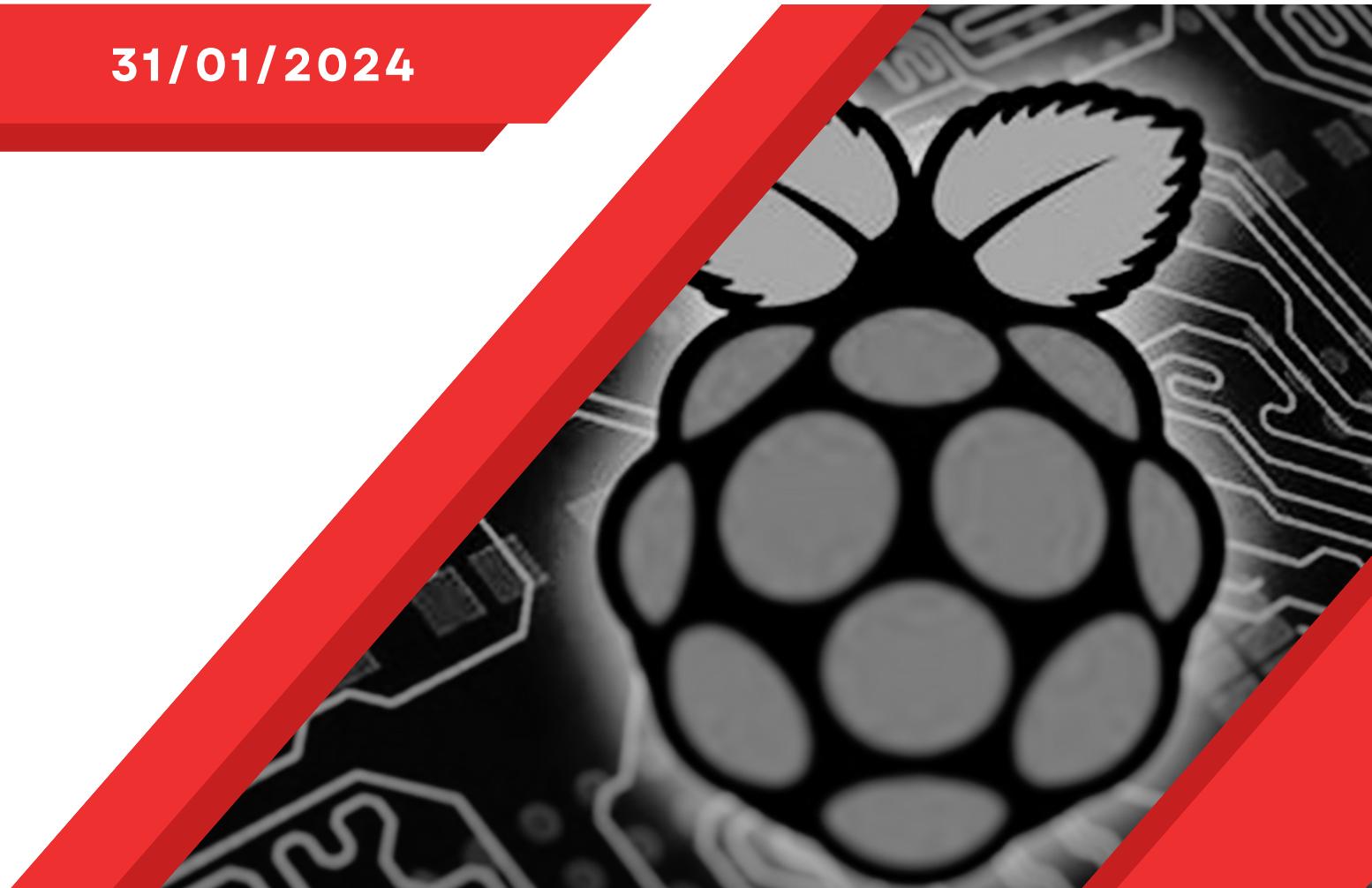


PROJET SYS-COM, INFORMATIQUE ET RÉSEAU

LORAWAN SNIFFING : OUTIL DE COLLECTE DE DONNÉES LORAWAN

31/01/2024**NOÉ CARINGI**

noe.caringi@insa-lyon.fr

HILLEL PARTOUCHE

hillel.partouche@insa-lyon.fr

SOUKAINA KHALIL

soukaina.khalil@insa-lyon.fr

MARGAUX MASSOL

margaux.massol@insa-lyon.fr

Table des matières

I Introduction	3
I.1 Contexte	3
I.1.1 Sniffer	4
I.2 Objectifs	4
II Déroulé du projet	4
II.1 Anticipation	4
II.2 Configuration Raspberry Pi	5
II.3 Création transmetteurs Arduino	5
II.4 Sniffing	5
II.5 Traitement des données par Python	6
III Prototype	6
III.1 Matériel	6
III.2 Code	8
III.2.1 Bibliothèques	8
III.2.2 Architecture	9
III.3 Émetteur	11
IV Bilan du projet	11
IV.1 Résultats	11
IV.2 Limites du prototype	11
V Pistes d'améliorations	14
V.1 Ecoute multi-bande	14
V.2 Wireshark	14
V.3 Amélioration du matériel	15
VI Conclusion	15
Bibliographie	17

I Introduction

I.1 Contexte

- LoRa - Une Technologie de Modulation Performante :

LoRa, abréviation de Long Range, constitue une technologie de modulation radiofréquence qui permet des échanges de signaux longue portée. Cette méthode de transmission se caractérise par une portée étendue, une basse consommation électrique et un débit modéré. Concrètement, LoRa offre une portée théorique de plus de 15 km en zone suburbaine, avec des débits ajustables entre 0,3 et 22 kbps, le tout associé à une puissance d'émission adaptative. (1)

- LoRaWAN - Un Protocole IoT Avancé :

LoRaWAN, ou LoRa Wide Area Network, désigne à la fois un protocole de communication et une architecture réseau spécialement conçus pour répondre aux exigences de l'Internet des Objets (IoT). Fonctionnant sur des fréquences ISM sans licence à l'échelle mondiale (868 MHz en Europe), il s'inscrit dans la catégorie des technologies LPWAN (Low Power Wide Area Network). LoRaWAN permet une communication bidirectionnelle entre les objets connectés et les passerelles, alternant entre la voie montante (Uplink) et la voie descendante (Downlink).

