Cette optimalité devra être approchée par simulations successives en faisant évoluer les limites afin d'optimiser un critère mathématique.

Nous avons effectué des premières simulations en fixant les limites géographiques afin de comprendre les phénomènes mis en jeu et de prendre en main le simulateur.

Lors de notre simulation manuelle, nous définissons les limites géographiques suivantes exprimées en mètres : [1500 ; 2500 ; 4000 ; 5500 ; 7500]. La représentation de cette répartition est visible sur la figure 4.17.

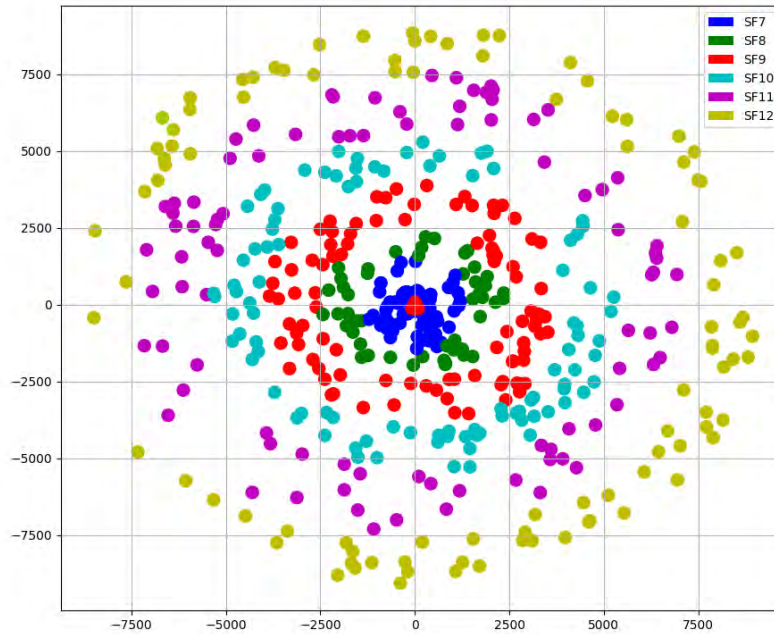


FIGURE 4.17 – Répartition des nœuds selon les spreading factors

4.2.9 Analyse du taux de livraison des trames (FDR)

Une fois la simulation effectuée, l'analyse peut être observée en une seule dimension. En effet, comme nous plaçons l'analyse au niveau de la gateway, la distance à prendre en considération est bien celle entre la gateway et un nœud donné. C'est d'autant plus vrai que nous étudions les collisions au niveau de la gateway, ainsi deux nœuds vont avoir des caractéristiques voisines lorsqu'ils sont à la même distance de la gateway. De cette manière, nous pouvons ramener le problème à une dimension afin de faire apparaître ce voisinage et de simplifier la représentation.

La figure 4.18 représente l'analyse du taux de livraison des trames en fonction de la distance avec la gateway. Les traits rouges verticaux sont les limites à partir desquelles les nœuds changent de spreading factor.

4.2.9.1 Réjection des trames de spreading factors différents

Il est aisé de remarquer que les nœuds qui évoluent avec un même spreading factor ont une corrélation forte de leur taux de livraison des trames. Cette propriété est issue de la très faible perturbation entre des trames à des spreading factors différents.

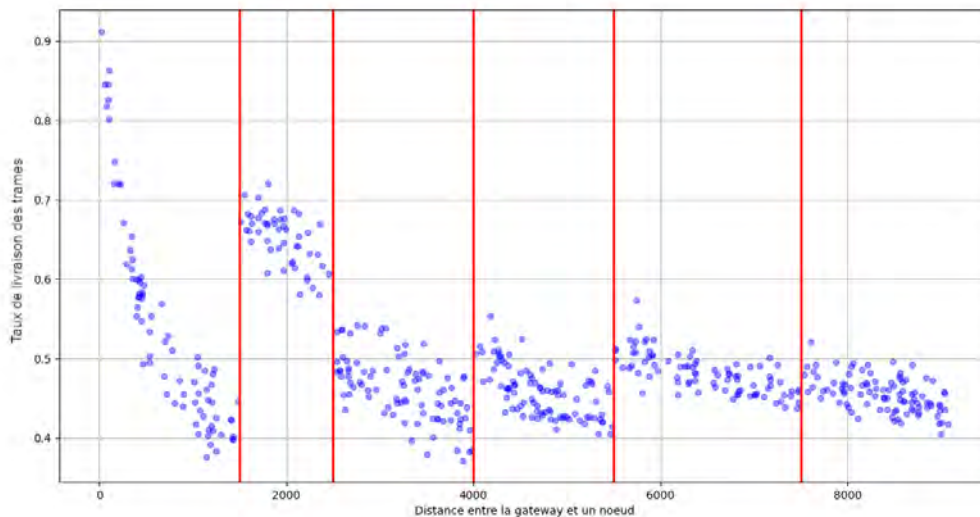


FIGURE 4.18 – Représentation du taux de livraison des messages par nœuds

4.2.9.2 Influence des trames de spreading factors égaux

Entre deux limites géographiques, les nœuds sont configurés avec un même spreading factor. C'est ce que nous définissons comme une bande iso-sf. Dans cette bande, les nœuds sont concurrents pour l'accès au médium.

4.2.9.3 Cas particulier de la première bande iso-sf

La première bande iso-sf, correspond à l'ensemble des nœuds les plus proches de la gateway. Ces nœuds sont sur une bande qui présente une tendance différente des autres bandes. Sa distribution en corne montre une disparité importante dans la qualité de service entre les différents nœuds. Cet effet vient du fait que ces nœuds se situent dans la bande favorable définie dans la section 4.1.4.3. Au sein de cette bande favorable, ils ont un taux de livraison des trames bien supérieur aux autres bandes car ils ont un RSSI moyen beaucoup plus élevé ce qui limite fortement les pertes de trames à cause du fading sur le canal. Chaque nœud va de cette manière plus régulièrement rentrer en concurrence sur l'accès au médium au niveau de la gateway. La forte disparité de la qualité de service est principalement due au fait des écarts importants de RSSI entre les nœuds, ce qui va les exposer à l'effet de capture qui donne l'avantage au plus proche. Des écarts importants de plus de 6 dB vont être observés entre les nœuds ce qui va favoriser les nœuds les plus proches de la gateway car ils éblouissent les nœuds plus éloignés.

4.2.9.4 Influence de la variation d'une limite géographique

Lorsque l'on déplace une limite géographique sans bouger les autres, on transfère des nœuds d'une bande adjacente à l'autre. La conséquence de ce transfert est que la densité de nœuds diminue sur une bande et augmente sur l'autre. Ce changement de densité au niveau de l'accès au médium va augmenter

globalement le taux de livraison moyen des trames sur une bande et le diminuer sur l'autre.

Cette propriété est critique dans la phase d'optimisation car il semble difficile d'optimiser de façon manuelle le réseau à cause des corrélations fortes entre les bandes. L'espace de recherche est immense et il devient nécessaire de résoudre ce problème numériquement.

4.2.9.5 A la recherche de l'optimalité

A ce niveau de réflexion, il est nécessaire de se demander qu'est-ce qu'un réseau LoRa mono gateway optimal ou plutôt quel est le critère d'optimalité ?

Il y a deux critères qui s'affrontent, la recherche d'extensibilité en nombre de nœuds et la qualité de service fournie à chaque nœud. Dans un contexte où il n'existe pas, dans le protocole, de profils applicatifs permettant une optimisation différenciée, il faut considérer que le réseau doit essayer de garantir une qualité de service équivalente pour tous les nœuds.

La recherche d'extensibilité va provoquer dans ses limites une baisse de qualité de service et peut entraîner de fortes disparités si les nœuds ne sont pas répartis équitablement dans les différents canaux.

Il est nécessaire de définir une qualité de service minimale pour le réseau et d'augmenter le nombre de nœuds jusqu'aux limites d'extensibilité en veillant à respecter une bonne répartition des nœuds dans les canaux.

Dans les réseaux sans-fil particulièrement, où le taux d'erreur bit est assez élevé, la qualité de service est très liée au taux de livraison des trames. La retransmission d'une trame sur le médium est problématique car les réseaux LPWAN ne sont pas synchronisés au niveau MAC et pour les nœuds de la bande de jeux équitables, l'accès au médium peut déjà être compliqué même sans les problématiques d'accès au médium. Notre critère d'optimisation doit chercher à maximiser le taux de livraison des trames.

4.3 Optimisation du taux de livraison des trames (FDR)

4.3.1 Critère d'optimisation

L'optimisation de notre réseau doit chercher à maximiser le nombre de nœuds sur le canal tout en garantissant une QoS minimale. La notion de qualité de service peut être définie selon plusieurs critères, dans les cas des LPWAN, le taux de livraison des trames doit être maximisé car la retransmission d'un message induit une forte latence due au respect du rapport cyclique.

Le taux de livraison est le rapport du nombre de messages délivrés sur le nombre total de messages émis. Ce taux est un pourcentage qui doit être calculé et maximisé pour chaque canal de transmission (combinaison d'un spreading factor et d'un canal fréquentiel).

Le critère d'optimisation peut se définir mathématiquement comme ceci :

$$L(x) = \max_j \left(\sum_i \min(FDR(SF_i, f_j)) \right) \quad (4.5)$$

- $L(x)$ fonction mathématique à optimiser
- n = ensemble des spreading factors (7 à 12)

- m = ensemble des canaux fréquentiels

La fonction mathématique $L(x)$ peut être calculée à la suite d’une simulation assez longue pour que les résultats statistiques convergent vers une tendance correcte pour les FDR.

La simulation est paramétrée avec un vecteur d’initialisation qui correspond aux limites de changement de spreading factor en mètres.

$$dist_{sf} = (1000, 2000, 3000, 4000, 5000) \quad (4.6)$$

L’objectif de notre programme d’optimisation est de déterminer le vecteur d’initialisation permettant d’optimiser le critère défini. C’est une optimisation multi-dimensionnelle dont les dimensions sont fortement corrélées.

4.3.2 Méthode d’optimisation CMA

Nous avons expérimenté la recherche de la solution optimale par les méthodes CMA-ES. La méthode CMA-ES [70] pour ”Covariance Matrix Adaptation Evolution Strategies” qui donne en français ”Stratégies d’évolution avec adaptation de matrice de covariance”, est une méthode qui repose sur l’adaptation de la matrice de covariance de la distribution multi-normale utilisée lors de l’optimisation.

L’idée étant d’évaluer à partir d’un vecteur d’initialisation, différentes petites variantes de valeurs autour de ce point de fonctionnement et d’en déduire après analyse statistique la valeur du vecteur pour le pas suivant.

4.3.2.1 Mise en échec de la méthode CMA

Afin de converger plus rapidement vers une solution à notre problème d’optimisation, nous avons cherché une méthode plus adaptée à notre cas d’usage.

