

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université Abderrahmane Mira de Béjaia
Département de Recherche Opérationnelle



Mémoire en vue d'obtention du Diplôme de Master
en Mathématiques Appliquées

Option :

Optimisation et Fiabilité des Réseaux de Communication

**OPTIMISATION DU PLACEMENT DES PASSERELLES DANS
LoRaWAN**

Réalisé par : **HADDAD DRIFA**
HAMENNI HALIMA

Soutenu le 03/07/2024, devant le jury composé de :

BELKHIRI Louisa	M.C.B	Présidente	Université de Béjaia.
CHERIFI Feriel	M.C.B	Examinatrice	Université de Béjaia.
NAIT MOHAND Nacim	M.C.B	Examinateur	Université de Béjaia.
ATMANI Mouloud	M.C.A	Encadrant	Université de Béjaia.
KHIMOUM Nouredine	M.C.A	Co-encadrant	Université de Béjaia.

Année Universitaire 2023 – 2024

Remerciement

Avant toutes choses, nous remercions dieu le tout puissant de nous avoir donné le courage, la force, la volonté, la détermination et la persévérance qu'il fallait pour la réalisation de ce modeste travail.

*Nous tenons tout d'abord à remercier chaleureusement nos encadrants, M^r **ATMANI Mouloud** et M^r **KHIMOUM Nouredine**, pour leur précieuse guidance, leur conseils avisés et leur soutien constant tout au long de ce projet. Leur expertise et leur patience ont été des atouts essentiels à la réussite de ce travail.*

Je remercie également mes professeurs, pour leurs enseignements et leurs encouragements. Leurs remarques pertinentes et leur disponibilité durant tout notre parcours universitaire.

*Nous tenons à exprimer notre reconnaissance à l'entreprise **Cevital** pour nous avoir accueilli chaleureusement au sein de leur équipe. Un grand merci à toute l'équipe de travail pour leur disponibilité et leur bienveillance. Leur volonté de partager leurs connaissances et de nous faire découvrir le maximum de choses sur leur travail a été une expérience extrêmement enrichissante et formatrice.*

Un grand merci à tous les membres du jury qui ont prit le temps d'examiner notre travail .

Finalement nous remercions toute personne qui a pu contribué de près comme de loin à la réussite de ce travail.

Dédicace

Je dédie ce mémoire à moi-même, en hommage à ma force, ma résilience et mes innombrables sacrifices. Chaque défi rencontré et chaque obstacle surmonté témoignent de ma détermination et de mon engagement sans faille. Ce travail est le reflet de ma capacité à persévérer et à transformer les difficultés en opportunités de croissance et de réussite.

*Je dédie ce mémoire à **Dieu**, infinie source de lumière, de sagesse et de force. C'est grâce à Sa guidance et à Sa bienveillance que j'ai pu traverser les défis de ce parcours académique et atteindre cet accomplissement. Que cette œuvre soit un humble témoignage de ma gratitude pour Ses bénédictions et Son soutien indéfectible.*

À ceux qui ont illuminé mon chemin et donné du sens à mes efforts, je dédie ce mémoire, à tous ceux qui ont marqué ma vie et contribué à mon succès.

À ma mère,

Ma guerrière et mon idole dans la vie. Ta détermination et ton amour inébranlable et votre soutien indéfectible m'ont toujours inspiré à donner le meilleur de moi-même. T'as été toujours à mes côtés et tu le seras à jamais aussi, tu m'as relevé au ciel et maintenant je le suis grâce à toi. Ma réussite est dédiée à toi. Merci pour tes innombrables sacrifices et pour avoir toujours cru en moi, même dans les moments les plus difficiles.

À mon père,

Mon premier héros et mon pilier dans la vie. Ta sagesse et tes encouragements m'ont guidé tout au long de ce voyage. Ta phrase "Tu es ma main droite" sonore toujours dans mon oreille et je te promets que je vais l'être. Merci pour ta patience, tes conseils et ton soutien infailible, qui m'ont aidé à surmonter chaque obstacle rencontré sur mon chemin. Merci pour ta confiance en moi qui est ma source de réussite aujourd'hui.

À ma grande sœur Lilia,

Ma deuxième maman. Ton amour, ta protection et tes conseils m'ont été d'une aide précieuse. Merci pour ta présence constante, pour tes mots rassurants et pour être un modèle de courage et de dévouement.

À ma petite sœur Lisa,

Ma lumière et mon étincelle. Ta joie de vivre et ton enthousiasme ont été une source inépuisable de motivation pour moi. Merci d'avoir égayé mes journées et d'avoir toujours su me rappeler l'importance de voir le côté positif de la vie et merci d'être l'amie fidèle.

À ma petite nièce adorée, Jordana,

Mon rayon de soleil et source d'inspiration, je te dédie ce mémoire pour te montrer que tout est possible en rêvant et en travaillant dur. Tu es la relève qui me pousse à toujours viser l'excellence.

À ma copine et binôme Halima,

Compagne de voyage académique et complice de mes inspirations. Ton soutien inébranlable, tes idées lumineuses et notre collaboration ont enrichi chaque étape de ce parcours. Ensemble, nous avons surmonté les défis et célébré les réussites. Merci pour ta présence précieuse et ta contribution indéniable à ce travail.

À mes amis,

Merci pour votre présence constante, votre soutien inébranlable et votre amitié précieuse qui ont illuminé mon chemin. Vos encouragements m'ont permis de persévérer et de croire en mes rêves. Ce mémoire est le fruit précieux de votre soutien inestimable.

À tous ceux qui ont croisé mon chemin et contribué à mon épanouissement,

Merci pour votre bienveillance, votre encouragement et votre soutien, qui m'ont aidé à devenir la personne que je suis aujourd'hui.

Ce mémoire reflète votre amour, votre soutien et votre encouragement. Merci du fond du cœur pour tout ce que vous faites pour moi. Je vous aime plus que les mots ne peuvent le dire.

HADDAD Drifa

Dédicace

À cet enfant qui ne grandit pas, qui reste debout - lui et ses rêves - au fond de chacun de nous.. Dans l'espoir que nous, on grandisse assez pour réaliser ces rêves-là.

Le voyage n'a pas été court et ne devait pas l'être, le rêve n'était pas proche et le chemin n'était pas parsemé de facilités. Mais je l'ai fait et je l'ai accompli.

Louange à Dieu avec amour, gratitude et reconnaissance, grâce à lui, je regarde aujourd'hui un rêve tant attendu devenir réalité et j'en suis fière.

*À ma force après Dieu, mon premier et éternel soutien. Je suis reconnaissante que Dieu t'ait choisie pour être ma mère, un pilier inébranlable et une bénédiction. Ta tendresse et ton soutien inlassable ont été une source constante de réconfort et d'inspiration. Ta patience, ta sagesse et ton amour m'ont toujours guidé et motivé. Merci pour tes innombrables sacrifices et pour avoir toujours été là pour moi. Ma super-héroïne, mon amour infini. " **À mon ange pur, ma mère** " .*

*À celui dont l'amour incommensurable et le soutien indéfectible ont toujours été ma plus grande source de force. Ta sagesse, ton dévouement et tes encouragements constants m'ont guidé tout au long de mon existence. Tu as toujours cru en moi, même dans les moments les plus difficiles et ta présence constante m'a permis de surmonter chaque obstacle. L'homme de ma vie et mon amour éternel. " **À mon héros, mon père** " .*

*À ceux pour qui il est dit 'Nous allons renforcer ton bras avec ton frère'. À celui qui m'a tendu la main sans relâche et sans fatigue dans mes moments de faiblesse. **À mon idole, "mon frère"** .*

*À celle qui a cru en mes capacités, qui me rappelle ma force et se tient derrière moi comme mon ombre, la prunelle de mes yeux, **À ma Bêta, "ma petite sœur"** .*

*À cette personne avec qui la phrase 'Ton pied, mon pied' a un vrai sens, l'amitié qui nous lie est un bien précieux qui nous rend plus fortes. Notre relation est un repère sécurisant dans les moments tristes, douloureux, de doute comme dans les moments de joie, de réussite et de bonheur. **À ma copine et binôme "Drifa"** .*

*À celle qui marque sa présence même qu'avec des mots virtuels, merci d'exister. **"Ranim"** .*

À mes amis de la première heure et de la dernière ligne droite, ceux qui ont été une pluie bienfaisante pendant les années arides, je suis infiniment reconnaissante. Votre constance dans un monde incertain, votre présence inébran-

lable à mes côtés année après année, est un soutien colossal pour moi. Merci à chacun de vous qui rend la vie plus douce par vos mots, vos petites actions et vos gestes inattendus. J'espère de tout cœur que nos rires résonneront toujours aussi fort et nos souvenirs se multiplieront sans cesse et que "**La Familia**" puisse durer pour toujours.

À tous ceux qui ont été présents et m'ont apporté leur soutien, grands et petits gestes confondus, qui ont contribué à mon bien être. **À ma famille et proches.**

C'est avec une profonde reconnaissance et une gratitude infinie que je vous dédie ce travail. Votre soutien indéfectible et votre amour inconditionnel ont été des forces motrices essentielles qui ont contribué à façonner ma personne et ont nourri mon parcours de vie. Ce mémoire est un humble témoignage de l'impact précieux de votre présence dans ma vie.

Last but not least , i wanna thank me, i wanna thank me for believing in me, i wanna thank me for doing all this hard work, i wannna thank me for having no days off, i wanna thank me for never quitting, i wanna thank me for always beeing a giver and tryna give more than i receive, i wanna thank me for tryna do more right than wrong, i wanna thank me for just being me at all times.

Always and forever, with all my love.

HAMENNI Halima

Table des matières

Liste des figures	V
Liste des tables	VI
Liste des abréviations	VI
<i>Introduction générale</i>	1
1 Internet des Objets et Transmission des Données	3
1.1 Introduction	3
1.2 Internet des Objets (IoT)	3
1.2.1 Architecture de IoT	4
1.2.2 Technologie de transmission	6
1.2.3 Structure de LoRa :	8
1.2.4 Alliance LoRa :	8
1.2.5 Traitement et Stockage des Données dans LoRa :	8
1.3 Conclusion	10
2 Technologie LoRaWAN	11
2.1 LoRaWAN	11
2.2 LoRa vs LoRaWAN	12
2.2.1 LoRa	12
2.2.2 LoRaWAN	13
2.3 Architecture du Réseau	13
2.4 LoRa	14
2.5 Classes de terminaux	14
2.5.1 Classe A	14
2.5.2 Classe B	15
2.5.3 Classe C	15
2.6 LoRa MAC	16
2.6.1 Protocoles d'accès dans LoRa MAC :	17
2.7 Types de trames utilisées par LoRaWAN	19
2.8 Connexion au réseaux LoRaWAN	20
2.9 Couverture dans les réseaux LoRaWAN	23
2.10 Relation entre SF, débit et latence :	23
2.11 Applications de technologie LoRaWAN	24
2.12 Défis de LoRa	25
2.13 Conclusion	25
3 Etude de l'état de l'art	26
3.1 Introduction	26
3.2 Type de solutions proposées pour LoRaWAN	26
3.2.1 ALOHA	26
3.2.2 AI/ML :	29

3.2.3	La synchronisation Slotted :	30
3.3	Étude sur les travaux de placement des gateways	34
3.3.1	Description du problème	34
3.3.2	Formulation de problème	36
3.3.3	Algorithmes de découpe proposés	37
3.4	Critique de l'article intitulé : "K-Means Spreading Factor Allocation for Large-Scale LoRa Networks"	42
3.5	Conclusion	44
4	Optimisation de placement et duplication des passerelles dans un réseau LoRaWAN	45
4.1	Introduction	45
4.2	Contexte et description générale	45
4.2.1	Structure du réseau	45
4.2.2	Fonctionnement du réseau	46
4.2.3	Gestion du réseau	46
4.3	Problématique	46
4.3.1	Déploiement des gateways	47
4.3.2	Sélection du facteur d'étalement de spectre (SF)	47
4.3.3	Technologie de réseau	47
4.4	Motivations et objectifs	47
4.5	Optimisation de placement et duplication des gateway dans un ré- seau LoRaWAN (OPG-LoRaWAN)	48
4.5.1	Hypothèses	48
4.5.2	Placement stratégique des passerelles	48
4.5.3	Duplication des passerelles pour une meilleure fiabilité	48
4.5.4	Optimisation continue et ajustements	49
4.6	Modélisation mathématique avec la programmation linéaire	49
4.6.1	Description de la Programmation Linéaire	49
4.6.2	Contraintes	49
4.6.3	Approches	50
4.7	Algorithmes de la solution proposée	51
4.7.1	OGP-LoRaWAN Algorithm	52
4.7.2	Activation des passerelles	53
4.8	Conclusion	55
5	Implémentation des résultats	56
5.1	Introduction	56
5.2	Outils de développement	56
5.2.1	PyCharm	57
5.3	Etapes implémentées	58
5.3.1	Déploiement des End devices	58
5.3.2	Déploiement des Gateways	59
5.3.3	Résultats du programme	63
5.4	Conclusion	64
	Conclusion générale	65
	Bibliographie	68
	Résumé	68

Table des figures

1.1	Internet of Things [2].	4
1.2	Architecture de IoT [33].	4
1.3	NB-IoT [34].	6
1.4	Sigfox [21].	7
1.5	Structure de LoRa [36].	8
1.6	Plateformes IoT.	8
1.7	Les services Cloud.	9
1.8	Les trois solutions techniques d'IoT.	9
2.1	Pile Technologique des Réseaux LoRaWAN [35].	12
2.2	Architecture du Réseau LoRaWAN [12].	13
2.3	Fenêtres de réception de la Classe A	15
2.4	Fenêtres de réception de la Classe B	15
2.5	Fenêtres de réception de la Classe C	16
2.6	Consommation d'énergie et capacité en downlink (débit).	16
2.7	Exemple sur ALOHA PUR	17
2.8	Exemple S-ALOHA	18
2.9	Exemple CSMA	18
2.10	CSMA 1-persistant	18
2.11	CSMA non persistant	19
2.12	Trame de LoRa	19
2.13	Données de la charge utile	20
2.14	DevAddr, NwkSKey et AppSKey en ABP.	21
2.15	DevAddr, NwkSKey et AppSKey en OTAA.	21
2.16	DevAddr, NwkSKey et AppSKey en OTAA.	22
2.17	DevAddr, NwkSKey et AppSKey en OTAA.	22
2.18	Applications de la technologie LoRaWAN	24
3.1	Collision avec deux autres trames : scénarios de chevauchement et de non-chevauchement	27
3.2	Représentation de la fraction des EDs situés à l'intérieur du domaine de contention du transmetteur.	29
3.3	Fonction de mise à l'échelle pour l'ED interférant et le nombre d'interférents.	29
3.4	La relation entre l'Intelligence Artificielle (IA), l'Apprentissage Machine (Machine Learning) et le Deep Learning	30
3.5	Probabilité de collision par rapport au nombre d'appareils finaux.	32
3.6	Structure d'une supertrame pour la plage de charge utile i dans le schéma TS-VP-LoRa.	33
3.7	L'architecture de découpe IoT dans les réseaux LoRa.	36
3.8	Algorithme de découpe basée sur BIRCH	38

3.9	Allocation de surface en fonction de l'algorithme K-Means	40
3.10	Algorithme K-means.	41
3.11	Distance entre les centres de clusters et les paramètres de transmission.	42
3.12	Algorithme de clustering Fuzzy C-Means (FCM).	42
4.1	Réseau avec passerelles actives et inactives.	51
5.1	Python Logo	57
5.2	PyCharm Logo	57
5.3	Déploiement des End Devices.	59
5.4	Déploiement avec une passerelle.	60
5.5	Déploiement avec une passerelle (min).	60
5.6	Déploiement avec deux passerelles.	62
5.7	Déploiement avec deux passerelles(min).	62
5.8	Déploiement des End devices.	63
5.9	Déploiement du réseau	63
5.10	Déploiement du réseau avec SF différent	64

Liste des tableaux

2.1	Différences entre LoRa et LoRaWAN.	13
2.2	Comparaison des méthodes d'activation ABP et OTAA	23
2.3	Comparaison des méthodes d'activation ABP et OTAA	24
4.1	Variables utilisées dans l'algorithme.	52

Liste des Abréviations

- **ABP** : Activation By Personalization.
- **ADR** : Adaptive Data Rate.
- **AS** : Application Server.
- **B2B** : Business-to-Business.
- **BIRCH** : Balaced Iterative Reducing and Clustering using Hierarchies.
- **BW** : Largeur de band.
- **CDMA** : Code Division Multiples Access.
- **CF-tree** : Clustering Feature tree.
- **CR** : Code Rate.
- **CRM** : Customer Relationship Management.
- **CRC** : Cyclic Redundancy Check.
- **CSMA** : Carrier Sense Multiple Access.
- **CSS** : Chirp Spread Spectrum.
- **DL** : Deep Learning.
- **DR** : débit.
- **ED** : Energie Detection.
- **EUI** : Extended Unique Identifier.
- **FCA-LoRa** : Financial Conduct Authority Long Range.
- **FCM** : Fuzzy C-Means.
- **FDMA** : Frequency Division multiple Access.
- **fNS** : Forwarding Network Serve.
- **GSM** : Global System for Mobile Communications.
- **hNS** : Home Network Server.
- **IaaS** : Infrastructure as a Service.
- **IEEE** : Institute of Electrical and Electronics Engineers.
- **IoT** : Internet of Things.
- **IoT-GSI** : Internet of Things Global Standards Initiative.

- **IP** : Internet Protocol.
- **ISM** : Industriel, Scientifique, Médical.
- **IA** : Intelligence Artificielle.
- **LoRa** : Long Range.
- **LoRaWAN** : Long Range Wide Area Network.
- **LoRaLAN** : Longe Range Local Area Network.
- **LPWAN** : Low-Power Wide Area Network.
- **LPWA** : Low-Power Wide Area.
- **LTE-M** : Long Term Evolution for Machines .
- **MAC** : Medium Access Control.
- **ML** : Machine Learning.
- **MTC** : Machine-Type Communication Devices.
- **NB-IoT** : Narrow band Internet of Things.
- **NFC** : Near Field Communication.
- **NS** : Network Server.
- **NP-CSMA** : No Persistent Frequency Division multiple.
- **OTAA** : Over The Air Activation.
- **OGP** : Optimization Gateway Placement.
- **PaaS** : Platform as a Service.
- **P-ALOHA** : Pur ALOHA.
- **PL** : Programmation Linéaire.
- **QoS** : Quality of Service.
- **RFID** : Radio-Frequency Identification.
- **RSSI** : Received Signal Strength Indication.
- **R-ALOHA** : Reservation ALOHA.
- **RL** : Reinforcement learning.
- **SaaS** : Software as a Service.
- **S-ALOHA** : Slotted ALOHA.
- **S-LoRa** : Symmetric Long Range.
- **S-LoRaWAN** : Symmetric Long Range Wide Area Network.
- **Sigfox** : SIGnal FOX.
- **SCM** : Supply Chain Management.
- **SF** : Spreading Factor.

- **SNR** : Signal-to-Noise-Ratio.
- **sNS** : Serving Network Server.
- **SSL** : Secure Sockets Layer.
- **TDMA** : Time Division Multiple Access.
- **TCP/IP** : Transmission Control Protocol/Internet Protocol.
- **TP** : Puissance de transmission.
- **ToA** : Temps de transmission.
- **TS-VP-LoRa** : Time-Sensitive Virtual Private Long Range.
- **UIT** : Union Internationale des Télécommunications.
- **vNS** : serveur de réseau visité.
- **WCSS** : Within Cluster Sum of Square.

Introduction générale

L'Internet des Objets (IoT) représente un réseau d'objets physiques intégrant des capteurs, logiciels et autres technologies afin de communiquer et échanger des données avec d'autres dispositifs et systèmes via Internet. L'objectif principal de l'IoT est de rendre les systèmes plus efficaces, économiques et intelligents. Ces objets peuvent inclure une vaste gamme de dispositifs, allant des appareils électroménagers aux systèmes industriels, en passant par les dispositifs portables. L'IoT est révolutionnaire car il permet une interaction continue entre le monde physique et numérique, favorisant l'innovation dans divers secteurs tels que la santé, l'agriculture, les villes intelligentes et bien plus encore.

Le principal défi abordé dans ce mémoire concerne l'optimisation du placement des passerelles dans un réseau LoRaWAN pour assurer une couverture réseau optimale tout en minimisant les ressources (passerelles) utilisées. La problématique posée est : "Comment optimiser le nombre et le placement des passerelles dans un réseau LoRaWAN afin de garantir une couverture complète et une utilisation efficace des ressources, notamment en termes de consommation d'énergie" .

La solution proposée implique l'utilisation de techniques avancées de programmation linéaire pour modéliser et résoudre le problème de placement des passerelles. En intégrant des algorithmes spécifiques et des stratégies d'optimisation, cette approche vise à minimiser le nombre de passerelles tout en garantissant une couverture complète et en maximisant l'efficacité énergétique des dispositifs finaux. D'autre part, notre solution vise à dupliquer la connectivité pour chaque utilisateur final (End Device) afin d'assurer la transmission des données en cas de panne d'une passerelle.

Le contenu de chaque chapitre est détaillé dans les paragraphes qui suivent :

Chapitre 1 : (Internet of Things et Transmission des Données) Ce chapitre présente les concepts de base de l'Internet des Objets (IoT), en décrivant son architecture, ses applications et ses défis. Il examine aussi les diverses technologies de transmission de données employées dans les systèmes IoT, en soulignant leurs avantages et leurs limites.

Chapitre 2 : (Technologie LoRaWAN) Ce chapitre se concentre sur la technologie LoRaWAN. Il distingue LoRa de LoRaWAN, présente l'architecture du réseau, les classes de terminaux, et les défis associés à cette technologie.

Chapitre 3 : (Etude de l'état de l'art) Ce chapitre est dédié à l'étude de l'état de l'art. Il examine les recherches existantes sur le placement des passerelles et propose des critiques d'articles pertinents, fournissant une base théorique pour les solutions développées.

Chapitre 4 : (Optimisation du Placement des Passerelles) Ce chapitre se concentre sur l'optimisation du placement des passerelles dans un réseau LoRaWAN. Il présente une modélisation mathématique du problème, utilisant des

techniques de programmation linéaire pour déterminer la configuration optimale des passerelles. Le chapitre traite aussi de la modulation adaptative pour améliorer la couverture et l'efficacité des réseaux.

Chapitre 5 : (Implémentation des résultats) Le dernier chapitre présente l'implémentation pratique des solutions proposées. Il détaille les outils utilisés, les étapes d'implémentation, et les résultats obtenus, illustrant l'efficacité des approches développées.

Ce mémoire offre une contribution significative à l'optimisation des réseaux LoRaWAN dans le contexte de l'IoT. En combinant des techniques de programmation linéaire et des algorithmes d'optimisation, les solutions développées permettent d'améliorer la couverture réseau tout en minimisant les ressources nécessaires. Les résultats obtenus sont prometteurs pour l'avenir des réseaux IoT, suggérant que l'intégration de techniques avancées (IA) pourrait encore améliorer la performance et la robustesse de ces réseaux.

