Retour d'expérience d'un déploiement LoRaWAN à Strasbourg

Guillaume Schreiner
ICube - UMR7357
300 Bd S. Brant, CS 10413,
F-67412 Illkirch CEDEX

Jean Melounou
Eurométropole de Strasbourg
1 parc de l'étoile
F-67076 Strasbourg CEDEX

Manuel Yguel
Strataggem
36, rue de l'Université
67000 STRASBOURG

Résumé

Contrairement au Wi-Fi ou au Zigbee qui se basent sur des technologies radio ayant une portée de quelques dizaines de mètres, le standard LoRaWAN approche des distances de plus de 10kms. Ces performances remettent fondamentalement en cause le précédent paradigme des réseaux maillés courte portée nécessitant plusieurs dizaines de nœuds intermédiaires pour couvrir la même distance radio.

La plateforme Inetlab du laboratoire ICube (UMR CNRS/Université de Strasbourg) a pour vocation de fournir les équipements et les logiciels destinés aux expérimentations des nouvelles technologies de l'Internet des Objets. En 2016 débute une expérimentation originale qui vise à déployer une infrastructure LoRaWAN complète (hardware et software) à l'échelle de l'agglomération, mêlant ICube à l'Eurométropole de Strasbourg et à la société Strataggem. Les buts multiples de ce déploiement sont d'évaluer le potentiel opérationnel de ce standard, l'émergence de nouvelles applications dans le contexte des villes intelligentes, les bénéfices à partager les antennes entre plusieurs acteurs locaux pour densifier la couverture radio, etc.

A travers cet article, nous vous présenterons tout d'abord la norme LoRaWAN dans le contexte de l'Internet des Objets afin de préciser ses avantages et ses inconvénients théoriques. Dans un second temps, nous partagerons notre retour d'expérience sur l'architecture de l'infrastructure LoRaWAN déployée, allant de la réalisation des objets connectés jusqu'à l'intégration dans le système d'information, en passant par l'étude des performances observées. Enfin, nous illustrerons le formidable potentiel des objets connectés à travers les cas d'usages déployés, qui s'appliquent aussi bien aux activités de recherche des laboratoires qu'aux applications métiers des différentes directions des services techniques de l'Université ou des collectivités.

Mots-clefs

IoT, LoRaWAN, Infrastructure, Smart City, OpenData, Bigdata

1 Introduction

Les objets connectés prennent une place de plus en plus prépondérante dans notre quotidien numérique. Selon les dernières études, on recense actuellement 7 milliards d'objets connectés en service en dehors des smartphones et tablettes. En 2025, on estime que plus de 25 milliards d'objets connectés seront activés. Bien plus qu'un argument marketing permettant de se distinguer des autres fabricants, la fonctionnalité d'avoir un objet connecté à Internet sera bientôt un prérequis obligatoire pour bon nombre d'industriels.

Au cours des deux dernières décennies, de nombreuses technologies radio et protocoles réseaux ont été développés afin de répondre aux différents scénarios d'usages de l'Internet des Objets (IoT). En résumé, on distingue deux grandes familles de technologies radio sans fil, les réseaux courtes et longues portées qui partagent cependant tous la même contrainte : être le plus économe en énergie. En effet, de nombreux objets déployés sont équipés de batteries afin d'être énergétiquement autonomes ce qui facilite leur déploiement.

Attentes des différents partenaires

La plateforme Inetlab [1] du laboratoire ICube [2] souhaite expérimenter le potentiel et les performances de la technologie IoT longue portée par rapport aux réseaux IoT courte portée. Dans cette démarche, Inetlab s'est doté d'une infrastructure appelée LRP IoT (Long Range Protocol for IoT) proposant un déploiement en conditions réelles sur différents campus de la ville. Cette expérimentation s'inscrit dans la durée pour évaluer la montée en puissance du trafic IoT longue portée et sa capacité à passer à l'échelle.

Dans le cadre de sa démarche Smart City (ville intelligente), ce réseau IoT expérimental permet à l'Eurométropole de Strasbourg [3] de développer et tester de nouveaux usages des objets connectés dans les domaines de la mobilité, de l'environnement et de la gestion de son patrimoine technique. Il constitue en outre un facteur d'attractivité et de développement économique pour la collectivité en stimulant l'innovation. Ce réseau est en effet mis à disposition de l'ensemble des acteurs du territoire qui le souhaitent, qu'il s'agisse d'étudiants, de startups ou d'entreprises. L'usage du réseau est gratuit mais strictement limité à des applications expérimentales et non commerciales.

Dans une même démarche de passage à l'échelle que Inetlab, la société Strataggem [4] souhaite expérimenter différents modèles d'architecture de déploiement IoT. Strataggem maîtrise l'ensemble de son infrastructure allant des objets connectés au cloud.

Cet article présente en détail la technologie IoT longue portée LoRaWAN, la solution technique retenue pour son déploiement et notre retour d'utilisation après trois années d'expérimentation.

2 Comparatif des technologies radio IoT

Afin de bien comprendre les différences notables entre les différentes technologies radio utilisées dans l'IoT, nous les classons dans le Tableau 1 selon les critères suivants :

- la consommation d'énergie relative ;
- le débit ;
- la portée radio ;

- la topologie de l'infrastructure ;
- le type de déploiement : autonome, opérateur.

Nous distinguons deux groupes principaux qui s'établissent selon le critère de portée radio :

- courte portée radio inférieure à 100 m ;
- longue portée radio supérieure à 1 km.