- Protocole LoRaWAN - Interaction Entre Nœuds et Passerelles :

Le protocole d'échange de données LoRaWAN repose sur le protocole LoRa MAC, qui définit l'interaction entre les nœuds (capteurs) et les passerelles. Dans Fig. 1, les nœuds échangent des données avec les passerelles à travers la couche radio LoRa et le protocole LoRa MAC. Les passerelles, connectées à Internet ou à un réseau privé via divers moyens tels que 3G, Ethernet ou WiFi, agrègent les messages des nœuds avant de les transférer vers le serveur de données.

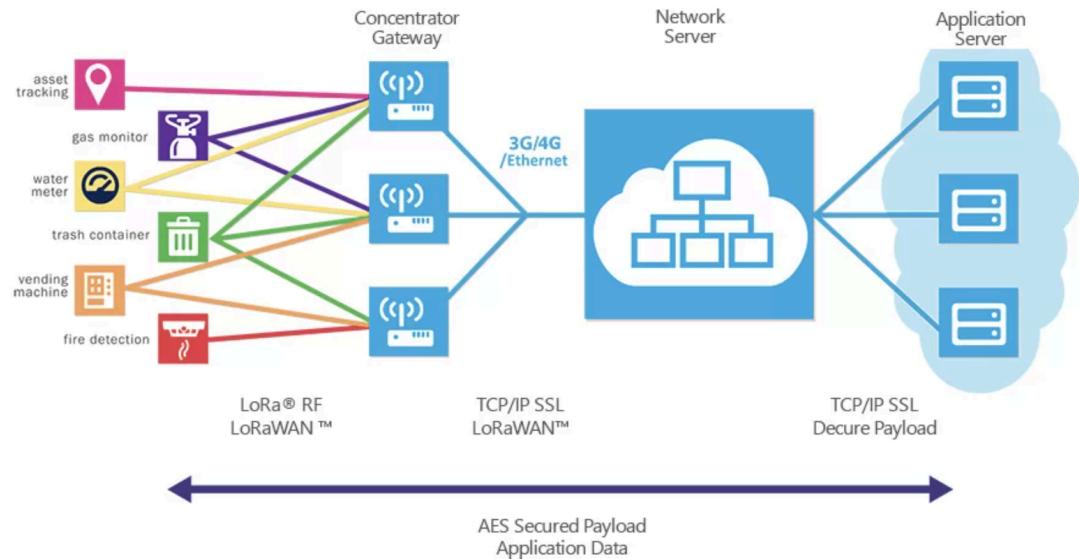


Fig. 1. – Schéma du protocole

- Structure du Réseau LoRaWAN - Agrégation des Données :

La structure du réseau LoRaWAN repose sur une organisation où les nœuds émettent et reçoivent des données via des passerelles. Ces passerelles, disposées stratégiquement, consolident les messages des nœuds et les transmettent vers le serveur de données. Cette architecture permet une gestion efficace des communications au sein du réseau LoRaWAN.

I.1.1 Sniffer

L'écoute passive des messages LoRa, également connue sous le nom de « sniffing », revêt une importance capitale dans la compréhension et la sécurisation des réseaux LoRaWAN. Le sniffer, appliqué avec succès dans des contextes similaires tels que les réseaux Wi-Fi et Bluetooth, offre une opportunité prometteuse pour dévoiler les mécanismes du LoRa et détecter d'éventuelles vulnérabilités spécifiques à cette technologie.

À la manière des pratiques de sniffing employées dans d'autres technologies sans fil comme le Wi-Fi et le Bluetooth, le LoRaWAN requiert des outils spécialisés pour une surveillance exhaustive. Actuellement, l'arsenal d'outils disponibles pour le sniffing LoRa demeure limité sur le plan de l'accès et de l'abordabilité, incitant ainsi le besoin pressant de développer une approche à la fois efficace et économique.

Dans le contexte spécifique du LoRa, caractérisé par sa portée longue distance, son bas débit, et sa faible consommation d'énergie, l'écoute passive revêt une importance particulière pour l'analyse des performances, la cartographie des réseaux, et la détection de potentielles failles de sécurité. Les solutions actuelles, bien que présentes, nécessitent une spécialisation matérielle et logicielle coûteuse, soulignant ainsi le besoin de démocratiser l'accès à des outils de sniffing LoRa plus accessibles et abordables. C'est dans ce contexte que s'inscrit notre projet, visant à combler cette lacune en développant un sniffer LoRaWAN efficace, économique et accessible.

I.2 Objectifs

L'objectif central de ce projet est la conception d'un outil de sniffing LoRaWAN, visant à capturer de manière simultanée toutes les communications échangées sur divers canaux. Les impératifs budgétaires guideront le choix de modules LoRa simples. L'objectif central de notre projet consiste à concevoir un outil de sniffing LoRaWAN qui soit à la fois techniquement efficace et économiquement viable. Nous visons la mise en place d'un dispositif capable de capturer simultanément toutes les communications sur divers canaux, offrant ainsi une vue exhaustive du réseau LoRaWAN. Pour répondre à des contraintes budgétaires, notre choix se portera sur des modules LoRa simples, tout en assurant une prise en charge complète des messages uplink (émanant des dispositifs connectés) et downlink (destinés à ces dispositifs).

La collecte des données se déroulera en temps réel, permettant une réactivité immédiate aux événements du réseau. Ces données seront ensuite transmises vers un hôte dans un format standardisé, tel que le PCAP (Packet Capture), facilitant ainsi leur interprétation ultérieure. Cette démarche technique nous permettra d'obtenir une vision claire et structurée des échanges au sein du réseau LoRaWAN.

II Déroulé du projet

Nous avons débuté le projet par une réunion avec Madame Goursaud qui a pu nous détailler l'ensemble des attendus du projet ainsi que l'histoire de la création du LoRa et les principes de télécommunications derrière son fonctionnement, suivi d'une deuxième réunion avec Monsieur Cunche qui nous a expliqué les attendus du projet et les premières pistes sur lesquels nous pouvions partir.

II.1 Anticipation

La suite du projet a été toutefois un peu compliquée notamment à cause du retard de livraison du matériel. Initialement prévu pour être livré au début du projet, le matériel a été retardé de plus de 7 semaines en raison de divers problèmes du côté des fournisseurs.