En effet, en partant d’un vecteur d’initialisation moyen, il est difficile de converger vers une solution car quand on déplace une limite, on optimise une bande mais on dégrade la bande adjacente. Si on représente cette fonction en 3D, cela revient à parcourir un désert bosselé afin de trouver une crevasse. Il a donc fallu trouver une méthodologie différente d’aborder le problème.

4.3.3 Proposition d’une nouvelle méthode d’optimisation par bandes successives

4.3.3.1 Méthode d’optimisation par bandes successives

Notre critère d’optimisation est de chercher à trouver la limite en nombre de nœuds dans une bande de telle sorte à ce que la qualité de service passe au dessus d’un seuil minimal. Il faut pour cela partir d’une bande avec un maximum de nœuds et réduire progressivement le nombre de nœuds jusqu’à que la QoS minimale dans la bande passe au dessus du seuil. À partir de ce moment-là, il ne faut plus toucher cette limite et réaliser la même méthode avec la limite suivante. Ce processus itératif peut être exécuté jusqu’à la dernière bande qui va contenir le reste des nœuds. Suivant le nombre de nœuds restant, cette bande peut avoir une QoS meilleure ou moins bien que les autres bandes.

4.3.3.2 Situation initiale

Lors d'un processus d'optimisation, il est nécessaire de déterminer un état initial du système que l'on cherche à optimiser. Cet état initial peut être proche d'une solution intuitive afin de converger rapidement ou très loin d'une solution optimale pour observer la convergence.

Dans le cadre de ces travaux, nous avons décidé de partir d'un état initial qui n'est avec certitude pas optimal afin d'étudier la convergence de notre critère d'optimalité.

Lorsque l'on cherche à répartir des éléments dans plusieurs catégories, ici des canaux de transmission, une solution non optimale est de mettre tous les éléments dans la même catégorie. Il est possible ensuite de définir et d'exécuter une procédure ou algorithme permettant pas après pas de répartir les éléments dans les différentes catégories.

L'état initial de notre optimisation consiste ainsi à définir les limites géographiques au plus loin de l'antenne de sorte à ce que tous les nœuds à portée de l'antenne soit configurés au spreading factor 7. Cet état initial peut être visualisé sur la figure 4.19.

Nous pouvons vérifier que cet état n'est pas optimal car tous les nœuds étant configurés avec le même spreading factor, il y a beaucoup de collisions qui entraînent un taux de réception moyen des trames très mauvais.

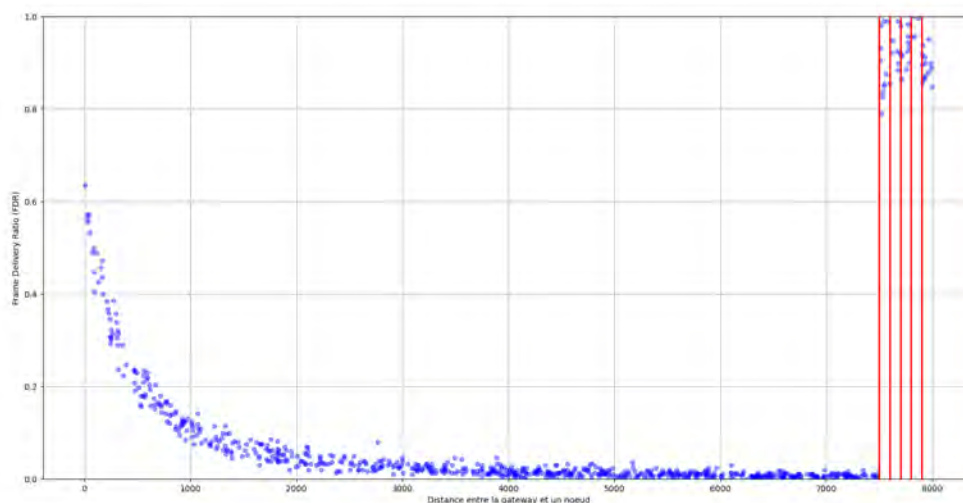


FIGURE 4.19 – État initial pour l'optimisation par bandes successives

4.3.3.3 Stratégie d'évolution lors de l'optimisation

Comme défini dans la partie 4.3.1, le critère d'optimalité tend à déterminer les zones géographiques de façon à avoir un taux de livraison minimal fixé autour de la gateway.

Dans un premier cas d'étude, nous fixons ce plancher à 40%. Afin de représenter cette limite, nous pouvons définir une limite horizontale à 40% sur nos graphiques (figure 4.20). L'objectif est maintenant

de déplacer les limites vers la gauche du graphique, en réduisant les limites circulaires autour de l'antenne de sorte que dans une bande donnée, le taux de livraison minimum soit supérieur au taux de 40%.

Le seuil de 40% est arbitraire et peut être ajusté suivant les contraintes du cahier des charges imposé, mais nous pensons que la valeur de 40% est classique d'un réseau LoRa où la distance provoque très régulièrement des pertes de trames par évanouissement.

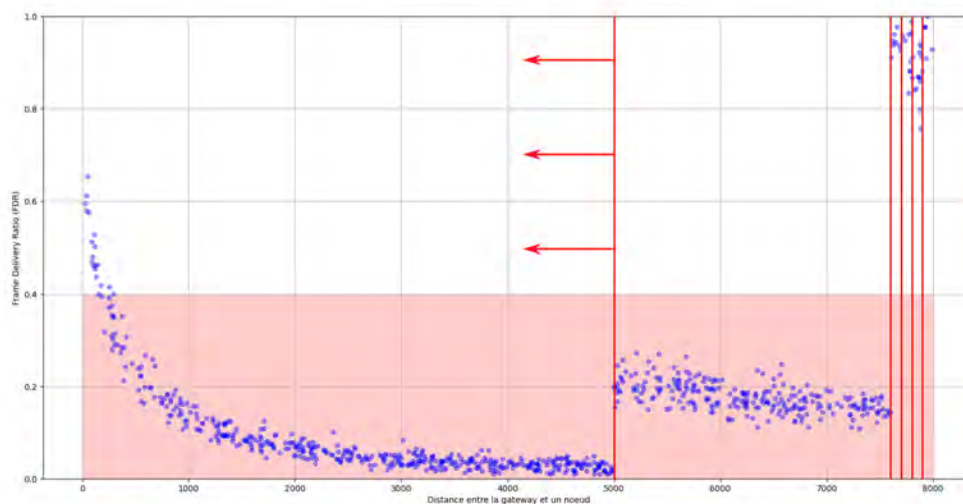


FIGURE 4.20 – Optimisation de la première bande par diminution du rayon de portée

Sur la figure 4.20, les flèches rouges représentent le sens de déplacement de la limite qui se rapproche de la gateway. Lorsque l'on réduit cette limite, on transfère des nœuds d'une bande à l'autre. Lors de la phase d'optimisation, on va progressivement augmenter le taux de livraison dans la bande adjacente gauche. Il faut définir un pas d'avancement entre deux simulations, c'est à dire le nombre de mètres que l'on retranche à la limite de portée. Ce pas d'avancement est arbitraire et va permettre de jouer sur le nombre de simulations nécessaires pour converger vers une solution. Plus le pas est grand et plus la précision finale sera faible mais plus la vitesse de convergence sera élevée.

Une fois que cette première limite optimale est atteinte, il ne faut plus bouger cette limite car elle fixe la répartition des nœuds dans la première bande.

4.3.3.4 Fin de l'optimisation

À la fin de la première étape d'optimisation, nous sommes dans une situation semblable à la situation initiale c'est à dire que tous les nœuds restants sont dans la deuxième bande et qu'en déplaçant la seconde limite, on va pouvoir optimiser la seconde bande sans modifier la première.

L'algorithme va consister à répéter l'étape intermédiaire successivement pour chaque bande jusqu'à la dernière. Nous aurons ainsi optimisé toutes les bandes et les nœuds au-delà de cette dernière limite seront tous configurés en spreading factor 12. Cette dernière bande peut ne pas respecter le critère d'optimalité suivant le nombre de nœuds restants.

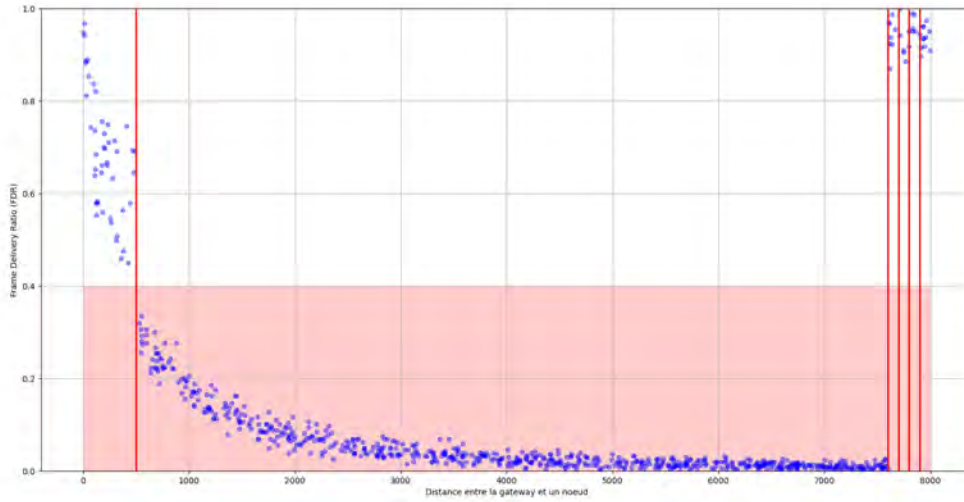


FIGURE 4.21 – Limite optimale pour laquelle le FDR minimal est de 40% dans la première bande

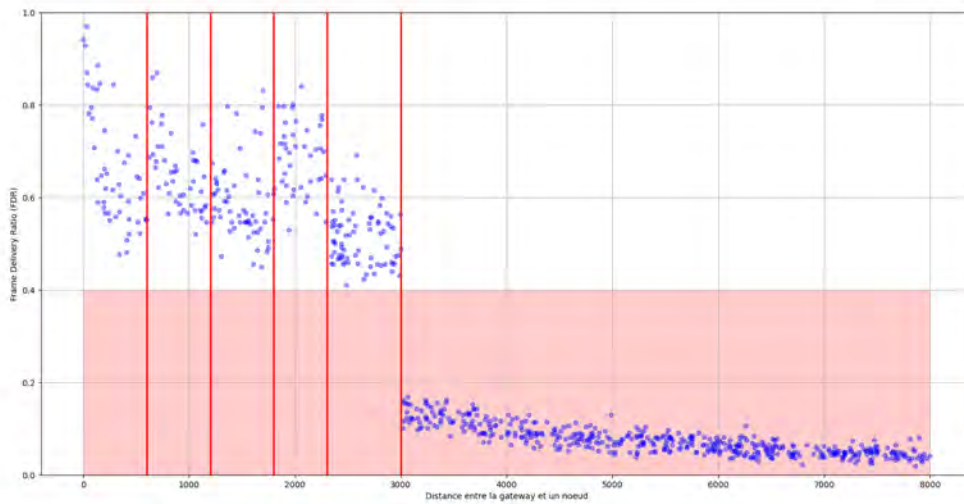


FIGURE 4.22 – État final à la fin de l'optimisation

Maintenant que nous sommes capables de simuler l'accès au médium dans un canal donné, nous pouvons étendre la simulation à l'ensemble des canaux disponibles sur une gateway.