1

Internet des Objets et Transmission des Données

1.1 Introduction

L'Internet des objets (IoT) connecte divers objets physiques à Internet pour collecter et échanger des données. Ces objets peuvent inclure des appareils domestiques, des véhicules, et des capteurs industriels. Les technologies de transmission des données à faible débit et longue distance, comme LoRaWAN, Sigfox et NB-IoT, sont cruciales pour l'IoT car elles permettent une communication efficace et fiable avec une faible consommation d'énergie, même dans des conditions de transmission difficiles. Chaque technologie présente des avantages et des limitations spécifiques, les rendant adaptées à différentes applications IoT.

1.2 Internet des Objets (IoT)

Internet of Things est un réseau d'éléments identifiables de manière unique qui communiquent sans interaction humaine à l'aide de la connectivité IP (Internet Protocol). Il existe plusieurs définitions de l'IoT dans la littérature [2] [33]. Le groupe de travail «Internet of Things Global Standards Initiative (IoT-GSI)» piloté par l'Union Internationale des Télécommunications (UIT), considère l'IoT comme :

« Infrastructure mondiale pour la société de l'information qui permet de disposer des services évolués en interconnectant des objets (physiques ou virtuels) grâce aux technologies de l'information et de la communication interopérable existant ou en évolution ».

L'IoT désigne un réseau d'objets physiques connectés à Internet, capables de collecter et d'échanger des données, voir la figure 1.1 :



FIGURE 1.1 – Internet of Things [2].

1.2.1 Architecture de IoT

L'architecture d'un réseau IoT peut varier en fonction du système mis en place. La structure la plus basique est composée de trois couches distinctes mais nous en va détaillé et présenté quatre couches, voir la figure 1.2 :

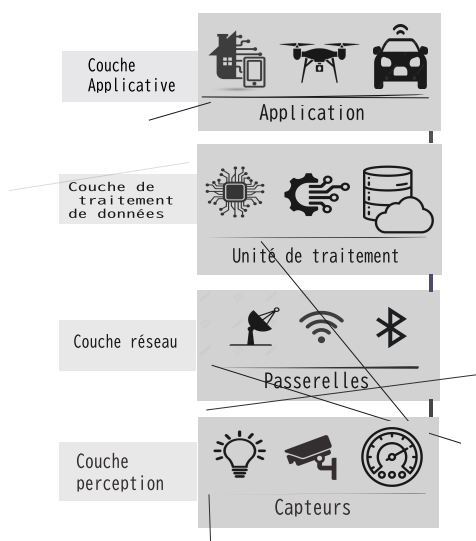


FIGURE 1.2 – Architecture de IoT [33].

A- Couche perception : Cette couche, également appelée couche physique, se compose des dispositifs terminaux qui collectent des données du monde réel. Ces données sont ensuite analysées par des applications numériques. Voici quelques appareils notables connectés à la couche de perception :

- a) **Capteurs :** les capteurs collectent des données environnementales (température, humidité, lumière, etc.) et les envoient à un serveur central pour analyse. Ils sont le point de départ de l'IoT :

- b) **Actionneurs** : convertissent les signaux électriques en actions physiques comme le mouvement ou le changement d'état.

B- Couche Transmission (Couche Réseau) : Dans un réseau IoT (Internet des objets), la couche réseau est cruciale pour permettre la communication entre les dispositifs IoT et l'infrastructure réseau. Elle utilise principalement le protocole IP pour l'adressage et l'acheminement des données, avec des versions comme IPv6 adaptées aux besoins en adressage massif. Les protocoles de routage tels que RPL sont employés pour gérer les conditions de faible puissance et les connexions instables. La couche réseau inclut également des mécanismes de sécurité pour protéger les données, des fonctions d'adaptation pour les changements de topologie et la mobilité, ainsi que des méthodes de gestion de la qualité de service (QoS) pour prioriser les transmissions critiques.

Elle comporte plusieurs réseaux de communication qui se sont classifiés par réseaux filaires et réseaux sans fil :

- **Les réseaux filaires** : tels que l'Ethernet ou la fibre optique, souvent surdimensionnés pour l'IoT en termes de débit et de coût.
- **Les réseaux sans fil à très courte distance** : comme le NFC et le RFID permettent l'échange de données entre des dispositifs proches les uns des autres..
- **Les réseaux sans fil courte distance** : le Wi-Fi, le Bluetooth et le Zigbee sont abordables et offrent un haut débit, mais ils ont une portée limitée et une consommation énergétique élevée pour les objets connectés.
- **Les réseaux cellulaires** : la 4G et la 5G offrent une large portée et une capacité élevée de transmission de données, mais elles consomment plus d'énergie et nécessitent des contrats avec les fournisseurs pour fonctionner.
- **Les réseaux cellulaires ou LPWAN privés** : des gateways fournissent une connectivité locale lorsque la couverture des réseaux publics est insuffisante.
- **Les réseaux LPWAN (Low-Power Wide Area Network)** : ces derniers transmettent des données de petits volumes sur de longues distances avec une faible consommation d'énergie.

C- Couche Traitement de Données : Cette couche fait référence à une partie d'un système informatique où les données sont manipulées et transformées selon les besoins spécifiques. Elle comprend des opérations telles que l'agrégation, la filtration, la transformation et l'analyse des données, souvent dans le but de les rendre utilisables pour des applications ou des décisions ultérieures. Cette couche joue un rôle crucial dans le traitement efficace et précis des informations, en transformant les données brutes en informations exploitables.

D- Couche Application : le dernier pilier de l'IoT concerne les outils de traitement et de visualisation des données, essentiels pour valoriser les informations collectées. Ils sont simples à installer et à configurer, comme les interfaces low-code ou no-code. Ces plateformes permettent de stocker, historiser, structurer, corrélérer, calculer et visualiser les données en temps réel. Les traitements peuvent inclure des algorithmes d'IA/ML pour des analyses complexes.

1. **Low-code** : ces plateformes offrent des outils visuels et des composants pré-conçus pour créer des applications avec moins de code, tout en permettant aux développeurs d'ajouter des personnalisations si nécessaire.

2. **No-code** : ces plateformes permettent aux non-programmeurs de créer des applications via des interfaces graphiques simples, sans écrire de code.

Ces outils peuvent accélérer le processus de développement d'applications, réduire les coûts,

1.2.2 Technologie de transmission

Les technologies de transmission LPWAN actuelles peuvent être classées en trois catégories selon leurs besoins en infrastructure supplémentaire :

A- NB-IoT

Narrow Band Iot est une technologie LPWA qui offre une connectivité robuste pour les appareils IoT via les réseaux mobiles existants. Elle est économe en énergie, a une excellente portée, est facile à déployer, sécurisée, fiable et coûte moins cher. [34],[24]

NB-IoT utilise les réseaux 4G et 5G et gère de grands volumes de données en direction descendante. Déploiement possible en :

- **Mode stand-alone** : utilise la bande GSM (2G), moins utilisée pour les communications mobiles.
- **Bande de garde** : opère en bordure des blocs LTE avec un espacement en fréquences d'au moins 200 kHz entre le bord du canal du système à bande étroite et le bord du bloc de l'opérateur à large bande.
- **In Band** : utilise les bandes de fréquences LTE d'un opérateur pour le haut débit mobile.

NB-IoT est étulisé dans pluieurs cas tel que : Les Compteurs intelligents, économies d'énergie, conservation d'eau, surveillance des températures, villes et bâtiments intelligents et agriculture intelligente, comme cela est montré par la figure 1.3 :



FIGURE 1.3 – NB-IoT [34].

B- Sigfox

Sigfox est une technologie de réseau LPWAN (Low Power Wide Area Network) [21] utilisée pour connecter des capteurs et appareils IoT de manière sécurisée, économique et à faible coût. Créée en 2010 et adoptée par plus de 70 opérateurs dans

le monde, elle permet une communication sans fil à basse consommation d'énergie, optimisée pour les petits messages. Sigfox utilise une technologie de modulation à bande étroite pour une transmission à bas débit mais avec une longue portée, et propose une infrastructure mondiale d'antennes. Les données collectées par les dispositifs IoT sont transmises périodiquement vers le cloud Sigfox, où elles sont traitées et sécurisées, avec une connectivité accessible via des API pour des applications tierces, voir la figure 1.4.

Sigfox a été largement utilisé pour divers cas d'utilisation, tels que le suivi d'actifs, l'agriculture connectée, la gestion des déchets, la santé et le suivi des animaux, grâce à sa couverture étendue et sa faible consommation d'énergie. Cependant, il est important de noter que Sigfox n'est plus opérable, ce qui affecte la continuité des services pour les applications IoT dépendant de cette technologie. Les entreprises utilisant Sigfox doivent chercher des alternatives pour maintenir leurs solutions IoT opérationnelles.

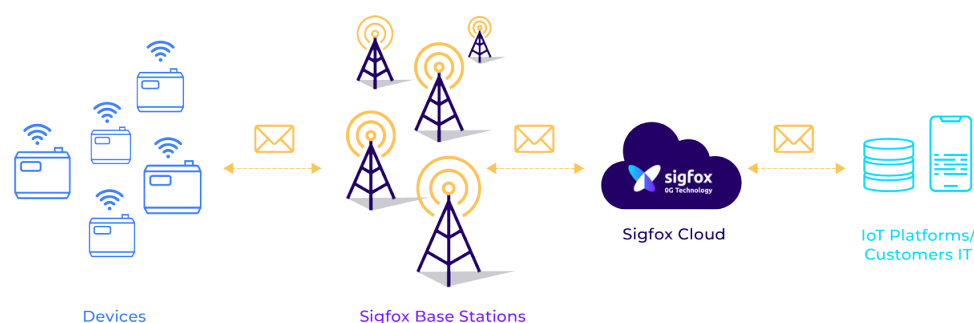


FIGURE 1.4 – Sigfox [21].

C- LoRa

L'Internet des objets utilise également la technologie LoRa. Les capteurs transmettent des données à faible débit sur de longues distances. Envoyer des données à faible débit présente l'avantage de réduire la consommation d'énergie des objets connectés, prolongeant ainsi leur durée d'utilisation [35].

- **Fréquence utilisée et débit** : LoRa a un débit de 980 bps à 21,9 kbps, inférieur à la 3G qui varie de 0,144 Mbps à 2 Mbps [30], et au WiFi qui va de 11 Mbps à 450 Mbps (2,4 GHz) et de 54 Mbps à 450 Mbps (5 GHz) [12]. LoRa utilise les bandes de fréquences ISM pour la communication des dispositifs, avec un rapport cyclique limité à 1% par heure [21]. Cela permet à un dispositif de transmettre des messages pendant 36 secondes sur une heure [22].

LoRa utilise une puissance de 20 mW pour l'émission de données [23], sur des bandes de fréquences spécifiques qui varient selon les continents [24, 11] :

- 868MHz et 433MHz pour l'Europe.
- 915MHz pour l'Amérique du Nord.
- 430MHz pour l'Asie.

1.2.3 Structure de LoRa :

pour que LoRa fonctionne efficacement, une structure appropriée a été mise en place.

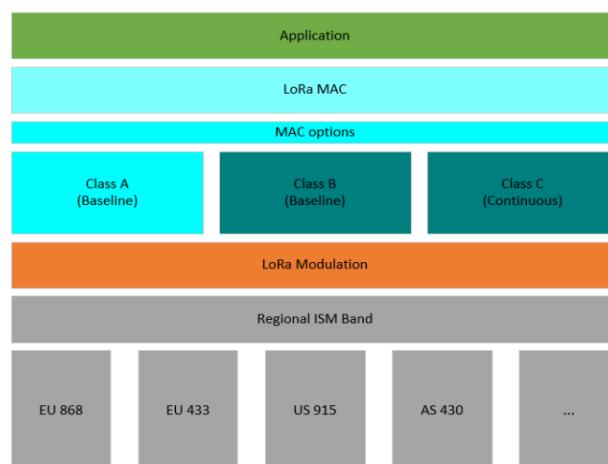


FIGURE 1.5 – Structure de LoRa [36].

La structure LoRa présentée par la figure 1.5 comprend plusieurs couches : la région d'utilisation, la modulation LoRa qui utilise le "Chirp Spread Spectrum" (CSS), la couche MAC (LoRaWAN) avec trois classes de communication, et la couche application pour le traitement des données [36].

1.2.4 Alliance LoRa :

LoRaWAN utilise la technologie LoRa pour les réseaux sans fil. Le protocole est ouvert, mais la technologie radio LoRa est propriétaire avec des brevets de Semtech Corporation. La LoRa Alliance, une organisation à but non lucratif comptant plus de 500 membres, supervise le développement de LoRaWAN depuis 2015.

1.2.5 Traitement et Stockage des Données dans LoRa :

les plateformes IoT (Cloud) gèrent et traitent les données des objets connectés pour offrir des services spécifiques, incluant la collecte, le stockage, l'analyse et l'exploitation des données, comme il est montré dans la Figure 1.6 :

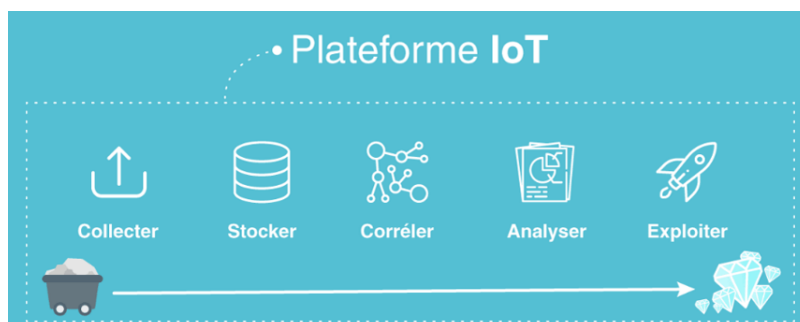


FIGURE 1.6 – Plateformes IoT.

* **Cloud/ Plateforme (informatique en nuage)** : cela permet d'accéder à des services informatiques via Internet à partir d'un fournisseur. Les services Cloud sont principalement montrés dans la figure 1.7 :

SaaS(Software as a Service) : offre des applications sur Cloud.

PaaS(Platform as a Service) : assure l'exécution des applications.

IaaS(Infrastructure as a Service) : mise en réseau, calcul et stockage.

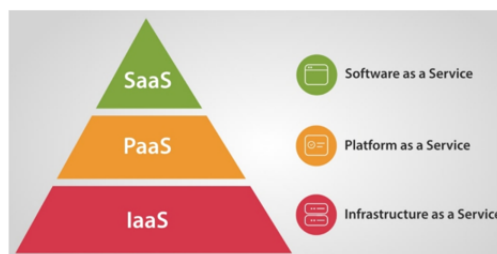


FIGURE 1.7 – Les services Cloud.

La figure 1.8 représente les trois solutions techniques sont possibles pour l'implémentation du 3^{ème} niveau :

Fog Computing : traitement décentralisé des données IoT au niveau des nœuds locaux (Fog), avant relais vers le Cloud.

Edge Computing : le traitement des données IoT se fait à l'extrémité du réseau.

Mist Computing : le traitement des données se fait localement dans le nœud capteur.

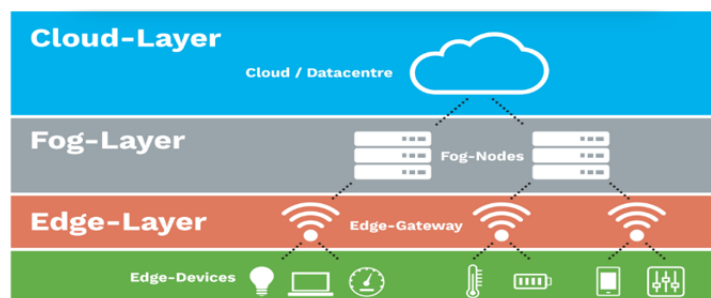


FIGURE 1.8 – Les trois solutions techniques d'IoT.

* **Type de plateforme IoT** : deux types de plateformes sont à distinguer :

- **Les plateformes propriétaires** : permet au prestataire de gérer la maintenance, la sécurité et le fonctionnement des applications.
- **plateformes open-source** : demandent plus de temps et de techniques car elles impliquent le développement de tous les services et la maintenance de l'infrastructure et des applications.

* **Investissements sur les plateformes IoT** : il existe plus de 600 plateformes IoT, ci-dessous quelques unes :

- Amazon Web Services IoT.
- IBM Watson.

- Microsoft Azure Cloud IoT.
- Google Cloud IoT.
- Oracle Integrated Cloud for IoT.
- ThingWorx Industrial.
- Cisco IoT Cloud.

1.3 Conclusion

L'Internet des objets (IoT) et les technologies de transmission à faible débit et longue distance jouent un rôle essentiel dans la transformation numérique actuelle. L'IoT facilite la connexion et la communication entre objets physiques, avec des avantages significatifs dans des secteurs variés tels que la santé, l'agriculture, l'industrie et la domotique. Ces technologies permettent une communication efficace et fiable, tout en réduisant la consommation d'énergie et en offrant une portée étendue. Ensemble, elles ouvrent la voie à des environnements intelligents et interconnectés, améliorant ainsi notre qualité de vie, l'efficacité des entreprises et contribuant à un avenir durable et innovant.

2

Technologie LoRaWAN

Introduction

Les réseaux LoRaWAN reposent sur une pile protocolaire composée de trois couches essentielles. Tout d'abord, la couche physique utilise la modulation LoRa pour permettre une communication sans fil à longue portée avec une faible consommation d'énergie. Ensuite, le protocole LoRaWAN constitue la couche d'accès au médium (MAC), standardisé et maintenu par la LoRa Alliance, qui assure des communications sécurisées, bidirectionnelles, ainsi que des fonctionnalités de mobilité et de localisation. Enfin, la couche applicative gère l'envoi et la réception des données entre les dispositifs IoT et les applications finales, permettant ainsi de développer une large gamme de solutions intelligentes et connectées.

Nous avons choisi la technologie LoRaWAN en raison de sa capacité à offrir une couverture étendue avec une faible consommation d'énergie, ce qui est idéal pour les applications IoT nécessitant une longue autonomie des dispositifs et une communication fiable sur de longues distances.

2.1 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 voir la figure 2.1. Il apporte un ensemble de mécanismes pour gérer la communication entre les terminaux et les passerelles.

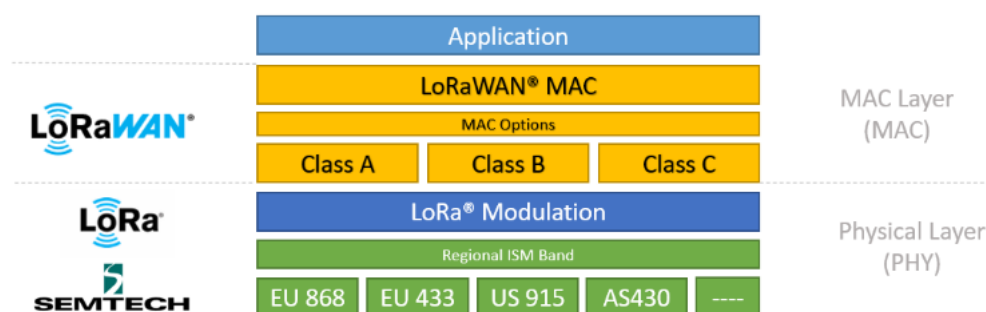


FIGURE 2.1 – Pile Technologique des Réseaux LoRaWAN [35].

Versions du protocole : l'un des principaux rôles de la LoRa Alliance est la spécification et l'évolution du protocole LoRaWAN. Voici les différentes versions ainsi que leur évolution [35].

- **Versión 1.0.0** (janvier 2015) : version initiale.
- **Versión 1.0.1** (février 2016) : ajout d'un nouveau plan de fréquences pour la Chine
- **Versión 1.0.2** (juillet 2016) : les sections sur la couche physique sont maintenant dans un document appelé "LoRaWAN Regional Parameters". Première version stable.
- **Versión 1.1** (octobre 2017) : amélioration de la sécurité et de l'itinérance. Ajout de nouvelles "Root Keys" et "Sessions Keys". Nouveaux "Frame Counters" et nouvelles commandes MAC. JoinEUI remplace AppEUI. Clarification sur la classe B et la classe C.
- **Versión 1.0.3** (juillet 2018) : version 1.0.3 = version 1.0.2 + section Classe B de la version 1.1.
- **Versión 1.0.4** (octobre 2020) : AppEUI devient JoinEUI. Nombreuses clarifications. Dernière version 1.0.x.

On constate que la version 1.1 a été publiée, mais elle n'a pas été adoptée. La LoRa Alliance a ensuite continué la clarification de la version 1.0.x en ajoutant la version 1.0.3 et la version 1.0.4 qui devrait être la dernière de la série 1.0.x.

2.2 LoRa vs LoRaWAN

La différence entre LoRa et LoRaWAN réside dans le fait qu'ils constituent deux éléments essentiels d'un système permettant une communication longue portée et à faible consommation d'énergie pour les appareils connectés. Voici les principales différences :

2.2.1 LoRa

- Technologie de la couche physique, comparable au matériel radio.
- Signifie "Long Range" et désigne une technique de modulation spécifique permettant une communication longue portée avec une faible consommation d'énergie.
- Définit la manière dont les appareils communiquent entre eux et avec les passerelles sur le réseau.

2.2.2 LoRaWAN

- Protocole de la couche réseau, comparable au langage utilisé par les radios.
- Offre des fonctionnalités telles que la sécurité et la communication bidirectionnelle.

LoRa et LoRaWAN sont des technologies prometteuses pour l'Internet des objets (IoT), en particulier pour les applications nécessitant une communication longue portée, peu gourmande en énergie et à faible débit de données. Le tableau 2.2 montre quelques différences entre les deux technologies :

Fonctionnalité	LoRa	LoRaWAN
Type	Technologie	Protocole
Fonction	Permet une communication longue portée	Définit les règles de communication
Transfert de données	Transporte les données	Gère l'envoi et la réception des données
Consommation d'énergie	Faible consommation	Ajoute une surcharge minimale

TABLE 2.1 – Différences entre LoRa et LoRaWAN.

2.3 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 [12] : les terminaux, les passerelles, les serveurs réseaux et les serveurs d'applications.

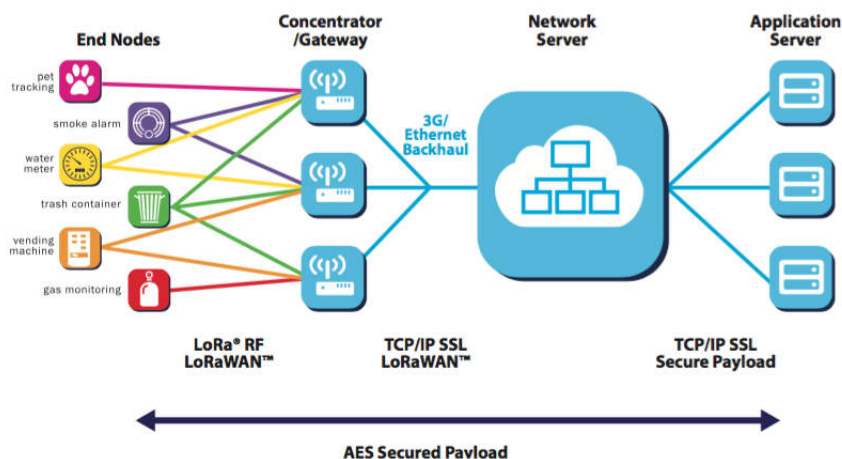


FIGURE 2.2 – Architecture du Réseau LoRaWAN [12].