En fonction de l'application envisagée, la portée radio est souvent déterminante pour envisager un déploiement extérieur ou intérieur dans un grand bâtiment sans la contrainte de recourir à une infrastructure lourde comme par exemple un réseau dense de points d'accès Wi-Fi [5]. D'autres protocoles comme le Zigbee [6] basé sur la norme IEEE 802.15.4 [7] ou le Bluetooth Low Energy (BLE) [8] plus économes en énergie sont envisageables à l'échelle du bâtiment. Ils permettent la création de réseaux maillés mais leur configuration reste compliquée à mettre en oeuvre à l'échelle de la ville. Les technologies radio longue portée comme la NB IoT [9], Sigfox [10] et LoRa [11] autorisent ce passage à l'échelle avec des débits suffisants pour de la télémétrie ou pour des actionneurs distants dans le cadre des applications envisagées pour notre expérimentation (voir section 3.3). Le critère de couverture radio totalement garantie ne peut être atteint que si l'utilisateur déploie son propre réseau d'antennes afin de combler les éventuelles zones blanches. Dans ce cas de figure, seul LoRa permet de gagner son indépendance en opérant soi-même sa propre infrastructure. Il s'agit du choix que nous avons fait dans le cadre du déploiement sur l'agglomération de Strasbourg.

	consommation	**débits**	**portée**	**infrastructure**	**opérateur/ autonome**
Wi-Fi	+	1,3 Gbits/s	100m <	étoile	autonome
BLE	+++	1 Mbit/s	100m <	point à point, maillé	autonome
Zigbee	++++	256 kbit/s	100m <	point à point, étoile, maillé	autonome
NB IoT	+++	250 kbit/s	30km >	cellulaire	opérateur
Sigfox	++++	environ 800 bits/ s	10km >	cellulaire	opérateur
LoRa	++++	de 250 à 5470 bit/s	10km >	point à point, étoile, maillé	les deux

Tableau 1 - Tableau de comparaison entre les différentes technologies radio

3 Infrastructure LoRaWAN

3.1 Principes généraux

3.1.1 La technologie radio LoRa

LoRa (Long-Range) est une technologie radio utilisant les bandes ISM[1] 868 MHz en Europe, 915 MHz aux USA, etc. Initialement développée par Cycléo en 2009 puis racheté par Semtech en 2012, LoRa fonctionne sur une modulation radio propriétaire qui permet d'atteindre des portées radio jusqu'à 5 km en zone urbaine dense et 15 km en zone dégagée. Les puces radio propriétaires sont vendues à bas coût pour viser des déploiements à large échelle supérieur à plusieurs milliers d'objets. La modulation LoRa propose différents modes d'émissions appelées « Spreading Factor » (SF) allant de 7 à 12. Plus le SF est fort, moins la bande passante est élevée. Le SF contraint la taille maximale d'un paquet allant de 51 à 222 octets maximum. Afin d'augmenter la robustesse aux interférences et de garantir l'intégrité des paquets, un CRC et un code correcteur (« Coding Rate ») sont inclus dans la trame LoRa.

Contrairement à Sigfox, la technologie pour le déploiement de points d'accès LoRa est libre d'accès. Cela permet à n'importe quelle entité de déployer sa propre infrastructure radio pour établir la couverture d'une zone géographique. LoRa bénéficie d'une standardisation liée à la norme LoRaWAN [12] décrite dans le paragraphe 3.1.2 et portée par la LoRa Alliance [13] et qui permet d'assurer la qualité des équipements et des services. Dans le cas du LoRa, l'accès au matériel pour les points d'accès et les terminaux étant ouvert, les modèles économiques sont très divers et très souples : abonnements en utilisant des infrastructures d'opérateurs, sans abonnement en déployant sa propre infrastructure et même mutualisation d'infrastructure comme le projet The Things Network [14].

3.1.2 La norme LoRaWAN

Afin de garantir l'interopérabilité entre fabricants, la LoRa Alliance a spécifié un standard appelé LoRaWAN qui décrit précisément le protocole de communication entre les différentes entités constituant l'infrastructure. Par rapport au modèle OSI, on place le standard LoRaWAN au niveau de la couche MAC (couche 2) directement au-dessus de la couche physique utilisant la technologie radio LoRa. Trois entités distinctes composent l'infrastructure LoRaWAN (voir Figure 1) :

- Les « end-devices » : il s'agit des terminaux ou objets connectés équipés de capteurs/actionneurs déployés pour instrumenter l'environnement ; ces objets utilisent un seul canal radio à la fois pour l'émission d'un message donné et les transmissions sont half-duplex ;
- Les « gateways » : ce sont les points d'accès ou « antennes » qui réceptionnent les messages des end-devices, elles ont la particularité d'écouter sur plusieurs canaux LoRa pour tous les SF à la fois ; les antennes écoutent généralement sur 8 canaux en même temps ; pour les messages à destination des objets, elles n'émettent que sur un seul canal à la fois ;
- Le « LoRaWAN server » : il s'agit d'un démon écoutant les messages relayés par les gateways ; il a pour rôle de décrypter le contenu des messages et de le présenter aux utilisateurs ; Les traitements postérieurs, permettant de faire suivre les messages ou de les stocker pour présentation ou analyse, sont hors standard et à la discrétion de l'opérateur.

1 bandes ISM (industriel, scientifique et médical)

