



HAL
open science

Performances et Gestion de l'itinérance dans les réseaux LoRaWAN

Mohamed Hammache

► **To cite this version:**

Mohamed Hammache. Performances et Gestion de l'itinérance dans les réseaux LoRaWAN. Autre [cs.OH]. Institut National Polytechnique de Toulouse - INPT, 2022. Français. NNT : 2022INPT0074 . tel-04248270

HAL Id: tel-04248270

<https://theses.hal.science/tel-04248270>

Submitted on 18 Oct 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Université
de Toulouse

THÈSE

En vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par :

Institut National Polytechnique de Toulouse (Toulouse INP)

Discipline ou spécialité :

Informatique et Télécommunication

Présentée et soutenue par :

M. MOHAMED HAMNACHE

le vendredi 25 novembre 2022

Titre :

Performances et Gestion de l'itinérance dans les réseaux LoRaWAN

Ecole doctorale :

Mathématiques, Informatique, Télécommunications de Toulouse (MITT)

Unité de recherche :

Institut de Recherche en Informatique de Toulouse (IRIT)

Directeur(s) de Thèse :

M. ANDRÉ-LUC BEYLOT

M. RAHIM KACIMI

Rapporteurs :

M. ALEXANDRE GUITTON, UNIVERSITE CLERMONT AUVERGNE

M. CONG-DUC PHAM, UNIVERSITE DE PAU ET DES PAYS DE L ADOUR

Membre(s) du jury :

M. MICHEL MAROT, TELECOM SUD PARIS, Président

M. ANDRÉ-LUC BEYLOT, TOULOUSE INP, Membre

M. FRANCK ROUSSEAU, INP GRENOBLE, Membre

MME CHRISTELLE CAILLOUET, UNIVERSITE COTE D'AZUR, Membre

MME OANA IOVA, INSA LYON, Membre

M. RAHIM KACIMI, UNIVERSITE TOULOUSE 3, Membre

REMERCIEMENTS

Mon cœur est plein de gratitude envers vous toutes et tous qui m'avez accompagné, motivé ou aidé de quelque façon que ce soit.

Je tiens à remercier tout d'abord mon directeur de thèse, André-Luc Beylot et mon co-directeur Rahim Kacimi. Ces années passées avec vous ont été un vrai plaisir. Merci à tous les deux de m'avoir enseigné la recherche, donné le goût de l'excellence tout en m'accueillant comme j'étais et en m'accompagnant de très près dans chaque étape. Merci André-Luc pour ta bienveillance et ta présence durant la période de la covid. Merci Rahim pour les moments passés à construire, à planifier et à mener des expérimentations dans un environnement de partage et d'innovation.

Je remercie également les membres de mon jury de soutenance, professeurs et chercheurs de renom, pour avoir pris du temps pour évaluer mes travaux. Vos questions pertinentes après la soutenance ont conduit à des discussions riches qui ont fait mûrir mes réflexions sur les sujets abordés.

Je remercie aussi l'ensemble du personnel de l'équipe RMESS de l'IRIT au sein de laquelle j'ai passé de très bons moments, avant et même pendant la covid ! Merci à tous les doctorants, devenus de véritables amis.

Merci aussi à tous les permanents de l'équipe, merci pour l'environnement favorable et les discussions fructueuses.

Je ne saurais jamais assez remercier ma famille qui a toujours été là pour moi et a joué un grand rôle dans toute ma vie et dans cette thèse : mes parents, mes frères et "Y" qui m'ont emmené à *repousser mes limites* en croyant en moi, des fois plus que moi-même. Merci pour votre présence, votre patience, votre soutien indéfectible et vos nombreux encouragements.

RÉSUMÉ

LoRaWAN est l'un des principaux protocoles de communication sans fil déployés pour répondre aux exigences des applications IoT (Internet of Things) nécessitant une communication à longue portée avec une faible consommation d'énergie. On compte actuellement plus d'un milliard de dispositifs IoT utilisés dans le monde et LoRaWAN apparaît comme l'une des solutions les plus prometteuses pour de nombreuses applications. Bien que l'étude de ces réseaux, des mécanismes associés et de leurs performances soit un sujet particulièrement brûlant, certains aspects restent peu explorés, notamment la gestion de la mobilité et de l'itinérance (*roaming*). De même, de nombreux travaux reposent exclusivement sur des approches de simulation pour leur validation ou manquent d'outils et de bancs de test pour conduire des expérimentations réelles.

Dans cette thèse, nous proposons d'apporter des solutions à des problèmes bien connus au sein de la communauté LoRaWAN, notamment l'allocation des paramètres de transmission pour augmenter les performances du réseau. Nous proposons alors L3SFA et L3SFA-TPC pour améliorer le taux de délivrance, la capacité du réseau et la consommation énergétique des terminaux. Par la suite, nous nous sommes concentrés sur des sujets moins explorés. Nous proposons LoRaRoam, le premier système prenant en charge l'itinérance active ou itinérance *Handover* dans les réseaux LoRaWAN.

Face au manque ou à l'absence d'outils de référence, nous nous sommes attachés au développement de solutions et au déploiement de bancs de test pour la validation des solutions proposées et pour l'analyse de leurs performances à large échelle. Nous avons conçu *LoRa Roaming Emulator* (LDE), le premier émulateur d'itinérance dédié à l'évaluation des réseaux LoRaWAN itinérants. De plus, nous avons proposé LoRaDL, un framework pour prendre en charge les communications sur la liaison descendante (commandes MAC, trames de données) facilitant ainsi la mise en œuvre des solutions centrées sur les passerelles (*gateway-centric*) et l'évaluation des performances de la liaison descendante.

ABSTRACT

LoRaWAN is one of the leading wireless communication protocols deployed to meet the requirements of IoT (Internet of Things) applications requiring long-range communication with low power consumption. Currently, more than one billion IoT devices in use around the world and LoRaWAN is emerging as one of the most promising communication protocols for many applications. Although the study of these networks, the associated mechanisms and their performances is a particularly hot topic, some aspects remain unexplored, in particular the management of mobility and roaming. Similarly, many works rely exclusively on simulation approaches for their validation or lack tools and testbeds to perform real experiments.

In this thesis, we propose to provide solutions to well-known problems within the LoRaWAN community, in particular the allocation of transmission parameters to increase network performance. We then propose L3SFA and L3SFA-TPC to improve the Data Extraction Rate (DER), the network capacity, and the terminal energy consumption. Thereafter, we focus on less explored topics. We propose LoRaRoam, the first system supporting Handover Roaming in LoRaWAN networks.

Faced with the lack or absence of reference tools, we have focused on developing solutions and deploying testbeds to validate the proposed solutions and analyze their performance on a large scale. We designed *LoRa Roaming Emulator* (LDE), the first roaming emulator dedicated to the evaluation of roaming LoRaWAN networks. In addition, we proposed LoRaDL, a framework to support downlink communications (MAC commands, data frames) facilitating the implementation of gateway-centric solutions and the evaluation of downlink performance.

TABLE DES MATIÈRES

Table des figures	xiii
Liste des tableaux	xv
Glossaire	xix
1 Introduction	1
1.1 Contexte	1
1.1.1 L’IoT au cœur de l’Internet du futur	1
1.1.2 LoRaWAN, la star des réseaux LPWANs	2
1.1.3 Les applications de la technologie LoRaWAN	3
1.2 Motivations	6
1.3 Méthodologie	6
1.3.1 Définition des problèmes de recherche visés	6
1.3.2 Validation multi-approche	7
1.3.3 Processus de développement	7
1.4 Contributions et structure du manuscrit	8
1.4.1 L3SFA : Stratégie d’équilibrage de charge pour l’allocation des facteurs d’étalement dans les réseaux LoRaWAN	8
1.4.2 LoRaRoam : Itinérance dans les réseaux LoRaWAN	9
1.4.3 LoRaDL : un <i>framework</i> pour prendre en charge les communications sur la liaison descendante LoRaWAN	9
2 Principes Fondamentaux des Réseaux LoRaWAN	11
2.1 LoRa	12
2.1.1 La modulation LoRa	12
2.1.2 Les paramètres de transmission radio	12
2.2 LoRaWAN	13
2.2.1 LoRaWAN : Architecture du Réseau	13

2.2.2	LoRaWAN : classes de terminaux	15
2.2.3	LoRaWAN : Activation des Terminaux	16
2.2.4	LoRaWAN : Sécurité	17
2.2.5	LoRaWAN : Itinérance	18
I	Allocation de Ressources Radio	21
3	Etat de l'art	23
3.1	Défis liés aux facteurs d'étalement	23
3.1.1	Densification des réseaux LoRaWAN	23
3.1.2	Orthogonalité imparfaite des facteurs d'étalement	24
3.1.3	Portée de la communication	24
3.1.4	Accès multiple au canal	24
3.2	Mesures de performance	25
3.2.1	Taux de délivrance	25
3.2.2	Consommation énergétique	25
3.2.3	Débit	25
3.2.4	Probabilité d'interruption	26
3.2.5	Probabilité de couverture	26
3.2.6	Taux d'occupation des canaux	27
3.3	Classification des stratégies d'allocation de facteurs d'étalement	27
3.3.1	Allocation aléatoire de SF	27
3.3.2	Allocation de SF fondée sur la distance	28
3.3.3	Allocation fixe de SF	30
3.3.4	Allocation de SF fondée sur un algorithme d'apprentissage machine	30
3.3.5	Allocation de SF fondée sur la couche physique	31
3.3.6	Allocation de SF fondée sur l'optimisation	33
3.4	Travaux les plus pertinents	35
3.5	Notre positionnement	36
4	L3SFA : Stratégie d'équilibrage de charge pour l'allocation de SF dans les réseaux LoRaWAN	37
4.1	Modélisation du problème d'allocation des facteurs d'étalement dans des réseaux LoRaWAN denses	37
4.1.1	Décalage des facteurs d'étalement	37
4.1.2	Charge SF	38

4.1.3	Allocation optimale des SF	39
4.1.4	Heuristiques d'équilibrage de charge et d'allocation de SF	40
4.2	Analyse de performances	44
4.2.1	Modèle de propagation	45
4.2.2	Validation du modèle de propagation par expérimentation	45
4.2.3	Scénarios de simulation	47
4.3	Pour aller plus loin : Impact du contrôle de la puissance de transmission	52
4.3.1	Interférences Inter-SF et Intra-SF	53
4.3.2	Contrôle de la puissance de transmission	53
4.4	Conclusions et discussions	55
II	Itinérance et Mobilité	57
5	Etat de l'art	59
5.1	Format de la trame LoRaWAN	59
5.1.1	Charge utile physique	60
5.1.2	Charge utile MAC	61
5.2	Activation des terminaux	61
5.2.1	Activation OTAA dans LoRaWAN v1.0.x	61
5.2.2	Activation OTAA dans LoRaWAN v1.1	62
5.3	Travaux pertinents sur l'itinérance LoRaWAN	64
5.3.1	Itinérance à base de 5G	64
5.3.2	Itinérance à base de blockchain	65
5.3.3	Itinérance à base de systèmes répartis	65
5.3.4	Itinérance à base de SCHC	65
5.3.5	Itinérance à base de DNS	66
5.4	Notre positionnement	66
6	LoRaRoam : Itinérance dans les réseaux LoRaWAN	69
6.1	Conception de la solution d'itinérance LoRaWAN	69
6.1.1	Architecture étendue d'itinérance LoRaWAN	69
6.1.2	Procédure d'activation	72
6.2	Implantation de l'itinérance LoRaWAN	75
6.3	Scénarios expérimentaux	76
6.3.1	Trafic faible	80
6.3.2	Trafic moyen	80

6.3.3	Trafic saturé	81
6.3.4	Interprétation des résultats d'expériences	81
6.4	Scénarios d'émulation	82
6.4.1	Conception de l'émulateur d'itinérance LoRaWAN	82
6.4.2	Description de l'émulation d'itinérance LoRaWAN	83
6.4.3	Configuration de l'environnement d'émulation	84
6.4.4	Paramètres d'émulation	85
6.4.5	Résultats de l'émulation de l'itinérance	86
6.5	Conclusions et discussions	88
III	Liaison Descendante LoRaWAN	91
7	Etat de l'art	93
7.1	Politique sur la liaison descendante en LoRaWAN	93
7.2	Commande MAC LoRaWAN	94
7.2.1	Encapsulation des commandes MAC	94
7.2.2	Structure des commandes MAC	94
7.3	Travaux pertinents	95
7.3.1	Études fondées sur la modélisation	96
7.3.2	Études fondées sur l'expérimentation	97
7.4	Notre positionnement	99
8	LoRaDL : Un framework pour prendre en charge les communications sur la liaison descendante LoRaWAN	101
8.1	Conception de LoRaDL	101
8.1.1	Control Engine	102
8.1.2	MAC Command Directory	103
8.1.3	MAC Command Formatter	104
8.1.4	Network Server Client	104
8.1.5	Payload Formatter	105
8.1.6	Downlink Generator	105
8.1.7	Application Server Client	105
8.2	Implantation de LoRaDL	105
8.2.1	LoRaMAC-DL	105
8.2.2	LoRaAppDL	106
8.3	Évaluation des performances	107

8.3.1	Scénario 1 : preuve de concept	107
8.3.2	Scénario 2 : limites de la liaison descendante du terminal	109
8.3.3	Scénario 3 : Impact de la charge de trafic	110
8.4	Conclusion	113
	Conclusions et perspectives	115
	Bibliographie	121

TABLE DES FIGURES

1.1	Avantages respectifs de Sigfox, LoRa et NB-IoT en termes de caractéristiques de l'IoT [98].	4
2.1	Pile Technologique des Réseaux LoRaWAN [3].	11
2.2	Architecture des réseaux LoRaWAN [103].	14
2.3	Fenêtres de réception de classe A [4].	15
2.4	Fenêtres de réception de classe B [4].	16
2.5	Fenêtres de réception de classe C [4].	17
2.6	Architecture des réseaux LoRaWAN.	18
3.1	Classification des différentes stratégies d'allocation de SF [96].	28
3.2	Cellule LoRaWAN [96].	29
4.1	Mécanisme de décalage de SF [66]	43
4.2	Stratégie d'équilibrage de charge pour l'allocation de facteurs d'étalement (seuls les SF7 à SF10 sont considérés sur ce schéma) [66]	43
4.3	Plateforme d'expérimentation.	46
4.4	RSSI : Modèles mathématiques versus mesures réelles [66].	47
4.5	Coordonnées GPS transmises par un terminal RN2483 équipé d'un module GPS Xbee.	48
4.6	Répartition des SF pour 5000 terminaux dans une cellule de rayon de 600 mètres pour chaque étape de L3SFA.	49
4.7	Performances de L3SFA avec positionnement uniforme des terminaux. [a-c] DER en fonction du nombre de terminaux pour une période moyenne d'émission de trames de $\frac{1}{100s}$, $\frac{1}{300s}$, $\frac{1}{600s}$, respectivement. [d-f] Répartition des SF pour 5000 terminaux. [g-i] Répartition finale des SF pour 5000 terminaux dans une cellule d'un rayon de 600 mètres.	51
4.8	Analyse des performances de L3SFA pour un positionnement de terminaux issu d'un jeu de données.	52

4.9	Analyse de performance de L3SFA-TPC.	54
5.1	Structure d'une trame LoRaWAN [5].	59
5.2	Échange de messages OTAA dans LoRaWAN v1.0.x	62
5.3	Échange de messages OTAA dans LoRaWAN v1.1	63
6.1	Architecture LoRaWAN étendue pour prendre en charge l'itinérance active.	70
6.2	Architecture système de la solution d'itinérance active	71
6.3	Format de JoinEUI.	73
6.4	Format du message JoinRequest.	73
6.5	Protocole d'itinérance LoRaWAN.	74
6.6	Configuration du scénario expérimental de l'itinérance LoRaWAN	77
6.7	Matériel expérimental	77
6.8	Configuration matérielle des expérimentations	80
6.9	Évaluation des performances en termes de latence d'activation pour la première visite du terminal dans un scénario d'itinérance.	81
6.10	Architecture de l'émulateur d'itinérance LoRaWAN.	83
6.11	Description de l'émulation d'itinérance LoRaWAN	84
6.12	Configuration du déploiement de l'émulation.	85
6.13	Latence d'activation dans un scénario d'itinérance.	87
6.14	Évaluation des performances en termes d'utilisation du processeur et de la mémoire.	87
7.1	Structure de la charge utile MAC	94
7.2	Structure de la commande LinkADRReq [2]	96
8.1	Architecture du framework LoRaDL	102
8.2	Structure de la commande MAC LinkADRReq	103
8.3	Configuration expérimentale : (a) Passerelle Mikrotik wAP LR8 LoRaWAN, (b) Terminal Pycom Lopy4, (c) Terminal Microchip RN2483.	107
8.4	(a) Trames par SF et type de message, (b) Trames de liaison montante par canal, et (c) Trames de liaison descendante par passerelle.	108
8.5	Taux d'extraction de paquets	109
8.6	Charge de trafic ρ en fonction de la configuration SF reçue en liaison descendante.	111
8.7	PER : Cas N°1 vs Cas N°2.	112
8.8	PER : liaisons montantes vs liaison descendantes	113

LISTE DES TABLEAUX

2.1	Paramètre de transmission radio LoRa en Europe	13
2.2	Abréviation des clés de sécurité.	17
3.1	Différentes mesures de performance des réseaux LoRaWAN.	27
3.2	Seuils de SNR pour une bande passante de 125 kHz en [dB].	32
3.3	Comparaison entre différents schémas d'allocation de facteurs d'étalement.	34
4.1	Définition des variables du modèle d'allocation de SF.	38
4.2	Seuils de SNR et de sensibilité pour une bande passante de 125 kHz en [dB].	38
4.3	Configuration de la Simulation	49
4.4	Seuils d'interférence [dbm].	50
5.1	Activation OTAA dans LoRaWAN v1.0.x versus v1.1	64
6.1	Tableau des souscriptions au réseau itinérant	78
6.2	Exemples de JoinEUI déduits	78
6.3	Paramètres de transmission du terminal domestique.	79
6.4	Paramètres d'émulation	85
6.5	Latence d'activation pour la première visite du terminal du réseau partenaire	86
7.1	Extrait de la liste des commandes MAC LoRaWAN	95
8.1	Résumé des scénarios expérimentaux	108
8.2	Taux moyen d'extraction de paquets	110

GLOSSAIRE

- 3GPP** *3rd Generation Partnership Project*. 8, 44, 45, 47
- ABP** *Activation By Personalization*. 16, 17, 72
- ADR** *Adaptive Data Rate*. 32, 61
- API** *Application Programming Interface*. 75, 76, 82, 104, 105
- AS** *Application Server*. 14, 105
- BW** *Bandwidth*. 24, 51, 86
- CE** *Control Engine*. 106
- CH** *Channel*. 103
- CPU** *Central Processing Unit*. 9, 85–88, 116
- CR** *Coding Rate*. 12, 24, 51, 86
- CSS** *Chirp Spread Spectrum*. 12
- DER** *Data Extraction Rate*. 8, 25, 35, 36, 40, 44, 48, 50, 52, 54
- DNS** *Domain Name System*. 9, 66, 71, 73–76, 78, 79, 81, 83, 86, 117
- DR** *Data Rate*. 32, 106
- EAB** *Equal Area Based*. 29, 30, 97
- EC-GSM-IoT** *Extended Coverage-Global System for Mobile-Internet of Things*. 2
- EDG** *End-Device Generator*. 82
- EIB** *Equal Interval Based*. 28–30, 32, 97
- ENSEEIH** *École Nationale Supérieure d'Electrotechnique, d'Electronique, d'Informatique, d'Hydraulique et des Télécommunications*. 76
- GC** *Generator Component*. 82
- GPS** *Global Positioning System*. 4, 46
- GSM-IoT** *Global System for Mobile-Internet of Things*. 2
- HLA** *Home Local Agent*. 69, 71, 73–76, 78, 79, 84, 86–88

- IIoT** *Industrial Internet of Things*. 2
- INLP** *Integer Nonlinear Programming*. 39
- ISM** *Industrial, Scientific, and Medical*. 53
- IoT** *Internet of Things*. 1–3, 5, 6, 9, 25, 28, 64, 115
- JRG** *JoinRequest Generator*. 82
- JS** *Join Server*. 63
- L3SFA-TPC** *Load Shifting Strategy for Spreading Factor Allocation-Transmission Power Control*. 8, 21, 36, 37, 53–55, 115, 116, 118
- L3SFA** *Load Shifting Strategy for Spreading Factor Allocation*. 8, 21, 36, 37, 40, 47–50, 52–55, 115, 116, 118
- LPWAN** *Low Power Wide Area Network*. 2, 3, 36, 65, 91
- LRE** *LoRaWAN Roaming Emulator*. 82, 83, 116
- LTE-M** *Long-Term Evolution Machine*. 2
- LoRADL** *Long Range Downlink*. 9, 10, 55, 91, 99, 101, 102, 107, 113, 116–118
- LoRARoAM** *LoRa Roaming*. 58, 66, 116, 117
- LoRAWAN** *Long Range Wide Area Network*. 3–19, 21, 23–27, 29–32, 35–37, 44, 45, 47, 50, 53, 55, 57–60, 62–66, 69, 72, 73, 75, 76, 80, 82, 84, 86, 88, 91, 94–99, 101–103, 106–108, 113, 115–117
- LoRA** *Long Range*. 3–5, 12–14, 24, 44, 45, 50
- M2M** *Machine to Machine*. 2
- MACD** *MAC command directory*. 103, 104, 106
- MACF** *MAC Commands Formatter*. 104, 106
- MAC** *Medium Access Control*. 3, 9, 11, 18, 19, 59–62, 66, 74, 82, 88, 91, 93, 94, 98, 99, 101–106, 108, 110, 113, 116, 117
- MA** *Master Agent*. 69, 71, 73–77, 84, 86–89, 117
- MIC** *Message Integrity Check*. 17, 18, 76
- MQTT** *Message Queuing Telemetry Transport*. 82
- MSCHC** *Mobile Static Context Header Compression*. 65
- NB-IoT** *Narrowband Internet of Things*. 2, 3
- NS** *Network Server*. 14, 32, 61, 64, 69, 71, 72, 76, 103, 104, 106, 107, 110, 111
- NET-ID** *network Identifier*. 71, 73, 74, 76–78, 82

- OTAA** *Over the Air Activation*. 16, 17, 61, 63, 72, 79, 81
- PER** *Packet Extraction Rate*. 111
- PL** *Payload Length*. 85
- QoS** *Quality of service*. 2
- RAM** *Random Access Memory*. 85
- RPC** *Roaming Protocol Component*. 83
- RSSI** *Received Signal Strength Indication*. 36, 37, 40, 41, 44–47, 49–51, 110, 115
- SCHC** *Static Context Header Compression*. 65
- SF** *Spreading Factor*. 12, 24–33, 35, 36, 38–42, 44, 47–53, 55, 86, 96, 97, 103, 110, 111
- SINR** *Signal to-inference-plus-noise ratio*. 27
- SIR** *Signal-to-Interference Ratio*. 52, 53
- SNR** *Signal to Noise Ratio*. 12, 26, 28, 31, 32, 36, 37, 40, 41, 44, 47, 49–51, 110, 115
- TP** *Transmission Power*. 12, 24, 26, 32, 103
- USRP** *Universal Software Radio Peripheral*. 98
- VLA** *Visited Local Agent*. 69, 71, 72, 74–76, 79, 84, 86–88
- HNS** *Home Network Server*. 19, 69, 72–76, 78–84, 86, 89
- vNS** *Visited Network Server*. 19, 71, 72, 74–76, 78, 79, 81–84, 86, 88, 89

INTRODUCTION

1.1 Contexte

1.1.1 L’IoT au cœur de l’Internet du futur

Ces dernières années, l’Internet des objets [16] [106] [82] ou en anglais *Internet of Things* IoT¹ a suscité un fort engouement de la communauté de recherche, des industriels et du grand public. Un véritable écosystème s’est rapidement constitué au travers de solutions technologiques variées dans des contextes applicatifs très hétérogènes. Considéré comme une partie de l’Internet du futur, il comprendra des milliards d’objets intelligents, communicants et dotés de capteurs et d’actionneurs, qui collectent, analysent et partagent des données avec d’autres objets, programmes et plateformes [106] [82].

Plusieurs définitions ont été données pour décrire ce nouveau paradigme de communication. Dans [82], les auteurs ont défini l’IoT comme “un ensemble d’objets connectés au monde virtuel où ils sont contrôlés à distance et peuvent servir de points d’accès physiques aux services Internet”. Dans [71], les auteurs l’ont décrit comme “un réseau mondial d’objets physiques utilisant l’Internet comme moyen de communication”. Les auteurs de [26], le voit comme “un réseau d’objets connectés pour détecter, surveiller et interagir au sein d’une entreprise et entre l’entreprise et sa chaîne d’approvisionnement, permettant l’agilité, la visibilité, le suivi et le partage d’informations pour faciliter la planification, le contrôle et la coordination en temps voulu des processus de la chaîne d’approvisionnement”.

On peut, sans se tromper, penser que désormais le futur de l’Internet sera constitué

1. Tout au long de ce manuscrit, nous emploierons le terme IoT pour faire référence à l’Internet des objets

d'objets connectés de manière hétérogène qui étendront encore les frontières du monde avec des entités physiques et des composants virtuels pour donner naissance à diverses solutions et applications conçues à dessein. Les terminaux **IoT** et **M2M** (*Machine to Machine*) [122], [77] sont connectés sans fil via des technologies de réseaux étendus à faible consommation énergétique connus sous le nom de **LPWAN** (*Low Power Wide Area Network*). Les réseaux **LPWAN** sont censés résoudre plusieurs problèmes liés aux technologies existantes et aux réseaux cellulaires. Les dernières technologies sont destinées à répondre aux applications de l'**IIoT** (*Industrial Internet of Things*).

1.1.2 LoRaWAN, la star des réseaux LPWANs

Les technologies **LPWAN** actuelles peuvent être réparties en trois catégories de réseaux en fonction de leurs besoins en infrastructures supplémentaires : (1) les réseaux fondés sur une infrastructure cellulaire comme le **NB-IoT** [131], [10], (2) les réseaux utilisant une infrastructure tierce comme SigFox [88] et (3) les **LPWAN** autonomes ne nécessitant aucune infrastructure tierce comme les réseaux à longue portée (LoRa) ou les réseaux étendus à longue portée (LoRaWAN) [120].

Pour la première catégorie, les **LPWAN** utilisant la technologie cellulaire ont une capacité élevée, des mécanismes dédiés à la qualité de service (**QoS**) et à la sécurité. Ce type de **LPWAN** peut être représenté par trois technologies principales : **NB-IoT** [131], **LTE-M** [86], et **GSM-IoT** (**EC-GSM-IoT**) [91]. Ces réseaux utilisent des bandes de fréquences sous licence, sont exploités par des réseaux commerciaux reposant sur l'abonnement aux données, et sont contrôlés par les opérateurs mobiles. Ils sont coûteux en raison des frais d'abonnement auprès des opérateurs de réseaux cellulaires. Ainsi, ils ne répondent pas à des exigences de fonctionnement autonome, doivent suivre les conditions de l'opérateur en termes de disponibilité de la couverture, de coût du service et de nombre d'appareils connectés. Ils visent un marché haut-de-gamme avec des débits relativement élevés ; leur fonctionnement est complexe. Par conséquent, ils sont les plus gourmands en termes de consommation énergétique comparés aux autres technologies **LPWAN** [76].

Dans la deuxième catégorie, on trouve SigFox [88] et Telensa [75] qui sont des réseaux brevetés qui sont proposés en tant que service par des fournisseurs. L'absence de normes et de technologies adaptées à l'IoT encourage le développement de réseaux spécialisés comme SigFox, qui est principalement conçu pour les applications à faible débit de données. SigFox offre une couverture assez large, une très bonne portée et une consommation énergétique efficace. Cependant, cette technologie souffre de problèmes de latence et de la taille très faible de la charge utile des trames. De plus, toute l'infrastructure est gérée par

un fournisseur, ce qui ne permet pas le contrôle du réseau ni la possibilité de proposer de nouveaux mécanismes, services ou protocoles.

Dans la dernière catégorie, on trouve principalement **LoRa/LoRaWAN**, une technologie *open source* qui permet de mettre en place à faible coût un réseau privé ou public de manière autonome et sans infrastructure tierce. Contrairement à SigFox et **NB-IoT**, **LoRaWAN** permet de développer des réseaux privés et de les intégrer simplement à un large panel de plateformes de réseaux mondiaux (par exemple, The Things Network) [102]. En raison de ses caractéristiques d'accès libre, **LoRaWAN** a suscité l'intérêt de la communauté des chercheurs dès son introduction sur le marché. En dépit de la diversité des technologies de la famille **LPWAN**, LoRa/LoRaWAN connaît un grand succès en raison de ses applications considérables dans l'industrie ou l'éducation par exemple. C'est sur cette technologie que porte la présente thèse de doctorat.

LoRa est une couche physique avec une technologie de communication à faible consommation d'énergie, résistante aux interférences et à longue portée fonctionnant dans des bandes de fréquences libres. **LoRaWAN** comprend le protocole de contrôle d'accès au support (**MAC**) pour la couche liaison de données et la couche réseau sur **LoRa**. C'est la solution soutenue et développée en tant que standard par l'Alliance **LoRa** [107]. Ces caractéristiques confient aux réseaux **LoRa** un potentiel important pour diverses applications **IoT**. Nous reviendrons plus avant sur les technologies **LoRa/LoRaWAN** dans le chapitre 2 et tout au long de ce manuscrit.

Un récapitulatif comparatif des caractéristiques des technologies emblématiques des trois grandes catégories recensées précédemment, **NB-IoT**, Sigfox et **LoRa**, est donné dans la figure 1.1.

1.1.3 Les applications de la technologie LoRaWAN

Dans cette section, nous passons en revue les principales applications de la technologie **LoRaWAN** sous forme d'un **voyage virtuel** aux quatre coins du monde. Nous détaillons les cas d'utilisation en les organisant par domaine d'application. Il est à noter que cette liste n'est pas exhaustive et a pour objectif de mettre l'accent sur la valeur ajoutée des réseaux **LoRaWAN**.

Applications pour les villes intelligentes

En Corée du Sud, un réseau qui atteint 99% de la population est en cours de déploiement à l'aide de **LoRaWAN**, tandis qu'Amsterdam, aux Pays-Bas, a été entièrement couverte par 10 passerelles [17] [24]. Londres et ses environs, en Angleterre, sont déjà

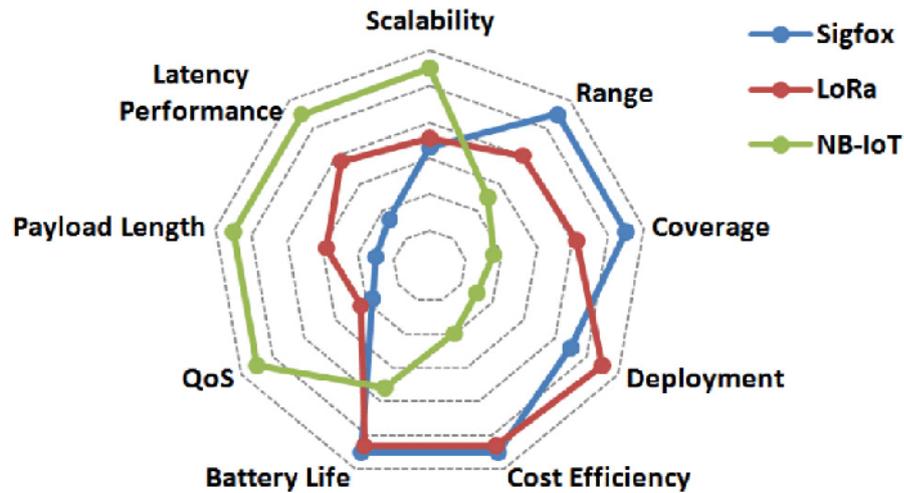


FIGURE 1.1 Avantages respectifs de Sigfox, LoRa et NB-IoT en termes de caractéristiques de l’IoT [98].

couverts de manière adéquate par **LoRa** [135].

Il existe plusieurs applications dans les transports publics et la surveillance du trafic routier. En Inde [78], certains bus sont équipés d’émetteurs-récepteurs **LoRa** et les arrêts de bus de récepteurs. À Nagoya, au Japon [28], il a été possible de suivre la localisation des bus en utilisant de tels émetteurs-récepteurs dotés de **GPS** pour capturer le positionnement des bus en temps réel. En Argentine [57], Cordoba et Buenos Aires sont des villes qui ont déjà mis en place des réseaux **LoRa** avec des terminaux composés de cartes BeagleBone Black, de **GPS** et de modules **LoRa**. À Selangor, en Malaisie, un déploiement a été effectué afin d’améliorer le monitoring du trafic dans les zones urbaines denses. Il permet d’améliorer la circulation des véhicules aux endroits visés [105]. À Marrakech, au Maroc [80], un prototype doté d’une caméra est présent dans les pare-brises de véhicules, détectant ainsi si un conducteur dort au volant. Les données collectées sont transmises aux applications en utilisant un réseau **LoRaWAN**.

Une autre classe d’applications est la surveillance urbaine. Les auteurs, dans [125], ont créé des capteurs qui mesurent la température, l’humidité, la poussière et le dioxyde de carbone dans l’air des deux villes, Yangon au Myanmar et Tampines à Singapour. La surveillance de la qualité de l’air a également été étudiée à La Plata, en Argentine [32]. La surveillance de l’environnement a fait l’objet d’expérimentations dans la ville de Hsinchu à Taïwan [132] ou bien encore à New York, aux États-Unis [40] où l’on s’est intéressé au contrôle des polluants urbains. Pour les deux applications, des réseaux **LoRaWAN** sont utilisés pour collecter les données dans différents quartiers.

LoRaWAN est encore utilisé dans d'autres applications telles que : la gestion des déchets [53], la gestion logistique dans les aéroports [110] ou la surveillance des emplacements libres des bateaux dans un port ou de voitures dans un parking.

Applications pour la ferme intelligente

L'agriculture de précision est la principale application dans ce contexte. À Campinas, au Brésil [33], des données précises de la couverture en zone rurale ont été collectées en faisant varier la hauteur du positionnement des passerelles, y compris à l'aide de drones. Ces résultats peuvent être exploités dans des champs. On trouve des études similaires à Turin, en Italie [127], et à Tronoh, en Malaisie [73], [74]. Les auteurs de [104] ont créé un réseau IOT pour surveiller des serres à champignons. À Skopje, en Macédoine [41], un vignoble est surveillé à l'aide de capteurs pour mesurer la température de l'air, l'humidité, la teneur en eau des feuilles et l'humidité du sol.

La surveillance de troupeaux est une autre application classique. À Montevideo, en Uruguay [25], chaque animal du troupeau dispose d'un collier ou d'une étiquette d'oreille avec un accéléromètre intégré à un émetteur LORA. À Ioannina, Grevena et Kavala, en Grèce [137], un réseau LORA pour la surveillance des animaux a été développé. Les données de surveillance sont consultables via une application Android appelée *CowTrack*. Le suivi de vaches dans les champs a également été étudié à Baotou, en Chine [89].

Applications pour les réseaux de distribution intelligents

Les auteurs de [130] ont montré l'utilisation de la technologie LORAWAN avec des compteurs d'énergie électrique en déployant des terminaux et des passerelles à Paris et en analysant les portées maximales. À Gand, en Belgique [87], le contrôle de la température dans les équipements liés aux lignes de distribution d'électricité a pu être effectué avec des émetteurs LORA. Ce service peut être utilisé pour optimiser le déploiement de l'énergie dans les villes intelligentes [12]. De même, des applications avec des compteurs intelligents pour la consommation d'eau et de gaz pourraient être déployées [92, 95, 115].

Applications pour la santé

LORA a été utilisé pour la transmission des données de la surveillance de la fréquence cardiaque, la respiration, le niveau de liquide sanguin et d'autres indicateurs [72], [29]. Dans des études menées à Sfax, en Tunisie, et à Marburg, en Allemagne [97], un réseau LORA permet de surveiller des patients éloignés des centres de santé. À Belgrade, en Serbie, et à Ljubljana, en Slovénie [83], la mise en œuvre de la télédétection a permis de

surveiller la santé des athlètes en utilisant plusieurs capteurs dans le corps. À Karachi, au Pakistan [14], les auteurs ont développé un système embarqué capable d'évaluer les constantes de santé des soldats et d'envoyer les données via un réseau **LORAWAN** à un centre de données.

Autres applications

Les réseaux **LORAWAN** sont exploités dans beaucoup d'autres applications telles que : la localisation [42, 51, 109], les campus intelligents [93] ou bien encore l'automatisation de l'industrie [114].

1.2 Motivations

LoRaWAN est l'un des protocoles de communication sans fil phares déployés pour prendre en charge les applications **IoT** nécessitant une communication à longue portée avec une faible consommation d'énergie. Plus d'un milliard de dispositifs **IoT** sont actuellement utilisés dans le monde. **LORAWAN** s'impose comme l'un des protocoles de communication prometteurs pour de nombreuses applications **IoT** avancées, la liste précédente en est la preuve irréfutable. Bien que plusieurs travaux de recherche soient menés pour améliorer les performances de ces réseaux ou pour proposer de nouveaux mécanismes, certains axes, tels que l'itinérance, restent peu explorés. Par ailleurs, beaucoup de travaux reposent uniquement sur des approches par simulation pour valider les travaux. Il existe un manque criant d'outils et de banc de tests pour mener des expérimentations réelles.

Pour ces raisons, dans cette thèse, nous proposons d'apporter des solutions à des problèmes connus de la communauté **LORAWAN** (Allocation de paramètres de transmission), à des orientations peu explorées (Handover roaming) et à un manque ou une absence d'outils, de bancs de test et de *framework* pour la validation des solutions proposées et pour l'analyse de performance.

1.3 Méthodologie

1.3.1 Définition des problèmes de recherche visés

Dans cette thèse, nous nous sommes intéressés à apporter des solutions à trois axes de recherche dans les réseaux **LORAWAN** : l'allocation des paramètres de transmission, l'itinérance et les communications sur la liaison descendante. Le choix de ces thématiques

est orienté par deux facteurs principaux : **l'opportunité** et **l'enchaînement des contributions**. Plus précisément :

1. **l'allocation de paramètres de transmission** : c'est un problème encore ouvert et qui présente encore maintenant un grand potentiel pour proposer de nouveaux mécanismes et stratégies ;
2. **l'étude de la liaison descendante** : cette thématique est en étroite liaison avec la précédente, notre contribution présente deux objectifs : 1) fournir un outil facile et pratique pour pouvoir tester les stratégies d'allocation de paramètres de transmission dans un environnement expérimental réel en comblant les lacunes de la simulation, et 2) analyser et étudier l'impact des communications sur la liaison descendante dans les réseaux [LORAWAN](#) ;
3. **l'itinérance** : cette orientation présente une opportunité de se positionner sur un problème peu abordé, voir non traité jusque-là. En effet, à la rédaction de ce manuscrit, aucune solution n'a été menée à terme ni déployée pour prendre en charge l'itinérance active (*Handover Roaming*) dans les réseaux [LORAWAN](#).

1.3.2 Validation multi-approche

Dans cette thèse, nous nous intéressons non seulement à la conception de ces différentes solutions, mais également à leur validation. Pour ce faire, nous adoptons une démarche combinant plusieurs approches et appliquées à différents scénarios. En effet, nous avons validé nos intuitions et nos propositions avec des expérimentations sur des équipements et des déploiements réels, ce qui est le cas du choix du modèle de propagation, de notre solution d'itinérance et de l'état de la liaison descendante. Nous avons évalué nos stratégies d'allocation de paramètres de transmission par simulation et en considérant plusieurs scénarios. De plus, pour combler les lacunes de la simulation, nous avons utilisé des données extraites d'un jeu de données pour alimenter notre simulateur. Enfin, nous avons proposé un émulateur, le premier en son genre, dédié à l'itinérance pour pousser plus loin l'évaluation de nos solutions.

1.3.3 Processus de développement

Puisque le déroulement de la thèse est accompagné par le développement de plusieurs outils et composants logiciels, à savoir un simulateur [LORAWAN](#), un émulateur d'itinérance et la solution d'itinérance, nous avons mis à disposition le code source sur GitHub [61].

1.4 Contributions et structure du manuscrit

Les travaux de cette thèse sont présentés en trois parties, chacune traitant d'un problème crucial pour les réseaux **LoRaWAN** et apportant des réponses aux défis soulevés. La première partie présente nos contributions sur l'allocation des paramètres radio, la deuxième partie apporte une solution à la gestion de l'itinérance et enfin la dernière partie expose un *framework* dédié aux communications sur la liaison descendante.

1.4.1 L3SFA : Stratégie d'équilibrage de charge pour l'allocation des facteurs d'étalement dans les réseaux LoRaWAN

Dans la première partie, nous nous intéressons à l'allocation des paramètres radio pour l'optimisation du rendement des réseaux **LoRaWAN** à forte charge dans des réseaux de grande taille.

Nous proposons une nouvelle stratégie appelée **L3SFA**, qui améliore le taux de délivrance de paquets global (**DER**). **L3SFA** prend en compte la charge globale du système en termes de trafic et de nombre de terminaux pour allouer les facteurs d'étalement aux équipements. Son objectif principal est d'équilibrer la charge sur les différentes classes correspondant aux de facteurs d'étalement afin d'améliorer le **DER**.