- **Terminaux** : les dispositifs sont équipés de modules LoRa pour envoyer et recevoir des messages sans fil via des passerelles LoRa.

- **Passerelles** : les passerelles agissent comme des relais entre les terminaux et le réseau, utilisant un concentrateur LoRa pour recevoir et transmettre les trames vers le Network Server. Les trames des terminaux LoRaWAN sont reçues par toutes les passerelles à proximité, qui les relaient au Network Server correspondant.
- **Network Server (NS)** : le Network Server gère un réseau LoRaWAN en filtrant les trames, en assurant la sécurité et en routant les données vers les Application Servers.
- **Application Server (AS)** : il traite les données des terminaux et envoie les informations aux terminaux via le Network Server. Plusieurs Application Servers peuvent être utilisés pour analyser les données à l'aide de technologies comme l'apprentissage automatique et l'intelligence artificielle.

2.4 LoRa

LoRa utilise la modulation Chirp Spread Spectrum (CSS) pour transmettre des signaux sur de grandes distances, avec un facteur d'étalement (SF) variant de 7 à 12, influençant la portée et la quantité de données transmises. CSS [36] permet d'ajuster la fréquence du signal dans le temps, réduisant les effets Doppler (changement fréquentiel) [37] et les pertes de données, tout en optimisant la consommation énergétique des dispositifs IoT [35]. Les nœuds LoRa peuvent être configurés avec différents paramètres SF (Spectre Factor), BW (Largeur de Band), CR (Code Rate), TP (Puissance de Transmission) pour diverses combinaisons de performances, affectant le débit, le temps de transmission, et la consommation d'énergie [26]. LoRaWAN inclut trois classes de communication qui définissent comment les transmissions et réceptions sont gérées, renforçant la sécurité des données et permettant des transmissions simultanées sur différents canaux grâce à des spectres de fréquences distincts [7].

2.5 Classes de terminaux

La spécification LoRaWAN [22] [24] définit trois classes de terminaux : A, B et C. Tous les terminaux doivent mettre en œuvre la classe A pour la communication bidirectionnelle, tandis que les classes B et C sont optionnelles et offrent des fonctionnalités supplémentaires.

2.5.1 Classe A

Les terminaux LoRaWAN doivent supporter la classe A pour la communication bidirectionnelle. Après une transmission montante initiée par le terminal, deux fenêtres de réception (RX1 et RX2) sont ouvertes pour écouter la liaison descendante. Le serveur peut répondre pendant RX1 ou RX2, mais pas les deux. Si aucune réponse n'est reçue, la prochaine transmission descendante est retardée jusqu'à la prochaine transmission montante comme indiqué dans la figure 2.3 :

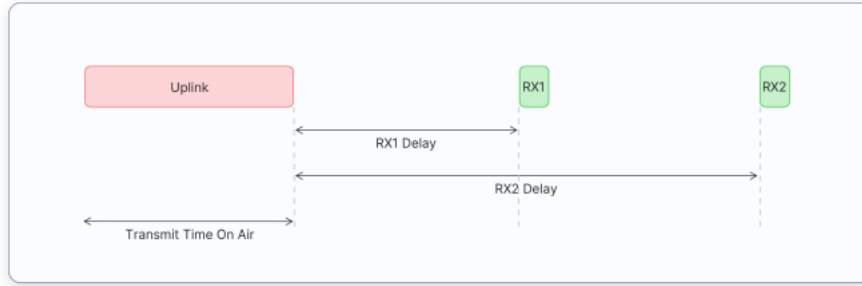


FIGURE 2.3 – Fenêtres de réception de la Classe A

Les terminaux de classe A, alimentés par batterie, ont une consommation minimale d'énergie et une latence élevée en liaison descendante, nécessitant une transmission préalable sur la liaison montante pour recevoir des données.

2.5.2 Classe B

Les terminaux de classe B ouvrent des fenêtres de réception synchronisées pour recevoir les messages descendantes du serveur réseau, utilisant des balises transmises par les passerelles. Ces fenêtres sont programmées à intervalles réguliers appelés périodes de balise. En plus, ils ouvrent des "slots ping" pour la réception de messages à des moments spécifiques après avoir envoyé des trames en liaison montante, comme indiqué dans la figure 2.4 :

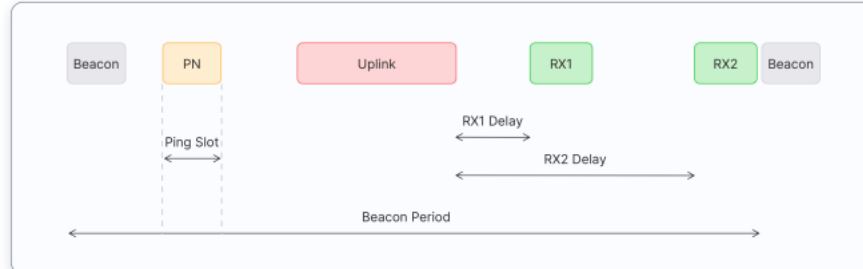


FIGURE 2.4 – Fenêtres de réception de la Classe B

2.5.3 Classe C

Les terminaux de classe C maintiennent leurs fenêtres de réception ouvertes en permanence, sauf pendant la transmission, offrant ainsi une communication à faible latence mais consommant plus d'énergie que les terminaux de classe A, comme illustré dans la figure 2.5 :

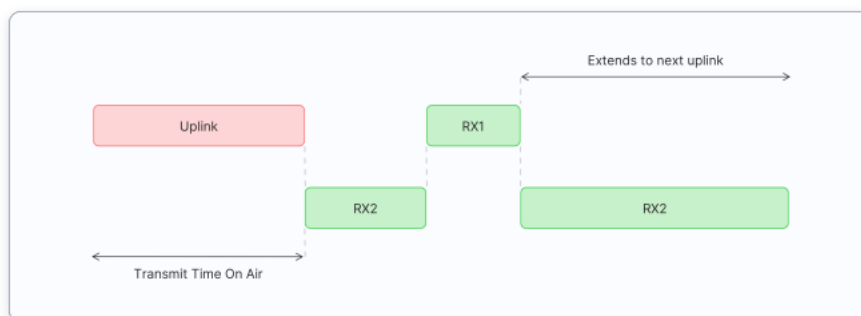


FIGURE 2.5 – Fenêtres de réception de la Classe C

D'après les figures précédentes, nous pouvons remarquer que :

- un terminal de classe B est également un terminal de classe A (les emplacements RX1 et RX2 sont toujours présents).
- un terminal de classe C est également un terminal de classe A (les emplacements RX1 et RX2 sont toujours présents).

La classe B et la classe C sont donc une extension de la classe A. Concernant leur consommation, nous pouvons les représenter dans la figure 2.6 :

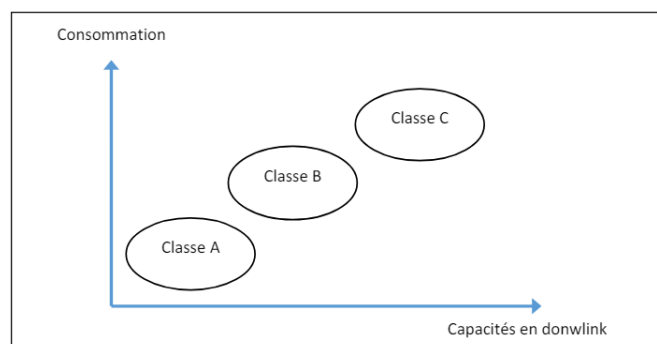


FIGURE 2.6 – Consommation d'énergie et capacité en downlink (débit).

2.6 LoRa MAC

Les protocoles d'accès dans LoRa MAC visent à optimiser l'utilisation du spectre radio pour assurer une communication fiable et efficace entre les dispositifs LoRa et les passerelles, regroupés en trois catégories principales[19].

- **La première catégorie** : est les protocoles d'accès aléatoire dans lesquels tous les nœuds ont la même supériorité sans temps fixe pour l'envoi de données comme le pur ALOHA et Carrier Sense Multiple Access (CSMA).

- **La deuxième catégorie** : est les protocoles d'accès contrôlés dans lesquels les données peuvent être envoyées par ce nœud qui est approuvé par tous les autres nœuds comme la Réserveation.

- **La troisième catégorie** : le protocole de canalisation partage la bande passante du lien de communication en fréquence, temps et code pour permettre à plusieurs nœuds d'y accéder simultanément, incluant TDMA(Time Division Multiple Access), FDMA(Frequency Division Multiple Access) et CDMA(Code Division Multiple Access) [15].

2.6.1 Protocoles d'accès dans LoRa MAC :

Les trafics aléatoires des terminaux informatiques nécessitent des protocoles d'accès au réseau, dont certains dérivent des méthodes des réseaux locaux fixes, on distingue [25] :

- Les protocoles à accès aléatoires (ALOHA, S-ALOHA, CSMA).
- Les protocoles à accès contrôle centralisé.
- Les protocoles à accès contrôle distribué (jeton, ...).
- Les protocoles d'accès adaptatifs (ADR).

Les protocoles MAC aléatoires permettent l'émission à tout moment, avec retransmission après collision. Dans ces protocoles il existe deux principales méthodes : sans écoute de la porteuse : ALOHA et Avec écoute de la porteuse : CSMA.

1- Méthode ALOHA : en 1970, les chercheurs de l'université de Hawaï ont imaginé une méthode simple d'accès au canal partagé. Le protocole ALOHA a été conçu en 1972 pour des transmissions radio terrestres mais peut s'appliquer à d'autres types de supports. Ce protocole a deux versions :

- ALOHA pur.
- ALOHA discrétisé.

- **ALOHA PUR :** dans cette méthode, une station transmet immédiatement et, en cas de collision, attend un ACK. Sans ACK, elle retransmet après un délai aléatoire. Cette méthode utilise 18% du débit total au mieux, comme la montre la figure 2.7 :

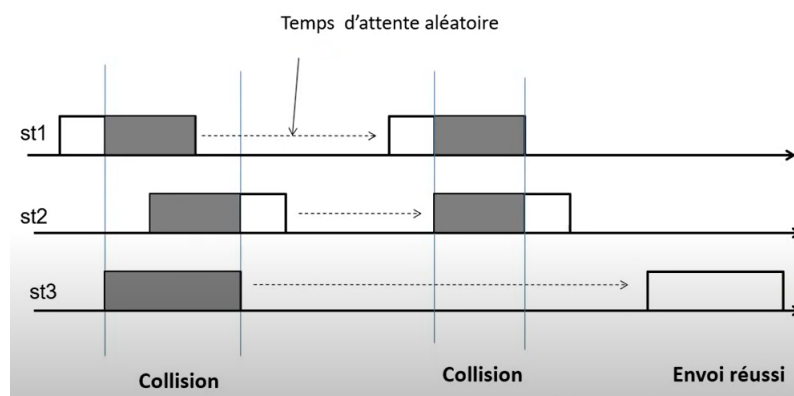


FIGURE 2.7 – Exemple sur ALOHA PUR

- **Slotted-ALOHA (ALOHA discrétisé) :** dans cette méthode, le temps est découpé en intervalles de durée T , avec des horloges synchronisées. La figure 2.8 montre que les trames sont transmises au début de chaque intervalle, doublant l'utilisation du canal à 37% [13].

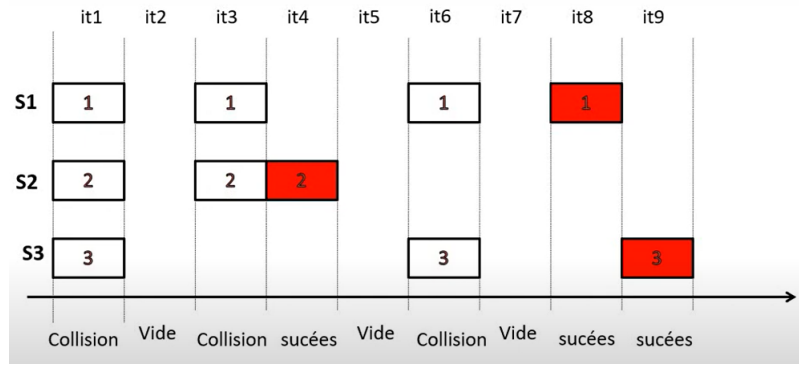


FIGURE 2.8 – Exemple S-ALOHA

Dans cette technique il y a des intervalles vides bien que les stations veulent émettre.

2- Protocole CSMA (Carrier Sense Multiple Access) : le protocole CSMA "écouter avant d'émettre" transmet si le canal est libre, sinon il diffère. Il réduit les conflits en retardant l'émission lorsque le canal est occupé. Après une collision, les stations attendent un temps aléatoire avant de réécouter voir la figure 2.9 :

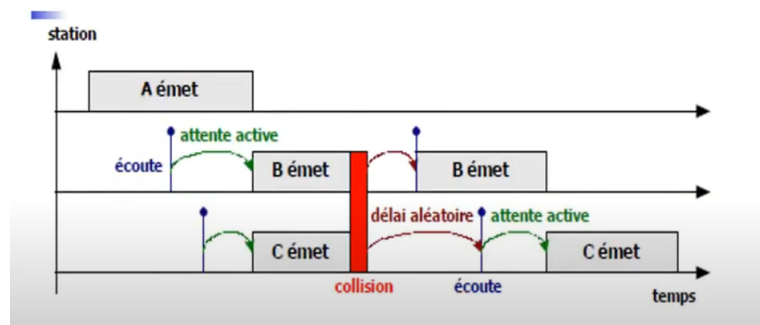


FIGURE 2.9 – Exemple CSMA

Nous allons présenté deux variantes de CSMA [29] :

- **CSMA persistante (1-persistent) :** comme le montre la figure suivante 2.10 :

- Tant que le canal est occupé continuer l'écoute .
- Dès que le canal est libre alors envoyer les données .

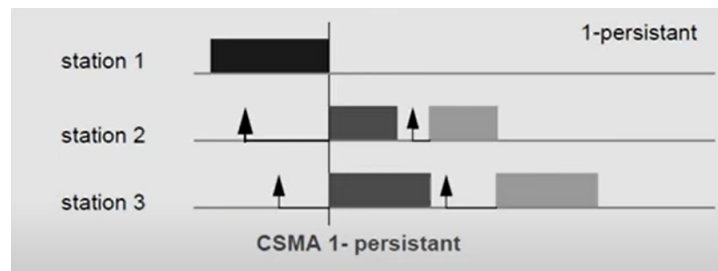


FIGURE 2.10 – CSMA 1-persistent

- **CSMA non persistante** : comme la montre la figure suivante 2.11 :
 - Une station peut ne pas rester tout le temps à l'écoute de la porteuse.
 - Si le canal est occupé alors attendre une durée aléatoire avant d'écouter une autre fois.

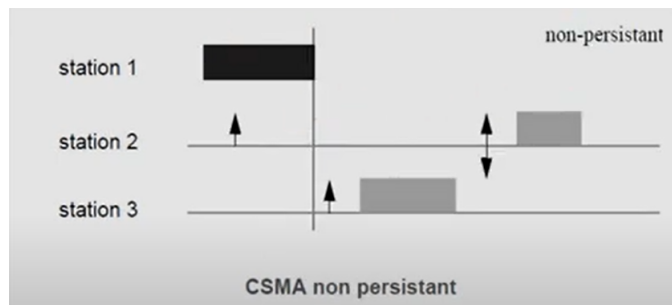


FIGURE 2.11 – CSMA non persistant

3- Adaptive Data Rate (ADR) : LoRaWAN utilise l'ADR pour ajuster dynamiquement le débit de données et la puissance des transmissions en fonction des conditions du canal et de la distance. Cela optimise la consommation d'énergie et la portée, en utilisant des facteurs d'étalement du spectre (SF) de SF7 à SF12 pour réguler le débit et la portée. Toutefois, l'utilisation de l'ADR doit être prudente pour éviter une consommation d'énergie excessive et une réduction de la portée[5].

2.7 Types de trames utilisées par LoRaWAN

La technologie LoRa envoie les données par des trames structurées. Une trame est composée de plusieurs parties voir la figure 2.12 :

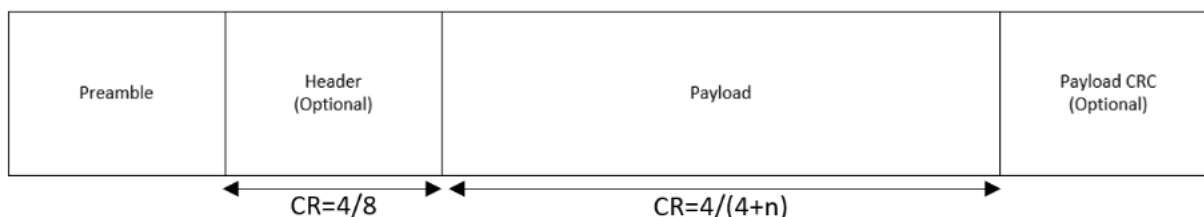


FIGURE 2.12 – Trame de LoRa

La trame LoRa commence par un préambule qui synchronise le dispositif avec une passerelle. L'en-tête, défini par un Code Rate (CR), spécifie le nombre d'octets dans la trame, indique si des données sont reçues ou envoyées, et précise la présence d'un CRC (Cyclic Redundancy Check) pour la vérification des erreurs. Les trames ont une taille comprise entre 51 et 222 octets. Le CR détermine la quantité de bits utiles à transmettre et inclut des bits supplémentaires pour assurer une communication fiable sans perte de données.[26]. L'encodage du paramètre « CR » est défini de la manière suivante :

$$Bits_{utile} = SF * CR$$

$$Bits_{correctif} = SF - Bits_{utile}$$

Le « Payload », qui signifie aussi « charge utile », contient toutes les données utiles à la trame, voir la figure 2.13. Dans la charge utile, les données présentent sont :

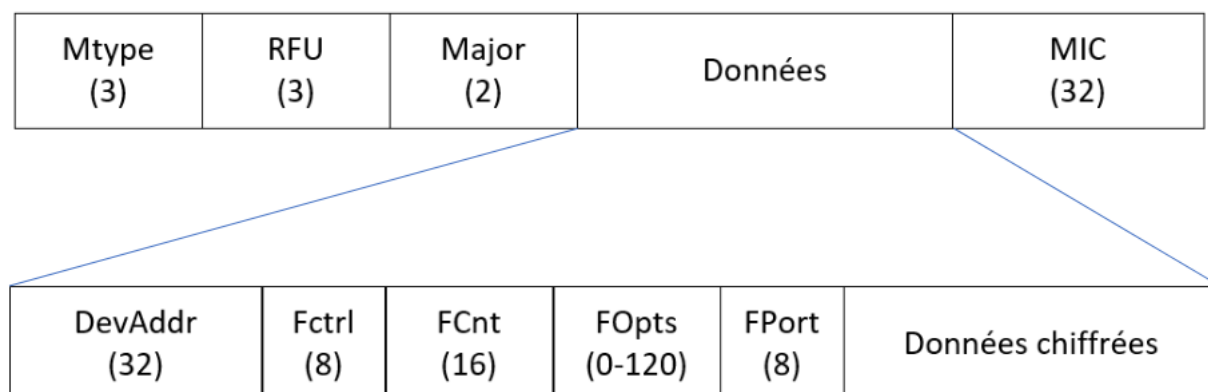


FIGURE 2.13 – Données de la charge utile

Les chiffres entre parenthèses représentent la taille en bits des différents champs. La description des différents champs [35] sont les suivant :

1. **Mtype** : permet de savoir si on reçoit une donnée sur le dispositif ou si on envoie une donnée.
2. **Major** : indique quelle version du protocole de LoRa est utilisée.
3. **MIC** : permet de savoir s'il y a eu une corruption de données dans la trame.
4. **DevAddr** : adresse du dispositif.
5. **FCtrl** : adapte le débit d'envoi ou de réception. Le champ adapte les acquittements en fonction des données reçues. Indique s'il y a des champs optionnels à la trame et la longueur du champ « FOpts ».
6. **FCnt** : compteur de répétitions de la trame envoyée.
7. **FOpts** : permet d'utiliser les commandes de la couche MAC.
8. **FPort** : contient le port de l'application où les données vont être envoyées ou reçues.
9. **RFU** : indique que ces bits ne sont pas actuellement utilisés mais sont réservés pour des fonctionnalités ou des extensions futures.

2.8 Connexion au réseaux LoRaWAN

LoRaWAN assure des transmissions sécurisées en activant les terminaux pour les associer au réseau. Chaque nœud doit être authentifié avec le DevAddr, la NwkSKey pour l'authentification, et la AppSKey pour le chiffrement. Deux méthodes sont utilisées : ABP (Activation par Personnalisation) et OTAA (Activation par Voie Hertzienne).

1- ABP (Activation By Personalization) :

En ABP (Activation par Personnalisation), les informations nécessaires à la communication (DevAddr, NwkSKey, AppSKey) sont préconfigurées et identiques dans

le dispositif, le Network Server et l'Application Server. C'est la méthode la plus simple souvent utilisée pour tester un prototype et établir une communication LoRaWAN, voir la figure 2.14 :

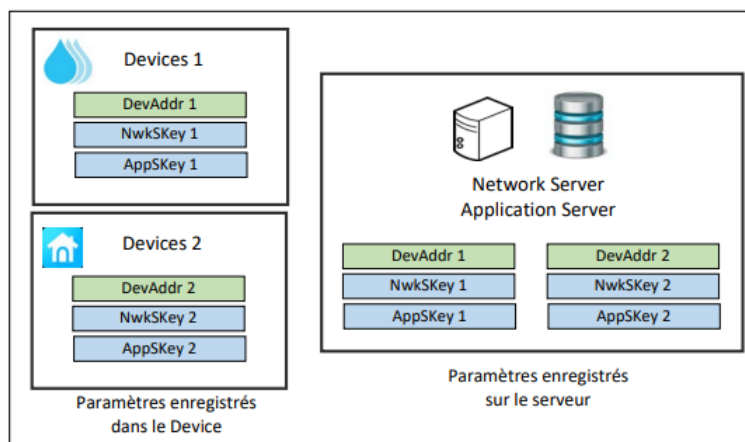


FIGURE 2.14 – DevAddr, NwkSKey et AppSKey en ABP.

Dès que le Device démarre, il peut envoyer et recevoir des messages LoRaWAN.

2- OTAA (Over The Air Activation) :

en mode OTAA, le DevAddr, l'AppSKey et le NwkSKey sont générés lors de la procédure de Join. Pour cette procédure, le dispositif LoRaWAN doit être configuré avec les éléments suivants :DevEUI,AppEUI/JoinEUI et AppKey.

Le Network Server doit posséder les mêmes DevEUI, AppEUI/JoinEUI et AppKey. L'objectif principal du Join-Request est d'établir la configuration finale des deux côtés : DevAddr, NwkSKey et AppSKey.

N.B : tous les éléments nommés "EUI" sont toujours uniques et ont une taille de 8 octets. Les clés sont toujours sur 128 bits (16 octets).

