

Université de Mons
Faculté des Sciences
Département d'Informatique
SERVICE RESEAU ET TELECOMMUNICATION

**Identifying IoT nodes by spying on their
radio signal**

Directeur : M^{me} Bruno QUOITIN

Mémoire réalisé par
Arnaud TULIPPE HECQ

Rapporteurs : M^r Prénom NOM
M^r Prénom NOM

en vue de l'obtention du grade de
Master en Sciences Informatiques



Année académique 2019-2020

Remerciements

Nous remercions ...

Table des matières

Introduction	2
1 Rappels et nomination des technologies	3
1.1 Signal radio	3
1.2 Traitement du signal	4
1.2.1 Modulation	4
1.2.2 Gestion du bruit	8
1.2.3 Transformée de Fourier	8
1.3 LoRa	10
1.3.1 couche physique LoRa	11
1.3.2 LoRaWAN	13
2 Travaux similaires et autres contributions	19
2.1 identification de device dans l'iot	19
2.1.1 historique et évolution des préoccupation de sécurité dans l'iot	19
2.1.2 approches d'identification dans l'iot	22
2.2 analyse de la technologie lora	23
2.2.1 analyse de la couche physique lora	23
2.2.2 analyse de la modulation CSS de Lora	23
2.3 identification de device lora	23
2.3.1 RFFI avec DCTFs	23
2.3.2 RFFI avec spectrogrammes	23
3 Expérimentations	24
3.1 Matériel	24
3.1.1 radio logicielle	24
3.1.2 Module d'émission Lora	25
3.1.3 logiciel	25
3.2 Librairie python	27
3.3 Génération et réception d'un signal LoRa	27

TABLE DES MATIÈRES

1

3.3.1	analyse du signal	27
3.3.2	automatisation du signal	28
3.4	Méthode "Constellation traces"	28
4	Résultats	30
4.1	Méthode des constellations Traces	30
4.1.1	paramétrage	30
4.1.2	training phase	30
4.1.3	testing phase	30
4.1.4	résultats	30
	Annexes	31
A	Première annexe	31
B	Deuxième annexe	32

Introduction

L'avènement de *L'Internet of Things* (IoT) a lancé une nouvelle ère d'appareils connectés, ouvrant de nouvelles possibilités de partage de l'information, d'automatisation et de protection. Bien que le concept lui-même soit prometteur, la technologie qui l'accompagne est essentielle. Les premières technologies utilisées pour l'IoT étaient les technologies sans fil déjà présentes comme le Wifi ou le Bluetooth. Elles ont cependant plusieurs limitations : une consommation en énergie élevée, une portée restreinte et parfois même un coût d'infrastructure trop important. Dans ces circonstances est apparu *LoRa*, une technologie développée en particulier pour l'IoT. Sa capacité à gérer les communications longue portée même dans des environnements urbains très denses est un grand atout pour le domaine.

L'expansion de l'IoT soulève une nouvelle problématique de sécurité. Entre autres, l'identification des nœuds au sein des réseaux est essentielle. Il a été découvert que des nœuds fabriqués avec les mêmes microprocesseurs et modèles d'émetteurs-récepteurs radio peuvent présenter de subtiles particularités dans les caractéristiques de leurs signaux. Cette variabilité intrinsèque de la transmission des signaux radio peut être exploitée pour distinguer les nœuds d'un réseau. En écoutant leurs signaux radio et en analysant leurs signatures distinctes, il devient possible de les identifier.

Ce travail est structuré en quatre parties. Le premier chapitre sert d'aperçu global du signal radio afin d'y développer et rappeler les concepts de télécommunication de base. Ce chapitre présente également les technologies LoRa et LoRaWAN à travers leurs caractéristiques et leur pertinence dans l'IoT. Le second chapitre rassemble les travaux qui ont déjà été effectués dans ce domaine. Le troisième chapitre est dédié à l'étude expérimentale du sujet. Les aspects pratiques y seront appliqués, notamment l'utilisation de radio logicielle afin de capturer des signaux radio. Ces signaux seront ensuite analysés grâce à diverses méthodes détaillées dans ce chapitre. La dernière partie du travail présentera les résultats obtenus en suivant l'analyse effectuée au chapitre précédent. Enfin le travail sera achevé en concluant sur de potentielles implications plus larges à ce sujet ainsi que des recherches plus approfondies.

Chapitre 1

Rappels et nomination des technologies

1.1 Signal radio

Un signal est une variation dans l'espace ou dans le temps d'une quantité physique contenant de l'information. Un signal peut être continu ou discret, on le nomme alors respectivement analogique ou numérique. Le type de signal dépend notamment de l'information qu'il contient. Un signal analogique est continu en amplitude, ce qui veut dire qu'il peut contenir un nombre infini de valeur, ainsi que prendre toutes les valeurs possibles, là où un signal numérique contient généralement un nombre fini de valeur (par exemple des 0 et 1). Les deux catégories ne sont pas incompatible car il est souvent nécessaire en télécommunication de pouvoir passer de l'un à l'autre.[?]

L'utilisation de signaux radio en télécommunication confère de nombreux avantages, comme la portée, la vitesse de transmission ou encore le coût de propagation. Pouvoir transporter de l'information sans avoir recours à du support matériel complet (pas besoin de câble, le signal passe dans l'air) réduit considérablement le cout de la transmission. Ajouté à cela, il est possible d'adapter un signal pour le rendre compatible avec diverse canaux de transmission et de réception, grâce à la modulation. La modulation est une technique permettant de modifier les propriétés du signal lui permettant de transporter de l'information.

En télécommunication, les signaux sont des ondes électromagnétiques appelées signal radio. Les signaux comportent de nombreuses caractéristiques qui les déterminent :

la fréquence, mesurée en Hertz (Hz). Elle détermine le nombre de cycle qu'accomplit le signal par seconde. Une onde radio possède une fréquence entre 9kHz et 300GHz.

La largeur de spectre, elle dépend de la fréquence car c'est l'écart entre la plus haute et la plus basse fréquence du signal. Une plus grande largeur permet de transmettre plus d'informations, mais consomme plus d'énergie.

L'amplitude. Selon le type de signal l'attribut possède différentes fonctions. Dans le cas d'un signal analogique l'amplitude détermine la magnitude de l'onde pour n'importe quel point dans le temps. Dans un signal numérique l'amplitude est interprétée différemment. Les signaux numériques sont encodés avec des valeurs discrètes, où chaque valeur représente un niveau (par exemple 0 ou 1). L'amplitude permet de faire la distinction entre ses niveaux.

la puissance, mesurée en Watt (W). C'est la force du signal, un attribut important pour la réception du signal notamment. Bien que le Watt soit utilisé pour décrire la puissance à l'émission ou la réception, les variations de puissances sont généralement exprimées en décibels (dB). Le décibel est une unité logarithmique permettant de mesurer plus facilement les relations entre les différents niveaux de puissance.

le *Signal to Noise Ratio* (SNR). Cet attribut mesure la qualité du signal. une valeur élevée indique que le pourcentage de bruit est faible.

le *Bit rate*, ou le taux de transmission mesure la quantité de donnée transmise en bit par seconde. Cet attribut est exclusif aux signaux numériques. On parle de *Baud rate* pour mesurer la quantité de symboles transmise par seconde. Ce n'est pas exactement l'équivalent du bit rate car un symbole peut contenir plusieurs bits, mais le Baud rate est utilisé pour les signaux numériques et analogiques.

1.2 Traitement du signal

1.2.1 Modulation

La réception d'un signal radio nécessite une antenne dont les dimensions dépendent de la longueur d'onde du signal. La longueur d'onde d'un signal représente la distance entre deux points consécutifs de même phase dans l'onde. La longueur d'onde s'obtient par la formule suivante :

$$c = f * \lambda \quad (1.1)$$

où c est la vitesse de la lumière,
 f est la fréquence,
 λ est la longueur d'onde.