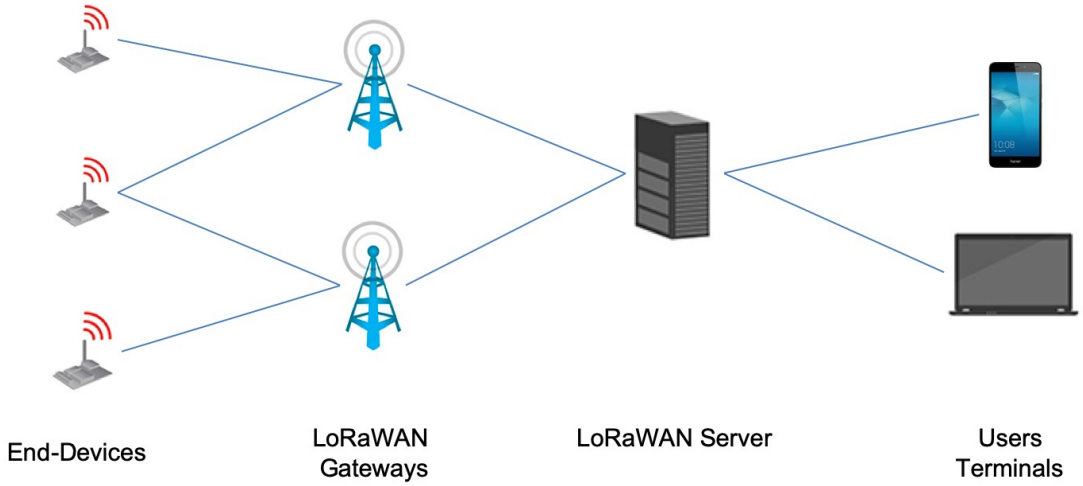

Figure 1 - Infrastructure LoRaWAN

3.1.3 ABP vs OTAA

Chaque end-device possède une adresse matérielle sur 64 bits unique appelée « end-device identifier » (DevEUI) ainsi qu'une adresse logique sur 32 bits appelée « end-device address » (DevAddr). Pour la sécurité, deux clés sont nécessaires pour le chiffrage symétrique AES 128 bits avec le LoRaWAN server. La première clé de sécurité « network session key » (NwkSkey) est utilisée pour les messages relatifs à la signalisation du protocole LoRaWAN. La seconde clé « Application session key » (AppSKey) est utilisée pour chiffrer les données utilisateurs à destination ou réception des end-devices.

Il existe deux modes supportés pour configurer ces paramètres d'accès au réseau. Un mode manuel appelé « Activation By Protocol » (ABP) ou un mode automatique appelé « Over-The-Air Activation » (OTAA).

Le mode ABP impose à l'utilisateur de configurer à la fois sur le end-device et sur le LoRaWAN server la DevAddr, la NetworksKey et la AppsKey. Une fois l'objet déployé, il n'est plus possible de changer ces paramètres sans accéder de manière physique au end-device. L'inconvénient majeur est de garder les mêmes clés de sécurité tout au long de la vie de l'objet. Si les clés sont compromises, tous les messages radio qui auraient été précédemment captés par un pirate pourraient être déchiffrés. De plus, la DevAddr, spécifique à un objet, présente dans l'entête des trames LoRaWAN est non chiffrée contrairement au payload. Elle permet donc d'identifier le trafic radio d'un objet.

Le mode OTAA implique l'enregistrement d'une clé symétrique appelée « Application Key » qui permet une négociation sécurisée et dynamique des DevAddr, NetworksKey et AppsKey avec le LoRaWAN server. Cette phase d'accès au réseau est appelée le « Join ». Il s'agit du mode recommandé qui permet de dynamiquement renouveler les paramètres réseaux entre le end-device et le LoRa Server. Dans ce cas la DevAddr est également spécifique à un objet, mais sans avoir assisté au Join, il est plus compliqué pour un pirate de rattacher l'adresse à un objet.

3.1.4 uplink, downlink et classes d'objets

Du point de vue des end-devices, on distingue le trafic montant appelé « uplink » du trafic descendant appelé « downlink ». Il est possible également de demander un acquittement de bonne réception de message, on parle alors de « confirmed uplink » et « confirmed downlink ».

Le end-device peut émettre des messages selon trois modes de fonctionnement décrit dans des classes.

La classe A correspond à un message uplink initié par le end-device qui va ensuite écouter sur les deux

fenêtres temporelles de réceptions l'éventuelle envoie d'un message downlink par le serveur en réponse. Ce mode convient bien pour des objets contraints énergétiquement qui restent en veille en dehors des phases d'émissions. En contrepartie, il est impossible d'envoyer un message downlink en dehors des fenêtres d'émission.

Pour répondre à ce besoin, LoRaWAN a spécifié la classe C et la classe B. La classe C implique que l'objet écoute en permanence le canal radio en attente d'un potentiel message. Ce mode a pour contrepartie d'être très énergivore et implique, en pratique, une source d'alimentation électrique continue. Enfin, il existe la classe B proposant une synchronisation grâce à des « beacons » (messages d'annonce périodiques) entre le end-devices et la gateway. Le end-device alterne des phases de veille et d'écoute, assurant ainsi un compromis entre efficacité énergétique et réactivité des messages downlinks.

3.1.5 Datarate et règle du 1%

Le « datarate » (DR) LoRaWAN correspond à la quantité d'information que peut transmettre un end-device au cours d'une communication. Il est constitué d'une combinaison d'un SF et d'une largeur de bande. Le datarate augmente quand le SF diminue et quand la largeur de bande augmente. En contrepartie, plus le datarate augmente plus la portée diminue. Pour transmettre la même quantité d'information, un datarate élevé entraînera une communication de durée plus courte. Ainsi avec un datarate plus élevé, un end-device peut rester éveillé moins longtemps et économiser de l'énergie, de plus, il encombre moins le spectre radio et cela permet à un plus grand nombre d'objets de communiquer. L'utilisation par la modulation LoRa des bandes de fréquence ISM 868 MHz en Europe implique de respecter un fair use de temps de communication de moins de 1% pour l'utilisation des canaux radio LoRaWAN 1, 2 et 3 (868.1 MHz, 868.3 MHz et 868.5 MHz respectivement). Ceci explique en partie l'asymétrie du trafic entre les uplinks émis par les end-devices et les downlinks émis par les gateways. En effet, de manière concurrente, le nombre d'objets avec lesquels une gateway peut communiquer est le produit du nombre de canaux supportés par une gateway multiplié par le nombre de SF. Généralement ce nombre est de 48 (6 SF et 8 canaux). Le module radio SX1301 d'une gateway ne peut pas démoduler deux messages avec le même SF et le même canal exactement en même temps. Lorsque cela arrive on parle de collision et seul un message sur les deux est démodulé. C'est une limitation fondamentale de la technologie LoRa qui a été bien étudiée. L'autre limitation vient du fait qu'une gateway a le même temps légal de parole qu'un objet et que donc son temps de parole est divisé entre tous les objets avec lesquels elle communique. C'est pour cela que le quatrième canal officiel se trouve dans une bande spécial où le temps de communication légal est de 10% et qui est réservé au message downlink des gateways.