Par la suite, pour valider notre proposition, nous avons considéré deux approches. Dans la première, les terminaux sont uniformément répartis dans l'espace. L'originalité de cette évaluation, en plus de la stratégie d'allocation des facteurs d'étalement proposée, est l'utilisation du modèle de propagation **3GPP** pour décrire le comportement des liens. Pour autant que nous le sachions, c'est la première fois qu'un tel modèle est utilisé pour les réseaux **LoRaWAN**. Nous plaçons pour ce choix, suite à des expérimentations que nous avons menées sur le campus de l'université de Paul Sabatier et à la confrontation des résultats avec ceux des modèles existants. Pour remédier à certaines des limites de la simulation, nous avons également envisagé une deuxième approche. Dans celle-ci, les terminaux émuloés sont approvisionnés à partir de trames extraites d'un jeu de données **LoRaWAN** recueillies lors d'un déploiement réel.

Nous avons étendu **L3SFA** grâce à une stratégie de contrôle de la puissance de transmission en proposant **L3SFA-TPC** qui diminue la consommation d'énergie globale en conservant les mêmes améliorations de performance en termes de **DER**.

1.4.2 LoRaRoam : Itinérance dans les réseaux LoRaWAN

Dans la deuxième partie, nous nous intéressons à l'itinérance active ou en anglais *Handover Roaming* dans les réseaux **LoRaWAN** pour répondre aux exigences de nouvelles applications qui requièrent la mobilité des terminaux.

Pour concrétiser cet objectif, nous avons conçu une extension de l'architecture **LoRaWAN** sans compromettre les mécanismes existants afin d'interconnecter différents opérateurs. Notre solution est conçue pour être intégrée dans les infrastructures déployées avec un minimum de configuration préalable. Nous avons tout d'abord introduit une souscription du réseau au service d'itinérance et une méthode d'approvisionnement des terminaux. Ensuite, nous avons conçu un ensemble de mécanismes d'itinérance, à savoir la résolution **DNS** et la migration du contexte entre le réseau domestique et le réseau visité (partenaire). Pour démontrer la faisabilité de notre conception, nous avons considéré des déploiements réels pour mener des expérimentations. Nous avons évalué les performances du système en termes de délai d'association dans plusieurs configurations et vérifié sa capacité à répondre aux exigences des terminaux de classe A.

Nous avons également conçu et mis en œuvre un émulateur d'itinérance **LoRaWAN** pour démontrer l'efficacité de notre solution. À cette fin, nous avons étudié le temps de réponse requis pour la résolution **DNS** et la migration du contexte du terminal, ainsi que l'utilisation du **CPU** et de la mémoire par les nouveaux composants ajoutés à l'architecture **LoRaWAN**.

Il est à noter que, au moment de la rédaction de ce manuscrit, cette contribution portant sur le *Handover Roaming* dans les réseaux **LoRaWAN** constitue la première implantation et le premier déploiement dans un environnement de test réel. Ce travail a été présenté devant la communauté de la LoRa Alliance.

1.4.3 LoRaDL : un *framework* pour prendre en charge les communications sur la liaison descendante LoRaWAN

Dans la troisième partie, nous portons notre intérêt aux communications sur la liaison descendante dans les réseaux **LoRaWAN**. L'objectif de cette contribution est de proposer un *framework* appelé **LoRaDL** permettant à un serveur de réseau **LoRaWAN** d'envoyer des commandes **MAC** afin de gérer les terminaux. Notre conception permet également aux applications **IoT** de transmettre des trames sur la liaison descendante jusque-là réservée aux commandes de configuration **MAC**. Elle facilite la mise en œuvre de nouvelles stratégies d'allocation de paramètres radio et fournit un cadre expérimental pour l'analyse des performances de la liaison descendante. Par la suite, pour montrer l'utilité et le

potentiel de notre outil, nous avons mené une étude expérimentale approfondie sur les performances de la communication en liaison descendante dans les réseaux [LORAWAN](#) en utilisant [LoRADL](#).

PRINCIPES FONDAMENTAUX DES RÉSEAUX LoRaWAN

Pour appréhender pleinement les réseaux LoRaWAN, nous allons dans un premier temps nous pencher sur la pile protocolaire et les choix technologiques. Comme le montre la figure 2.1, on peut distinguer trois couches constituant cette pile :

- **LoRa** représente la couche physique (PHY) et correspond à la modulation sans fil utilisée sur le lien de communication à longue distance ;
- **LoRaWAN** est un protocole d'accès au *medium* qui constitue donc la couche MAC des réseaux LoRaWAN. Ce protocole, libre de droit, standardisé et maintenu par la *LoRa Alliance*, fournit des services de communications bidirectionnelles sécurisées, de mobilité et de localisation ;
- **Application** représente la couche applicative qui permet de gérer l'envoi de données des applications.

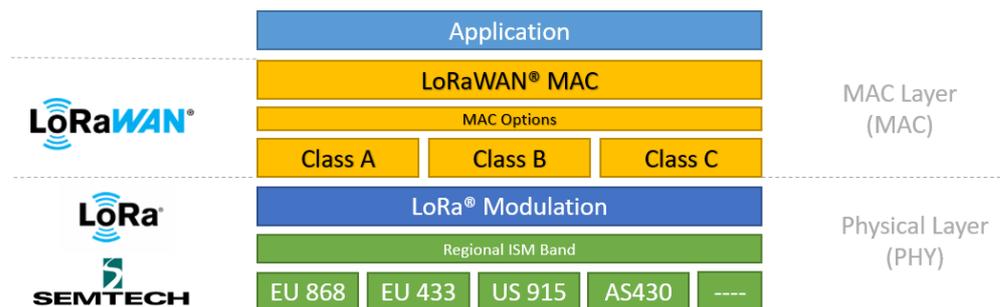


FIGURE 2.1 Pile Technologique des Réseaux LoRaWAN [3].

2.1 LoRa

2.1.1 La modulation LoRa

LoRa (*Long Range*) est une modulation propriétaire à étalement de spectre, appelée *Chirp Spread Spectrum* (**CSS**), développée par Semtech en 2004 et incorporée comme couche physique à la technologie **LoRaWAN**. Cette technique de modulation est à l'origine des principales caractéristiques offertes par les réseaux **LoRaWAN** : longue portée et transmissions à faible puissance, grande robustesse et résistance à l'effet Doppler. La qualité de transmission utilisant la modulation **LoRa** est déterminée par la combinaison optimale d'un ensemble de paramètres de transmission détaillés dans la section suivante.

2.1.2 Les paramètres de transmission radio

Une transmission radio dans **LoRaWAN** peut être caractérisée par une combinaison de paramètres de configuration. Nous pouvons en définir principalement cinq qui ont un impact direct sur la consommation d'énergie, la portée de transmission et la robustesse.

- **Facteur d'Étalement (SF)** : Le facteur d'étalement ou en anglais *Spreading Factor* (**SF**) est le paramètre qui contrôle la façon avec laquelle les *chirps* sont étalés. Un facteur d'étalement plus faible diminue le rapport signal à bruit (**SNR**), et donc la sensibilité et la portée, mais diminue également le temps de transmission de la trame. Les valeurs du facteur d'étalement sont comprises entre 7 et 12 ; le choix est fait en fonction d'un compromis entre la consommation d'énergie et la robustesse.
- **Puissance de Transmission (TP)** : LoRa offre la possibilité d'envoyer des trames avec différentes puissances d'émission dans la plage [2dBm;14dBm] pour la région européenne. L'utilisation d'une **TP** élevée améliore la probabilité de réception des trames. En revanche, elle augmente la consommation d'énergie de l'émetteur et le niveau d'interférence.
- **Taux de Codage (CR)** : De façon quelque peu surprenante **CR** caractérise (une proportion de) la taille prise par le code de correction d'erreur ajouté à une trame avant sa transmission pour offrir une protection contre les erreurs et les interférences. LoRa utilise quatre valeurs de **CR**. Le taux de codage τ vaut alors $\tau = \frac{4}{4+CR}$. τ prend ses valeurs dans (4/5, 4/6, 4/7, 4/8). Il ne fait aucun doute que l'utilisation d'un taux de codage τ plus faible offre une plus grande immunité contre les erreurs dues aux interférences, mais augmente le temps de vol et, par conséquent, la consommation d'énergie.

- **Bande Passante (BW)** : la bande passante représente la largeur d'une bande continue de fréquences. La communication LoRa peut s'effectuer en utilisant l'une des valeurs suivantes : $BW \in \{125, 250, 500\} \text{ kHz}$ en fonction des paramètres régionaux pour les liaisons montantes et descendantes. Plus cette largeur de bande est élevée, plus le temps de vol est court, mais plus la sensibilité est faible. Cela permet d'atteindre un débit de données élevé au prix d'un bruit plus important.
- **Fréquence de la porteuse (CF)** : c'est la fréquence de l'onde porteuse, mesurée en Hertz, qui est modulée pour transmettre des signaux.

Le tableau 2.1 ci-dessous résume les valeurs des paramètres de transmission LoRa utilisées en Europe.

TABLE 2.1 Paramètre de transmission radio LoRa en Europe

Paramètres	Valeurs
Facteur d'Étalement	7, 8, 9, 10, 11, 12
Puissance de Transmission	2, 5, 8, 11, 14 [dBm]
Bande Passante	125 kHz, 250 kHz, 500 kHz
Taux de Codage	4/5, 4/6, 4/7, 4/8
Fréquence de la porteuse	868.1, 868.3, 868.5, ...

2.2 LoRaWAN

LoRaWAN est un protocole de contrôle d'accès au support libre de droit, standardisé par la *LoRa Alliance*, qui fonctionne par-dessus la couche physique **LoRa**. Il apporte un ensemble de mécanismes pour gérer la communication entre les terminaux et les passerelles.

2.2.1 LoRaWAN : Architecture du Réseau

Un réseau **LoRaWAN** est représenté sous forme d'une topologie en étoile d'étoiles. La figure 2.2 illustre l'architecture ce réseau telle que définie dans la spécification. Elle comporte principalement 4 composants [50] : les terminaux, les passerelles (ou *gateways*), les serveurs réseaux (ou *network servers*) et les serveurs d'applications (ou *application servers*).

- **Terminaux** : ce sont des capteurs ou des actionneurs dotés d'un émetteur-récepteur **LoRa** qui permet d'envoyer des messages sans fil, modulés par **LoRa**, aux passerelles sur la voie montante ou de recevoir, en retour, des messages des passerelles sur la voie descendante.

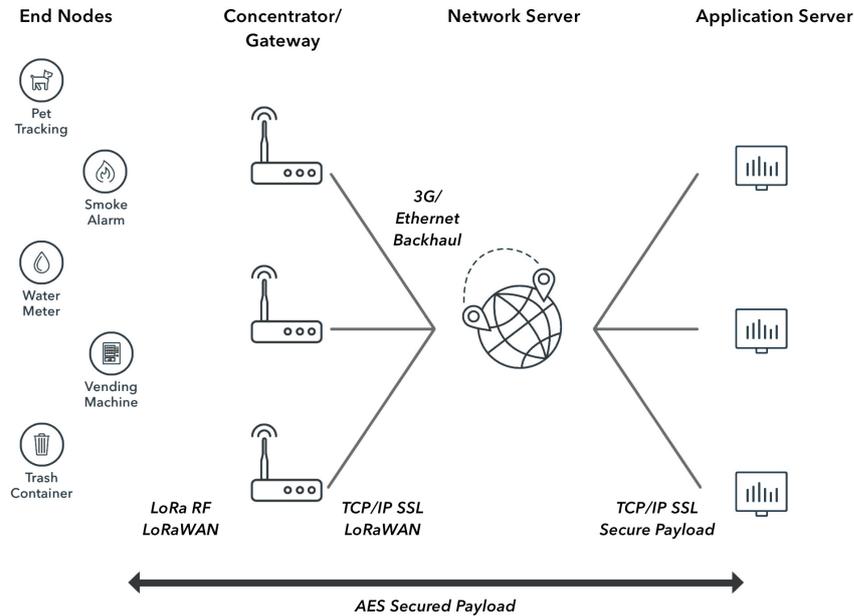


FIGURE 2.2 Architecture des réseaux LoRaWAN [103].

- **Passerelles** : elles représentent un pont entre les terminaux et le réseau. Pour recevoir les trames émises par les terminaux, les passerelles sont équipées d'un concentrateur **LoRa** et peuvent, en substance, être considérées comme une sorte de relais pour acheminer les trames au Network Server.
Dans un réseau **LoRaWAN**, les trames envoyées par les terminaux sont reçues par toutes les passerelles à leur portée, qui les relaient à leur tour au Network Server associé.
- **Network Server (NS)** : il représente la composante intelligente d'un réseau **LoRaWAN** dédié à la gestion et au contrôle de l'infrastructure. Il filtre les trames dupliquées provenant de différentes passerelles, effectue des contrôles de sécurité. Il envoie alors des accusés de réception et des commandes de contrôle aux passerelles, qui sont ensuite envoyés aux terminaux sur le lien descendant. Enfin, il achemine les trames destinées à un Application Server.
- **Application Server (AS)** : il traite les messages de données applicatives envoyés par les terminaux. Il engendre également toutes les données utiles de la liaison descendante de la couche application et les envoie aux terminaux connectés par l'intermédiaire du **NS**. Un réseau **LoRaWAN** peut comporter plus d'un **AS**. Les données applicatives collectées peuvent être exploitées en appliquant des

techniques telles que l'apprentissage automatique et l'intelligence artificielle pour apporter des solutions et construire des applications.

2.2.2 LoRaWAN : classes de terminaux

La spécification **LoRaWAN** [5–7] définit trois types de terminaux ou classes appelées A, B et C. Tous les terminaux doivent mettre en œuvre la classe A, tandis que les classes B et C sont des extensions optionnelles de la spécification des terminaux de classe A. Toutes les classes de terminaux prennent en charge la communication bidirectionnelle (liaison montante et descendante).

Classe A

Tous les terminaux doivent prendre en charge la mise en œuvre de la classe A. La communication de classe A est toujours initiée par le terminal qui peut envoyer un message sur la liaison montante à tout moment. Une fois cette transmission terminée, le terminal ouvre deux courtes fenêtres de réception pour écouter les liaisons descendantes. Il y a un délai entre la fin de la transmission sur la liaison montante et le début des fenêtres de réception (RX1 et RX2 respectivement). Le serveur de réseau peut répondre pendant la première fenêtre de réception (RX1), ou pendant la seconde fenêtre de réception (RX2), mais il n'utilise pas les deux fenêtres. Dans le cas où le serveur de réseau ne répond pas pendant ces deux fenêtres de réception, la prochaine transmission sur la liaison descendante ne pourra pas avoir lieu avant la prochaine transmission montante. La figure 2.3 illustre les différents intervalles de temps.

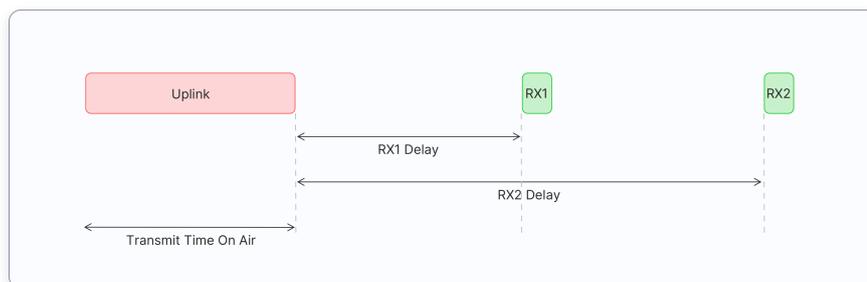


FIGURE 2.3 Fenêtres de réception de classe A [4].

Les terminaux de classe A sont souvent alimentés par des batteries, ont la plus faible consommation d'énergie, passent la plupart du temps en mode veille et ont une latence élevée en liaison descendante (pour recevoir sur la liaison descendante, le terminal doit envoyer une trame sur la liaison montante).

Classe B

En plus des fenêtres de réception initiées par la classe A, les terminaux de la classe B ouvrent des fenêtres de réception planifiées pour recevoir les messages de liaison descendante du serveur de réseau. En utilisant des balises synchronisées transmises par la passerelle, les terminaux ouvrent périodiquement des fenêtres de réception. Le temps entre deux balises est appelé période de balise. Le terminal ouvre des “créneaux ping” de liaison

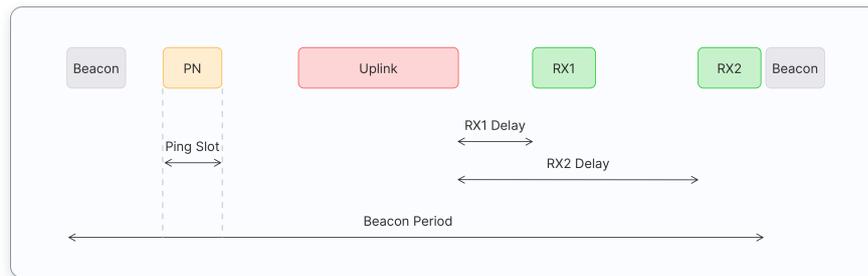


FIGURE 2.4 Fenêtres de réception de classe B [4].

descendante à des moments programmés pour recevoir des messages. Les terminaux de la classe B ouvrent également des fenêtres de réception après avoir envoyé une trame en liaison montante, comme illustré dans la figure 2.4.

Classe C

Les terminaux de la classe C étendent la classe A en gardant les fenêtres de réception ouvertes à moins qu'ils ne soient en train de transmettre, comme le montre la figure 2.5. Cela permet une communication à faible latence, mais consomme beaucoup plus d'énergie que pour les terminaux de la classe A. Les terminaux de la classe C sont souvent alimentés par le secteur et n'ont aucune latence en liaison descendante grâce à l'ouverture continue de fenêtres de réception.

2.2.3 LoRaWAN : Activation des Terminaux

La couche **LoRaWAN** offre un ensemble de mécanismes et de fonctionnalités permettant d'assurer des transmissions sûres et robustes. Elle prend en charge la procédure d'activation d'un terminal pour l'associer au réseau. Plus précisément, chaque nœud doit être authentifié (activé) pour rejoindre et être associé à un réseau LoRaWAN. Deux méthodes d'activation sont proposées : par voie hertzienne (*Over The Air Activation Otaa*) ou bien par personnalisation (*Activation by Personalization ABP*).

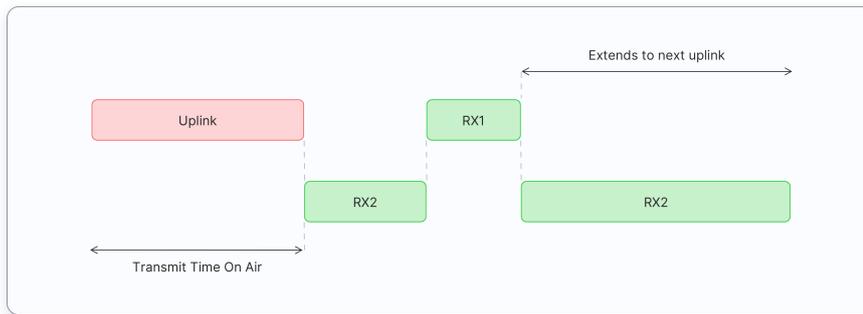


FIGURE 2.5 Fenêtres de réception de classe C [4].

Les terminaux utilisant la méthode **OTAA** reçoivent des clés racines. Lors de l'activation **OTAA**, un terminal exécute une procédure d'adhésion à un réseau **LoRaWAN**, au cours de laquelle un DevAddr dynamique est attribué à un nœud et les clés racines sont utilisées pour déduire les clés de session. Par conséquent, le DevAddr et les clés de session changent à l'établissement de chaque nouvelle session.

À l'inverse, lors d'une activation **ABP**, un DevAddr fixe et des clés de session pour un réseau présélectionné sont codés en dur dans le terminal; ils sont donc inchangés pendant toute la durée de vie d'un terminal **ABP**. Avec cette méthode, un terminal saute la procédure d'adhésion, ce qui semble plus simple, mais l'utilisation d'**ABP** présente un certain nombre de limitations en termes de sécurité et la possibilité d'implanter des mécanismes d'itinérance. Dans la pratique, cette méthode est utilisée à des fins de tests. Ces limitations sont contournées par l'introduction de l'activation **OTAA**.

2.2.4 LoRaWAN : Sécurité

Abréviation	Signification
AppKey	Application Key
NwkKey	Network Key
AppSKey	Application Session Key
FNwkSIntKey	Forwarding Network Session Integrity Key
SNwkSIntKey	Serving Network Session Integrity Key
NwkSEncKey	Network Session Encryption Key

TABLE 2.2 Abréviation des clés de sécurité.

LoRaWAN fournit un mécanisme de sécurité efficace combinant le chiffrement et la vérification du code d'intégrité du message **MIC** (*Message Integrity Check*) [49]. Pour cela, les terminaux utilisent des clés de chiffrement de 16 octets qui correspondent à la

clé racine AppKey pour la spécification LoRaWAN 1.0.x ou aux clés racine NwkKey et AppKey pour la spécification LoRaWAN 1.1. Elles sont stockées à la fois sur le NS et sur le nœud, de manière à empêcher leur extraction et leur réutilisation par des acteurs malveillants. Ces clés correspondent respectivement aux clés grises et vertes illustrées sur la figure 2.6. Pour faciliter la lecture de cette section, la table 2.2 donne la signification des abréviations des clés de sécurité utilisées. Dans le reste de ce manuscrit, nous nous

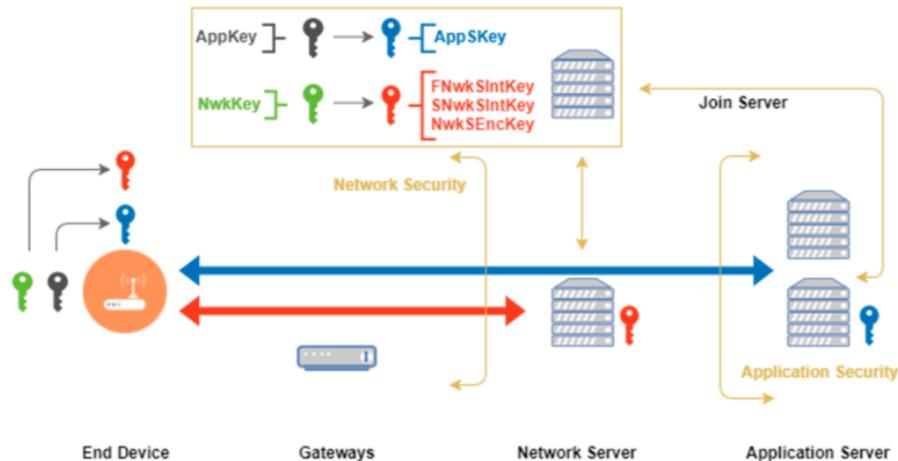


FIGURE 2.6 Architecture des réseaux LoRaWAN.

concentrons sur la spécification LoRaWAN 1.1 dans laquelle la clé de chiffrement AppKey, clé bleue sur la figure 2.6, est utilisée pour déduire une clé de chiffrement de session appelée AppSKey de 16 octets qui sera par la suite utilisée pour chiffrer les données utiles de l'application au niveau du terminal et au niveau du serveur d'application. D'autre part, la clé de chiffrement NwkKey, clé rouge sur la figure 2.6, est utilisée pour déduire des clés de session de 16 octets appelées FNwkSIntKey, SNwkSIntKey et NwkSEncKey afin de fournir une couche de sécurité entre un terminal et un serveur de réseau. Ces clés sont utilisées pour le calcul du MIC (FNwkSIntKey, SNwkSIntKey) et le chiffrement des commandes de contrôle d'accès au support (NwkSEncKey). L'utilisation de deux clés différentes permet une authentification en deux étapes, ce qui offre la possibilité d'une délégation éventuelle de la couche MAC à un autre opérateur sans compromettre la confidentialité de la couche applicative.

2.2.5 LoRaWAN : Itinérance

Comme classiquement, l'itinérance **LoRaWAN** permet à différents réseaux dans différentes régions de combiner leurs zones de couverture afin de fournir une extension de

connectivité aux terminaux et aux passerelles LoRa au-delà de leurs réseaux domestiques. Outre cette extension, **LoRaWAN** offre également une fonction d’itinérance exclusive appelée “densification du réseau”. Cela permet à plusieurs opérateurs dont la couverture se chevauche de fournir une couverture plus large et de prendre en charge la mobilité des terminaux.

LoRaWAN autorise deux types d’itinérance [79] : l’itinérance passive et l’itinérance active (ou itinérance *handover*).

- **L’itinérance passive** consiste à rediriger les trames d’un serveur de réseau visité (**vNS**), appelé **Forwarding Network Server** (**fNS**), vers le serveur du réseau domestique appelé **Home Network Server** (**hNS**). Dans ce contexte, pendant la communication, les terminaux ignorent totalement qu’ils sont connectés à un **vNS**. Par conséquent, aucune *RejoinRequest* n’est nécessaire. Ce type d’itinérance est pris en charge dans LoRaWAN v1.0.x [5], [6].
- A l’inverse, LoRaWAN v1.1 [7] permet de mettre en place un accord d’**itinérance Handover** en plus de la version passive. Lors d’une itinérance *handover*, les trames provenant des terminaux sont reçues sur un vNS appelé **Serving Network Server** (**sNS**). Une fois reçues, les trames sont déchiffrées à l’aide des clés du réseau et les données chiffrées de l’application sont extraites et envoyées au **hNS**. Le **vNS** gère donc la couche **MAC**, d’où la nécessité d’un *RejoinRequest*.

En effet, LoRaWAN v1.0.x ne permet que l’itinérance passive en raison de l’absence de mécanismes de sécurité qui sont améliorés dans LoRaWAN v1.1 (voir section 2.2.4).

Première partie

Allocation de Ressources Radio

Une transmission entre les terminaux et les passerelles dans les réseaux [LORAWAN](#) est configurable par un ensemble de paramètres radio. Ces paramètres deviennent cruciaux et ont un impact significatif sur la qualité de la communication en termes de capacité du réseau à gérer un grand nombre de terminaux, de taux de collision, de taux de délivrance de messages, etc. En effet, la sélection d'une combinaison de paramètres peut influencer fortement les performances obtenues.

Parmi tous ces paramètres, le facteur d'étalement est l'un des plus importants, car il a un impact direct sur le débit global. Il ne fait aucun doute que le choix du bon facteur d'étalement pour envoyer une trame contribue fortement à minimiser les interférences et permet la coexistence de plusieurs terminaux dans la même région sans détériorer de manière significative la qualité du réseau.

Organisation de la partie

Cette partie comprend deux chapitres. Dans le premier chapitre, nous présentons un état de l'art des stratégies d'allocation des facteurs d'étalement et les défis associés. Nous présentons également une classification de ces stratégies et les métriques les plus répandues pour leur évaluation.

Dans le deuxième chapitre, nous présentons [L3SFA](#) et [L3SFA-TPC](#), nos stratégies d'allocation de ressources radio pour l'optimisation des taux de réception des trames au niveau des passerelles et la consommation énergétique des terminaux finaux.

ÉTAT DE L'ART

Dans ce chapitre, nous aborderons les défis scientifiques liés à l'allocation des paramètres de transmission radio. Puis, nous exposerons les métriques d'évaluation les plus pertinentes utilisées dans la littérature pour l'analyse des performances des réseaux **LoRaWAN**. Enfin, nous recenserons les différents travaux menés au cours de ces dernières années et leur classification permettra le positionnement de notre solution.

3.1 Défis liés aux facteurs d'étalement

3.1.1 Densification des réseaux LoRaWAN

La densification est la capacité des réseaux **LoRaWAN** à prendre en charge un grand nombre d'objets connectés autour d'une seule passerelle. Plus grande sera cette capacité et plus le réseau passera à l'échelle en termes de charge de trafic et de nombre de terminaux. La densification de ces réseaux est cruciale pour plusieurs applications telles que les villes et les bâtiments intelligents. Pour cela, deux solutions sont envisageables : augmenter le nombre de passerelles ou bien optimiser la configuration du réseau en adoptant une bonne stratégie d'allocation de paramètres radio. En effet, une telle stratégie peut contribuer fortement à minimiser les interférences, ce qui permet la coexistence de plus de terminaux dans la même région.

De nombreux chercheurs ont étudié les problèmes liés au passage à l'échelle du réseau **LoRaWAN**. L'augmentation du nombre de terminaux entraîne une augmentation des collisions et une diminution potentielle du débit écoulé par le réseau. Dans [69], les auteurs ont développé un modèle d'interférence pour évaluer l'impact du terminal interférant sur une transmission en cours. D'après les résultats de la simulation dans [69], lorsque le

nombre d'émetteurs par passerelle augmente, le débit diminue, car l'augmentation de la taille du réseau entraîne une plus grande perte de paquets.

3.1.2 Orthogonalité imparfaite des facteurs d'étalement

Dans [13], les auteurs ont proposé un schéma d'allocation de facteurs d'étalement en tenant compte de l'orthogonalité imparfaite. Dans [13] et [133], les auteurs ont également considéré l'orthogonalité imparfaite dans le cadre de l'analyse **LoRAWAN**. Les auteurs ont mené une analyse numérique et expérimentale pour démontrer le risque de collision même dans le cas où les paquets sont transmis avec des **SF** différents. Selon [133], l'orthogonalité imparfaite réduit le débit du réseau ou la probabilité de succès dans la mesure où elle entraîne davantage de collisions dans le réseau.

3.1.3 Portée de la communication

Le protocole **LoRAWAN** est connu pour permettre une communication à longue portée. Grâce à la modulation **LoRA**, les communications sont plus résistantes aux interférences, ce qui ouvre un nouveau spectre à des applications telles que les villes intelligentes, l'industrie 4.0 ou l'agriculture pour lesquelles les terminaux peuvent se retrouver en intérieur ou en extérieur. Dans des environnements de déploiement aussi diversifiés, les pertes de propagation et l'atténuation du signal doivent être contrées afin d'améliorer la qualité du signal reçu, ce qui conduit à une amélioration de la portée de communication [44], [48].

Pour analyser la portée de la communication, les auteurs dans [48] ont mené de vastes campagnes de mesures en intérieur et en extérieur, dans des environnements urbains et ruraux. Sur la base de ces mesures, des modèles de propagation sont proposés pour les communications **LoRAWAN** et comparés aux modèles empiriques largement utilisés. Ils sont par la suite utilisés pour étudier la couverture. Les résultats montrent que des portées de couverture allant jusqu'à 8 et 45 km ont été obtenues dans les zones urbaines et rurales, respectivement.

3.1.4 Accès multiple au canal

Les réseaux **LoRAWAN** visent à permettre à plusieurs milliers de terminaux de communiquer simultanément, selon les besoins des applications. En plus de l'allocation optimale de la ressource (**SF**, **BW**, **CR** et **TP**), fournir un mécanisme efficace pour l'accès multiple aux canaux de transmission disponibles améliore les performances de ces réseaux et leur passage à l'échelle [111].

3.2 Mesures de performance

3.2.1 Taux de délivrance

Le taux de délivrance ou en anglais le *Data Extraction Rate* (**DER**) [31] est défini comme le rapport entre le nombre de messages reçus avec succès par la passerelle LoRa et le nombre de messages émis par le terminal LoRa dans une fenêtre d'estimation. Pour un système **LoRAWAN**, le **DER** [31] peut être exprimé comme suit en utilisant les formules d'accès d'ALOHA sous des hypothèses de trafics poissonniens :

$$DER = e^{-2N \cdot T_f \cdot \lambda} \quad (3.1)$$

où N est le nombre de terminaux utilisant $SF = f$, T_{SF} le temps de vol des paquets en utilisant $SF = f$, et λ le taux de transmission des paquets pour chaque terminal. Le **DER** est la mesure de performance la plus courante pour évaluer les réseaux **IoT** et les réseaux **LoRAWAN** en particulier.

3.2.2 Consommation énergétique

Dans [101], les auteurs ont considéré la métrique de performance de la consommation d'énergie pour l'évaluation de la stratégie d'allocation optimale de **SF**. Le modèle énergétique considéré dans [101] dispose de deux modes : (1) mode transmission de données et (2) mode sommeil. Selon le modèle énergétique considéré, la consommation moyenne $W(s)$ en mAh sur une heure pour tous les terminaux peut être définie comme suit :

$$W(s) = \frac{3600}{N} \sum_{i=1}^N \frac{T_f}{T_i} \cdot I_{tx} + \left(1 - \frac{T_f}{T_i}\right) \cdot I_{slp} \quad (3.2)$$

où I_{tx} et I_{slp} indiquent respectivement l'intensité du courant pour la transmission et en mode veille, T_i est un intervalle de transmission moyen pour le i ème terminal, T_f et N sont respectivement le temps de vol du paquet en utilisant $SF = f$ et le nombre de terminaux. La consommation de courant sera doublée pour les terminaux avec $SF = f + 1$ par rapport à ceux utilisant $SF = f$.

3.2.3 Débit

Le protocole **LoRAWAN** repose sur le schéma ALOHA. Dans les travaux de [81], les auteurs ont effectué une analyse détaillée du débit du réseau **LoRAWAN**. Pour le

modèle considéré, le débit de la liaison montante est défini par l'équation 3.3 :

$$S = \sum_i r_i P_i^{data} \quad (3.3)$$

$$P_i^{data} = e^{-2r_i T_i^{data}} \cdot e^{-2r_i T_i^{ACK1}} \quad (3.4)$$

P_i^{data} , donné dans l'équation 3.4, est constitué de deux termes : $e^{-2r_i T_i^{data}}$ est la probabilité de collision de la i ème transmission avec une autre transmission de la liaison montante sur le même canal, et $e^{-2r_i T_i^{ACK1}}$ indique la probabilité qu'une transmission $ACK1$ était déjà en cours sur le canal i lorsqu'une transmission de la liaison montante commence. T_i^{data} , et T_i^{ACK1} sont respectivement le temps de transmission des données et des ACK sur le canal de la i ème transmission. r_i est le débit d'arrivée des paquets sur ce même canal.

3.2.4 Probabilité d'interruption

La probabilité d'interruption [15] est définie comme la distance du terminal à laquelle le **SNR** reçu passe en dessous de la valeur seuil définie dans le tableau 3.2. Pour les réseaux **LoRaWAN**, le **SNR** peut être défini comme suit :

$$SNR = \frac{TP \|h_1\|^2 g(d_i)}{\eta} \quad (3.5)$$

où **TP** est la puissance de transmission, η est la variance du bruit gaussien, h_1 est le coefficient de canal et $g(d_i)$ est la fonction d'atténuation de la propagation du signal en fonction de la distance d_i du terminal i à la passerelle. La communication LoRa entre les terminaux et la passerelle est réussie lorsque le **SNR** du signal reçu est supérieur au seuil de réception requis pour le **SF** donné, noté q_{SF} . La probabilité d'interruption de service peut être exprimée comme suit :

$$H_1 = \exp\left(\frac{\eta q_{SF}}{TP_1 g(d_1)}\right) \quad (3.6)$$

La distance entre le terminal concerné et la passerelle est d_1 .

3.2.5 Probabilité de couverture

La probabilité de couverture P_c pour un terminal est définie comme la probabilité qu'un terminal puisse atteindre un certain seuil β du rapport signal à bruit plus interférence

(*signal-to-interference-plus-noise ratio* SINR). Cette probabilité peut être exprimée comme suit :

$$P_c = P[SINR > \beta] \quad (3.7)$$

3.2.6 Taux d'occupation des canaux

L'occupation du canal est la fraction de temps pendant laquelle le canal est occupé. Il est déterminant pour évaluer l'efficacité du système et peut être exploitée pour une adaptation optimale du choix du canal disponible. Dans LORAWAN [38], l'analyse de l'occupation du canal doit tenir compte des différents SF.

Le tableau 3.1 donne un résumé des mesures de performance utilisées dans les différents travaux de recherche.

TABLE 3.1 Différentes mesures de performance des réseaux LoRaWAN.

Mesure de performance	Références
Taux de délivrance	[30, 31, 34, 37, 39, 52, 101, 136]
Consommation énergétique	[13, 101]
Débit	[13, 31, 37, 39, 101, 123, 124]
Probabilité d'interruption	[15, 124]
Probabilité de couverture	[15, 123]
Taux d'occupation des canaux	[38, 58]

3.3 Classification des stratégies d'allocation de facteurs d'étalement

Les auteurs de [96] ont proposé une classification, en 6 catégories, des différents schémas d'allocation de facteur d'étalement dans les réseaux LORAWAN. Cette classification s'avère intéressante pour les concepteurs de réseaux afin de définir les exigences du système et d'identifier la catégorie d'allocation de SF appropriée. Les différentes classes sont illustrées dans la figure 3.1 et détaillées dans ce qui suit.

3.3.1 Allocation aléatoire de SF

Dans l'approche d'allocation de facteur d'étalement aléatoire, ou en anglais *Random spreading factor allocation*, chaque terminal se voit attribuer un facteur d'étalement aléatoire dans l'ensemble {7, 8, 9, 10, 11, 12}. Cette stratégie d'allocation a été discutée

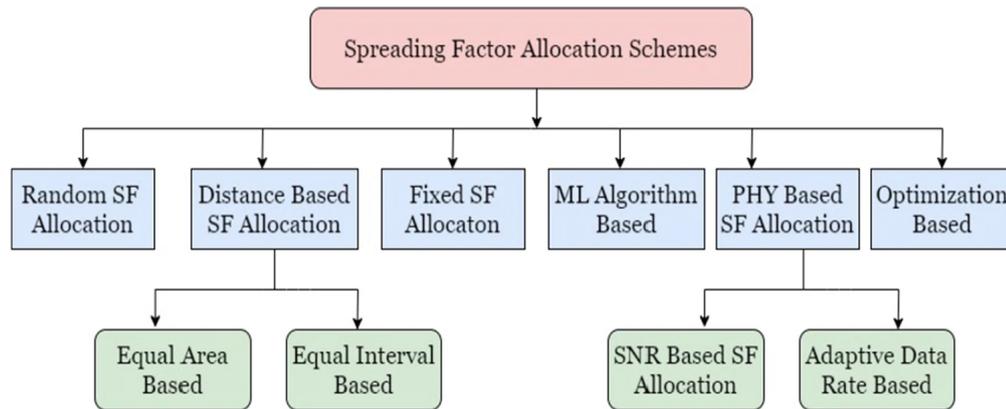


FIGURE 3.1 Classification des différentes stratégies d'allocation de SF [96].

par [52], [128], [70], [90]. La particularité de cette approche est qu'elle ne dépend ni de la répartition des terminaux autour de la passerelle, ni de leurs valeurs de SNR (Signal to Noise Ratio). Les auteurs dans [70] ont proposé une stratégie appelée CE-MAC (*capture effect-MAC*) pour améliorer le débit du réseau. Les auteurs ont mené l'analyse des performances de leur approche en considérant une configuration de réseau dans laquelle les terminaux sont répartis de manière aléatoire et uniforme sur une cellule de 5 km de rayon, et autour d'une seule passerelle située au centre de la cellule. Les résultats de la simulation ont montré que le schéma d'allocation aléatoire de SF est plus performant que la stratégie *Equal-Interval-Based* (EIB) (décrite dans la section 3.3.2) en termes de débit et de taux de collision de trames lorsque le nombre de terminaux dans la cellule est faible. Par conséquent, nous pouvons conclure que l'adaptation du schéma d'allocation aléatoire de SF sera envisageable pour les applications IoT avec un réseau de petite taille.

3.3.2 Allocation de SF fondée sur la distance

Les auteurs dans [70], [90], [94] ont présenté une approche d'allocation de SF reposant sur la distance. Les SF sont affectés aux terminaux en fonction de (la distribution de) leur distance à la passerelle. Plus concrètement, des SF faibles sont affectés aux terminaux proches de la passerelle et vice versa. En effet, les valeurs SNR sont plus faibles pour les terminaux éloignés. Ces terminaux ont donc besoin d'un SF plus élevé pour avoir une bonne connectivité. On peut diviser cette approche selon les deux stratégies suivantes.

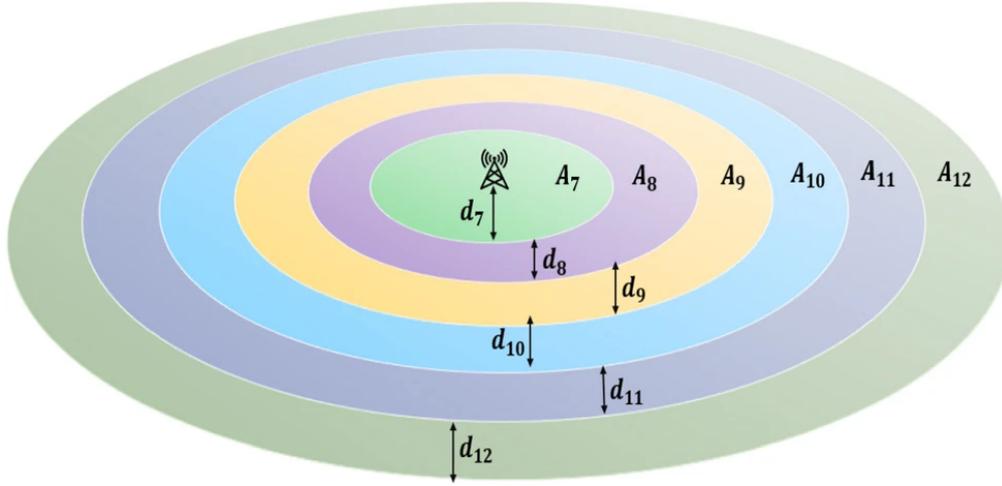


FIGURE 3.2 Cellule LoRaWAN [96].