Il est donc possible d'adapter les caractéristiques d'un signal pour le rendre compatible à différentes antennes, via la modulation. La modulation est le procédé par lequel un ou plusieurs attributs du *baseband signal* (le signal modulant, contenant l'information à transmettre) vont être altéré par le *carrier signal* (un signal porteur, utilisé pour être combiné avec le signal modulant) pour devenir un signal modulé, le *modulated signal*.

En plus de sa compatibilité, un signal modulé a l'avantage d'être facilement transmissible sur une grande portée sans perdre en puissance.

Parmi ces différents attributs, certains sont utilisés pour effectuer une modulation. Les trois modulations les plus utilisées sont basées sur les attributs de la fréquence, l'amplitude et la phase. La modulation en fréquence (*Frequency modulation, FM*) consiste à encoder l'information en faisant varier la fréquence en maintenant l'amplitude constante. La modulation en amplitude (*Amplitude modulation, AM*) est le procédé inverse, c'est à dire encoder l'information en faisant varier l'amplitude tout en gardant la fréquence constante. La modulation en phase (*Phase Modulation, PM*) fait varier la phase de la porteuse proportionnellement à l'amplitude instantanée du baseband signal.

La modulation en amplitude est plus ancienne et est encore utilisée dans beaucoup de systèmes. Cette technique possède moins de contrainte et est notamment plus simple à implémenter.

Soient $u(t)$ un baseband signal et $v(t)$ un carrier signal, la modulation en amplitude s'effectue en multipliant les deux signaux pour obtenir le signal modulé

$$s(t) = u(t) \cdot v(t) \quad (1.2)$$

Prenons par exemple

$u(t) = \sin(2\pi f_u t)$ avec $f_u = 5$ Hz,

$v(t) = \cos(2\pi f_c t)$ où $f_c = 50$ Hz.

La Figure 1.1 montre le signal modulé $s(t)$ via la modulation en amplitude.

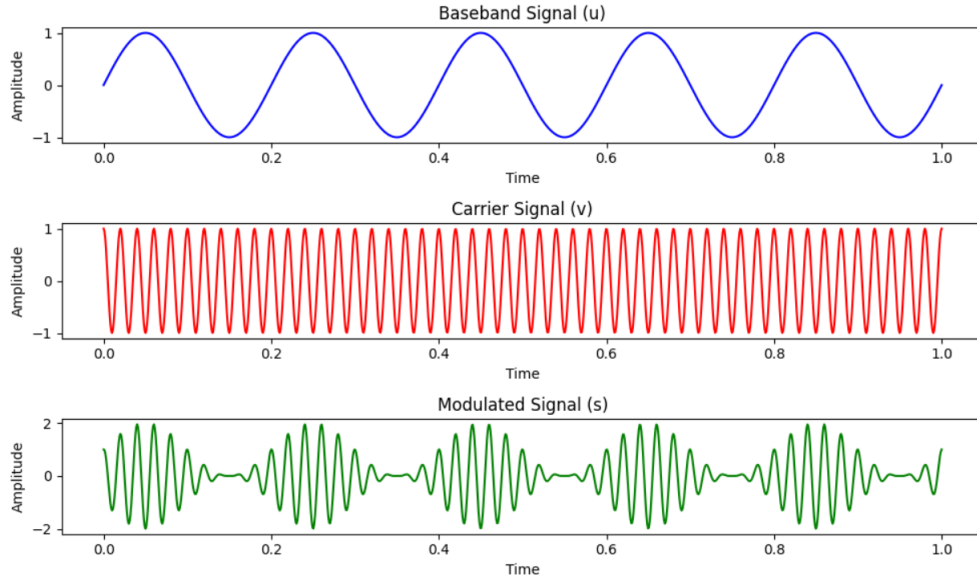


FIGURE 1.1 – Exemple de modulation en amplitude

La modulation en fréquence permet d'obtenir des transmissions de meilleures qualités plus résistantes à leur environnement tout en gardant une puissance d'émission constante.

Soient $u(t)$ un baseband signal et $v(t)$ un carrier signal, le signal modulé en fréquence $s_{fm}(t)$ est le résultat suivant :

$$u(t) = \sin(2\pi f_u t) \quad (1.3)$$

$$v(t) = \cos(2\pi f_c t + \phi_c) \quad (1.4)$$

$$s_{fm}(t) = \cos(2\pi f_c t + \Delta f \cdot u(t) \cdot t + \phi_c) \quad (1.5)$$

Prenons par exemple

$$u(t) = \sin(2\pi f_u t) \text{ avec } f_u = 5 \text{ Hz,}$$

$$v(t) = \cos(2\pi f_c t + \phi_c) \text{ où } f_c = 50 \text{ Hz.}$$

La Figure 1.2 montre le signal modulé $s_{fm}(t)$ via la modulation en fréquence pour une phase initiale du carrier signal $\phi_c = 0$ avec une dérivation en fréquence $\Delta f = 10\text{Hz}$.

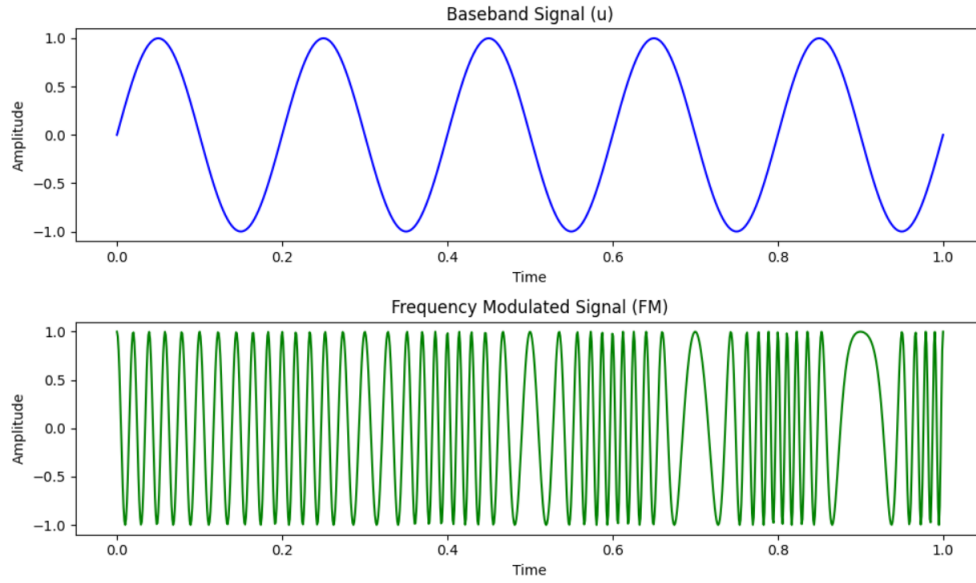


FIGURE 1.2 – Exemple de modulation en fréquence

La modulation en phase permet généralement d'obtenir une meilleur utilisation de la bande passante que les autres modulations car les variations de phase peut encoder plus d'informations, ce qui augmente la quantité de données transmises.

Soient $u(t)$ un baseband signal et $v(t)$ un carrier signal, le signal modulé en phase $s_{pm}(t)$ est le résultat suivant :

$$u(t) = \sin(2\pi f_u t) \quad (1.6)$$

$$v(t) = \cos(2\pi f_c t + \phi_c) \quad (1.7)$$

$$s_{pm}(t) = \cos(2\pi f_c t + K_p \cdot u(t)) \quad (1.8)$$

Prenons par exemple

$$u(t) = \sin(2\pi f_u t) \text{ avec } f_u = 5 \text{ Hz,}$$

$$v(t) = \cos(2\pi f_c t + \phi_c) \text{ où } f_c = 50 \text{ Hz.}$$

La Figure 1.3 montre le signal modulé $s_{pm}(t)$ en phase pour une phase initiale du carrier signal $\phi_c = 0$ avec un index de modulation de phase $K_p = 8$.