En pratique, en considérant des messages de longueur variable (entre 1 et 222 octets), une fréquence de communication allant de 1 message par heure à 1 message par 10 minutes et par objet, chaque gateway peut supporter en écoute environ un millier d'objets communicants pour peu que ceux-ci soient majoritairement dans un rayon d'environ 500 m de la gateway.

La solution à ces limitations est d'augmenter le nombre de gateways dans une zone donnée. Ainsi les objets peuvent utiliser des datarates plus élevés et le temps de communication diminue minimisant ainsi le risque de collision.

3.1.6 ADR

L'exemple ci-dessus montre bien qu'il est très intéressant d'utiliser le datarate le plus élevé possible pour communiquer afin de limiter les interférences radio. Comme cela réduit la portée radio, il faut trouver un mécanisme pour appliquer le datarate maximum pour lequel la communication est toujours possible avec le end-device.

La LoRaWAN est une norme très intéressante car il est possible de moduler automatiquement le datarate des end-devices grâce au mécanisme d' « Adaptative Data Rate » (ADR). Si le end-device active cette fonctionnalité, le serveur analyse à la réception d'un message uplink la puissance du signal radio capté par la gateway. En fonction de la marge radio définie et de la qualité du signal radio capté, le serveur peut demander au end-device via un message downlink d'augmenter ou d'abaisser son datarate.

3.1.7 Commandes MAC, Coding Rate

LoRaWAN définit des commandes MAC permettant d'échanger des messages de signalisation réseau entre le end-device et le serveur. Ces commandes sont ajoutées dans les entêtes des messages LoRaWAN en plus de la charge utile des données utilisateurs. Typiquement, les commandes MAC échangées permettent la reconfiguration des paramètres réseau du end-device (datarate, intervalle d'écoute pour les messages downlink, canaux LoRa disponibles).

3.2 Solution déployée

Pour chaque maillon de l'infrastructure LoRaWAN, il existe plusieurs implémentations matérielles et logicielles disponibles. Depuis 2017, l'offre s'est considérablement diversifiée, tant dans la communauté open source qu'industrielle, prouvant ainsi le potentiel de cette technologie.

3.2.1 Serveur LoRaWAN

Initialement, Inetlab a développé en 2016 sa propre implémentation d'un LoRaWAN server. Le but était de pouvoir modifier le standard en fonction des améliorations souhaités. Au regard des avancées des différents projets, nous avons basculé vers l'implémentation LoRa Server [15] en 2018. Les principales motivations qui nous ont poussé à choisir cette implémentation sont :

- la qualité du code écrit en Golang ;
- le dynamisme du créateur du projet et de la communauté gravitant autour ;
- le suivi des évolutions du standard.

Au-dessus d'une API REST très complète, LoRa Server propose une interface graphique conviviale pour déléguer aux utilisateurs du réseau l'enregistrements de nouveaux objets. LoRa Server est également multi-tenant et permet de segmenter les utilisateurs en organisation. Chaque organisation a la possibilité de déclarer ses propres gateways. Au final, toutes les gateways de l'ensemble des organisations sont mutualisées pour profiter d'une couverture optimale entre utilisateurs. Au niveau du projet, l'Eurométropole est propriétaire de ses propres bornes déclarées dans une organisation dédiée. Elle profite des autres bornes déployées par Inetlab et inversement (voir Figure 2).

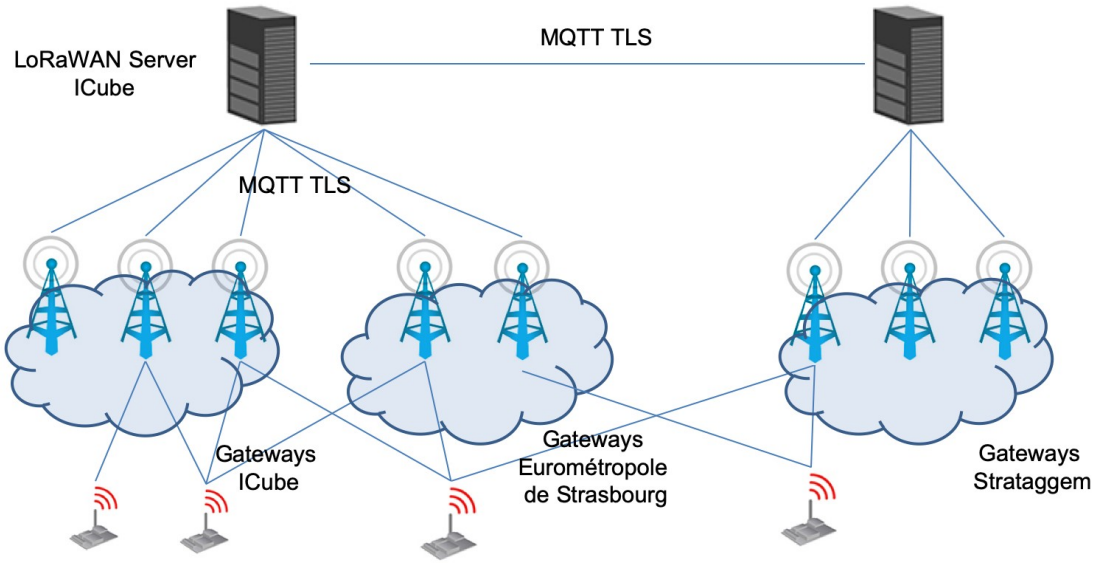

Figure 2 - Infrastructure mutualisée LoRaWAN

Dans notre déploiement nous avons créé une instance unique de LoRa Server à l'aide une machine virtuelle KVM dotée de 4 VCPU et 8 GB de RAM.