Pendant la procédure de Join, les mêmes paramètres sont générés de part et d'autre : DevAddr,NwkSKey et AppSKey, voir les figures 2.15 et 2.16 :

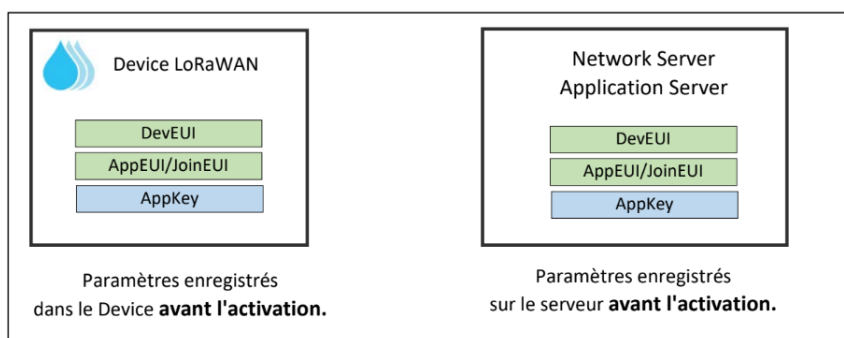


FIGURE 2.15 – DevAddr, NwkSKey et AppSKey en OTAA.

Avant d'effectuer le Join-Request, la configuration initiale stockée dans le dispositif et sur le serveur inclut les éléments suivants :

- * **DevEUI** : le DevEUI est l'identifiant unique du dispositif LoRaWAN, similaire à une adresse MAC en Ethernet.
- * **AppKey** : la clé AES 128 est utilisée pour l'authentification du Join-Request,

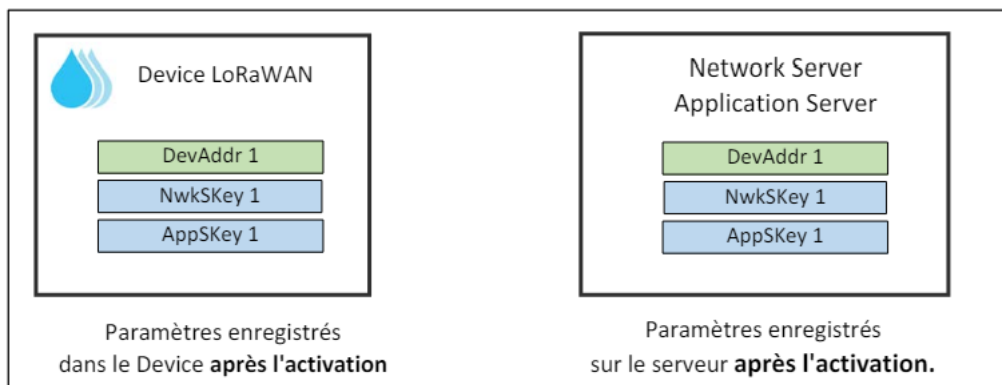


FIGURE 2.16 – DevAddr, NwkSKey et AppSKey en OTAA.

le chiffrement du Join-Accept, et la génération des clés de session. Elle doit rester strictement confidentielle.

- * **AppEUI/JoinEUI** : dans les versions LoRaWAN 1.0.3 et antérieures, ce paramètre était l'AppEUI. À partir de LoRaWAN 1.0.4, il est renommé en JoinEUI pour identifier le Join Server [22].

Pour rappel, la configuration finale après la procédure de Join a les objectifs suivants :

- Utilisée pour l'authentification avec le Network Server.
- Utilisée pour le chiffrement des données avec l'Application Server.
- Identifiant de 32 bits au sein d'un réseau LoRaWAN.

qui sont représentés par la figure 2.17 :

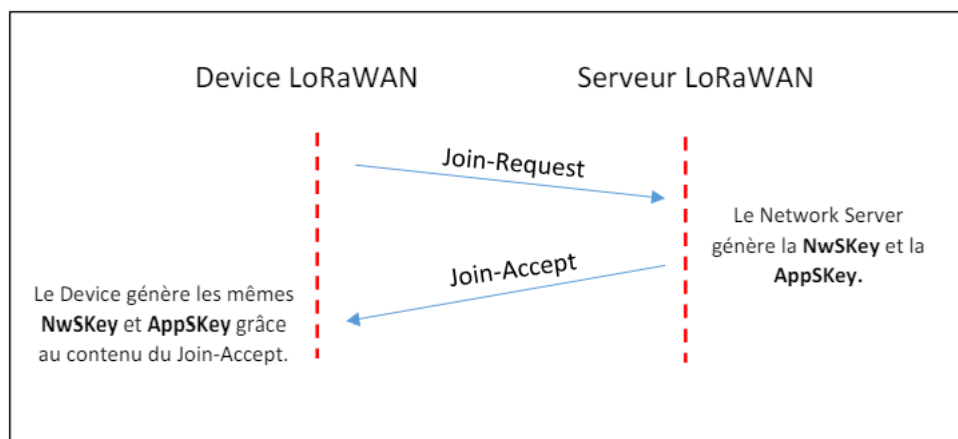


FIGURE 2.17 – DevAddr, NwkSKey et AppSKey en OTAA.

En utilisant OTAA, les terminaux reçoivent des clés racines et obtiennent un DevAddr dynamique à chaque session. En revanche, avec ABP, les terminaux utilisent un DevAddr fixe et des clés de session statiques, ce qui simplifie la configuration mais limite la sécurité et l'itinérance.

Le tableau 2.2 montre une petite comparaison entre les méthodes d'activation ABP et OTAA.

	ABP	OTAA
Sécurité globale	Moins sécurisé que l'OTAA	Plus de sécurité
Gestion du Frame Counter	- La sauvegarde en mémoire non volatile est obligatoire. - Possibilité de désactiver le contrôle du Frame Counter (faille de sécurité)	Supporté par l'OTAA
Changement de réseau	Complicé et non sécurisé	Supporté par l'OTAA avec un Join Server
Modification RX Delay DLSettings	Gestion par "MAC commands"	Supporté par l'OTAA
Ajout de canaux	Gestion par "MAC commands"	Supporté par l'OTAA

TABLE 2.2 – Comparaison des méthodes d'activation ABP et OTAA

2.9 Couverture dans les réseaux LoRaWAN

L'itinérance LoRaWAN combine les zones de couverture de réseaux différents pour étendre la connectivité des terminaux et des passerelles, avec une fonctionnalité de "densification du réseau" pour une couverture étendue et une meilleure mobilité des terminaux.

LoRaWAN autorise deux types d'itinérance [18] : l'itinérance passive et l'itinérance active (ou itinérance handover).

- **L'itinérance passive** : l'itinérance passive redirige les trames du Forwarding Network Server (fNS) vers le Home Network Server (hNS) sans que les terminaux en soient conscients, et ne nécessite pas de RejoinRequest.[22], [23].

- A l'inverse, LoRaWAN v1.1 [24] a introduit l'itinérance handover en plus de l'itinérance passive. Cela permet au Serving Network Server (sNS) de recevoir et traiter les trames des terminaux avant de les transférer au Home Network Server (hNS), nécessitant ainsi une RejoinRequest.

2.10 Relation entre SF, débit et latence :

Voici la relation entre le Spreading Factor (SF), le débit, et la latence dans un réseau LoRaWAN : LoRaWAN Regional Parameters Semtech LoRa Modulation Basics

- **Spreading Factor (SF)** : paramètre de modulation qui influence la portée et la robustesse de la communication, il varie entre 7 et 12. Plus le SF est élevé, plus la portée est grande mais le débit est réduit.
- **Débit (kbps)** : vitesse à laquelle les données sont transmises. Un SF plus élevé réduit le débit.
- **Latence (ms)** : temps de retard dans la transmission des données. Un SF plus élevé augmente la latence en raison du temps de transmission plus long.

Ce tableau 2.3 montre que, dans un réseau LoRaWAN, il y a un compromis entre le débit et la latence en fonction du Spreading Factor utilisé [3] [8].

SF	Débit	Latence
SF7	5.47	Très faible
SF8	3.13	Faible
SF9	1.76	Modérée
SF10	0.98	Élevée
SF11	0.54	Très élevée
SF12	0.29	Très très élevée

TABLE 2.3 – Comparaison des méthodes d’activation ABP et OTAA

2.11 Applications de technologie LoRaWAN

LoRaWAN trouve de nombreuses applications diverses dans différents secteurs d’activité voir la figure 2.18 :

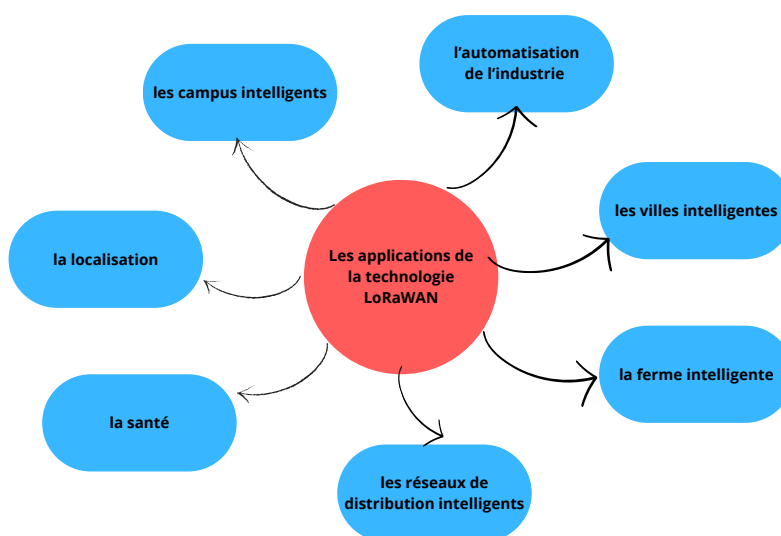


FIGURE 2.18 – Applications de la technologie LoRaWAN

- **Villes intelligentes** : les villes intelligentes utilisent LoRaWAN pour connecter des capteurs et gérer efficacement les ressources urbaines, améliorant ainsi la qualité de vie.
- **Ferme intelligente** : LoRaWAN révolutionne l’agriculture avec une surveillance et un contrôle à distance efficaces des cultures, de l’irrigation et du bétail.
- **Santé** : LoRaWAN améliore la santé par le suivi à distance des maladies chroniques et la surveillance des activités physiques.
- **Localisation** : LoRaWAN est utilisé pour localiser des objets sur de longues distances, même dans des environnements difficiles.
- **Campus intelligents** : LoRaWAN améliore les campus intelligents avec des réseaux de capteurs efficaces et sécurisés.
- **Réseaux de distribution intelligents** : LoRaWAN optimise les réseaux de distribution d’énergie en permettant une gestion et une surveillance en temps réel, réduisant ainsi les pertes et améliorant l’efficacité énergétique.

- **Automatisation de l'industrie** : LoRaWAN est utilisé dans l'automatisation industrielle pour assurer une communication fiable entre les machines et les systèmes de gestion, augmentant ainsi la productivité et réduisant les coûts de maintenance.

2.12 Défis de LoRa

LoRa et LoRaWAN sont des technologies sans fil longue portée et basse consommation pour les applications IoT industrielles [28]. Malgré ses avantages LoRa a aussi des défis comme :

- **Interférences** : LoRa est résistant aux perturbations mais peut être affecté par d'autres appareils utilisant les mêmes fréquences.
- **Portée** : la portée de LoRa dépend du terrain et du matériel, malgré sa capacité à couvrir de grandes distances.
- **Débits de données** : faible débit et données limitées, LoRa est optimal malgré une portée réduite à 2,4 GHz avec des débits plus élevés.
- **Coût** : le coût des dispositifs LoRa et des infrastructures réseau peut être limitant pour certaines applications.
- **Sécurité** : la sécurité des communications LoRa est cruciale pour les données sensibles.
- **Interopérabilité** : l'interopérabilité entre réseaux LoRa de différents fournisseurs reste un défi.

En dépit de ces défis, LoRa reste une technologie prometteuse pour l'IoT en raison de sa longue portée, de sa faible consommation d'énergie et de sa capacité à connecter un grand nombre d'appareils sur de vastes zones géographiques.

2.13 Conclusion

En résumé, LoRa et LoRaWAN sont des technologies essentielles pour l'IoT, offrant une connectivité longue portée et à faible consommation d'énergie. LoRa permet des communications robustes sur de longues distances, tandis que LoRaWAN gère efficacement les réseaux IoT basés sur LoRa.

Elles peuvent transformer des secteurs comme l'agriculture, la logistique, les villes intelligentes et l'industrie, malgré des défis en matière de sécurité des données, de gestion de réseau et d'interopérabilité. Leur popularité croissante et leurs améliorations continues les placent en position clé pour l'avenir de l'IoT.

3

Etude de l'état de l'art

3.1 Introduction

Les solutions innovantes pour optimiser les réseaux LoRaWAN portent essentiellement sur deux axes principaux : l'amélioration des protocoles d'accès et le placement stratégique des gateways. Ces efforts sont cruciaux pour augmenter l'efficacité, la couverture, et la fiabilité des réseaux IoT utilisant la technologie LoRaWAN.

Concernant les protocoles d'accès, l'utilisation du protocole ALOHA, un système d'accès aléatoire simple, montre des avantages en termes de simplicité mais présente des défis en cas de congestion du réseau. Pour surmonter ces limitations, l'intégration de techniques de Machine Learning (ML) propose une voie prometteuse. Le ML peut optimiser dynamiquement les protocoles d'accès, améliorant ainsi la gestion des collisions et l'efficacité générale du réseau en ajustant les paramètres de communication selon les conditions en temps réel.

D'autre part, le placement des gateways est également fondamental pour maximiser la portée et la qualité de service du réseau LoRaWAN. Le choix du Facteur d'Étalement (SF) est essentiel, car il influence directement la portée du signal et la capacité du réseau. Un placement judicieux des gateways, prenant en compte les valeurs optimales de SF, peut considérablement étendre la couverture et minimiser les interférences, garantissant ainsi une meilleure connectivité et fiabilité.

3.2 Type de solutions proposées pour LoRaWAN

Les solutions proposées pour LoRaWAN sont variées et dépendent des besoins spécifiques de chaque application.

Dans les articles étudiés, les types de solutions proposées pour LoRaWAN sont basés sur :

3.2.1 ALOHA

Les articles basés sur le protocole Aloha couvrent un large éventail de sujets, allant des études de performances et des améliorations de ce protocole classique

à ses applications dans des domaines spécifiques tels que les réseaux de capteurs sans fil et les communications par satellite. Ces articles explorent souvent les défis uniques posés par Aloha, tels que la gestion des collisions et l'efficacité du canal, tout en proposant des solutions novatrices pour optimiser ses performances. De plus, certains articles utilisent des simulations ou des modèles mathématiques pour étudier le comportement d'Aloha dans des scénarios variés, offrant ainsi des perspectives précieuses sur son utilisation dans différents contextes de réseau. Enfin, les comparaisons avec d'autres protocoles de communication sans fil aident à évaluer l'efficacité relative d'Aloha et à identifier ses forces et ses faiblesses dans des situations spécifiques.

Parmi les articles basés sur ALOHA on a étudié :

Performance of Unslotted ALOHA with Capture and Multiple Collisions in LoRaWAN

Dans cet article ils ont fait l'étude sur les performances de unslotted ALOHA où ils ont posé les problèmes suivants [16] :

- Surcharge de canal.
- Envoi types de paquets non confirmés.
- Perte de données.

En modélisant avec le modèle ALOHA qui concerne uniquement l'augmentation de la portée de communication.

Ils ont ensuite proposé un nouveau modèle d'ALOHA avec capture et collision multiples dans LoRaWAN. Tout les noeuds utilisent le même taux de codage et les mécanismes de correction d'erreurs, comme la figure 3.1 :

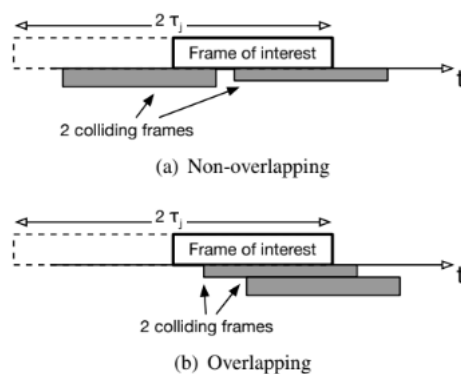


FIGURE 3.1 – Collision avec deux autres trames : scénarios de chevauchement et de non-chevauchement .

Où ils ont proposé les solutions ci-dessous :

- Déployer les passerelles loin les unes des autres.
- Utiliser une modulation plus robuste pour les noeuds.
- Avoir plus de passerelles dans les zones avec une densité plus élevée de noeuds.

LoRa beyond ALOHA : An Investigation of Alternative Random Access Protocols

Dans cet article [20] ils ont fait l'étude sur les protocoles d'accès où ils ont posé les problèmes suivants :

- Interférence causée par les transmissions simultanées d'un grand nombre d'appareils connectés.
- ALOHA souffre d'une évolutivité limitée (nombre maximal de dispositifs servis).
- Absence de garantie de Qualité de service dans ALOHA.

Ils ont proposé les solutions suivantes :

- Introduire un soutien QoS dans LoRa grâce à l'utilisation de mécanismes de planification pour les transmissions de données déterministes dans le cas des utilisations industrielles critiques du temps.
- Renforcement de l'évolutivité de LoRa par biais d'accès aléatoire (ALOHA/CSMA).

En se basant sur les mécanismes cités ci-dessous :

- Pure-ALOHA.
- Slotted-ALOHA (en boucle).
- Accès multiple sans porte-sens persistant (NP-CSMA).

Modèle d'Interférence entre EDs :

Le modèle étudié évalue l'impact des interférences entre les appareils de détection d'énergie (ED) dans un réseau sans fil, en se focalisant sur la distribution des EDs autour d'un transmetteur central. La variable clé H , représente la fraction des EDs dans un anneau autour du transmetteur, influençant directement le niveau d'interférence estimé. Trois scénarios principaux sont modélisés comme le montre la figure 3.2 :

$H \approx 0$: Peu d'interférence, avec une approximation simple.

$H \approx 1$: Forte interférence, approximée par une formule plus complexe.

$H \approx 0.5$: Interférences modérées dues à un équilibre entre EDs internes et externes.

Une fonction de correction $g(H)$ dans la figure 3.3 est utilisée pour ajuster les prédictions en fonction de la densité d'EDs, améliorant la précision du modèle. Les validations par simulations montrent que le modèle est fiable pour prédire et gérer les interférences dans des environnements réseau densément peuplés.

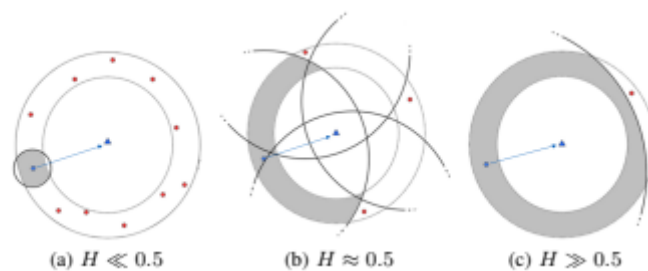


FIGURE 3.2 – Représentation de la fraction des EDs situés à l'intérieur du domaine de contention du transmetteur.

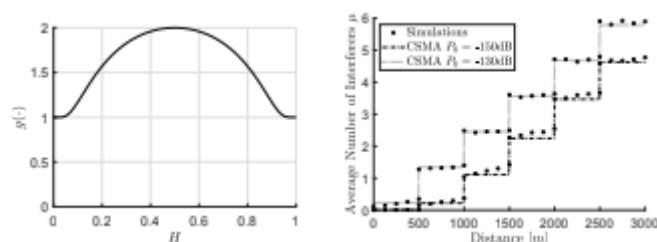


FIGURE 3.3 – Fonction de mise à l'échelle pour l'ED interférant et le nombre d'interférents.

Parmi les défis qu'ils ont rencontré :

- Erreur de synchronisation pour S-ALOHA.
- L'intensité d'interférences pour un seuil de détection sélectionnée dans CSMA.

Malgré les défis rencontrés on trouve certains points forts dont :

- Effet de capture de puissance exécutée par LoRa améliore considérablement le débit de canal de tous les mécanismes d'accès en général la densité de l'appareil.
- La possibilité pour les messages LoRa de survivre aux collisions partielles d'un message réduit le temps de vulnérabilité du message et le temps dans S-ALOHA.
- S-ALOHA offre une fiabilité plus élevée que P-ALOHA (nombre d'applications est faible).
- CSMA est plus fiable que ALOHA et plus efficace en énergie mais juste à proximité (petit SF)

3.2.2 AI/ML :

Les articles basés sur l'intelligence artificielle (IA) et l'apprentissage automatique (machine learning) constituent un domaine de recherche en plein essor, abordant une multitude de sujets allant de l'amélioration des algorithmes existants à l'exploration de nouvelles applications. Ces articles explorent souvent les avancées récentes dans les techniques d'apprentissage automatique, telles que les réseaux de neurones profonds, les algorithmes d'apprentissage par renforcement et les méthodes d'apprentissage semi-supervisé. Ils se concentrent également sur l'application de ces techniques à divers domaines tels que la vision par ordinateur, le traitement du langage naturel, la robotique, la santé et bien d'autres. De plus, certains articles examinent les implications éthiques et sociétales de l'IA et de

l'apprentissage automatique, ainsi que les défis techniques et les opportunités futures dans ce domaine en constante évolution. En résumé, les articles basés sur l'IA et le machine learning offrent un aperçu approfondi des progrès technologiques et de leur impact sur notre société.

LoRaWAN Meets ML : A Survey on Enhancing Performance with Machine Learning

Cet article est une enquête sur l'amélioration des performances de LoRaWAN avec Machine Learning (ML) [14], voir la figure 3.4 :

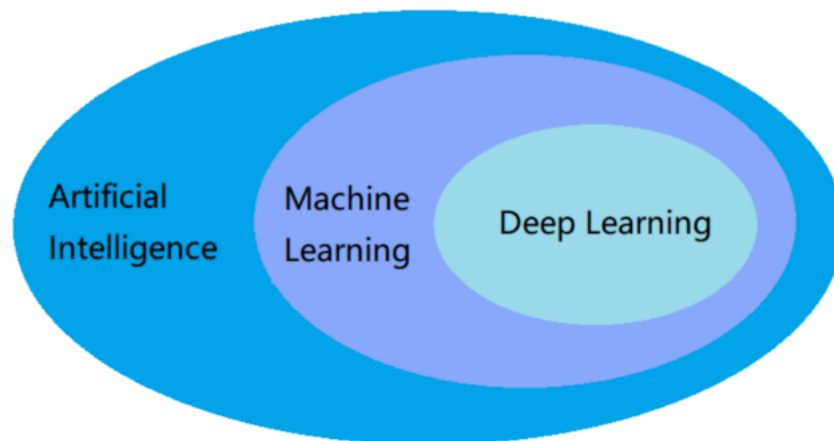


FIGURE 3.4 – La relation entre l'Intelligence Artificielle (IA), l'Apprentissage Machine (Machine Learning) et le Deep Learning

Ils ont posé les problèmes suivants :

- La configuration des paramètres LoRa.
- L'interférence et l'ADR optimisé.

Où ils ont proposé :

- Une gestion efficace des ressources (allocation de canaux, SF, TP et BW) par le biais de ML .