Allocation Equal-interval-based (EIB)

La stratégie *Equal-interval-based* (EIB) considère un déploiement composé d'une seule passerelle LORAWAN offrant une couverture sur une cellule de rayon d . Le rayon de la cellule est divisé en autant d'intervalles égaux qu'il y a de SF disponibles. Si le rayon de la cellule est d et que le nombre de SF disponibles (7, 8, 9, 10, 11, 12) est de six, d est divisé en six intervalles égaux de longueur $\frac{d}{6}$ chacun. L'équation 3.8 donne la longueur des intervalles d_i pour chaque $SF = i$. Par la suite, l'intervalle le plus proche de la passerelle se voit attribuer le SF le plus bas et ainsi de suite. Cependant, l'inconvénient est que les surfaces considérées sont croissantes et par exemple, dans le cas d'une répartition uniforme des terminaux, un grand nombre se voit attribuer des SF élevés, ce qui augmente le temps de vol et entraîne davantage de collisions.

$$d_7 = d_8 = d_9 = d_{10} = d_{11} = d_{12} = \frac{d}{6} \quad (3.8)$$

Les anneaux affectés aux différents $SF = i$ ont des aires A_{SF} croissantes :

$$A_7 < A_8 < A_9 < A_{10} < A_{11} < A_{12} \quad (3.9)$$

Allocation Equal-area-based (EAB)

A l'inverse de la solution précédente, l'approche *Equal-area-based* (EAB) divise la surface de la cellule en un nombre de SF d'anneaux concentriques de superficie égale,

comme le montre l'équation 3.10.

$$A_7 = A_8 = A_9 = A_{10} = A_{11} = A_{12} \quad (3.10)$$

La taille des intervalles est donc décroissante, comme donnée dans l'équation 3.11. Comme la superficie de chaque zone est la même, on aura le même nombre moyen de terminaux attribués à chaque SF si les terminaux sont répartis uniformément dans la cellule.

$$d_7 > d_8 > d_9 > d_{10} > d_{11} > d_{12} \quad (3.11)$$

L'approche EAB est plus intéressante que EIB pour produire une allocation de SF pertinente. En effet, on observe que EAB attribue des SF plus faibles à un nombre relativement plus élevé de terminaux que EIB. Les SF plus faibles attribués à un plus grand nombre de terminaux entraînent moins de collisions, car ils ont moins de temps de vol. Cette analyse dépend fortement du déploiement et de la répartition des terminaux. L'hypothèse de déploiement uniforme des terminaux ne correspond pas souvent à la réalité.

3.3.3 Allocation fixe de SF

Dans l'approche d'allocation fixe de SF [52], [128], tous les terminaux d'une cellule se voient affecter le même SF. Dans [128], les auteurs ont considéré un déploiement de terminaux dans une cellule de rayon compris entre 6 et 100 mètres autour d'une, deux puis quatre passerelles. Les résultats de l'analyse montrent que lorsque le SF augmente dans la stratégie d'allocation de SF fixe, le taux de délivrance des paquets diminue. En effet avec l'augmentation du SF, le temps de vol de chaque paquet augmente, ce qui entraîne plus de collisions entre les transmissions des terminaux.

3.3.4 Allocation de SF fondée sur un algorithme d'apprentissage machine

Les stratégies reposant sur des algorithmes d'apprentissage machine sont devenues récemment les méthodes les plus étudiées pour l'allocation des facteurs d'étalement dans les réseaux LoRaWAN [39], [15], [136], [124], [108], [38]. Dans cette approche d'allocation de SF, des algorithmes d'apprentissage automatique, tels que *k-means*, l'apprentissage profond, l'apprentissage par renforcement, etc, sont utilisés pour attribuer les SF aux terminaux de manière optimale. Dans cette partie, nous avons recensé les algorithmes d'apprentissage les plus répandus pour répondre à ce problème :

- Le clustering *k-means* est un algorithme d'apprentissage non supervisé. Pour améliorer les performances du système, dans les travaux rapportés dans [39], [15] et [38], les auteurs ont adopté l'algorithme d'apprentissage automatique *k-means* pour allouer les SF appropriés aux terminaux.
- L'approche de l'apprentissage par renforcement est plus judicieuse pour l'allocation de SF dans un réseau dynamique où les paramètres du système sont susceptibles de changer fréquemment. Lorsque cette approche de paramétrage est adoptée dans un réseau LORAWAN, l'algorithme prend l'action appropriée, c'est-à-dire qu'il suggère le meilleur SF possible pour les terminaux en fonction des actions des autres. Dans [124], les auteurs ont proposé une solution fondée sur de l'apprentissage par renforcement pour l'allocation de SF dans le but d'améliorer le débit du réseau et la consommation d'énergie. Enfin, les auteurs dans [108] ont adopté l'apprentissage par renforcement profond pour optimiser l'allocation des ressources radio.
- D'autres travaux se sont orientés vers les arbres de décision utilisés généralement pour résoudre un problème de classification [134]. Dans le contexte des réseaux LORAWAN, cet algorithme est utilisé pour créer un modèle d'apprentissage qui prédit les SF optimaux pour les terminaux ou les classes de terminaux utilisant le même SF.

L'un des défis majeurs pour la conception de solutions fondées sur des algorithmes d'apprentissage est la collecte de données pour obtenir un *dataset* d'entraînement efficace et un jeu de test suffisant. Ce défi peut être contré par une méthode d'apprentissage par renforcement, mais la tâche de définition de l'environnement, de l'ensemble des actions et des états reste d'une complexité non négligeable.

3.3.5 Allocation de SF fondée sur la couche physique

Dans cette section, nous allons exposer les méthodes d'allocation de SF reposant sur les paramètres de la couche physique. On peut distinguer deux catégories.

Allocation de SF fondée sur le SNR

Dans cette approche, les terminaux se voient attribuer des facteurs d'étalement en fonction de leurs valeurs de SNR [94]. Un signal transmis par un terminal éloigné de la passerelle est reçu avec moins de puissance en raison de son affaiblissement lors de sa propagation entre le terminal et la passerelle. L'objectif de cette stratégie est d'attribuer le SF le plus bas possible à chaque terminal afin de respecter le seuil de SNR permettant

une communication réussie avec la passerelle. Les valeurs des seuils de **SNR** correspondant à chaque **SF** sont indiquées dans le tableau 3.2. Le schéma d'allocation de **SF** fondé sur le **SNR** est également connu sous le nom de *Path-loss Based SF Allocation* (PLB).

TABLE 3.2 Seuils de SNR pour une bande passante de 125 kHz en [dB].

SF	7	8	9	10	11	12
SNR	-7.5	-10	-12.5	-15	-17.5	-20

D'après les résultats présentés dans [94], il est observé que l'allocation de **SF** fondée sur le **SNR** améliore le taux de délivrance par rapport à l'**EIB**. En revanche, ce schéma d'allocation des **SF** nécessite plus de traitement du signal et un récepteur plus complexe que les schémas d'allocation aléatoires et à base de distance présentés respectivement dans les sections 3.3.1 et 3.3.2.

Stratégie Adaptive Data Rate

L'un des mécanismes intégré dans la technologie **LORAWAN** est l'*Adaptive Data Rate* (**ADR**) [84], [128], [70], [90] qui contrôle les paramètres de transmission (**SF**, **TP**), appelé aussi *Data Rate* (**DR**), pour la communication ascendante entre le terminal et la passerelle. Une fois le mécanisme activé au niveau de la passerelle et du terminal, le serveur de réseau (**NS**) peut contrôler le terminal et lui envoyer des commandes pour adapter ses paramètres de transmission.

ADR adapte dynamiquement les paramètres de transmission en fonction de l'historique des performances des trames reçues depuis chaque terminal afin de minimiser l'utilisation de la batterie et maximiser le débit. Plus précisément, le serveur réseau (**NS**) collecte les métadonnées des m paquets les plus récents reçus sur la liaison montante, entre autre le **SF** et la **TP**, pour un terminal donné. Ensuite, le **NS** retient le **SNR** maximum reçu et la valeur du **DR** correspondante parmi les m valeurs de **SNR** des paquets reçus sur la liaison montante. Cette valeur est appelée SNR_{max} . Elle est utilisée par la suite pour calculer le SNR_{margin} [1] donné par l'équation 3.12 :

$$SNR_{margin} = SNR_{max} - SNR_{SF} - margin_{default} \quad (3.12)$$

où SNR_{SF} représente le **SNR** seuil pour chaque **SF** donné dans le tableau 3.2 et $margin_{default}$ est une valeur par défaut fixée à 15 par l'algorithme de *The Things Network* [1]. Une fois cette marge calculée, elle est utilisée pour augmenter le débit de données tant qu'il y a assez de marge. S'il y a encore de la marge après avoir atteint le débit maximal, on peut diminuer la puissance de transmission.

3.3.6 Allocation de SF fondée sur l'optimisation

Un autre schéma d'allocation de SF est une approche fondée sur la formulation d'un problème d'optimisation. Dans cette approche, un ensemble d'équations d'évaluation des performances du système doit être optimisé en tenant compte de certaines contraintes.

Dans les travaux effectués dans [31], [13], [101], [30] et [34], cette approche a été adoptée pour allouer de manière optimale les SF aux terminaux répartis dans le réseau. Cette classe de stratégies fournit un modèle théorique et mathématique qui permet de concevoir un mécanisme d'allocation du facteur d'étalement approprié.

Une comparaison entre les différents schémas d'allocation de facteurs d'étalement est résumé dans le tableau 3.3.

TABLE 3.3 Comparaison entre différents schémas d'allocation de facteurs d'étalement.

Allocation de SF	Performance	Adaptation	Complexité	Mobilité	Références
SF fixes	Faible	Non	Faible	Oui	[124, 128]
SF Aléatoires	Modérée	Non	Modérée	Oui	[52, 70, 90, 124, 128]
EIB	Modérée	Non	Modérée	Non	[47, 52, 94]
EAB	Élevée	Non	Modérée	Non	[52, 94]
SF fondé sur le SNR	Élevée	Oui	Élevée	Oui	[37, 113]
ADR	Élevée	Oui	Élevée	Non	[1, 52, 84]
SF fondé sur l'apprentissage	Élevée	Oui	Élevée	Oui	[15, 37, 38, 108, 136]
SF fondé sur l'optimisation	Élevée	Oui	Élevée	Dépend du modèle d'optimisation	[13, 30, 31, 34, 101, 123]

3.4 Travaux les plus pertinents

Ces dernières années, la recherche a été particulièrement productive sur l'allocation des paramètres radio pour la recherche de la meilleure combinaison des paramètres de transmission dans les réseaux **LoRaWAN**. Il ne fait aucun doute qu'un bon réglage du taux de codage, du facteur d'étalement et de la puissance de transmission permettrait d'obtenir un *Data Extraction Rate* (**DER**) global élevé et une utilisation efficace des canaux. Dans les travaux [22], [56] et [27], les auteurs étudient la densification des réseaux **LoRaWAN** en se concentrant sur la manière d'augmenter la capacité du réseau dans les déploiements autour d'une seule passerelle.

D'autres études portent sur l'optimisation d'un critère de performance spécifique par de nouveaux schémas d'allocation de facteurs d'étalement. Plusieurs approches ont été envisagées pour aboutir aux solutions suivantes :

- des algorithmes comme EXP-SF et EXP-AT dans [37] visent à augmenter le débit de données global, EXPLoRa-TS proposé dans [39] effectue une répartition identique de la charge de trafic entre les facteurs d'étalement tout en considérant que chaque appareil final transmet une quantité variable de données en fonction de son profil d'application ;
- dans [116],[118] les auteurs ont modélisé l'allocation des ressources **LoRaWAN** comme un problème de maximisation ou de minimisation résolu à l'aide de métaheuristiques comme les algorithmes génétiques et de méthodes d'apprentissage comme l'apprentissage par renforcement ;
- d'autres études visent l'application de méthodes d'apprentissage automatique [39], [116], [15]. Par exemple, l'apprentissage non supervisé de type *K-means* a été appliqué dans les articles [39] et [15] afin de trouver des clusters appropriés pour l'allocation des **SF**. D'autres articles reposent sur l'apprentissage par renforcement où une passerelle joue le rôle d'un agent pour contrôler l'environnement (appareils finaux) en envoyant des actions (configurations). Des réseaux neuronaux convolutifs ont également été utilisés pour résoudre des problèmes d'optimisation dans [116].

Ces travaux se focalisent essentiellement sur l'évaluation des performances des réseaux **LoRaWAN** tels que [22], [56] sans fournir de nouveaux mécanismes ni de solutions pour en améliorer les performances. En effet, la plupart de ces papiers se penchent sur les questions de densification de réseau, de couverture de communication et d'efficacité énergétique afin de déterminer la durée de vie des batteries, mais ils n'étudient pas les questions fondamentales de LoRa. En effet :

- ils négligent, dans le modèle mathématique ou la simulation, certains phénomènes

physiques LoRa tels que l'effet de capture (interférences intra-SF) et l'orthogonalité imparfaite des SF (interférences inter-SF) [37], [39], [118] qui ont un impact direct sur les performances du réseau ;

- ils utilisent principalement le modèle de *path loss* Log-Distance [116], [9], [117] avec un *path loss* à une distance $d_0 = 40$ mètres égal à 127.41 dB. Cette hypothèse est pénalisante et ne peut permettre qu'un déploiement autour d'une passerelle avec de courtes distances, ce qui contredit l'objectif principal des technologies LPWAN ;
- ils ne prennent pas suffisamment en compte l'impact d'autres facteurs importants, tels qu'une forte charge de trafic ou une grande taille de réseaux (nombre de dispositifs dans les réseaux LoRAWAN à forte charge [52]).

Après avoir pris le recul nécessaire pour passer en revue les solutions proposées, nous avons jugé bon de nous pencher à nouveau sur la question de l'allocation des paramètres radio afin d'élaborer de nouvelles stratégies, notamment pour les systèmes LoRAWAN fortement chargés et des réseaux de grande taille.

3.5 Notre positionnement

Dans cette première partie du manuscrit, nous présentons L3SFA, une solution dédiée à l'allocation de facteurs d'étalement dans les réseaux LoRaWAN. Nous étendons par la suite cette solution en L3SFA-TPC pour prendre en compte la puissance de transmission afin d'améliorer les rendements énergétiques. Nos solutions visent la satisfaction des objectifs suivant :

- **Passage à l'échelle** : proposer de nouvelles solutions pour les réseaux LoRAWAN fortement chargés en considérant à la fois différentes charges de trafic en termes de débit et de nombre de terminaux ;
- **Approche** : proposer des solutions fondées sur une modélisation sous forme d'un problème d'optimisation, utilisation de méta-heuristiques fondées sur les seuils de RSSI et de SNR ;
- **Évaluation** : validation des solutions proposées par simulation en termes de DER et consommation énergétique en :
 - (i) **utilisant le modèle de propagation 3GPP** pour la première fois dans le contexte des réseaux LoRAWAN et validation de ce modèle par des **expérimentations** sur le campus de l'université Paul Sabatier ;
 - (ii) **combinant la simulation et un jeu de données issues d'expérimentations réelles** menées sur la ville d'Anvers en Belgique.

L3SFA : STRATÉGIE D'ÉQUILIBRAGE DE CHARGE POUR L'ALLOCATION DE SF DANS LES RÉSEAUX LoRAWAN

Dans ce chapitre, nous présentons [L3SFA](#), *Load Shifting Strategy for Spreading Factor Allocation*, une stratégie d'allocation de facteurs d'étalement dans les réseaux [LoRAWAN](#) à forte charge, et [L3SFA-TPC](#) une extension [L3SFA](#) pour le contrôle de la puissance de transmission dans le but d'améliorer la consommation énergétique des terminaux.

4.1 Modélisation du problème d'allocation des facteurs d'étalement dans des réseaux LoRaWAN denses

Nous considérons un réseau LoRaWAN composé de N terminaux autour d'une seule passerelle. Chaque terminal est initialisé avec $SF = i$ selon le [RSSI](#) et les seuils [SNR](#) fournis dans le [Tableau 4.2](#). Afin d'allouer un facteur d'étalement dans la gamme SF7 à SF12, pour un dispositif final donné, nous définissons un schéma d'équilibrage de charge.

4.1.1 Décalage des facteurs d'étalement

Les terminaux sont initialisés avec un $SF = j$ en fonction des seuils définis dans le [Tableau 4.2](#) avec $j \in \{7, 8, 9, 10, 11, 12\}$. On note alors N_j le nombre de terminaux positionnés initialement sur le $SF = j$. Il est possible de prendre des SF plus grands que ce SF initial mais pas de plus faibles. On note $M_{j,i}$ le nombre de terminaux pouvant utiliser un $SF = j$ mais décalés à un $SF = i$ plus élevé et $M_{j,j}$ le nombre de terminaux

TABLE 4.1 Définition des variables du modèle d'allocation de SF.

Variable	Description
N	Nombre de terminaux
N_i	Nombre de terminaux utilisant $SF = i$ avant décalage
M_i	Nombre de terminaux utilisant $SF = i$ après décalage
T_i	Temps de vol en utilisant $SF = i$
p	Période de génération de la trame (par terminal)
$M_{j,i}$	Terminals changeant de $SF = j$ vers $SF = i$
$M_{j,j}$	Terminals avec $SF = j$ inchangé

qui conservent leur $SF = j$ inchangé. La définition des variables du modèle d'allocation de SF sont résumées dans le tableau 4.1.

TABLE 4.2 Seuils de SNR et de sensibilité pour une bande passante de 125 kHz en [dB].

SF	SNR	Sensibilité
7	-7.5	-126.5
8	-10	-127.25
9	-12.5	-131.25
10	-15	-132.75
11	-17.5	-133.25
12	-20	-134.5

Pour simplifier la présentation, nous notons M_i le nombre total de terminaux auxquels a été attribué un SF i , après la procédure de décalage. Le processus de décalage est illustré dans la figure 4.2 et M_i peut être calculé comme suit :

$$M_i = \sum_{j=7}^i M_{j,i} \tag{4.1}$$

4.1.2 Charge SF

Soit ρ_i la charge du trafic engendrée par l'utilisation d'un $SF = i$ après le processus de décalage des SF.

$$\rho_i = \frac{M_i \times T_i}{p} \tag{4.2}$$

La valeur de ρ_i est calculée à l'aide de l'équation (4.2), où T_i est le temps de vol correspondant à l'envoi d'une trame avec un $SF = i$ et p est la période de génération de

trame par terminal. On suppose que les trames sont de taille constante et l'intensité de trafic identique pour tous les terminaux.

Par conséquent, sous l'hypothèse d'arrivées poissonniennes, le taux de délivrance de trames DER_i obtenu pour un $SF = i$ avec une charge ρ_i peut être calculé comme suit :

$$DER_i = e^{-2\rho_i} \quad (4.3)$$

4.1.3 Allocation optimale des SF

Nous définissons le problème d'allocation des SF comme un problème d'optimisation. Plus précisément, nous considérons le problème de maximisation formulé dans les équations (4.4a-4.4d).

$$\text{maximize} \quad \frac{1}{N} \sum_{i=7}^{12} DER_i \times M_i \quad (4.4a)$$

subject to

$$M_{j,i} \in \mathbb{N}, \quad (4.4b)$$

$$\sum_{j=i}^{12} M_{i,j} = N_i, \quad (4.4c)$$

$$M_{j,i} = 0 \quad \text{if } j > i \quad (4.4d)$$

La fonction objectif donnée par l'équation (4.4a) vise à maximiser le DER global en fonction du DER_i correspondant à l'utilisation de $SF = i$ et du nombre de terminaux utilisant ce même SF une fois la phase de décalage des facteurs d'étalement terminée.

L'équation (4.4b) apporte une contrainte forte au problème d'optimisation. En effet, le nombre de terminaux décalant leur SF doit être un nombre entier positif. Le processus de décalage n'est possible que si l'équation (4.4d) est satisfaite. En effet, le décalage du SF se fait dans un sens, d'un SF vers des SF supérieurs ; l'autre sens n'est pas possible.

Le processus de décalage est contrôlé par l'équation (4.4c). Cette contrainte garantit que le nombre total de terminaux décalés de $SF = i$ vers des SF supérieurs doit être égal au nombre de terminaux utilisant $SF = i$ avant la phase de décalage. Cela garantit que $\sum_{i=7}^{12} M_i = N$.

Le problème d'optimisation se révèle d'une complexité ardue pour un solveur à solution exacte. Effectivement, d'une part, nous sommes confrontés à un problème de programmation non linéaire en nombres entiers (INLP) et d'autre part, la taille du

réseau est fortement extensible.

Il ne fait aucun doute qu'une recherche exhaustive dans l'espace des solutions n'est pas possible en raison du grand nombre de combinaisons, et aucun algorithme en temps polynomial n'est connu pour être applicable dans un cas général, en particulier lorsque le problème à optimiser ne présente pas une forme linéaire ni convexe.

4.1.4 Heuristiques d'équilibrage de charge et d'allocation de SF

Dans cette section, nous présentons une stratégie d'équilibrage de la charge pour l'allocation des facteurs d'étalement, appelée **L3SFA**. La stratégie proposée vise une bonne allocation de **SF** aux terminaux afin d'obtenir un meilleur **DER**. Cette politique d'allocation comporte trois phases principales (voir Algorithme 1) décrites dans les sections suivantes : 1) Initialisation, 2) Allocation de SF, 3) Ajustement de SF.

Algorithm 1 L3SFA

Input : N : Terminaux

Output : Terminaux avec SF alloués, répartition de SF.

- 1: Initialisation des terminaux
 - 2: Allocation de SF
 - 3: Ajustement de SF
-

Initialisation des terminaux

L3SFA commence par une phase d'initialisation. Durant cette première phase, deux actions sont entreprises : 1) Puisque le mécanisme d'allocation de **SF** est incrémental, tous les terminaux sont initialisés avec un **SF** égal à 7. 2) Les terminaux sont ensuite triés en fonction de leur distance à la passerelle ou de leur valeur de **RSSI**. Cette action intuitive permet d'attribuer en priorité des **SF** faibles aux terminaux les plus proches de la passerelle ou à ceux qui ont les valeurs de **RSSI** les plus faibles. Ceci est détaillé par les lignes 1 à 2 dans l'Algorithme 2.

Allocation de SF

Après l'initialisation, la stratégie **L3SFA** commence la deuxième phase de pré-allocation des facteurs d'étalement.

L'algorithme SF LoRa (lignes 3 à 12) parcourt la liste des terminaux triée, produite lors de la première phase et, pour chaque terminal, un **SF** approprié est attribué en fonction des valeurs de **SNR** et de **RSSI** reçues sur la passerelle. Ces valeurs sont

comparées aux seuils **RSSI** et **SNR** donnés dans le tableau 4.2. La condition d'attribution des **SF** définie à la ligne 7 est nécessaire et suffisante pour garantir une transmission réussie. En effet, un **RSSI** supérieur à la sensibilité assure la réception des trames par la passerelle et combiné à un **SNR** supérieur au seuil de **SNR** du récepteur, les trames seront certainement décodées.

Algorithm 2 SF_LoRa

Input : N : end-devices

Output : N with SF allocated, SF_Distribution

```

1: Sort end-devices by distance  $d$  to the gateway ;
2: Init all end-devices with SF =7 and SF_Distribution to zeros
3: for  $n$  in  $N$  do
4:   SF =  $n.sf$ 
5:   Get SNR and RSSI thresholds for the corresponding SF from the table 4.2
6:   for  $s$  in range (SF,12) do
7:     if  $n.snr < SNR_{sf}$  or  $n.rssi < sensibility_{sf}$  then
8:       if  $n.sf < 12$  then
9:          $n.sf = n.sf + 1$ 
10:      end if
11:     Get SNR and RSSI thresholds for the corresponding SF from the table 4.2
12:   end if
13:    $n.sf = SF\_Shifting(n.sf, SF\_Distribution)$ 
14: end for
15: SF_Distribution[ $n.sf$ ] = SF_Distribution[ $n.sf$ ] + 1
16: end for

```

Ajustement de SF

L'allocation de **SF** effectuée dans la phase deux (Allocation de SF) n'est pas définitive. Cette allocation est contrôlée par la dernière phase du mécanisme **RSSI**.

Avant d'attribuer définitivement un **SF** à un terminal, SF_LoRa effectue un contrôle de charge sur la répartition des **SF** (ligne 13). Ce contrôle est effectué avec l'heuristique SF_Shifting.

L'algorithme 3 et la Fig. 4.1 donnent une description du mécanisme d'ajustement de **SF**. SF_shifting doit avoir une connaissance globale du système d'allocation des **SF** pour effectuer les ajustements. En effet, il prend en entrée un **SF** pré-affecté à chaque terminal, la SF_Distribution (un vecteur du nombre de terminaux utilisant $SF = i$ pour $i \in \{7, 8, 9, 10, 11, 12\}$) et les charges (un vecteur des nombres seuils de terminaux pour chaque $SF = i$) calculées à l'aide de l'équation 4.2.

Algorithm 3 SF_Shifting

Input : SF, SF_Distribution, load

Output : SF

```

1: for s in range (SF,12) do
2:   if SF_Distribution[s] < load[s] then
3:     return SF
4:   else
5:     if SF_Distribution[s] ≥ load[SF] and SF_Distribution[s+1] < load[s+1] then
6:       if s < 12 then
7:         return s+1
8:       end if
9:     end if
10:  end if
11: end for

```

Comme le montre la figure 4.1, l'algorithme SF_shifting effectue une simple vérification des charges induites par l'utilisation de chacun des SF pour décider le SF à attribuer définitivement aux terminaux. En effet, après qu'un terminal a reçu une affectation, SF_shifting vérifie :

1. Si la classe de $SF = j$ courante (allouée lors de la phase d'allocation de SF) n'est pas surchargée, alors SF_shifting continue à utiliser $SF = j$.
2. Si la classe de $SF = j$ courante est surchargée mais que toutes les classes de $SF = i, i > j$ le sont également, alors SF_shifting continue à utiliser $SF = j$.
3. Si la classe de $SF = j$ courante (allouée lors de la phase d'allocation de SF) est surchargée et que l'une des classes supérieures $SF = i, i > j$ ne l'est pas, alors SF_shifting fait passer le $SF = i$ à la première classe de SF non surchargée.

Sans perte de généralité, nous illustrons dans la figure 4.2 le fonctionnement du mécanisme d'ajustement de SF en donnant un exemple.

Considérons un système simple utilisant seulement quatre valeurs de SF (de 7 à 10) où N terminaux envoient des trames à une seule passerelle. Nous supposons que la phase d'allocation de SF alloue $SF = 7$ au terminal k . SF_Shifting vérifiera si cette allocation est pertinente ou s'il est préférable de la mettre à jour. Le graphe dans la Fig. 4.2 illustre une telle configuration. Sous ces hypothèses, un terminal dont la valeur de SF est de 7 fait face à deux situations possibles :

1. Il conserve l'utilisation du $SF = 7$ si la classe des terminaux de ce SF n'est pas surchargée ou si elle l'est, mais que toutes les autres classes le sont également.

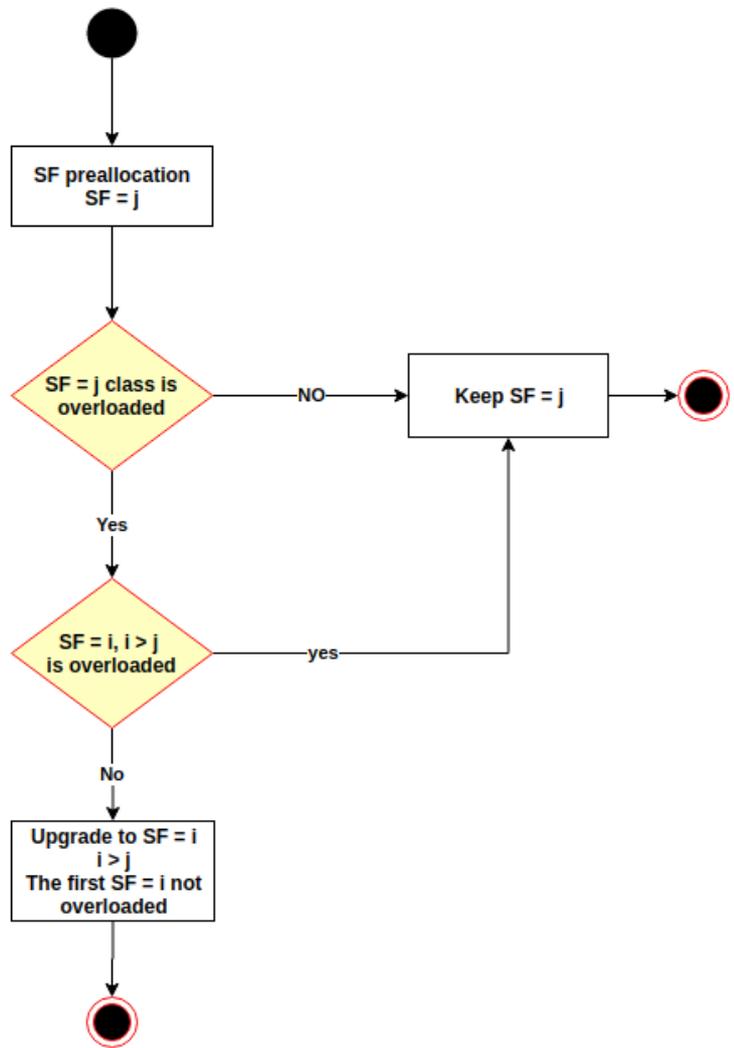


FIGURE 4.1 Mécanisme de décalage de SF [66]

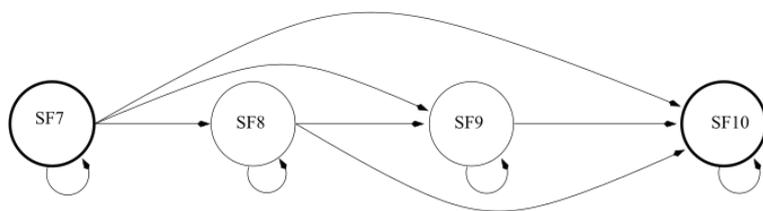


FIGURE 4.2 Stratégie d'équilibrage de charge pour l'allocation de facteurs d'étalement (seuls les SF7 à SF10 sont considérés sur ce schéma) [66]

2. Il est déplacé vers $SF = j$ avec $j > 7$ si la classe des terminaux utilisant $SF = 7$ est surchargée et que la classe des terminaux utilisant $SF = j$ ne l'est pas ($j \in \{8, 9, 10\}$ étant le premier SF non surchargé).

L'utilisation d'une telle stratégie garantit que chaque terminal se verra attribuer un SF approprié, ce qui est qualifié de bon pour deux raisons principales : 1) l'allocation de SF dans la phase deux assure qu'une trame sera reçue et pourra être décodée si aucune collision ne se produit, 2) l'ajustement des SF dans la phase trois contrôle la charge des SF, ce qui réduit considérablement le nombre de collisions. Comme on pouvait s'y attendre, le fait de faire passer, lorsque c'est possible, certains terminaux à des classes de SF plus élevées réduit le phénomène d'effet de capture. Par ailleurs, le fait d'avoir des classes de SF faible surchargées est moins pénalisant en termes de DER que des classes surchargées de SF élevé. Ceci est dû au temps passé en l'air par les trames qui augmente avec le SF et par conséquent augmente la période de vulnérabilité et donc potentiellement la probabilité de collision.

4.2 Analyse de performances

Dans cette section, nous analysons les performances de la stratégie d'allocation des facteurs d'étalement à travers des simulations approfondies utilisant l'implantation des algorithmes détaillés dans la section 4.1.

Nous avons développé un outil de simulation appelé LoRa-L3SFA [59] pour démontrer l'efficacité de la stratégie proposée. Le simulateur LoRa-L3SFA repose à la fois sur Simpy, un framework de simulation à événements discrets écrit en Python, et un modèle de perte fondé sur le modèle 3GPP Urban Macro. Ce simulateur est également fondé sur LoRaSim [106] et LoRaFREE [27]. En fait, pour accentuer la viabilité de la stratégie d'allocation SF sur les réseaux LoRaWAN, nous avons amélioré LoRaSim avec de nouvelles fonctionnalités données comme suit :

- il prend en compte l'orthogonalité imparfaite des facteurs d'étalement et l'impact du *fading* sur la base des seuils d'interférence indiqués dans le tableau 4.4 pour la vérification des collisions de trames ;
- il intègre un modèle de propagation 3GPP. À notre connaissance, il n'existe aucun autre travail qui utilise ce modèle dans les études de performance LoRa ;
- il prend également en compte des terminaux émules pour lesquels des valeurs de RSSI et de SNR réalistes, provenant d'un jeu de données expérimentales, sont utilisées.

4.2.1 Modèle de propagation

Nous considérons qu'une transmission dans les réseaux **LORAWAN** est reçue avec succès par la passerelle, si la puissance du signal reçu P_{rx} est supérieure à la sensibilité du récepteur. P_{rx} est donnée par l'équation 4.5.

$$P_{rx} = P_{tx} - PL + GL \quad (4.5)$$

où P_{tx} est la puissance de transmission, PL est l'affaiblissement de propagation et GL cumule tous les gains et pertes généraux.

Plusieurs modèles existent pour décrire l'affaiblissement de propagation PL dans différents environnements (urbain, suburbain, rural, espace libre) tels que : log-distance [112], Okumura-Hata [121] et les modèles **3GPP** [8].

Dans cette thèse, nous utilisons le modèle de propagation **3GPP Urban Macro** [8] qui est couramment utilisé pour modéliser les déploiements dans les zones de macro-cellules urbaines. Nous avons choisi ce modèle, car il correspond aux environnements dans lesquels les réseaux LoRa sont censés être déployés. L'utilisation de ce modèle de propagation dépend de la distance de communication d et peut être décrit comme suit :

$$\begin{aligned} PL = & 44.9 - 6.55 \times \log_{10}(GH) \times \log_{10}\left(\frac{d}{1000}\right) \\ & + 45.5 \\ & + 35.46 \times 1.1 \times NH \times \log_{10}(f_c) \\ & - 13.82 \times \log_{10}(GH) \\ & + 0.7 \times NH \\ & + \kappa \end{aligned} \quad (4.6)$$

où GH est la hauteur de l'antenne de la passerelle en mètres, NH la hauteur de l'antenne du terminal en mètres, f_c correspond à la fréquence de la porteuse en MHz , d est la distance entre la passerelle et le terminal en mètres, et κ est un facteur constant ($\kappa = 0$ dB pour des cellules macro-suburbaines et $\kappa = 3$ dB pour des cellules macro-urbaines selon le vocabulaire **3GPP**).

4.2.2 Validation du modèle de propagation par expérimentation

Afin de mieux comprendre comment le **RSSI** se comporte en fonction de la distance entre les terminaux et la passerelle **LORA**, nous avons mis en œuvre une étude expérimentale. L'objectif est également de valider et de justifier le modèle de propagation

retenu. En termes de hardware, nous avons utilisé pour les expériences une passerelle MultiTech Conduit et un terminal microchip RN2483 fixé sur une Raspberry Pi3. En outre, nous avons équipé cette plateforme d'un module GPS Xbee (voir Fig. 4.3). Le terminal LoRaWAN est fixé à un vélo et transmet les coordonnées GPS toutes les 20 secondes avec les paramètres radio suivants : $SF = 12$, $BW = 125kHz$, $Ptx = 14dBm$ et $CR = 4/5$.

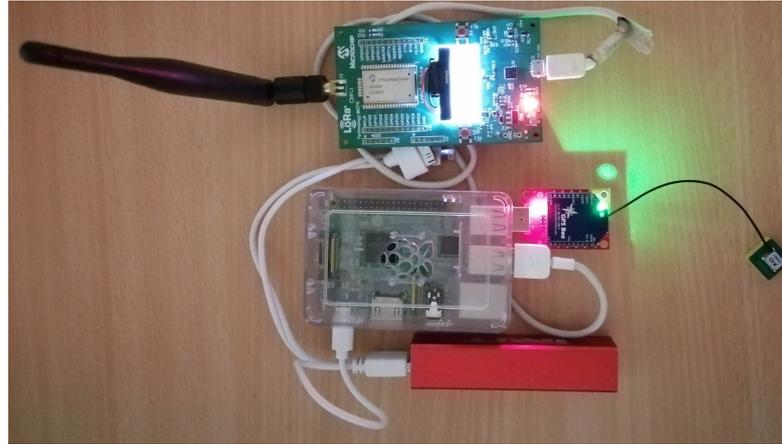


FIGURE 4.3 Plateforme d'expérimentation.

La Fig. 4.5 montre le circuit d'expérimentation tracé à l'aide des coordonnées GPS envoyées par le terminal microchip RN2483. Le but de cette expérimentation est de comparer les modèles de propagation avec des mesures provenant d'un environnement réel et utilisant un matériel réel. Les résultats de cette étude comparative sont illustrés dans la Fig. 4.4.

Comme tous les modèles de propagation dépendent fortement de la distance d entre le terminal et la passerelle, nous avons tracé l'évolution du RSSI en fonction de la distance pour le modèle log-distance, les modèles 3GPP et les valeurs de RSSI issues de l'expérimentation (les distances sont calculées à l'aide des coordonnées GPS). Nous pouvons constater que toutes les courbes ont la même apparence et présentent une tendance décroissante du RSSI lorsque l'on s'éloigne de la passerelle.

$$PL = PL(d_0) + 10\gamma \log_{10}\left(\frac{d}{d_0}\right) + \chi \quad (4.7)$$

Le modèle log-distance (équation 4.7) est très pénalisant et donne des valeurs de RSSI surestimées par rapport aux résultats empiriques obtenus par les expériences. Ceci est dû à une surestimation de la valeur de l'affaiblissement du chemin à une distance de référence d_0 , où $PL(d_0)$ à la distance $d_0 = 40$ mètres est supposé être $127.41dB$. γ

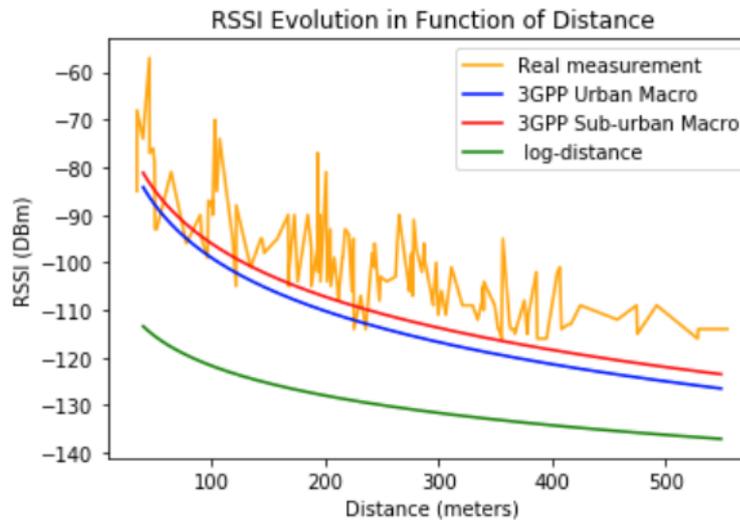


FIGURE 4.4 RSSI : Modèles mathématiques versus mesures réelles [66].

est l'exposant de l'affaiblissement de propagation avec une valeur de 2.08 et χ est une variable aléatoire distribuée selon une loi normale de moyenne nulle (en dB) avec un écart type σ . Cette variable n'est utilisée que lorsqu'il existe un effet de *shadowing*. Dans le cas contraire, cette variable est égale à zéro (nous supposons alors que $\chi = 0$).

Les modèles 3GPP tels que présentés dans la Fig. 4.4 semblent en revanche bien adaptés aux réseaux LORAWAN. Ils ne sont pas trop pénalisants et donnent des résultats qui reflètent ceux obtenus par expérimentation. En conclusion, nous avons retenu le modèle 3GPP pour un environnement macro-urbain pour nos simulations dans la mesure où il donne des valeurs de RSSI qui correspondent à des environnements réels où les réseaux LORAWAN sont les plus adoptés.

4.2.3 Scénarios de simulation

Dans cette partie, nous allons comparer les résultats obtenus en utilisant notre stratégie d'équilibrage de charge L3SFA pour l'allocation de SF et l'allocation SF fondée uniquement sur les seuils de RSSI et de SNR. Dans tous les scénarios, nous considérons une application qui envoie des trames de 20 octets selon un processus de Poisson. Nous supposons également que la période moyenne p d'envoi des trames est la même pour tous les terminaux.