Les données entre le serveur et les gateways sont sécurisées via le protocole MQTT [16] avec TLS et une QoS activée au niveau 2 garantissant la retransmission des paquets entre les gateways et le serveur en cas de coupure réseau. Le serveur MQTT utilisé sur l'ensemble de l'infrastructure est Mosquitto [17]. Nous utilisons une architecture MQTT distribuée qui nous permet d'utiliser une file d'attente sur chaque noeud (gateway ou LoRaWAN server) en cas de coupure réseau ou d'une maintenance du serveur. Les différents instances MQTT sont reliées entre elles par l'intermédiaire de « bridges » MQTT. En bout de chaîne, une instance MQTT dédiée autorise la consultation des données en temps réel pour les utilisateurs authentifiés sur la base de données PostgreSQL de LoRa Server.

3.2.2 Exploitation des données uplinks/downlinks

Coté utilisateur, il existe trois possibilités pour récupérer les messages uplinks des end-devices :

- via l'interface graphique, les messages ne sont pas enregistrés directement par LoRa Server dans une base de données ;
- via une intégration HTTP, le contenu du message est transféré vers un service web, généralement couplé à une base de donnée ;
- via le protocole MQTT, l'utilisateur utilise un client MQTT ou une application basée sur une librairie MQTT pour intégrer le flux des messages vers son système d'information.

Pour l'envoi des messages downlink à destination des end-devices, seul le protocole MQTT est autorisé à l'heure actuelle.

Avec ses utilisateurs, l'Eurométropole a expérimenté l'intégration HTTP vers le service en ligne TagoIO [18] permettant de stocker des données issues de l'IoT et de créer des tableaux de bord partagés. Cette solution est limitée car le flux de données est unidirectionnel dans le sens LoRa Server vers service web. Via l'intégration HTTP, il est impossible de faire descendre un message du service web vers LoRa Server pour lancer un downlink à destination du end-device.

Inetlab exploite la remontée des messages via le protocole MQTT grâce à une application Python qui insère les messages uplinks reçus dans une base NoSQL de type Elasticsearch [19]. Les motivations principales de ce choix sont le moteur de recherche performant, le passage à l'échelle en termes de volume d'enregistrements grâce au clustering ainsi que l'écosystème des outils de visualisation disponibles, comme par exemple Kibana [19] ou Grafana [20].

Nous avons également expérimenté Thingsboard [21], un outil auto-hébergé de collecte de données orienté IoT permettant de créer des tableaux de bord. La version « Community » couplée au module iot-gateway permet de récupérer directement les données issus de LoRa Server via MQTT. A l'heure actuelle, seuls les messages uplink ont été remontés avec succès. Sur le même modèle que LoRa Server, Thingsboard propose également une gestion des utilisateurs dans des organisations. La configuration relativement complexe de Thingsboard nécessite toutefois un fort investissement pour un paramétrage efficace.

3.2.3 gateways

Depuis 2016, énormément d'équipementiers (par ex. Kerlink, Multitech, Cisco) proposent maintenant des gateways LoRaWAN. Notre approche scientifique d'évaluation des performances du protocole LoRaWAN nous impose d'avoir la main sur l'ensemble de l'infrastructure et de pouvoir faire une introspection interne des performances des gateways. Nous avons donc retenu la solution matérielle basée sur un concentrateur LoRa IMST IC880A relié via le bus SPI à une carte Raspberry Pi 3. Sur la base de ces composants, Strataggem, partenaire du projet, a développé une gateway sur mesure alimentée via PoE (voir Figure 3).

Figure 3 - Exemple de gateway LoRaWAN équipée d'une antenne omnidirectionnelle

Nous avons testé trois types d'antennes. Le Tableau 2 résume les performances mesurées :

type d'antenne	portée observée
indoor 20 cm	1 km
outdoor type trident 50 cm	4 km
outdoor omnidirectionnelle 100 cm	15 km (jusqu'à 70 km observé entre un ballon sonde et une gateway du réseau)

Tableau 2 - Performances des antennes gateways

Au niveau logiciel, plusieurs démons sont nécessaires pour relayer les messages. Le démon packet_forwarder [22] développé par Semtech convertit les paquets radio LoRa en paquets IP/UDP. Bien que le standard spécifie une remontée des donnés vers le serveur sur ce format de données, nous les convertissons en local sur la gateway en paquet MQTT grâce au démon LoRa Gateway Bridge [23] pour sécuriser les paquets et profiter des mécanismes de QoS.

3.2.4 End-devices

Nous avons pu tester différentes combinaisons de puces radio et microcontrôleurs programmés sous Arduino ou Micropython :

- Carte d'évaluation Microchip LoRaMote 10422 ;
- Modem Microchip RN2483 + Arduino (Mega, FIO) ;
- Arduino MKR1300 ;
- Pycom LoPy (v1 et v4).

Outre les problèmes de firmwares des modems LoRa décrits dans section 4.2.1, le principal problème de ces montages testés est la performance énergétique lors du passage en état de veille. En effet, certains microcontrôleurs ont des erreurs de conception matérielle empêchant d'atteindre les seuils de consommation énergétique minimum spécifiés.

Coté performances radio, deux d'antennes ont été testées :

- interne : de type « fouet », souple et d'une longueur de 11 cm, on peut facilement les cacher à l'intérieur du boîtier de l'objet ;
- externe : rigide et d'une longueur de 20 cm, on fixe son embase à l'extérieur du boîtier.