Cette situation a eu un impact significatif sur notre calendrier et nous a contraints à adapter notre approche et à tirer parti du temps disponible pour approfondir notre compréhension du projet, effectuer des recherches préliminaires et anticiper les défis futurs. Nos premières recherches sont focalisées

sur la collecte d'informations sur LoRa, ses concepts et la manière dont allait fonctionner notre matériel. Bien que compliquées, ces recherches ont été fructueuses et elles nous ont permis de gagner un temps précieux pour la suite de notre projet. Ainsi, nous avons pu anticiper au maximum l'arrivée tardive du matériel, en prévenant les différents points de blocages que nous allions pouvoir rencontrer, en nous organisant entre nous pour effectuer les recherches et les futures tâches à réaliser.

Une fois le matériel reçu, le développement du prototype LoRaWAN a été marqué par une série d'étapes méthodiques visant à construire un système fonctionnel pour la capture et l'analyse des communications LoRa.

II.2 Configuration Raspberry Pi

Dans un premier temps, nous avons pris connaissance de l'intégralité du matériel et avons procédé à la configuration de tous les Raspberry Pi mis à notre disposition. Initialement, ces micro-ordinateurs nécessitaient l'utilisation d'un écran, d'un clavier et d'une souris pour être manipulés. Par la suite, nous avons effectué des configurations via SSH, ce qui nous a permis de contrôler sans périphériques les Raspberry Pi depuis n'importe quel ordinateur du réseau. De plus, nous avons mis en place l'accès VNC afin d'avoir la possibilité, au besoin, d'utiliser l'interface graphique des systèmes embarqués.

II.3 Création transmetteurs Arduino

La création des transmetteurs a été orchestrée à l'aide de la carte Arduino Uno R3, agissant comme le générateur de paquets LoRa. Suivant les consignes de la documentation (2), nous avons alors souder nos modules de transmission avec les antennes pour pouvoir rendre ces dernières opérationnelles et pour garantir la connectivité stable de l'ensemble du matériel avec la carte Arduino. La décision a été prise de réaliser la soudure de deux modules de chaque catégorie, équipés d'antennes, afin d'initier les tests d'émission et de réception de paquets LoRa entre ces modules. Cette approche vise à explorer toutes les configurations envisageables, tout en conservant du matériel en stock pour pallier d'éventuels dommages.

II.4 Sniffing

Il nous a paru d'abord important, avant d'essayer de sniffer des paquets LoRa, de réussir à transmettre et recevoir nos propres paquets à l'intérieur de notre système. Cela s'est fait en plusieurs étapes que nous allons détailler ci-dessous.

Pour commencer, nous avons entrepris de tester l'émission et la réception en utilisant deux modules similaires avec les programmes de la bibliothèque Arduino RadioHead dédiée. Après avoir correctement effectué les branchements et assigné les ports appropriés, nous avons réussi à recevoir nos premiers paquets LoRa.

Ensuite, une étape supplémentaire a consisté à vérifier la possibilité de communication entre les divers modules LoRa. Cependant, nous avons constaté que cela n'était pas réalisable, impliquant ainsi que chaque module serait limité à la capture d'un type spécifique de communication.

Notre prochaine étape consistait à vérifier la possibilité de communication entre les différents modules LoRa. Cependant, nous avons constaté que cette communication n'était pas effective, limitant ainsi chaque module à sniffer un type spécifique de transmission.

Une première tentative a consisté à élaborer un script Arduino permettant au récepteur de balayer plusieurs fréquences. Bien que cela se soit avéré efficace lorsque l'émetteur envoyait continuellement des paquets, nos mentors, Madame Goursaud et Monsieur Cunche, nous ont conseillé que cette approche n'était pas appropriée. Ils ont souligné que chaque module devait écouter une fréquence spécifique pour avoir une chance de capturer les paquets LoRa, qui étaient relativement rares dans notre contexte.

Dans notre démarche suivante, nous avons entrepris de rechercher la fréquence la plus courante pour notre module dans le territoire français. Suite à des investigations et aux conseils de M. Samuel Pélassier, nous avons opté pour des fréquences aux alentours de 868 MHz avec des variations minimales de 0.1 à chaque fois essai pour élargir notre champ d'observation et de constation.

Cependant, à cette étape, nous n'observions toujours pas la réception de paquets d'émetteurs externes. Plusieurs hypothèses ont émergé à ce sujet. Actuellement, notre émetteur communique en broadcast, **ce qui suggère que les paquets non adressés à notre récepteur pourraient être ignorés**.

De plus, nous avons maintenu un sous-ensemble spécifique de spreading factors, de largeurs de bande et de coding rates standard, **ce qui pourrait expliquer pourquoi notre sniffer ne détecte pas les paquets envoyés par un émetteur avec une configuration différente**.

Pour éliminer la première hypothèse, nous avons entrepris des essais pour recevoir des paquets même lorsqu'ils ne nous étaient pas directement destinés. Nous avons examiné les changements d'adresses de destination et exploré la possibilité d'écouter des messages qui ne nous étaient pas adressés. Cette approche s'est avérée efficace, notamment grâce à l'utilisation du mode promiscuous. Lorsque l'émetteur transmet vers une adresse quelconque, le paquet est néanmoins capturé. Ainsi, l'hypothèse d'un mauvais adressage a été écartée.

Concernant la configuration du « Spreading Factor - Largeur de bande - Coding Rate », nous avons également mené des essais en configurant l'émetteur et le récepteur de deux manières différentes, et les transmissions étaient toujours possibles. Par conséquent, l'hypothèse d'un écart de configurations a également été écartée.

II.5 Traitement des données par Python

En parallèle, un script Python a été développé pour traiter les paquets reçus. Ce script a été conçu pour explorer les données, identifier les motifs et aider à résoudre les problèmes rencontrés. L'objectif final était la visualisation des communications LoRa dans un format .pcap, qui offre une analyse détaillée des échanges entre les dispositifs du réseau.

III Prototype

III.1 Matériel

Notre prototype de sniffer (schéma en Fig. 2 et photo en Fig. 3) est composé de quatres composants essentiels: une antenne, le récepteur LoRa, une carte Arduino et un Raspberry Pi. Détailons les caractéristiques et le rôle de chacun:

- **L'antenne** : antenne conçue pour le réseau Sigfox (réseau bas débit). Elle est GSM et fonctionne, notamment, pour des fréquences de 824 à 960 MHz. Les fréquences autorisées pour LoRaWan se trouvent de 867 à 869 MHz, cette antenne fonctionne donc pour notre Sniffer, mais elle est de faible portée. Pour ces fréquences, l'antenne présente un gain inférieur à 0 dBi ainsi qu'un taux d'ondes permanant (VSWR) inférieur à 2
- **Le récepteur** : module radio RFM95W LoRa fonctionnant, notamment, pour les fréquences autour de 868 MHz. Ce module est transceiver, il est donc capable de réception tout comme d'émission de paquets LoRa. Il fonctionne avec la bibliothèque arduino RadioHead ; bien documentée, qui rend son utilisation facile. L'intérêt d'un module RFM95 est sa portée ; jusqu'à 2km avec une simple antenne et

jusqu'à 20km avec une antenne directionnelle et quelques améliorations (d'après le constructeur). Ce module n'est toutefois pas compatible avec les modules d'autres séries, notamment la série RFM69.