4.4 Étude de l’extensibilité de l’accès au médium des réseaux LoRa

Les réseaux LoRa se basent sur les bandes ISM 868 MHz et principalement dans les deux sous-bandes à 868,0 MHz et à 868,7 MHz. A l’intérieur de ces deux sous-bandes, il est possible de définir 4 canaux fréquentiels de 125 kHz de bande passante. La capacité totale est ainsi de 4 canaux fréquentiels à 125 kHz limités à 1% de rapport cyclique et 4 canaux fréquentiels à 125 kHz limités à 0,1%. Ces deux sous-bandes sont limitées à une puissance maximale d’émission de 25 mW soit +14 dBm. Nous allons dans cette partie étudier l’extensibilité par canal fréquentiel puis l’extensibilité globale.

4.4.1 Limite d’extensibilité par canal

L’étude d’extensibilité consiste à simuler l’accès au médium en se limitant à un canal fréquentiel unique et de chercher la limite en nombre de nœuds avec une QoS minimale. Nous fixons toujours cette QoS minimale à 40% de taux de réception correcte des trames.

4.4.1.1 Canaux de rapport cyclique à 1%

Afin de définir la limite de capacité d’un canal radio LoRa à 125 kHz et avec un rapport cyclique de 1%, nous utilisons notre algorithme d’optimisation. En sortie de cette optimisation, nous pouvons avoir la taille de la cellule radio autour de la gateway en fonction du nombre de nœuds.

Les figures 4.23 et 4.24 permettent de visualiser la taille de la cellule pour 500 et 1000 nœuds. Le critère visé est un taux de réception des trames minimum de 40%. La taille de la cellule idéale varie bien proportionnellement au nombre de nœuds, ici elle passe de 5 km pour 500 nœuds à 2,8 km pour 1000 nœuds.

Afin de conclure sur la capacité d’un canal à 1%, il faut comptabiliser le nombre de nœuds dont le taux de livraison est supérieur à 40%. Pour la simulation avec 1000 nœuds la capacité du canal en nombre de nœuds est ainsi de 359 nœuds pour une cellule de 3 km.

4.4.1.2 Canaux de rapport cyclique à 0,1%

La même évaluation doit être menée pour un canal à rapport cyclique de 0,1%. Il est nécessaire de souligner que ces canaux vont instinctivement pouvoir avoir une capacité plus grande mais chaque nœud va émettre 10 fois moins de messages! La comparaison entre un canal à 1% et un canal à 0,1% doit toujours tenir compte de cet aspect.

La figure 4.25 montre que la limite de FDR de 40% permet d’avoir une capacité de 2183 nœuds pour une taille de cellule de 3 km.

4.4.2 Limite d’extensibilité globale

Après l’étude de la capacité d’un canal à 1% et d’un canal à 0,1% il est possible de conclure sur l’extensibilité globale d’une gateway LoRa en terme de nombre de nœuds.

Comme le montre la figure 3.2, le composant SX1301 qui est intégré dans les gateways LoRa et qui gère la partie numérique du frontend radio, est composé de 8 démodulateurs multi-spreading factor. Le

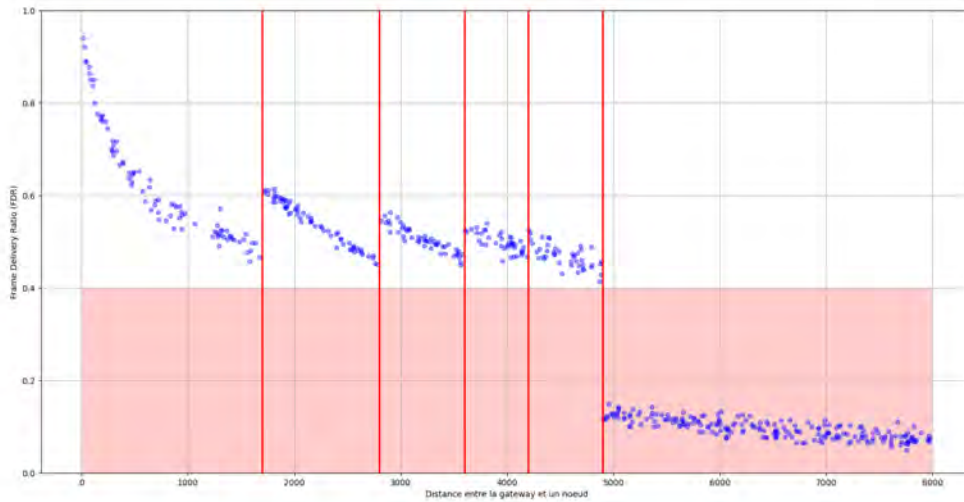


FIGURE 4.23 – Optimisation pour 500 nœuds

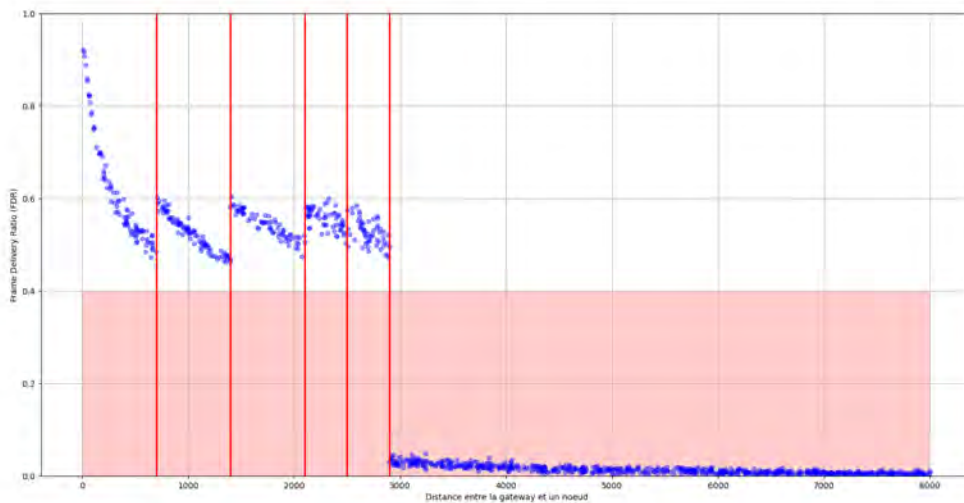


FIGURE 4.24 – Optimisation pour 1000 nœuds

SX1301 fonctionne avec deux frontend analogiques de type SX125x qui numérisent chacun une bande de fréquence fixée. Chaque SX125x numérise une bande dans laquelle il est possible de placer un ou plusieurs démodulateurs sur une fréquence intermédiaire. L'idée est de fixer un frontend sur un canal donné et de répartir 4 démodulateurs à l'intérieur de cette bande pour réaliser des sous-canaux. Il est possible de

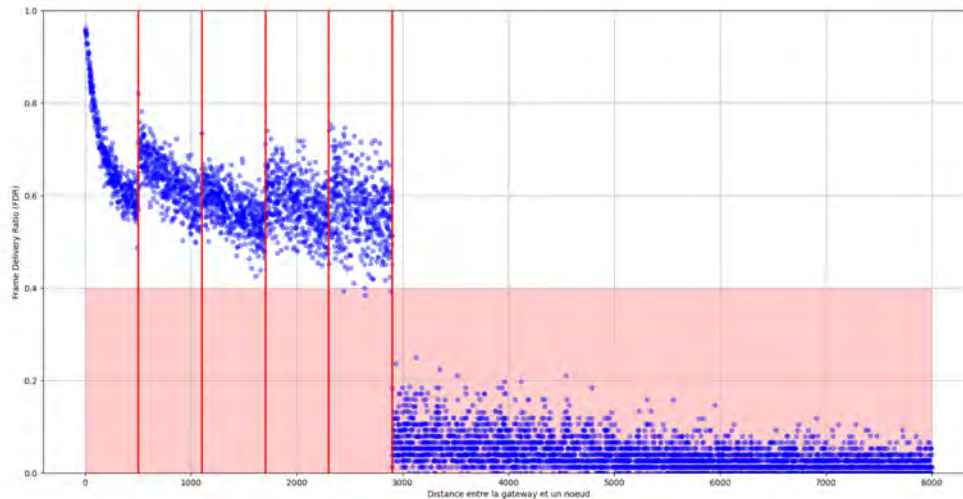


FIGURE 4.25 – Optimisation pour 6000 nœuds

placer un frontend à 433 MHz et un à 868 MHz mais dans le cadre des réseaux LoRaWAN, deux frontends à 868 MHz sont paramétrés. En accord avec la législation des bandes ISM (figure 2.2), il est possible de numériser les deux premières bandes et de les diviser en 4 sous canaux fréquentiels. Le résultat est ainsi de 4 sous canaux à 1% et de 4 sous canaux à 0,1%.

Si la répartition fréquentielle des nœuds est homogène alors il est possible de conclure qu'une gateway LoRa a une extensibilité globale de 1436 nœuds à 1% avec une portée de 3 km et un FDR minimal de 40% et 8732 nœuds à 0,1% avec une portée de 3 km et un FDR minimal de 40%.

Attention, les canaux avec un rapport cyclique de 0,1% correspondent à des cas d'usages très limités en nombre de messages!

Il est important de noter que les nœuds à spreading factor 12 sont sacrifiés car nous n'avons pas de leviers disponibles pour les optimiser. Ils constituent la longue traîne jusqu'à la limite de portée. Un réseau multi-gateway permettra via une couverture contiguë d'avoir des leviers pour optimiser ces nœuds.

4.5 Algorithme de réglage automatique des paramètres de la couche physique

4.5.1 Automatic Data Rate de LoRaWAN

Dans les spécifications de LoRaWAN et de son empilement protocolaire déployé en grande majorité par les opérateurs de part le monde, un mécanisme d'adaptation automatique du spreading factor et de la puissance d'émission est décrit. Ces spécifications définissent les mécanismes protocolaires mais restent très vagues sur l'algorithme de décision côté nœud ou côté serveur. Le papier "Adaptive configuration of LoRa networks for dense IoT deployments" [71] publié en 2018 dans "IEEE/IFIP Network Operations

and Management Symposium” montre la simulation du mécanisme de base de l’ADR via OMNeT++.

4.5.1.1 Activation de l’ADR

Dans le header LoRaWAN un bit ADR est présent et permet d’indiquer à l’agent à l’autre bout de la transmission sa volonté de rentrer dans une phase de négociation des paramètres physiques.