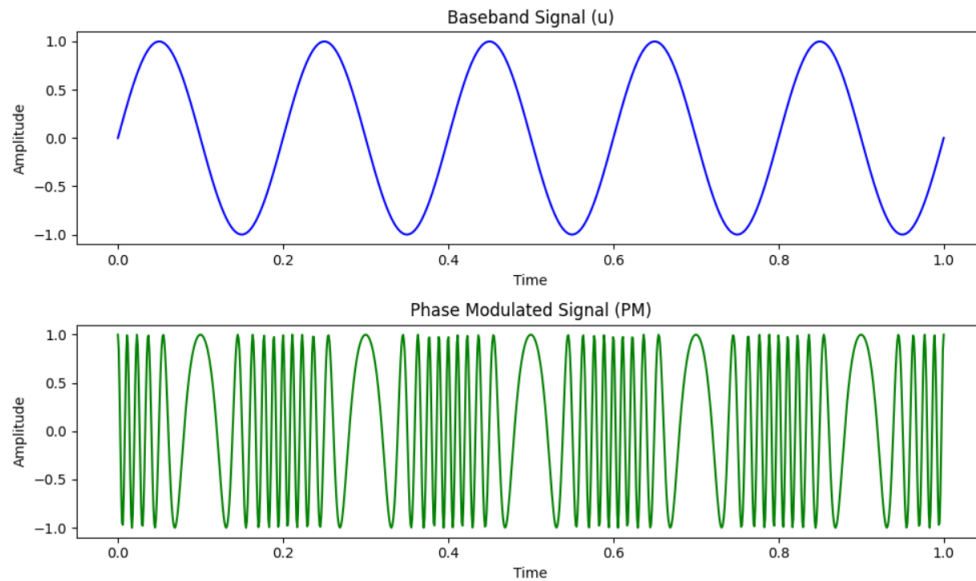


FIGURE 1.3 – Exemple de modulation en phase

1.2.2 Gestion du bruit

L'un des attributs cités concerne le bruit. Un signal est toujours affecté de petites fluctuations plus ou moins importantes, et dont les origines peuvent être diverses. Ces perturbations, appelée bruit ou *noise* en télécommunication se définissent par l'altération non souhaitée de l'intégrité d'un signal. Le bruit peut prendre différentes formes, des perturbations essentiellement impulsionnelles engendrées par des commutations de courants ou alors du bruit de fond généré dans les câbles et les composants électroniques en raison des mécanismes statistiques de la conduction électrique. Il est possible de réduire voir éliminer l'influence des perturbations impulsionnelles. En revanche, le bruit de fond est lui irréductible. Tout signal sans bruit n'existe pas, même à l'émission. Il est cependant possible que le bruit devienne invisible si son niveau est très faible. L'attribut SNR est donc un critère de la qualité du signal.

1.2.3 Transformée de Fourier

Pour effectuer une analyse de signal, sa représentation est capitale. Les Figure 1 et 2 représentent des signaux en fonction du temps écoulé. Il est possible de représenter des signaux selon une autre composante, la fréquence.

La transformée de Fourier est un outil fondamental utilisé pour analyser et décomposer des signaux complexes en composantes fréquentielles. En trans-

formant un signal dans le domaine temporel en sa représentation dans le domaine fréquentiel, la transformée de Fourier révèle les différentes composantes fréquentielles présentes dans le signal. En fonction du type de signal, la transformée de Fourier est adaptée.

Pour les signaux continus, la *CFT* (Transformée de Fourier continue) convertit une fonction du temps en fonction de la fréquence en intégrant le signal par rapport aux sinusoides de toutes les fréquences possibles. Cette transformation fournit les informations d'amplitude et de phase pour chaque composante de fréquence présente dans le signal.

Pour les signaux discrets et échantillonnés, la *DFT* (Transformée de Fourier discrète) calcule un ensemble fini de composantes de fréquence. Il est calculé à l'aide d'un nombre fini d'échantillons, ce qui donne des composantes de fréquence discrètes. Il existe une méthode simplifiée pour les signaux discrets appelé *FFT* (Fast Fourier Transform). Il s'agit d'un moyen plus rapide de calculer la transformée de Fourier, en particulier pour les signaux numériques comportant un grand nombre de points de données. L'avantage principal de cet algorithme permet de réduire le temps de calcul en divisant la DFT en sous problèmes. La FFT est une méthode très utilisée pour l'analyse de signaux.

1.3 LoRa

LoRa (Long Range) est une technologie de communication sans fil qui permet de transmettre des données sur de longues distances avec une faible consommation d'énergie. Elle a été développée par la société française Cycleo et est maintenant gérée par la fondation LoRa Alliance, qui regroupe plusieurs entreprises et organisations du monde entier.

LoRa est principalement utilisée dans l'IoT. Elle se distingue par sa portée étendue, qui peut atteindre plusieurs kilomètres en milieu urbain et plusieurs dizaines de kilomètres en milieu rural, ainsi que par sa faible consommation d'énergie, qui permet de prolonger la durée de vie des appareils connectés. Une longue portée avec une puissance limitée induit une plus faible bande passante que les autres technologies sans fil (le Wifi, la 4G, Bluetooth etc).

LoRa utilise une bande de fréquences qui varie selon les régions du monde où LoRa est déployée :

- en Europe, la bande de fréquences autorisée est comprise entre 863 et 870 MHz, ce qui correspond à l'*ISM radio band*, un bande dédié aux recherches qui ne nécessite pas de licence d'émission.
- aux États-Unis, elle se situe entre 902 et 928 MHz,
- en Chine, la fréquence autorisée varie entre 779 et 787 MHz,
- les régions restantes ont elles aussi une fourchette unique.

La technologie LoRa utilise la modulation appelé *Chirp Spread Spectrum Modulation* (CSS). La modulation CSS utilise un signal chirp, c'est à dire un signal modulé en fréquence linéaire. Ce signal a une amplitude constante mais balaie tout le spectre de la bande passante de manière linéaire dans une période de temps définie. Cette technique de modulation sera détaillée à la section 1.3.1.2

La technologie LoRa utilise également une technique de multiplexage en temps partagé (*Time Division Multiple Access*) pour permettre à plusieurs appareils de partager la même bande de fréquences de manière à maximiser l'utilisation de la capacité de transmission. Elle utilise également une technique de diffusion de données (*multicast*) pour envoyer les mêmes données à plusieurs appareils simultanément, ce qui permet de réaliser des économies de bande passante et d'énergie (source : LoRa Alliance)

En plus de sa portée étendue et de sa faible consommation d'énergie, LoRa se distingue par sa sécurité de transmission, qui est assurée grâce à l'utilisation de codes de sécurité uniques et à la possibilité de chiffrer les données transmises. Lora n'est pas exclusivement lié au protocole LoraWan. Ce protocole sera décrit en détails à la section 1.3.2. Si LoRa opère à un niveau plus bas que la plupart des protocoles réseau, LoRaWan via son infrastructure (notamment les *gateway*

permet entre autre aux appareils LoRa de pouvoir utiliser différents protocoles et d'être compatibles avec un grand nombre de protocoles de communications comme *TCP/IP* (*transport layer protocol*), *HTTP* (*hypertext transfer protocol* ou *MQTT* (*message queuing telemetry transport*)).

Toutes ces particularités font de LoRa une technologie complémentaire à celles déjà existantes plutôt que rivale. LoRa se compose de deux éléments principaux : la couche physique de la technologie et LoRaWAN, la couche MAC (*Media Access Control*, une sous couche de la couche liaison de données dans le modèle OSI *Open Systems Interconnection*). la couche physique de LoRa gère la fréquence radio ainsi que la modulation. LoRaWAN gère les aspects réseaux comme la sécurité, la propagation, l'adressage et la sécurité.