- **La carte arduino** : carte de développement microcontrôleur basée sur l'ATMMega328P. Cette carte dispose de toutes les entrées nécessaires pour connecter le module radio LoRa en notre possession. Elle s'utilise en téléversant des scripts ino par connexion série avec un ordinateur. Sa console est alors lisible en série lorsque l'ordinateur est connecté par câble USB-série FTDI.
- **Le Raspberry Pi** : micro-ordinateur polyvalent. Il supporte le logiciel permettant la connexion à la carte Arduino ce qui permet de faire le lien entre le récepteur LoRa et le traitement des paquets réalisé en Python.

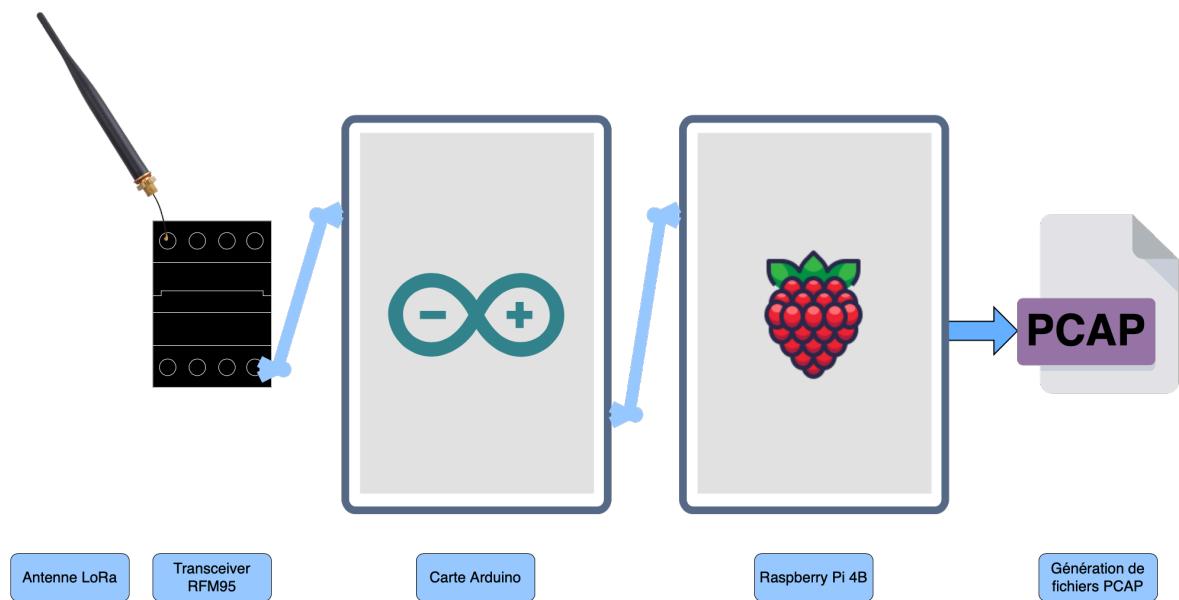


Fig. 2. – Schéma du prototype

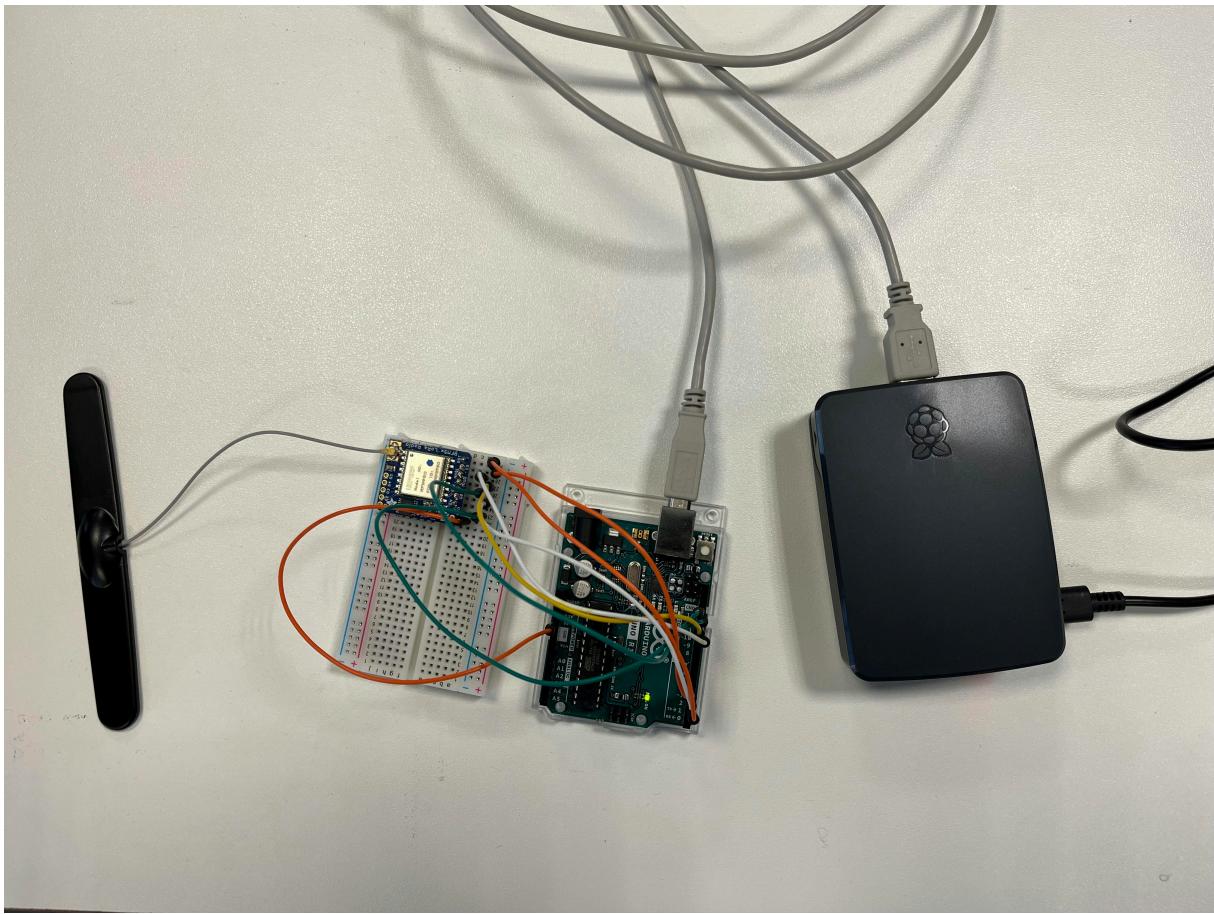


Fig. 3. – Prototype de sniffer

Salut

III.2 Code

Deux scripts assurent le fonctionnement de notre prototype. Le premier, en ino, utilise la bibliothèque RadioHead (3). Il assure la réception de paquet et imprime dans le moniteur série, à chaque réception, les éléments de l'en-tête du paquet LoRa ainsi que son contenu s'il n'est pas chiffré. Le second, en Python utilise les bibliothèques PySerial (4) et Scapy (5). Grâce à PySerial, il récupère le contenu de la sortie série de l'arduino et transforme ce texte en fichier PCAP par Scapy.

L'entièreté du code est accessible sur la branch main de notre dépôt github (6)

III.2.1 Bibliothèques

- **RadioHead**