Côté nœud : La couche applicative d’un device est libre de choisir un spreading factor et une puissance d’émission afin d’émettre des messages à destination de la gateway (uplink). Lorsque la couche applicative veut déléguer cette tâche au serveur de réseau elle peut activer via une commande MAC le mode ADR.

Côté serveur de réseau : Le serveur de réseau peut, si il est en position de définir des paramètres physiques pour le device, activer le bit ADR dans un message à destination du device (downlink). L’activation de ce bit est une proposition à rentrer en mode ADR que le device peut accepter ou refuser.

4.5.1.2 Commandes MAC de réglage des paramètres de la couche physique

La couche MAC LoRaWAN définit des commandes de niveau MAC permettant au serveur de proposer un reparamétrage de la couche physique au device.

LinkADRReq : Cette commande est émise par le serveur de réseau à destination du device avec la proposition d’un spreading factor, d’une puissance d’émission, d’une liste de canaux fréquentiels à utiliser ainsi que d’une période de retransmission en cas d’échecs.

LinkADRAns : Cette commande permet au device d’acquiescer les différents paramètres pour indiquer au serveur qu’il accepte le reparamétrage.

4.5.1.3 Nœuds mobiles

Lorsque un nœud est mobile, il est difficile d’établir des statistiques précises car l’écart type de la puissance du signal reçu va varier énormément suivant les obstacles rencontrés et la diversité des multi-chemins. Il est dans ce cas non recommandé de demander au serveur de réseau de proposer un reparamétrage de la couche physique qui risque d’être très erroné.

4.5.2 Les trois zones comportementales

À l’issue de la phase de simulation, nous observons sur la figure 4.23 trois zones distinctes, une zone proche de l’antenne avec les devices en spreading factor 7, une zone éloignée avec le reste des devices en spreading factor 12 et entre les deux une zone intermédiaire : les spreading factors de 8 à 11.

4.5.2.1 Zone proche

La zone proche de la gateway rassemble les nœuds qui sont les mieux exposés et qui vont pouvoir émettre avec un spreading factor de 7. C’est une zone où les inégalités sont fortes à cause de la décroissance forte des pertes de propagation. Afin d’optimiser cette bande il est nécessaire de limiter son rayon et de ne pas faire chuter le taux de livraison en dessous d’un seuil acceptable pour les nœuds en limite de zone. Il va également être judicieux d’avoir un ajustement de la puissance d’émission du nœud très fin pour tendre à réduire les inégalités d’accès au médium par effet de capture.

4.5.2.2 Zone éloignée

La zone éloignée regroupe l'ensemble des nœuds qui sont tellement éloignés de la gateway que l'on ne peut les recevoir qu'avec un spreading factor de 12. Le rayon de cette zone est difficilement réglable car nous ne pouvons via un algorithme définir qu'une des deux limites de la bande. Dans un contexte mono-gateway, il faut prêter assez peu attention à cette bande qui sera surtout importante lors d'un déploiement de multiples gateways afin de couvrir une zone plus grande. En effet, après avoir défini la capacité d'une cellule LoRa, il est possible de déployer un réseau de gateways afin que les cellules s'entrelacent pour couvrir une zone géographique.

4.5.2.3 Zone intermédiaire

Le gros enjeu de l'extensibilité des réseaux LoRa réside dans cette zone intermédiaire. La faible variation des pertes de propagation place les nœuds dans des conditions très proches. Nous remarquons sur la figure 4.23 que les bandes optimales sont très étroites ce qui ne facilite pas l'estimation. L'algorithme de paramétrage automatique va devoir trouver une stratégie différente, non basée sur la puissance du signal reçu pour cette bande.

4.5.2.4 Interprétation graphique des zones comportementales

L'existence de ces zones comportementales est observable lorsque l'on confronte le modèle du bilan de liaison aux taux de livraison simulés. La figure 4.26 indique clairement les trois zones : proche, intermédiaire et éloignée.

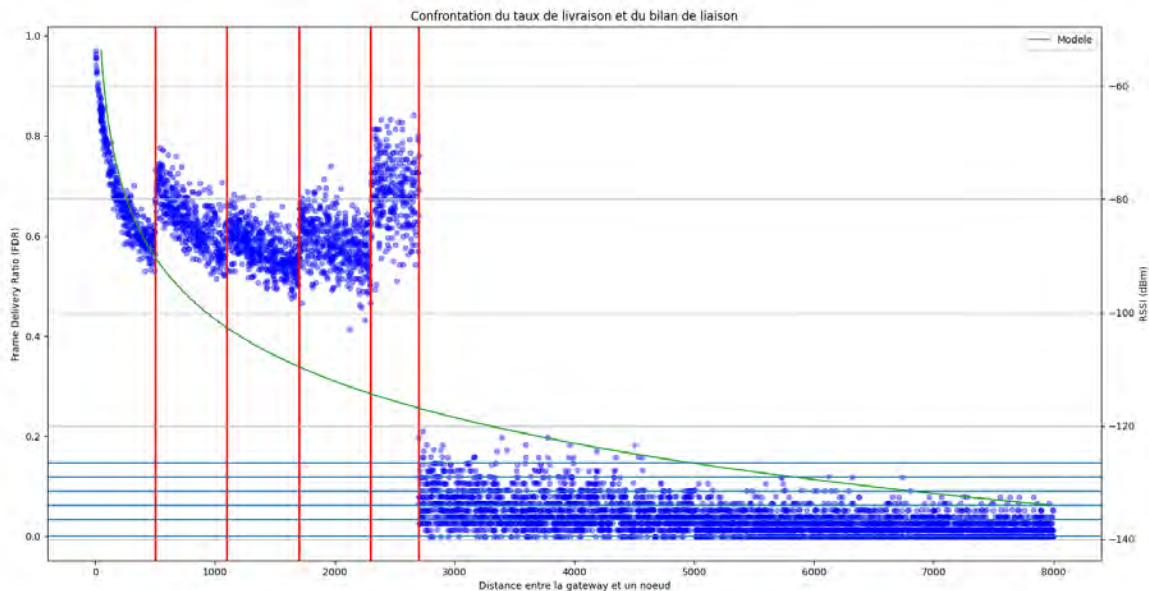


FIGURE 4.26 – Confrontation du taux de livraison et du bilan de liaison

Les variations de taux de livraison des trames représentent directement une variation de la perte de trames.

La première cause de perte de trames est une atténuation tellement forte qu’au niveau du récepteur la puissance reçue est en dessous du seuil de sensibilité. Cette atténuation peut être due à un obstacle temporaire comme à de mauvaises conditions de propagation à cause de la distance ou du positionnement en intérieur ou sous terre. Cette cause est donc fonction du bilan de liaison en fonction de la distance.

La deuxième cause de perte de trames concerne les collisions. Plus le nombre de nœuds présents sur le médium est élevé, plus il y a de collisions lorsque que plusieurs émetteurs concurrents émettent simultanément. Cette cause est donc fonction du taux d’occupation du médium.

4.5.3 Proposition d’algorithme

Les spécifications de LoRaWAN étant assez floues sur l’algorithme de choix pour la proposition du paramétrage d’un nœud, nous proposons en se basant sur nos simulations et les observations associées, un nouvel algorithme décisionnel.

4.5.3.1 Les leviers technologiques disponibles

Avant de définir un algorithme, il est utile de se replacer dans un environnement concret et de lister les leviers technologiques disponibles.

Périodicité d’émission : Dans les réseaux LoRa, le rapport cyclique imposé par la législation ISM contraint fortement la période d’émission entre deux trames. Un algorithme de paramétrage automatique va devoir envoyer des messages sur le médium pour le sonder afin de prendre sa décision, mais il est à noter que l’algorithme ne doit nécessiter que très peu de messages pour être applicativement viable.

RSSI : La puissance du signal reçu est un paramètre précieux pour positionner un nœud dans une bande. Cependant un nœud ne peut pas avoir d’informations directes après l’émission pour savoir à quelle puissance sa trame a été reçue. Cette information doit être transmise lors d’un message retour de la gateway vers le nœud. Comme les trames descendantes sont encore plus précieuses, il faut limiter au strict minimum l’utilisation de ces messages descendants par l’algorithme.

Puissance d’émission : Lorsqu’un nœud ou une gateway reçoit une trame, elle ne peut pas savoir à quelle puissance d’émission elle a été émise. C’est la même problématique qu’avec le RSSI, l’indication doit être passée dans un second message ou mieux, être directement transportée dans un champ dédié à l’intérieur de la trame.

Rapport signal sur bruit : Le rapport signal sur bruit permet d’informer sur l’état du plancher de bruit par rapport aux pertes de propagation. Les seuils de sensibilité sont fixés par rapport à un plancher de bruit donné, si le bruit est plus important en réalité, les seuils de sensibilité réels vont remonter ce qui va entraîner plus de pertes de trames. Notre algorithme peut se servir de cet indicateur pour réajuster la décision.

Taux de livraison : Un taux de livraison sur un très faible nombre de messages peut être calculé. L’information qu’il va délivrer sera très grossière mais peut permettre de prendre une décision sur plusieurs trames afin de faire une moyenne et de limiter l’impact d’un interférent passager sur la prise de décision. Cette analyse statistique peut aussi être effectuée en temps-réel à la suite du paramétrage pour suivre la pertinence de la décision dans le temps.

4.5.3.2 Sondage du médium

Un sondage efficace du médium doit permettre en quelques messages de trouver le réglage optimal pour les nœuds les plus évidents.

Cas du spreading factor 7 : Les nœuds les plus proches de la gateway peuvent via la mesure du RSSI rapidement se positionner. En effet, si ces nœuds émettent deux ou trois trames balises à spreading factor 7 et que la gateway reçoit ces trames avec un RSSI supérieur à -100 dBm, elle peut automatiquement les paramétrer afin de rester avec ce réglage.

Cas du spreading factor 12 : Les nœuds les plus éloignés vont également pouvoir se positionner rapidement. Via l'émission de quelques trames balises à spreading factor 12 et si la gateway reçoit ces trames avec un RSSI moyen inférieur à -125 dBm, elle peut les paramétrer afin de rester avec ce réglage.

Cas de la bande intermédiaire : Pour la bande intermédiaire, la décision est bien plus difficile. Les bandes optimales sont très étroites ce qui donne des limites en terme de RSSI qui sont trop peu éloignées en comparaison de l'écart type des RSSI reçus. Plusieurs approches peuvent être étudiées afin de répartir au mieux les nœuds de façon homogène au sein des spreading factor de 8 à 11. Nous proposons une approche aléatoire mais il serait possible d'imaginer une approche par profils applicatifs.

4.5.3.3 Approche aléatoire pour la bande intermédiaire

Le paramétrage optimal des nœuds étant très difficile dans cette bande, une stratégie facile à mettre en œuvre consiste à répartir aléatoirement les nœuds pour les spreading factors de 8 à 11.