Comme décrit dans la section 4.1, le mécanisme L3SFA comporte trois étapes successives. L3SFA initialise les terminaux avec un $SF = 7$ puis les trie en fonction de la distance d ou de leur RSSI et alloue enfin les SF tout en contrôlant la charge

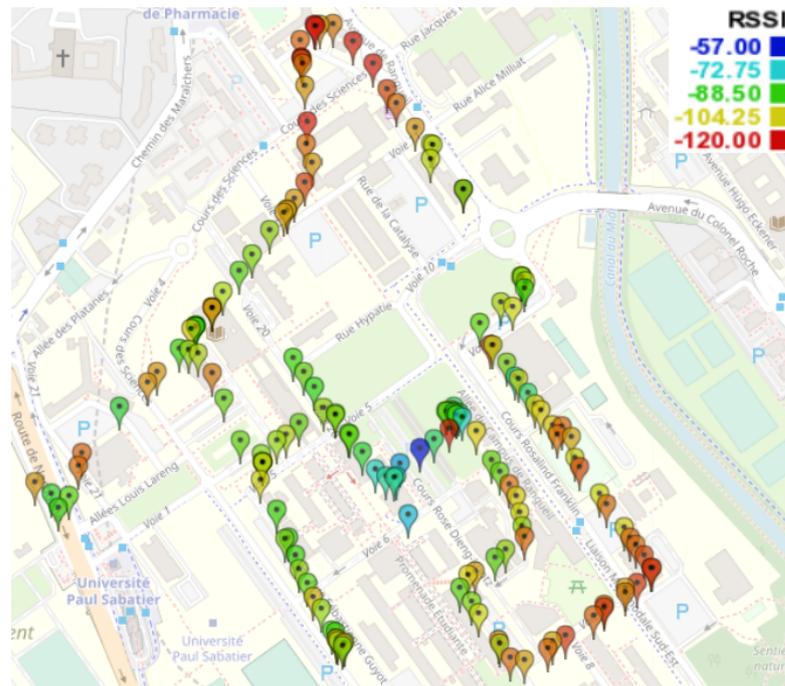


FIGURE 4.5 Coordonnées GPS transmises par un terminal RN2483 équipé d'un module GPS Xbee.

induite sur les différentes classes de SFs. Ces opérations conduisent à une affectation de SF. Nous simulons la transmission de trames par N terminaux utilisant les SF alloués précédemment pendant 2 heures. Nous examinons la performance de notre approche en termes de taux d'extraction de données (DER). En effet, dans un déploiement LoRa efficace, toutes les trames transmises doivent être reçues au niveau de la passerelle. Plus le DER est élevé, plus le déploiement est efficace.

Pour accroître la fiabilité, la pertinence et la qualité des résultats de la simulation, deux scénarios ont été envisagés :

Scénario 1 - Positionnement uniforme de terminaux

Dans cette approche, les terminaux sont positionnés selon un tirage uniforme dans un espace bidimensionnel formant une cellule de $rayon = 600$ mètres. Plusieurs simulations sont effectuées pour chaque scénario avec différentes valeurs de graines aléatoires afin d'assurer une plus grande précision et fiabilité des résultats.

Nous avons exécuté 2400 simulations pour N terminaux, $N \in [500, 10000]$ avec un pas de 500, sous différentes configurations. Tous les paramètres de simulation, les variables du modèle de propagation et de transmission, et les paramètres L3SFA sont résumés

dans le tableau 4.3.

Pour mieux comprendre le comportement de L3SFA, nous représentons dans la Fig. 4.6 la répartition du facteur d'étalement pour 5000 terminaux déployés de manière uniforme dans une cellule de 600 mètres de rayon. Nous pouvons voir sur la Fig. 4.6a qu'initialement tous les terminaux se voient attribuer un $SF = 7$. Ensuite, une pré-allocation des SF en fonction des seuils SNR et RSSI est effectuée. On observe, lors de cette première allocation de SF, que les SF8, SF9 sont attribués et les SF10, SF11, SF12 ne sont pas attribués. Enfin, la Fig. 4.6c représente l'ajustement des SF en considérant une charge $\rho = 0,5$, nous pouvons voir clairement que certains terminaux sont décalés vers des SF plus élevés pour alléger la charge du système et offrir de meilleurs taux de réception globaux.

TABLE 4.3 Configuration de la Simulation

Paramètre	Valeur
Terminaux	500-10000 éparpillés uniformément
BW	125 [kHz]
CR	4/5
TP (P_{tx})	14 [dBm]
SF	7, ..., 12
canaux	3 up-links [868.1, 868.3, 868.5]
Réceptions simultanées	8
Seuils SNR	Table 4.2
Sensibilité	Table 4.2
Interférences Inter-SF and Intra-SF	Table 4.4
p	100, 300, 600 (s)
Charge utile	20 [octet]
Modèle de propagation[15]	3GPP Urban Macro
Hauteur terminal (NH)	1 [m]
Hauteur passerelle (GH)	15 [m]

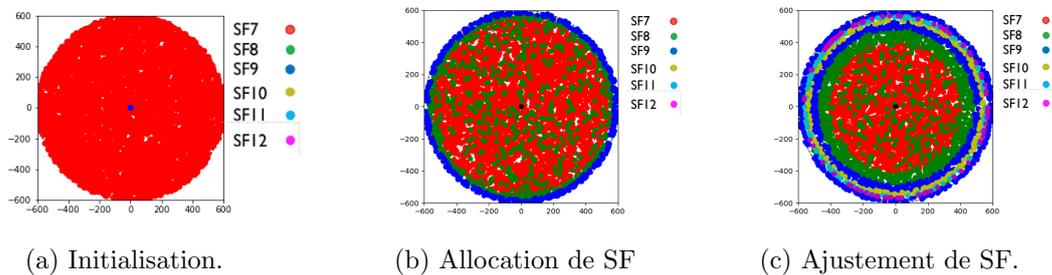


FIGURE 4.6 Répartition des SF pour 5000 terminaux dans une cellule de rayon de 600 mètres pour chaque étape de L3SFA.

TABLE 4.4 Seuils d'interférence [dbm].

SF	7	8	9	10	11	12
7	6	-8	-9	-9	-9	-9
8	-11	6	-11	-12	-13	-13
9	-15	-13	6	-13	-14	-15
10	-19	-18	-17	6	-17	-18
11	-22	-22	-21	-20	6	-20
12	-25	-25	-25	-24	-23	6

Pour ce scénario, nous avons tracé dans les Fig. 4.7d, à 4.7f la répartition des facteurs d'étalement correspondant à la phase de pré-allocation de L3SFA (sans contrôle de charge) et à la phase d'allocation finale après l'ajustement des SF pour un réseau LoRa de 5000 terminaux. De même, nous avons mis en évidence l'impact de la redistribution du SF sur l'obtention d'un meilleur DER global dans les figures 4.7a à 4.7c. Bien évidemment, la charge de trafic diminue en utilisant respectivement une fréquence d'émission de trames de $\frac{1}{100}$, $\frac{1}{300}$, et $\frac{1}{600}$.

Nous observons sur les Fig. 4.7a à 4.7c que L3SFA conduit à de meilleurs résultats en termes de DER que l'allocation initiale. En effet, le décalage de certains terminaux vers des SF plus élevés allège la charge du système et réduit les collisions dues à l'effet de capture, ce qui augmente la capacité du réseau LORA.

Par exemple, pour une fréquence d'émission de trames de $\frac{1}{600s}$ et pour atteindre un DER supérieur à 80%, L3SFA peut tolérer jusqu'à 8500 terminaux contre seulement 6000 en utilisant une allocation de SF fondée uniquement sur les seuils de RSSI et de SNR sans phase d'ajustement de SF (il n'y a pas de contrôle de charge des classes SF).

La répartition finale de SF est illustrée dans les Fig. 4.7g, 4.7h, 4.7i pour $\rho = 0.5$, $\rho = 0.3$, $\rho = 0.2$ respectivement où 5000 terminaux sont déployés dans une cellule d'un rayon de 600 mètres. Comparé à la Fig. 4.6b qui représente la répartition de SF pour 5000 terminaux avant le processus d'ajustement de SF, nous remarquons que certains terminaux ont changé de facteur d'étalement, ce qui entraîne la formation de nouveaux anneaux à l'intérieur de la cellule. Cela correspond au décalage d'un ensemble de terminaux vers des SF plus élevés.

Scénario 2 - Génération de terminaux à partir d'un jeu de données

Dans le scénario 2, nous avons utilisé un jeu de données réel de messages LORAWAN obtenus dans le centre-ville d'Anvers [11] pour positionner les terminaux. Ce jeu de données contient 130 430 messages qui ont été recueillis du 17 novembre 2017 au 19 juillet

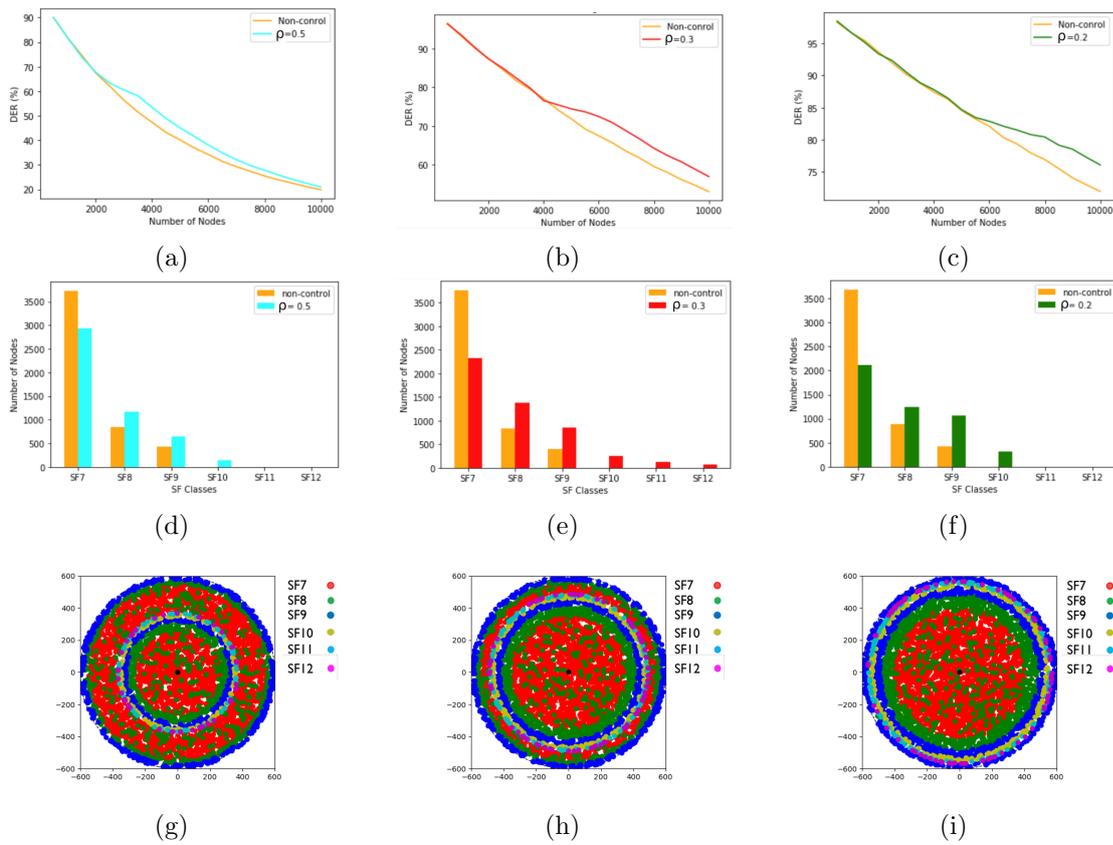


FIGURE 4.7 Performances de L3SFA avec positionnement uniforme des terminaux. [a-c] DER en fonction du nombre de terminaux pour une période moyenne d'émission de trames de, $\frac{1}{100s}$, $\frac{1}{300s}$, $\frac{1}{600s}$, respectivement. [d-f] Répartition des SF pour 5000 terminaux. [g-i] Répartition finale des SF pour 5000 terminaux dans une cellule d'un rayon de 600 mètres.

2019.

Ce jeu de données contient les charges utiles des terminaux ainsi que des méta-données décrivant l'état du canal (RSSI, SNR) et les paramètres de transmission radio (SF, BW, CR, etc.). Nous avons utilisé ces valeurs de RSSI et de SNR comme entrées des simulations pour émuler les terminaux avec des paramètres radio mesurés, comme dans un déploiement réel. Tous les autres paramètres de simulation sont les mêmes que dans le premier scénario et sont donnés dans la table 4.3.

Comme pour l'étude précédente avec une répartition uniforme des terminaux, nous évaluons la simulation menée avec des terminaux réels émulés pour démontrer l'efficacité de notre solution et améliorer la fiabilité de notre évaluation. Les résultats obtenus sont présentés dans la Fig. 4.8. La Fig. 4.8a montre clairement que le décalage de certains

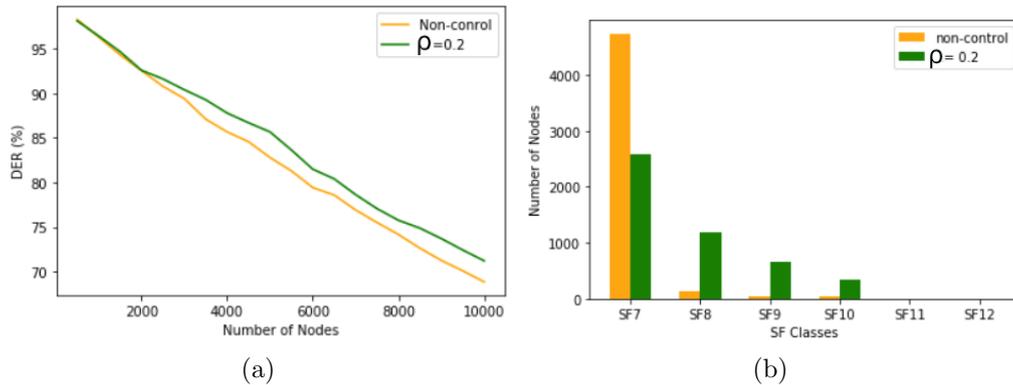


FIGURE 4.8 Analyse des performances de L3SFA pour un positionnement de terminaux issu d'un jeu de données.

terminaux vers des SF plus élevés (voir Fig. 4.8b) afin d'alléger les classes de SF souffrant d'une charge élevée conduit à un meilleur DER global. Par conséquent, l'utilisation de L3SFA augmente la capacité du réseau. En effet, d'après la Fig. 4.8, pour atteindre un DER supérieur à 85%, le réseau peut tolérer 6000 terminaux contre seulement 4500 sans ajustement de SF.

4.3 Pour aller plus loin : Impact du contrôle de la puissance de transmission

Dans les réseaux LoRaWAN, les transmissions simultanées utilisant différents facteurs d'étalement peuvent être traitées au niveau de la passerelle. Cependant, les différentes puissances des signaux reçus de ces transmissions simultanées peuvent augmenter considérablement le taux d'erreur. Cela est dû au fait que les facteurs d'étalement ne sont pas complètement orthogonaux, ce qui provoque des interférences mutuelles. La matrice SIR [36] donnée dans la table 4.4 représente les seuils du rapport signal sur interférence entre les facteurs d'étalement.

La réception de deux signaux qui se chevauchent en utilisant des facteurs d'étalement différents ou similaires sur le même canal ne signifie pas que les deux sont perdus. En pratique, si la différence entre les puissances de leurs signaux est inférieure au seuil de brouillage correspondant, les deux signaux, ou au moins le plus fort, peuvent être reçus avec succès. Par conséquent, un mécanisme d'isolation des signaux est nécessaire pour recevoir avec succès toutes les transmissions simultanées. Dans la transmission sans fil, cette isolation peut être réalisée soit par le contrôle de la puissance de transmission, soit

par l'affectation des canaux.

4.3.1 Interférences Inter-SF et Intra-SF

Le phénomène d'interférences Inter-SF survient lorsque deux trames sont envoyées dans le même canal avec des SF différents et sont reçues simultanément au niveau de la passerelle. En effet, les facteurs d'étalement n'étant pas parfaitement orthogonaux, deux trames concurrentes, i et j , envoyées avec des facteurs d'étalement différents, sf et sf' respectivement, ne sont toutes deux correctement reçues que si les puissances des signaux reçus sont supérieures aux seuils définis dans la matrice SIR. Sinon, si l'équation 4.8 est vérifiée, seule la trame i est correctement reçue.

$$RSSI_{sf}^i - RSSI_{sf'}^j \leq SIR_{sf,sf'} \quad (4.8)$$

A l'opposé, le phénomène d'interférences Intra-SF survient lorsque deux trames sont envoyées dans le même canal en utilisant le même facteur d'étalement $sf = sf' = k$. Cela ne signifie pas nécessairement que les deux trames sont perdues à cause de l'effet de capture. En effet, si la différence absolue des puissances de ces deux signaux reçus est inférieure à $6dB$ (Equation 4.9), les deux trames sont perdues. Dans le cas contraire, la trame dont le signal est le plus fort pourra être décodée.

$$| RSSI_{sf}^i - RSSI_{sf'}^j | \leq 6 \quad (4.9)$$

4.3.2 Contrôle de la puissance de transmission

Dans cette section, nous présentons une stratégie de contrôle de puissance de transmission appelée *TPC_LoRa* qui étend l'algorithme L3SFA à L3SFA-TPC donné dans l'algorithme 4. Plus précisément, l'algorithme *TPC_LoRa* examine la différence de puissance entre toutes les paires de trames provenant de tous les terminaux envoyées sur le même canal. Comme défini dans les spécifications LORAWAN, la puissance d'émission varie de 2 à 14 dBm dans la réglementation européenne de la bande ISM avec un pas de 3 comme suit : $TP \in \{2, 5, 8, 11, 14\}$. La stratégie proposée calcule les étapes maximales possibles pour diminuer la puissance d'émission en fonction de la différence entre la puissance d'émission et les seuils définis dans la matrice SIR, comme indiqué par les lignes [5 – 14] de l'algorithme 5. De plus, le nombre de pas est utilisé pour diminuer la puissance d'émission du terminal dont la trame sera perdue.

Pour valider la solution TPC_LoRa combinée à L3SFA, nous avons simulé 5000 terminaux avec les mêmes paramètres de simulation que ceux indiqués dans le tableau

Algorithm 4 L3SFA-TPC

Input : N : Terminaux

Output : Terminaux avec SF alloué et TP ajustée, répartition de SF.

- 1: Initialisation des terminaux
- 2: Allocation de SF
- 3: Ajustement de SF
- 4: Contrôle de la puissance de transmission (Algorithme 5)

4.3 pour $p = 100$ secondes, et nous avons évalué la consommation d'énergie et le DER.

$$E = \sum_{n=1}^N T_n \times Tx_n \times Nframe_n \times V \quad (4.10)$$

L'équation 4.10 donne la consommation d'énergie cumulée E sur toutes les trames envoyées par N terminaux où T_n est le temps de vol pour le terminal n utilisant le SF alloué, Tx_n est la consommation de transmission en mA , $Nframe_n$ est le nombre de trames envoyées par le terminal n et enfin V est la tension.

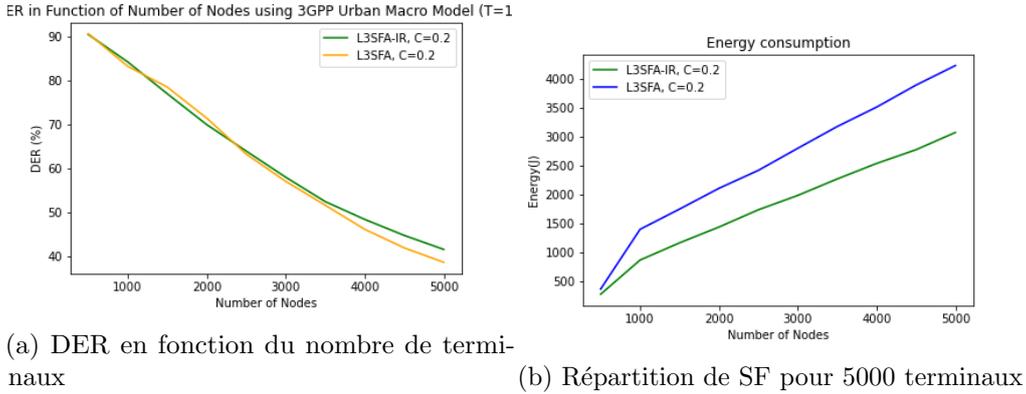


FIGURE 4.9 Analyse de performance de L3SFA-TPC.

Nous avons représenté sur la Fig. 4.9b la consommation d'énergie cumulée E en *Joule* sur toutes les trames envoyées par 5000 dispositifs finaux pour les deux solutions L3SFA et L3SFA-TPC avec une charge de trafic fixée à $\rho = 0,2$. Il est intéressant de noter que nos résultats démontrent que l'adoption de *TPC_LoRa* diminue la consommation d'énergie de 27% par rapport à L3SFA. De plus, L3SFA-TPC présente de meilleures performances en termes de DER pour $\rho = 0.2$ et conserve les mêmes performances que L3SFA tout en améliorant la consommation énergétique.

4.4 Conclusions et discussions

Dans cette première contribution, nous avons abordé la question de l'affectation du facteur d'étalement dans les réseaux LoRaWAN à grande échelle. Nous avons montré que le problème est similaire à un problème d'équilibrage de charge, et par conséquent, nous avons présenté une nouvelle approche pour l'allocation des SF qui vise à trouver la meilleure répartition des terminaux parmi les différentes classes de SF.

Les résultats de l'évaluation des performances de L3SFA sous différents scénarios et en utilisant différentes approches d'évaluation suggèrent que l'équilibrage de la charge conduit à de meilleures performances en termes de taux de délivrance tout en garantissant le passage à l'échelle pour des réseaux de grande taille (denses).

De plus, l'extension de L3SFA avec un algorithme de contrôle de la puissance de transmission démontre que l'approche L3SFA-TPC permet d'économiser 27% de consommation d'énergie tout en maintenant la même performance en termes de taux de délivrance et de densité du réseau.

En outre, nous envisageons de déployer nos algorithmes sur notre serveur de réseau LoRaWAN et de mener des expériences pour étudier comment ils se comportent dans une infrastructure réelle, en particulier sur les communications sur la liaison descendante. Cela a constitué notre motivation pour fournir un framework, appelé LoRADL, dédié à la prise en charge des communications sur la liaison descendante. Les détails sur cette contribution sont exposés dans la partie III de ce manuscrit.

Finalement, l'utilisation de nos algorithmes d'allocation de paramètres radio dans un système avec de multiples passerelles est également une piste de recherche prometteuse.

Algorithm 5 TPC_LoRa

Input : N : end-devices.**Output :** N with Tx adjusted.

```

1: for  $i$  in  $N$  do
2:   for  $j$  in  $N$  do
3:     if  $i \neq j$  then
4:       if  $i.freq == j.freq$  then
5:         if  $(i.rssi - j.rssi) < SIR[i.sf - 7][j.sf - 7]$  then
6:           if  $i.sf == j.sf$  then
7:              $diff = SIR[i.sf][j.sf] - (i.rssi - j.rssi)$ 
8:           else
9:              $diff = SIR[i.sf][j.sf] - (i.rssi - j.rssi)$ 
10:          end if
11:           $steps = \lceil diff/3 \rceil$ 
12:        end if
13:        if  $steps > 4$  then
14:           $steps = 4$ 
15:        end if
16:         $newRSSI = 0$ 
17:         $newTx = 0$ 
18:        for  $s$  in range  $(1, steps)$  do
19:          if  $j.rssi - s \times 3 > Sensitivity_{j.sf}$  &  $j.tx - s \times 3 \geq 5$  then
20:             $newRSSI = j.rssi - s \times 3$ 
21:             $newTx = j.tx - s \times 3$ 
22:          end if
23:          if  $newRSSI \neq 0$  and  $newTx \neq 0$  then
24:             $j.rssi = newRSSI$ 
25:             $j.tx = newTx$ 
26:          end if
27:        end for
28:      end if
29:    end if
30:  end for
31: end for

```

Deuxième partie

Itinérance et Mobilité

Pour un ensemble d'applications telles que le suivi, la localisation et la surveillance de la logistique intelligente, les terminaux sont susceptibles de quitter les cellules de leur opérateur domestique. Malheureusement, dans ces scénarios de mobilité, les opérateurs ne peuvent pas systématiquement fournir une couverture plus large. En effet, procurer une couverture mondiale par un seul opérateur n'est pas économiquement viable ni techniquement pratique. Cependant, **LoRaWAN** est une technologie sans fil *open source* qui permet aux opérateurs et aux particuliers de créer des réseaux privés ou publics. Par conséquent, l'interconnexion de tous ces réseaux via des accords d'itinérance permet d'atteindre une forme de couverture mondiale. L'unification de ces réseaux par l'activation de l'itinérance **LoRaWAN** nécessite la capacité (i) de découvrir le réseau d'origine de l'appareil final mobile, (ii) de vérifier si l'opérateur propose un service d'itinérance pour le terminal, et (iii) de récupérer le contexte des clés de session du terminal.

Pour mieux mettre en évidence les principales contributions et le fil conducteur de cette contribution, nous décrivons les mécanismes conçus, en présentant les outils et en mettant en évidence l'approche d'évaluation adoptée afin de valider la conception de notre solution d'itinérance. Les principales contributions sont au nombre de trois :

- Conception de la solution d'itinérance **LoRaWAN** : nous commençons par la description de la conception de notre solution architecturale de l'interconnexion entre opérateurs. Nous avons ensuite introduit un abonnement au réseau et un mécanisme d'approvisionnement des terminaux avec des JoinEUI. Nous terminons par le protocole d'itinérance.
- Expérimentation : La validation de notre proposition passait nécessairement par une phase expérimentale en vraie grandeur.
- Émulation : Le banc de test permet de mettre en exergue la faisabilité de la solution. L'analyse des performances et le passage à l'échelle a donné lieu à une deuxième étape de validation. Pour ce faire, nous avons mis en place un émulateur dédié à l'itinérance active.

Organisation de la partie

Cette partie comprend deux chapitres. Dans le premier chapitre, nous présentons un état de l'art des solutions d'itinérance existantes et des procédures d'activation des terminaux dans un réseau LoRaWAN. Nous présentons également la structure d'une trame LoRaWAN nécessaire à la compréhension des mécanismes proposés.

Dans le deuxième chapitre, nous présentons [LORAROAM](#), notre solution d'itinérance active pour l'interconnexion d'opérateurs, la mise en place du banc de test et la conception d'un émulateur dédié à l'itinérance [LORAWAN](#).

ETAT DE L'ART

5.1 Format de la trame LoRaWAN

Avant d'aborder les mécanismes d'activation des terminaux, il est important de bien comprendre la structure des trames **LoRaWAN**. Dans la suite de cette section, on se focalisera principalement sur les charges utiles composant les couches physiques et **MAC** respectivement. Selon les spécifications **LoRaWAN** [7], [5], [6] une trame est structurée

Radio PHY layer:

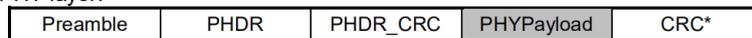


Figure 5: Radio PHY structure (CRC* is only available on uplink messages)

PHYPayload:

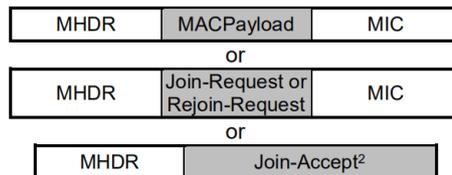


Figure 6: PHY payload structure

MACPayload:

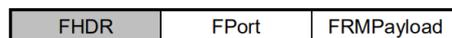


Figure 7: MAC payload structure

FHDR:

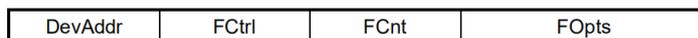


Figure 8: Frame header structure

FIGURE 5.1 Structure d'une trame LoRaWAN [5].

comme illustrée dans la figure 5.1. Le détail des champs la composant est donné comme suit :

- **Préambule** : le préambule est une suite de 8 symboles pour la détection du début des trames.
- **PHDR** : LoRa Physical HeaDeR.
- **PHDR_CRC** : header CRC, où CRC est un code de redondance cyclique pour la détection d'erreurs sur l'en-tête.
- **PHYPayload** : représente la charge utile physique.

5.1.1 Charge utile physique

La charge utile physique est encapsulée dans le champ **PHYPayload** de la trame et elle se compose des champs : **MHDR**, **MIC** et **MACPayload** ou **Join-Request** ou **Join-Accept**. Dans cette partie, nous nous intéressons particulièrement aux messages de type Join-Request. Le détail des différents champs est donné comme suit :

- **MHDR** : est l'en-tête **MAC**. Il spécifie le type de message (join request, liaison montante, liaison descendante, etc) et la version du format de la trame **LoRAWAN** ;
- **MIC** : *Message Integrity Code* de 4 octets. Il est calculé à partir des champs **MHDR** et **MACPayload** ;
- **MACPayload** : représente la charge utile **MAC** ;
- **Join-Request** : est un message envoyé par un terminal pour rejoindre (activation) un réseau **LoRAWAN** et il comprend les champs suivants :
 - **JoinEUI** : appelé aussi AppEUI, est un identifiant d'application unique de 64 bits dans l'espace d'adressage IEEE EUI64 qui identifie de manière unique l'entité capable de traiter la trame Join-Request ;
 - **DevEUI** : un identifiant unique de 64 bits dans l'espace d'adressage IEEE EUI64 qui identifie de manière unique le terminal ;
 - **DevNonce** : une valeur unique, aléatoire, de 2 octets, tirée par le terminal. Le serveur de réseau utilise le DevNonce de chaque terminal pour garder la trace de leurs demandes de connexion pour détecter les attaques par rejeu (sécurité) ;
- **Join-Accept** : est un message envoyé par un serveur de réseau comme réponse au terminal demandant son activation. Le détail des champs comportant ce message ne sera pas discuté dans cette partie.

5.1.2 Charge utile MAC

La charge utile **MAC** des trames de données contient un en-tête de trame **FHDR** suivi d'un champ optionnel de port **FPort** et d'un champ optionnel de charge utile de trame **FRMPayload** qui contient les données applicatives. L'en-tête de la trame **FHDR** est composé des champs suivants :

- **DevAddr** : une adresse du terminal engendrée par le **NS** après la procédure d'activation.
- **FCtrl** : un octet de contrôle de trame utilisé pour l'activation de l'**ADR**, l'activation des accusés de réception, la définition de la taille du champ **FOpts**.
- **FCnt** : un compteur de trames.
- **FOpts** : jusqu'à 15 octets d'options de trame utilisées pour transporter les commandes **MAC**.

5.2 Activation des terminaux

5.2.1 Activation OTAA dans LoRaWAN v1.0.x

Dans la version LoRaWAN v1.0.x [5], [6], la procédure d'adhésion nécessite l'échange de deux messages **MAC** entre le terminal et le serveur de réseau : un **Join-request** du terminal au serveur de réseau et un **Join-accept** du serveur de réseau au terminal. Avant l'activation, le **JoinEUI**, le **DevEUI** et l'**AppKey** doivent être approvisionnés dans le terminal. L'**AppKey** est une clé secrète AES-128 bits appelée aussi clé racine. La même **AppKey** doit être fournie sur le serveur réseau où l'appareil final va s'enregistrer. Le **JoinEUI** et le **DevEUI** ne sont pas secrets et sont visibles par tous. Les étapes d'activation **OTAA** des terminaux sont illustrées dans la figure 5.2 et données comme suit :

- **Étape 1** : la procédure d'activation est toujours lancée par le terminal. Il envoie le message **Join-request** au réseau à rejoindre. Le message **Join-request** est composé des champs suivants : **AppEUI**, **DevEUI** et **DevNonce** et il est toujours envoyé non chiffré.
- **Étape 2** : le serveur de réseau traite le message **Join-request** et engendre deux clés de session (**NwkSKey** et **AppSKey**) et le message **Join-accept** si le terminal est autorisé à rejoindre ce réseau. Le message **Join-accept** est ensuite chiffré avec l'**AppKey**.
- **Étape 3** : le serveur de réseau renvoie le message chiffré **Join-accept** au terminal en liaison descendante.
- **Étape 4** : le serveur de réseau conserve la **NwkSKey** et distribue l'**AppSKey** au

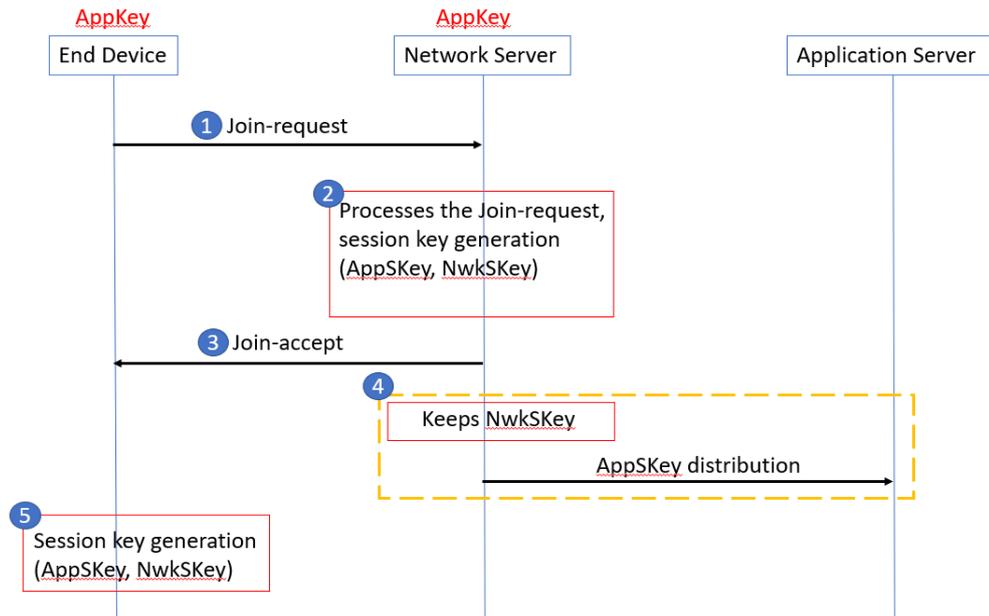


FIGURE 5.2 Échange de messages OTAA dans LoRaWAN v1.0.x

serveur d'application.

- **Étape 5 :** le terminal déchiffre le message Join-accept et utilise l'AppKey pour déduire les deux clés de session, la clé de session du réseau **NwkSKey** et la clé de session de l'application **AppSKey**.

L'utilisation d'une seule clé racine **AppKey** pour déduire les clés de session, à savoir la **NwkSKey** et la **AppSKey**, ne permet pas de prendre en charge l'itinérance active (*Handover Roaming*), seule l'itinérance passive est possible. La version LoRaWAN v1.1 propose de nouveaux mécanismes et améliorations permettant par exemple de prendre en charge l'itinérance active.

5.2.2 Activation OTAA dans LoRaWAN v1.1

À l'instar de la version v1.0.x, dans LoRaWAN v1.1 [7], la procédure d'activation des terminaux nécessite l'échange des mêmes messages **MAC** entre le terminal et le serveur de réseau. Cependant, les modifications apportées sont illustrées dans la figure 5.3 et données comme suit :

1. un nouveau composant, appelé **Join Server** (JS), est ajouté à l'architecture **LoRaWAN**. Ce composant est dédié à la déduction des clés de session. Cette opération était auparavant effectuée, dans LoRaWAN v1.0.x, par le serveur de

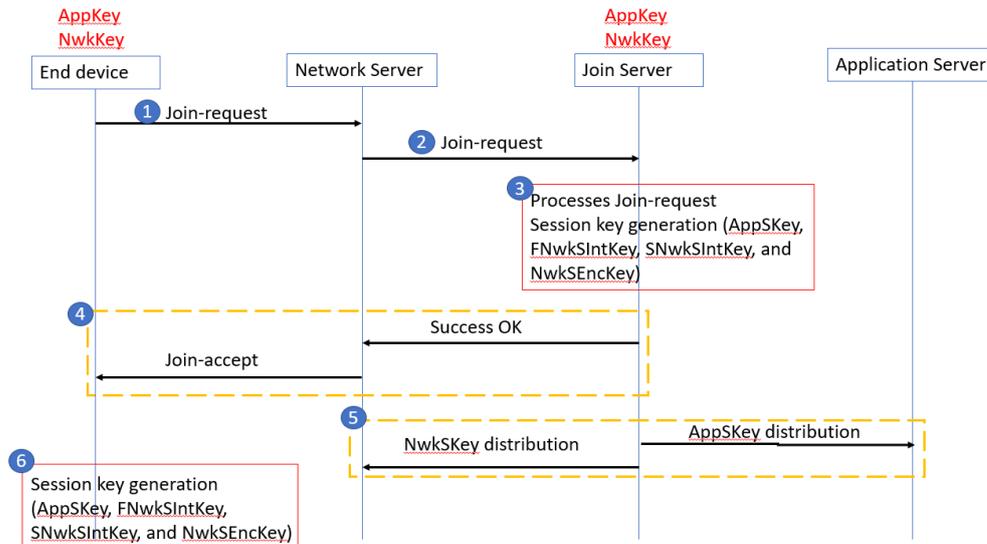


FIGURE 5.3 Échange de messages OTAA dans LoRaWAN v1.1

réseau ;

2. en plus de la clé **AppKey**, une nouvelle clé racine est introduite, appelée **NwkKey**. Par conséquent, les changements suivants sont à prévoir. Avant l'activation, les clés **JoinEUI**, **DevEUI**, **AppKey** et **NwkKey** doivent être stockées dans le terminal. L'**AppKey** et le **NwkKey** sont des clés secrètes AES-128 bits connues sous le nom de clés racine. L'**AppKey**, le **NwkKey** et le **DevEUI** correspondants doivent être fournis au **JS** ;
3. le **JS** traite le message **Join-request** et déduit toutes les clés de session **AppSKey**, **FNwkSIntKey**, **SNwkSIntKey** et **NwkSEncKey** si le terminal est autorisé à rejoindre le réseau ;
4. le **JS** envoie l'**AppSKey** au serveur d'application et les trois clés de session réseau (**FNwkSIntKey**, **SNwkSIntKey** et **NwkSEncKey**) au serveur de réseau ;
5. le terminal déchiffre le message **Join-accept** en utilisant les deux clés **AppKey**, **NwkKey** et déduit les clés de session.

Les modifications apportées dans la version v1.1 comparée à la version v1.0.x pour l'activation **OTAA** sont résumées dans la table 5.1.

Note importante :

Comme susmentionné, le **JS** est une nouvelle entité introduite dans la version v1.1 des réseaux **LORAWAN** pour la déduction des clés de session dans le but de décharger le

TABLE 5.1 Activation OTAA dans LoRaWAN v1.0.x versus v1.1

	LoRaWAN v1.0.x	LoRaWAN v1.1
Clés racines	AppKey	AppKey, NwkKey
Clés déduites	NwkSKey, AppSKey	FNwkSIntKey, SNwkSIntKey, NwkSEncKey et AppSKey
Join Server	Inclus dans le NS	Oui, entité séparée
Type d'itinérance	Itinérance passive	Itinérance passive Itinérance active (<i>Handover</i>)

NS de tout calcul supplémentaire. Ce calcul était auparavant effectué au niveau du NS dans la version v1.0.x. Dans le reste de ce manuscrit et pour faciliter la schématisation et l'explication de nos solutions, **nous considérons que le JS est inclus dans le NS**. En effet, nous ne nous intéressons qu'aux clés de session et non à leur calcul.

5.3 Travaux pertinents sur l'itinérance LoRaWAN

Récemment, l'industrie et les organismes de recherche académiques se sont intéressés à l'étude de l'itinérance dans les réseaux **LoRaWAN**. En effet, l'itinérance est une préoccupation fondamentale qui a attiré de plus en plus l'attention afin de répondre aux besoins des applications **IoT** en termes de prise en charge de la mobilité. Parmi les contributions récentes de la recherche, nous examinons cinq directions différentes proposées pour rendre possible la mobilité inter-opérateurs des terminaux dans le contexte des réseaux **LoRaWAN**.

5.3.1 Itinérance à base de 5G

Dans [126], les auteurs ont proposé un mécanisme permettant l'itinérance entre les réseaux **LoRaWAN** et la 5G. Ils ont introduit un mécanisme de *Handover Roaming* qui s'appuie sur la 5G pour authentifier les terminaux dans le réseau. À notre connaissance, les tests n'ont été effectués que pour des scénarios d'itinérance passive ; les scénarios d'itinérance active sont toujours en cours d'exploration.

De plus, la solution proposée, telle qu'elle a été conçue par leurs auteurs, suppose que les terminaux sont destinés à être équipés d'une interface 5G et d'une interface LoRa. Cela augmente donc considérablement la consommation d'énergie et aura un impact considérable sur les coûts de déploiement, ce qui va à l'encontre de l'esprit d'efficacité énergétique de **LoRaWAN** et des déploiements à bas coût.

5.3.2 Itinérance à base de blockchain

Dans [45] et [46], les auteurs ont proposé une infrastructure LPWAN décentralisée pour l'itinérance passive LORAWAN fondée sur une solution *blockchain*. Dans ce travail, ils ont proposé un mécanisme d'approvisionnement du terminal par un JoinEUI pour effectuer une recherche de son réseau domestique fondée sur un *Smart Contract Solidity*. Ils ont étendu leur travail dans [46] avec une couche de sécurité supplémentaire utilisant des signatures numériques afin de vérifier l'intégrité des messages comme une solution concurrente à la vérification MIC pour une approche décentralisée. Malgré l'originalité de cette solution, elle est limitée à LoRaWAN v1.0.x [5], [6] et ne prend en charge que le *roaming* passif. De plus, aucune évaluation des performances n'est donnée sur l'impact de l'adoption d'une solution décentralisée sur les exigences des terminaux de la classe A en termes de délais engendrés.