La bibliothèque RadioHead pour Arduino (3) est une bibliothèque open source conçue par Mike McCauley. Elle est entièrement orientée objet et est pensée pour l'envoi et la réception de messages sous forme de paquets via divers systèmes radio sur microprocesseurs embarqués. La bibliothèque est composée de deux types de classes, les **pilotes** et les **gestionnaires**. Les pilotes offrent un accès de bas niveau aux systèmes radio pour la communications de paquets. Les gestionnaires fournissent des fonctionnalités de communications de paquets de haut niveau.

Pour notre projet, nous opterons pour l'utilisation du pilote RH_RF95 afin d'assurer une gestion exhaustive de notre module radio.

- **PySerial**

PySerial (4) est une bibliothèque Python open source d'accès au port série. Nous l'utilisons pour récupérer le contenu de la sortie série de la carte Arduino

- **Scapy**

Scapy (5) est une bibliothèque Python dédiée à la manipulation de paquets. Elle permet la création, la modification et l'envoi de paquets. Nous l'utiliserons pour transformer le texte récupéré en série, en fichier PCAP lisibles par Wireshark

III.2.2 Architecture

Pour résumer notre architecture, voici le fonctionnement de nos scripts illustré en Fig. 4:

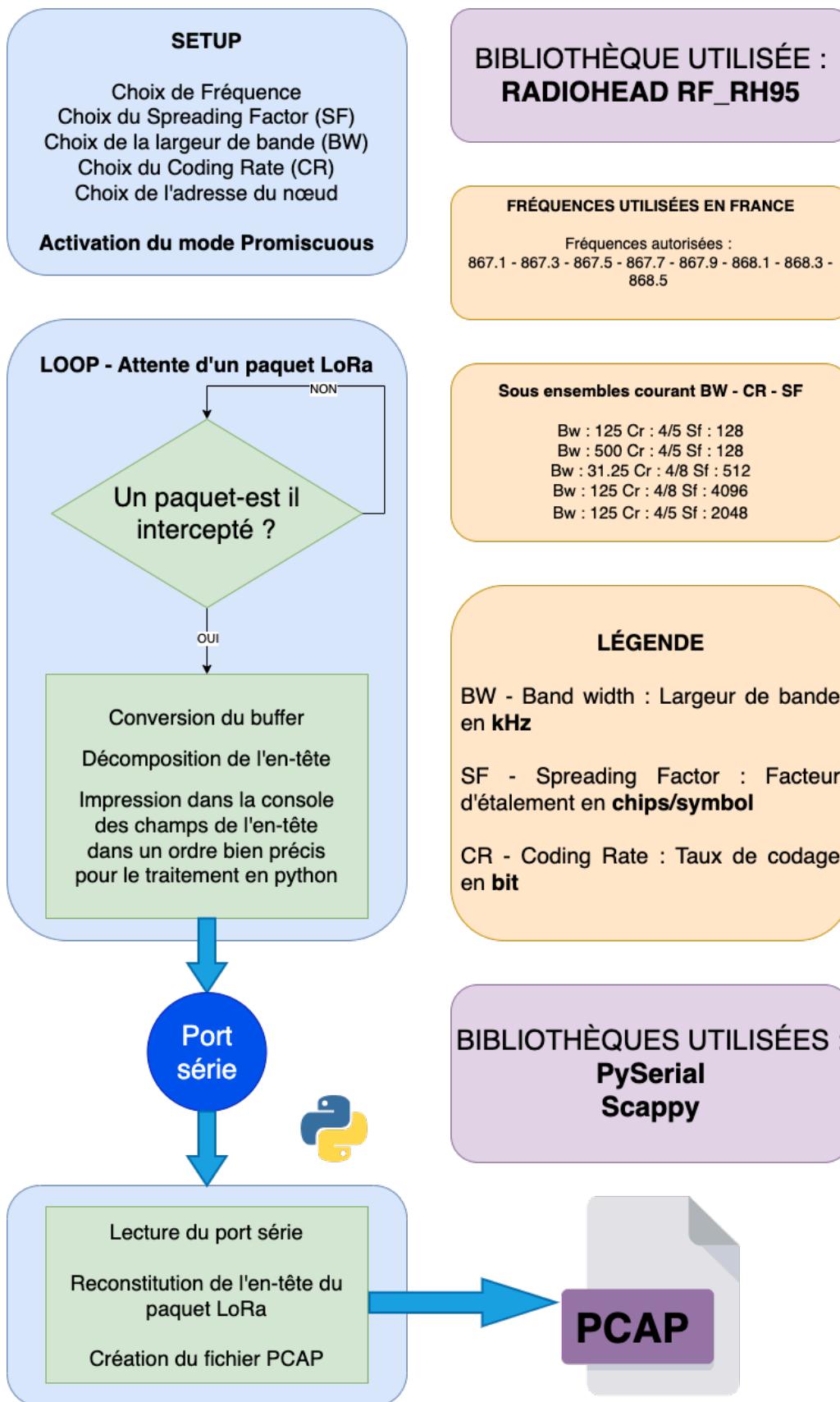


Fig. 4. – Architecture du code

III.3 Émetteur

Pour assurer le fonctionnement de notre sniffer, nous avons conçu un émetteur avec du matériel similaire au récepteur. Il offre la possibilité de configurer divers paramètres tels que la fréquence d'émission, le spreading factor, la largeur de bande, le coding rate, ainsi que les adresses source et destination pour les émissions. La polyvalence de notre émetteur nous permet de simuler toutes situations d'émetteur étrangers.