Plusieurs inconvénients sont liés à cette stratégie :

- **Débits différents :** Il est important de souligner que lors d'un changement de spreading factor le temps d'émission et le temps d'attente successif à une émission vont être très différents. Il est possible de pondérer les différents spreading factors afin de favoriser les plus faibles.
- **Performances plus faibles que l'optimalité :** En autorisant les nœuds à se déplacer entre plusieurs spreading factors, il arrive régulièrement qu'une majorité de nœuds se retrouvent au même spreading factor en laissant les autres libres. Si le générateur aléatoire est de bonne qualité, ce cas arrive statistiquement assez peu mais il arrive tout de même ce qui permet d'affirmer que les performances vont être au mieux équivalentes à une répartition fixe.

4.5.3.4 Proposition de l'algorithme de décision

L'algorithme de décision pour le placement initial des nœuds est proposé dans les programmes 1 et 2 sous forme de pseudo-code.

La fonction *Principale* est appelée lors de l'émission d'une trame. C'est elle qui va décider du paramétrage à appliquer en commençant par une phase de détection de zone via l'appel à la fonction *Détection de zone* puis par une émission dans le paramétrage correspondant à la zone déterminée.

Programme 1: Sondage du canal pour déterminer la zone

Variable: Zone = Intermédiaire

```
1 Fonction Sondage de la zone proche
2   Pour essais ← 2 à 1 Faire
3     Émission d'une trame sonde en SF7;
4     Tant que Réception de trame ou timeout Faire
5       Attente;
6     FinTantQue
7     Si Réception de trame Alors
8       Zone = Proche;
9       Retourner
10    FinSi
11  FinPour
12 FinFonction
13 Fonction Sondage de la zone éloignée
14   Pour essais ← 2 à 1 Faire
15     Émission d'une trame sonde en SF12;
16     Tant que Réception de trame ou timeout Faire
17       Attente;
18     FinTantQue
19     Si Réception de trame Alors
20       Zone = Éloignée;
21       Retourner
22    FinSi
23  FinPour
24 FinFonction
25 Fonction Détection de la zone
26   Appel de la fonction Sondage de la zone proche;
27   Appel de la fonction Sondage de la zone éloignée;
28   Si Zone = Aucune Alors
29     Zone = Intermédiaire;
30   FinSi
31   Retourner Zone
32 FinFonction
```

Programme 2: Reparamétrage automatique de la couche physique

```
1 Fonction Répartition aléatoire dans la zone intermédiaire
   | Variable: AleaSF
2   | AleaSF = AleatoireUniforme(entre 8 et 11);
3   | Transmission à SpreadingFactor AleaSF;
4 FinFonction
5 Fonction Principale
   | Variable: Zone
6   | Zone = Appel de la fonction Détection de la zone;
7   | Suivant Zone Faire
8     | Cas où Zone = Proche Faire
9     | | Transmission à SpreadingFactor 7;
10    | FinCas
11    | Cas où Zone = Intermédiaire Faire
12    | | Appel de la fonction Répartition aléatoire dans la zone intermédiaire;
13    | FinCas
14    | Cas où Zone = Éloignée Faire
15    | | Transmission à SpreadingFactor 12;
16    | FinCas
17  | FinAlternative
18 FinFonction
```

4.5.3.5 Implémentation du protocole dans LoRaWAN

Il est important de vérifier l'implémentabilité de cette nouvelle stratégie dans LoRaWAN afin de pouvoir l'évaluer sur les réseaux LoRa les plus déployés actuellement. Le levier défini dans la norme qui permet d'implémenter ce mécanisme est l'ADR (Adaptative Data Rate) qui nous le rappelons décrit l'encodage des trames mais pas la stratégie à adopter.

La couche MAC LoRaWAN v1.1 définit plusieurs trames de commande MAC. Parmi ces commandes, certaines permettent d'implémenter notre stratégie :

- LinkADRReq (0x03) : le serveur de réseau demande à l'objet de changer de paramétrage suite à une décision prise par l'algorithme
- LinkADRAns (0x03) : l'objet acquitte le serveur de réseau de son reparamétrage
- LinkCheckReq (0x02) : l'objet demande au serveur de réseau de valider la connectivité
- LinkCheckAns (0x02) : le serveur de réseau acquitte la connectivité et transmet des informations sur la qualité de la réception

Via ces commandes MAC, nous proposons d'utiliser des requêtes LinkCheckReq au démarrage du nœud pour l'émission des trames balises puis lors de la prise de décision du reparamétrage, un échange de type LinkADRReq pour appliquer les nouveaux paramètres de transmission.

4.5.4 Conclusion et perspectives de ces travaux

L'implémentation et l'évaluation de ce protocole est à l'heure actuelle difficilement envisageable sur le *testbed* car nous n'avons pas encore assez déployé de nœuds par rapport aux passerelles. La prochaine

étape de ce projet sera donc de développer une architecture de *testbed* pour contrôler une flotte d'objets dans la ville ou dans un quartier donné.

Nous avons dans le cadre de cette thèse, présenté une première piste d'architecture pour ce *testbed* d'objets. Nous pouvons faire sortir notre *testbed* d'objets du laboratoire vers la ville en déployant dans un quartier des contrôleurs afin de reprogrammer et de faire l'abstraction réseau des nœuds via MQTT. Le positionnement de ces nœuds doit être plus proche du sol afin d'être dans des conditions d'usages classiques.

Dans ce même quartier, nous déploierons plus finement notre testbed de passerelles sur les toits afin de couvrir la zone de façon contiguë et de pouvoir mener des scénarios d'expérimentation multi-passerelles et multi-nœuds en scénario urbain. L'ensemble des contrôleurs et des passerelles seront raccordés au réseau de fibre de l'opérateur de la collectivité afin de les connecter au cœur de réseau IoT de notre laboratoire.

Notre équipe de recherche démarre le projet "Vol de Nuit" avec la métropole de Toulouse afin d'imaginer, de concevoir et de prototyper ce type de *testbed* à l'échelle d'un quartier permettant à des laboratoires et des entreprises d'avoir accès à une plateforme de mise au point de leurs systèmes au cœur de la ville.

Conclusion générale et perspectives de recherche

Conclusion générale

Testbed LoRa Métropolitain

L'objectif de notre équipe de recherche est d'imaginer et de créer les logiciels et matériels nécessaires à l'expérimentation en conditions réelles de réseaux et protocoles. Ces expérimentations nécessitent une rigueur dans le choix des outils, des architectures et des méthodologies afin de viser une reproductibilité maximale. En effet dans le cadre de réseaux sans fil, il est délicat de pouvoir garantir une reproductibilité stricte car il est très compliqué de maîtriser le médium de transmission. Cependant, un descriptif précis, de l'environnement d'étude, des matériels mis en œuvre, d'une publication ouverte et libre des jeux de données et du matériel expérimental permet de fournir un bon taux de reproductibilité.

Dans le cadre de cette thèse, nous avons contribué à cette discipline en créant un *testbed* qui sort du laboratoire pour s'étendre sur toute la métropole. Le passage à large échelle est principalement dû au caractère longue distance des communications LoRa (plusieurs kilomètres) qui nécessitent d'étendre la zone d'étude. Ce déploiement est établi en zone non maîtrisée, ce qui ne garantit pas une reproductibilité forte mais permet une évaluation des protocoles au plus proche des applications finales.

Les contributions architecturales les plus importantes dans ce *testbed* se situent au niveau de la couche de transport des messages, nous avons travaillé à ce niveau afin de rendre notre réseau interopérable. Nous avons montré à travers plusieurs expériences, des paradigmes et des méthodologies pour imaginer, prototyper et éprouver des nouveaux protocoles. De plus cette méthodologie est transposable à d'autres technologies, comme nous avons pu le montrer avec l'Ultra Wide Band via le second *testbed* de l'équipe : LoCuRa4IoT [72].

La réussite de ce *testbed* repose sur l'implication d'associations, d'enseignants et de chercheurs qui grâce à la diffusion ouverte et libre des informations et des outils, ont permis de développer largement ce réseau et de l'utiliser pour la pédagogie et la recherche.

Architectures interopérables pour l'Internet des Objets

L'interopérabilité est un sujet majeur de l'Internet des Objets. Nous avons pu montrer que les architectures mises en œuvre, sont complexes, constituées de nombreux agents matériels et logiciels qui dialoguent ensemble. Avec l'objectif de réduire la complexité de développement et de maintien en production mais également de pérenniser le déploiement de ces réseaux dans le temps, il est strictement nécessaire de mettre

en place dès le départ une solution d'interopérabilité.

Différentes solutions existent, nous avons étudié plus spécifiquement le protocole MQTT qui se développe fortement depuis quelques années dans cet objectif, démontrant que notre choix en début de thèse était judicieux. Nous avons montré les forces de ce protocole pour casser les relations client/serveur entre les agents et permettre des architectures plus dynamiques via le mécanisme de multicast. Cette architecture permet de casser les silos applicatifs et le chaînage fragile de nombreuses applications en se focalisant sur une architecture centralisée beaucoup plus simple et résiliente.

Nous avons déployé ce paradigme dans le cadre de la maison intelligente de l'IUT de Blagnac, du *testbed* LoCuRa4IoT et dans notre *testbed* LoRa métropolitain. Cette architecture est devenue au fur et à mesure de cette thèse omniprésente au laboratoire. Nous avons développé de nombreux outils pour gérer, simuler, collecter, stocker, visualiser et analyser les données, les réseaux et les protocoles. Durant la période de cette thèse, nous avons fait évoluer ces différents outils ce qui a permis de montrer la facilité d'évolutivité en respectant la définition et l'implémentation de formats des données qui circulent sur MQTT.

Nous avons ouvert la voie à des travaux sur une autre forme d'interopérabilité qui permettrait le partage d'informations en temps réel entre différentes entités via le respect de politiques de partage. Un broker MQTT est un cœur foisonnant où circule l'ensemble des flux de données d'un ensemble d'applications. Cet élément est très critique et se doit d'être maîtrisé au niveau politique par l'entité qui est responsable du réseau. Lorsque plusieurs entités cherchent à partager un sous ensemble de ces flux d'informations, il devient nécessaire de définir, des règles de partage, des limites, des conventions sur le format des données et sur leur transmission.

Extensibilité des réseaux LoRa

Nous avons profité du développement de notre *testbed* urbain et de l'étude de sa couverture pour contribuer à l'étude de la extensibilité et de la portée des réseaux LoRa. Ces réseaux se sont beaucoup développés durant la période de cette thèse, avec le développement au niveau français de réseaux opérateurs. Le développement rapide de ces réseaux principalement par les industriels a amené très peu de publications scientifiques sur la thématique du déploiement et de l'étude de couverture de ces réseaux. Notre volonté a été de chercher à regagner du terrain au niveau scientifique via le développement du *testbed* métropolitain afin de mener des études et des expérimentations.