1.3.1 couche physique LoRa

1.3.1.1 Découpage de la couche physique

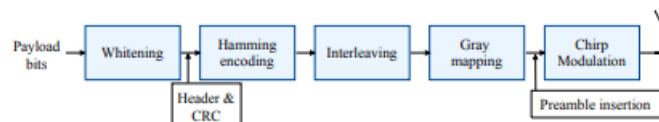


FIGURE 1.4 – Etapes de la transformation des données dans un émetteur LoRa

Les étapes de la conception de l'envoi de données dans la couche physique de LoRa sont les suivantes :

- Le codage de canal est une technique utilisée dans les systèmes de communication sans fil pour améliorer la robustesse et la fiabilité de la transmission des données. Dans le cas de LoRa, le codage de canal utilise la méthode *FEC* (*Forward error correction*) pour corriger les erreurs causées par du bruit. La méthode FEC ajoute de l'information redondante sur les données.
- Le mélange de canal (ou *channel interleaving*) suit la codage de canal. Cette technique consiste à réarranger les bits de data avant de les transmettre, en les intercalant entre eux de manière à les disperser sur le spectre des fréquences de la transmission. Cela permet de réduire l'impact des erreurs consécutives (*burst error*).
- Le blanchiment de canal (ou *channel whitening*) est la dernière étape avant la modulation du signal. Cette technique consiste à utiliser une transformation aléatoire ou pseudo-aléatoire des données avant de les transmettre, de manière à répartir le spectre des fréquences de la transmission sur une large gamme de fréquences. Cela permet également la

récupération d'horloge pour le récepteur.

- La modulation CSS est l'étape principale de LoRa. En effet, les étapes précédantes sont communes à de nombreuses technologies, mais la particularité de LoRa provient de la modulation. Cette étape est détaillée dans la section suivante du chapitre.

Chacune des étapes décrites doit être inversement réalisée pour le récepteur. Ainsi pour la récupération de donnée à l'arrivée, l'appareil récepteur gère la démodulation, le déblanchiment, le démellement et de décodage.

Cette analyse a été faite en *reverse engineering* ([citer article](#)). Le reverse engineering consiste à analyser un produit ou un système afin de comprendre comment il fonctionne ou d'identifier ses principes de conception. Dans le contexte de LoRa, le reverse engineering examine la technologie derrière LoRa afin de comprendre ses principes de base et sa conception.

1.3.1.2 Modulation CSS

Contrairement aux modulation classique en amplitude ou en fréquence, la modulation CSS étale le signal sur une large bande de fréquence. La modulation en fréquence est linéaire et utilise des chirps. un chirp est un signal dont la fréquence change en continue tout en conservant une amplitude constante. Il existe deux types de chirps : les *upchirp* et *downchirp*. Dans un upchirp la fréquence augmente avec le temps tandis que dans un downchirp la fréquence diminue.

le signal est donc séparé sur une large bande de fréquence, permettant par exemple plusieurs transmission sans causer d'interférence. La modulation CSS est l'une des principale contribution au fait que LoRa possède une faible consommation et une longue portée. Cette technioque est très bien intégrée aux appareils a faible puissance utilisé par les technologies LoRa.

1.3.1.3 Spreading factor

LoRa permet d'envoyer des paquets sur un longue distance à faible puissance. Selon l'environnement dans lequel les appareils LoRa sont présents, il peut être utile de pouvoir ajuster certaines capacités.

L'étalement, ou *spreading factor* permet de déterminer le taux de variation de fréquence pour un signal. Modifier le spreading factor ajuste différentes propriétés de la communications. Par exemple, si on augmente le spreading factor, les quatre conséquences principales sont :

- l'augmentation de la portée. Comme la largeur de bande est plus large, la communication peut atteindre une portée supérieure.

- Augmentation de la résistance aux interférences. Comme le signal est étalé sur une bande plus large, il y a moins de risque de subir des interférences.
 - Plus petit débit de données. Comme la bande est large, dans un temps défini moins de données sont transmises.
 - Plus faible consommation. Les données transmises à un taux plus faible consomment moins d'énergie, ce qui prolonge la durée de vie des appareils dont l'économie d'énergie est une priorité.
- Diminuer le spreading factor engendre l'effet inverse.

1.3.1.4 Structure d'un paquet LoRa

un paquet LoRa est structuré en 5 parties différentes :

- Le préambule : la première partie du paquet, composée d'un nombre variable d'upchirps. La valeur par défaut est fixée à 8 upchirps.
- L'identificateur réseau : après le préambule, le paquet contient deux symboles modulés pour l'identification réseau.
- Des symboles de synchronisation de fréquence. Après l'identificateur réseau, il y a des downchirps pour faire la distinction entre les offsets d'échantillonnage de temps ou de fréquence.
- La tête (*header*) du paquet. Elle contient les informations relatives à la taille du paquet, le code rate, la présence ou non d'un CRC (*cyclique redundancy check*) et une checksum.
- Le payload. La dernière partie du paquet contient le payload d'une taille maximale de 255 bits et l'éventuelle CRC de 16 bits.

1.3.2 LoRaWAN

LoRaWAN est un protocole de type *low power, wide area network* (LP-WAN) désigné pour la communication longue portée. Ce protocole opère avec la technologie LoRa et lui fournit une infrastructure capable de maintenir une communication à longue portée et à faible coût dans l'*IoT*.

1.3.2.1 aspects généraux de la technologie

LoRaWAN bénéficie donc d'une faible puissance de consommation et d'une portée accrue. Elle est également efficace dans différents environnements. Le signal est capable de pénétrer divers terrains et structures. Le déploiement d'une infrastructure LoRa ne nécessite pas de licence, et son réseau peut être public ou privé.

Le cœur de LoRaWAN réside sur la gestion de l'énergie, permettant aux appareils de fonctionner avec une consommation d'énergie minimale, prolongeant

leur durée de vie tout en garantissant une fonctionnalité à long terme. A cette caractéristique de faible consommation d'énergie s'ajoute ses capacités en termes de portée, capable de pénétrer divers environnements. Cela rend la technologie efficace aussi bien milieu rural qu'urbain. LoRaWAN opère sur une bande de fréquence qui ne nécessite pas de licence d'émission, par exemple sur la bande ISM pour *Industrial, Scientific, and Medical*. Les bandes ISM, (868 MHz en Europe ou 915 MHz aux USA) sont disponibles pour l'utilisation de différentes technologies, incluant LoRaWAN.

LoRaWAN possède des capacités de géolocalisation, permettant au réseau de détecter et de localiser précisément les appareils au sein de son domaine. LoRaWAN utilise différentes méthodes pour localiser ses appareils comme **Received signal strength indication** ou RSSI, *Time difference on Arrival* ou TDOA, une triangulation ou alors une combinaison de plusieurs des méthodes.

LoRaWAN utilise des protocoles de sécurité end-to-end, aussi bien dans un réseau public intégré que dans un réseau privé. L'architecture LoRaWAN (décrite en détails dans la section 1.3.2.2) contient plusieurs couches de sécurité. Au niveau des *end devices*, une routine d'identification est imposée avant l'accès au réseau. Seul les appareils de confiance sont donc autorisés à communiquer. Ensuite, une fois la communication commencée, les données sont chiffrées avant d'être transmises dans le réseau. Le framework sécuritaire de Lora ne se limite pas à l'authentification et au chiffrement. LoRaWAN gère également la mise à jour en continu par les airs, ainsi qu'une supervision continue sur d'éventuelles intrusions.