L'antenne externe plus rigide et plus grande offre de meilleure performances. Mais pour des applications qui nécessitent un objet plus petit ou plus discret, l'antenne fouet est un choix pertinent. Dans ce cas, le end-device devra se trouver dans une zone plus proche de la gateway.

3.2.5 Peering opérateur

En partant d'un déploiement initial opéré par Inetlab, nous avons souhaité intégrer en plus de l'Eurométropole les bornes de Strataggem, qui est lui-même opérateur d'un réseau LoRaWAN sur l'agglomération de Strasbourg. En tant qu'opérateur, Strataggem souhaite pouvoir conserver sa propre instance de LoRaWAN server. Dans le cadre de l'expérimentation, nous avons pris parti de router directement tous les paquets captés sans distinction entre opérateurs. Techniquement, nous avons élaboré un peering entre serveurs MQTT grâce à un lien de type bridge et nous avons déclarés les gateways des opérateurs distants sur chaque instance du LoRaWAN server. Cette solution n'est que moyennement satisfaisante car elle ne facilite pas le passage à l'échelle. Elle risque également de dépasser le quota d'usage de 1% en émission des gateways (voir paragraphe 3.1.5), car chaque opérateur maintient un quota différent pour ses gateways utilisées.

3.3 Exemples de cas d'usage déployés

3.3.1 Instruments d'études climatologique

Dans le cadre d'études de micro-climats urbains, Inetlab déploie à l'échelle d'un quartier des capteurs environnementaux tels que des stations météo, pluviomètres, pyranomètres, etc. Initialement, les données de ces instruments équipés de mémoire interne étaient relevées à la main. En 2015, une première tentative de déploiement d'instruments connectés basée sur des réseaux maillés Zigbee s'est révélée trop contraignante car la distance maximale de 50 mètres entre chaque noeud nécessite un réseau trop dense par rapport aux zones instrumentées plus larges. L'utilisation du LoRa avec une topologie en étoile permet de manière simple, et relativement abordable financièrement de déployer à l'endroit souhaité nos instruments.

3.3.2 Monitoring et télégestion de l'éclairage public

L'Eurométropole expérimente sur un quartier entier la migration de la gestion de son éclairage public actuellement piloté par courant porteur. Via le réseau LoRaWAN, ce système est en mesure de :

- commander à distance des scénarios d'illuminations sur des ouvrages d'art ou des édifices publics ;
- télégérer des armoires de commande d'éclairage public (ordre d'allumage ou d'extinction, relevé de consommation, remontée d'alertes en cas de panne) ;
- télégérer des points lumineux (changement d'horaires, remontée d'alertes).

4 Bilan

4.1 Statistiques et performances observées

4.1.1 Monitoring et performances du réseau

Depuis le début de l'expérimentation, Inetlab a étudié la montée en charge du réseau LoRaWAN. Nous avons captés plus de 36 millions de messages à l'aide de plus de 50 gateways. Depuis juin 2019, le peering opérateur entre Inetlab et Strataggem nous permet de considérablement augmenter le nombre de messages remontés (voir Figure 4).

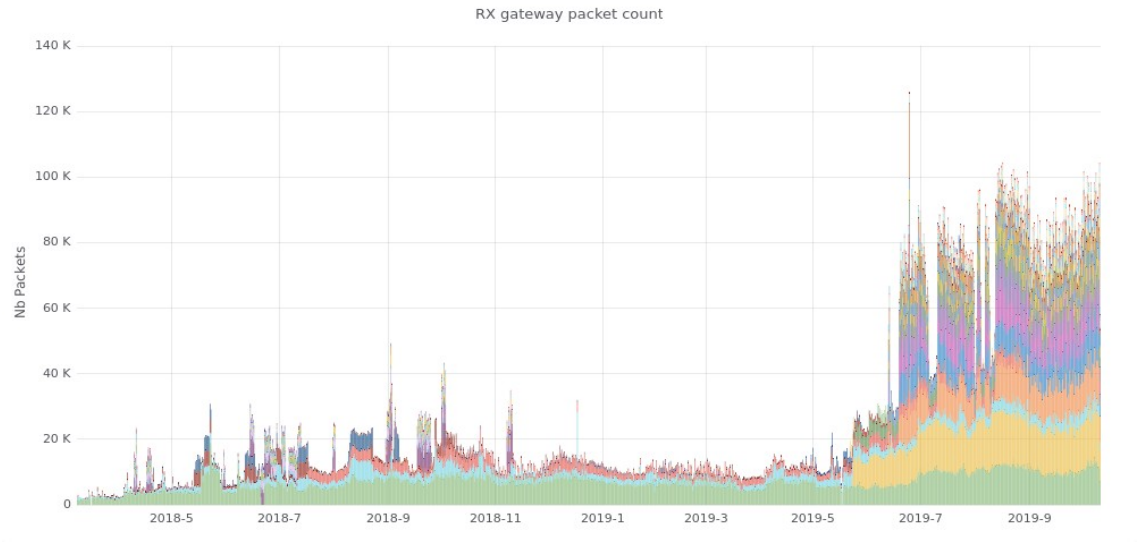

Figure 4 - Nombre de paquets reçus entre mars 2018 et octobre 2019

La gateway qui physiquement se trouve sur le point le plus haut (voir Figure 3) équipée d'une antenne omnidirectionnelle de 1 mètre a capté plus d'un quart des messages (voir Figure 5) mettant ainsi en évidence l'importance du positionnement des antennes.