Les résultats obtenus sont :

- Certains documents traitent des questions de gestion des ressources .
- Technique d'optimisation et d'amélioration de l'ADR, l'atténuation, des interférences, des collisions et les simulateurs pour LoRa/LoRa WAN.
- Fournir un aperçus systématique des différents domaines de performance où ML/DL/RL ont été appliqué pour collecter des données en utilisant une méthode ML pour allocation de ressources.

La solution proposé est :

- Approche ML

3.2.3 La synchronisation Slotted :

Les articles basés sur la synchronisation slotted se concentrent sur l'amélioration des performances des protocoles de communication sans fil en introduisant

des créneaux temporels pour la transmission des données. Ces créneaux temporels, ou slots, permettent une coordination plus efficace des transmissions entre les nœuds du réseau, réduisant ainsi les collisions et améliorant la capacité du réseau. Ces articles explorent différentes techniques de synchronisation slotted, telles que l'utilisation de signaux de synchronisation périodiques ou l'allocation dynamique des créneaux en fonction de la charge du réseau. Ils évaluent également les avantages de la synchronisation slotted en termes de débit de données, de latence et de scalabilité du réseau. En outre, certains articles se penchent sur les défis associés à la synchronisation slotted, tels que la gestion de la synchronisation dans des réseaux mobiles ou la synchronisation entre les nœuds hétérogènes. En résumé, les articles basés sur la synchronisation slotted offrent des perspectives importantes pour améliorer les performances des réseaux sans fil en introduisant une coordination temporelle plus précise entre les nœuds du réseau.

Improving LoRaWAN Performance Using Reservation ALOHA

Dans cet article [17] ils ont fait l'étude sur les performances en utilisant la réservation ALOHA où ils ont posé les problèmes suivants :

- La salacité.
- Contrôle d'accès moyen à l'accès aléatoire (MAC).
- L'évolutivité du réseau.

Ils ont trouvé qu'il y a trois catégories des protocoles d'accès :

- **Protocoles d'accès aléatoires** : noeuds avec la même supériorité, pas de temps fixe pour l'envoi de données (P-ALOHA, CSMA).
- **Protocoles d'accès contrôlés** : données envoyées par noeuds approuvées par tous les autres (réservation).
- **Protocoles de canalisation** : bande passante disponible de lien de communication est partagée en fréquence, temps et en code à plusieurs noeuds pour accéder à ce canal simultanément (TDMA, FDMA et CDMA).

Les propositions données sont :

Normaliser la communication des réseaux LoRaWAN à l'aide de R-ALOHA au lieu du protocole ALOHA standard . Cette approche est basée sur un service de synchronisation distribuée qui convient aux noeuds finaux IoT à faible coût .

R-ALOHA : est un protocole qui permet à un grand nombre de station avec un trafic irrégulier de réserver des fentes pour transmettre leur cadre dans les cycles futurs . Chaque cycle a des mini-slots alloués pour faire des réservations . Les stations utilisent l'ALOHA en fente pendant les mini-slots pour demander des fentes.

Où les solutions proposées sont :

- ALOHA.
- Les performances de réseaux LoRaWAN peuvent être améliorés en utilisant l'un des protocoles de réservation (R-ALOHA).

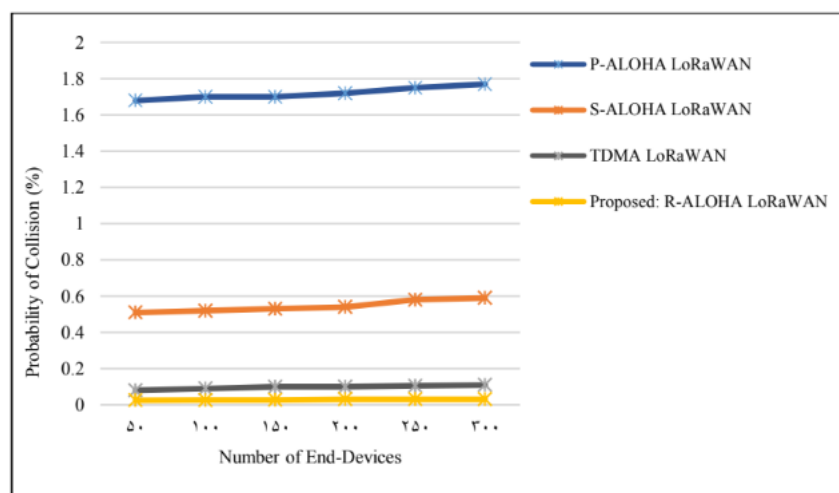


FIGURE 3.5 – Probabilité de collision par rapport au nombre d'appareils finaux.

La figure 3.5 présente la probabilité de collision en fonction de l'augmentation du nombre de dispositifs en bout de réseau. Les résultats pour les modèles montrent que le modèle de réseau LoRaWAN R-ALOHA proposé offre les meilleurs et les plus bas résultats, avec une valeur moyenne de 0.029% , comparativement aux P-ALOHA, S-ALOHA et TDMA qui présentent des valeurs moyennes de 1,7%, 0,54% et 0,09% respectivement.

D'où les résultats obtenus sont :

R-ALOHA LoRaWAN offre des meilleures performances par rapport à P-ALOHA, S-ALOHA et TDMA. Il améliore considérablement les performances du réseau concernant la probabilité de collision, le débit maximale et le cycle de tâche maximale.

Time-slotted LoRa MAC with variable payload support

Dans cet article [17] ils ont fait l'étude sur LoRa MAC chronométrique avec prise en charge de la charge utile variable où ils ont posé le problème de collision des paquets dans les récepteurs.

Leur supposition s'est basée sur la synchronisation constitue une couche LoRaWAN où les balises sont périodiquement transmises par les passerelles, voir la figure 3.6.

Ils ont utilisé les méthodes suivantes :

- Slotted-ALOHA/CSMA : pour viser à atteindre un nombre maximal de transmission de charge utile en un période de temps limité (entre balises), sur la base d'une distribution de canal complètement aléatoire.

- S-ALOHA : pour aider à éliminer les collisions partielles car il utilise des Time-Slots pour la transmission des données.
- FCA-LoRa : une meilleure performance qui utilise le mécanisme CSMA/CA.

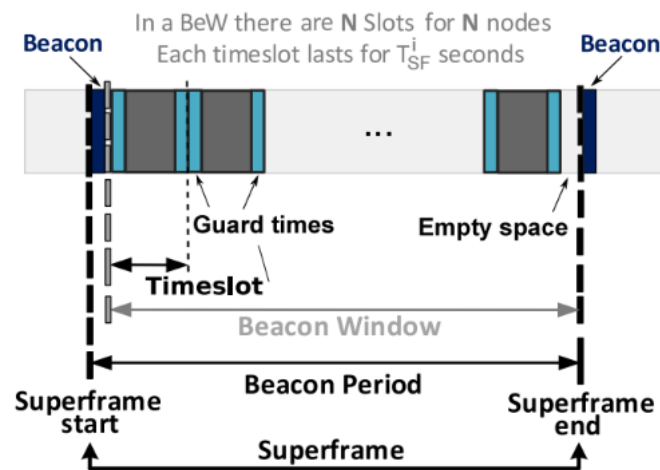


FIGURE 3.6 – Structure d'une supertrame pour la plage de charge utile i dans le schéma TS-VP-LoRa.

Où les solutions trouvées sont :

- Chaque passerelle doit avoir une taille de paquet.
- Sélectionner le canal de fréquence.

Gestion de la qualité de service (QoS) dans un réseau LoRaWAN avec mobilité

Dans cette thèse [6] ils ont fait l'étude sur la gestion de la qualité de service dans un réseau LoRaWAN avec mobilité, ils se sont basés sur le problème d'évolutivité de LoRaWAN tel que seul 120 nœuds peuvent être gérés avec une transmission de paquets toutes les 22 minutes avec configuration.

Leurs propositions se sont basées sur :

- P-ALOHA : étudier le débit de charge utile effectif lié au protocole qui diminue d'un facteur 2.22x.
- S-ALOHA : il contient plusieurs points comme :
 - Développement d'un algorithme de synchronisation fiable optimisé pour les appareils à faible puissance avec 5.37ms et le taux de réussite est de 90%
 - Ce temps est utilisé pour maintenir en alignement tous les nœuds finaux du réseau en permettant le déploiement d'un réseau.
 - Les capacités de faible puissance de la Classe A sont préservées.

En proposant la méthode de synchronisation fiable comme solution.

Leurs résultats sont :

- Amélioration théorique de la capacité de réseau 2x.
- Dans un réglage plus élevé, l'amélioration mesurée est jusqu'à 5.8x avec une réduction démontrée des collisions de paquets de 26% dans un déploiement réel avec 24 nœuds fonctionnant pendant des heures.

- Le débit total du S-LoRaWAN proposé n'est que de 8 bits dans le paquet down-link avec une conséquence négligeable en termes de consommation d'énergie.
- S-LoRaWAN ne nécessite aucune modification de la pile logicielle LoRaWAN et peut-être déployée sur le firmware LoRaLAN non modifié.

3.3 Étude sur les travaux de placement des gateways

Les études se sont concentrées sur l'évaluation de la scalabilité LoRa tout en considérant l'interférence co-SF qui provient de collisions lors de l'utilisation de la même configuration SF sur le même canal tandis que d'autres ont supposé que les SF sur un canal sont parfaitement orthogonaux. SF représente le ratio entre le taux de chirp et le taux du symbole de données et affecte directement le taux des données et la plage qu'un dispositif LoRa peut atteindre loin d'une passerelle LoRaWAN. Ils effectuent des coupes dans des réseaux de capteurs sans fil virtuels afin d'améliorer la gestion de la location des ressources physiques avec plusieurs fournisseurs d'applications simultanés [9]. Leurs études se sont basés sur :

- L'étude de la découpe de réseau sur différentes configurations SF pour évaluer la pérennité du système et trouver celui qui convient le mieux aux appareils LoRa dans chaque tranche.
- Proposition d'un algorithme d'interslicage dynamique pour réserver la bande passante de manière équitable sur toutes les passerelles LoRa, comparé à une stratégie de coupure fixe simple.
- Intégration d'un modèle énergétique pour LoRaWAN pour analyser la consommation d'énergie dans chaque tranche et proposition d'un algorithme de découpe intérieure satisfaisant aux exigences de QoS de chaque tranche.

3.3.1 Description du problème

Chaque appareil adopte une configuration spécifique de facteur d'étalement (SF) pour la transmission des données. LoRa étale chaque symbole à un taux de 2^{SF} puces par symbole avec $SF = 7, \dots, 12$, produisant ainsi un débit de données calculé selon l'équation 3.1, où le débit des puces (R_c) et la largeur de bande assignée à la tranche l de la passerelle LoRa m ($b_{l,m}$) influent sur le débit de données. Une largeur de bande de 125 kHz par canal est utilisée conformément à la réglementation européenne, améliorant le débit de données mais réduisant la sensibilité de l'appareil LoRa.

$$R_{k,l,m} = SF \cdot \frac{R_c}{2^{SF}} = SF \cdot \frac{b_{l,m}}{2^{SF}} \text{ bits/s} \quad (3.1)$$

Ce passage décrit l'impact de l'augmentation du facteur d'étalement sur la transmission de données dans un système de communication. Selon l'équation (2), augmenter le facteur d'étalement réduit le débit de données transmis mais augmente la puissance du signal. La puissance reçue (P_{rx}) est influencée par la puissance transmise (P_{tx}), le gain d'antenne du canal (g_{rx} et g_{tx}), la perte de propagation (L) et un composant de masquage log normal ε . Cette relation dépend de la distance entre l'émetteur et le récepteur.

$$P_{k,l,m}^{rx} = \frac{P_{k,l,m}^{tx} g_{k,l,m}^{rx} g_{k,l,m}^{tx}}{L} \cdot e^\varepsilon \quad (3.2)$$

En ce qui concerne les interférences, le rapport signal sur interférence plus bruit (SINR) varie en fonction du facteur d'étalement (SF) utilisé par chaque appareil dans un système LoRa. L'équation (3) décrit la SINR pour un appareil donné, où la puissance du signal reçu (P_{rx}) est confrontée à la somme de la puissance du bruit σ^2 et de la puissance des signaux interférants provenant d'autres appareils P_i^{rx} . Chaque élément de la matrice représente le seuil de marge de puissance minimale $V_{i,j}$ nécessaire pour qu'un paquet envoyé avec un SF donné soit correctement décodé malgré l'interférence des paquets envoyés avec d'autres SF. En d'autres termes, un paquet peut survivre aux interférences si la marge de puissance nécessaire est supérieure à la puissance d'interférence du signal correspondant.

$$SINR_{i,j} = \frac{P_i^{rx}}{\sigma^2 + \sum_{n \in \delta_j} P_n^{rx}} \quad (3.3)$$

Dans cet article le modèle de perte de propagation adopté pour évaluer les performances des dispositifs LoRa dans des environnements denses.

L'équation 3.4 décrit ce modèle, où la perte de chemin (L) est déterminée par la perte de chemin à la distance de référence L_0 , l'exposant de distance de perte de chemin (n), la longueur du chemin (d) et la distance de référence d_0 . En utilisant ce modèle, les performances des dispositifs LoRa peuvent être évaluées en fonction de la distance dans des environnements denses.

$$L = L_0 + 10 \cdot n \cdot \log_{10} \frac{d}{d_0} \quad (3.4)$$

Ce travail explore le concept de découpage de réseau dans un contexte de réseaux LoRa, comparant ses avantages en matière de résilience aux interférences par rapport à d'autres technologies IoT telles que Sigfox et NB-IoT. L'approche consiste à diviser virtuellement les ressources physiques des passerelles LoRa pour créer plusieurs tranches de réseau virtuelles, permettant ainsi une meilleure gestion de la qualité de service (QoS) selon les besoins spécifiques de chaque dispositif IoT. Chaque tranche est définie avec une priorité et une allocation de bande passante spécifique au niveau de la passerelle, optimisant la transmission des données vers les serveurs LoRa en évitant la duplication des paquets. Le découpage de réseau vise donc à améliorer la flexibilité et l'efficacité de la gestion du réseau LoRa face aux diverses exigences des applications IoT, comme le montre la figure 3.7 :

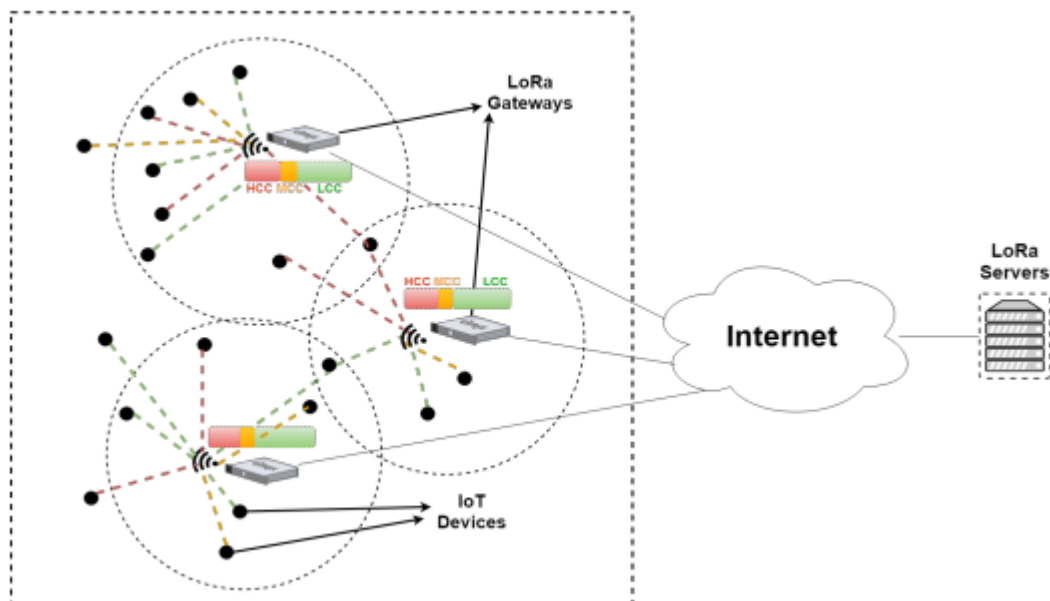


FIGURE 3.7 – L’architecture de découpe IoT dans les réseaux LoRa.

3.3.2 Formulation de problème

Cette étude aborde l’optimisation du découpage de réseau dans les environnements IoT, un problème structuré en trois parties principales :

- l’admission des dispositifs LoRa et leur association avec des tranches spécifiques ;
- la détermination de la meilleure stratégie pour réserver des ressources entre différentes tranches ;
- l’allocation optimale des ressources à l’intérieur des tranches.

Chaque tranche est conçue en tenant compte des besoins en termes de délai et de fiabilité des dispositifs, attribuant à chaque dispositif la tranche qui correspond le mieux à ses exigences de latence. Ensuite, les passerelles réservent des capacités spécifiques pour chaque tranche en fonction des besoins de débit, et chaque dispositif au sein d’une tranche est valorisé par une métrique d’utilité spécifique. Finalement, une allocation optimisée des ressources est réalisée au sein des tranches, assignant chaque dispositif au flux virtuel le plus efficace afin de maximiser l’utilité globale. Ce cadre vise à maximiser l’efficacité et la performance du réseau IoT par une gestion fine des ressources.

Ce document aborde l’optimisation des tranches de réseau virtuel dans les réseaux IoT gérés via des passerelles physiques communes. Il établit un cadre de contraintes pour s’assurer que chaque appareil IoT est associé de manière efficace et précise à une tranche de réseau unique, même dans un environnement avec plusieurs passerelles. Les principales contraintes discutées sont :

$$\text{Max} \sum_{k \in K} \sum_{l \in L} \alpha_{k,l} U_{k,l,m} \forall m \in M \quad (3.5)$$

$$C1 : \sum_{l \in L} \alpha_{k,l} = 1 \forall k \in K \quad (3.6)$$

$$C2 : \sum_{k \in K} \beta_{k,m} P_{k,l,m} \leq P_m^{max}, \forall m \in M, \forall l \in L \quad (3.7)$$

$$C3 : \sum_{k \in K} \alpha_{k,l} \beta_{k,m} r_{k,l,m} \leq R_{l,m}^{max}, \forall l \in L, \forall m \in M \quad (3.8)$$

$$C4 : \beta_{k,m} = \begin{cases} 1 & \text{si le dispositif } k \text{ est attribué à la passerelle } m. \\ 0 & \text{sinon.} \end{cases} \quad (3.9)$$

Contrainte (3.6) : elle stipule que chaque appareil doit choisir uniquement une tranche de réseau, assurant ainsi une association exclusive même lorsque plusieurs passerelles physiques sont impliquées.

Contrainte (3.7) : cette contrainte limite la puissance totale de transmission de chaque passerelle pour prévenir les interférences excessives et gérer la consommation d'énergie.

Contrainte (3.8) : elle garantit que la somme du trafic montant des membres d'une tranche ne dépasse pas la capacité maximale de débit de données que la tranche peut gérer à travers chaque passerelle, évitant ainsi la congestion du réseau.

Contrainte (3.9) : assure une association binaire entre chaque appareil IoT et la passerelle LoRa correspondante, facilitant la gestion claire des connexions réseau.

Ces contraintes sont cruciales pour maintenir la performance et l'efficacité du réseau tout en respectant les exigences spécifiques de chaque tranche de réseau, garantissant ainsi une communication fluide et fiable au sein des environnements IoT complexes.

3.3.3 Algorithmes de découpe proposés

Dans les réseaux LoRa, la gestion des ressources est centralisée dans le cloud pour une coordination efficace, avec les serveurs LoRa prenant les décisions finales sur l'attribution des périphériques et la définition des passerelles. L'algorithme d'optimisation en trois étapes utilise une méthode d'élagage itératif équilibré et de regroupement pour attribuer les périphériques à la tranche appropriée, réserver dynamiquement les ressources des passerelles et transmettre les paquets aux serveurs LoRa pour une utilité maximale.

Définition de découpe basée sur BIRCH

En raison de la nature ultra-dense dans un IoT, l'algorithme BIRCH détaillé dans la figure 3.8 est adopté, qui appartient à la famille de clustering hiérarchique agglomératif et a été prouvé comme la meilleure méthode de clustering disponible pour gérer de grands ensembles de données. L'objectif principal de cette méthode est de définir des tranches en vérifiant les exigences de qualité de service (QoS) en passant d'un grand ensemble de dispositifs à un groupe de sous-ensembles avec des exigences de QoS similaires. Les dispositifs les plus urgents sont ceux qui ont le délai instantané le plus proche d_k de leur budget de délai de paquet PDB_k et qui se voient attribuer la priorité la plus élevée. u_k désigne le facteur d'urgence du dispositif k , avec $u_k = d_k / PDB_k$. Étant donné K_l dispositifs dans un cluster l , ce dernier sera considéré comme un point d'utilité u_k de chaque dispositif dans un cluster avec $k = 1, 2, \dots, K_l$.

Chaque nœud dans l'arbre CF est un cluster de sous-clusters défini par une caractéristique de clustering CF comme suit :

$$CF = ((K_l, LS, SS) = (K_l \sum_{k=1}^{K_l} u_k, \sum_{k=1}^{K_l} u_k^2) \quad (3.10)$$

Dans ce contexte, l'algorithme BIRCH utilise une structure d'arbre appelée CF-tree (Clustering Feature tree) pour organiser dynamiquement les données à mesure que de nouveaux dispositifs MTCN (Machine-Type Communication Devices) sont insérés. Cette structure est construite en fonction de deux paramètres principaux : un facteur de branchement B et un seuil T . Chaque nœud parent dans l'arbre peut contenir jusqu'à B enfants, et chaque nœud enfant peut contenir au maximum T entrées.

Dans ce problème spécifique, B représente le nombre de tranches L créées, et Kl représente le groupe de dispositifs admis dans la tranche l . Ainsi, les nœuds l dérivent de la racine et représentent les tranches créées, où chaque tranche est composée d'un groupe de sous-clusters.

Les paramètres Kl , LS (la somme linéaire des points d'utilité des Kl dispositifs) et SS (la somme des carrés des points d'utilité des kl dispositifs) sont utilisés pour définir les caractéristiques de clustering (CF) de chaque sous-cluster au sein de la tranche. Les entrées dans le CF-tree ne sont donc pas considérées comme des dispositifs individuels, mais plutôt comme des ensembles de sous-clusters C qui appartiennent à la tranche l et regroupent des dispositifs LoRa ayant des points d'utilité presque similaires.

Pseudo-code 1 BIRCH-based Slicing Admission algorithm

Input : Set of devices K , diameter D , branching factor L , threshold T

```

1 begin
2   Initialize as many clusters as devices
   for each  $k \in K$  do
3     Start from root
       Search for closest child node according to  $D$ 
       Search for closest subcluster according to  $D$ 
       if number of entries  $< T$  then
4         Add  $k$  to subcluster  $C_{l,l}$ 
           Update CF of  $C_{l,l}$ 
5       else if number of childs  $< B$  then
6         Create a new subcluster  $C_{l,l'}$ 
           Add  $k$  to  $C_{l,l'}$ 
           Update CF of the parent node  $S_l$ 
7       else if number of parents  $< B$  then
8         Split child nodes and redistribute CF entries according to closest
            $D$ 
9       else
10        Split parent nodes
11      end
12    end
13  Update CF entries in CF-tree
14 end

```

Output: Set of groups $G_l(l=1,2,\dots,L)$

FIGURE 3.8 – Algorithme de découpe basée sur BIRCH

Méthode K-means

Dans la section 1 du document, il est expliqué que le cadre d'apprentissage automatique développé est basé sur l'algorithme k-means, conçu par Lloyd en 1982 et MacQueen en 1967. Cet algorithme est une méthode de clustering partitif qui vise à diviser un ensemble de données en plusieurs groupes distincts et non chevauchants [32].