Avec toutes ces caractéristiques, LoRaWAN s'est développé dans de nombreux domaines aussi bien environnementaux qu'industriels. Les principales utilisations de LoRaWAN actuelles sont les suivantes :

- la surveillance environnementale en général. LoRaWAN peut être déployé pour surveiller des niveaux de températures, d'humidité, de bruits ou encore d'autres paramètres dans n'importe quel milieu. Une compagnie Hollandaise, Sensoterra, utilise notamment LoRaWAN pour surveiller la qualité des sols.
- Les *smart cities*. LoRaWAN est actif sur différents aspects comme la gestion intelligente de l'éclairage, la gestion des déchets, la surveillance, etc.
- l'embarqué industriel. La maintenance et la surveillance de matériel et de l'équipement peut être gérée par LoRaWAN ? TataSteel, une compagnie indienne, utilise LoRaWAN pour ces équipements industriels.
- la prévention de catastrophe naturelle. Que ce soit en prévision, pendant ou après d'éventuelles catastrophes naturelles, la longue portée et la surveillance en temps réel sont des atouts cruciaux pour ce genre d'évènement.

Cependant, toutes ses caractéristiques entraînent un certain nombre de li-

limitations. La restriction de la fréquence en fonction de la région peut rendre le déploiement d'une même infrastructure à différent endroit dans le monde plus difficile. Cela peut aussi entraîner des problèmes de compatibilité entre régions, notamment pour des chaînes logistique ou d'approvisionnement qui traverse plusieurs régions.

Une faible consommation de puissance avec une grande portée a un impact sur la taille et la vitesse de l'information. La taille du payload d'un message est limitée entre 51 et 241 octets. La vitesse de transmission est également peu élevée, atteignant un maximum de 5.5kbps sur une largeur de bande de 125hz.

la communication au sein d'un réseau LoraWan se fait en grande partie de manière asynchrone. La synchronisation dépend de la classe de l'appareils, qui est détaillé dans la section topologie. C'est un avantage pour maintenir une grande autonomie de batterie pour les appareils. LoraWan possède un système pour limiter les collision entre message si plusieurs appareils communiquent simultanément. Ce système est basé sur une combinaison entre *Listen before talk* LBT et des délais aléatoires (citer papier). Il est néanmoins possible que dans un environnement très dense des collisions puisse encore se produire. La communications asynchrone et le système d'évitement de collision entraine une augmentation du temps entre les envois et la réception de message.

1.3.2.2 Topologie de Lorawan

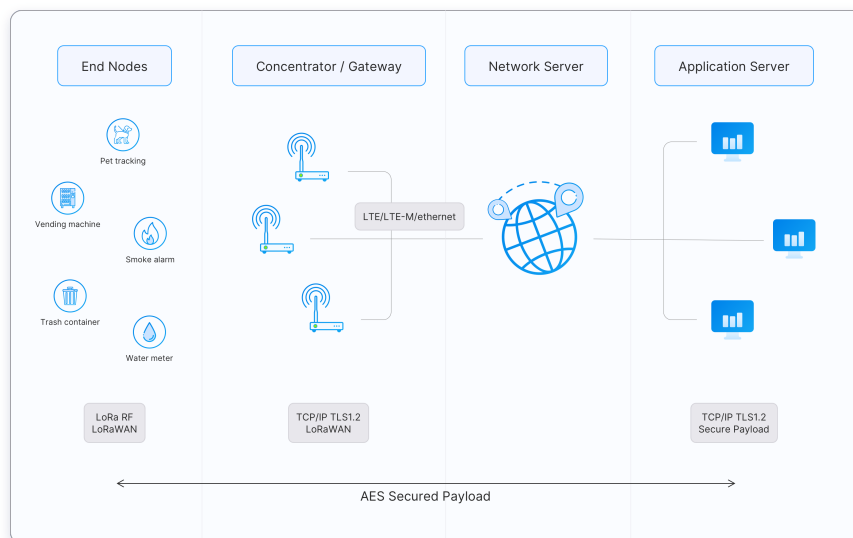


FIGURE 1.5 – Topologie de l'infrastructure LoraWan

La figure 4 montre les 4 types d'appareils qui composent la topologie d'un infrastructure LoraWan.

Les end devices sont les noeuds qui collectent les informations à envoyer à travers le réseau. Ils sont catégorisés en trois sous classes : A, B et C. Les appareils de classe A sont les plus économe en énergie. Ils ont été créé pour conserver leur énergie et comunique exclusivement en communication asynchrone. Les appareils de classe A écoute les message provenant des serveur uniquement après avoir eux même transmis un message. la classe A regroupe les appareils les moins énergivore. Les appareils de classe B sont assez similaires avec ceux de classe A, mais sont occasionnellement synchronisés avec les servers du réseau. Ils possèdent des capacités supérieur de reception leur permettant de se synchroniser avec le schedule des serveurs, ce qui augmente considérablement l'efficacité du temps de réponses dans le réseau. Finalement, les appareils de classe C sont en écoute permanente de message provenant des serveur. Ils sont plus les plus réactif mais également les plus énergivores. Les end devices sont donc classés selon deux paramètres : leur réactivité et leur consommation d'énergie. En fonction de eur classe, ils ont également la possibilité de recevoir des message server après avoir transmis de l'information. L'envoie d'un message d'un end devices vers les serveurs est appelé *uplink message* et l'envoie d'un message depuis les serveur vers les end devices est appelée *downlink message*.

Les gateways font le role d'intermédiaire entre les end devices et le serveur réseau. Ils recoivent les transmissions depuis les end devices dans leur zone de couverture et forward les messages vers le serveur réseau. Les gateways peuvent écouter plusieurs fréquence simultanément (*multichanneling*) là où les end devices n'écotentent qu'une seule fréquence. Les gateway gèrent la communication radio avec les end devices en utilisation la modulation de LoRa.

Le serveur réseau est la composante centrale de l'infrastructure. Il gère tout le réseau, que ce soit les données reçu des gateways, l'identification et l'activation des end devices dans le réseau, le routing ou encore l'adapation du data rate. Le serveur réseau supervise également l'aspect sécurité au sein du réseau en gérant les clefs de chiffage et les protocoles de sécurité.

Le serveur application de LoraWan reçoit les données forwardés depuis le serveur réseau. C'est l'interface entre le réseau de LoraWan et les différents service ou application d'utilisateurs finaux. Les utilisateurs interagissent avec le serveur d'application pour n'importe quelle action à effectuer sur le réseau ou pour la récupération de données du réseau. Les données reçues par le serveur réseau sont traduite par le server d'application avant d'être interprétée par l'utilisateur final.

1.3.2.3 Sécurité

La sécurité dans l'architecture LoraWan se concentre sur trois axes principaux :

- l'authentification : qui communique avec qui.
- L'intégrité : les données ne sont pas altérées entre l'émetteur et le receveur.
- La confidentialité : les données ne sont visibles par personne au sein du réseau hormi l'émetteur et le récepteur.

La sécurité repose sur le chiffrement des données. Les données sont chiffrées en utilisant l'algorithme de cryptographie AES (*advanced encryption standard*). La taille des clés est de 128 bits. Ce choix est motivé par un équilibre entre une sécurité suffisante et une consommation réduite de ressources.

Il y a deux types de clés utilisées dans LoraWan. La *root key* est la clé partagée entre un end device et le serveur réseau. Cette clé est utilisée pour l'authentification initiale et l'établissement d'une communication entre les deux éléments du réseau. Cette clé n'est jamais transmise par l'air, elle est stockée dans un *join server*. Un join server est un serveur dédié au contenu sensible à l'activation du matériel dans un réseau LoRaWAN. Il authentifie le réseau et les applications des serveurs. Il gère les *root keys*, il génère également le second type de clés de LoraWan, les *session keys*.

Les *session keys* sont des clés générées dynamiquement par le join server et utilisées durant l'échange de données pendant une session. Il y a deux *session keys* différentes, la *AppSKEY* pour le chiffrement des *payload* d'application, et la *nwkSKEY* pour les fonctionnalités du réseau (le chiffrement à la couche MAC, la vérification d'intégrité, etc).

1.3.2.4 Session

L'établissement d'une session entre un end device et le réseau LoraWan peut se faire de deux façons différentes.