Nous avons utilisé le *testbed* pour valider un modèle de propagation des réseaux LoRa en menant une campagne de mesure. Le relevé d'un ensemble de points de mesure à différentes distances et sous différents angles a permis de valider que le modèle de Okumura Hata était applicable. Une fois le modèle paramétré, nous l'avons implémenté et cette implémentation a fourni un socle pour le développement d'un simulateur de réseaux LoRa à événements discrets.

L'objectif du développement d'un simulateur LoRa était de pouvoir évaluer par la simulation les limites de capacité d'une station de base LoRa dans un contexte urbain. Cette évaluation est trop complexe pour être mise en œuvre en pratique car elle nécessiterait le déploiement de milliers de nœuds. Nous avons donc validé le modèle unitaire d'un nœud ainsi que les propriétés de gestion des collisions et des éblouissements puis nous avons testé la montée en charge du réseau en nombre de nœuds.

Perspectives de recherche

Mutualisation du service de collecte

Les concepts et architectures étudiés dans le cadre de cette thèse remettent en question le métier d'opérateur pour l'Internet Des Objets. Le métier d'un opérateur LPWAN est de fournir un service d'émission et de réception de trames au travers de stations de base. Son métier est de fournir une couverture maximale du territoire et une disponibilité forte des services de collecte. Un opérateur doit être en mesure d'intégrer des technologies de modulation et de traitement du signal toujours plus performantes de façon à collecter de plus en plus de trames et de plus en plus loin. Cet opérateur peut collaborer avec des entreprises et/ou des laboratoires afin de créer et d'intégrer ces modulations et ces protocoles. Dès lors, d'autres entreprises peuvent venir se greffer sur les opérateurs que l'on désigne ici comme «opérateurs de collecte». Le métier de ces entreprises est d'implémenter les couches hautes du modèle OSI, jusqu'à la transmission des données au client final. Comme un objet est capable de transmettre suivant plusieurs couches physiques, il adapte sa transmission à son état, son cas d'usage etc. Ces entreprises garantissent que ces changements de couche physique soient transparents pour le client final.

Les travaux issus de cette thèse proposent des méthodologies de mutualisation des opérateurs de collecte ; c'est à dire de ne plus être obligé de déployer un réseau national ou mondial mais de pouvoir fédérer son réseau urbain avec ceux d'autres opérateurs.

Passerelle générique pour les réseaux LPWAN

Les réseaux LPWAN actuels nécessitent de changer l'ensemble du matériel afin d'être mis à jour lors de la sortie d'une nouvelle modulation. Notre proposition est de créer une passerelle générique dont la couche physique pourrait être mise à jour à distance sans changement de matériel. Pour cela, les travaux sur la radio logicielle (SDR) permettent d'imaginer que le front-end radio soit générique et que le traitement du signal et l'implémentation des modulateurs et démodulateurs soient logiciels. De cette façon, un opérateur de collecte peut mettre à jour les logiciels embarqués sur ses passerelles de façon à s'adapter aux nouvelles améliorations en terme de traitement du signal.

Il est également possible de totalement déporter ces logiciels de traitement du signal au niveau de serveurs via la transmission des échantillons (I et Q) issus du front-end radio via un réseau IP. Au niveau des serveurs, les logiciels de traitement du signal peuvent moduler et démoduler les signaux de façon centralisée indépendamment de l'ensemble des passerelles déployées sur le réseau.

Protocoles de transport pour l'Internet des Objets

Lors des travaux sur l'interopérabilité dans le cadre de cette thèse, nous avons analysé l'importance de la couche transport dans les réseaux de l'Internet des Objets. Les infrastructures d'Internet sont de plus en plus complexes et deux concepts s'opposent, l'application du modèle existant d'Internet aux objets et la remise en cause de certains principes. Dans le cadre de cette thèse, notre positionnement a été modéré en proposant des solutions se basant sur le standard IP permettant d'être interopérable avec Internet tout en travaillant sur la couche transport pour ajouter de nouvelles fonctionnalités nécessaires aux objets.

L'idée est que la couche transport soit constituée d'une charge utile et de méta-données lui permettant de prendre des décisions sur le transport de cette charge utile. Les différents équipements et infrastructures peuvent lire et écrire ces méta-données afin de rendre un service de transport.

Réseaux LoRa fédérés

Un réseau LoRa métropolitain de recherche est complexe à déployer et à maintenir essentiellement parce qu'il nécessite de maîtriser un ensemble de points hauts sur lesquels déployer l'infrastructure radio. Dans une ville comme Toulouse, les équipes de recherche des différents laboratoires en réseaux et protocoles déploient des réseaux de quelques passerelles. Afin de fédérer ces différents acteurs, nous sommes en train de développer des mécanismes de mutualisation des infrastructures. De cette manière, via une interopérabilité au niveau de la couche transport, nous pouvons créer des réseaux LPWAN virtuels qui au niveau gestion du réseau reposent sur différentes entités.

L'avenir des réseaux LoRa réside dans cette capacité à mutualiser les déploiements et à partager les ressources physiques tout en garantissant la neutralité d'accès aux réseaux et aux données.

OperaMetrix et le déploiement de réseaux LoRa

Ces travaux de thèse ont abouti à la fondation de l'entreprise OperaMetrix dont l'ambition est de créer et d'exploiter des réseaux LoRa pour l'industrie et les collectivités.

La métropole de Toulouse a lancé le projet «Vol de nuit» afin de définir de façon expérimentale les besoins en connectivité de la ville de demain. De nombreuses technologies se développent pour des usages très différents et laissent présager d'un avenir multi-technologique où l'interopérabilité ne sera pas une option.

Bibliographie

- [1] Site web du consortium pour le bus CAN. <https://www.can-cia.org/can-knowledge/>.
- [2] Spécifications du protocole Modbus. <https://www.modbus.org/specs.php>.
- [3] Spécifications du protocole Profibus. <https://www.profibus.fr/fr/profibus>.
- [4] Spécifications du protocole LIN. <http://www.lin-subbus.org/>.
- [5] B. Warneke, M. Last, B. Liebowitz, and K. S. J. Pister. Smart dust : communicating with a cubic-millimeter computer. *Computer*, 34(1) :44–51, 2001.
- [6] Spécifications du protocole Bluetooth. <https://www.bluetooth.com/specifications/bluetooth-core-specification/>.
- [7] Site web de la ZigBee Alliance. <https://zigbeealliance.org/>.
- [8] Adrien Van Den Bossche. *Proposition of a new deterministic medium access method for time-constrained wireless personal area networks*. Theses, Université Toulouse le Mirail - Toulouse II, July 2007.
- [9] P. Thubert, C. Bormann, L. Toutain, and R. Cragie. Ipv6 over low-power wireless personal area network (6lowpan) routing header. RFC 8138, RFC Editor, April 2017.
- [10] Thierry Val. *Contribution à l'ingénierie des systèmes de communication sans fil*. PhD thesis, 07 2002.
- [11] Ultra Wideband (UWB) : quand Apple veut démocratiser la géolocalisation précise. [Lien vers la page web](#).
- [12] Brian Robertson, éditions LEDUC.S (2016). *La révolution Holacracy, le système de management des entreprises performantes*.
- [13] Andrea De Mauro, Marco Greco, and Michele Grimaldi. A formal definition of big data based on its essential features. *Library Review*, 65 :122–135, 03 2016.
- [14] Rob Hierons. Machine learning. tom m. mitchell. published by mcgraw-hill, maidenhead, u.k., international student edition, 1997. isbn : 0-07-115467-1, 414 pages. price : U.k. £22.99, soft cover. *Software Testing, Verification and Reliability*, 9(3) :191–193, 1999.
- [15] M. B. Alaya, S. Medjiah, T. Monteil, and K. Drira. Toward semantic interoperability in onem2m architecture. *IEEE Communications Magazine*, 53(12) :35–41, 2015.
- [16] Jennings, Shelby, Arkkio, Keranen, and Bormann. Sensor measurement lists (senml). RFC 8428, RFC Editor, 09 2018.

- [17] Site web de la plateforme IoT Mainflux. <https://mainflux.readthedocs.io/en/latest/messaging/>.
- [18] M. A. Razzaque, M. Milojevic-Jevric, A. Palade, and S. Clarke. Middleware for internet of things : A survey. *IEEE Internet of Things Journal*, 3(1) :70–95, 2016.
- [19] N. Naik. Choice of effective messaging protocols for iot systems : Mqtt, coap, amqp and http. In *2017 IEEE International Systems Engineering Symposium (ISSE)*, pages 1–7, 2017.
- [20] Roy T. Fielding, James Gettys, Jeffrey C. Mogul, Henrik Frystyk Nielsen, Larry Masinter, Paul J. Leach, and Tim Berners-Lee. Hypertext transfer protocol – http/1.1. RFC 2616, RFC Editor, June 1999. <http://www.rfc-editor.org/rfc/rfc2616.txt>.
- [21] Z. Shelby, K. Hartke, and C. Bormann. The constrained application protocol (coap). RFC 7252, RFC Editor, June 2014. <http://www.rfc-editor.org/rfc/rfc7252.txt>.
- [22] Oasis message queuing telemetry transport (mqtt). Standard, OASIS Standard., March 2019.
- [23] Oasis advanced message queuing protocol (amqp). Standard, OASIS Standard., October 2012.
- [24] A. Minaburo, L. Toutain, C. Gomez, D. Barthel, and JC. Zuniga. Schc : Generic framework for static context header compression and fragmentation. RFC 8724, RFC Editor, April 2020.
- [25] Kenton Varda. Protocol buffers : Google’s data interchange format. Technical report, Google, 6 2008.
- [26] GNU Radio. *The free and open source software radio ecosystem*, 2020. <https://www.gnuradio.org/>.
- [27] Node-RED. *Low-code programming for event-driven applications*, 2020. <https://nodered.org/>.
- [28] U. Ramacher, W. Raab, U. Hachmann, D. Langen, J. Berthold, R. Kramer, A. Schackow, C. Grassmann, M. Sauermann, P. Szreder, F. Capar, G. Obradovic, W. Xu, N. Bröls, K. Lee, E. Weber, R. Kuhn, and J. Harrington. Architecture and implementation of a software-defined radio base-band processor. In *2011 IEEE International Symposium of Circuits and Systems (ISCAS)*, pages 2193–2196, 2011.
- [29] M. Kiran, P. Murphy, I. Monga, J. Dugan, and S. S. Baveja. Lambda architecture for cost-effective batch and speed big data processing. In *2015 IEEE International Conference on Big Data (Big Data)*, pages 2785–2792, 2015.
- [30] LoRa Alliance. *LoRaWAN Specifications v1.0*, 01 2015.
- [31] Chiraz Houaidia, Hanen Idoudi, Adrien Van den Bossche, Leila Saidane, and Thierry Val. Inter-flow and intra-flow interference mitigation routing in wireless mesh networks. *Computer Networks*, 120(COMNET-D-16-826R1) :141–156, juin 2017.
- [32] Oana Andreea Hotescu, Katia Jaffres-Runser, Adrien Van Den Bossche, and Thierry Val. Synchronizing Tiny Sensors with SISP : a Convergence Study (regular paper). In *ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWIM 2017)*, Miami Beach, USA, 21/11/17-25/11/17, pages 279–287, <http://www.acm.org/>, novembre 2017. ACM : Association for Computing Machinery.
- [33] Rejane Dalce, Adrien Van Den Bossche, and Thierry Val. Optimization of a CSMA/CA based MAC protocol designed for confined WSNs (regular paper). In *IEEE International Conference on Wireless Communications in Unusual and Confined Areas (ICWCUCA 2012)*, Clermont-Ferrand, France, 28/08/12-30/08/12, page (electronic medium), <http://ieeexplore.ieee.org/>, 2012. IEEEExplore digital library.