L'algorithme k-means (voir la figure 3.10) réorganise un ensemble de données $S = \{x_1, \dots, x_n\}$ de n observations en k clusters distincts et non chevauchants, notés $S = \{S_1, \dots, S_k\}$. Chaque cluster S_i est exclusif de sorte que $S_i \cap S_j = \emptyset$ pour $i \neq j$, et l'union de tous les clusters $\bigcup_{i=1}^k S_i$ reconstitue l'ensemble original S . L'objectif de k-means est de minimiser la variance intra-cluster, connue sous le nom de Somme des Carrés Intra-Clusters (WCSS). Ce processus permet de s'assurer que les données au sein de chaque cluster sont aussi similaires que possible, ce qui optimise la qualité du clustering en termes de compacité des clusters autour de leurs centroïdes :

$$WCSS = \sum_{i=1}^k \sum_{x \in S_i} d(x, c^{(i)})^2 \quad (3.11)$$

Le fonctionnement de l'algorithme k-means peut se résumer de la manière suivante :

- Sélectionner un ensemble de k centroïdes initiaux ;
- Étape d'assignation : attribuer chaque point de données au centroïde le plus proche ;
- Étape de mise à jour : réévaluer les centroïdes pour tous les clusters ;
- Répéter les étapes 2 et 3 jusqu'à convergence (par exemple, lorsque les centroïdes cessent de changer ou qu'un nombre maximum d'itérations est atteint).

Contrairement à d'autres paradigmes de clustering de données, tels que le clustering libre, le nombre de clusters k à retourner est un paramètre d'entrée fourni par l'utilisateur final et trouver une valeur appropriée dépend strictement du problème et des données, et n'est généralement pas connu à l'avance. Typiquement, on teste plusieurs candidats pour k et on sélectionne la meilleure valeur en étudiant la fonction objective dans l'équation (9) et/ou au moyen d'indices de validation internes. Les stratégies courantes comprennent :

Le "Elbow Plot" consiste à tracer la WCSS en fonction de k et à choisir la première valeur de k correspondant au point où la courbe devient plate. La logique derrière ce critère est qu'il est inutile d'ajouter plus de clusters s'ils n'apportent pas une meilleure modélisation des données (la courbe s'aplatit car la WCSS ne change pas de manière significative).

L'indice de Davies-Bouldin mesure la séparation intra-cluster par rapport à la variance inter-cluster. Soit S_i la dispersion statistique du cluster i , c'est-à-dire la distance moyenne entre les motifs et le centroïde, et soit $M_{i,j}$ la distance entre les centroïdes appartenant aux clusters i et j . Pour qu'une solution de clustering soit bonne, S_i doit être petite (cluster compact), tandis que $M_{i,j}$ doit être grande (les différents clusters sont bien éloignés), d'où pour chaque paire de clusters on peut définir le score de pénalité suivant.

$$R_{i,j} = \frac{S_i + S_j}{M_{i,j}} \quad (3.12)$$

3.3. ÉTUDE SUR LES TRAVAUX DE PLACEMENT DES GATEWAYS40

et le score de Davies-Bouldin pour le cluster i est défini comme :

$$DBI_i = \text{Max}_{j \neq i} R_{i,j} \quad (3.13)$$

Finalemnt, le score de Davies-Bouldin pour la solution de clustering globale est obtenu en faisant la moyenne des scores de chaque cluster :

$$DBI = \frac{1}{k} \sum_{i=1}^k DBI_i \quad (3.14)$$

L'indice de Davies-Bouldin est orienté négativement : plus il est proche de 0, meilleure est la solution de clustering.

Algorithm 1 K-Means-based SF Allocation

Input: $ED := \bar{N}$ uniformly deployed nodes

Output: $L := \{l_0, l_1, \dots, l_6\}$

```

1:  $l_6 := R$ 
2: for  $i$  in  $\{5, \dots, 1\}$  do                                ▷ For each SF ring, starting from the outermost ring
3:    $K := \text{GetKfromSeries}(i)$                                   ▷ Set number of centroids for this iteration
4:    $C := \text{Kmeans}(ED, K)$                                     ▷ Compute the centroids
5:    $B := \text{boundary}(C)$                                        ▷ Compute the boundary of C
6:    $I := \{x \in ED \mid x \in \text{conv}B\}$                           ▷ Select nodes that are inside the boundary B
7:    $l_i := \frac{\max(|I_x|) + \max(|I_y|)}{2}$                             ▷ Compute the new SF ring limit
8:    $SF_{i+7} := \{x \in ED \mid x \notin \text{Ball}[(0,0), l_i]\}$     ▷ Allocate  $SF_{i+7}$  to nodes outside the circle of radius  $l_i$ 
9:    $ED := \{x \in ED \mid x \in \text{Ball}[(0,0), l_i]\}$             ▷ Remove nodes outside the circle of radius  $l_i$ 
10:  $l_0 := 0$ 
11:  $SF_7 := ED$                                                 ▷ Allocate  $SF_7$  to remaining nodes
12: return L

```

FIGURE 3.9 – Allocation de surface en fonction de l'algorithme K-Means

D'une autre manière l'algorithme de k-means s'écrit comme c'est mentionné dans la figure 3.10 :

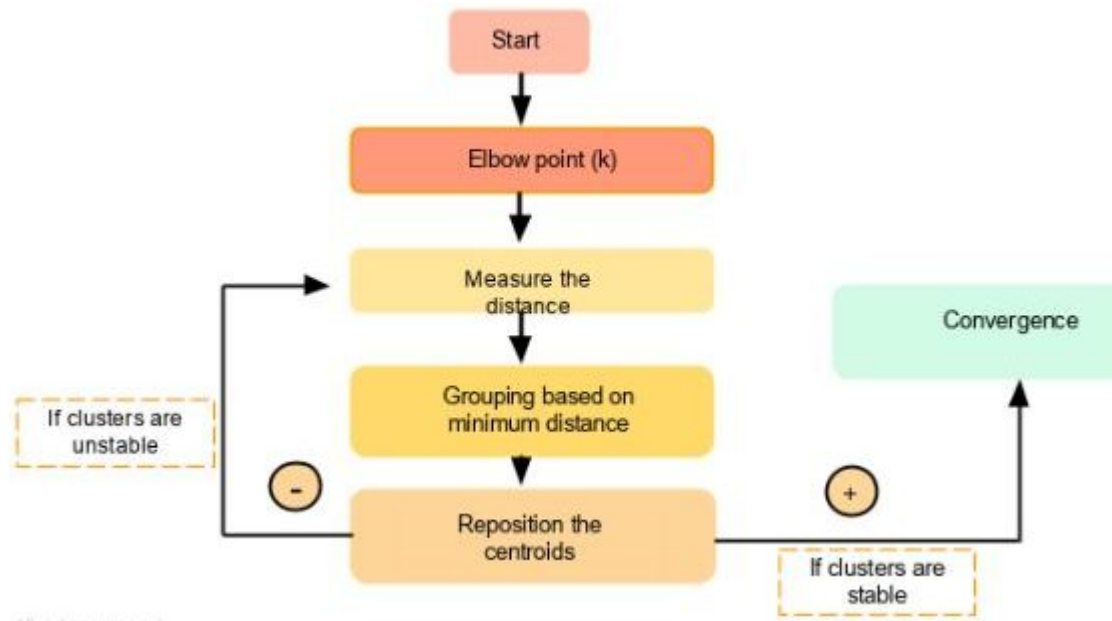


FIGURE 3.10 – Algorithme K-means.

Méthode Fuzzy C-Means (FCM) clustering of LoRa transmission settings

L'algorithme de clustering FCM (voir la figure 3.12) est une méthode non supervisée largement utilisée pour l'analyse des caractéristiques. Il fonctionne en classant les points de données en clusters en fonction de leurs espaces de caractéristiques. Contrairement aux méthodes traditionnelles comme le k-means, le FCM offre une flexibilité accrue en permettant aux éléments d'avoir des degrés d'appartenance à plusieurs clusters simultanément, ce qui est particulièrement utile lorsque les caractéristiques des données peuvent varier [10].

Dans une étude récente, le FCM a été utilisé pour évaluer la qualité des paramètres de transmission dans un réseau. Les paramètres sont projetés dans un espace tridimensionnel basé sur des métriques de Qualité de Service (QoS), et le processus de clustering est optimisé pour minimiser une fonction de coût évaluant la distance entre les paramètres et les centres des clusters. Chaque paramètre se voit alors attribuer un degré d'appartenance à chaque cluster, ce qui permet de prédire la qualité de la liaison de transmission utilisée par un dispositif.

L'objectif du clustering FCM est de mettre à jour les valeurs de la matrice d'appartenance M et les positions des centres des clusters H , convergent vers une solution lorsque l'erreur seuil ϵ est atteinte après un certain nombre d'itérations. Les valeurs d'appartenance à chaque cluster sont floues, variant entre 0 et 1, ce qui permet une représentation plus nuancée de l'appartenance des éléments aux clusters.

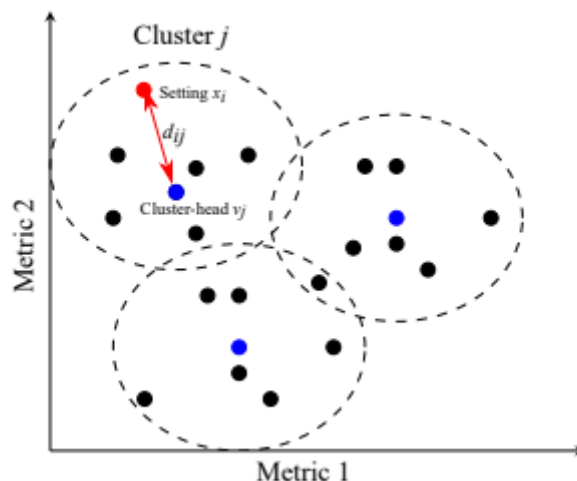


FIGURE 3.11 – Distance entre les centres de clusters et les paramètres de transmission.

Algorithm 3.1 Fuzzy C-Means (FCM) clustering algorithm

- 1: **Input:** $X = [x_{11}, \dots, x_{np}]$
 - 2: **Output:** (\mathbf{M}, \mathbf{H})
 - 3: $t = 0$
 - 4: **while** $F_m(\mathbf{M}_t, \mathbf{H}_t) > \varepsilon$ **do**
 - 5: $t = t + 1$
 - 6: Update \mathbf{H}_t from Equation 3.10
 - 7: Update \mathbf{M}_t from Equation 3.9
 - 8: $(\mathbf{M}, \mathbf{H}) = (\mathbf{M}_t, \mathbf{H}_t)$
-

FIGURE 3.12 – Algorithme de clustering Fuzzy C-Means (FCM).

Les simulations montrent que l'algorithme FCM est efficace et précis dans la classification des configurations en trois clusters distincts. Ces groupements permettent de classer les configurations selon leur appartenance aux clusters. L'algorithme prend en compte plusieurs paramètres et conditions, incluant les spécificités des technologies LoRa et les facteurs environnementaux, qui ont un impact sur la qualité de service (QoS).

3.4 Critique de l'article intitulé : "K-Means Spreading Factor Allocation for Large-Scale LoRa Networks"

Ce travail de recherche contribue au problème de l'allocation du facteur de propagation (SF) LoRa en proposant un algorithme basé sur le regroupement K-means [32]. Ils évaluent les performances du réseau en tenant compte des probabilités de panne d'un modèle LoRaWAN de classe-A en mode non confirmé, sans retransmissions. L'algorithme proposé permet une répartition différente des utilisateurs sur les SF, rendant ainsi l'allocation de SF flexible. Cette répartition se traduit par des paramètres réseau dépendant de l'application. Les résultats de simulation

prennent en compte différents scénarios réseau et des paramètres réalistes pour illustrer comment la distance par rapport à la passerelle et le nombre de noeuds dans chaque SF affectent la fiabilité de la transmission. Les résultats théoriques et de simulation montrent que leur approche d'allocation de SF améliore la probabilité de couverture moyenne du réseau jusqu'à 5 points de pourcentage par rapport au modèle de référence. De plus, leurs résultats montrent un fonctionnement plus équitable du réseau où la différence de performance entre les noeuds de meilleur et de pire cas est significativement réduite. Cela se produit parce que leur méthode cherche à égaliser l'utilisation de chaque SF. En montrant que la performance de pire cas dans un scénario de déploiement peut être améliorée de 1,53 fois.

Cet article de recherche scientifique sur l'allocation des facteurs d'étalement (SF) dans les réseaux LoRa, se concentre sur l'utilisation de l'algorithme de K-means pour cette tâche.

Critiquer l'article nécessite d'évaluer ses forces et ses faiblesses, ainsi que sa contribution à la recherche dans le domaine des réseaux LoRa. Voici une critique générale de l'article :

Forces :

- **Méthodologie innovante** : l'article propose une approche novatrice pour l'allocation des facteurs d'étalement (SF) dans les réseaux LoRa en utilisant l'algorithme de K-means, ce qui peut potentiellement améliorer l'efficacité et les performances du réseau.
- **Analyse complète** : l'étude comprend une analyse approfondie de la scalabilité et de la performance de la méthode proposée, avec des simulations informatiques pour étayer les résultats.
- **Considération des paramètres** : l'article prend en compte plusieurs paramètres pertinents, tels que le cycle de service, l'exposant de perte de trajet et la bande de fréquence, ce qui contribue à une évaluation plus précise des performances du réseau.

Faiblesses :

- **Limitations de la méthodologie** : l'utilisation de l'algorithme de K-means pour l'allocation de SF peut ne pas être optimale dans toutes les situations, en particulier dans les environnements de réseau complexes où la topologie peut varier considérablement.
- **Manque de validation pratique** : bien que l'article discute de l'implémentation potentielle de l'algorithme dans LoRaWAN, aucune validation pratique sur le terrain n'est fournie pour confirmer l'efficacité de l'approche proposée dans des conditions réelles.
- **Complexité de l'analyse** : l'article présente une analyse détaillée des résultats, mais la complexité des simulations et des données présentées peut rendre la compréhension difficile pour les lecteurs et il pourrait être utile de simplifier la présentation des résultats.

En conclusion, le document présente une approche intéressante pour l'allocation des SF dans les réseaux LoRa, mais il pourrait bénéficier d'une meilleure contextualisation, d'une validation pratique supplémentaire et d'une simplification de la présentation des résultats pour améliorer son accessibilité et son impact.

3.5 Conclusion

Dans le cadre du développement et de l'optimisation de réseaux LoRaWAN, les efforts récents se sont concentrés sur l'amélioration des protocoles d'accès et le placement stratégique des gateways, deux éléments essentiels pour renforcer l'efficacité et la portée de cette technologie IoT.

Du côté des protocoles d'accès, des modèles tels qu'ALOHA et des techniques de machine learning (ML) sont explorés pour gérer les communications dans des environnements à haute densité de dispositifs IoT. ALOHA, par exemple, offre une méthode simple mais efficace pour gérer l'accès aléatoire non coordonné, bien que susceptible aux collisions sous fortes charges. L'intégration de techniques de ML permet d'optimiser ces protocoles, en prédisant et en adaptant les comportements du réseau pour réduire les collisions et améliorer le débit.

En ce qui concerne le placement des gateways, la sélection du facteur d'étalement (SF) joue un rôle crucial. Le SF, qui détermine la portée de la communication et la résistance aux interférences, doit être soigneusement choisi pour équilibrer la couverture réseau et la capacité. Un placement judicieux des gateways, prenant en compte le SF optimal, peut significativement améliorer la couverture réseau et l'efficacité énergétique, surtout dans des zones urbaines ou topographiquement complexes.

En conclusion, l'association de protocoles d'accès avancés, comme ALOHA enrichi par des approches de ML, avec un placement stratégique des gateways en fonction des caractéristiques de SF, permet aux réseaux LoRaWAN de mieux servir les exigences croissantes des applications IoT. Ces solutions proposées contribuent à créer un réseau LoRaWAN plus robuste, plus performant et plus adaptable aux divers besoins des utilisateurs finaux et des environnements opérationnels.

Dans le chapitre à venir, nous nous concentrerons sur une solution visant à optimiser les clusters pour garantir un meilleur équilibre de charge sur les passerelles tout en utilisant au maximum deux passerelles, à minimiser l'utilisation du SF pour économiser l'énergie des appareils finaux et à assurer une couverture globale du réseau.

4

Optimisation de placement et duplication des passerelles dans un réseau LoRaWAN

4.1 Introduction

L'optimisation des clusters pour un accès réseau performant est un domaine crucial dans le développement de systèmes sans fil à grande échelle. L'objectif principal est de minimiser le nombre de gateways tout en garantissant une couverture réseau totale, en utilisant au moins deux gateways par nœud, en minimisant le facteur d'étalement de spectre (SF) et en assurant une couverture totale du réseau.

Ce chapitre explore la problématique posée dans ce travail où nous sommes basées sur le déploiement d'un réseau LoRaWAN en minimisant le nombre de gateways.

Le chapitre explore ensuite la modélisation mathématique avec de la programmation linéaire qui est adaptée au problème posé.

Enfin, le chapitre conclut en un algorithme qui mènera vers la solution volue.

4.2 Contexte et description générale

Dans ce chapitre notre travail consiste à faire une configuration optimale des gateways dans un réseau LoRaWAN pour un accès réseau performant.

4.2.1 Structure du réseau

Un réseau LoRaWAN se compose de quatre éléments principaux :

- **End-devices (Noeuds capteurs)** : il s'agit des dispositifs finaux chargés de collecter et de transmettre les données aux passerelles.
- **Passerelles** : elles agissent en tant que points d'accès au réseau, recevant les données des capteurs et les relayant vers les serveurs.
- **Serveur réseau LoRaWAN** : le Serveur réseau LoRaWAN occupe une place centrale au sein du réseau. Il assure la gestion des nœuds capteurs et des passe-

relles, ainsi que la validation des données, leur stockage et leur mise à disposition pour les applications.

- **Serveur d'application** : quant au Serveur d'application, il sert d'interface logicielle permettant aux utilisateurs d'interagir avec les données issues des nœuds capteurs. Ce dernier met à disposition divers outils pour visualiser, analyser et traiter efficacement ces précieuses informations.

4.2.2 Fonctionnement du réseau

Le réseau fonctionne selon :

- Les capteurs recueillent des informations qu'ils mettent en forme en paquets LoRaWAN.
- Ces paquets sont ensuite acheminés vers les passerelles à portée.
- Les dites passerelles se chargent de transmettre ces paquets aux serveurs par le biais d'une connexion réseau telle que l'Ethernet ou le cellulaire.
- Une fois réceptionnés, les serveurs décryptent les données et les traitent avant de les stocker dans une base de données dédiée.
- Il est alors possible pour les applications d'accéder à ces données depuis les serveurs afin de procéder à leur analyse et visualisation.

4.2.3 Gestion du réseau

La gestion d'un réseau LoRaWAN efficace requiert la prise en compte de divers aspects fondamentaux :

- **Placement des passerelles** : une disposition stratégique des passerelles s'avère primordiale afin d'assurer une couverture adéquate du réseau. Des outils de simulation de propagation radio peuvent être utiles pour optimiser leur positionnement.
- **Configuration des passerelles** : les passerelles doivent être configurées correctement pour les canaux LoRaWAN appropriés, la puissance de transmission et d'autres paramètres.
- **Sécurité du réseau** : la mise en place de mesures sécuritaires vigoureuses demeure essentielle afin de prémunir le réseau contre tout accès non autorisé ou intrusion malveillante.
- **Surveillance du réseau** : un suivi continu du fonctionnement du réseau se révèle indispensable pour détecter et résoudre promptement tout dysfonctionnement potentiel.

4.3 Problématique

Déployer un réseau LoRaWAN performant avec un nombre minimal de passerelles tout en garantissant une couverture réseau complète et en minimisant la consommation d'énergie des capteurs nécessite d'assurer certains points :

4.3.1 Déploiement des gateways

Pour déployer les gateways il faut :

- Utiliser un placement stratégique des gateways pour maximiser la couverture et minimiser le nombre de gateways nécessaires.
- Considérer des facteurs tels que la densité des nœuds, la topologie du terrain et les obstacles environnementaux.
- Envisager l'utilisation de techniques d'optimisation basées sur des algorithmes pour déterminer le placement optimal des gateways.

4.3.2 Sélection du facteur d'étalement de spectre (SF)

La sélection du facteur d'étalement de spectre se fait comme suit :

- Choisir un SF approprié qui offre un équilibre entre la couverture réseau et la consommation d'énergie.
- Des SF plus élevés offrent une meilleure couverture mais augmentent la consommation d'énergie, tandis que des SF plus bas réduisent la consommation d'énergie mais limitent la couverture.
- Adapter le SF en fonction des besoins spécifiques de l'application et des contraintes environnementales.

4.3.3 Technologie de réseau

- La technologie de réseau adaptée à l'application et à l'environnement est LoRaWAN.
- Cette technologie offre des caractéristiques en termes de portée, de débit et de consommation d'énergie.

4.4 Motivations et objectifs

Ce travail a été fait grâce à des motivations et en ayant des objectifs qui sont :

- **Motivations :**

l'Internet des objets révolutionne de nombreux secteurs, de l'agriculture à l'industrie en passant par les villes intelligentes. Les réseaux LoRaWAN s'imposent comme une technologie clé pour connecter des objets à faible consommation d'énergie sur de grandes distances.

Ce projet vise à explorer l'optimisation du placement et de la duplication des passerelles dans les réseaux LoRaWAN. En optimisant la configuration du réseau, nous pouvons :

- **Réduire le nombre de passerelles nécessaires**
- **Économiser l'énergie des capteurs**
- **Améliorer la couverture réseau**
- **Optimiser les performances du réseau**

- **Objectifs :** nos objectifs sont :

- **Mettre en œuvre un algorithme d'optimisation du placement des passerelles :** cet algorithme identifiera automatiquement les emplacements optimaux pour les passerelles afin de maximiser la couverture réseau et de minimiser le nombre de passerelles nécessaires.

- **Développer un outil de visualisation des résultats** : cet outil permettra de visualiser la couverture réseau, l'emplacement des passerelles et les performances du réseau pour une meilleure compréhension et prise de décision.

En réalisant ce projet, nous espérons contribuer à l'avancement de la technologie LoRaWAN et à la création de réseaux IoT plus efficaces, économiques et fiables.

4.5 Optimisation de placement et duplication des gateway dans un réseau LoRaWAN (OPG-LoRaWAN)

L'optimisation du placement et de la duplication des gateways dans un réseau LoRaWAN est un aspect crucial pour garantir une couverture réseau adéquate, une communication fiable et une utilisation efficace des ressources. Plusieurs facteurs doivent être pris en compte pour optimiser le déploiement des gateways, notamment :

4.5.1 Hypothèses

Le problème posé repose sur les hypothèses suivantes :

- **Les gateways ne sont pas limités en terme d'énergie et capacité mémoire (le nombre de capteurs à couvrir n'est pas limité)** : cette hypothèse suppose que les passerelles peuvent gérer un nombre illimité de connexions sans contraintes énergétiques ou de mémoire. Cela permet de se concentrer sur la réduction du nombre total de passerelles nécessaires pour une couverture réseau complète et redondante.
- **Les noeuds sont statiques** : les capteurs et passerelles sont immobiles une fois installés, garantissant des connexions constantes. Cette stabilité simplifie l'optimisation du réseau, permettant aux algorithmes d'utiliser des données fixes sans ajustements dynamiques.