5.3.3 Itinérance à base de systèmes répartis

Les auteurs présentent dans [85] une solution fondée sur des serveurs distribués pour gérer l'itinérance passive dans le contexte des réseaux LoRaWAN v1.0.x [5], [6]. Ils ont étendu l'architecture LORAWAN avec de nouvelles entités appelées *Distribution Servers* (DS) qui fonctionnent comme des *brokers*. Comme cette solution repose sur une architecture de système distribué, elle offre les mêmes avantages de décentralisation que ceux présentés dans la direction précédente.

5.3.4 Itinérance à base de SCHC

Dans les deux travaux [20], [19], les auteurs ont proposé le protocole *Mobile Static Context Header Compression* (MSCHC) fondé sur le protocole *Static Context Header Compression* (SCHC) [18] pour gérer la mobilité des terminaux. Ils ont proposé une solution permettant d'assurer la continuité des sessions pendant la mobilité des terminaux grâce à un mécanisme basé sur des règles. Plus précisément, ce mécanisme repose sur un contexte composé de plusieurs règles stockées sur le terminal et sur le réseau. Chaque règle de contexte contient principalement une liste de champs d'en-tête IPv6 avec une valeur cible, un opérateur de correspondance et une action de compression/décompression. Un examen attentif de la proposition révèle que le stockage des règles sur les terminaux et l'exécution des actions de compression/décompression peut être très coûteuse en termes de consommation d'énergie.

5.3.5 Itinérance à base de DNS

Dans [21], les auteurs ont proposé IoTRoam, une architecture d'itinérance passive **LORAWAN**. IoTRoam est principalement fondée sur un modèle d'itinérance opérationnelle pour interconnecter différents réseaux IoT. En outre, une preuve de concept est fournie, avec une mise en œuvre de la solution d'itinérance passive proposée. Cependant, cette évaluation montre que la solution est opérationnelle, mais n'a pas été complètement menée en condition de forte charge, pour démontrer l'évolutivité de la proposition.

Malgré la pertinence des travaux susmentionnés, il convient de noter qu'ils se concentrent uniquement sur l'itinérance passive [21], [45], [46] et [85] ou qu'ils proposent des solutions inefficaces sur le plan énergétique [20], [19], [18] et [126]. Ainsi, à la différence de ces directions, nous étudions la capacité d'itinérance *handover* interopérateur afin de proposer une nouvelle solution sans compromettre la standardisation de la technologie **LORAWAN** rapportée dans la spécification [5], [6] et [7] et en soulageant les terminaux de tout traitement supplémentaire entraînant une consommation d'énergie supplémentaire. En outre, nous avons réalisé une étude expérimentale exhaustive pour montrer comment le délai de migration du contexte répond aux exigences de la classe A sous différentes charges de trafic.

5.4 Notre positionnement

Dans cette deuxième partie du manuscrit, nous présentons **LORAROAM**, une solution dédiée à l'itinérance des terminaux dans les réseaux **LORAROAM**. Nos solutions visent la satisfaction des objectifs suivants :

- **Handover Roaming** : proposition d'une solution d'itinérance active fondée sur la résolution **DNS** et la migration de contexte des terminaux (migration de la couche **MAC**) avec un minimum de configuration, sans ajouter une charge supplémentaire au niveau des terminaux et en se fondant sur la standardisation LoRaWAN v1.1 ;
- **Implantation** : implantation de la solution proposée et intégration à la *stack* chirpstack. À notre connaissance, cette implantation est la première permettant la mise en place d'un banc de test qui prenne en charge l'itinérance active ;
- **Expérimentation** : mise en place d'un banc de test et analyse des délais engendrés par notre protocole d'itinérance dans différents scénarios ;
- **Émulation** : conception et implantation d'un émulateur dédié à l'itinérance **LORAWAN**. Cet outil, qui est un premier en ce genre, étend **LORAROAM** pour permettre d'effectuer des expérimentations à large échelle, et remédier au manque

de matériel et au coût de leur déploiement.

LoRaRoam : Itinérance dans les Réseaux LoRaWAN

6.1 Conception de la solution d'itinérance LoRaWAN

6.1.1 Architecture étendue d'itinérance LoRaWAN

Comme nous l'avons décrit dans la section 2.2.1 du chapitre 2, dans les scénarios classiques, les terminaux communiquent avec leur NS domestique (HNS) via les passerelles. Cependant, ces mêmes terminaux peuvent se trouver hors de la couverture de leur opérateur domestique et visiter des zones couvertes par d'autres opérateurs. Une solution naturelle consiste à exploiter ces infrastructures qui appartiennent à d'autres opérateurs par le biais d'accords d'itinérance. Les principes s'inspirent naturellement des solutions qui ont été déployées avec succès depuis trente ans dans le contexte des réseaux mobiles. Ainsi, afin d'établir une interconnexion entre les opérateurs participant au dispositif d'itinérance, nous avons étendu l'architecture LoRaWAN avec des composants logiciels locaux (HLA et VLA) et centraux (MA) comme le montre la figure 6.1.

Sur la base de la description ci-dessus, nous avons développé les composants suivants et illustré leurs interactions dans la Figure 6.2 :

Master Agent (MA)

Le MA est un composant central qui joue un rôle d'intermédiaire pour l'interconnexion de plusieurs réseaux LoRaWAN indépendants appartenant à différents opérateurs. En effet, la mise en œuvre d'un mécanisme d'itinérance devient essentielle pour permettre les communications entre plusieurs réseaux. Pour effectuer une telle opération, le MA

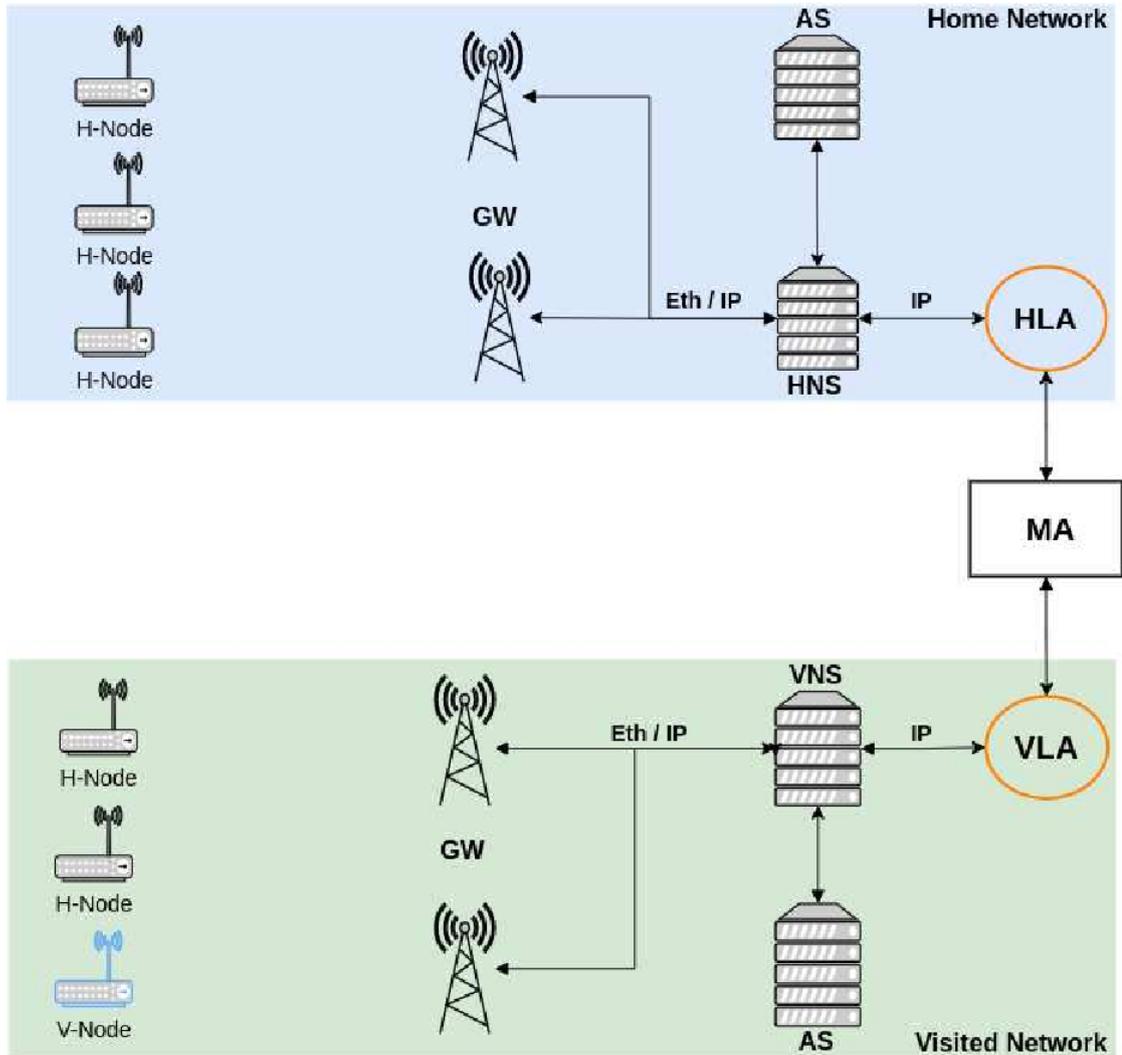


FIGURE 6.1 Architecture LoRaWAN étendue pour prendre en charge l'itinérance active.

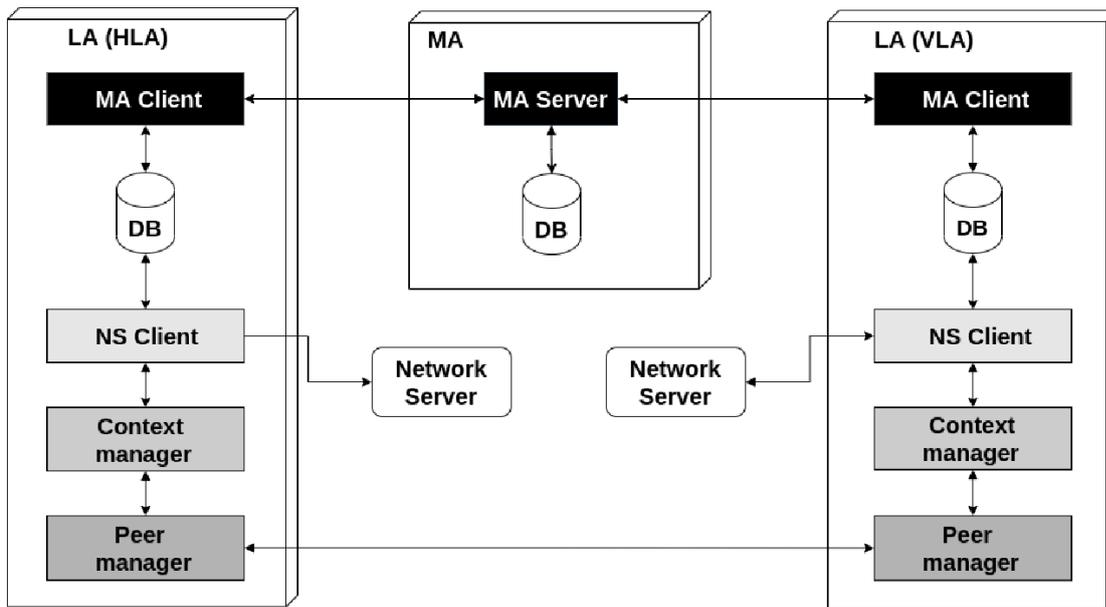


FIGURE 6.2 Architecture système de la solution d'itinérance active

commence par fournir une procédure d'abonnement au réseau afin d'attribuer un **NET-ID** unique de 32 bits à chaque réseau participant à l'itinérance. Ce **NET-ID** est utilisé après avoir été extrait des trames reçues pour effectuer une résolution **DNS** de la recherche du réseau domestique du terminal. Il est à noter que cette résolution **DNS** est différente de la résolution **DNS** classique, car la résolution proposée permet de trouver le réseau domestique auquel appartient le terminal à partir d'un **JoinEUI unique**.

Local Agent (LA)

Le LA est un composant local connecté au **NS**. Il se comporte de deux manières différentes, en tant qu'agent local visité (**VLA**) lorsqu'il gère les trames provenant de terminaux itinérants reçues sur le serveur de réseau visité (**vNS**) ou en tant qu'agent local domestique (**VLA**) pour gérer l'échange de trames provenant d'un réseau partenaire dans le cadre d'accords d'itinérance. Un **VLA** (ou **HLA**) est constitué de quatre sous-composants, comme le montre la figure 2.2 :

- le **MA** Client représente une interface de connexion avec le **MA**, son rôle consiste à se procurer les services fournis par le **MA** pour la souscription d'un nouveau partenaire itinérant ou pour une résolution **DNS** afin de rechercher le réseau domestique du terminal ;
- le client **NS** est le composant qui interagit avec le serveur de réseau (**NS**). Il met

- en œuvre un processus qui commence par traiter les trames entrantes, les filtre en fonction du type de message (liaison montante confirmée ou non, JoinRequest) et gère les terminaux ;
- le composant Device Context Manager récupère le contexte du terminal sur le serveur du réseau domestique contenant les informations F/SNwkSIntKey, NwkSEncKey, FCntUp, FCntDwn (spec 1.1) et DevAddr. Ensuite, il l'envoie au VLA correspondant. Une fois reçu, il est stocké dans le vNS ;
 - le Peer Manager est le composant qui gère l'échange de transmissions de trames entre le vNS et le hNS après que la JoinRequest a réussi et que le contexte du terminal soit parvenu à son extrémité. Il crée un tunnel direct avec le hNS pour réacheminer ses trames de données.

6.1.2 Procédure d'activation

Pour être associé et prendre part à un réseau **LoRaWAN**, un terminal doit le rejoindre. Du côté du terminal, la procédure d'adhésion consiste à envoyer une demande pour joindre ou rejoindre le réseau et à recevoir un JoinAccept de la part du réseau.

Comme décrit dans la section 2.2.3, **LoRaWAN** fournit deux méthodes d'activation différentes, à savoir **ABP** et **OTAA**. La méthode **ABP** est généralement utilisée à des fins de test et pour des déploiements non destinés à la production. Cette méthode exige que les clés de chiffrement soient codées en dur sur les terminaux. Par conséquent, la communication est effectuée sans *handshaking*. A l'opposé, **OTAA** est utilisé, selon la spécification **LoRaWAN**, lorsqu'un terminal est déjà déployé ou qu'il est réinitialisé. Cette méthode d'activation est nécessaire pour la mise en place de l'itinérance. En effet, **OTAA** engendre à chaque fois les clés de session pendant le *handshaking* où les messages JoinRequest et JoinAccept sont échangés. Au cours de cette procédure, les terminaux sont identifiés à l'aide des champs DevEUI et JoinEUI. La procédure d'association dans le cadre de la nouvelle architecture d'itinérance **LoRaWAN** proposée nécessite deux phases principales. :

Approvisionnement des terminaux en JoinEUI

Le JoinEUI est un identifiant global d'application dans l'espace d'adressage IEEE EUI64. Il est utilisé comme identifiant unique par le *Join Server* qui gère la procédure d'activation et déduit les clés de session. Rappelons que tout le long de cette thèse, le *Join Server* est considéré comme une entité faisant partie du **NS**.

Avant d'autoriser un terminal à se déplacer et à transiter par un réseau partenaire,

l'agent local domestique doit lui attribuer un JoinEUI spécifique. Le JoinEUI est engendré de manière unique par le HLA afin de permettre la résolution DNS pour la recherche du réseau domestique auquel le terminal appartient. En effet, le JoinEUI est divisé en deux champs égaux de quatre octets chacun, comme illustré dans la Figure 6.3. Les quatre octets les plus significatifs représentent un NET-ID unique attribué par le MA à chaque réseau LORAWAN participant au partenariat d'itinérance. À la réception des trames, ce champ est extrait et utilisé pour trouver le serveur de réseau domestique d'un terminal donné. Les quatre derniers octets les moins significatifs représentent un identifiant unique universel attribué par le HLA. Il convient de noter que l'utilisation de cette stratégie de création de JoinEUI garantit que tous les terminaux mobiles ayant des accords d'itinérance seront identifiés de manière unique et que le trafic engendré sera acheminé vers le HNS correspondant.

Field	Net-ID	uuid
Size (Byte)	4	4

FIGURE 6.3 Format de JoinEUI.

Adhésion au réseau

Une fois la phase d'approvisionnement terminée, les terminaux sont en mesure de se déplacer sur les réseaux partenaires. En effet, lorsque les terminaux quittent la zone de couverture fournie par leur réseau domestique et atteignent une autre zone couverte par un réseau partenaire, ils doivent pouvoir transiter par ce réseau afin de continuer à envoyer leur trafic vers leur réseau domestique. Dans ce paragraphe, nous présentons un protocole d'itinérance handover décrit comme suit :

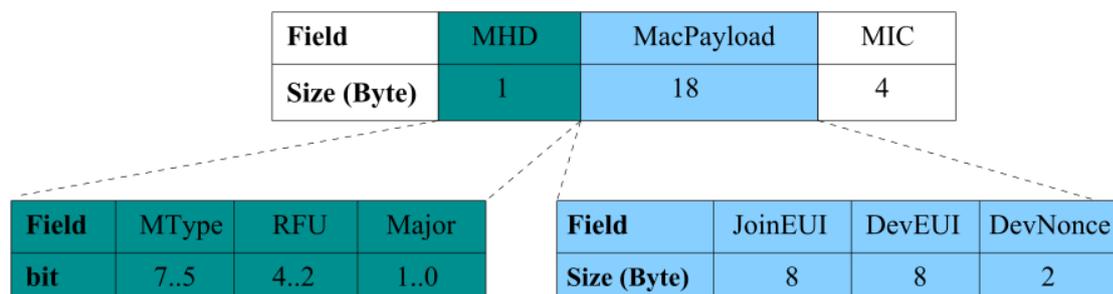


FIGURE 6.4 Format du message JoinRequest.

La procédure d'activation lors de la première visite du terminal d'un opérateur partenaire est résumée dans la Figure 6.5. Lorsque le handover est activé, le **vNS** gère les couches physique et **MAC**. Le terminal et le **vNS** doivent effectuer un *handshaking* (activation). Plus précisément, cette activation consiste en l'échange de deux messages : un **JoinRequest** envoyé par le terminal et un **JoinAccept** envoyé par le **vNS**. Le **JoinRequest** est envoyé en clair et contient le **JoinEUI** procuré dans la phase précédente, le **DevEUI**, et une valeur *nonce* comme indiqué dans la Figure 6.4. Comme il est envoyé en clair, toute personne interceptant le trafic peut lire son contenu, y compris le **vNS**. Lorsque le **vNS** reçoit des trames de sources inconnues, elles sont transmises au **VLA**. Le **VLA** effectue une vérification de la structure de la trame et du type de message. Si la vérification réussit, le **VLA** construit une requête qui se compose du **JoinRequest** reçue sur le **vNS** et d'un **VLAReq** qui contient l'adresse IP du **vNS** et son nom de domaine. Cette demande est ensuite envoyée au **MA**. Lorsque le **MA** reçoit la demande du **VLA**, il effectue une vérification de la source de la demande. Un **JoinRequest** valide envoyé par un appareil final itinérant doit contenir un **NET-ID** connu du **MA**. En effet, le **MA** extrait le **NET-ID** des 4 octets les plus significatifs du **JoinEUI** reçu dans le **JoinRequest**. Si le **MA** connaît ce **NET-ID**, il procède à une résolution **DNS** et transmet la requête au **HLA** correspondant. Ensuite, il répond au **VLA** avec un **DNSResp** comprenant l'adresse IP du **HNS** et son nom de domaine.

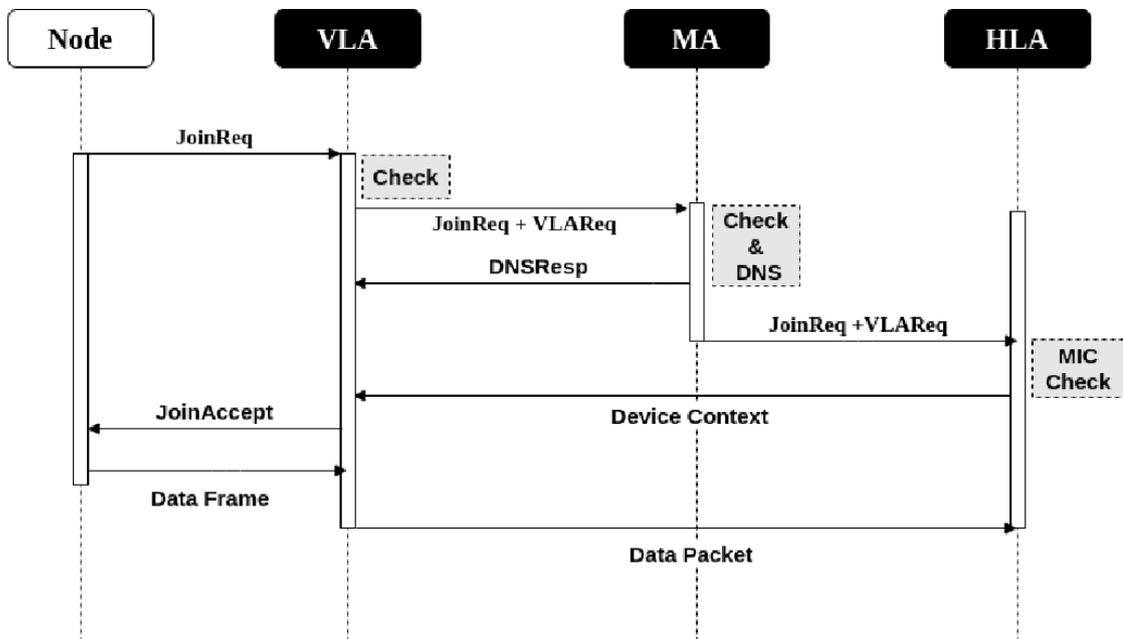


FIGURE 6.5 Protocole d'itinérance LoRaWAN.

Dès que la demande du **MA** est reçue par le **HLA**, celui-ci effectue un contrôle d'intégrité. En effet, seules les entités possédant la `NwkKey` peuvent affirmer que le message n'a pas été altéré en utilisant le code d'intégrité du message. En retour, le **HLA** répond au **VLA** correspondant en se servant des informations incluses dans le `VLAReq` en envoyant le contexte du terminal. Une fois que le contexte du terminal est reçu par le **VLA**, il est stocké sur le **vNS**. À ce stade, le **vNS** est en mesure de répondre au terminal par un `JoinAccept`. Une fois que le terminal est activé, il peut commencer à envoyer des trames. A la réception d'une trame sur le **vNS**, celui-ci la transmet au **hNS** correspondant en se fondant sur le `DNSResp` déjà stocké lors des phases précédentes. Notons que si un changement est intervenu sur l'adresse IP et/ou le nom de domaine du **hNS**, le **MA** envoie une mise à jour au **VLA**.

Toutes les communications entre le **VLA**, le **MA** et le **HLA** relatives à la résolution du **DNS** et à la récupération du contexte du terminal sont effectuées uniquement lors de la première visite du réseau ou pour la mise à jour du contexte du terminal. Bien sûr, lorsque le contexte du terminal est stocké sur le **vNS**, cela signifie que la couche MAC est entièrement gérée par le **vNS** qui est capable de gérer son activation. Ainsi, le **vNS** peut traiter les prochains `JoinRequests` ou `ReJoinRequests` provenant de terminaux mobiles connus dont un contexte valide a déjà migré et été stocké.

6.2 Implantation de l'itinérance LoRaWAN

Pour valider la conception de la solution d'itinérance **LoRaWAN** présentée dans la section 6.1, nous avons mis en œuvre et déployé un banc d'essai fondé sur `ChirpStack`. En effet, `ChirpStack` [35] est une stack **LoRaWAN**, *open source*, qui fournit des composants logiciels pour les réseaux **LoRaWAN**. Plus précisément, elle fournit un serveur de réseau et un serveur d'applications prêts à être utilisés ainsi qu'une interface web conviviale pour la gestion des terminaux et des `API` pour l'intégration d'autres services. Nous justifions l'utilisation de `ChirpStack` par son architecture modulaire, qui permet de l'intégrer aux infrastructures existantes et de l'ouvrir à d'éventuelles extensions.

La *stack* logicielle `Chirpstack`, jusqu'à la rédaction de ce manuscrit, implante une résolution **DNS** spécifique à des fins d'itinérance passive. Nous avons étendu l'architecture **LoRaWAN** avec trois nouveaux composants principaux pour mettre en œuvre un nouveau mécanisme de résolution **DNS** fondé sur le `JoinEUI` et pour délocaliser le contrôle de la couche MAC du terminal mobile qui sera dans ce cas maintenu par le **vNS** lors de son itinérance. Les implantations des composants sont détaillées comme suit :

- le **MA** est un composant central écrit en python, qui expose des points de

terminaison d'une [API](#) pour fournir deux services principaux : 1) il crée et distribue des [NET-ID](#) pour les opérateurs ; 2) ces [NET-ID](#) sont ensuite utilisés dans le second service pour la résolution [DNS](#) fondée sur le JoinEUI afin de trouver le réseau domestique auquel appartiennent les terminaux ;

- le [VLA/HLA](#) est un composant python qui est divisé en plusieurs sous-composants décrits comme suit : il invoque les services exposés par le [MA](#) pour l'abonnement des opérateurs et la résolution [DNS](#) en utilisant le composant [MA Client](#).

Toutes les interactions avec le [NS ChirpStack](#) sont gérées par le client ChirpStack via son [API](#). Ce sous-composant communique avec le Device Context Manager, soit pour gérer et stocker le contexte du terminal sur le [vNS](#) afin de prendre en charge la couche MAC, soit pour envoyer le contexte du terminal au [vNS](#) une fois qu'il a été récupéré et formaté par le Device Context Manager. Avec l'aide du paquet pyThingPark, nous avons implanté le contrôle du formatage de la trame et la vérification du [MIC](#) avant de procéder à la récupération du contexte du terminal.

Le sous-composant Peer Manager crée une connexion directe entre le [vNS](#) et le [hNS](#) après la réussite de la procédure d'association et la migration du contexte du terminal. En fait, à la réception d'un message d'un terminal mobile, le [VLA](#) transmet le message au [HLA](#) correspondant.

La première version V1.0 de la solution est disponible librement sur GitHub [61] et peut être consultée.

6.3 Scénarios expérimentaux

Pour mettre en œuvre et étudier le comportement de notre solution d'itinérance handover, nous avons sélectionné deux sites de déploiement de réseaux [LoRaWAN](#), séparés par une distance de 6 km dans une zone urbaine. Notre premier réseau est déployé à l'[ENSEEIH](#)T situé dans le centre-ville de Toulouse jouant le rôle du [hNS](#). Le second réseau est déployé à l'Institut de Recherche en Informatique de Toulouse (IRIT) sur le campus de l'Université Paul Sabatier (UPS) et joue le rôle du [vNS](#). Le [MA](#) est déployé sur le site de l'[ENSEEIH](#)T. La configuration du scénario expérimental est schématisée dans la Figure 6.6.

Dans le cadre de notre déploiement, nous avons utilisé deux passerelles de type Mikrotik wAP LR8b [129] sur les deux sites, comme illustré dans la figure 6.7a. Les deux passerelles sont configurées pour fonctionner en mode de transfert de paquets. Ainsi, à la réception d'une trame, la passerelle la transmet à son serveur de réseau domestique

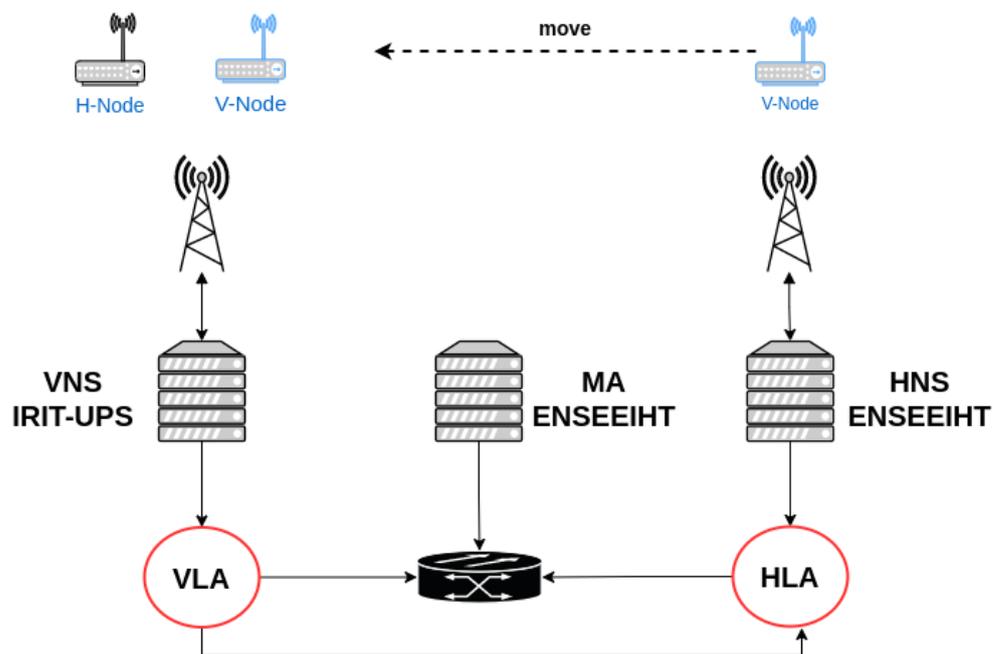


FIGURE 6.6 Configuration du scénario expérimental de l'itinérance LoRaWAN

ChirpStack. Toutes nos expériences sont menées en utilisant soit des terminaux Pycom Lopy4 ou Microchip RN2483 fixés sur un Raspberry Pi3. Tous les terminaux sont alimentés par une source d'énergie externe, comme le montrent les figures 6.7b- 6.7c.



(a) Passerelle Mikrotik



(b) Pycom Lopy4



(c) Microchip RN2483

FIGURE 6.7 Matériel expérimental

Avant de considérer la mobilité des terminaux, les réseaux IRIT-UPS et ENSEEIHT doivent souscrire au niveau du MA pour obtenir un NET-ID unique. En effet, le MA maintient une table de correspondance qui associe chaque nom de domaine et adresse IP aux réseaux correspondants sur la base du NET-ID attribué. Le tableau 6.1 donne un extrait de la table des réseaux enregistrés sur le MA pour la configuration du réseau décrite dans la section 6.3.

Réseau	Net-ID	Nom de Domaine	IP
hNS ENSEEIHT	4C683808	hns.io	ip @
vNS IRIT-UPS	CBEF7B6E	vns.io	ip @

TABLE 6.1 Tableau des souscriptions au réseau itinérant

Le **HLA** utilise le **NET-ID** attribué pour créer des JoinEUIs qui seront fournis aux terminaux mobiles. En pratique, le **HLA** maintient une table de correspondance qui associe les JoinEUIs au terminal mobile. Un extrait de cette table est donné dans le Tableau 6.2. A titre d'exemple, en supposant que le **HLA** obtienne un **NET-ID** correspondant à **4C683808**, il fournit des JoinEUI uniques fondés sur le **NET-ID** comme suit : les quatre octets les plus significatifs représentent le **NET-ID** suivi de quatre octets uniques choisis par l'opérateur du réseau domestique. Par exemple, le **HLA** crée le JoinEUI suivant : **4C683808E77E4E01** pour le terminal lopy4-1.

Terminal	DevEUI	JoinEUI
lopy4-1	BEEFDEADBEEF0001	4C683808E77E4E01
lopy4-2	BEEFDEADBEEF0002	4C683808A9713E02
lopy4-3	BEEFDEADBEEF0003	4C683808C6768E03
RN2483-1	BEEFDEADBEEF0004	4C683808E27DCE04
RN2483-2	BEEFDEADBEEF0005	4C683808B09E5E05
RN2483-3	BEEFDEADBEEF0006	4C6838081F73E060

TABLE 6.2 Exemples de JoinEUI déduits

La formulation d'une telle solution pour l'itinérance handover reposant sur la résolution **DNS** et la migration du contexte du terminal depuis le **hNS** vers le **vNS** soulève des inquiétudes, notamment en ce qui concerne la latence induite par le protocole lors de la première visite du terminal. Cette latence affecte directement les fenêtres de réception de la liaison descendante du terminal de classe A. En effet, la classe A est décrite comme la moins coûteuse en termes de consommation d'énergie, ce qui se traduit par une plus grande autonomie de la batterie. Évidemment, le mécanisme proposé a un coût puisque la plupart du temps, le terminal est en veille. Ce coût peut être traduit comme suit : après avoir envoyé une trame sur la liaison montante et un éventuel JoinRequest, le terminal de classe A ouvre une fenêtre de réception pendant une seconde pour écouter une trame descendante, un éventuel JoinAccept. Si aucune trame sur la liaison descendante n'est reçue pendant cette fenêtre, il ouvre une deuxième fenêtre de réception pendant une seconde supplémentaire.

Nous allons analyser et évaluer l'impact des délais induits par la résolution **DNS**

pour la recherche du réseau domestique des terminaux et la migration de contexte sur les fenêtres de réception des terminaux de la classe A. Nous avons fourni un algorithme de monitoring dans le **VLA/HLA** pour mesurer le temps total allant de la réception initiale du JoinRequest à la migration du contexte et son stockage sur le **vNS**. Ce temps est donné comme suit :

$$T = T_{DNS}(MA) + T_{rContext}(HLA) + T_{sContext}(VLA) \quad (6.1)$$

L'équation 6.1 donne le délai T causé par le protocole relativement à la résolution du **DNS** et à la migration du contexte lors de la première visite du réseau par le terminal. $T_{DNS}(MA)$ représente le temps de la résolution du **DNS** par le MA, $T_{rContext}(HLA)$ le temps de récupération du contexte du terminal à partir du **hNS** et $T_{sContext}(VLA)$ le temps de migration de ce contexte et son stockage sur le **VLA**.

Après avoir été activés au niveau du réseau domestique, les terminaux ont été déplacés vers une zone non couverte par leur opérateur. Nous avons rapporté dans la Figure 6.9 les délais enregistrés par les terminaux mobiles lors de la première visite du réseau partenaire. Plusieurs demandes d'activations ont été envoyées par la méthode **OTAA** pour rejoindre le réseau visité. Les expériences ont été effectuées en considérant que les terminaux n'ont pas encore rejoint le réseau visité et qu'aucun contexte correspondant n'est stocké sur le **vNS**.

Pour obtenir une plus grande précision des résultats expérimentaux, trois scénarios sous différentes charges de trafic ont été envisagés :

Paramètres	SF	PL	BW	Temps de vol
Valeur	12	3 octets	125 kHz	827.39 ms

TABLE 6.3 Paramètres de transmission du terminal domestique.

Soit ρ la charge de trafic [66] induite par les terminaux domestiques en utilisant les paramètres décrits dans le tableau 6.3.

$$\rho = \frac{N \times T_{SF12}}{C \times p} \quad (6.2)$$

La charge de trafic ρ est calculée à l'aide de l'équation (6.2), où T_{SF12} est le temps de vol en utilisant SF 12, N est le nombre de terminaux domestiques, C est le nombre de canaux de réception au niveau de la passerelle ($C = 3$) et p est la période de génération des trames par terminal.

Dans nos expériences, nous avons défini les paramètres de transmission comme indiqué

dans le tableau 6.3 et considéré une période de génération de paquets $p = 3$ secondes. Le choix de cette configuration expérimentale est motivé par le but de l'expérience qui est de charger fortement le réseau. Pour ce faire, nous pouvons utiliser de nombreux terminaux, ce qui n'est pas pratique en raison du manque de matériel et de la complexité de configuration. On peut également fixer le temps de vol au maximum (cette valeur est obtenue en utilisant SF 12) et la période d'envoi de trames par terminal au minimum. Le paramétrage de p à 3 secondes est justifié par la spécification du terminal **LORAWAN** de la classe A, où deux fenêtres ouvertes d'une seconde chacune sont considérées après l'envoi d'une trame ascendante. Nous avons pu vérifier cette valeur expérimentalement.

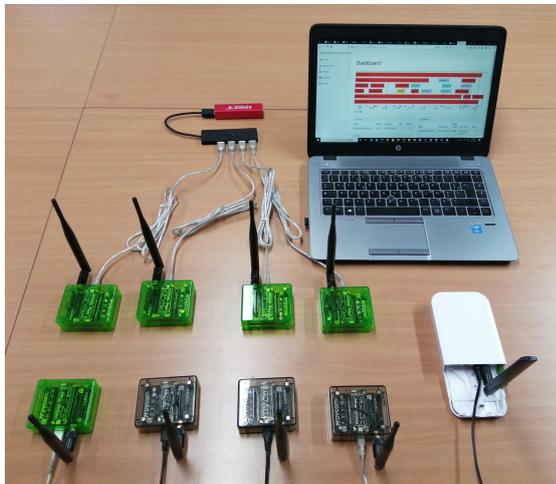


FIGURE 6.8 Configuration matérielle des expérimentations

6.3.1 Trafic faible

Dans ce scénario, nous avons considéré une expérience de contrôle avec un réseau sans aucun trafic envoyé par les terminaux domestiques. Plus précisément, nous n'avons pris en compte que le trafic engendré par le terminal mobile visitant l'opérateur avec qui des accords d'itinérance sont établis, en supposant qu'aucun trafic n'est envoyé par les terminaux domestiques. Par conséquent, la charge de trafic des terminaux domestiques est de $\rho = 0$.

6.3.2 Trafic moyen

Dans ce scénario, nous avons considéré une configuration de réseau composée de $N = 4$ terminaux domestiques envoyant des trames au **HNS** avec une période $p = 3$

secondes. La charge de trafic induite est calculée à l'aide de l'équation 4.2, on obtient $\rho = 0,37$. Un terminal mobile envoie des JoinRequest au vNS pendant que les terminaux domestiques envoient des trames.

6.3.3 Trafic saturé

Dans ce scénario, nous avons considéré une configuration de réseau similaire à la précédente. Cependant, nous avons étendu la taille du réseau en considérant $N = 8$ terminaux domestiques (voir la configuration donnée dans la figure 6.8) envoyant des trames à une fréquence d'une trame chaque $p = 3$ secondes. Cela induit une charge de trafic de $\rho = 0,74$.

6.3.4 Interprétation des résultats d'expériences

Les figures 6.9a et 6.9b représentent la fonction de répartition empirique du délai (T) nécessaire à la résolution DNS et à la migration du contexte du hNS vers le vNS, mesurée par de multiples activations OTAA. Il convient de noter que T est enregistré lors de la première visite du réseau partenaire par le terminal mobile sous différentes charges de trafic ρ . Les résultats indiquent que le contexte peut être récupéré et que le terminal est activé en moyenne en 360 ms voire 450 ms dans les cas les plus défavorables lors de l'itinérance dans le réseau partenaire.

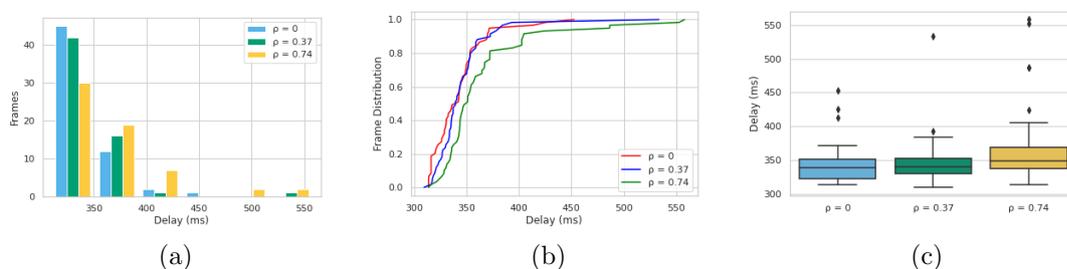


FIGURE 6.9 Évaluation des performances en termes de latence d'activation pour la première visite du terminal dans un scénario d'itinérance.

Par conséquent, 95% des terminaux peuvent être enregistrés et activés sur le vNS en un temps maximum de 380 ms pour une charge de trafic ρ de 0,37 et de 420 ms pour un ρ de 0,74. Cela prouve que les mécanismes d'itinérance proposés n'augmentent pas de manière significative le temps d'activation, même en cas de forte charge de trafic. Ainsi, ils se conforment bien à la contrainte des terminaux de Classe A qui correspond aux deux fenêtres de réception de 2s ouvertes pour la réception des trames descendantes.

Rappelons que ces résultats correspondent à la première visite d'un réseau partenaire par un terminal. En effet, après la migration du contexte, le **vNS** gère la couche **MAC**, donc la procédure d'activation du terminal. Par conséquent, lorsque ce même terminal tente de rejoindre le réseau, cela ressemble à une activation classique entre un terminal et son **hNS**.

La figure 6.9c confirme les résultats décrits dans le dernier paragraphe. En effet, elle montre que la majorité des terminaux sont enregistrés et activés en moins de 360 *ms*. Elle montre également quelques valeurs aberrantes dues aux appels de l'**API** ChirpStack et à la disponibilité des paquets sur le *broker* **MQTT** du **vNS** qui fait partie du logiciel ChirpStack. Comme défini dans la documentation ChirpStack, ce *broker* est la partie où les trames sont poussées depuis le *packet forwarder* ChirpStack une fois reçues sur la passerelle.