La première méthode est une méthode dynamique appelée *Over the Air Activation* ou OTAA et se déroule de la façon suivante :

- Le device possède initialement deux identifiants, un DevEUI et un appEUI.
- La requête pour rejoindre le réseau est initiée par le end device. Il envoie un message *join request* au serveur réseau. La *join request* contient ses identifiants, ainsi qu'un nombre aléatoire généré par le device.
- Le serveur réseau accepte (ou décline) la requête et vérifie les identifiants du device dans ses enregistrements.
- le serveur génère ensuite un nombre aléatoire appelé *DevNonce* et renvoie

un message *join accept* contenant le DevNonce, l'adresse du device ainsi que les clefs (NwkSKey et appSKey) de session.

- le device reçoit le message *join accept*. Il extrait les clefs envoyées et calcule ses propres clefs de session avec ses paramètres (les clefs envoyées par le *join server* ainsi que le devNonce).
- le device fait maintenant partie du réseau. Chaque message que le device va envoyer au serveur sera chiffré avec ses clefs.

La seconde méthode est hardcodée et permet à un end device de rejoindre directement le réseau sans passer par l'identification. Cette méthode est appelée *activation by personalisation* ou ABP. Voici la procédure de la session :

- Le device possède à l'avance son adresse ainsi que ses clefs de session.
- Le device est déployé au préalable dans la zone de couverture du réseau LoraWan.
- Sans avoir à initialiser la procédure *join request*, le end device transmet directement ses données au serveur en utilisant ses clefs préconfigurées.

L'échange de clef avec le serveur n'a pas lieu.

Cette seconde procédure a comme avantage d'être plus rapide à exécuter car toute la partie d'initialisation est passée. Le processus d'initialisation peut être contraignant en ressource ce qui rend la méthode ABP moins énergivore. Cependant l'utilisation de clef hardcodée directement dans les devices est une pratique moins sécuritaire. Comme pour la taille de clefs, il y a un équilibre entre consommation d'énergie et sécurité.

Chapitre 2

Travaux similaires et autres contributions

Ce chapitre a pour but d'établir un état de l'art dans le domaine de l'internet of things. Comme l'iot est très vaste, seulement certains aspects seront présentés. Tout d'abords, il est important de comprendre que la sécurité dans ce domaine a évoluer autant bien par ces méthodes que par son importance, ainsi un historique est présenté dans la première section du chapitre. Ensuite, la seconde section se concentre uniquement sur la technologie Lora. Cette section donne une approche analytique de signaux. Finalement, la dernière partie de ce chapitre se consacre aux approches existante pour l'identification d'appareils utilisant Lora. Cette section présentera les travaux qui sont au centre des expérimentations de ce travail.

2.1 identification de device dans l'iot

Avant de s'intéresser à l'identification d'appareil Lora, il est nécessaire d'étendre ce concept à l'iot. L'internet of things a connu une forte évolution depuis les début des années 2000, avec des priorités dans son évolution qui ont également changé au fil du temps.

2.1.1 historique et évolution des préoccupations de sécurité dans l'iot

Afin de bien comprendre les enjeux des différentes époques, voici un petit historique des deux décennies précédentes. Les technologies mais également le concept même d'internet of things ont évolué.

Les premières années de l'iot (2000) Bien que le concept d'appareils connectés remonte aux années 70, l'avènement de l'internet of things arrive en fin de millénaire.

Ce concept est associé à la technologie RFID *Radio Frequency Identification*, qui permet d'utiliser les ondes radios afin d'identifier des objets ou des personnes. Le but initial était de rendre tout objet dans le monde identifiable par un code EPC ou *Electronic Product Code*, un peu comme le code barre. Durant ses années, plusieurs entreprises lancent leur première appareils connectés. Tout cet enthousiasme pour la connectivité met au second plan les questions de sécurité. Ainsi la première partie du développement de l'iot se concentre surtout sur la qualité de communication entre les objets plutôt que sur leur sécurité.

Les premières préoccupations sécuritaires (2010s) Vers la fin des années 2000, l'augmentation du nombre d'appareils est si grande qu'elle a atteint tous les domaines de la société. Certains domaines étant plus critique que d'autre d'un point de vue sécurité (l'énergie, les transports, la santé, etc), l'intégrité des données, la confidentialité et les accès réseaux deviennent le centre de l'attention. Le concept de certificats x.509, initialement développé pour le world wide web avant les années 2000, a un regain d'attention dans cette période. Plus largement la structure de la technologie PKI (*Public Key infrastructure*, qui utilise les certificats x.509) a été adaptée pour s'intégrer aux problématiques de l'embarqué. Un certificat est un document digital permettant de vérifier d'identité d'une entité, comme d'un appareil, un utilisateur ou une organisation. Il se base sur la liaison d'une clé public à l'entité établie par une *Certificate Authority* ou CA. La CA agit en temps de tier de confiance et assure la légitimité de l'information grâce au certificat. Ainsi, les trois axes principaux de la sécurité dans l'iot émergent : l'authentification, l'intégrité des données et la confidentialité.

L'avènement du *edge computing* (année 2010-2020). Le nombre d'appareils connecté dépassé le nombre d'humains, forçant une transition vers l'ipv6 tant le nombre d'appareils est élevé et continue d'augmenter. L'information a pris de la valeur, et de l'ampleur. Ainsi, viens se greffer de nouveau enjeux économiques en plus des enjeux sécuritaires. La quantité de donnée générée nécessitent de revoir le stockage de l'information. C'est ainsi que va apparaître le Edge computing, qui est une réponse directe aux besoins des architectures de gérer autant de données en périphérie de réseau. Le concept du edge computing vise à effectuer des calculs et des analyses des données directement sur les appareils connectés, plutôt que de les envoyer vers un centre de données centralisé. Cela réduit la latence, améliore l'efficacité du réseau et permet des analyses en temps réel. Le premier malware spécialement centré sur l'iot fait son apparition. Mirai exploite une faille lui permettant de récupérer les mots de passe d'appareils afin de s'en servir pour lancer des attaques DDoS *distributed denial of service* à grande échelle. En quelques années l'iot est passé d'un gadget d'entreprise à un véritable enjeu économique et sécuritaire, centré autour de l'information. Les seules perspectives de législation concernant la sécurité de l'internet of thing n'apparaîtront de travers en fin de décennies avec La loi européenne sur le *Règlement générale sur la protection des*

données. cette loi ne couvre pas la sécurité des appareils mais plutôt l'utilisation des données sur internet en général.

L'apparition de la *blockchain* (fin année 2010) Le stockage et la transmission de données, l'authentification d'appareils ou encore la confidentialité sont au centre des préoccupations. Initialement technologie utilisée dans les cryptomonnaies, la technologie Blockchain est un mécanisme de base de données qui permet un partage transparent des informations au sein d'un réseau. Une base de données Blockchain stocke les données dans des blocs qui sont reliés entre eux dans une chaîne. Les données sont chronologiquement cohérentes, car il n'est pas possible de supprimer ou modifier la chaîne sans le consensus du réseau. Par conséquent, la technologie Blockchain peut servir de livre inaltérable ou immuable pour le suivi des ordres, des paiements, des comptes et d'autres transactions. Le système dispose de mécanismes intégrés qui empêchent les entrées de transactions non autorisées et créent une cohérence dans la vue partagée de ces transactions. L'implémentation de la blockchain pour l'IoT confère les avantages suivants :

- l'immutabilité. la blockchain permet de créer un enregistrement immuable de toutes les interactions et communications des appareils. Cet enregistrement peut être utilisé pour détecter et empêcher l'accès non autorisé ou la modification des appareils ou des données dans l'IoT.
- La décentralisation. il est possible de créer un système décentralisé pour l'authentification et la communication des appareils. Chaque appareil IoT se connecte au réseau blockchain et se voit attribuer une identité numérique unique, qui est vérifiée grâce à l'utilisation de signatures numériques ou de contrats intelligents. Cela élimine le besoin d'une autorité centrale pour authentifier les appareils et annule ainsi les risques de *single point of failure*.
- La confidentialité. La technologie Blockchain peut sécuriser la communication entre les appareils IoT grâce à l'utilisation de la cryptographie à clé publique ou asymétrique. Cela permet l'échange sécurisé d'informations entre appareils sans avoir recours à des intermédiaires.