Le défi principal est d'optimiser la configuration des passerelles pour la couverture et la redondance. Sans contraintes d'énergie, de mémoire et de mobilité, la modélisation linéaire assure que chaque capteur se connecte à au moins deux passerelles, tout en minimisant le nombre total de passerelles nécessaires.

4.5.2 Placement stratégique des passerelles

- Prioriser les emplacements élevés et dégagés pour une meilleure propagation du signal.
- Couvrir les zones à forte densité de capteurs avec des passerelles dédiées.

4.5.3 Duplication des passerelles pour une meilleure fiabilité

- Déployer des paires de passerelles redondantes.
- Utiliser des techniques de basculement automatique pour garantir la disponibilité du réseau en cas de défaillance d'une passerelle.

4.5.4 Optimisation continue et ajustements

- Analyser les données de couverture et de qualité du signal pour identifier les zones à améliorer.
- Ajuster le placement des passerelles et le SF en fonction des besoins et des observations.

4.6 Modélisation mathématique avec la programmation linéaire

Dans cette section, nous utiliserons la modélisation linéaire pour optimiser le placement et la duplication des passerelles, visant à minimiser leur nombre tout en assurant une couverture réseau complète. Chaque capteur se connectera à au moins deux passerelles différentes pour garantir la redondance et la fiabilité du réseau.

4.6.1 Description de la Programmation Linéaire

La programmation linéaire (PL) est une méthode de modélisation mathématique utilisée pour optimiser un objectif, tel que la minimisation des coûts ou la maximisation des profits, sous certaines contraintes linéaires. Elle est largement utilisée dans divers domaines tels que la gestion, l'ingénierie et les sciences sociales.

Dans notre contexte, nous avons opté pour la programmation linéaire en raison de sa capacité à traiter efficacement des objectifs et des contraintes linéaires, tout en manipulant des variables continues. Notre objectif est de minimiser le nombre de passerelles nécessaires tout en assurant une couverture réseau complète. Chaque capteur doit être connecté à au moins deux passerelles pour garantir la redondance et la fiabilité du réseau.

4.6.2 Contraintes

Pour résoudre le problème, il est nécessaire de tenir compte des contraintes suivantes :

- Utiliser au moins deux (2) gateways pour chaque noeud.
- Minimiser le facteur d'étalement de spectre (SF) pour réduire la consommation d'énergie des capteurs et par conséquent augmenter le débit .
- Assurer la couverture totale du réseau (tous les capteurs doivent être connectés à au moins deux gateways).

Explication des contraintes

Plusieurs approches peuvent être envisagées pour résoudre ce problème :

a- Déploiement stratégique des gateways : cela nécessite de tenir compte des points suivants :

- Analyse de la distribution des capteurs.
- Placement optimisé des gateways.
- Utiliser des algorithmes d'optimisation.

b- Sélection appropriée du facteur d'étalement de spectre (SF) : cela se fait aussi en tenant compte des points suivants :

- Évaluer l'impact du SF sur la consommation d'énergie et le débit.
- Considérer la portée et la fiabilité des transmissions.
- Utiliser des techniques d'adaptation dynamique du SF.

c- Mise en œuvre de mécanismes de redondance des passerelles :

- Utiliser des gateways redondantes

En combinant ces approches, on peut concevoir un réseau IoT répondant aux contraintes de couverture totale, de consommation d'énergie minimale et de débit maximal. Le choix de la solution dépendra des spécificités de l'application et des exigences de performance. Il est crucial de considérer les caractéristiques uniques de l'environnement de déploiement, des capteurs et des applications pour trouver la solution optimale.

4.6.3 Approches

La programmation linéaire (PL) est appropriée pour notre problème en raison de sa capacité à gérer efficacement des objectifs et des contraintes linéaires, des variables continues, et un objectif d'optimisation unique.

1) **Modélisation linéaire** la modélisation linéaire repose sur l'utilisation des variables et d'une fonction objectif, comme expliqué ci-dessous.

a) **Variables** : description des variables :

- X_{ij} : variable binaire indiquant si le capteur i est affecté au gateway j .

$$X_{ij} = \begin{cases} 1 & \text{si le gateway } j \text{ est à la portée du noeud } i \\ 0 & \text{sinon.} \end{cases}$$

- **Explication de la variable X_{ij} : Affectation des capteurs aux passerelles**

La variable X_{ij} est une variable binaire qui indique si un capteur i est affecté à une passerelle j . Elle prend la valeur 1 si le capteur i est effectivement affecté à la passerelle j , et 0 dans le cas contraire. Cette variable est définie en fonction de la portée de la passerelle j et de la distance entre le capteur i et la passerelle j .

- Y_j : variable binaire indiquant si le gateway j est actif.

$$Y_j = \begin{cases} 1 & \text{si le gateway } j \text{ est actif} \\ 0 & \text{sinon.} \end{cases}$$

- **Explication de la variable Y : Activation des passerelles**

La variable Y est également une variable binaire qui indique si une passerelle j est active ou non. Elle prend la valeur 1 si la passerelle j est active, et 0 dans le cas contraire. Cette variable est définie en fonction de la couverture réseau requise et de la capacité de la passerelle j à couvrir cette zone, voir la figure 4.1 :

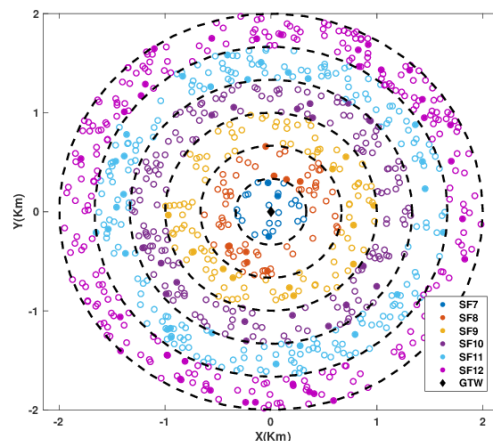


FIGURE 4.1 – Réseau avec passerelles actives et inactives.

Dans ce schéma, la passerelle j est active et couvre un ensemble de capteurs. La couleur bleu-clair indique la zone de couverture de la passerelle j , et les capteurs situés dans cette zone sont connectés à la passerelle j , telle que :

$$X_{ij} = \begin{cases} 1 & \text{si } d(i, j) \leq \text{portee}_i(SF_i) \\ 0 & \text{sinon.} \end{cases}$$

tel que

$$d(i, j) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \quad (4.1)$$

b) Fonction objectif :

$$M = \min \sum_{j=1}^m y_j \quad (4.2)$$

$$\sum_{j=1}^{|P|} X_{ij} Y_j \geq 2 \quad (4.3)$$

$$X_{ij} y_j \in \{0, 1\} \quad (4.4)$$

tel que : P est le cardinal du nombre des Gateways.

2) Minimisation du nombre de gateways actifs : réduire le nombre de gateways actifs au minimum nécessaire pour couvrir le réseau et économiser des ressources.

4.7 Algorithmes de la solution proposée

Ci-dessous on trouve le tableau 4.1 qui représente les variables utilisées dans l'algorithme :

Variable	Description	Valeur
ED_i	End device ou noeud capteur (objet) i	x,y positions du ED
G_j	Gateway j	x,y : position du G
P_{ED}	portée radio d'un objet	$\simeq 5$ km
P_G	portée radio d'un Gateway	$\simeq 50$ km
$SF[]$	Etalement de spectre	7-12
$M_{ED}(x, y)$	Matrice d'emplacement des ED	La matrice avec x,y de chaque ED
$M_G(x, y)$	Matrice d'emplacement du Gateway	La matrice avec x,y de chaque Gateway
N_{ED}	Nombre de ED déployés	N_{ED} à lire
$d_{ED,G}$	Distance entre le ED et le Gateway	Déduite selon la loi de PYTHAGORE
M_{AG}	Matrice d'Activation des Gateways	x,y position des Gateways
Max_{SF}	La valeur maximale de SF	7-12

TABLE 4.1 – Variables utilisées dans l'algorithme.

4.7.1 OGP-LoRaWAN Algorithm

Avant de détailler l'algorithme, nous donnerons un aperçu de sa méthode et des étapes clés qu'il utilise pour optimiser le placement des passerelles dans les réseaux LoRaWAN.

L'algorithme OGP-LoRaWAN optimise l'activation des gateways dans un réseau LoRaWAN en fonction de la position des nœuds finaux (ED). Initialement, les nœuds sont déployés de manière aléatoire dans la zone de couverture. Ensuite, les identités et les coordonnées des gateways pré-déployées sont récupérées. Toutes les gateways sont désactivées, et la liste des gateways actives est vide au départ. L'algorithme vérifie pour chaque nœud final s'il existe déjà un gateway active à proximité. Si ce n'est pas le cas, un nouvel gateway est activé avec les paramètres appropriés. Sinon, le gateway actif existant est utilisé pour servir le nœud final. Cette méthode permet de minimiser l'activation des gateways, optimisant ainsi l'utilisation des ressources et la consommation d'énergie du réseau.

Voici l'algorithme d'optimisation de placement des gateways :

Algorithm 1 OGP-LoRaWAN Algorithm

```

1: Input :  $N_{ED}(x,y)$ ,  $N_{AG}$   $Max_{SF}$ ,  $SF_{min}$ ;
2: Output :  $M_{AG}[]$ ;
3:  $M_{ED}(x,y) \leftarrow \text{Deploy}(N_{ED})$ ;
4:  $M_G[] = 1 \leftarrow \text{Virtual-Deploy-G}(\text{Long-zone}, \text{large-zone})$ ; {Récupérer les identités
   et les  $(x,y)$  de Gateways }
   ;
5:  $M_{AG}[] \leftarrow 0$ ; {Désactiver tous les Gateways}
    $A_G \leftarrow \text{nil}$ ; { La liste des Gateways actifs est initialisée à nil.}

   {Elle sert à tester d'abord si un Gateway déjà actif est proche d'un ED};
6: while (i=1 à  $N_{ED}$ ) do
7:   if ( $A_G[] = \text{nil}$ ) then
   ( $A_G[], SF[], M_{AG}[], M_G[]$ )  $\leftarrow \text{Acivate}_G(i, Max_{SF}, M_G[], N_{AG})$ ;
8:   else
   ( $A_G[], SF[], M_{AG}[], M_G[]$ )  $\leftarrow \text{Acivate}_G(i, Max_{SF}, A_G[], N_{AG}$ ;
9:   end if
10: end while
    =0

```

4.7.2 Activation des passerelles

Avant de présenter en détail l'algorithme d'activation des passerelles, nous dresserons un aperçu des principes fondamentaux qu'il utilise pour gérer efficacement le processus d'activation et d'intégration des passerelles dans le réseau LoRaWAN.

Dans notre approche, nous supposons un déploiement virtuel des passerelles dans chaque cellule de deux (2) km^2 (voir plus). Après selon le déploiement aléatoires des noeuds capteurs (ED), notre solution essaye de voir quels sont les gateways à activer afin de respecter les contraintes établies auparavant, à savoir :

- La couverture .
- La redondance des passerelles en cas de panne.

Algorithm 2 Activate Gateways

```

1: Input :  $M_G[]$ 
2: Output :  $AG[], SF[], M_{AG}[], M_G[]$ 
3: Fonction  $Activate_G(ed, max_{sf}, M_{Get}, N_{ag})$ 
4: var  $n_g, g, sf$  : entier
5: ( $n_g$  : nombre de Gateway pour un  $ED = ed$ )
6: ( $g$  : pour parcourir tous les Gateways de  $M_{Get}$  fournis en entrée)
7:  $n_g \leftarrow 1$ ;  $sf \leftarrow 7$  while ( $sf \leq 12$  et  $n_g < N_{ag}$ ) do
8:   (Tester si  $sf \leq 12$  et le nombre max de Gateway n'est pas encore atteint)
10:   $g \leftarrow 1$ 
11:  while ( $g \leq \text{Taille}(G[])$ ) do
12:    (Parcourir tous les Gateways)
13:    if ( $\text{distance}(ed, g) \leq \text{Portée}(ed)$ ) then
14:      (Si un Gateway est dans la portée radio d'un ED)
15:       $SF[ed] \leftarrow sf$ 
16:      (enregistrer le  $sf$  minimal pour un ED)  $M_{AG}(g) = 0$ 
17:      (tester par la suite si le Gateway  $g$  n'a pas déjà été activé)
18:       $M_{AG}(g) \leftarrow 1$ 
19:      (Activer le Gateway)
20:       $AG[] \leftarrow g$ 
21:      (Ajouter  $g$  dans la liste des Gateways actives)
22:       $M_G[] \leftarrow M_G[] - g$ 
23:      (Retirer le Gateway  $g$  de la liste globale des Gateways après son transfert vers les Gateways actives)
24:    end if
25:     $g \leftarrow g + 1$ 
26:  end while
27:   $sf \leftarrow sf + 1$ ;  $n_g \leftarrow n_g + 1$ 
28: end while
29: return ( $AG[], SF[], M_{AG}[], M_G[]$ ) = 0

```

Les deux algorithmes présentés offrent des approches distinctes mais complémentaires pour optimiser respectivement le placement des passerelles et le processus d'activation dans les réseaux LoRaWAN. Leur mise en œuvre vise à améliorer la couverture, la fiabilité et l'efficacité des réseaux IoT. Dans le prochain chapitre, nous examinerons en détail les résultats obtenus à travers l'application de ces algorithmes, mettant en lumière leurs performances et leurs contributions potentielles à l'évolution des réseaux LoRaWAN.

4.8 Conclusion

En conclusion, ce chapitre a exploré l'optimisation du placement et de la duplication des passerelles dans un réseau LoRaWAN, en mettant en lumière l'importance de stratégies bien définies pour maximiser la couverture et minimiser le nombre de passerelles. Nous avons formulé des hypothèses claires et utilisé la modulation adaptative pour ajuster les paramètres de communication en fonction de la distance des nœuds aux passerelles. En appliquant la programmation linéaire, nous avons pu modéliser notre problématique et obtenir des solutions optimales qui démontrent une amélioration notable en termes de couverture et d'efficacité d'affectation. Les résultats obtenus offrent des perspectives prometteuses pour le déploiement futur de réseaux IoT, suggérant que l'intégration de techniques avancées pourrait encore améliorer la performance et la robustesse des réseaux LoRaWAN.

5

Implémentation des résultats

5.1 Introduction

Dans ce chapitre, nous abordons l'implémentation pratique des solutions proposées pour l'optimisation des réseaux LoRaWAN. Après avoir détaillé les concepts théoriques, les algorithmes développés, et les méthodes d'optimisation dans les chapitres précédents, il est essentiel de démontrer l'efficacité de ces solutions à travers des implémentations concrètes. Ce chapitre présente les outils de développement utilisés, les étapes d'implémentation suivies, et les résultats obtenus.

L'objectif principal est de valider les approches développées en les appliquant à des scénarios réels ou simulés, afin d'évaluer leur performance et leur robustesse. Nous commencerons par une présentation des outils de développement, notamment PyCharm, un environnement de développement intégré (IDE) utilisé pour la programmation en Python. Ensuite, nous décrirons en détail les étapes d'implémentation, depuis le déploiement des End Devices et des passerelles (Gateways), jusqu'à l'analyse des résultats du programme.

Les résultats obtenus seront analysés pour illustrer les améliorations apportées en termes de couverture réseau et d'efficacité énergétique, confirmant ainsi la pertinence et l'efficacité des solutions proposées dans le contexte de l'Internet des Objets (IoT) et des réseaux LoRaWAN.

5.2 Outils de développement

Python (voir la figure 5.1) est un langage de programmation interprété, multiparadigme et multiplateformes, ce qui signifie qu'il peut être exécuté sur différents systèmes d'exploitation sans nécessiter de compilation. Il est connu pour sa lisibilité, sa simplicité d'utilisation et sa grande communauté d'utilisateurs.

Le choix du langage de programmation Python a été basé sur de nombreuses raisons, notamment :

- **Lisibilité** : la syntaxe de Python est claire et facile à apprendre, ce qui le rend accessible aux débutants comme aux développeurs expérimentés.

- **Simplicité d'utilisation** : Python est conçu pour être intuitif et facile à utiliser, ce qui permet aux développeurs de se concentrer sur la logique de leur programme plutôt que sur les complexités du langage.
- **Polyvalence** : Python peut être utilisé pour un large éventail de tâches, du développement web à la science des données en passant par l'intelligence artificielle.
- **Grande communauté** : Python possède une communauté d'utilisateurs active et large, ce qui signifie qu'il existe de nombreuses ressources disponibles pour apprendre et obtenir de l'aide.
- **Bibliothèques riches** : Python dispose d'un vaste écosystème de bibliothèques tierces pour presque toutes les tâches imaginables, ce qui permet aux développeurs de gagner du temps et d'être plus productifs.



FIGURE 5.1 – Python Logo

5.2.1 PyCharm

PyCharm est un IDE développé par JetBrains, particulièrement reconnu pour son support du langage Python. Disponible en deux éditions, Professionnelle et Communautaire, il offre un ensemble complet d'outils pour le développement de logiciels Python, allant de l'écriture et du débogage du code à la gestion de projets et au déploiement d'applications. Sa version est 2024.2.3, (voir la figure 5.2).



FIGURE 5.2 – PyCharm Logo

* Bibliothèques et frameworks pris en charge :

PyCharm offre une prise en charge native d'un large éventail de bibliothèques et de frameworks Python populaires, notamment :

- NumPy
- Pandas
- Matplotlib
- Scikit-learn
- TensorFlow

- PyTorch
- Django
- Flask

Et bien d'autres

* **Bibliothèques utilisées :**

Voici quelques-unes des bibliothèques utilisées avec PyCharm :

- **NumPy** : une bibliothèque essentielle pour les calculs numériques en Python, fournissant des structures de données et des opérations mathématiques performantes.
- **Matplotlib** : une bibliothèque pour la visualisation de données, permettant de créer des graphiques et des visualisations en 2D.
- **SciPy** : une bibliothèque utilisée pour les calculs scientifiques et techniques, construite sur NumPy.

Ces bibliothèques peuvent être facilement installées et gérées via PyCharm grâce à son intégration avec des outils comme pip et conda. PyCharm offre également des fonctionnalités avancées pour la gestion de projets, l'analyse de code, le débogage, et l'intégration avec des systèmes de contrôle de version comme Git.

5.3 Etapes implémentées

Il est important de présenter les étapes de notre réseau comme suit :

5.3.1 Déploiement des End devices

Notre objectif est de générer et de visualiser aléatoirement les positions d'un réseau de capteurs (End Devices) dans un environnement bidimensionnel. Voici une description des étapes de ce programme :

- **Définition de la fonction endDevice** : la fonction commence par définir le nombre de capteurs à déployer (nbrED). Une matrice 2x100 est ensuite créée à l'aide de la fonction `np.random.rand`, générant des coordonnées aléatoires pour chaque capteur dans un intervalle $[0, 100]$, représentant les positions X et Y des capteurs dans un espace de 100 km par 100 km. Les valeurs sont arrondies à deux décimales pour plus de précision.
- **Affichage des coordonnées** : la matrice des positions est imprimée dans la console pour une visualisation textuelle. La fonction `plt.scatter` de la bibliothèque `matplotlib.pyplot` est utilisée pour créer un nuage de points représentant les positions des capteurs dans l'environnement.
- **Configuration et affichage du graphique** : les axes du graphique sont étiquetés comme "Largeur de l'environnement (km)" pour l'axe X et "Hauteur de l'environnement (km)" pour l'axe Y. Le graphique est ensuite affiché à l'aide de `plt.show()`, permettant de visualiser la distribution aléatoire des capteurs.
- **Exécution du script** : si le script est exécuté directement (et non importé comme un module), la fonction `endDevice` est appelée pour générer et afficher les positions des capteurs.

Donc le programme génère une matrice de positions aléatoires pour un ensemble de capteurs dans un environnement 2D et les affiche sous forme de nuage de points, permettant ainsi de visualiser leur distribution spatiale, comme la montre la figure 5.3 :

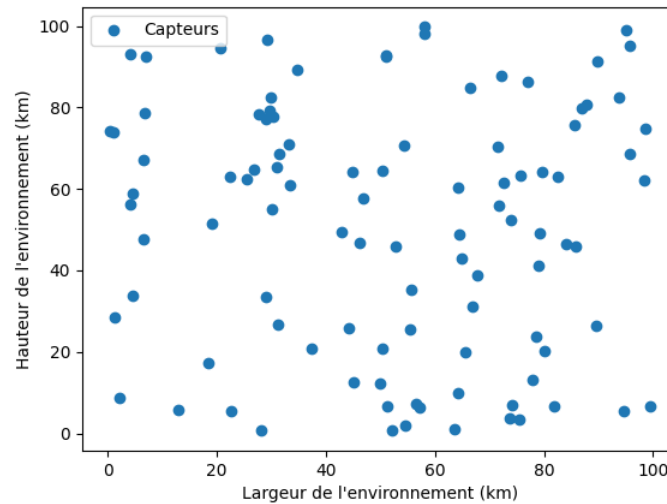


FIGURE 5.3 – Déploiement des End Devices.

5.3.2 Déploiement des Gateways

Avec une seule passerelle : le but du programme est de simuler le déploiement d'un réseau de capteurs et de passerelles dans un champ donné, afin d'analyser les connexions possibles entre les End devices et les passerelles. Les End devices sont déployés de manière aléatoire et possèdent des rayons de connexion variables. Le programme détermine ensuite les passerelles les plus proches pour chaque End device, même si elles se trouvent hors du rayon de connexion défini.

a- **Structuration du programme :** le programme est structuré en plusieurs fonctions :

- **EndDevice :** déploie aléatoirement les End devices et génère les rayons de connexion.
- **deployGateways :** déploie aléatoirement les passerelles et établit les connexions entre les End devices et les passerelles.
- **plotNetwork :** visualise le réseau de capteurs et de passerelles ainsi que les connexions entre eux.

b- **Étapes d'Implémentation**

- **Déploiement des End devices :** cette fonction génère aléatoirement les positions des End devices dans un environnement de taille spécifiée (*field_size*). Chaque capteur a un rayon de connexion aléatoire compris entre 7 et 12 unités de mesure.
- **Déploiement des Passerelles et Établissement des Connexions** cette fonction déploie aléatoirement les passerelles et calcule les distances entre chaque capteur et chaque passerelle. Ensuite, elle établit les connexions entre les capteurs et les passerelles en fonction des distances et des rayons de connexion.

- **Visualisation du Réseau** cette fonction visualise le réseau en utilisant Matplotlib. Elle affiche les capteurs et les passerelles, ainsi que les connexions entre eux. Les rayons de connexion de chaque capteur sont également représentés par des cercles.
- c- **Exécution du programme** : il commence par l'appel des fonctions, afin d'initialiser les capteurs et leurs rayons de connexion, déploie les passerelles et affiche le réseau(voir la figure 5.4).