6.4 Scénarios d'émulation

Pour mieux évaluer notre solution et montrer sa pertinence, nous avons conçu et implanté un émulateur appelé *LoRaWAN Roaming Emulator* (**LRE**).

La conception d'un émulateur pour l'itinérance dans les réseaux **LoRaWAN** est motivée par deux objectifs principaux. D'une part, il s'agit de faire face au nombre limité de terminaux disponibles tout en réalisant des expériences à grande échelle. D'autre part, cela permet de surmonter la tâche fastidieuse et chronophage de déploiement de centaines de terminaux. En outre, cet outil fournit une nouvelle approche pour étudier l'itinérance **LoRaWAN** et valider certains aspects tels que la mise à l'échelle.

6.4.1 Conception de l'émulateur d'itinérance LoRaWAN

L'émulateur est constitué de deux composants principaux comme illustré dans la figure 6.10. La description du **LRE** est donnée comme suit :

- le *Generator Component* (**GC**) : ce composant remplace les terminaux réels et les émule. Tout d'abord, il crée des terminaux à l'aide du sous-composant *End-Device Generator* (**EDG**). Plus précisément, cette partie crée des terminaux dans le **hNS** en leur attribuant des devEUIs, des AppKeys et calcule les clés d'authentification à partir de l'AppKey attribuée. Ensuite, le *JoinRequest Generator* (**JRG**) envoie des messages JoinRequest pour les terminaux créés. Rappelons que la structure d'un JoinRequest est composée de trois champs principaux comme suit : JoinEUI (8 octets), DevEUI (8 octets), DevNonce (2 octets). Le JoinEUI est forgé sur la base du **NET-ID** attribué en adoptant la même stratégie que celle développée dans

- la section 6.1.2. Les terminaux et les JoinRequest sont enregistrés et stockés dans une base de données ;
- le *Roaming Protocol Component* (RPC) : ce composant émule l'échange de messages pour le protocole d'itinérance illustré dans la figure 6.5. Il récupère les joinRequest dans la base de données et les utilise pour alimenter le protocole d'itinérance afin d'effectuer la résolution DNS et la migration du contexte de l'appareil final du hNS vers le vNS.

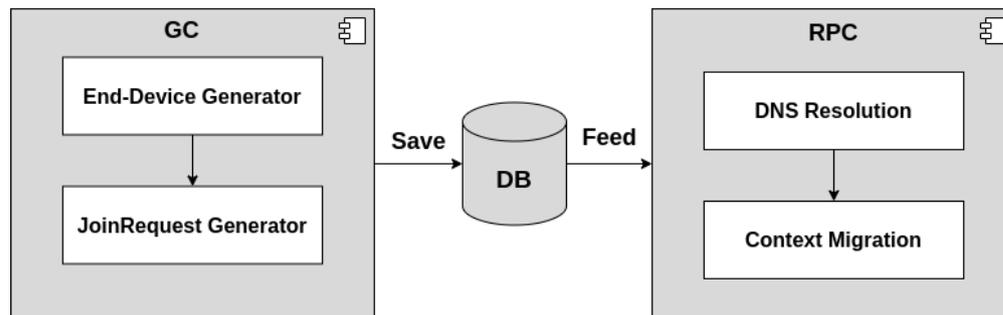


FIGURE 6.10 Architecture de l'émulateur d'itinérance LoRaWAN.

6.4.2 Description de l'émulation d'itinérance LoRaWAN

La figure 6.11 donne une description du processus de l'émulation d'itinérance en utilisant notre émulateur LRE.

Dans ce qui suit, nous donnons la description du *workflow* d'émulation du protocole de *roaming* entre plusieurs opérateurs. Plus précisément, l'émulateur commence par la création du terminal sur le hNS en lui attribuant un ensemble de paramètres comprenant le nom du terminal, le devEUI et le joinEUI. Ensuite, il procède à son activation et à la création de son contexte en générant les clés d'authentification composées de l'AppKey, des clés de réseau et d'application, du devAddr et des compteurs de trames. Une fois les terminaux émulés créés, l'émulateur prépare pour chaque terminal un joinRequest et le stocke dans une base de données. Enfin, les joinRequest stockés sont utilisés pour s'abonner au vNS. Notons que pour fournir des scénarios d'émulation réalistes, nous avons conçu un générateur de requêtes *multi-threading*(MRG). En effet, chaque *thread* gère une exécution du protocole d'itinérance pour un terminal donné, depuis le traitement du JoinRequest jusqu'à la résolution du DNS et la migration du contexte du terminal depuis le réseau domestique vers le réseau visité.

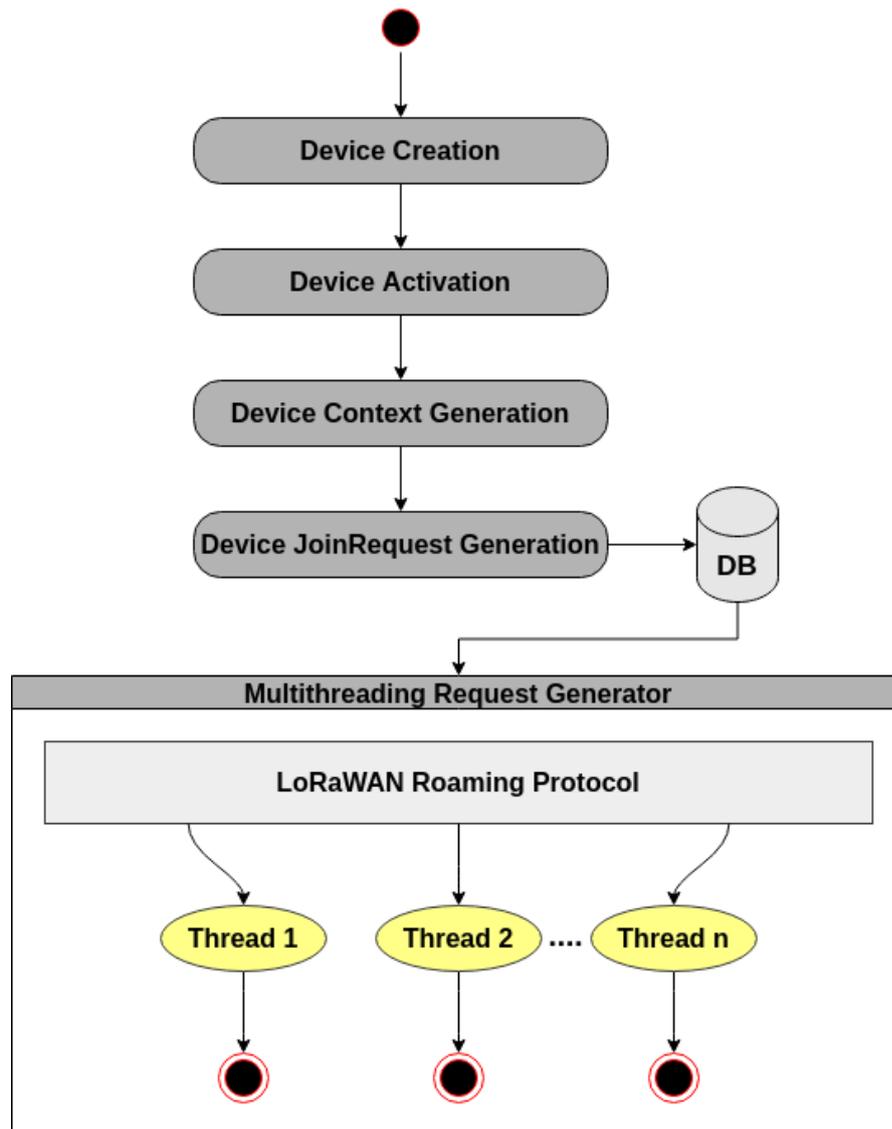


FIGURE 6.11 Description de l'émulation d'itinérance LoRaWAN

6.4.3 Configuration de l'environnement d'émulation

Pour évaluer la solution d'itinérance proposée par émulation, nous avons considéré une configuration d'un environnement expérimental, telle que résumée dans le tableau 6.4, composée de deux déploiements d'opérateurs réseau **LoRaWAN**, respectivement un serveur réseau visité (**vNS**) avec un **VLA** et un serveur réseau domestique (**hNS**) avec **HLA** déployés dans deux machines virtuelles Ubuntu. Nous avons également déployé un **MA** sur la machine hôte. La configuration du déploiement de l'émulation est illustrée

dans la figure 6.12. L'ensemble de l'environnement d'émulation est déployé sur une seule machine hôte avec 8 CPU et 16 Go de RAM.

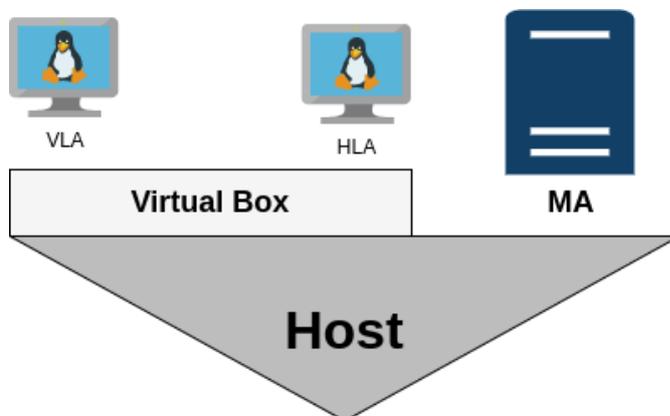


FIGURE 6.12 Configuration du déploiement de l'émulation.

6.4.4 Paramètres d'émulation

Les paramètres de transmission pris en compte dans nos émulations pour le calcul du temps de vol sont donnés ci-dessous et résumés dans la Table 6.4 :

TABLE 6.4 Paramètres d'émulation

Paramètres de transmission	Valeurs
Payload Length (PL)	23 octets
Bandwidth (BW)	125 kHz
Coding Rate (CR)	1/4
Spreading Factor (SF)	Aléatoire (7,12)
Temps de vol	en fonction de (BW,PL,CR,SF)
Environnement	Valeurs
Terminaux émulsés	10-100
Opérateurs	2 [hNS, vNS]
Système	2 * Ubuntu 18 Machines Virtuelles
CPU	8 Cœurs
Mémoire	16 GB

- Payload Length (PL) : dans ce scénario d'émulation, nous prêtons attention aux trames joinRequest qui sont composées d'un en-tête suivi d'un JoinEUI, d'un DevEUI et d'un devNonce dans le champ de charge utile physique. Par conséquent, la longueur de la charge utile est fixée à 23 octets, ce qui correspond à une trame joinRequest ;

- Bandwidth (**BW**) : elle est choisie dans un ensemble de trois valeurs possibles $[125 \text{ kHz}, 250 \text{ kHz}, 500 \text{ kHz}]$. Dans ces émulations, nous n'avons considéré que la bande 125 kHz car la plupart des déploiements **LoRaWAN** utilisent cette bande passante, tandis que les bandes de 250 kHz sont rarement utilisées. Enfin, la bande de 500 kHz est réservée à la liaison descendante ;
- Coding Rate(**CR**) : comme défini dans la section 2.1.2, ce paramètre est utilisé pour la correction des erreurs afin d'améliorer la protection contre les interférences. Dans cette émulation, nous avons considéré $CR = 4/5$ car il s'agit de la valeur par défaut des équipements **LoRaWAN** et nous ne sommes pas particulièrement préoccupés par la robustesse des communications ;
- Spreading Factor (**SF**) : dans les émulations, nous avons considéré un tirage aléatoire (et uniforme) du facteur d'étalement pour chaque JoinRequest généré, allant de 7 à 12.

Ces paramètres sont utilisés pour calculer le temps de vol de chaque JoinRequest envoyé. Ensuite, pour ce temps dû à la transmission radio, tous les JoinRequest envoyés au **vNS** sont retardés d'une période égale au temps de vol calculé.

6.4.5 Résultats de l'émulation de l'itinérance

Dans cette section, nous présentons les résultats de l'émulation de l'itinérance en termes de latence d'activation des terminaux pour la première visite du réseau partenaire, puis d'utilisation du **CPU** et de la mémoire pour chaque composant qui étend l'architecture **LoRaWAN**, voir par le **MA**, le **VLA** et le **HLA**.

TABLE 6.5 Latence d'activation pour la première visite du terminal du réseau partenaire

Terminaux	Médiane (ms)	Moyenne (ms)
10	282.8	283.5
20	296.7	307.2
30	289.4	307.7
40	296.4	323.8
50	307.2	348.3
100	344.5	361.9

La figure 6.13 et le tableau 6.5 montrent que la solution d'itinérance handover proposée donne de bons résultats en termes de délai occasionné par la résolution **DNS** et la migration du contexte du terminal depuis le **hNS** vers le **vNS**. Les résultats indiquent que le temps moyen pour récupérer le contexte et activer les terminaux est de 283,5 ms pour un scénario impliquant 10 terminaux et de 361,9 ms pour un scénario avec 100

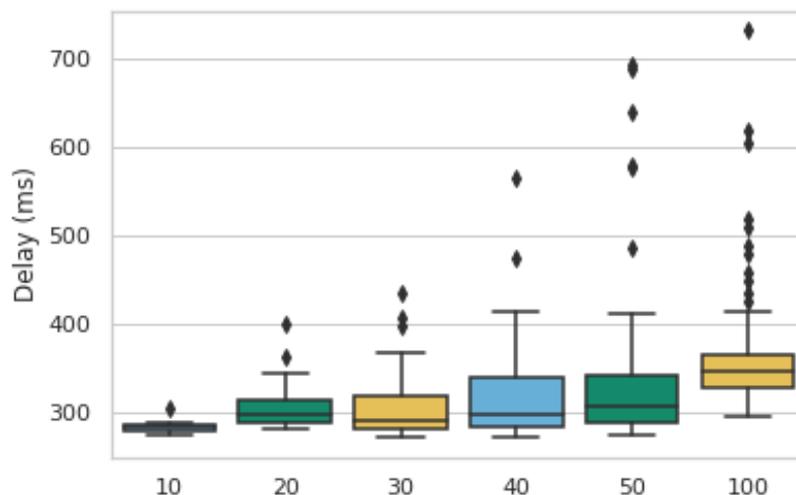


FIGURE 6.13 Latence d'activation dans un scénario d'itinérance.

terminaux. Cela démontre que les mécanismes d'itinérance conçus n'augmentent pas de manière significative le temps d'activation avec le nombre de terminaux itinérants. Ainsi, ces délais répondent bien aux exigences de la classe A des terminaux en termes de fenêtres de réception de 2s ouvertes lors de l'écoute sur la voie descendante.

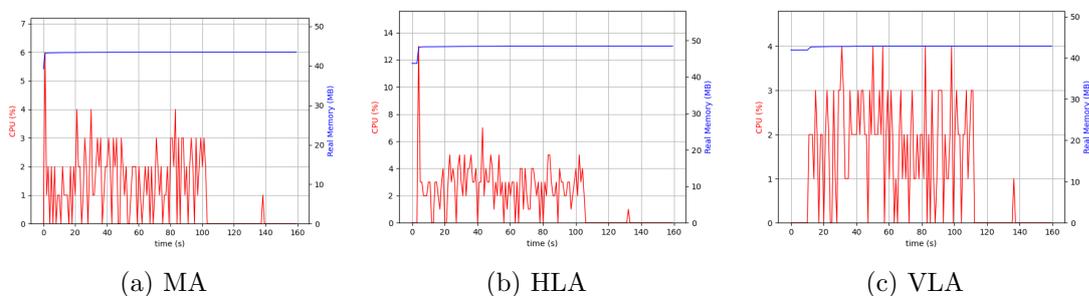


FIGURE 6.14 Évaluation des performances en termes d'utilisation du processeur et de la mémoire.

Dans les figures 6.14a, 6.14b et 6.14c, nous avons tracé l'utilisation du CPU et de la mémoire pour le MA, le HLA et le VLA, respectivement, dans un scénario où 100 terminaux tentent de rejoindre le réseau de l'opérateur visité après avoir été activés sur leur serveur de réseau domestique.

Le tracé en rouge représente le pourcentage d'utilisation de la CPU pendant l'émulation.

Les pulsations enregistrées correspondent à la réception et au traitement des requêtes sur les différents composants (**MA**, **HLA**, **VLA**). Le tracé en bleu représente l'utilisation de la mémoire par les processus des composants une fois déployés et lors du traitement des requêtes.

Les figures illustrent la différence de charge **CPU** et d'utilisation de la mémoire avant que le générateur n'envoie les JoinRequest et pendant le traitement des demandes, capturées dans une fenêtre de 160 secondes. L'utilisation du processeur par le **MA**, le **HLA** et le **VLA** augmente légèrement lors de la réception de demandes envoyées par un terminal itinérant, pour atteindre une utilisation de 4% d'un seul cœur de processeur. Le programme résidant dans la mémoire lors du traitement des demandes est de 43,4 *Mo* pour le **MA** et le **VLA** et de 48,3 *Mo* pour le **HLA**.

6.5 Conclusions et discussions

Les résultats de l'évaluation des performances en termes de latence d'activation pour la première visite d'un terminal itinérant suggèrent, dans l'approche expérimentale, que le contexte peut être récupéré et le terminal activé en une moyenne de 360 ms et 361,9 ms en considérant 100 terminaux pour l'approche d'émulation. La comparaison entre les résultats de ces deux approches montre à la fois l'efficacité de la solution d'itinérance proposée en termes de contraintes de délai pour les terminaux de classe A et l'efficacité de notre solution pour émuler des environnements réels. Cela souligne la pertinence de disposer d'un émulateur efficace, proche d'un environnement réel, et d'un émulateur dédié pour étudier les performances de l'itinérance dans les réseaux **LoRaWAN**.

Les résultats de l'émulation de l'itinérance ont également montré l'efficacité de la solution d'itinérance pour fournir une réponse rapide aux demandes des terminaux visitant un réseau partenaire sans surcharger les **CPU** et les mémoires des serveurs hébergeant les composants **MA**, **VLA** et **HLA**.

Par ailleurs, la liaison descendante **LoRaWAN** dans un scénario d'itinérance est l'une des principales préoccupations actuelles des réseaux que la recherche devrait aborder et prendre en compte. La mise en œuvre de la solution d'itinérance *handover* apporte déjà quelques réponses à cette préoccupation. En effet, contrairement à l'itinérance passive, lors de l'adoption de l'itinérance *handover*, le **vNS** gère les terminaux dans le réseau, grâce à la migration du contexte du terminal. Ainsi, la couche **MAC** est entièrement gérée par l'opérateur visité, qui est capable d'envoyer des messages en liaison descendante contenant des commandes **MAC** pour l'adaptation du débit de données. Cela n'est pas possible en itinérance passive sans fournir de nouveaux mécanismes pour transférer les

messages de la liaison descendante du **HNS** au **VNS** et enfin au terminal.

Des perspectives sont envisageables pour aller plus loin dans l'analyse de notre solution. Par exemple, on pourrait envisager un algorithme d'élection sur le **MA** afin d'élire le meilleur **VNS** parmi ceux qui sont disponibles et ainsi de mieux gérer les terminaux mobiles en définissant une certaine stratégie.

Troisième partie

Liaison Descendante LoRaWAN

Même si les réseaux [LPWAN](#) sont majoritairement conçues pour des communications dans le sens montant, LoRaWAN autorise aussi les transmissions en liaison descendante de manière distincte en réponse aux trames en liaison montante. La couche [MAC](#) de [LoRAWAN](#) propose trois classes de terminaux pour répondre aux besoins des différentes applications, à savoir la classe A, la classe B et la classe C. Chaque terminal choisit l'une de ces trois classes. Le détail de chaque classe est donné dans la section [2.2.2](#). Le mode de réception des trames sur la liaison descendante (synchronisé, non synchronisé) dépend de la classe du terminal. Dans cette partie, nous exposons [LoRADL](#), le premier *framework* qui prend en charge les messages de la liaison descendante pour l'envoi de commandes [MAC](#) ou de données applicatives aux terminaux de la classe A.

Organisation de la partie

Cette partie comprend deux chapitres. Dans le premier, nous proposons un état de l'art des travaux portant sur l'étude de la liaison descendante. Nous présentons également la structure d'une commande [MAC](#) et son encapsulation dans une trame [LoRAWAN](#) nécessaire à la compréhension des mécanismes proposés.

Dans le second, nous présentons [LoRADL](#), notre solution pour prendre en charge les communications sur les liaisons descendantes [LoRAWAN](#) et la mise en place de tests pour l'étude de leurs performances.

ETAT DE L'ART

7.1 Politique sur la liaison descendante en LoRaWAN

Les terminaux de classe A sont en mesure d'envoyer des trames sur la liaison montante selon un mode **confirmé ou non confirmé** décidé avant chaque transmission. Dans le premier cas, aucun accusé de réception n'est envoyé par le serveur de réseau au terminal. Dans le second, un accusé de réception peut lui être envoyé via une passerelle. Le terminal est en mesure de recevoir l'accusé de réception sur l'une des deux fenêtres de réception (RX1 et RX2) qui s'ouvrent respectivement une et deux secondes après la transmission sur la liaison montante. Le terminal ne dispose pas de mécanismes pour prévoir la fenêtre sur laquelle l'accusé de réception sera reçu. S'il ne reçoit rien dans la première fenêtre, il doit allumer sa radio dans la seconde fenêtre également. Dans le cas où aucun message n'est reçu dans RX2 non plus, la trame de la liaison montante est retransmise après un temps aléatoire sur un canal de liaison montante différent. Cela peut soulever un problème de surconsommation énergétique par le terminal due aux retransmissions dans le cas où la passerelle a épuisé ses ressources dédiées aux transmissions sur la liaison descendante en raison de la contrainte du rapport cyclique.

Les messages sur la liaison descendante ne sont pas limités aux accusés de réception dans le cas de transmissions de trames confirmées. En effet, un terminal peut recevoir une **trame de contrôle** contenant des commandes **MAC** ou une **trame de données applicatives** sur l'une de ses fenêtres de réception (RX1 ou RX2) après une trame de liaison montante confirmée ou non confirmée. Dans notre contribution, nous nous proposons de fournir une solution pour prendre en charge les trames sur la liaison descendante contenant des commandes **MAC** et/ou des données applicatives.

7.2 Commande MAC LoRaWAN

7.2.1 Encapsulation des commandes MAC

Pour l'administration du réseau, un ensemble de commandes **MAC** est échangé entre le serveur de réseau et la couche **MAC** du terminal. Ces commandes ne sont pas visibles pour le serveur d'application ni pour l'application fonctionnant sur le terminal.

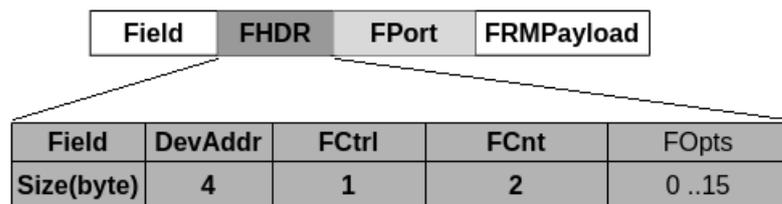


FIGURE 7.1 Structure de la charge utile MAC

Une seule trame de données (voir la figure 7.1) peut contenir n'importe quelle séquence de commandes **MAC**, injectées dans le champ **FOpts** ou envoyées comme une trame de données constituée de la séquence de commandes dans le champ **FRMPayload**, avec le champ **FPort** à 0. Les commandes **MAC** injectées dans le champ **FOpts** sont toujours envoyées chiffrées et ne peuvent pas dépasser 15 octets. Les commandes **MAC** envoyées en tant que **FRMPayload** sont également chiffrées et ne peuvent pas dépasser la longueur maximale du **FRMPayload**.

7.2.2 Structure des commandes MAC

Une commande **MAC** se compose d'un **identifiant de commande CID** d'un octet, suivi d'une séquence d'octets spécifiques à la commande, éventuellement vide pour les commandes **MAC** sans charge utile. La longueur des octets suivants dépend du type de commande **MAC** identifié dans le **CID** précédent. La table 7.1 donne un extrait de la liste des commandes **MAC** définies dans la spécification **LORAWAN** [7], [5] et [6]. Les commandes **MAC** peuvent être envoyées individuellement ou dans un groupe de commandes simultanées.

Pour bien illustrer la structure d'une commande **MAC**, nous prenons comme exemple la commande **Link ADR Request (LinkADRReq)** envoyée par un serveur de réseau à un terminal en liaison descendante pour mettre à jour ses paramètres de transmission (SF, TP, CH). Cette commande est identifiée par le **CID 0x03**, et comporte quatre autres octets, comme l'illustre le schéma suivant 7.2, qui montre la structure de la commande **LinkADRReq**. Cette commande est composée des champs suivants :

TABLE 7.1 Extrait de la liste des commandes MAC LoRaWAN

CID	Commande	Transmise par		Description
		Terminal	NS	
0x03	LinkCheckReq	X		Utilisée par un terminal pour vérifier sa connectivité à un réseau.
0x03	LinkCheckAns		X	Réponse au LinkCheckReq.
0x03	LinkADRReq		X	Demande de modifier le SF, la TP, la redondance ou le canal.
0x03	LinkADRAns	X		Accuse réception du LinkADRReq
0x06	DevStatusReq		X	Demande l'état du terminal.
0x06	DevStatusAns	X		Renvoie l'état du terminal (niveau de batterie et son état radio).

- **Data Rate** : correspond aux 4 bits de poids fort du champ **Requested Data Rate and TX Output Power (DataRate_TXPower)** de la commande LinkADRReq. Cela indique le nouveau débit de données (SF) que le terminal doit utiliser lors de l'envoi de trames sur la liaison montante.
- TX Output Power** : correspond aux 4 bits de poids faible du champ **Requested Data Rate and TX Output Power (DataRate_TXPower)** de la commande LinkADRReq. Cela indique la nouvelle puissance de transmission TP que le terminal doit utiliser lors de l'envoi sur la liaison montante ;
- **Channel Mask** : correspond au canal (CH) que le terminal doit utiliser lors de l'envoi dans le sens montant ;
- **Channel Mask Control** contrôle l'interprétation du ChMask précédemment défini ;
- **Number of Transmissions Per Uplink Frame** : correspond aux bits [3 : 0] du champ de redondance et indique le nombre de retransmissions de chaque trame envoyée sur la liaison montante.

7.3 Travaux pertinents

Dans cette section, nous présentons les travaux portant sur l'étude des performances et des mécanismes des réseaux **LoRaWAN** tenant compte des messages envoyés en liaison descendante. Nous proposons de classer ces travaux en deux catégories selon la méthode de validation retenue (modélisation ou expérimentation).

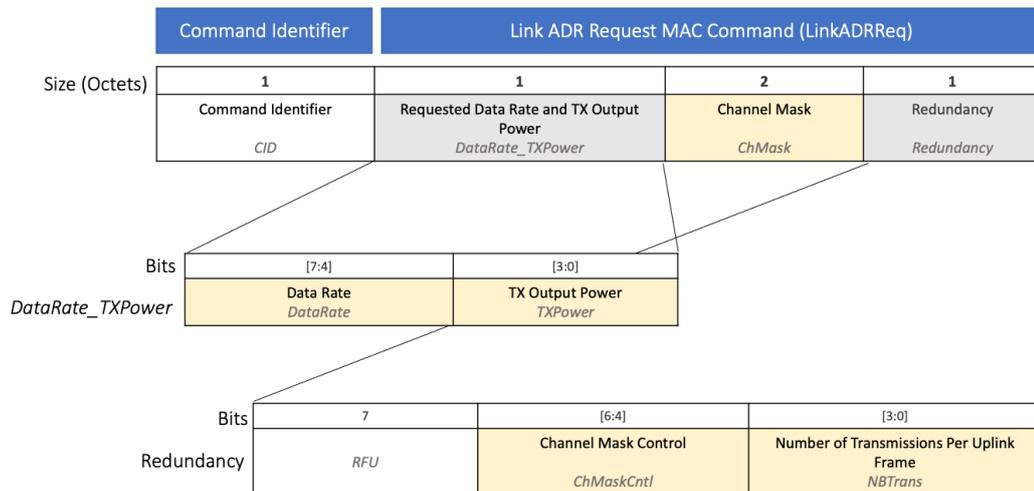


FIGURE 7.2 Structure de la commande LinkADRReq [2]

7.3.1 Études fondées sur la modélisation

Des études sur la mise à l'échelle des réseaux **LORAWAN** sont présentées dans [55], [99], [128], [56]. Les auteurs dans [128] ont proposé un modèle d'erreur pour la modulation LoRa pour différents couples taux de codage – **SF**. Ils ont effectué une implantation complète du protocole **LORAWAN** en utilisant le simulateur NS-3. Plus précisément, ils ont mené une étude sur la mise à l'échelle axée sur l'impact des messages confirmés par rapport aux messages non confirmés, et sur l'impact du trafic en liaison descendante dans les réseaux **LORAWAN** à grande échelle. Leur analyse montre l'impact négatif du trafic descendant sur le taux de livraison du trafic montant confirmé. En revanche, l'augmentation de la densité des passerelles peut limiter cet effet, mais pas l'éliminer. Dans [55], les auteurs ont évalué le passage à l'échelle de la transmission sur la liaison descendante dans la classe B où les trames de la liaison descendante sont envoyées dans un canal avec des rapports cycliques de 1% et 10%. Les résultats de la simulation ont montré que l'utilisation de la sous-bande avec un rapport cyclique de 10% offrait un meilleur débit que celle avec un rapport cyclique de 1% ; un rapport cyclique de 10% permet bien évidemment à la passerelle de passer plus de temps à transmettre qu'un rapport de 1%. Il en résulte que le passage à l'échelle du réseau est limité par l'utilisation du rapport cyclique du côté de la passerelle.

Des analyses qui prennent en compte la collision des trames ont été menées dans [22],

[23], [54] et [90]. Les auteurs de [54] ont proposé une approche plus précise et spécifique pour prédire les collisions et les pertes de trames. En particulier, le cas de la transmission de trames confirmées en liaison descendante a été inclus dans leur modèle. Dans [22], un modèle mathématique a été fourni pour évaluer la performance de **LORAWAN** lorsque les trames montantes sont envoyées en mode confirmé. Les auteurs ont expliqué que l'utilisation de l'approche classique d'accès au support de type ALOHA sous-estime la probabilité de collision et ont développé un modèle mathématique précis qui tient compte des particularités de **LORAWAN** liées aux politiques de retransmission. Les auteurs de [23] ont étendu le modèle mathématique existant dans [22] en prenant en compte les pertes dues à la propagation et à l'effet de capture, c'est-à-dire la possibilité qu'une trame soit reçue même si elle croise dans le temps des trames provenant d'autres terminaux, mais que sa puissance est suffisamment élevée. Dans [90], afin de maximiser la probabilité moyenne de succès des trames du réseau, une méthode d'allocation sous-optimale des SF est proposée pour des terminaux uniformément répartis dans l'espace. Il est démontré que la méthode proposée surpasse les schémas de répartition de **SF EAB** et **EIB**.

L'analyse des délais est une préoccupation capitale dans le domaine de la recherche **LORAWAN**. Les auteurs dans [43] ont évalué la performance du délai des trames en liaison descendante confirmées pour les terminaux de la classe B. Ce travail propose un modèle fondé sur les chaînes de Markov pour estimer le délai de livraison d'une trame confirmée avec retransmissions sous l'impact du nombre de canaux disponibles, du débit de données et du nombre de terminaux. Les résultats de l'étude suggèrent que les passerelles souffrent de l'incapacité d'utiliser la plupart des créneaux de ping de la classe B en raison de la limitation du rapport cyclique. Ainsi, sur la liaison descendante, le nombre de créneaux ping a un impact sur le délai de communication mais un effet mineur sur le débit.

7.3.2 Études fondées sur l'expérimentation

Les auteurs dans [100] ont effectué une étude empirique de l'impact de la présence du trafic descendant sur la performance de la liaison montante pour les réseaux **LORAWAN** fonctionnant dans la bande 868 MHz. Les résultats montrent que, dans la pratique, les transmissions en liaison descendante peuvent compromettre les performances de la liaison montante et doivent être prises en compte lors du déploiement de ces réseaux. Les auteurs ont également démontré l'effet de la sélection des canaux de fréquence et des **SF** sur l'amélioration des performances d'un réseau **LORAWAN**. De plus, dans [119], les auteurs se sont penchés sur la question de l'orthogonalité entre les liaisons montantes et descendantes. Pour répondre à cette question, les auteurs ont considéré un banc de

test réel combinant des USRPs et des GNU Radio et ont observé une non-orthogonalité. D'après les résultats obtenus, cela a un impact sur le taux de délivrance en le réduisant de 20%, dans les expérimentations menées, lorsque des transmissions simultanées se produisent dans les deux sens.

Malgré la pertinence des travaux recensés, il convient de noter qu'ils se concentrent uniquement sur l'analyse de l'impact des communications en liaison descendante sur les performances des réseaux LoRaWAN et en particulier sur les messages d'accusés de réception des trames montantes, avec des approches de validation fondées sur la simulation et/ou des modèles analytiques [55], [99], [128] et [56]. Rares sont les travaux présentant une étude expérimentale de la liaison descendante [100] [119]. Ces études souffrent du manque d'un cadre expérimental reproductible, facile à mettre en place et comportant un nombre significatif de terminaux. Il convient de noter aussi que ces analyses portent sur l'impact des messages d'accusés de réception des trames sur la voie montante. Elles négligent le trafic des trames de contrôle contenant des commandes MAC et/ou des données applicatives. Or ce trafic peut rapidement constituer une charge de trafic significative dans le cadre de réseaux de grande taille nécessitant une reconfiguration fréquente ou ayant des contraintes applicatives fortes conduisant à un échange fréquent de trames de données applicatives avec les terminaux.

De plus, pour plusieurs approches d'allocation de paramètres de transmission [37], [39], [38], [52], [70], [94], [108] (voire le chapitre 3 pour plus de détails), l'évaluation est limitée et fondée sur des simulations ou des modèles mathématiques sans démontrer la faisabilité de ces stratégies dans un environnement réel en configurant les terminaux à l'aide de commandes MAC en liaison descendante.

Ce manque d'outils et de solutions constitue notre motivation pour fournir un *framework* qui prend en charge les messages envoyés sur la liaison descendante pour :

- les trames de contrôle contenant des commandes MAC de configuration permettant, principalement, de tester les stratégies d'allocation de paramètres de transmission et la surveillance de l'état des terminaux ;
- les trames de données applicatives.

Ce *framework* apporte une solution pratique pour construire des bancs de test à large échelle, facile à entretenir et à reproduire dans le but d'effectuer des analyses concrètes des performances des réseaux LoRaWAN.

7.4 Notre positionnement

Dans cette troisième partie du manuscrit, nous présentons **LORADL**, le premier *framework* conçu pour prendre en charge les communications sur la liaison descendante dans les réseaux **LORAWAN**. Notre solution vise la satisfaction des objectifs suivants :

- conception et implantation d'une solution pratique pour prendre en charge l'envoi de trames de configuration contenant des commandes **MAC** et des trames de données d'applications sur la liaison descendante ;
- mise en place d'un banc de test combinant **LORADL** et Chirpstack ;
- étude de la performance de la liaison descendante.

LoRADL : UN FRAMEWORK POUR PRENDRE EN CHARGE LES COMMUNICATIONS SUR LA LIAISON DESCENDANTE LoRAWAN

L'objectif de ce travail est de proposer une solution pratique et flexible permettant à un serveur de réseau **LoRAWAN** d'envoyer des commandes **MAC** afin de gérer les terminaux. En complément, notre conception permet également aux applications IoT de transmettre des trames sur la liaison descendante jusqu'alors réservée aux commandes de configuration **MAC**, facilite la mise en œuvre de nouvelles stratégies d'affectation de facteurs d'étalement, et fournit un cadre expérimental pour l'analyse des performances de la liaison descendante.

8.1 Conception de LoRaDL

Dans cette section, nous détaillons la conception d'un framework dédié aux liaisons descendantes dans les réseaux **LoRAWAN** appelé **LoRADL** (LoRa DownLink). La conception de cette solution vise à permettre aux utilisateurs d'envoyer des commandes **MAC** à partir du serveur de réseau (NS) pour la reconfiguration des terminaux ou d'envoyer des trames de données sur la liaison descendante à partir du serveur d'application. **LoRADL** peut être représenté selon la figure 8.1.

Tout d'abord, afin de mettre à jour les paramètres de transmission, à savoir le facteur d'étalement, la puissance et le canal de transmission, nous avons proposé un ensemble de mécanismes pour étendre le serveur de réseau ChirpStack permettant l'envoi de commandes **MAC** sur la liaison descendante. Comme nous l'avons indiqué, **LoRAWAN**

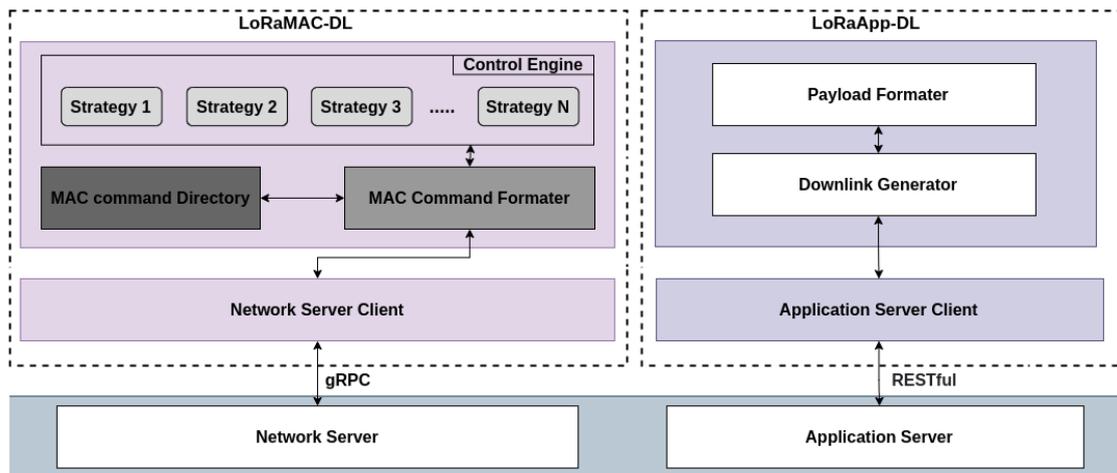


FIGURE 8.1 Architecture du framework LoRaDL

fournit deux méthodes pour envoyer des commandes **MAC** sur la liaison descendante. Dans ce travail, nous avons choisi d’exploiter le champ **Fopts** présenté dans le chapitre précédent pour intégrer les commandes **MAC** permettant la configuration des terminaux.

Deuxièmement, notre solution permet de mettre à jour les paramètres des applications. En effet, dans les réseaux **LoRaWAN** et pour certains scénarios, les applications sont susceptibles d’envoyer des données aux terminaux sur la liaison descendante afin de mettre à jour ou de configurer certains paramètres de l’application. Ainsi, **LoRaDL** permet d’envoyer des trames de contrôle contenant des configurations applicatives intégrées dans le champ **FRMPayload**.

Comme illustré dans la figure 8.1, **LoRaDL** est composé de deux modules principaux, **LoRaMAC-DL** et **LoRaAPP-DL**. Nous allons en détailler les différents composants de chacun de ces modules.

8.1.1 Control Engine

Le composant *Control Engine* (CE) représente un répertoire qui contient les stratégies d’allocation des paramètres de transmission. Plus précisément, ce module est conçu pour permettre aux utilisateurs d’implanter de nouvelles stratégies pour l’allocation des paramètres de transmission, d’une manière reproductible et commune. En effet, comme décrit dans la fonction Stratégie i de l’algorithme 6, chaque stratégie ajoutée suit un format standard divisé en trois parties : 1) l’entrée N correspond à l’ensemble des terminaux du réseau, 2) l’algorithme ou la stratégie d’allocation des paramètres de transmission qui permet à l’utilisateur de développer ses propres algorithmes et enfin, 3)

une sortie standard qui consiste en un tuple des paramètres de transmission, à savoir **SF**, **TP** et **CH**.

Algorithm 6 Strategie i

Entrées : N : Terminaux

Sorties : Terminaux avec SF, TP et CH alloués.

- 1: Stratégie d'allocation des paramètres de transmission
 - 2: Retourner SF, TP, CH alloués
-

Le composant CE fournit donc une approche modulaire, rapide et facile pour ajouter et tester de nouvelles stratégies d'allocation de paramètres de transmission pour les terminaux du réseau.

8.1.2 MAC Command Directory

Le composant *MAC command directory* (**MACD**) liste les différentes commandes **MAC** envoyées par le **NS** aux terminaux tels que définies dans la spécification **LORAWAN**. Nous préconisons de dédier un composant indépendant pour lister les commandes **MAC** afin d'anticiper les éventuels changements apportés par les nouvelles versions de la spécification **LORAWAN**.

De plus, le module **MACD** utilise les différentes structures de commandes **MAC** définies dans la spécification **LORAWAN** pour les interpréter de manière flexible et extensible.