IV Bilan du projet

IV.1 Résultats

Au cours de ces dernières semaines, notre projet a abouti à la création d'un prototype fonctionnel capable de capturer des paquets LoRa émis par un émetteur similaire, indépendamment des adresses de destination et source, du mode ou du spreading factor. Cette flexibilité opérationnelle est rendue possible lorsque l'émetteur et le récepteur sont configurés sur la même fréquence de transmission et d'écoute. La portée de notre système s'étend sur environ 900 mètres, et les détails de la méthode d'acquisition de cette métrique sont explicités au Chapitre IV.2.

Pour assurer une communication efficace, notre système établit une liaison série avec une carte Raspberry Pi 4. Cette connexion permet le traitement des trames reçues au moyen d'un script Python. Ainsi, notre solution offre une intégration pratique avec des outils informatiques, offrant des possibilités étendues de traitement et d'analyse des données. L'ensemble du processus vise à garantir une interopérabilité optimale et une gestion efficace des transmissions LoRa, renforçant ainsi la fiabilité et la fonctionnalité globale de notre prototype.

IV.2 Limites du prototype

Malgré l'évolution du projet, nous sommes encore confrontés à un défi persistant, puisque pour le moment nous n'avons pas réussi à capter de paquets LoRa étrangers (c'est-à-dire non-émis par notre propre transmetteur). Après avoir écarter les hypothèses d'incohérence d'adressage, ou de configurations, deux nouvelles hypothèses ont été émises: les paquets LoRa sont trop rares et éparses et nos périodes d'observations trop courtes pour pouvoir en percevoir, et notre matériel n'est pas suffisamment performant pour nous permettre de recevoir des paquets émis à une distance conventionnelle.

Ne pouvant nous fier entièrement à notre module de réception, nous avons d'abord voulu nous assurer de la présence de signaux LoRa sur bande de fréquence que nous avons choisie de sniffer. Pour cela nous avons utilisé l'analyseur de spectre portable TinySA afin de visualiser la puissance du signal présent dans la bande de fréquence LoRa que nous cherchons à sniffer puis de le comparer au spectre du signal lorsque nous émettons avec notre propre module.

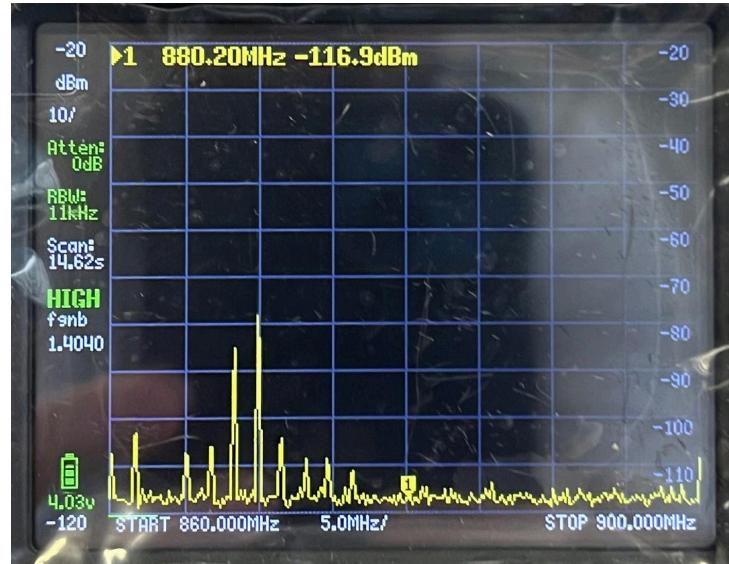


Fig. 5. – Signal observé avec la TinySA sur la bande de fréquence [860MHz,900MHz] lors de la transmission de paquets LoRa



Fig. 6. – Signal observé avec la TinySA sur la bande de fréquence [860MHz, 871MHz]

Fig. 5, la graduation horizontale de 5MHz nous permet de vérifier le spectre du signal lors de la transmission de paquets LoRa sur la fréquence 868.1 MHz. Dans un deuxième temps, nous avons également inspecté la présence de signaux de même spectre sur les autres canaux de fréquences normalisées et autorisées en France. Malheureusement, lorsque nous coupions notre transmetteur, nous pouvons voir Fig. 6 qu'aucun signal de spectre semblable ne subsistait, même à plus faible puissance, dans cette bande fréquence. Les pics de signal que nous observons, bien que occupant des fréquences utilisées en LoRa, n'ont pas le même spectre que celui que nous obtenons lorsque nous transmettons. Sachant que ces canaux sont également utilisés par d'autres réseaux et protocoles, il ne nous est pas possible d'affirmer si ces signaux perçus sont ou non du LoRa. Cependant, si ces signaux proviennent de modules LoRa, nous n'arrivons toujours pas à les capter, ce qui signifierait que le problème réside ailleurs.

Dès lors, nous avons cherché à éprouver la possibilité que notre matériel manque de performance. En effet, le signal LoRa, généralement de faible intensité en milieu urbain, nécessite une antenne suffisamment puissante pour être capté. Il nous fallait donc évaluer la portée de notre récepteur, ce qui

explique notre procédure en prenant des mesures de différentes métriques en fonction de nos déplacements sur le campus:

- Le RSSI (Received Signal Strength Indicator) :

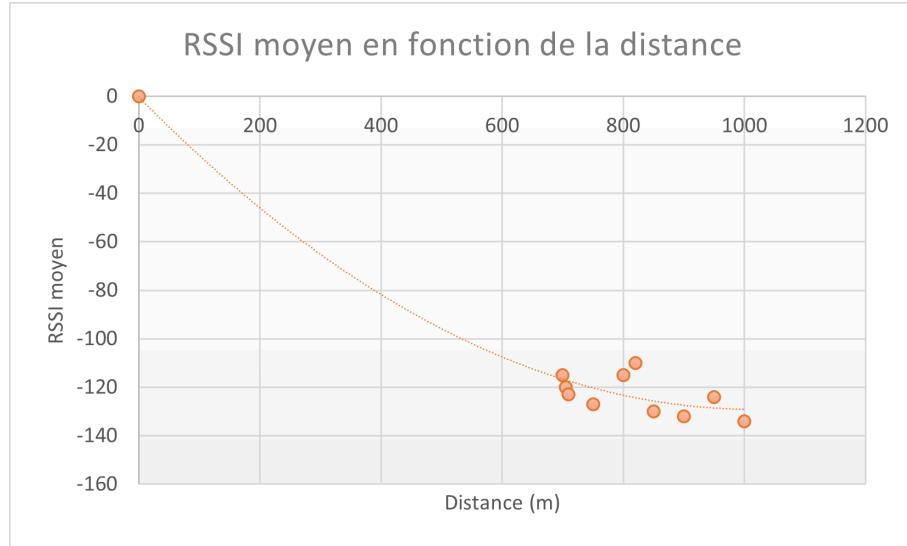


Fig. 7. – RSSI moyen d'un paquet LoRa reçu en fonction de la distance émetteur/récepteur