Le *Zero Trust Model* (2020) les menaces de sécurité sont de plus en plus sophistiquées. Comment faire encore confiance aux infrastructures qui doivent gérer autant d'appareils ? La réponse est de ne plus leur faire confiance. Le *zero Trust model* est donc un modèle basé sur l'absence totale de confiance et une vérification constante, que la demande d'accès provienne de l'intérieur ou de l'extérieur du réseau. Dans un modèle de sécurité classique, une fois qu'un utilisateur ou un appareil accède au réseau interne, on lui fait souvent implicitement confiance, ce qui lui permet une grande liberté d'actions au sein du réseau. Le *Zero Trust model* suppose cependant que des menaces peuvent exister à la fois à l'intérieur et à l'extérieur du périmètre du réseau et nécessite donc une vérification continue de la confiance. Un modèle qui s'applique sur ce principe devrait en principe contenir

les éléments suivants :

- La vérification d'identité. Les utilisateur et les appareils doivent subir une authentification avant d'accéder à n'importe quel services ou ressources du réseau.
- Le *Least privilege acces*. les permissions sont accordées de manières limités selon le besoin de l'utilisateur ou de l'appareil.
- La micro segmentation. Diviser le réseau en segment pour limiter son accès par les appareils.
- La surveillance en continue. L'analyse du trafic, du comportement et des activités des appareils.
- le chiffage des données.

Le *quantum computing*(futur ?)

Avec la future arrivée des ordniateurs quantiques, les mécanismes de chiffrage basé sur la comlxité mathématique comme RSA ou ECC sont voués à disparaître. La puissance de calcul des ordinateurs quantique est déjà considérée comme une véritable menace pour la sécurité informatique. Fort heureusement, c'est également un nouveau champ de possibilité qui s'ouvre pour la sécurité, avec le développement du *post quantum cryptography*. Un premier protocole résistant au menaces quantiques, *Quantum Key Distribution* permet d'établir des canaux de communications entre différents appareils dand l'iot. Ce protol n'est pas encore en service dans l'iot, mais les premiers est réalisé en laboratoire sont très prometteurs (cite article).

2.1.2 approches d'identification dans l'iot

Comme tout appareil au sein d'un réseau, un appareil connecté dans l'iot possède de base différent moyen d'authentification. Son adresse MAC, son adresse IP, des informations relatives à sa manufacture comme un numéro de série par exemple. De manière plus spécifiques, les noeuds au sein d'un réseau Lora-WAN possède un DevEUI, un numéro unique accordé par le réseau à l'appareil. Cependant ses informations peuvent être compromises si appareils sont victimes de *devices spoofing*, c'est à dire qu'un appareil malveillant usurpe l'identité de sa cible, afin d'accéder au sein du réseau. D'autres attaques comme *Man in The Middle* ou *Replay attacks* peuvent également compromettre l'identité d'un appareil si on se base uniquement sur ses identifiant classiques. Il faut donc pouvoir identifier les appareils mais sans se fier à leur informations. Il existe diverse méthodes basée sur différentes approches pour pouvoir indentifier un appareil.

La première approche possible est de fier non pas à l'appareil directement ni aux données qu'il reçoit car celles ci pourraient également être compromises, mais à la routine sur sa communication. Cette approche, appelée *Traffic related pattern* où s'intéresse au comportement d'un appareil au sein d'un réseau. L'ar-

ticle publié par (citer article) propose notamment d'analyser via machine learning le *traffic pattern* c'est à dire le comportement du trafic.

Une autre approche s'intéresse à la position géographique d'un appareil. en effet il est possible de mesurer la qualité de la réception d'un signal, le *Received Signal strength Indicator* ou RSSI. Une étude a été menée sur des appareils lora par (citer article). L'article utilise le *Path Loss* pour estimer le RSSI. Via machine learning ils sont capables de recréer un système de positionnement permettant l'identification d'appareil Lora.

La dernière approche se concentre sur l'analyse des caractéristiques uniques des fréquences radios. Plutôt que de s'intéresser à la routine entre les communications ou la distance d'où elles ont lieu, le *Radio Frequency Fingerprinting* se concentre sur les propriétés physiques des signaux.

2.2 analyse de la technologie lora

2.2.1 analyse de la couche physique lora

des travaux ont été réalisés pour permettre l'analyse de la couche physique de lora (article) fait en reverse engineering.

2.2.2 analyse de la modulation CSS de Lora

papier belge sur la modulation CSS

2.3 identification de device lora

radio frequency fingerprinting identification (RFFI). trouver des caractéristiques hardware pour identifier des devices.

2.3.1 RFFI avec DCTFs

article sur les méthodes de constellations traces

2.3.2 RFFI avec spectrogrammes

article sur les spectrogrammes.

Chapitre 3

Expérimentations

3.1 Matériel

3.1.1 radio logicielle

La radio logicielle (*SDR*, pour *Software-DefinedRadio*) est une technologie qui permet de mettre en œuvre des systèmes de radio à l'aide de logiciels plutôt que de matériel.

Dans les systèmes de radio traditionnels, les différentes fonctions de la radio, comme l'accord sur une fréquence spécifique, la modulation et la démodulation du signal, et le filtrage du bruit, sont mises en œuvre à l'aide de composants matériels tels que des oscillateurs, des amplificateurs et des filtres. En revanche, les systèmes SDR utilisent des logiciels pour effectuer ces fonctions, ce qui les rends beaucoup plus flexible car chaque composante est reconfigurable. Les radios logicielle sont capable d'opérer sur une large portée de fréquence, aussi bien très basse fréquence comme haute fréquence. Les *SDR* peuvent jouer le rôle d'émetteur ou de récepteur voir les deux.

3.1.1.1 RTL-SDR

La première radio utilisée comme récepteur, possède différentes composantes :
 rtl2832U : digitalise les signaux RF et les envoie à l'ordinateur. Tuner chip : le tuner permet d'ajuster la fréquence. Grâce à ça la sdr peut couvrir une large portée. port usb : pour raccorder la sdr à l'ordinateur.

3.1.1.2 hackRf

autre radio logicielle. plus (cher et) complète. meilleure qualité de signal que les rtl-sdr "classiques".

3.1.2 Module d'émission Lora

3.1.2.1 module RN2483

Le microchip RN2483 est un module de technologie spécifique à LoRa. Cet appareil permet de communiquer à longue portée et à faible consommation grâce à l'utilisation de la modulation basée sur LoRa.

quelques spécificités du module :

technologie LoRa faible puissance (idéale pour de l'IoT car faible consommation) fréquence à 433, 868 et 915MHz (regarder la région adéquate) AT command : configurable via un set de commande compatible avec le protocole LoRa-WAN pour établir ou rejoindre ce type de réseau.

fonctionne par entrée de commande (aucun retour écran donc aucune faute possible)

différentes commandes/ utilisation :

sys get ver : demande la version du module, reçoit en réponse radio set (param) (value) : ajuste le paramètre pour l'adapter à la valeur souhaitée.

3.1.2.2 pycom fipy

besoin d'un environnement python, qq configuration nécessaire.

3.1.2.3 module arduino

besoin d'un IDE arduino, possibilité de configurer une largeur de bande bien plus faible que pour les autres modules, très pratique pour analyser le signal en détail.

3.1.3 logiciel

3.1.3.1 GNU radio

GNU Radio est un toolkit qui permet de créer des flux de traitement de signal en utilisant des blocs prédéfinis. Ces blocs peuvent être combinés pour créer des chaînes de traitement de signal pour simuler des modulations CSS, capturer des signaux et en extraire des séquences de chirp.