- [34] M. Aref and A. Sikora. Free space range measurements with semtech lora™ technology. In *2014 2nd International Symposium on Wireless Systems within the Conferences on Intelligent Data Acquisition and Advanced Computing Systems*, pages 19–23, 2014.
- [35] M. Centenaro, L. Vangelista, A. Zanella, and M. Zorzi. Long-range communications in unlicensed bands : the rising stars in the iot and smart city scenarios. *IEEE Wireless Communications*, 23(5) :60–67, 2016.
- [36] J. Petajarvi, K. Mikhaylov, A. Roivainen, T. Hanninen, and M. Pettissalo. On the coverage of lpwans : range evaluation and channel attenuation model for lora technology. In *2015 14th International Conference on ITS Telecommunications (ITST)*, pages 55–59, 2015.
- [37] M. Lauridsen, B. Vejlgard, I. Z. Kovacs, H. Nguyen, and P. Mogensen. Interference measurements in the european 868 mhz ism band with focus on lora and sigfox. In *2017 IEEE Wireless Communications and Networking Conference (WCNC)*, pages 1–6, 2017.
- [38] Nicolas Gonzalez, Adrien Van den Bossche, and Thierry Val. Specificities of the lora physical layer for the development of new ad hoc mac layers. In *17th International Conference on Ad Hoc Networks and Wireless (AdHoc-Now 2018)*, pages 163–174, St Malo, FR, August 2018. Springer. Thanks to Springer editor. This papers appears in volume 11104 of Lecture Notes in Computer Science ISSN : 0302-9743 ISBN 978-3-030-00246-6 The original PDF is available at : https://link.springer.com/chapter/10.1007/978-3-030-00247-3_16.
- [39] K. Mikhaylov, . Juha Petaejaervi, and T. Haenninen. Analysis of capacity and scalability of the lora low power wide area network technology. In *European Wireless 2016 ; 22th European Wireless Conference*, pages 1–6, 2016.
- [40] O. Georgiou and U. Raza. Low power wide area network analysis : Can lora scale ? *IEEE Wireless Communications Letters*, 6(2) :162–165, 2017.
- [41] Z. Qu, G. Zhang, H. Cao, and J. Xie. Leo satellite constellation for internet of things. *IEEE Access*, 5 :18391–18401, 2017.
- [42] U. Raza, P. Kulkarni, and M. Sooriyabandara. Low power wide area networks : An overview. *IEEE Communications Surveys Tutorials*, 19(2) :855–873, 2017.
- [43] A. Mahmood, E. Sisinni, L. Guntupalli, R. Rondón, S. A. Hassan, and M. Gidlund. Scalability analysis of a lora network under imperfect orthogonality. *IEEE Transactions on Industrial Informatics*, 15(3) :1425–1436, 2019.
- [44] D. Croce, M. Gucciardo, S. Mangione, G. Santaromita, and I. Tinnirello. Impact of lora imperfect orthogonality : Analysis of link-level performance. *IEEE Communications Letters*, 22(4) :796–799, 2018.
- [45] Licia Amichi, Megumi Kaneko, Ellen Fukuda, Nancy Rachkidy, and Alexandre Guitton. Joint allocation strategies of power and spreading factors with imperfect orthogonality in lora networks. *IEEE Transactions on Communications*, PP :1–1, 02 2020.
- [46] Q. Lone, E. Dublé, F. Rousseau, I. Moerman, S. Giannoulis, and A. Duda. Wish-walt : A framework for controllable and reproducible lora testbeds. In *2018 IEEE 29th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, pages 1–7, 2018.
- [47] T. Petrić, M. Goessens, L. Nuaymi, L. Toutain, and A. Pelov. Measurements, performance and analysis of lora fabian, a real-world implementation of lpwan. In *2016 IEEE 27th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, pages 1–7, 2016.

- [48] M. N. Ochoa, A. Guizar, M. Maman, and A. Duda. Evaluating lora energy efficiency for adaptive networks : From star to mesh topologies. In *2017 IEEE 13th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, pages 1–8, 2017.
- [49] H. Lee and K. Ke. Monitoring of large-area iot sensors using a lora wireless mesh network system : Design and evaluation. *IEEE Transactions on Instrumentation and Measurement*, 67(9) :2177–2187, 2018.
- [50] D. Lundell, A. Hedberg, C. Nyberg, and E. Fitzgerald. A routing protocol for lora mesh networks. In *2018 IEEE 19th International Symposium on "A World of Wireless, Mobile and Multimedia Networks" (WoWMoM)*, pages 14–19, 2018.
- [51] Joachim Tapparel, Orion Afisiadis, Paul Mayoraz, Alexios Balatsoukas-Stimming, and Andreas Burg. An open-source lora physical layer prototype on gnu radio. 02 2020.
- [52] Revere engineering de la couche physique LoRa. *Lien vers le rapport en ligne*.
- [53] SX1276 Datasheet. <http://www.semtech.com/apps/filedown/down.php?file=sx1276.pdf>.
- [54] Packet forwarder source code. *Lien du dépôt du code source du packet forwarder*.
- [55] Chirpstack Gateway Bridge. <https://github.com/brocaar/chirpstack-gateway-bridge>.
- [56] IETF Draft of QUIC protocol. <https://tools.ietf.org/html/draft-ietf-quic-transport-27>.
- [57] Site web de Docker. <https://www.docker.com/>.
- [58] Site web de GDB. <https://www.gnu.org/software/gdb/>.
- [59] Site web de OpenOCD. <http://openocd.org/>.
- [60] Librairie Radiohead. <https://www.airspayce.com/mikem/arduino/RadioHead>.
- [61] Semtech Application Note AN1200.22. *LoRa Modulation Basics*, 2015.
- [62] Claire Goursaud and Jean-Marie de la Gorce. Dedicated networks for iot : Phy / mac state of the art and challenges. In *IOT 2015*, 2015.
- [63] N. Gonzalez A. Van Den Bossche, R. Dalce and T. Val. Specificities of the lora physical layer for the development of new ad hoc mac layers. St Malo, France, 2018. International Conference on Ad Hoc Networks and Wireless.
- [64] Orestis Georgiou and Usman Raza. Low power wide area network analysis : Can lora scale? *IEEE Wireless Communication Letters*, PP, 01 2017.
- [65] Y. Okumura. Field strength and its variability in vhf and uhf land mobile radio service. *Rev. Elect. Commun. Laboratory*, 16 :825–873, 01 1968.
- [66] M. Hatay. Empirical formula for propagation loss in land mobile radio services. *Vehicular Technology, IEEE Transactions on*, 29 :317–325, 08 1980.
- [67] Distribution de Rayleigh. *Page wikipedia de la distribution de Rayleigh*.
- [68] Librairie scientifique SciPy. <https://www.scipy.org/>.
- [69] Librairie de simulation à évènements discrets Simpy. <https://simpy.readthedocs.io/en/latest/>.

- [70] Andreas Ostermeier, Andreas Gawelczyk, and Nikolaus Hansen. A derandomized approach to self-adaptation of evolution strategies. *Evolutionary Computation*, 2(4) :369–380, 1994.
- [71] M. Slabicki, G. Premsankar, and M. Di Francesco. Adaptive configuration of lora networks for dense iot deployments. In *NOMS 2018 - 2018 IEEE/IFIP Network Operations and Management Symposium*, pages 1–9, 2018.
- [72] A. Van den Bossche, R. Dalcé, N. Gonzalez, and T. Val. Locura : A new localisation and uwb-based ranging testbed for the internet of things. In *2018 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pages 1–6, 2018.

Glossaire

ADR Adaptive Data Rate. 123, 128

base64 Encodage d'une chaîne de caractères en utilisant 64 caractères. 70, 71

CMA Covariance Matrix Adaptation. 8, 91, 116

CMA-ES Covariance Matrix Adaptation Evolution Strategy. 116

CSS Chirp Spread Spectrum. 61

FDR Frame Delivery Ratio. 8, 91, 113, 115, 116, 119, 120, 122, 145

FER Frame Error Rate. 110

GDB GNU Debugger. 85, 87

GNSS Système de positionnement par satellites. 86

GPIO General Purpose Input/Output. 94

IIoT Industrial Internet Of Things. 5, 11, 22

IMU Centrale Inertielle. 86

IoT Internet Of Things. 5, 11, 12, 27–29, 31, 35, 43, 49, 55, 57, 58, 122, 129, 147

JSON JavaScript Object Notation. 6, 12, 29, 42, 44, 45, 70, 71, 143

LPWAN Low Power Wide Area Network. 6, 7, 29, 51, 53–59, 64–69, 73, 75, 76, 78, 82–84, 87, 89, 93, 96, 115, 133, 134, 144

NAT Network Address Translation. 72, 85

PER Packet Error Rate. 57

QoS Quality of Service. 6, 12, 48–51, 98, 115, 116, 120, 143, 144

RSSI Received Signal Strength Indication. 57, 71, 101, 103–105, 107, 114, 125, 126, 145