Fig. 7 évalue la perte en puissance et en qualité du signal au fur et à mesure que nous éloignons l'émetteur du récepteur et que des obstacles s'interposent au signal. Nous voyons que la qualité du signal chute très rapidement pour un signal LoRa. Au delà de -137, nous ne recevions plus le paquet émis. En effet, au bout d'environ 700m, nous observions régulièrement des pertes de paquets bien que le signal soit à nouveau perçu quelque secondes plus tard. Nous avons donc décidé de recenser également le nombre moyen de paquets perdus en fonction de la distance.

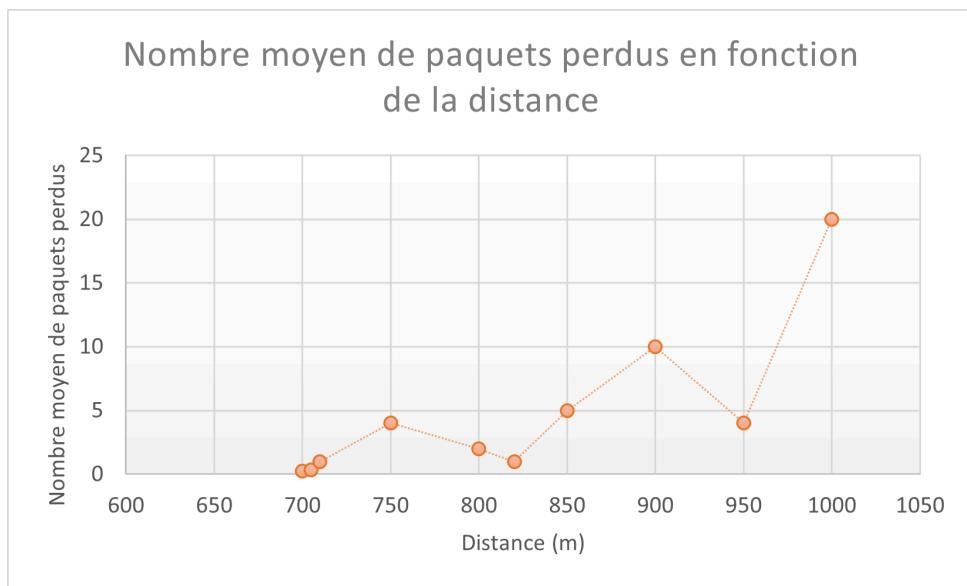


Fig. 8. – Nombre moyen de paquets LoRa perdus en fonction de la distance émetteur/récepteur

Nous avons ainsi déterminé avec Fig. 8 que la distance maximale à laquelle nos modules pouvaient encore communiquer est de moins d'un kilomètre en extérieur en champ quasi-libre. Sachant qu'un

module LoRa peut avoir une portée allant jusqu'à dix kilomètre, ces observations mettent en évidence le manque de performance de notre système et de notre matériel bien spécifiquement. Nous avons ainsi tiré la conclusion que les messages LoRa externes, en raison de leur nature dispersée, de leur faible fréquence, de leur espace temporel, et de leur émission sur un canal parmi plusieurs canaux dédiés au LoRa, étaient intrinsèquement difficiles à sniffer avec notre matériel limité. En effet, à ce stade, nous n'écoutes qu'un canal à la fois, sur des plages de temps de quelques minutes, dans un bâtiment isolé, et avec une antenne de faible puissance.

V Pistes d'améliorations

Contraints par des limites de temps et de matériel, notre projet peut bénéficier d'améliorations significatives, et il reste encore beaucoup de travail à accomplir pour parvenir à un sniffer LoRa performant. Ci-dessous, nous présentons quelques axes potentiels pour approfondir davantage notre projet :

V.1 Ecoute multi-bande

Comme mentionné au Chapitre IV.2, notre module actuelle ne peut sniffer qu'un seul canal à la fois. Cette contrainte nous restreint non seulement à capturer seulement une partie du trafic, mais elle diminue également la probabilité de détecter des paquets LoRa externes au cours d'une observation donnée. Ainsi, une évolution cruciale de notre système monochannel vers un système multibande représente la principale amélioration à implémenter avant de poursuivre le développement de ce projet.

Nous avions prévus d'utiliser un ensemble composé d'une carte *Arduino Uno*, d'un module radio et d'une antenne pour chaque canal que nous voulons écouter. Tous ces ensembles communiqueront avec une carte *RaspberryPi 4* afin de transmettre les trames reçues sur leur canal respectif et qu'ils puissent être traités ensemble ou séparément.

V.2 Wireshark

Faisant partie des fonctionnalités d'origines du projet, les traces des paquets sniffés devaient pouvoir être visualisées, triées et traitées via la plateforme WireShark. Cet outil libre et gratuit d'analyse de trafic réseau supporte depuis peu le protocole LoRaWAN (7) et permettra une plus grande accessibilité et facilité d'exploitation du système.

Pour cette étape, il est essentiel de convertir les traces des paquets capturés au format .pcap afin de les importer dans WireShark pour une analyse approfondie. Cependant, nous avons rencontré des difficultés au niveau de cette conversion. La librairie WireShark, notamment l'outil *text2pcap*, qui est conçu pour convertir des fichiers texte au format .pcap, présente des erreurs de construction et d'installation fréquentes. De plus, elle ne permet pas une conversion directe des traces LoRa au format .pcap. Dans le but de contourner cette limitation, nous avons envisagé d'adapter nos traces LoRa aux champs d'un protocole utilisable avec *text2pcap*, tel que UDP par exemple. Malheureusement, la documentation de cette librairie (8) ne précise pas explicitement la syntaxe textuelle à appliquer pour utiliser *text2pcap*. En raison de contraintes de temps, nous n'avons pas pu approfondir notre compréhension de cette librairie prometteuse.