3.1.3.2 gqrx

logiciel open source d'analyse de fréquence radio pour les SDR.

installer gqrx via apt. (ubuntu)

sélectionner le périphérique pour analyse

image choix périphérique

visualisation du spectre
deux forme d'affichage, en spectre et en cascade.

L'affichage du spectre fournit une représentation graphique en temps réel du spectre RF sur une gamme de fréquences. Il montre la puissance du signal de différentes fréquences sur une plage de fréquences spécifiée. L'axe des x représente la fréquence, tandis que l'axe des y affiche la force du signal (mesurée en dB).

L'affichage en cascade est un spectrogramme qui visualise la force du signal au fil du temps. Il montre une série d'instantanés de spectre empilés les uns sur les autres, où l'intensité de la couleur représente la force du signal. Chaque ligne horizontale du tracé en cascade représente une vue du spectre capturée à un moment précis, créant ainsi un enregistrement historique de l'activité du signal. L'axe vertical représente la fréquence et l'axe horizontal représente le temps.

image affichage spectre

configuration de la réception :

input control (pas trop touché)

FFt settings : très important règle la ff size, le rafraichissement d'image. le laps de temps. l'averaging

Le paramètre Panadapter dB fait référence à l'échelle verticale dans la vue du spectre. Il représente la force du signal des fréquences radio reçues affichées sur l'axe vertical du graphique du spectre. Le réglage du paramètre Panadapter dB modifie l'échelle verticale de la force du signal affichée dans la vue du spectre.

Le paramètre Waterfall dB concerne l'intensité de la couleur ou l'ombrage des fréquences affichées dans le tracé en cascade. Le réglage du paramètre Waterfall dB modifie l'intensité utilisée pour afficher la force du signal dans le tracé en cascade, permettant ainsi d'ajuster le contraste ou la visibilité des signaux plus faibles ou plus forts.

3.1.3.3 Universal radio hacker, URH

logiciel open source. similaire à gqrx pour fonction d'analyse du signal. Il est possible de visualiser les signaux de manière analogique, démodulé, en spectrogramme ou en vue I/Q. Il est possible de découper les signaux, notamment pour supprimer les parties "vides" dans les enregistrements. Possibilité de sauvegarder des signaux enregistrés dans des fichiers. Urh supporte différents formats pour les signaux :

.complex files with complex64 samples (32 Bit float for I and Q, respectively). This is the default signal file format.

.complex16u using two unsigned 8 Bit integers for I and Q

.complex16s using two signed 8 Bit integers for I and Q

.complex32u using two unsigned 16 Bit integers for I and Q (since v2.7)

.complex32s using two signed 16 Bit integers for I and Q (since v2.7).

Il est également possible de lire des fichiers de données qui n'ont pas été enregistré avec URH tant qu'ils sont dans les formats supportés.

3.2 Librairie python

utilisation de python : pourquoi ? librairie très utile dans le domaine comme NumPy, Pandas, Scikit-learn, PyCM et FiPy. librairie compatible avec la sauvegarde et l'utilisation des données. Bonne documentation notamment sur les formats des nombres complexes. Capacités pour le machine learning. Utilisation de diverses algorithmes comme kmeans déjà implémenté dans des librairies. Intégration avec des modules, la librairie fipy gère l'un es modules utilisé pour les expérimentations.

numpy : pour complexe conjugué matplotlib : pour les plot des diagrammes
datashader : pour la coloration du diagramme de constellations

3.3 Génération et réception d'un signal LoRa

dans un premier temps les signaux sont généré manuellement sans automatisation, le but étant de reconnaître et d'analyser la structure d'un signal Lora.

script pour générer un signal depuis un module (soit rn, soit fipy, soit arduino) Il faut paramétrer la largeur de bande, le CR, le spreading factor, la fréquence d'émission (fixé à 868Mhz dans la région europe), le type de modulation (Lora ...)

Ensuite pour la réception, lancer un logiciel d'écoute comme gqrx ou urh. Sélectionner la sdr connecté à l'ordinateur (rtl sdr ou hackrf). Il faut configurer également l'échantillonnage, soit la quantité d'échantillons qui vont être lu chaque seconde. Il faut ajuster la fréquence d'écoute.

3.3.1 analyse du signal

affichage du signal capturé, d'abords sous forme analogique, puis en spectrogramme. Décomposition du signal, on observe des "chirps". En analogique, augmentation de la fréquence (unchirp) et diminution de la fréquence (downchirp). En spectrogramme, augmentation est plus visuelle encore, on voit de manière net les chirps.

Dans le signal, on reconnaît donc le préambule du signal composé de 10 upchirps et 2 downchirps (selon la théorie).

Attention, la fréquence d'écoute des sdr ne doit pas être exactement à 868Mhz. En effet, voilà à quoi ressemble si la fréquence d'écout est la même que la fréquence d'émission. Il faut prendre en compte la largeur de bande du signal, dans le cas ou le singal a une largeur de bande de 125KHz, il faut décaler la fréquence d'écoute

pour recentrer le signal, ainsi on évite d'avoir des fréquences qui sont interprétées comme "négatives" par URH. Dans la figure, la fréquence est décalée de 125/2 soit 867,935Mhz.

3.3.2 automatiser le signal

Dans un second temps, besoin d'automatiser la génération et l'enregistrement des signaux, car il va falloir en générer un grand nombre pour les méthodes d'identification. La première étape consiste à éditer le script pour que la configuration des paramètres soit faite au préalable. Ensuite, sans devoir passer par un logiciel d'écoute, il faut sauvegarder le signal. La librairie RTL-SDR de python permet de configurer la rtl sdr sans devoir passer gqrx ou URH. Il est alors possible avec juste un script python d'écouter et d'enregistrer un signal radio. Il faut ajouter au script une méthode qui supprime dans la partie sauvegardée la section ne comportant pas le signal.

3.4 Méthode "Constellation traces"

objectif, identification du device via son frequency offset.

selon l'article (citer article), possible de performer la méthode soit uniquement sur le préambule, soit sur l'intégralité du signal.

idée : le received signal contient le baseband signal ainsi qu'une rotation factor instable. pour pouvoir récupérer cette partie du signal, besoin d'effectuer une opération différentielle suivante :

$$x(t).x(t + n)e^{-j2\pi n}$$

apparition d'un nouveau rotation factor, mais stable. Besoin de trouver deux inconnues,

textitdelta f et n. n est le differential interval. il se calcule de la manière suivante :

$$R_s = BW/2sf$$

$$N = f_s/R_s$$

delta f est la difference entre le transmitter carrier frequency et le receiver carrier frequency.

application : récupérer les samples I/Q du signal n ayant au préalable "clean" le signal. utilisation d'un gradient pour observer la densité sur le plot. normalement des zones plus denses apparaissent. coloration : utilisation de la librairie data shader.

clustering, le but de conserver les parties les plus dense (95pourcent du point le plus dense) plot avec les différents appareils. librairie panda et numpy. découpage en zones (bins ?) sous forme d'une grille, calcul de nobre de points dans chaque zone, la zone avec le plus grand nombre de point sers de maximum.

Chapitre 4

Résultats

4.1 Méthode des constellations Traces

4.1.1 paramétrage

3 modules lora
module rn2483
module pycom fipy
module arduino
paramètres d'émission : SF = 7, BW = 125, Freq = 868Mhz, CR = 4/5, mod
= Lora
plot des DCTF (differential constellation traces figure)

4.1.2 training phase

during the training phase, you use differential constellation trace figures (DCTFs) from different modules. The objective is to identify clusters in these figures and learn the characteristics of each module.

4.1.3 testing phase

In your case, during the testing phase, you have new DCTFs from modules that the model has not seen during training. The model uses what it learned to predict the category or cluster for each DCTF.

4.1.4 résultats

Annexe A

Première annexe

Annexe B

Deuxième annexe