SaaS Software As A Service. 84

SCHC Static Context Header Compression and Fragmentation. 29

SDK Software Development Kit. 21

SDR Software Defined Radio. 61, 144

UWB Ultra Wide Band. 84, 86

VM Machine Virtuelle. 28

VPN Virtual Private Network. 69, 85

WSN Wireless Sensor Network. 55

Table des figures

1.1	Le mainframe du CERN en octobre 1963 (http://information-technology.web.cern.ch/about/computer-centre/computing-history)	13
1.2	La carte logique du réseau ARPANET en 1977 (https://fr.wikipedia.org/wiki/ARPANET)	14
1.3	Modélisation d'un système embarqué standard	16
1.4	Modélisation d'un système embarqué communicant	18
1.5	Différentes topologies d'un réseau ZigBee	20
1.6	Nouvelle forme d'organisation des entreprises autour de la donnée	23
1.7	De la collecte des informations à l'agrégation et la modélisation	24
1.8	Architecture en silos applicatifs	26
1.9	Exemple d'un format de données JSON	29
1.10	Graphe de flux des silos applicatifs	30
1.11	Notion de composant interopérable	31
1.12	Notion de réseau interopérable	32
1.13	Modèle d'objet connecté à plusieurs interfaces de communication	33
1.14	Modèle des mécanismes réseaux d'un capteur connecté	34
1.15	Modèle des mécanismes réseaux d'un actionneur connecté	34
1.16	Modèle des mécanismes réseaux d'un objet connecté avec un bus de terrain	35
1.17	Modèle d'une passerelle avec empilement protocolaire intégré	38
1.18	Modèle d'une passerelle avec couche réseau déportée	39
1.19	Modèle d'une passerelle avec couche MAC déportée	40
1.20	Architecture lambda pour le traitement des données	41
1.21	Exemple de format de message au sein de Node-RED	42
1.22	Principe de l'interface de Node-RED	43
1.23	Exemple de message avec la représentation JSON	44
1.24	Exemple d'arborescence de topics MQTT	47
1.25	MQTT avec une QoS de 0	50

1.26	MQTT avec une QoS de 1	50
1.27	Problématique de retransmission avec une QoS de 1	51
1.28	MQTT avec une QoS de 2	51
2.1	Liste des bandes ISM en Europe (crédit ebds.eu)	59
2.2	Détails de la bande ISM 868 MHz (crédit ebds.eu)	60
2.3	Visualisation de la modulation LoRa observée à l'analyseur de spectre (SDR waterfall)	61
2.4	Format de trame LoRa à longueur variable (crédit Semtech)	63
2.5	Fonction python qui calcule le temps d'émission sur le canal	63
2.6	Temps de transmission des trames LoRa en seconde	64
2.7	Longueur du rapport cyclique en seconde des trames LoRa sur un canal à 1%	64
2.8	Longueur du rapport cyclique en seconde des trames LoRa sur un canal à 0,1%	64
2.9	Exemple de format de donnée utilisable	67
2.10	Architecture lambda pour le traitement des données	68
2.11	Architecture LoRaWAN via le Packet Forwarder telle que définie par Semtech (crédit Semtech)	70
2.12	Diagramme de séquence d'un relayage de trame	71
2.13	Diagramme de séquence d'une transmission de trame	72
3.1	Architecture de la puce sx1301 (Semtech)	77
3.2	Architecture interne de la puce sx1301 (Semtech)	78
3.3	Exemple d'installation d'une antenne LoRa sur un toit de Toulouse	79
3.4	Aperçu global d'une antenne ainsi que la gateway associée	80
3.5	Architecture de la collecte des trames LoRa	81
3.6	Concept de commutation de trames LoRa via MQTT	82
3.7	Ensemble de services réseau LPWAN	83
3.8	Comparaison des concepts de conteneurisation et de virtualisation	84
3.9	Programmation et réception des messages par l'infrastructure <i>testbed</i>	85
3.10	Architecture logicielle et matérielle du <i>testbed</i> urbain	86
3.11	Service réseau de visualisation des liens radio par géolocalisation	88
4.1	Noeud Feather LoRa M0+	94
4.2	Taux de réjection (dB) entre deux trames émises simultanément	94
4.3	Synchronisation entre les deux noeuds de l'expérimentation	95
4.4	Visualisation de la longueur des trames selon le spreading factor utilisé (l'abscisse est le temps)	96
4.5	Schématisation d'un bilan de liaison	97
4.6	Gateway LoRa sur le toit de l'IUT de Blagnac	100

4.7	Bilan de liaison entre la gateway d'étude et un nœud mobile	101
4.8	Cartographie des différents points de mesures autour de l'IUT de Blagnac	102
4.9	Résultats issus de la campagne de mesure sur l'axe nord	103
4.10	Définition de la bande favorable et de la bande de jeux équitables	104
4.11	Distributions de probabilités pour les différents distances	106
4.12	Distributions de probabilités de Rice et de Rayleigh	107
4.13	Modélisation de la distribution des RSSI pour une distance de 1500 mètres	107
4.14	Topologie utilisée dans notre simulateur	109
4.15	Fonction de répartition des noeuds	111
4.16	Répartition des nœuds en mètres autour de la gateway	112
4.17	Répartition des nœuds selon les spreading factors	113
4.18	Représentation du taux de livraison des messages par nœuds	114
4.19	État initial pour l'optimisation par bandes successives	117
4.20	Optimisation de la première bande par diminution du rayon de portée	118
4.21	Limite optimale pour laquelle le FDR minimal est de 40% dans la première bande	119
4.22	État final à la fin de l'optimisation	119
4.23	Optimisation pour 500 nœuds	121
4.24	Optimisation pour 1000 nœuds	121
4.25	Optimisation pour 6000 nœuds	122
4.26	Confrontation du taux de livraison et du bilan de liaison	124

Liste des tableaux

- 1.1 Principaux protocoles de transport pour l'IoT 28
- 4.1 Spreading Factor matrix for 10 ms delay 95
- 4.2 Définition des paramètres de simulation 111

Liste des publications personnelles

Conférences et workshops internationaux avec actes édités et comité de lecture

Adrien Van Den Bossche, Rejane Dalce, Nicolas Gonzalez, Thierry Val

LocURa : A New Localisation and UWB-Based Ranging Testbed for the Internet of Things (regular paper)

Dans : IEEE International Conference on Indoor Positioning and Indoor Navigation (IPIN 2018), Nantes, France, 24/09/2018-27/09/2018, IEEEExplore digital library, (support électronique), septembre 2018.

Résumé Accès : <https://oatao.univ-toulouse.fr/22454/>

Nicolas Gonzalez, Adrien Van Den Bossche, Thierry Val

Specificities of the LoRa physical layer for the development of new ad hoc MAC layers (regular paper)

Dans : International Conference on Ad Hoc Networks and Wireless (AdHoc-Now 2018), St Malo, France, 05/09/2018-07/09/2018, Springer, (support électronique), 2018.

Résumé Accès : <https://oatao.univ-toulouse.fr/22451/>

Adrien Van Den Bossche, Nicolas Gonzalez, Thierry Val, Damien Brulin, Frédéric Vella, Nadine Vigouroux, Eric Campo

Specifying an MQTT Tree for a Connected Smart Home (regular paper)

Dans : International Conference On Smart homes and health Telematics (ICOST 2018), Singapore, 10/07/2018-12/07/2018, Springer, (support électronique), juillet 2018.

Résumé Accès : <https://oatao.univ-toulouse.fr/22416/>

Nicolas Gonzalez, Adrien Van Den Bossche, Thierry Val

Hybrid Wireless Protocols for the Internet Of Things (regular paper)

Dans : IFIP/IEEE International Conference on Performance Evaluation and Modeling in Wired and Wireless Networks (PEMWN 2016), Paris, France, 22/11/2016-24/11/2016, IEEEExplore digital library, (support électronique), novembre 2016.

Résumé Accès : <http://ieeexplore.ieee.org/document/7842895/>

Conférences et workshops nationaux avec actes édités et comité de lecture

Nicolas Gonzalez, Matthieu Herrb, Laurent Guerby, Medhi Abaakouk

Routage haut-débit et filtrage de DDOS avec DPDK et Packet Journey (regular paper)

Dans : Journées Réseaux de l'Enseignement et de la Recherche, Nantes, 14/11/2017-17/11/2017, RENATER, (en ligne), novembre 2017.

Résumé Accès : <https://oatao.univ-toulouse.fr/22336/>

Conférences sans actes publiés

Nicolas Gonzalez

Réseau IoT LoRa ouvert sur Toulouse : déploiement, architecture et enjeux citoyens Dans : Capitole du Libre, Toulouse, 16/11/2019-17/11/2019 (conférencier invité).

Résumé Accès : [Lien de la vidéo sur youtube.com](#)

Nicolas Gonzalez

Déploiement d'un testbed LoRaWAN métropolitain ouvert

Dans : RESCOM 2019, Grenoble, 17/01/2019-18/01/2019 (conférencier invité).

Accès : <http://rescom2019.imag.fr/>

Nicolas Gonzalez

Déploiement d'une infrastructure LoRaWAN Tetaneutral en libre accès sur Toulouse

Dans : Capitole du libre, Toulouse, 17/11/2018-17/11/2018.

Résumé Accès : <https://2018.capitoledulibre.org/>

Nicolas Gonzalez

Libérez vos objets avec LoRaWAN

Dans : Capitole du Libre, Toulouse, 18/11/2017-19/11/2017.

Résumé Accès : [Lien de la vidéo sur youtube.com](#)

Nicolas Gonzalez, Adrien Van Den Bossche, Thierry Val

Plateforme pédagogique en libre accès pour l'expérimentation de l'Internet des Objets

Dans : Open Source Innovation Spring 2017 (OSIS 2017), Paris, France, 11/05/2017-11/05/2017.

Accès : <http://www.open-source-innovation-spring.org>

RÉSUMÉ

**Architectures protocolaires interopérables
pour le réseau de collecte dans l’Internet des Objets**

La connexion de nombreux éléments électroniques se base sur des réseaux sans fil ambiants, économes en énergie et longue distance. Le modèle technique et économique des réseaux cellulaires peut servir de base mais les nombreux cas d’usages nécessitent de casser ce monopole technologique et de penser les réseaux de demain comme interopérables et multi-technologies. Les réseaux LPWAN émergents sont un compromis entre les réseaux cellulaires consommateurs en énergie et les réseaux de capteurs qui ont une portée réduite. Ce nouveau paradigme ouvre de nouveaux champs de recherche en permettant d’hybrider les réseaux locaux et les réseaux cellulaires. Cette thèse traite de l’interopérabilité de ces réseaux sous différents aspects architecturaux. La contribution majeure de cette thèse réside dans le déploiement d’un réseau LoRa métropolitain sur la ville de Toulouse permettant aux chercheurs de créer et d’évaluer de nouveaux protocoles en conditions réelles pour la ville de demain.

Mots-clés : Internet des Objets, Ville intelligente, Réseaux sans fil, LoRa, Interopérabilité

ABSTRACT

**Interoperable protocol architectures
for the backhaul of the Internet of Things**

The connection of many electronic devices is based on ambient, energy-efficient and long range wireless networks. The technical and economic model of cellular networks can serve as a basis, but the many use cases require breaking this technological monopoly and thinking of the networks of tomorrow as interoperable and multi-technology. Emerging LPWAN networks are a compromise between energy consuming cellular networks and wireless sensor networks which have a reduced radio range. This new paradigm opens up new fields of research by making it possible to hybridize local networks and cellular networks. This thesis deals with the interoperability of these networks under different architectural aspects. The major contribution of this thesis lies in the deployment of a metropolitan LoRa network in the city of Toulouse allowing researchers to create and evaluate new protocols in real conditions for the city of tomorrow.

Keywords : Internet Of Things, Smart City, Wireless networks, LoRa, Interoperability