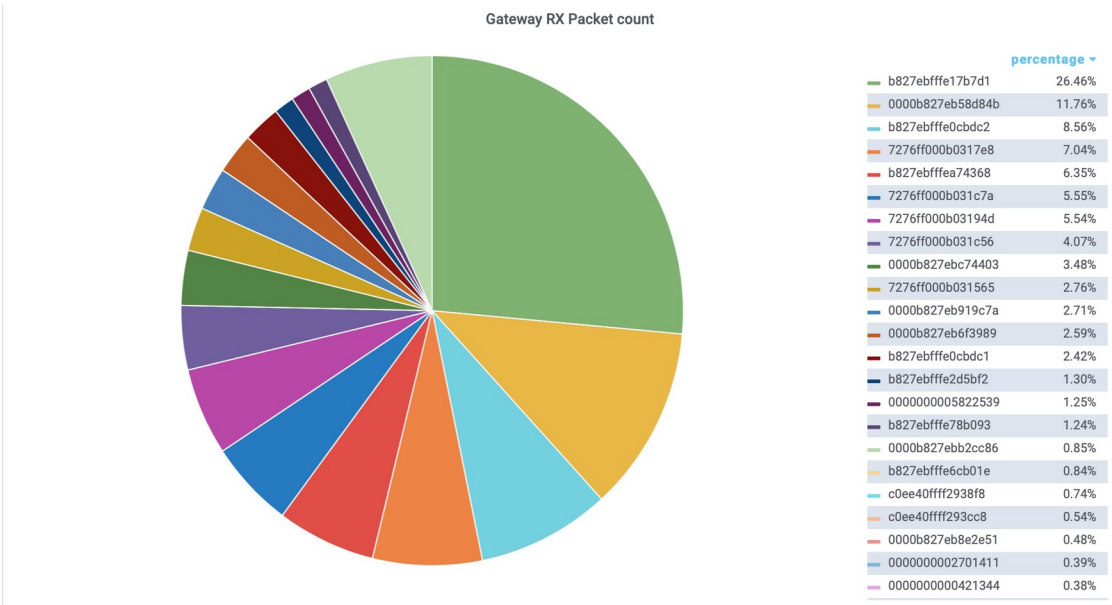

Figure 5 - Pourcentage de paquets reçus par gateway

Concernant les puissances émises par les objets, la puissance la plus faible SF7 correspond à 43% des messages suivie de la puissance la plus forte SF12 qui compte 33% des messages (voir Figure 6). Les autres puissances correspondent à des états transitoires le plus souvent liés à l'ADR qui fixe automatiquement la puissance. Il est à noter que généralement lors d'une phase d'accès au réseau (Join),

le end-device s'enregistre avec la puissance maximale SF12 pour être sûr de trouver une gateway à portée, puis il baisse progressivement sa puissance selon l'algorithme ADR implémenté par le serveur.

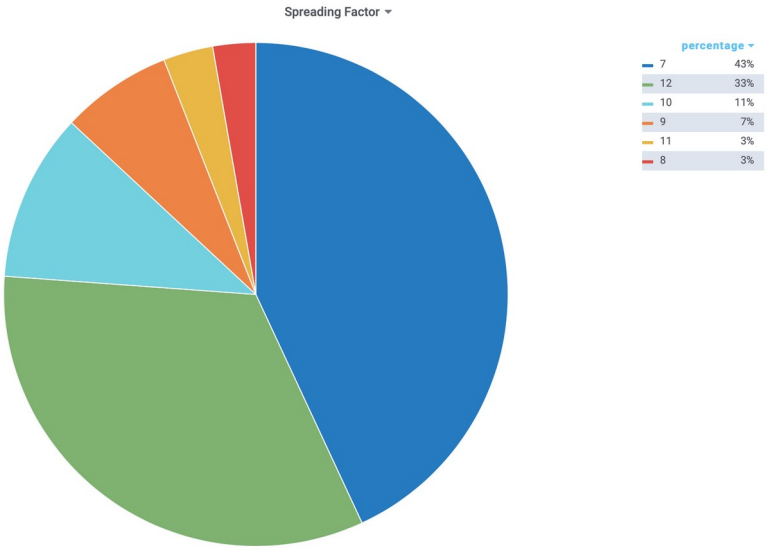

Figure 6 - Répartition des paquets reçus par puissance d'émission

En terme de taille de paquet, plus de 50% des paquets font moins de 30 octets (voir Figure 7) soit une taille bien en dessous de celle maximum autorisée de 51 octets pour la puissance maximale SF12.

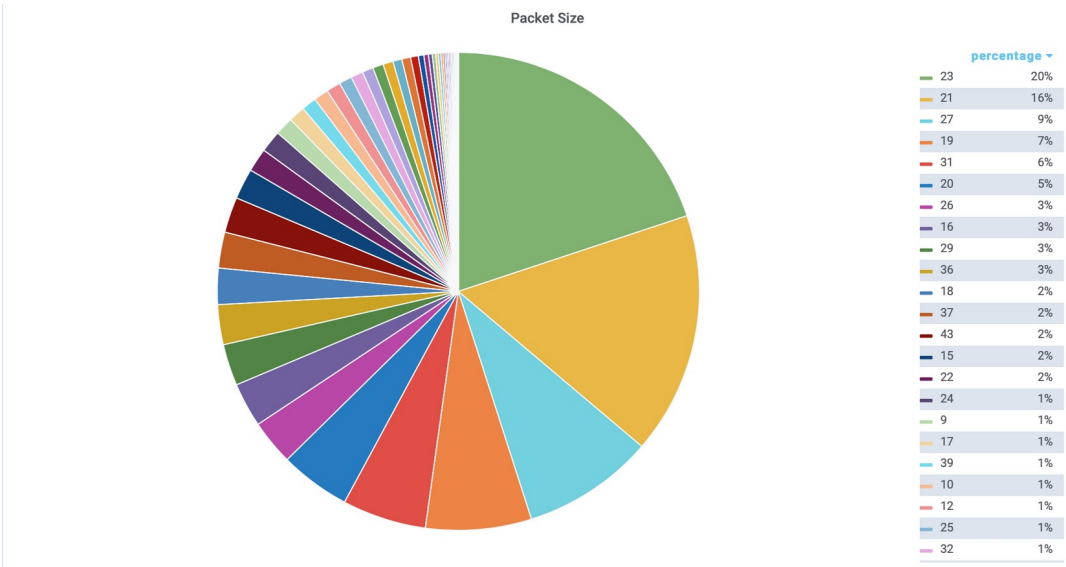

Figure 7 - Répartition des paquets reçus par taille

Au niveau des fréquences utilisées, quatre sur dix monopolisent 99% des messages émis voir Figure 8). On observe que les trois fréquences spécifiées par défaut par la norme (868.1MHz, 868.3 MHz et 868.5 MHz) ne sont que très peu modifiées par les fabricants.