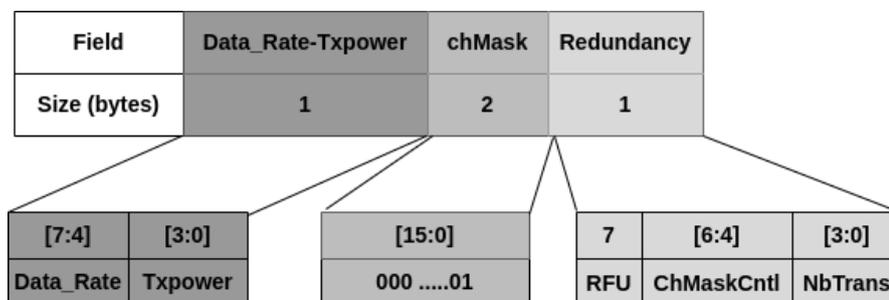


FIGURE 8.2 Structure de la commande MAC LinkADRReq

À titre d'exemple, considérons la structure de la commande LinkADRReq illustrée dans la figure 8.2 conformément à la spécification **LORAWAN**. LinkADRReq est une commande **MAC** de 4 octets transmise par le **NS** pour demander au terminal de mettre à jour ses paramètres de transmission. Elle se compose principalement de : (i) Débit de données (DR) et puissance de transmission (TP) qui sont des valeurs spécifiques de la région géographique ; (ii) ChMask signifie masque de canal et définit les canaux à utiliser

pour les prochaines transmissions sur la liaison montante; *(iii)* NbReq indique enfin le nombre de transmissions pour tous les messages de liaison montante non confirmés.

Cette commande **MAC** est formatée à l'aide du composant **MACD** en interprétant sa structure en un schéma de données comprenant tous les champs nécessaires. Un exemple de l'interprétation de la commande LinkADRReq est donné comme suit :

```
1 {
2   "LinkADRReq" :{
3     "cid" :2 ,
4     "dr"  :0 ,
5     "tx_power" :5 ,
6     "chMask" : "FFFF" ,
7     "redundacy" :{
8       "RFU" :1 ,
9       "chMaskCntl" : " " ,
10      "NbTrans" :2
11    }
12  }
13 }
```

8.1.3 MAC Command Formatter

Le composant *MAC Command Formatter* (**MACF**) combine les paramètres de configuration produits par une ou plusieurs stratégies d'allocation de paramètres de transmission engendrés par le CE et la structure de commande **MAC** correspondante récupérée du composant **MACD** pour construire une commande **MAC** formatée. Plus précisément, ce composant remplit les champs de la commande **MAC** en fonction du schéma de données qui la décrit.

8.1.4 Network Server Client

Le composant *Network Server Client* (**NSC**) représente l'interface de connexion avec l'API gRPC du serveur de réseau (**NS**). En pratique, après que le **MACF** a formaté la commande **MAC**, le client l'envoie au serveur de réseau pour qu'elle soit placée dans la file d'attente des messages de la liaison descendante correspondant à un terminal donné. Dès qu'une trame est reçue sur la liaison montante, le **NS** envoie une trame sur la liaison descendante avec des commandes **MAC** dans le champ Fopts.

8.1.5 Payload Formatter

Le composant *Payload Formatter* (PF) traite les données utiles de l'application qui sont destinées à être envoyées sur la liaison descendante vers un terminal. Cette charge utile est ensuite incorporée dans le champ FRMPayload.

8.1.6 Downlink Generator

Le composant *Downlink Generator* (DG) crée des trames pour la liaison descendante destinées à un terminal donné en intégrant les données utiles formatées dans le champ FRMPayload et en choisissant un port supérieur à 0.

8.1.7 Application Server Client

Le composant *Application Server Client* (ASC) représente l'interface de connexion avec l'API RESTful du serveur d'application (AS). Plus précisément, après avoir formaté et créé une trame pour la liaison descendante en utilisant les deux modules PF et DG, le client l'envoie à l'AS pour qu'elle soit mise dans la file d'attente de liaison descendante correspondant à un terminal donné. Dès qu'une trame est reçue sur la liaison montante, l'AS envoie la trame sur la liaison descendante.

8.2 Implantation de LoRaDL

Dans cette section, nous détaillons la mise en œuvre de la solution que nous proposons permettant d'envoyer des trames sur la liaison descendante comportant des commandes MAC et/ou la charge utile des applications. Cette solution vient pour étendre le serveur de réseau et le serveur d'application de la stack ChirpStack dans le but de faciliter la mise en place d'un outil permettant de créer et d'envoyer des trames de contrôle en liaison descendante aux terminaux.

Notre objectif est, par la suite, de construire une plateforme expérimentale extensible afin d'étudier les performances de ces envois de trames en liaison descendante dans des déploiements réels.

8.2.1 LoRaMAC-DL

LoRaMAC-DL est un module logiciel qui s'interface avec le serveur réseau de ChirpStack pour fournir aux utilisateurs un moyen facile, rapide et reproductible d'expérimenter l'envoi de commandes MAC personnalisées et contrôlées sur la liaison descendante pour permettre :

- la reconfiguration des terminaux avec de nouveaux paramètres de transmission à l'aide de la commande `MAC LinkADRReq` ;
- le monitoring des terminaux à l'aide de la commande `DevStatusReq` ;
- la configuration de la durée des fenêtres de réception des terminaux.

Le module `LoRaMACDL` est constitué de plusieurs composants implantés en python. La description des commandes `MAC` dans le composant `MACD` est donnée sous la forme d'un document JSON pour permettre une lecture intuitive fondée sur la structure des commandes `MAC` définie dans la spécification `LoRaWAN`. En ayant une représentation des commandes `MAC` fournies par le composant `MACD`, le composant `MACF`, un script python vient formater ces commandes `MAC` en indiquant les paramètres de transmission issus d'une stratégie du composant `CE` dans les champs correspondants en fonction de la commande à envoyer aux terminaux.

Le composant python `NSC` représente l'interface de connexion entre le `ChirpStack NS` et le module `LoRaMACDL` au travers de communications `gRPC` afin de créer en premier lieu une file d'attente de trames de liaison descendante comprenant des commandes `MAC` pour un terminal donné, puis d'ajouter ces trames à cette file. Lors de la réception d'une trame sur la liaison montante d'un terminal, si sa file d'attente n'est pas vide, une trame lui sera envoyée en liaison descendante. Enfin, à la réception au niveau du terminal, les commandes `MAC` imbriquées dans le champ `Fopts` sont exécutées.

Une trame de configuration envoyée sur la liaison descendante a pour but de mettre à jour les paramètres de transmission du dispositif d'extrémité. Par exemple, une `LinkADRReq` envoyée par le `ChirpStack NS` et engendrée par le module `LoRaMACDL` permet d'indiquer à un terminal donné, d'envoyer ses prochaines trames en utilisant un *Data Rate (DR)* de 5 qui correspond à *SF* 7, une puissance de transmission égale à 5 *dBm* et un canal d'indice 1 qui correspond au canal 868.3.

8.2.2 LoRaAppDL

Le module `LoRaAppDL` est composé de trois éléments implantés en python. Le composant `PF` traite les données de l'application et les structure pour les intégrer dans le champ `FRMPayload`. Cette charge utile formatée est ensuite utilisée pour créer une trame en liaison descendante par le composant `DG` pour un terminal donné. Le composant python `ASC` représente l'interface de connexion entre le serveur d'application `ChirpStack` et le module `LoRaAppDL` en utilisant des communications `RESTful`. Il a le rôle de créer une file d'attente de trames contenant des données applicatives destinées à être envoyées en liaison descendante. Lorsqu'une trame montante est reçue d'un terminal, les trames dans la file lui seront envoyées.

La première version V1.0 du *framework* [LORADL](#) est disponible librement sur GitHub [60] et peut être consultée.

8.3 Évaluation des performances

Nous avons considéré un déploiement de ChirpStack et une instance LoRaMAC connectée au [NS](#) ChirpStack. Toutes nos expérimentations sont menées en utilisant des terminaux PYCOM LOPY4 ou des terminaux MICROCHIP RN2483 greffés sur un RASPBERRY PI3. Tous les nœuds sont alimentés par une source d'alimentation externe, comme le montrent les figures 8.3a-8.3c.

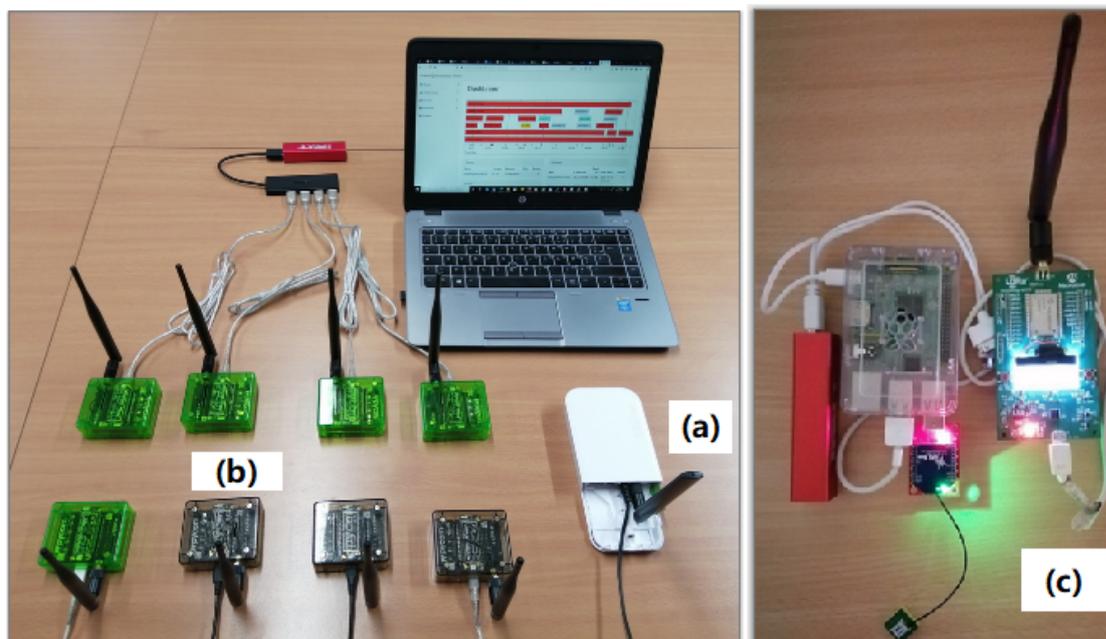


FIGURE 8.3 Configuration expérimentale : (a) Passerelle Mikrotik wAP LR8 LoRaWAN, (b) Terminal Pycom Lopy4, (c) Terminal Microchip RN2483.

Pour démontrer la faisabilité et l'utilité de notre framework et mettre en évidence certains cas d'utilisation potentiels pour l'étude des performances de la liaison descendante [LORAWAN](#), nous avons considéré trois scénarios résumés dans le Tableau 8.1.

8.3.1 Scénario 1 : preuve de concept

Initialement, nous avons considéré un déploiement de deux passerelles dans un environnement intérieur espacées de 12 mètres et séparées par 4 bureaux. Nous avons

scénario	Terminaux	Passerelles	DL	P_DL	P_UL
1	1	2	20	60	5
2	3	2	40	60	5
3	11	2	/	/	5
			30	60	5
			/	/	10
			30	60	10

TABLE 8.1 Résumé des scénarios expérimentaux

mené cette expérimentation avec un seul terminal Microchip RN2483 positionné à côté de la passerelle 1 et configuré pour envoyer des trames en liaison montante toutes les $P_{UL} = 5$ secondes.

Du point de vue du serveur de réseau, nous avons considéré une séquence prédéfinie de 20 configurations de paramètres de transmission comprenant un facteur d'étalement, un canal et une puissance de transmission comme stratégie d'allocation de ressources. Il est à noter que cette stratégie suit une politique incrémentale, où, soit le SF, soit la TP sont incrémentés pour chaque nouvelle configuration et un seul canal est activé parmi les canaux de la liaison montante LORAWAN par défaut (868.1, 868.3, 868.5). Les configurations sont envoyées à l'appareil final sur la liaison descendante, chaque $P_{DL} = 60$ secondes en utilisant le module LoRaMACDL. L'objectif de ce scénario est de démontrer la faisabilité et l'utilité pratique de notre proposition pour la reconfiguration des terminaux via des commandes MAC en liaison descendante. Nous avons reporté dans les Fig. 8.4a et Fig. 8.4b, la répartition des trames par SF et par type de message (liaison montante/liaison descendante) et la répartition des trames de liaison montante par canal respectivement. Le terminal est initialisé avec $SF = 7$. Ces figures illustrent le changement de SF et de canal du terminal lors de la réception d'une commande MAC LinkADRReq en liaison descendante en adoptant le SF et le canal inclus dans les trames.

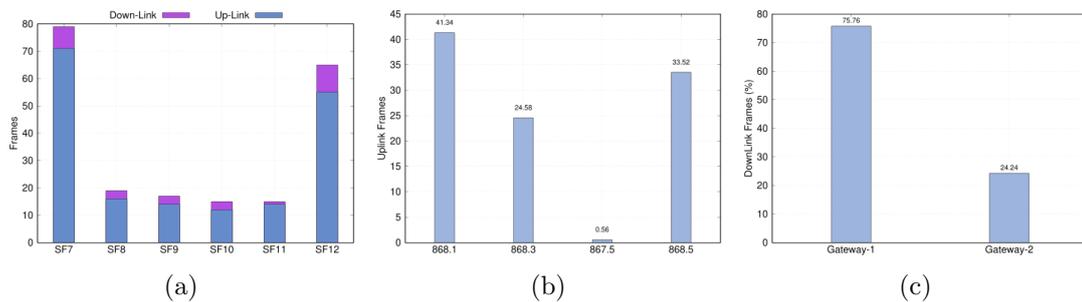


FIGURE 8.4 (a) Trames par SF et type de message, (b) Trames de liaison montante par canal, et (c) Trames de liaison descendante par passerelle.

8.3.2 Scénario 2 : limites de la liaison descendante du terminal

Dans ce scénario, nous avons considéré un déploiement avec deux passerelles Mikrotik en intérieur et trois terminaux Microchip RN2483 (ED1, ED2, ED3). Deux d'entre eux sont déployés à l'intérieur et restent statiques. L'objectif de ces deux terminaux est de charger le réseau. Le dernier terminal est mobile. L'intérêt de considérer un terminal mobile est d'atténuer les valeurs de RSSI et de SNR lorsqu'il s'éloigne des passerelles. Cela permettra d'étudier et de montrer, d'une part, la faible sensibilité des terminaux lors de la réception de trames en liaison descendante par rapport aux passerelles lors de la réception de trames en liaison montante. On observera également la manière avec laquelle le serveur de réseau sélectionne une passerelle pour transmettre sur la liaison descendante vers le terminal.

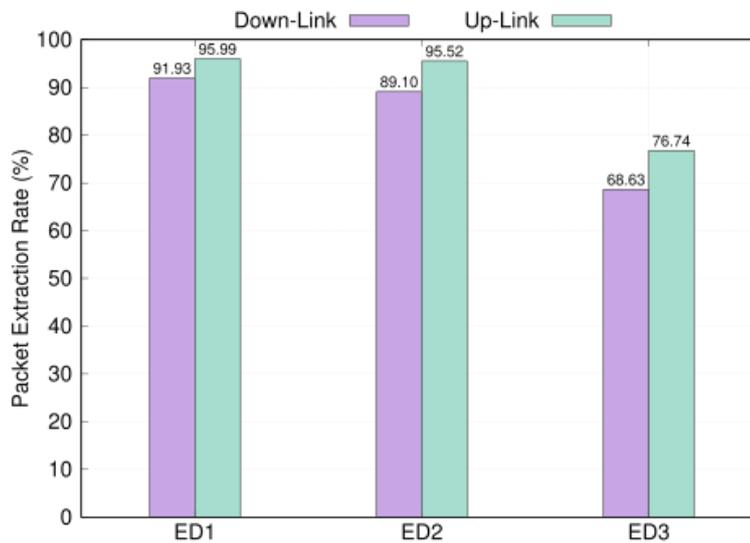


FIGURE 8.5 Taux d'extraction de paquets

Pour montrer la faible sensibilité des terminaux par rapport à celle des passerelles, nous avons reporté dans la figure 8.5 le taux d'extraction de paquets (PER) à la fois sur les liaisons montantes et descendantes et pour tous les terminaux. La fig. 8.5 montre que l'écart entre le PER de la liaison descendante et celui de la liaison montante augmente en s'éloignant de la passerelle. Ceci est clairement visible dans le PER enregistré pour le terminal mobile ED3.

Dans les conditions expérimentales du scénario 2, ED1 est positionné près des passerelles, ED2 un peu plus loin et ED3 en mouvement suivant une trajectoire qui s'éloigne des passerelles. Il ne fait aucun doute que le fait de s'éloigner des passerelles atténue les

valeurs de **RSSI** et de **SNR** pour les trames reçues en liaison montante et descendante, ce qui entraîne une différence entre les PER en liaison montante et descendante de $76,74\% - 68,63\% = 8,11\%$ pour ED3 et beaucoup moins de $95,99\% - 91,93\% = 4,05\%$ pour ED1.

8.3.3 Scénario 3 : Impact de la charge de trafic

Enfin, dans ce scénario, nous étudions l’impact de la charge de trafic sur la liaison descendante. À ce titre, nous avons déployé deux passerelles en intérieur et 11 terminaux situés autour de ces deux passerelles. Plus précisément, nous avons déployé 4 Microchip RN2483 et 7 Pycom Lopy4.

Pour analyser l’impact de la charge de trafic, nous considérons 4 cas, résumés dans le tableau 8.1 et détaillés comme suit :

- cas N°1 (liaisons montantes) : nous considérons deux expérimentations, toutes deux avec 11 terminaux envoyant uniquement sur les liaisons montantes avec une fréquence d’un message par $P_{UL} = 5$ secondes et un message par $P_{UL} = 10$ secondes par terminal respectivement ;
- cas N°2 (liaisons montantes et descendantes) : nous considérons également deux expérimentations, toutes deux avec 11 terminaux. Tous les terminaux envoient des trames en liaison montante chaque $P_{UL} = 5$ secondes puis chaque $P_{UL} = 10$ secondes. De plus, le serveur de réseau (**NS**) envoie un total de 30 messages en liaison descendante contenant des commandes **MAC** chaque $P_{DL} = 60$ secondes aux quatre terminaux Microchip RN2483.

P_UL	Cas N°1 : PER (moyenne)	Cas N°2 : PER (moyenne)
5	99.14 %	96.06 %
10	99.5 %	98.32 %

TABLE 8.2 Taux moyen d’extraction de paquets

Initialement, avant d’envoyer les commandes **MAC** sur la liaison descendante, tous les terminaux Microchip RN2483 sont initialisés pour utiliser $SF = 7$, et les sept terminaux Pycom Lopy4 sont configurés pour utiliser un **SF** fixe correspondant respectivement à $SF = \{7, 7, 8, 9, 10, 11, 12\}$. Seuls les paramètres de transmission des terminaux Microchip RN2483 sont mis à jour au travers de commandes **MAC** envoyées en liaison descendante par le **NS**. Cette stratégie d’allocation permet de garder la trace du nombre de terminaux utilisant un **SF** donné pour calculer la charge de trafic globale engendrée.

On considère que ρ_i , donnée dans l’équation 8.1, est la charge de trafic [66] engendrée

par les terminaux utilisant SF_i . T_i et M_i sont respectivement le temps de vol et le nombre de terminaux utilisant SF_i , et P_{UL} est la période de d'envoi des trames.

$$\rho_i = \frac{M_i \times T_i}{P_{UL}} \quad (8.1)$$

La charge globale du trafic envoyée par les terminaux dans le réseau est donnée par l'équation 8.2.

$$\rho = \sum_{i=7}^{12} \rho_i \quad (8.2)$$

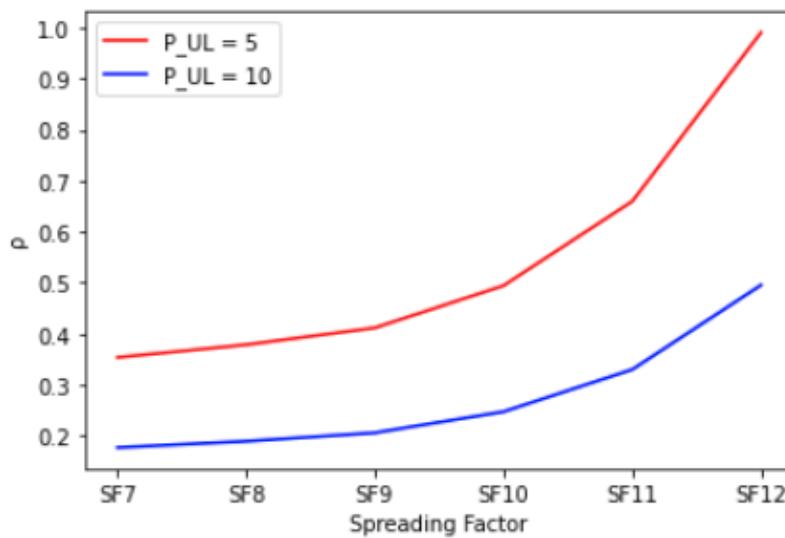


FIGURE 8.6 Charge de trafic ρ en fonction de la configuration SF reçue en liaison descendante.

La Figure 8.6 illustre l'évolution de la charge de trafic globale en fonction des SF envoyés par le NS en liaison descendante afin de mettre à jour le débit de données des terminaux.

La figure 8.7 illustre la différence entre le PER de la liaison montante au niveau de la passerelle pour les deux scénarios, liaison montante uniquement et (liaison montante + liaison descendante). La figure montre une différence significative entre les taux de réception pour les terminaux RN2483 qui sont configurés pour recevoir des trames en liaison descendante. En effet, l'ajout de trafic descendant dans le réseau a un impact direct sur le PER de la liaison montante au niveau de la passerelle. Ces résultats sont confirmés par les moyennes des taux d'extraction de paquets présentées dans le tableau 8.2, dans lequel cette différence est clairement visible. Ainsi, la coexistence des deux

trafics dans le réseau a un impact direct sur le taux d'extraction des données au niveau de la passerelle.

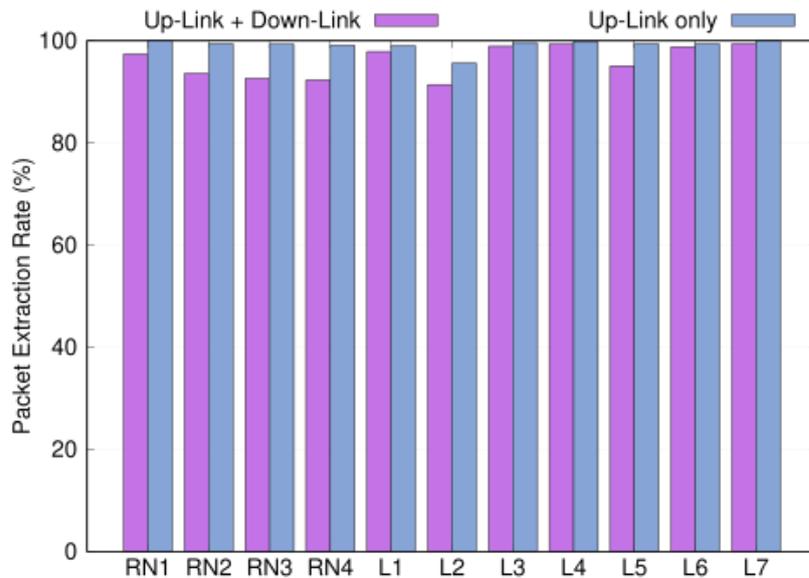


FIGURE 8.7 PER : Cas N°1 vs Cas N°2.

En revanche, dans la figure. 8.8, qui représente le taux d'extraction de données de la liaison montante au niveau de la passerelle par rapport aux taux d'extraction de données de la liaison descendante au niveau des nœuds, il est intéressant de noter que ces taux sont nettement meilleurs sur la liaison montante que sur la liaison descendante. Cela suggère que la performance de la liaison descendante souffre de la charge de trafic sur la liaison montante.

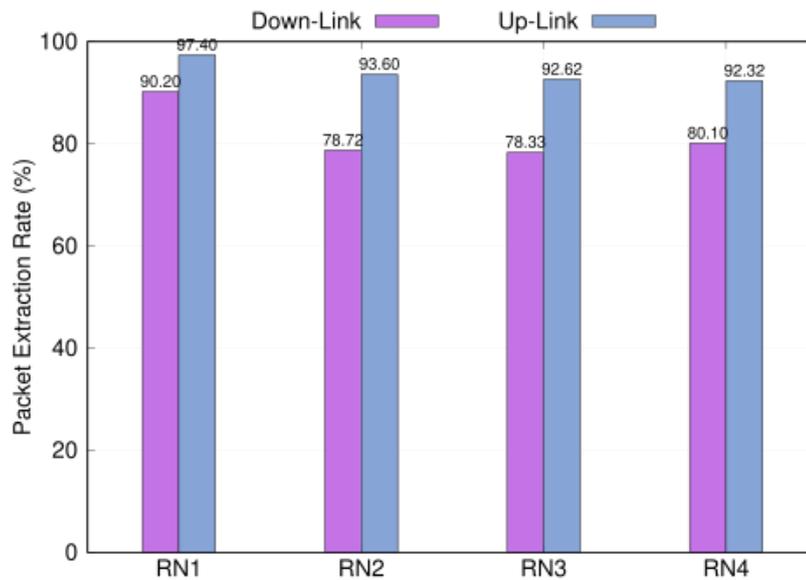


FIGURE 8.8 PER : liaisons montantes vs liaison descendantes

8.4 Conclusion

Dans ce chapitre, nous avons proposé et validé un framework pratique appelé **LORADL** qui permet aux opérateurs et aux utilisateurs **LORAWAN** d'expérimenter des stratégies de paramétrage de terminaux dans un déploiement réel et d'avoir le plein contrôle sur l'envoi des liaisons descendantes. Plus précisément, ce framework fournit une prise en charge de l'envoi de commandes de configuration **MAC** et de données applicatives sur la liaison descendante.

De plus, nous avons démontré la faisabilité et l'utilité de disposer d'un tel support à travers une analyse expérimentale poussée en testant plusieurs scénarios de configuration. Notre analyse a permis d'identifier un ensemble de limitations de la liaison descendante par rapport à la liaison montante en raison de la sensibilité de réception des terminaux qui est plus faible que celle des passerelles.

Enfin, nous avons étudié l'impact de la charge de trafic sur les performances de la liaison descendante et l'impact de la coexistence de ce trafic et le trafic de la liaison montante sur les performances du réseau **LORAWAN**.

CONCLUSIONS ET PERSPECTIVES

Conclusions

Les réseaux **LORAWAN** ont suscité beaucoup d'intérêt dans la communauté des communications sans fil en raison de leur grand potentiel en termes de haute densité de terminaux autour d'une seule passerelle, d'efficacité énergétique, de la faiblesse de leur coût, de leurs capacités, en particulier à longue portée, et de la facilité et de la flexibilité de leur déploiement. Toutes ces caractéristiques les rendent particulièrement attractifs et compétitifs pour une grande variété d'applications **IOT**.

Forts de ces caractéristiques prometteuses, des travaux de recherche de plus en plus nombreux sont menés pour en améliorer les performances ou pour proposer de nouveaux mécanismes. Cependant, certains axes de recherche restent peu explorés et n'ont pas donné lieu à des expérimentations suffisamment approfondies.

Dans cette thèse, nous proposons d'apporter des solutions à des problèmes en pleine effervescence au sein de la communauté **IOT** portant sur l'allocation des paramètres radio, à des thématiques en émergence portant sur l'itinérance dans les réseaux **LORAWAN** et à des orientations peu explorées proposant d'étudier les performances de la liaison descendante. En adoptant une méthodologie fortement motivée par l'évaluation multi-approche, combinant l'expérimentation, la simulation et l'émulation, nous avons implanté un ensemble d'outils et mis en place plusieurs bancs de test.

Nous avons donc proposé **L3SFA** et **L3SFA-TPC**, deux stratégies d'allocation de paramètres de transmission dans les réseaux **LORAWAN**. Ces stratégies s'inspirent des mécanismes d'équilibrage de charge en combinant les stratégies fondées sur les paramètres physiques (**SNR** et **RSSI**) et sur la charge dans le réseau pour choisir les paramètres. Les résultats de l'évaluation des performances de **L3SFA** dans plusieurs scénarios montrent que l'équilibrage de la charge conduit à de meilleures performances en termes de taux de délivrance tout en garantissant le passage à l'échelle pour des réseaux de grande taille (denses). L'extension de **L3SFA** avec un mécanisme de contrôle de la puissance de

transmission permet de réduire de surcroît la consommation d'énergie des terminaux.

Ensuite, nous nous sommes intéressés à l'itinérance dans les réseaux **LoRaWAN**. Nous avons proposé **LoRaRoam**, la première solution implantée prenant en charge l'itinérance active. Après avoir identifié les défis liés à la mise en place d'une interconnexion entre opérateurs **LoRaWAN**, nous avons apporté des solutions architecturales et protocolaires, fourni de nouveaux mécanismes et mis en place un banc de test expérimental. Pour remédier au manque de matériel pour élaborer des expérimentations à large échelle, nous avons conçu **LRE**, le premier émulateur dédié à l'analyse de performance de l'itinérance. Les résultats de l'évaluation des performances menée avec notre banc de test montrent l'efficacité de notre solution d'itinérance en termes de latence d'activation pour la première visite d'un terminal itinérant. En particulier, les contraintes de délai pour les terminaux de classe A sont respectées. Les résultats de l'émulation de l'itinérance ont abouti aux mêmes conclusions en termes de délais, mais ont aussi montré que la solution ne surcharge ni les **CPU** ni les mémoires des serveurs hébergeant les différents agents logiciels définis.

Enfin, nous nous sommes intéressés aux communications sur la liaison descendante dans les réseaux **LoRaWAN**. Pour aborder le sujet par la pratique et l'expérimentation, nous avons conçu **LoRaDL**, le premier *framework* prenant en charge les trames de contrôle imbriquant des commandes **MAC** et les trames de données applicatives envoyées sur la liaison descendante. Les résultats de nos expérimentations ont montré la faisabilité de notre proposition. Ils ont confirmé la faible sensibilité des terminaux comparée à celle des passerelles et ont mis en exergue l'impact du trafic descendant sur les performances des réseaux **LoRaWAN**.

Perspectives

Plusieurs perspectives sont envisageables dans la suite de nos travaux pour enrichir les résultats obtenus, apporter de nouveaux mécanismes et associer les outils développés (simulateur, émulateur, **LoRaDL**) pour produire un cadre expérimental complet. Nous présentons dans cette partie, les perspectives les plus prometteuses et l'utilisation potentielle de nos solutions (**L3SFA/L3SFA-TPC**, **LoRaRoam** et **LoRaDL**).

1) **L3SFA/L3SFA-TPC : évaluation expérimentale.**

Dans le chapitre 4, nous avons mené une évaluation de **L3SFA** et **L3SFA-TPC** par simulation en mettant en place plusieurs scénarios et en utilisant différentes approches d'évaluation fondées à la fois sur une répartition uniforme des terminaux et sur un jeu de données contenant des traces de trames **LoRaWAN**. Même si cette évaluation présente

une diversité d'approches, il serait intéressant d'étendre notre étude à un environnement réel pour étudier comment nos mécanismes se comportent dans une infrastructure réelle. En particulier, l'impact sur les communications en liaison descendante pour la configuration des terminaux et la performance de la stratégie pourra être testé. Cela sera possible en implantant les deux stratégies dans le *framework* **LoRADL** présenté dans le chapitre 8 pour transmettre les attributions de paramètres de transmission aux terminaux en liaison descendante. L'utilisation de réseaux multi-passerelles nous paraît également très intéressante pour obtenir de meilleurs résultats et augmenter la capacité du réseau. La diversité spatiale que cette solution procure est en effet un gage d'amélioration substantielle. En revanche, le choix des paramètres de transmission devient plus compliqué si l'on veut justement en tirer profit.

2) LoRaRoam : vers une solution complète et un déploiement à large échelle

Dans le chapitre 6, nous avons présenté **LoRAROAM**, la première solution implantée et déployée prenant en charge l'itinérance active (*Handover*) dans les réseaux **LoRAWAN**. Nous nous dirigeons vers l'extension de notre solution pour fournir une solution complète prête à être déployée pour interconnecter plusieurs opérateurs en prenant en compte la sécurité et la tolérance aux pannes. Nous envisageons d'ajouter des mécanismes pour sécuriser l'échange entre les serveurs de réseau lors de la migration du contexte des terminaux. Pour accentuer la capacité de notre solution à fournir une grande tolérance aux pannes, le composant **MA** peut être distribué et dupliqué pour assurer d'une part une disponibilité élevée du service **DNS** et de souscription des opérateurs et d'autre part pour implanter des solutions d'équilibrage de charge sur les instances de **MA**.

De plus, nous souhaitons lancer des invitations à des organismes académiques et industriels pour rejoindre le réseau d'itinérance pour effectuer des tests à large échelle. Nous avons d'ores et déjà entamé ce processus lors de la présentation de notre solution, **LoRaRoam**, devant le groupe de travail académique de la **LoRa Alliance**.

3) LoRaDL : vers un outil d'évaluation puissant

Dans le chapitre 8, nous avons présenté **LoRADL**, le premier *framework* permettant de prendre en charge la transmission des trames de contrôle contenant des commandes **MAC** et des trames de données applicatives sur la liaison descendante. La conception modulaire de cette solution nous permet d'intégrer les stratégies d'allocation de paramètres de transmission de la littérature, limitées à ce jour à une évaluation par simulation ou par modèle analytique, pour étudier leur performance dans un environnement réel. Cette initiative fournira un outil permettant l'évaluation et la comparaison de ces stratégies

expérimentalement.

4) Perspectives à long terme : l'union fait la force

Les solutions que nous avons présentées dans cette thèse traitent de problèmes en effervescence. Tout le long de cette thèse, nous avons apporté des contributions regroupant des algorithmes d'allocation de paramètres radio ([L3SFA](#), [L3SFA-TPC](#)), des mécanismes et protocoles d'itinérance, des outils de simulation et d'émulation (LoRa-L3SFA, LRE), un *framework* ([LoRADL](#)) pour prendre en charge la liaison descendante et la conception et déploiement de bancs de tests. Le choix et la conception de toutes ces solutions sont guidés par nos projections à long terme. Le but est de combiner toutes ces solutions pour construire une **pile LoRaWAN** complète sous forme d'une boîte à outils fournissant un cadre de recherche regroupant plusieurs techniques de validation et prenant en charge plusieurs axes de recherche. L'idée de **l'union fait la force** pour construire un outil de référence de type **pile LoRaWAN** nous est apparue dès le début de la thèse, lorsque nous avons constaté la difficulté de reproduire les résultats présentés dans la littérature, l'absence d'un cadre de test unifié permettant de comparer les résultats et la difficulté de reproduire nos expérimentations.

PUBLICATIONS

Conférences internationales

- Mohamed Hammache, Rahim Kacimi, and André-Luc Beylot. L3sfa: Load shifting strategy for spreading factor allocation in lorawan systems. In *2020 IEEE 45th Conference on Local Computer Networks (LCN)*, pages 216–224, 2020. doi:[10.1109/LCN48667.2020.9314777](https://doi.org/10.1109/LCN48667.2020.9314777)
- Mohamed Hammache, Rahim Kacimi, and André-Luc Beylot. Unifying lorawan networks by enabling the roaming capability. In *2021 IEEE 46th Conference on Local Computer Networks (LCN)*, pages 371–374, 2021. doi:[10.1109/LCN52139.2021.9524996](https://doi.org/10.1109/LCN52139.2021.9524996)
- Mohamed Hammache, Rahim Kacimi, and André-Luc Beylot. Fundamentals and Performance of the LoRaWAN Downlink: an Experimental Approach. In *IEEE Global Communications Conference (GLOBECOM 2022)*, Rio de Janeiro, Brazil, December 2022. IEEE, IEEE. URL: <https://hal.archives-ouvertes.fr/hal-03741970>

Journaux

- Mohamed Hammache, Rahim Kacimi, and André-Luc Beylot. Enabling an inter-operator roaming capability in lorawan networks. *Ad Hoc Networks*, 139:103025, 2023. doi:<https://doi.org/10.1016/j.adhoc.2022.103025>

Soumis

- Mohamed Hammache, Rahim Kacimi, and André-Luc Beylot. Joint load-balancing and power control strategy to maximize the data extraction rate of lorawan networks. *Computer Networks (révisions mineures)*, 2022

Conférences francophones

- Mohamed Hammache, Rahim Kacimi, and André-Luc Beylot. Optimisation de l'allocation des facteurs d'étalement dans les Réseaux LoRaWAN. In *6èmes Ren-*

- contres Francophones sur la Conception de Protocoles, l'Évaluation de Performance et l'Expérimentation des Réseaux de Communication (CORES 2021)*, La Rochelle, France, May 2021. URL: <https://hal.archives-ouvertes.fr/hal-03220095>
- Mohamed Hamnache, Rahim Kacimi, and André-Luc Beylot. Itinérance dans les réseaux LoRaWAN . In *7èmes Rencontres Francophones sur la Conception de Protocoles, l'Évaluation de Performance et l'Expérimentation des Réseaux de Communication (CORES 2022)*, Saint-Rémy-Lès-Chevreuse, France, May 2022. Conférence co-localisée avec Algotel 2022. URL: <https://hal.archives-ouvertes.fr/hal-03657148>

BIBLIOGRAPHIE

- [1] Adaptive data rate. URL : <https://www.thethingsindustries.com/docs/reference/adr/>.
- [2] Implementing adaptive data rate (ADR) | DEVELOPER PORTAL. URL : <https://lora-developers.semtech.com/documentation/tech-papers-and-guides/implementing-adaptive-data-rate-adr/implementing-adaptive-data-rate/>.
- [3] LoRa and LoRaWAN : Technical overview | DEVELOPER PORTAL. URL : <https://lora-developers.semtech.com/documentation/tech-papers-and-guides/lora-and-lorawan/>.
- [4] Lorawan device classes. URL : <https://www.thethingsnetwork.org/docs/lorawan/classes/>.
- [5] LoRaWAN® 1.0.4 specification package. URL : https://lora-alliance.org/resource_hub/lorawan-104-specification-package/.
- [6] LoRaWAN® specification v1.0.3. URL : https://lora-alliance.org/resource_hub/lorawan-specification-v1-0-3/.
- [7] LoRaWAN® specification v1.1. URL : https://lora-alliance.org/resource_hub/lorawan-specification-v1-1/.
- [8] 3GPP. Etsi tr 125 996 v10.0.0, 2011. URL : https://www.etsi.org/deliver/etsi_tr/125900_125999/125996/10.00.00_60/tr_125996v100000p.pdf.
- [9] K. Q. Abdelfadeel, D. Zorbas, V. Cionca, and D. Pesch. *free* —fine-grained scheduling for reliable and energy-efficient data collection in lorawan. *IEEE Internet of Things Journal*, 7(1) :669–683, 2020.
- [10] Ansuman Adhikary, Xingqin Lin, and Y.-P. Eric Wang. Performance evaluation of nb-iot coverage. In *2016 IEEE 84th Vehicular Technology Conference (VTC-Fall)*, pages 1–5, 2016. doi:10.1109/VTCFall.2016.7881160.
- [11] Michiel Aernouts, Rafael Berkvens, Koen Van Vlaenderen, and Maarten Weyn. Sigfox and lorawan datasets for fingerprint localization in large urban and rural areas. *Data*, pages 1–13, 2018. doi:10.3390/data3020013.