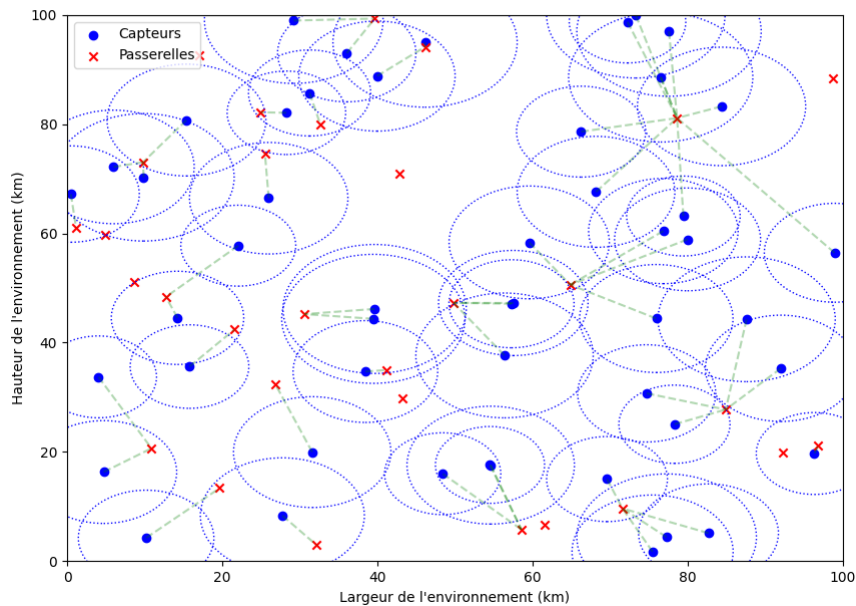


FIGURE 5.4 – Déploiement avec une passerelle.

En cause d'existence de passerelles qui n'ont aucun lien avec les End devices, on minimise le nombre de passerelles utilisées en éliminant les passerelles inactives, comme le montre la figure 5.5 ci-dessous :

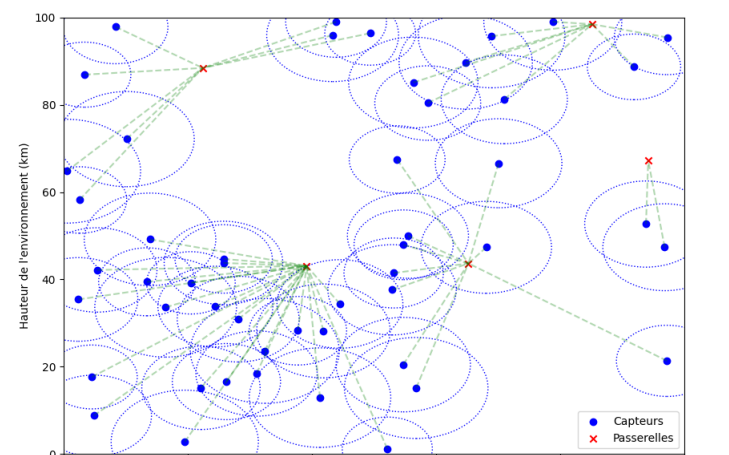


FIGURE 5.5 – Déploiement avec une passerelle (min).

Le programme permet de simuler et de visualiser efficacement un réseau de

capteurs et de passerelles. Il est modulable et peut être ajusté en fonction des besoins spécifiques, tels que le nombre de capteurs, de passerelles, ou la taille de l'environnement.

2- Avec deux passerelles :

a- Initialisation et Déploiement des End devices (Capteurs) :

- La fonction `endDevice` initialise le déploiement des capteurs dans un environnement de taille définie (par défaut, 100x100 km).
- Les positions des capteurs sont générées aléatoirement et stockées dans une matrice.
- Les positions des capteurs sont ensuite visualisées à l'aide d'un graphique à points.

b- Déploiement des Passerelles :

- La fonction `deployGateways` initialise les positions potentielles des passerelles sur une grille régulière couvrant l'environnement.
- Elle calcule la matrice des distances entre chaque capteur et les positions potentielles des passerelles.
- Chaque capteur est connecté aux deux passerelles les plus proches qui se trouvent dans une certaine portée (par défaut, 30 km).
- Un ensemble initial de passerelles nécessaires est déterminé en assurant que chaque capteur soit couvert par au moins deux passerelles.
- Le programme optimise ensuite le nombre de passerelles en vérifiant si certaines peuvent être supprimées tout en maintenant la connectivité requise pour chaque capteur.
- Le nombre final de passerelles nécessaires est affiché.

c- Visualisation du Réseau :

- La fonction `plotNetwork` visualise le réseau en affichant les positions des capteurs et des passerelles optimisées.
- Elle ajoute des cercles de couverture pour chaque capteur afin de montrer leur portée.
- Elle trace des lignes indiquant les connexions entre chaque capteur et ses deux passerelles les plus proches.
- Le graphique final illustre l'organisation du réseau et les connexions entre capteurs et passerelles.

d- Exécution Principale :

- Le programme est exécuté en appelant successivement les fonctions `endDevice`, `deployGateways`, et `plotNetwork`, en passant les résultats de l'une à l'autre pour compléter le déploiement et la visualisation du réseau (voir la figure 5.6).

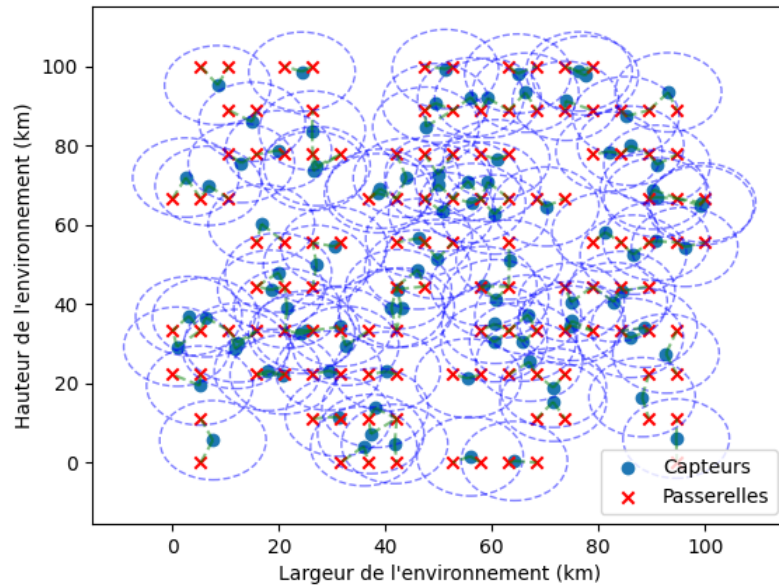


FIGURE 5.6 – Déploiement avec deux passerelles.

En cause du nombre considérable de passerelles, on minimise ce dernier comme le montre la figure 5.7 ci-dessous :

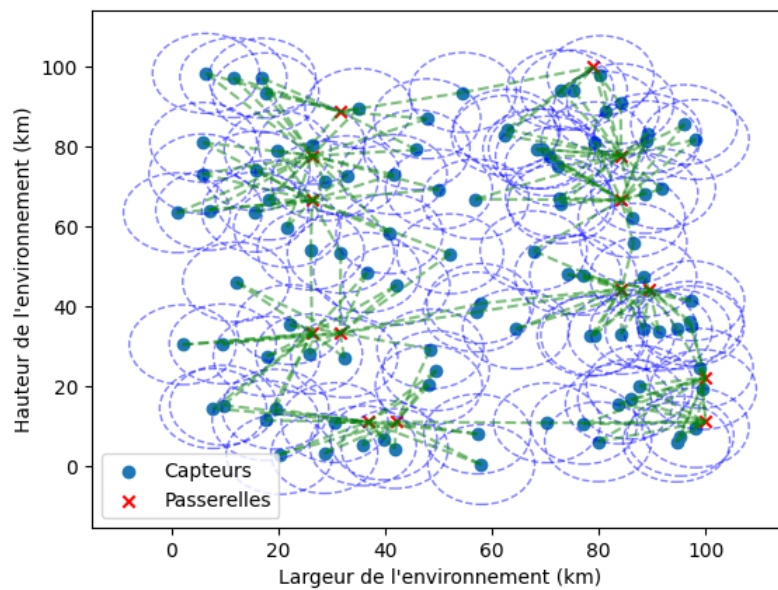


FIGURE 5.7 – Déploiement avec deux passerelles(min).

Ce programme permet de simuler et d'optimiser le déploiement de capteurs et de passerelles dans un environnement donné, en assurant la connectivité et la couverture nécessaires tout en minimisant le nombre de passerelles utilisées.

5.3.3 Résultats du programme

Nous effectuons une génération aléatoire et utilisons les résultats obtenus comme un exemple représentatif :

La figure 5.8 ci-dessous montre le déploiement des End devices :

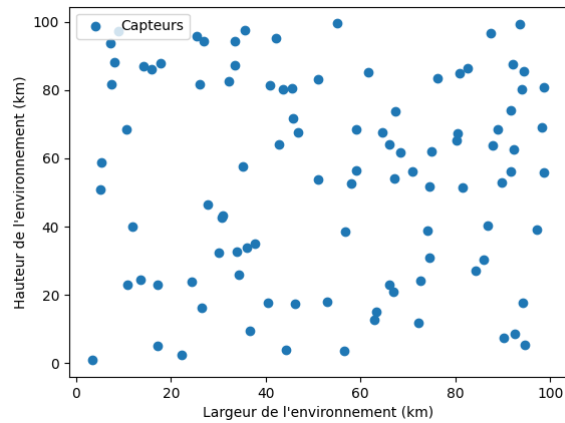


FIGURE 5.8 – Déploiement des End devices.

La figure 5.9 ci-dessous montre le déploiement du réseau :

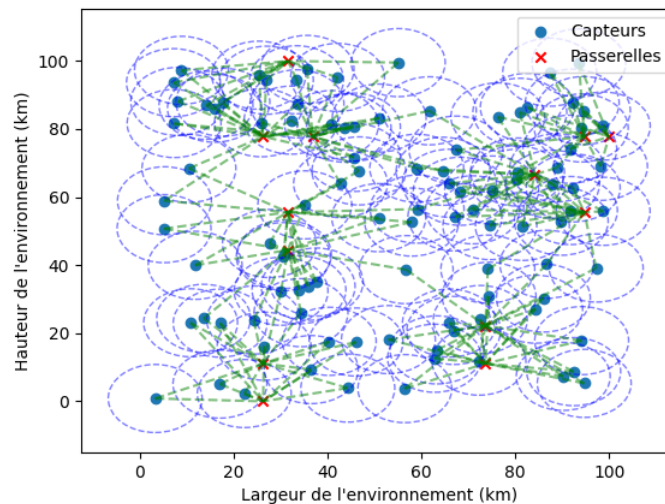


FIGURE 5.9 – Déploiement du réseau

La figure 5.10 ci-dessous illustre le déploiement du réseau avec SF différent :

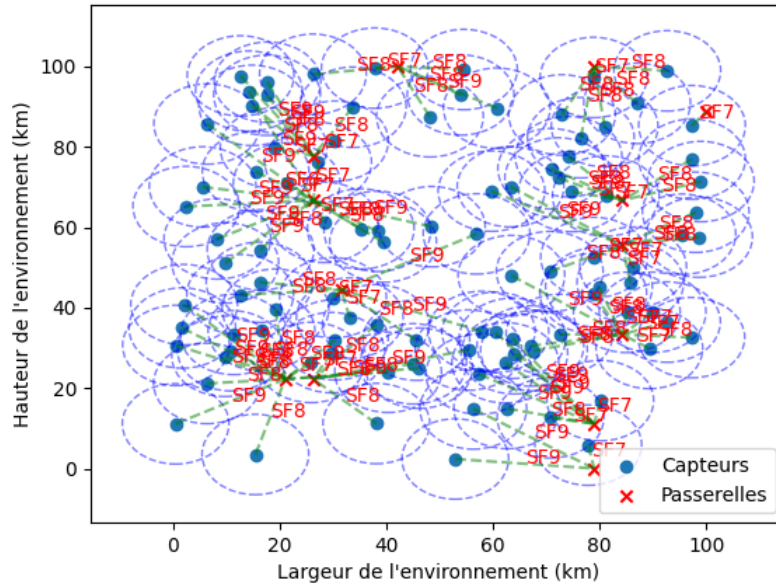


FIGURE 5.10 – Déploiement du réseau avec SF différent

Les résultats obtenus démontrent l'efficacité de notre approche dans l'optimisation du déploiement des passerelles, assurant une couverture adéquate avec un nombre minimal de passerelles nécessaires.

Dans cet exemple le nombre minimal de passerelles utilisées pour 100 End devices est : 13 passerelles.

5.4 Conclusion

Dans ce chapitre, nous avons implémenté et validé les résultats de l'optimisation du nombre de passerelles dans un réseau de capteurs en utilisant Python. Nous avons déployé aléatoirement des capteurs dans un environnement de 100 km x 100 km, puis calculé les positions optimales des passerelles en s'assurant que chaque capteur était connecté à au moins deux passerelles. Un algorithme d'optimisation a permis de réduire le nombre de passerelles nécessaires tout en maintenant une couverture fiable. La visualisation finale a montré les positions des capteurs et des passerelles optimisées ainsi que les rayons de couverture des capteurs, validant ainsi notre approche. Cette optimisation réduit les coûts d'installation et de maintenance tout en assurant une performance et une fiabilité accrues du réseau de capteurs.

Conclusion générale

L'Internet des Objets (IoT) connecte des dispositifs physiques à Internet pour une communication continue et des échanges de données en temps réel, révolutionnant divers domaines comme la domotique et la gestion industrielle.

Le déploiement des réseaux LoRaWAN rencontre divers défis, tels que l'accès optimal pour les dispositifs IoT dans des environnements complexes, le placement stratégique des gateways pour une couverture maximale avec un minimum de coûts, et la nécessité d'interopérabilité entre différents équipements conformes aux normes. Pour y répondre, des solutions comme l'attribution dynamique de canaux et l'utilisation de modèles d'optimisation intégrés aux données géographiques sont explorées. Ces approches visent à garantir une connectivité fiable tout en optimisant les ressources disponibles, soutenant ainsi une adoption efficace et évolutive des réseaux LoRaWAN.

Ce projet de fin d'étude se concentre sur l'optimisation du placement et de la duplication des passerelles dans les réseaux LoRaWAN, une technologie clé de l'IoT. Les défis incluent la couverture optimale du réseau, la réduction des interférences et l'assurance d'une communication fiable et efficace. Pour y répondre, une approche utilisant la programmation linéaire et l'algorithme OGP-LoRaWAN a été proposée.

En conclusion, ce mémoire apporte une contribution significative en proposant des solutions pratiques pour améliorer les performances des réseaux LoRaWAN. Les perspectives futures incluent :

- Exploration de techniques d'intelligence artificielle (IA) pour une gestion dynamique avancée des ressources réseau.
- Développement de systèmes d'apprentissage automatique pour prédire et ajuster dynamiquement la capacité des gateways en fonction des besoins du réseau.
- Intégration de l'IA dans la planification et l'optimisation du placement des gateways pour une couverture optimale et une efficacité accrue.
- Collaboration avec l'industrie et la recherche pour standardiser et optimiser les algorithmes d'IA adaptés aux spécificités des réseaux IoT basés sur LoRaWAN.

Bibliographie

- [1] ADHIKARY, A., LIN, X., AND WANG, Y.-P. E. Performance evaluation of nb-iot coverage. In *2016 IEEE 84th Vehicular Technology Conference (VTC-Fall)* (2016), pp. 1–5.
- [2] AKYILDIZ, I. F., AND KASIMOGLU, I. H. Wireless sensor and actor networks : research challenges. *Ad Hoc Networks* 2, 4 (2004), 351–367.
- [3] ALLIANCE, L. Lorawan regional parameters. <https://lorawan-alliance.org/resource-hub/lorawanr-regional-parameters>, 2021.
- [4] BELTRAMELLI, L., MAHMOOD, A., OSTERBERG, P., AND GIDLUND, M. Lora beyond aloha : An investigation of alternative random access protocols. In *Proceedings of the IEEE Conference on LoRaWAN* (2024).
- [5] BENKAHLA, N. *Gestion de la qualité de service (QoS) dans un réseau LoRaWAN avec mobilité*. Thèse de doctorat, Ecole Supérieure des Communications de Tunis, 2021. NNT : tel-03283203.
- [6] BENKAHLA, N. *Gestion de la qualité de service (QoS) dans un réseau LoRaWAN avec mobilité*. Thèse, Ecole Supérieure des Communications de Tunis, 2021. NNT : tel-03283203.
- [7] BOR, M., AND ROEDIG, U. Lora transmission parameter selection. In *Proceedings of the 13th International Conference on Distributed Computing in Sensor Systems (DCOSS)* (Ottawa, Canada, 2017), pp. 27–34.
- [8] CORPORATION, S. Lora modulation basics. <https://www.semtech.com/lora/what-is-lora>, 2021.
- [9] DAWALIBY, S., BRADAI, A., AND POUSSET, Y. Adaptive dynamic network slicing in lora networks. *À compléter* (2021).
- [10] DJOUDI, A. *Toward a Self-Reconfigurable LoRaWAN Network for Smart City Applications*. Thèse de doctorat, Université Gustave Eiffel, 2023. NNT : 2023UEFL2022, tel-04195062.
- [11] ELDEFRAWY, M., BUTUN, I., PEREIRA, N., AND GIDLUND, M. Formal security analysis of lorawan. *Computer Networks* 148 (2019), 328–339.
- [12] ERTÜRK, M. A., AYDIN, M. A., BÜYÜKAKKASLAR, T., AND EVIRGEN, H. A survey on lorawan architecture, protocol and technologies. *Future Internet* 11 (October 2019), 216.
- [13] EXPLAINED, S. Cdma/cd vs csma/ca, 2023.
- [14] FARHAD, A., AND PYUN, J.-Y. Lorawan meets ml : A survey on enhancing performance with machine learning. In *Proceedings of the IEEE Conference on LoRaWAN and Machine Learning* (2024).
- [15] FRENZEL, L. *Fundamentals of Communications Access Technologies : FDMA, TDMA, CDMA, OFDMA, AND SDMA*. 2013. Last modified on Jan. 22.

- [16] HEUSSE, M., CAILLOUET, C., AND DUDA, A. Performance of unslotted aloha with capture and multiple collisions in lorawan. In *Proceedings of the IEEE Conference on LoRaWAN* (2024).
- [17] HUSSEIN, D., AND IBRAHIM, D. M. Improving lorawan performance using reservation aloha. *International Journal of Information Technology and Management* 19, 2 (April 2020), 123–135.
- [18] JRADI, H., SAMHAT, A. E., NOUVEL, F., MROUE, M., AND PRÉVOTET, J.-C. Overview of the mobility related security challenges in lpwans. *Computer Networks* 186 (2021), 107761.
- [19] KHATER, E. M., AND IBRAHIM, D. M. Proposed ST-Slotted-CS-ALOHA protocol for time saving and collision avoidance. *ISeCure* 11, 3 (2019).
- [20] LAMPORT, L. *LaTeX : A Document Preparation System*. Addison-Wesley, Reading, Massachusetts, 1994.
- [21] LAVRIC, A., PETRARIU, A. I., AND POPA, V. Sigfox communication protocol : The new era of iot ? In *2019 International Conference on Sensing and Instrumentation in IoT Era (ISSI)* (2019), pp. 1–4.
- [22] LORA ALLIANCE. LoRaWAN® 1.0.4 specification package. URL. <https://lora-alliance.org/resource-hub/lorawan-104-specification-package/>.
- [23] LORA ALLIANCE. LoRaWAN® specification v1.0.3. URL. <https://lora-alliance.org/resource-hub/lorawan-specification-v1-0-3/>.
- [24] LORA ALLIANCE. LoRaWAN® specification v1.1. URL. <https://lora-alliance.org/resource-hub/lorawan-specification-v1-1/>.
- [25] MAMMERI, Z. Université paul sabatier (toulouse iii), 2007.
- [26] N. SORNIN (SEMTECH), AND A. YEGIN (ACTILITY). *LoRaWAN Specification v1.1*. LoRa Alliance, 2017.
- [27] RAOUF, M. M., 2019.
- [28] RUSH, C. Qu’est-ce que la technologie lora, lorawan. En ligne.
- [29] RÉSEAUX. Le protocole csma (1-persistent non persistent), 2024.
- [30] SUNDARAM, J. P. S., DU, W., AND ZHAO, Z. A survey on lora networking : Research problems, current solutions, and open issues. *IEEE Communications Surveys & Tutorials* 22, 1 (2020), 371–388.
- [31] TRIANTAFYLLOU, A., ZORBAS, D., AND SARIGIANNIDIS, P. Time-slotted lora mac with variable payload support. In *Proceedings of the IEEE Conference on Wireless Communications and Networking* (2024).
- [32] ULLAH, M. A., IQBAL, J., HOELLER, A., SOUZA, R. D., AND ALVES, H. K-means spreading factor allocation for large-scale lora networks. In *Proceedings of the IEEE International Conference on Communications (ICC)* (2024).
- [33] VERMESAN, O., AND FRIESS, P., Eds. *Internet of Things : From Research and Innovation to Market Deployment*. River Publishers, 2014.
- [34] WANG, H., AND FAPOJUWO, A. O. A survey of enabling technologies of low power and long range machine-to-machine communications. *IEEE Communications Surveys & Tutorials* 19, 4 (2017), 2621–2639.
- [35] WIKIPEDIA. LoRaWAN. En ligne, mai 2021. <https://fr.wikipedia.org/wiki/LoRaWAN>.

-
- [36] WIKIPÉDIA. Chirp spread spectrum. En ligne, mai 2021. <https://en.wikipedia.org/wiki/Chirp-spread-spectrum>.
- [37] WIKIPÉDIA. LoRa. En ligne, mai 2021. <https://en.wikipedia.org/wiki/LoRa>.

Résumé

Résumé :

Le déploiement des réseaux LoRaWAN comme technologie récente pour les applications IoT nécessite de mettre en place des passerelles d'accès entre les utilisateurs et les fournisseurs de services. Notre projet traite le problème de déploiement des passerelles afin d'assurer une meilleure couverture et une continuité de service (connectivité). Nous avons proposé une solution d'optimisation de placement des passerelles (Gateways) en satisfaisant ces contraintes tout en minimisant le nombre de passerelles à déployer. Les résultats de simulation ont montré que notre solution réduit le nombre de Gateways déployés en assurant la couverture des utilisateurs (end devices).

Mots clés : IoT, LoRa, LoRaWAN, Programmation linéaire, Optimisation, couverture réseau.

Abstract :

The deployment of LoRaWAN networks as a recent technology for IoT applications requires the establishment of access gateways between users and service providers. Our project addresses the gateway deployment problem to ensure improved coverage and service continuity (connectivity). We proposed an optimization solution for gateway placement, satisfying these constraints while minimizing the number of gateways to be deployed. Simulation results have demonstrated that our solution reduces the deployed gateways while ensuring coverage for end devices.

Keywords : IoT, LoRa, LoRaWAN, Linear Programming, Optimization, network coverage .