Lors de nos recherches, nous avons également rencontrés plusieurs projets visant à importer des traces LoRa en .pcap. Notamment une librairie nommée *LoRaTap* (9) définissant un format d'encapsulation permettant de stocker le traffic LoRa dans des fichiers .pcap. Nous avons cherché à contacter l'auteur de cette librairie afin d'avoir plus d'informations sur l'architecture et le mode de fonctionnement de son projet, mais nous n'avons pas obtenu de réponse et n'avons pas su adapter cette librairie

à notre système dans les temps impartis. Nous continuons cependant à penser qu'il serait intéressant de creuser ces deux pistes afin d'essayer d'articuler *text2pcap* et *LoRaTap* ensemble.

V.3 Amélioration du matériel

Egalement mentionné au Chapitre IV.2, notre matériel actuel présente des limitations en termes de couverture pour du LoRa, et des améliorations significatives sont possibles. Une fois notre prototype entièrement finalisé, incluant une écoute multi-bande et une compatibilité avec Wireshark, il serait particulièrement bénéfique d'équiper les récepteurs d'antennes plus performantes spécialement conçues pour une utilisation LoRa. Des antennes avec un gain de 2dBi, adaptées au LoRa, sont disponibles pour un coût similaire, offrant ainsi une solution plus performante à un prix abordable.

VI Conclusion

En conclusion, ce projet a été une aventure passionnante qui nous a plongés dans l'univers captivant de l'Internet des Objets (IoT) et de la technologie LoRaWAN. Nous avons navigué à travers des défis techniques stimulants, de la conception du dispositif de sniffing à la mise en œuvre des mécanismes de collecte et d'analyse des données.

L'adoption étendue de notre projet LoRaWAN Sniffer ouvre la voie à une transformation significative, apportant des avantages notables dans plusieurs domaines y compris le domaine sécurité et optimisation des Réseaux IoT, Optimisation des Réseaux...

- Sécurité des Réseaux IoT : D'une part, l'extraction d'informations à partir de l'en-tête des paquets LoRaWAN Sniffer peut présenter des implications techniques importantes en termes de confidentialité. Les données contenues dans l'en-tête, telles que les adresses source et destination, les informations de séquence et d'autres métadonnées, sont essentielles pour comprendre la communication. Cependant, leur interception non autorisée pourrait compromettre la confidentialité des échanges. D'autre part, la capacité de capter et d'analyser les paquets LoRaWAN peut contribuer à l'amélioration de la sécurité des réseaux IoT en identifiant les vulnérabilités potentielles. Cela permet aux gestionnaires de réseau de renforcer les mesures de sécurité en réponse aux faiblesses détectées tels que le chiffrement des données en transit et des protocoles d'authentification fiables.
- Détection d'Intrusions : En surveillant activement les communications LoRaWAN, le sniffer peut jouer un rôle crucial dans la détection d'intrusions. Il peut repérer des schémas de comportement suspects ou des activités non autorisées, facilitant ainsi la réaction rapide aux menaces potentielles.
- Optimisation des Réseaux : L'analyse des paquets peut fournir des informations précieuses sur les performances du réseau, les zones de couverture, les taux de réussite de transmission, etc. Ces données sont essentielles pour optimiser l'infrastructure LoRaWAN, améliorer la qualité de service et garantir une connectivité fiable.

L'impact potentiellement révolutionnaire de notre projet LoRaWAN Sniffer sur le domaine des réseaux et sécurité ouvre la voie à une transformation significative voire révolutionnaire.

Ce projet nous a offert une plongée immersive dans un domaine émergent, nous permettant de découvrir les complexités et les possibilités infinies de l'IoT. Il a élargi notre vision et ouvert la porte à de nouvelles perspectives technologiques. Nous espérons que notre contribution à l'exploration du sniffer LoRaWAN et l'exploitation de cette technologie ajoutera une pierre à l'édifice de sa compréhension et son évolution.

Enfin, nous sommes reconnaissants envers l'ensemble de l'équipe et des personnes qui ont rendu possible cette expérience unique. Nos remerciements vont tout d'abord à Mr Mathieu Cunch et Mdm Clare Goursaud, dont les conseils éclairés et l'expertise ont été cruciaux pour orienter nos efforts dans la bonne direction.

Nous tenons également à remercier Mr Samuel Pélissier, Mr Herve Rivano et Mr Walid Bechkit pour leur collaboration précieuse et leurs temps pour répondre à nos diverses questions. Leur expertise dans la technologie nous était une clé pour une compréhension approfondie.

Ce projet restera un jalon significatif dans notre parcours académique, nous préparant à affronter de nouveaux défis et à explorer davantage les frontières passionnantes de la technologie et bien précisément de l'IOT. Un grand merci à tous pour cette aventure inoubliable.

Bibliographie

1. HUCHER T, Concept G. Protocole LoRa/LoRaWAN [Internet]. 2023. Disponible sur: <https://www.giga-concept.fr/technologies/protocole-lorawan/>
2. Industries A. Adafruit RFM69HCW and RFM96/RFM95/RFM98 LoRa Packet Radio Breakouts [Internet]. 2024. Disponible sur: <https://learn.adafruit.com/adafruit-rfm69hcw-and-rfm96-rfm95-rfm98-lora-packet-padio-breakouts>
3. McCauley M, Arduino. RadioHead Packet Radio library for embedded microprocessors [Internet]. 2024. Disponible sur: <https://www.arduino.cc/reference/en/libraries/radiohead/>
4. Liechti C. pySerial [Internet]. 2023. Disponible sur: <https://pythonhosted.org/pyserial/>
5. Biondi P. Scapy [Internet]. 2024. Disponible sur: <https://scapy.net/>
6. Partouche H, Caringi N, Khalil S, Massol M. LORA Sniffer [Internet]. 2024. Disponible sur: https://github.com/MargauxMsl/LORA_Sniffer
7. Wireshark. Display Filter Reference: LoRaWAN Protocol [Internet]. 2023. Disponible sur: <https://www.wireshark.org/docs/dref/l/lorawan.html>
8. Narayanan A. Wireshark - text2pcap Manual Page [Internet]. 2023. Disponible sur: <https://www.wireshark.org/docs/man-pages/text2pcap.html>
9. Jong E de. LoRaTap [Internet]. 2022. Disponible sur: <https://github.com/eriknl/LoRaTap>