- [12] Fadi Al-Turjman. Mobile couriers' selection for the smart-grid in smart-cities' pervasive sensing. *Future Generation Computer Systems*, 82 :327–341, 2018. URL : <https://www.sciencedirect.com/science/article/pii/S0167739X17310737>, doi:<https://doi.org/10.1016/j.future.2017.09.033>.
- [13] Licia Amichi, Megumi Kaneko, Nancy El Rachkidy, and Alexandre Guitton. Spreading factor allocation strategy for lora networks under imperfect orthogonality. In *ICC 2019 - 2019 IEEE International Conference on Communications (ICC)*, pages 1–7, 2019. doi:[10.1109/ICC.2019.8761235](https://doi.org/10.1109/ICC.2019.8761235).
- [14] Muhammad Arsalan, Asghar Ashraf Musani, Syed Asad Ailia, Nabeel Baig, and Engr. M. Kashif Shaikh. Military uniform for health analytics for field intelligent zone (muhafiz) protecting the ones that protect our land. In *2018 2nd International Conference on Smart Sensors and Application (ICSSA)*, pages 64–68, 2018. doi:[10.1109/ICSSA.2018.8535755](https://doi.org/10.1109/ICSSA.2018.8535755).
- [15] Muhammad Asad Ullah, Junnaid Iqbal, Arliones Hoeller, Richard Demo Souza, Alves, and Hirley. K-means spreading factor allocation for large-scale lora networks. *Sensors (Basel, Switzerland)*, 19(21) :4723, Oct 2019. 31671700[pmid]. URL : <https://pubmed.ncbi.nlm.nih.gov/31671700>, doi:[10.3390/s19214723](https://doi.org/10.3390/s19214723).
- [16] Luigi Atzori, Antonio Iera, and Giacomo Morabito. The internet of things : A survey. *Computer Networks*, 54(15) :2787–2805, 2010. URL : <https://www.sciencedirect.com/science/article/pii/S1389128610001568>, doi:<https://doi.org/10.1016/j.comnet.2010.05.010>.
- [17] Eyuel D. Ayele, Chiel Hakkenberg, Jan Pieter Meijers, Kyle Zhang, Nirvana Meratnia, and Paul J. M. Havinga. Performance analysis of lora radio for an indoor iot applications. In *2017 International Conference on Internet of Things for the Global Community (IoTGC)*, pages 1–8, 2017. doi:[10.1109/IoTGC.2017.8008973](https://doi.org/10.1109/IoTGC.2017.8008973).
- [18] Wael Ayoub, Mohamad Mroue, Abed Ellatif Samhat, Fabienne Nouvel, and Jean-Christophe Prévotet. SCHC-Based Solution for Roaming in LoRaWAN. In *The 14-th International Conference on Broadband and Wireless Computing, Communication and Applications (BWCCA-2019)*, Anvers, Belgium, November 2019. URL : <https://hal.archives-ouvertes.fr/hal-02266618>.
- [19] Wael Ayoub, Fabienne Nouvel, Abed Ellatif Samhat, Mohamad Mroue, and Jean-Christophe Prévotet. Mobility management with session continuity during handover in lpwan. *IEEE Internet of Things Journal*, 7(8) :6686–6703, 2020. doi:[10.1109/JIOT.2020.2985925](https://doi.org/10.1109/JIOT.2020.2985925).
- [20] wael ayoub, Abed Ellatif Samhat, Fabienne Nouvel, Mohamad Mroue, Hassan Jradi, and Jean-Christophe Prévotet. Media independent solution for mobility management in heterogeneous LPWAN technologies. *Computer Networks*, 182 :107423, December 2020. URL : <https://hal.archives-ouvertes.fr/hal-02938593>, doi:[10.1016/j.comnet.2020.107423](https://doi.org/10.1016/j.comnet.2020.107423).
- [21] Sandoche Balakrichenan, Antoine Bernard, Michel Marot, and Benoit Ampeau. IoTRoam : design and implementation of an open LoRaWAN roaming architecture. In *IEEE Global Communications Conference (GLOBECOM)*, Madrid, Spain, December 2021. URL : <https://hal.archives-ouvertes.fr/hal-03100628>.

- [22] Dmitry Bankov, Evgeny Khorov, and Andrey Lyakhov. Mathematical model of lorawan channel access. In *2017 IEEE 18th International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, pages 1–3, 2017. doi:10.1109/WoWMoM.2017.7974300.
- [23] Dmitry Bankov, Evgeny Khorov, and Andrey Lyakhov. Mathematical model of lorawan channel access with capture effect. In *2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, pages 1–5, 2017. doi:10.1109/PIMRC.2017.8292748.
- [24] Steve Battle and Benedict Gaster. Lorawan bristol. In *The 21st International Database Engineering Applications Symposium, IDEAS '17*, page 287–290, New York, NY, USA, 2017. doi:10.1145/3105831.3105835.
- [25] Bruno Bellini and Alfredo Amaud. A 5 wireless platform for cattle heat detection. In *2017 IEEE 8th Latin American Symposium on Circuits Systems (LASCAS)*, pages 1–4, 2017. doi:10.1109/LASCAS.2017.7948089.
- [26] Mohamed Ben-Daya, Elkafi Hassini, and Zied Bahroun. Internet of things and supply chain management : a literature review. *International Journal of Production Research*, 57(15-16) :4719–4742, 2019. arXiv:https://doi.org/10.1080/00207543.2017.1402140, doi:10.1080/00207543.2017.1402140.
- [27] Martin Bor, Utz Roedig, Thiemo Voigt, and Juan Alonso. Do lora low-power wide-area networks scale? In *Proc. of the 19th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems, MSWIM'16*, page 59–67, 2016. doi:10.1145/2988287.2989163.
- [28] Takuya Boshita, Hidekazu Suzuki, and Yukimasa Matsumoto. Iot-based bus location system using lorawan. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pages 933–938, 2018. doi:10.1109/ITSC.2018.8569920.
- [29] M. Talha Buyukakkaslar, Mehmet Ali Erturk, Muhammed Ali Aydin, and Luca Vollero. Lorawan as an e-health communication technology. In *2017 IEEE 41st Annual Computer Software and Applications Conference (COMPSAC)*, volume 2, pages 310–313, 2017. doi:10.1109/COMPSAC.2017.162.
- [30] Qingsong Cai and Jia Lin. Improving the scalability of LoRa networks through dynamical parameter set selection. In Songtao Guo, Kai Liu, Chao Chen, and Hongyu Huang, editors, *Wireless Sensor Networks*, pages 3–18.
- [31] Christelle Caillouet, Martin Heusse, and Franck Rousseau. Optimal sf allocation in lorawan considering physical capture and imperfect orthogonality. In *2019 IEEE Global Communications Conference (GLOBECOM)*, pages 1–6, 2019. doi:10.1109/GLOBECOM38437.2019.9013602.
- [32] Agustin Candia, Soledad Natacha Represa, Daniela Giuliani, Miguel Ángel Luengo, Andrés Atilio Porta, and Luis Armando Marrone. Solutions for smartcities : proposal of a monitoring system of air quality based on a lorawan network with low-cost sensors. In *2018 Congreso Argentino de Ciencias de la Informática y Desarrollos de Investigación (CACIDI)*, pages 1–6, 2018. doi:10.1109/CACIDI.2018.8584183.

- [33] Dick Carrillo and Jorge Seki. Rural area deployment of internet of things connectivity : Lte and lorawan case study. In *2017 IEEE XXIV International Conference on Electronics, Electrical Engineering and Computing (INTERCON)*, pages 1–4, 2017. doi:10.1109/INTERCON.2017.8079711.
- [34] Matteo Cesana, Alessandro Redondi, and Jorge Ortin. A framework for planning lorawan networks. In *2018 IEEE 29th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, pages 1–7, 2018. doi:10.1109/PIMRC.2018.8580875.
- [35] ChirpStack. Chirpstack : open-source lorawan network server stack. URL : <https://www.chirpstack.io/>.
- [36] Daniele Croce, Michele Gucciardo, Stefano Mangione, Giuseppe Santaromita, and Ilenia Tinnirello. Impact of lora imperfect orthogonality : Analysis of link-level performance. *IEEE Communications Letters*, 22(4) :796–799, 2018. doi:10.1109/LCOMM.2018.2797057.
- [37] Francesca Cuomo, Manuel Campo, Alberto Caponi, Giuseppe Bianchi, Giampaolo Rossini, and Patrizio Pisani. Explora : Extending the performance of lora by suitable spreading factor allocations. In *2017 IEEE 13th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, pages 1–8, 2017. doi:10.1109/WiMOB.2017.8115779.
- [38] Francesca Cuomo, Domenico Garlisi, Alessio Martino, and Antonio Martino. Predicting lorawan behavior : How machine learning can help. *Computers*, 9(3), 2020. URL : <https://www.mdpi.com/2073-431X/9/3/60>, doi:10.3390/computers9030060.
- [39] Francesca Cuomo, Julio César Carrasquel Gámez, Antonio Maurizio, Laura Scipione, Manuel Campo, Alberto Caponi, Giuseppe Bianchi, Giampaolo Rossini, and Patrizio Pisani. Towards traffic-oriented spreading factor allocations in lorawan systems. In *2018 17th Annual Mediterranean Ad Hoc Networking Workshop (Med-Hoc-Net)*, pages 1–8, 2018. doi:10.23919/MedHocNet.2018.8407091.
- [40] Matteo Danieletto, Li Li, and Joel T. Dudley. Application of i-como device towards geographic disease enrichment pattern revealed from electronic medical record at a large urban academic medical center. In *Proceedings of the 11th EAI International Conference on Pervasive Computing Technologies for Healthcare, PervasiveHealth '17*, page 282–287, New York, NY, USA, 2017. Association for Computing Machinery. doi:10.1145/3154862.3154913.
- [41] Danco Davcev, Kosta Mitreski, Stefan Trajkovic, Viktor Nikolovski, and Nikola Koteli. Iot agriculture system based on lorawan. In *2018 14th IEEE International Workshop on Factory Communication Systems (WFCS)*, pages 1–4, 2018. doi:10.1109/WFCS.2018.8402368.
- [42] Olivier Debauche, Meryem Elmoulat, Mahmoudi Saïd, Slimane Boukraa, Pierre Manneback, and Frederic Lebeau. Web monitoring of bee health for researchers and beekeepers based on the internet of things. *The 8th International Symposium on Frontiers in Ambient and Mobile Systems (FAMS 2018)*, 130 :991–998, 04 2018. doi:10.1016/j.procs.2018.04.103.

- [43] Francois Delobel, Nancy El Rachkidy, and Alexandre Guitton. Analysis of the delay of confirmed downlink frames in class b of lorawan. In *2017 IEEE 85th Vehicular Technology Conference (VTC Spring)*, pages 1–6, 2017. doi:[10.1109/VTCSpring.2017.8108412](https://doi.org/10.1109/VTCSpring.2017.8108412).
- [44] Silvia Demetri, Marco Zúñiga, Gian Pietro Picco, Fernando Kuipers, Lorenzo Bruzzone, and Thomas Telkamp. Automated estimation of link quality for lora : A remote sensing approach. In *2019 18th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, pages 145–156, 2019. doi:[10.1145/3302506.3310396](https://doi.org/10.1145/3302506.3310396).
- [45] Arnaud Durand, Pascal Gremaud, and Jacques Pasquier. Resilient, crowd-sourced lpwan infrastructure using blockchain. In *Proceedings of the 1st Workshop on Cryptocurrencies and Blockchains for Distributed Systems, CryBlock'18*, page 25–29, New York, NY, USA, 2018. Association for Computing Machinery. doi:[10.1145/3211933.3211938](https://doi.org/10.1145/3211933.3211938).
- [46] Arnaud Durand, Pascal Gremaud, and J. Pasquier-Rocha. Decentralized lpwan infrastructure using blockchain and digital signatures. *Concurrency and Computation : Practice and Experience*, 32, 2020.
- [47] Minar El-Aasser, Tallal Elshabrawy, and Mohamed Ashour. Joint spreading factor and coding rate assignment in lorawan networks. In *2018 IEEE Global Conference on Internet of Things (GCIoT)*, pages 1–7, 2018. doi:[10.1109/GCIoT.2018.8620147](https://doi.org/10.1109/GCIoT.2018.8620147).
- [48] Rida El Chall, Samer Lahoud, and Melhem El Helou. Lorawan network : Radio propagation models and performance evaluation in various environments in lebanon. *IEEE Internet of Things Journal*, 6(2) :2366–2378, 2019. doi:[10.1109/JIOT.2019.2906838](https://doi.org/10.1109/JIOT.2019.2906838).
- [49] Mohamed Eldefrawy, Ismail Butun, Nuno Pereira, and Mikael Gidlund. Formal security analysis of lorawan. *Computer Networks*, 148 :328–339, 2019. URL : <https://www.sciencedirect.com/science/article/pii/S1389128618306145>, doi:<https://doi.org/10.1016/j.comnet.2018.11.017>.
- [50] Mehmet Ali Ertürk, M.Ali Aydin, Talha Büyükakkaşlar, and Hayrettin Evirgen. A survey on lorawan architecture, protocol and technologies. *Future Internet*, 11 :216, 10 2019. doi:[10.3390/fi11100216](https://doi.org/10.3390/fi11100216).
- [51] Bernat Carbonés Fargas and Martin Nordal Petersen. Gps-free geolocation using lora in low-power wans. In *2017 Global Internet of Things Summit (GIoTS)*, pages 1–6, 2017. doi:[10.1109/GIoTS.2017.8016251](https://doi.org/10.1109/GIoTS.2017.8016251).
- [52] Arshad Farhad, Dae-Ho Kim, and Jae-Young Pyun. Resource allocation to massive internet of things in lorawans. *Sensors*, 20 :20, 05 2020. doi:[10.3390/s20092645](https://doi.org/10.3390/s20092645).
- [53] Petr Fedchenkov, Arkady Zaslavsky, and Inna Sosunova. Enabling smart waste management with sensorized garbage bins and low power data communications network. In *Proceedings of the Seventh International Conference on the Internet of Things, IoT '17*, New York, NY, USA, 2017. Association for Computing Machinery. doi:[10.1145/3131542.3140264](https://doi.org/10.1145/3131542.3140264).

- [54] Guillaume Ferre. Collision and packet loss analysis in a lorawan network. In *2017 25th European Signal Processing Conference (EUSIPCO)*, pages 2586–2590, 2017. doi:10.23919/EUSIPCO.2017.8081678.
- [55] Joseph Finnegan, Stephen Brown, and Ronan Farrell. Evaluating the scalability of lorawan gateways for class b communication in ns-3. In *2018 IEEE Conference on Standards for Communications and Networking (CSCN)*, pages 1–6, 2018. doi:10.1109/CSCN.2018.8581759.
- [56] Orestis Georgiou and Usman Raza. Low power wide area network analysis : Can lora scale? *IEEE Wireless Communications Letters*, 6(2) :162–165, 2017. doi:10.1109/LWC.2016.2647247.
- [57] Fabrizio J. Grión, Gabriel O. Petracca, Daniel F. Lipuma, and Enrique R. Amigó. Lora network coverage evaluation in urban and densely urban environment simulation and validation tests in autonomous city of buenos aires. In *2017 XVII Workshop on Information Processing and Control (RPIC)*, pages 1–5, 2017. doi:10.23919/RPIC.2017.8214345.
- [58] Rami Hamdi, Marwa Qaraqe, and Saud Althunibat. Dynamic spreading factor assignment in lora wireless networks. In *ICC 2020 - 2020 IEEE International Conference on Communications (ICC)*, pages 1–5, 2020. doi:10.1109/ICC40277.2020.9149243.
- [59] Mohamed Hamnache. LoRa-L3SFA. original-date : 2022-10-07T16 :09 :43Z. URL : <https://github.com/mohamedhamnache/LoraSim3>.
- [60] Mohamed Hamnache. LoRaDL. original-date : 2022-10-07T16 :09 :43Z. URL : <https://github.com/mohamedhamnache/LoRaDL>.
- [61] Mohamed Hamnache. mohamedhamnache/lora-dns : Lora-handover-roaming, May 2021. URL : <https://github.com/mohamedhamnache/LoRa-DNS>.
- [62] Mohamed Hamnache, Rahim Kacimi, and André-Luc Beylot. Optimisation de l'allocation des facteurs d'étalement dans les Réseaux LoRaWAN. In *6èmes Rencontres Francophones sur la Conception de Protocoles, l'Évaluation de Performance et l'Expérimentation des Réseaux de Communication (CORES 2021)*, La Rochelle, France, May 2021. URL : <https://hal.archives-ouvertes.fr/hal-03220095>.
- [63] Mohamed Hamnache, Rahim Kacimi, and André-Luc Beylot. Fundamentals and Performance of the LoRaWAN Downlink : an Experimental Approach. In *IEEE Global Communications Conference (GLOBECOM 2022)*, Rio de Janeiro, Brazil, December 2022. IEEE, IEEE. URL : <https://hal.archives-ouvertes.fr/hal-03741970>.
- [64] Mohamed Hamnache, Rahim Kacimi, and André-Luc Beylot. Itinérance dans les réseaux LoRaWAN . In *7èmes Rencontres Francophones sur la Conception de Protocoles, l'Évaluation de Performance et l'Expérimentation des Réseaux de Communication (CORES 2022)*, Saint-Rémy-Lès-Chevreuse, France, May 2022. Conférence co-localisée avec Algotel 2022. URL : <https://hal.archives-ouvertes.fr/hal-03657148>.

- [65] Mohamed Hammache, Rahim Kacimi, and André-Luc Beylot. Joint load-balancing and power control strategy to maximize the data extraction rate of lorawan networks. *Computer Networks (révisions mineures)*, 2022.
- [66] Mohamed Hammache, Rahim Kacimi, and André-Luc Beylot. L3sfa : Load shifting strategy for spreading factor allocation in lorawan systems. In *2020 IEEE 45th Conference on Local Computer Networks (LCN)*, pages 216–224, 2020. doi:10.1109/LCN48667.2020.9314777.
- [67] Mohamed Hammache, Rahim Kacimi, and André-Luc Beylot. Unifying lorawan networks by enabling the roaming capability. In *2021 IEEE 46th Conference on Local Computer Networks (LCN)*, pages 371–374, 2021. doi:10.1109/LCN52139.2021.9524996.
- [68] Mohamed Hammache, Rahim Kacimi, and André-Luc Beylot. Enabling an inter-operator roaming capability in lorawan networks. *Ad Hoc Networks*, 139 :103025, 2023. doi:https://doi.org/10.1016/j.adhoc.2022.103025.
- [69] Jetmir Haxhibeqiri, Floris Van den Abeele, Ingrid Moerman, and Jeroen Hoebeke. Lora scalability : A simulation model based on interference measurements. *Sensors*, 17(6), 2017. URL : <https://www.mdpi.com/1424-8220/17/6/1193>, doi:10.3390/s17061193.
- [70] Yujun Hou, Zujun Liu, and Dechun Sun. A novel mac protocol exploiting concurrent transmissions for massive lora connectivity. *Journal of Communications and Networks*, 22(2) :108–117, 2020. doi:10.1109/JCN.2020.000005.
- [71] Xin Huang, Paul Craig, Hangyu Lin, and Zheng Yan. Seciot : A security framework for the internet of things. *Sec. and Commun. Netw.*, 9(16) :3083–3094, nov 2016. doi:10.1002/sec.1259.
- [72] Matti Hämäläinen and Xinrong Li. Recent advances in body area network technology and applications. *International Journal of Wireless Information Networks*, 24, 03 2017. doi:10.1007/s10776-017-0348-1.
- [73] Ahmad Rizan Ibrahim, Nik Hisham Nik Ibrahim, Ahmad Nizar Harun, Mohamed Rawidean Mohd Kassim, Shamsul Effendy Kamaruddin, and Gunawan Witjaksono. Bird counting and climate monitoring using lorawan in swiftlet farming for ir4.0 applications. In *2018 2nd International Conference on Smart Sensors and Application (ICSSA)*, pages 33–37, 2018. doi:10.1109/ICSSA.2018.8535955.
- [74] Ahmad Rizan Ibrahim, Nik Hisham Nik Ibrahim, Ahmad Nizar Harun, Mohamed Rawidean Mohd Kassim, Shamsul Effendy Kamaruddin, Liang Kim Meng, Chong Jin Hui, and Gunawan Witjaksono. Automated monitoring and lorawan control mechanism for swiftlet bird house. In *2018 International Conference on Intelligent and Advanced System (ICIAS)*, pages 1–5, 2018. doi:10.1109/ICIAS.2018.8540562.
- [75] Augustine Ikpehai, Bamidele Adebisi, Khaled M. Rabie, Kelvin Anoh, Ruth E. Ande, Mohammad Hammoudeh, Haris Gacanin, and Uche M. Mbanaso. Low-power wide area network technologies for internet-of-things : A comparative review.

- IEEE Internet of Things Journal*, 6(2) :2225–2240, 2019. doi:[10.1109/JIOT.2018.2883728](https://doi.org/10.1109/JIOT.2018.2883728).
- [76] Waheb A. Jabbar, M. Ismail, and Rosdiadee Nordin. Mba-olsr : A multipath battery aware routing protocol for manets. In *2014 5th International Conference on Intelligent Systems, Modelling and Simulation*, pages 630–635, 2014. doi:[10.1109/ISMS.2014.114](https://doi.org/10.1109/ISMS.2014.114).
- [77] Waheb A. Jabbar, Hiew Kuet Shang, Saidatul N. I. S. Hamid, Akram A. Almomhammedi, Roshahliza M. Ramli, and Mohammed A. H. Ali. Iot-bbms : Internet of things-based baby monitoring system for smart cradle. *IEEE Access*, 7 :93791–93805, 2019. doi:[10.1109/ACCESS.2019.2928481](https://doi.org/10.1109/ACCESS.2019.2928481).
- [78] Jerrin George James and Sreekumar Nair. Efficient, real-time tracking of public transport, using lorawan and rf transceivers. In *TENCON 2017 - 2017 IEEE Region 10 Conference*, pages 2258–2261, 2017. doi:[10.1109/TENCON.2017.8228237](https://doi.org/10.1109/TENCON.2017.8228237).
- [79] Hassan Jradi, Abed Ellatif Samhat, Fabienne Nouvel, Mohamad Mroue, and Jean-Christophe Prévotet. Overview of the mobility related security challenges in lpwans. *Computer Networks*, 186 :107761, 2021. URL : <https://www.sciencedirect.com/science/article/pii/S1389128620313384>, doi:<https://doi.org/10.1016/j.comnet.2020.107761>.
- [80] Aimad Karkouch, Hajar Mousannif, and Hassan Al Moatassime. Cads : A connected assistant for driving safe. *Procedia Computer Science*, 127 :353–359, 01 2018. doi:[10.1016/j.procs.2018.01.132](https://doi.org/10.1016/j.procs.2018.01.132).
- [81] Arzad Alam Kherani and K.M. Poonam Maurya. Improved packet detection in lora-like chirp spread spectrum systems. In *2019 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS)*, pages 1–4, 2019. doi:[10.1109/ANTS47819.2019.9118076](https://doi.org/10.1109/ANTS47819.2019.9118076).
- [82] Alex Koohang, Carol Springer Sargent, Jeretta Horn Nord, and Joanna Paliszkiwicz. Internet of things (IoT) : From awareness to continued use. 62 :102442. URL : <https://www.sciencedirect.com/science/article/pii/S0268401221001353>, doi:<https://doi.org/10.1016/j.ijinfomgt.2021.102442>.
- [83] Anton Kos, Veljko Milutinović, and Anton Umek. Challenges in wireless communication for connected sensors and wearable devices used in sport biofeedback applications. *Future Gener. Comput. Syst.*, 92(C) :582–592, mar 2019. doi:[10.1016/j.future.2018.03.032](https://doi.org/10.1016/j.future.2018.03.032).
- [84] Rachel Kufakunesu, Gerhard P. Hancke, and Adnan M. Abu-Mahfouz. A survey on adaptive data rate optimization in lorawan : Recent solutions and major challenges. *Sensors*, 20(18), 2020. URL : <https://www.mdpi.com/1424-8220/20/18/5044>, doi:[10.3390/s20185044](https://doi.org/10.3390/s20185044).
- [85] Janos Lamberg-Liszky and Tadas Lisauskas. An alternative roaming model inlorawan, 2018.

- [86] Mads Lauridsen, Istvan Z. Kovacs, Preben Mogensen, Mads Sorensen, and Steffen Holst. Coverage and capacity analysis of lte-m and nb-iot in a rural area. In *2016 IEEE 84th Vehicular Technology Conference (VTC-Fall)*, pages 1–5, 2016. doi:10.1109/VTCFall.2016.7880946.
- [87] Joannes I. Laveyne, Brecht Zwaenepoel, Greet Van Eetvelde, and Lieven Vandevelde. Potential of domestically provided ancillary services to the electrical grid. In *2017 52nd International Universities Power Engineering Conference (UPEC)*, pages 1–6, 2017. doi:10.1109/UPEC.2017.8231906.
- [88] Alexandru Lavric, Adrian I. Petrariu, and Valentin Popa. Sigfox communication protocol : The new era of iot? In *2019 International Conference on Sensing and Instrumentation in IoT Era (ISSI)*, pages 1–4, 2019. doi:10.1109/ISSI47111.2019.9043727.
- [89] Qi Li, Zhanghua Liu, and Junsheng Xiao. A data collection collar for vital signs of cows on the grassland based on lora. In *2018 IEEE 15th International Conference on e-Business Engineering (ICEBE)*, pages 213–217, 2018. doi:10.1109/ICEBE.2018.00041.
- [90] Jin-Taek Lim and Youngnam Han. Spreading factor allocation for massive connectivity in lora systems. *IEEE Communications Letters*, 22(4) :800–803, 2018. doi:10.1109/LCOMM.2018.2797274.
- [91] Stefan Lippuner, Benjamin Weber, Mauro Salomon, Matthias Korb, and Qiuting Huang. Ec-gsm-iot network synchronization with support for large frequency offsets. In *2018 IEEE Wireless Communications and Networking Conference (WCNC)*, pages 1–6, 2018. doi:10.1109/WCNC.2018.8377168.
- [92] Yan-Ting Liu, Bo-Yi Lin, Xiao-Feng Yue, Zong-Xuan Cai, Zi-Xian Yang, Wei-Hong Liu, Song-Yi Huang, Jun-Lin Lu, Jing-Wen Peng, and Jen-Yeu Chen. A solar powered long range real-time water quality monitoring system by lorawan. In *2018 27th Wireless and Optical Communication Conference (WOCC)*, pages 1–2, 2018. doi:10.1109/WOCC.2018.8373792.
- [93] Marine Lorient, Ammar Aljer, and Isam Shahrour. Analysis of the use of lorawan technology in a large-scale smart city demonstrator. In *2017 Sensors Networks Smart and Emerging Technologies (SENSET)*, pages 1–4, 2017. doi:10.1109/SENSET.2017.8125011.
- [94] Aamir Mahmood, Emiliano Sisinni, Lakshmikanth Guntupalli, Raúl Rondón, Syed Ali Hassan, and Mikael Gidlund. Scalability analysis of a lora network under imperfect orthogonality. *IEEE Transactions on Industrial Informatics*, 15(3) :1425–1436, 2019. doi:10.1109/TII.2018.2864681.
- [95] Anto Merline Manoharan and Vimalathithan Rathinasabapathy. Smart water quality monitoring and metering using lora for smart villages. In *2018 2nd International Conference on Smart Grid and Smart Cities (ICSGSC)*, pages 57–61, 2018. doi:10.1109/ICSGSC.2018.8541336.

- [96] Poonam Maurya, Aatmjeet Singh, and Arzad Alam Kherani. A review : Spreading factor allocation schemes for lorawan. *Telecommun. Syst.*, 80(3) :449–468, jul 2022. doi:10.1007/s11235-022-00903-4.
- [97] Afef Mdhaffar, Tarak Chaari, Kaouthar Larbi, Mohamed Jmaiel, and Bernd Freisleben. Iot-based health monitoring via lorawan. In *IEEE EUROCON 2017 -17th International Conference on Smart Technologies*, pages 519–524, 2017. doi:10.1109/EUROCON.2017.8011165.
- [98] Kais Mekki, Eddy Bajic, Frédéric Chaxel, and Fernand Meyer. A comparative study of lpwan technologies for large-scale iot deployment. 5 :1–7, 03 2019. doi:10.1016/j.ictc.2017.12.005.
- [99] Juha Petaejaevaervi Mikhaylov, Konstantin and Tuomo 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.
- [100] Konstantin Mikhaylov, Juha Petäjäjärvi, and Ari Pouttu. Effect of downlink traffic on performance of lorawan lpwa networks : Empirical study. In *2018 IEEE 29th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, pages 1–6, 2018. doi:10.1109/PIMRC.2018.8580721.
- [101] Shusuke Narieda, Takeo Fujii, and Kenta Umabayashi. Energy constrained optimization for spreading factor allocation in lorawan. *Sensors*, 20(16), 2020. URL : <https://www.mdpi.com/1424-8220/20/16/4417>, doi:10.3390/s20164417.
- [102] The Things Network. The things network. URL : <https://www.thethingsnetwork.org/>.
- [103] The Things Network. Lorawan architecture, Jun 2020. URL : <https://www.thethingsnetwork.org/docs/lorawan/architecture.html>.
- [104] Nik Hisham Nik Ibrahim, Ahmad Rizan Ibrahim, Ibrahim Mat, Ahmad Nizar Harun, and Gunawan Witjaksono. Lorawan in climate monitoring in advance precision agriculture system. In *2018 International Conference on Intelligent and Advanced System (ICIAS)*, pages 1–6, 2018. doi:10.1109/ICIAS.2018.8540598.
- [105] Ruhaizan Fazrren Ashraff Mohd Nor, Fadhlán H. K. Zaman, and Shamry Mubdi. Smart traffic light for congestion monitoring using lorawan. In *2017 IEEE 8th Control and System Graduate Research Colloquium (ICSGRC)*, pages 132–137, 2017. doi:10.1109/ICSGRC.2017.8070582.
- [106] Jeretta Horn Nord, Alex Koohang, and Joanna Paliszkievicz. The internet of things : Review and theoretical framework. *Expert Systems with Applications*, 133 :97–108, 2019. URL : <https://www.sciencedirect.com/science/article/pii/S0957417419303331>, doi:<https://doi.org/10.1016/j.eswa.2019.05.014>.
- [107] Alfonso Osorio, Maria Calle, Jose D. Soto, and John E. Candelo-Becerra. Routing in lorawan : Overview and challenges. *IEEE Communications Magazine*, 58(6) :72–76, 2020. doi:10.1109/MCOM.001.2000053.

- [108] Gyubong Park, Wooyeob Lee, and Inwhee Joe. Network resource optimization with reinforcement learning for low power wide area networks. *EURASIP J. Wirel. Commun. Netw.*, 2020(1), sep 2020. doi:10.1186/s13638-020-01783-5.
- [109] Nico Podevijn, Jens Trogh, Abdulkadir Karaagac, Jetmir Haxhibeqiri, Jeroen Hoebeke, Luc Martens, Pieter Suanet, Kim Hendrikse, David Plets, and Wout Joseph. Tdoa-based outdoor positioning in a public lora network. In *12th European Conference on Antennas and Propagation (EuCAP 2018)*, pages 1–4, 2018. doi:10.1049/cp.2018.0574.
- [110] Olaf Poenicke, Martin Kirch, Klaus Richter, and Sebastian Schwarz. Lorawan for iot applications in air cargo - development of a gse tracking system for dhl air cargo hub leipzig. In *Smart SysTech 2018; European Conference on Smart Objects, Systems and Technologies*, pages 1–6, 2018.
- [111] Zhijin Qin and Julie A. McCann. Resource efficiency in low-power wide-area networks for iot applications. In *GLOBECOM 2017 - 2017 IEEE Global Communications Conference*, pages 1–7, 2017. doi:10.1109/GLOCOM.2017.8254800.
- [112] Theodore Rappaport. *Wireless Communications : Principles and Practice*. Prentice Hall PTR, USA, 2nd edition, 2001.
- [113] Brecht Reynders, Wannes Meert, and Sofie Pollin. Power and spreading factor control in low power wide area networks. In *2017 IEEE International Conference on Communications (ICC)*, pages 1–6, 2017. doi:10.1109/ICC.2017.7996380.
- [114] Mattia Rizzi, Paolo Ferrari, Alessandra Flammini, Emiliano Sisinni, and Mikael Gidlund. Using lora for industrial wireless networks. In *2017 IEEE 13th International Workshop on Factory Communication Systems (WFCS)*, pages 1–4, 2017. doi:10.1109/WFCS.2017.7991972.
- [115] Moreshwar Salpekar, Pankaj Mohan Gupta, and Pravesh Kumar Tejan. Smart water/gas distribution system with focus on user safety. In *2018 International Conference on Smart City and Emerging Technology (ICSCET)*, pages 1–3, 2018. doi:10.1109/ICSCET.2018.8537309.
- [116] R. M. Sandoval, A. Garcia-Sanchez, and J. Garcia-Haro. Optimizing and updating lora communication parameters : A machine learning approach. *IEEE Transactions on Network and Service Management*, 16(3) :884–895, 2019.
- [117] R. M. Sandoval, D. Rodenas-Herraiz, A. Garcia-Sanchez, and J. Garcia-Haro. Deriving and updating optimal transmission configurations for lora networks. *IEEE Access*, 8 :38586–38595, 2020.
- [118] Ruben Sandoval, Antonio-Javier Garcia-Sanchez, and Joan Garcia-Haro. Performance optimization of lora nodes for the future smart city/industry. *EURASIP Journal on Wireless Communications and Networking*, 2019, 12 2019. doi:10.1186/s13638-019-1522-1.
- [119] Rachida Saroui, Alexandre Guitton, Oana Iova, and Fabrice Valois. Uplink and downlink are not orthogonal in LoRaWAN! In *IEEE VTC2022-Fall*, Londres,

- United Kingdom, September 2022. URL : <https://hal.archives-ouvertes.fr/hal-03789064>.
- [120] Jothi Prasanna Shanmuga Sundaram, Wan Du, and Zhiwei Zhao. A survey on lora networking : Research problems, current solutions, and open issues. *IEEE Communications Surveys Tutorials*, 22(1) :371–388, 2020. doi:10.1109/COMST.2019.2949598.
- [121] Yuvraj Singh. Comparison of okumura, hata and cost-231 models on the basis of path loss and signal strength. *International Journal of Computer Applications*, 59 :37–41, 12 2012. doi:10.5120/9594-4216.
- [122] Emiliano Sisinni, Abusayeed Saifullah, Song Han, Ulf Jennehag, and Mikael Gidlund. Industrial internet of things : Challenges, opportunities, and directions. *IEEE Transactions on Industrial Informatics*, 14(11) :4724–4734, 2018. doi:10.1109/TII.2018.2852491.
- [123] René Sørensen, Nasrin Razmi, Jimmy Nielsen, and Petar Popovski. Analysis of lorawan uplink with multiple demodulating paths and capture effect. pages 1–6, 05 2019. doi:10.1109/ICC.2019.8761142.
- [124] Duc-Tuyen Ta, Kinda Khawam, Samer Lahoud, Cedric Adjih, and Steven Martin. Lora-mab : Toward an intelligent resource allocation approach for lorawan. In *2019 IEEE Global Communications Conference (GLOBECOM)*, pages 1–6, 2019. doi:10.1109/GLOBECOM38437.2019.9013345.
- [125] Min Ye Thu, Wunna Htun, Yan Lin Aung, Pyone Ei Ei Shwe, and Nay Min Tun. Smart air quality monitoring system with lorawan. In *2018 IEEE International Conference on Internet of Things and Intelligence System (IOTAIS)*, pages 10–15, 2018. doi:10.1109/IOTAIS.2018.8600904.
- [126] E. M. Torroglosa-Garcia, J. M. A. Calero, J. B. Bernabe, and A. Skarmeta. Enabling roaming across heterogeneous iot wireless networks : Lorawan meets 5g. *IEEE Access*, 8 :103164–103180, 2020. doi:10.1109/ACCESS.2020.2998416.
- [127] Maksudjon Usmonov and Francesco Gregoretti. Design and implementation of a lora based wireless control for drip irrigation systems. In *2017 2nd International Conference on Robotics and Automation Engineering (ICRAE)*, pages 248–253, 2017. doi:10.1109/ICRAE.2017.8291389.
- [128] Floris Van den Abeele, Jetmir Haxhibeqiri, Ingrid Moerman, and Jeroen Hoebeke. Scalability analysis of large-scale lorawan networks in ns-3. *IEEE Internet of Things Journal*, 4(6) :2186–2198, 2017. doi:10.1109/JIOT.2017.2768498.
- [129] Lorenzo Vangelista. Frequency shift chirp modulation : The lora modulation. *IEEE Signal Processing Letters*, 24(12) :1818–1821, 2017. doi:10.1109/LSP.2017.2762960.
- [130] Nadège Varsier and Jean Schwoerer. Capacity limits of lorawan technology for smart metering applications. In *2017 IEEE International Conference on Communications (ICC)*, pages 1–6, 2017. doi:10.1109/ICC.2017.7996383.

- [131] Hai Wang and Abraham O. Fapojuwo. A survey of enabling technologies of low power and long range machine-to-machine communications. *IEEE Communications Surveys Tutorials*, 19(4) :2621–2639, 2017. doi:[10.1109/COMST.2017.2721379](https://doi.org/10.1109/COMST.2017.2721379).
- [132] Shie-Yuan Wang, Ji-Jhe Zou, Yo-Ru Chen, Chun-Chia Hsu, Yu-Hsiang Cheng, and Chia-Hung Chang. Long-term performance studies of a lorawan-based pm2.5 application on campus. In *2018 IEEE 87th Vehicular Technology Conference (VTC Spring)*, pages 1–5, 2018. doi:[10.1109/VTCSpring.2018.8417489](https://doi.org/10.1109/VTCSpring.2018.8417489).
- [133] Antoine Waret, Megumi Kaneko, Alexandre Guitton, and Nancy El Rachkidy. Lora throughput analysis with imperfect spreading factor orthogonality. *IEEE Wireless Communications Letters*, 8(2) :408–411, 2019. doi:[10.1109/LWC.2018.2873705](https://doi.org/10.1109/LWC.2018.2873705).
- [134] Tugrul Yatagan and Sema Oktug. Smart spreading factor assignment for lorawans. In *2019 IEEE Symposium on Computers and Communications (ISCC)*, pages 1–7, 2019. doi:[10.1109/ISCC47284.2019.8969608](https://doi.org/10.1109/ISCC47284.2019.8969608).
- [135] Fanghao Yu, Ziming Zhu, and Zhong Fan. Study on the feasibility of lorawan for smart city applications. In *2017 IEEE 13th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, pages 334–340, 2017. doi:[10.1109/WiMOB.2017.8115748](https://doi.org/10.1109/WiMOB.2017.8115748).
- [136] Guibing Zhu, Chun-Hao Liao, Theerat Sakdejayont, I-Wei Lai, Yoshiaki Narusue, and Hiroyuki Morikawa. Improving the capacity of a mesh lora network by spreading-factor-based network clustering. *IEEE Access*, 7 :21584–21596, 2019. doi:[10.1109/ACCESS.2019.2898239](https://doi.org/10.1109/ACCESS.2019.2898239).
- [137] Nicholas Zinas, Sotirios Kontogiannis, George Kokkonis, Stavros Valsamidis, and Ioannis Kazanidis. Proposed open source architecture for long range monitoring. the case study of cattle tracking at pogoniani. In *Proceedings of the 21st Pan-Hellenic Conference on Informatics*, PCI 2017, New York, NY, USA, 2017. Association for Computing Machinery. doi:[10.1145/3139367.3139437](https://doi.org/10.1145/3139367.3139437).

Résumé

LoRaWAN est l'un des principaux protocoles de communication sans fil déployés pour répondre aux exigences des applications IoT (Internet of Things) nécessitant une communication à longue portée avec une faible consommation d'énergie. On compte actuellement plus d'un milliard de dispositifs IoT utilisés dans le monde et LoRaWAN apparaît comme l'une des solutions les plus prometteuses pour de nombreuses applications. Bien que l'étude de ces réseaux, des mécanismes associés et de leurs performances soit un sujet particulièrement brûlant, certains aspects restent peu explorés, notamment la gestion de la mobilité et de l'itinérance (*roaming*). De même, de nombreux travaux reposent exclusivement sur des approches de simulation pour leur validation ou manquent d'outils et de bancs de test pour conduire des expérimentations réelles.

Dans cette thèse, nous proposons d'apporter des solutions à des problèmes bien connus au sein de la communauté LoRaWAN, notamment l'allocation des paramètres de transmission pour augmenter les performances du réseau. Nous proposons alors L3SFA et L3SFA-TPC pour améliorer le taux de délivrance, la capacité du réseau et la consommation énergétique des terminaux. Par la suite, nous nous sommes concentrés sur des sujets moins explorés. Nous proposons LoRaRoam, le premier système prenant en charge l'itinérance active ou itinérance *Handover* dans les réseaux LoRaWAN.

Face au manque ou à l'absence d'outils de référence, nous nous sommes attachés au développement de solutions et au déploiement de bancs de test pour la validation des solutions proposées et pour l'analyse de leurs performances à large échelle. Nous avons conçu *LoRa Roaming Emulator* (LRE), le premier émulateur d'itinérance dédié à l'évaluation des réseaux LoRaWAN itinérant. De plus, nous avons proposé LoRaDL, un framework pour prendre en charge les communications sur la liaison descendante (commandes MAC, trames de données) facilitant ainsi la mise en œuvre des solutions centrées sur les passerelles (*gateway-centric*) et l'évaluation des performances de la liaison descendante.

Mots clés : LoRaWAN, allocation de paramètres de transmission, itinérance, liaison descendante, expérimentations.

Abstract

LoRaWAN is one of the leading wireless communication protocols deployed to meet the requirements of IoT (Internet of Things) applications requiring long-range communication with low power consumption. Currently, more than one billion IoT devices in use around the world and LoRaWAN is emerging as one of the most promising communication protocols for many applications. Although the study of these networks, the associated mechanisms and their performances is a particularly hot topic, some aspects remain unexplored, in particular the management of mobility and roaming. Similarly, many works rely exclusively on simulation approaches for their validation or lack tools and testbeds to perform real experiments.

In this thesis, we propose to provide solutions to well-known problems within the LoRaWAN community, in particular the allocation of transmission parameters to increase network performance. We then propose L3SFA and L3SFA-TPC to improve the Data Extraction Rate (DER), the network capacity, and the terminal energy consumption. Thereafter, we focus on less explored topics. We propose LoRaRoam, the first system supporting Handover Roaming in LoRaWAN networks.

Faced with the lack or absence of reference tools, we have focused on developing solutions and deploying testbeds to validate the proposed solutions and analyze their performance on a large scale. We designed *LoRa Roaming Emulator* (LDE), the first roaming emulator dedicated to the evaluation of roaming LoRaWAN networks. In addition, we proposed LoRaDL, a framework to support downlink communications (MAC commands, data frames) facilitating the implementation of gateway-centric solutions and the evaluation of downlink performance.

Keywords : LoRaWAN, transmission parameters allocation, roaming, downlink, experiments.