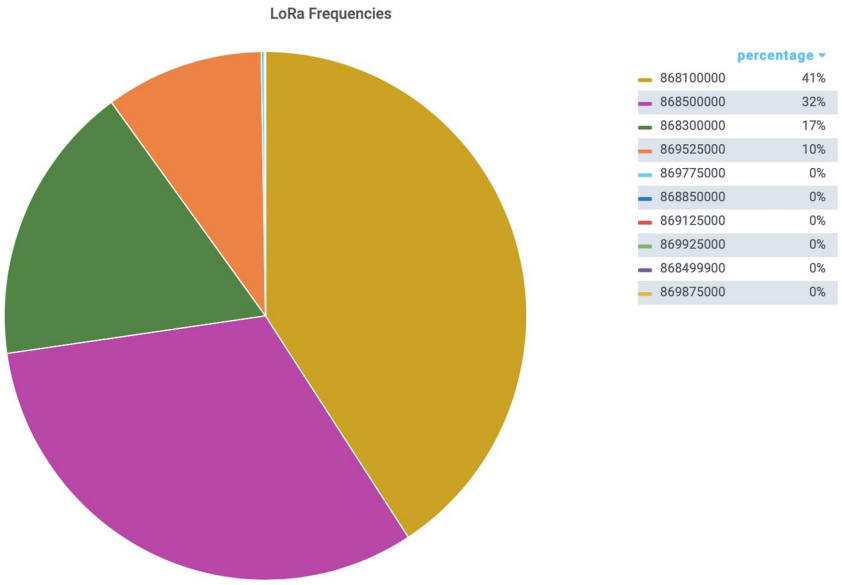

Figure 8 - Répartition des paquets reçus par fréquences

4.1.2 Cartographie radio

Afin de valider la couverture radio de notre déploiement, nous avons conçu des end-devices équipés de GPS. Régulièrement, des campagnes de mesures sont effectuées par des personnes en situation de mobilité (pied, vélo, transport en commun, voiture). Toutes les 30 secondes, le end-device relève sa coordonnée GPS puis l'envoie sur le réseau LoRaWAN. Si le paquet est capté par une gateway, on considère la position courante comme couverte. La Figure 9 montre un aperçu de la couverture radio observée. Les marqueurs rouges symbolise les gateways et les taches de couleur les positions couvertes. Les cercles bleus correspondent à la portée maximum pour une gateway donnée, soit 15 km à pleine puissance avec la gateway la plus haute (voir Figure 3).

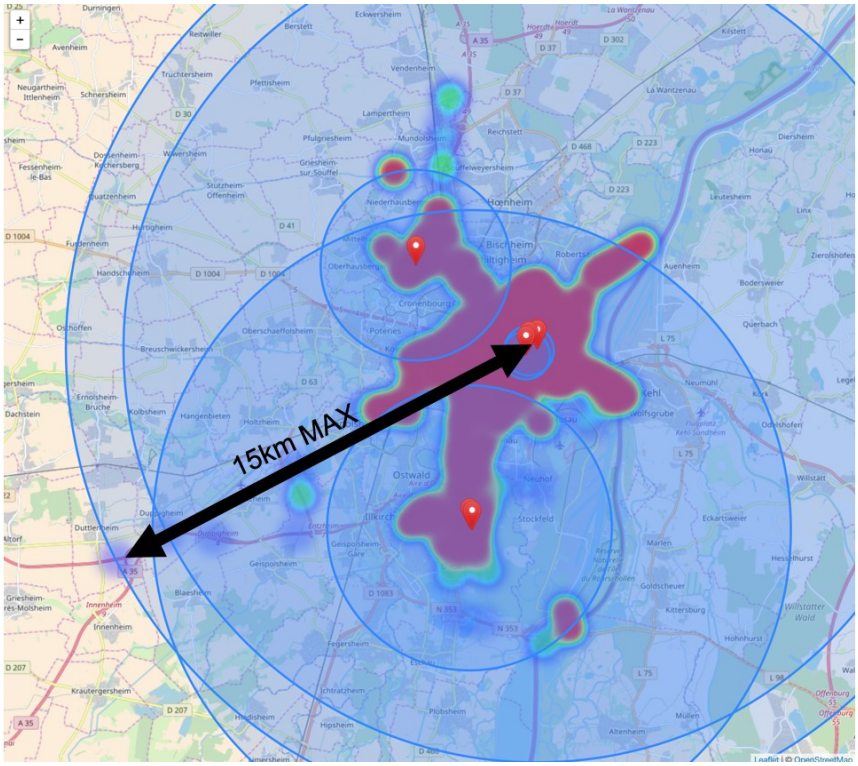

Figure 9 - Cartographie de la couverture radio

4.2 Difficultés rencontrées

4.2.1 Interopérabilité techniques

Lors des premiers essais de déploiement menés en 2016, la relative jeunesse des implémentations a mis en évidence des faiblesses d'implémentations aussi bien côté serveur que end-devices, notamment pour les procédures d'enregistrement dynamique au réseau (OTAA) ou la gestion de l'ADR. Via une mise à jour des firmwares des modems, ces problèmes ont été corrigés. Pour une flotte de end-devices déployés, il convient de vérifier que tous leurs modems ont la même version de firmware afin de garantir un comportement homogène et plus facilement analyser les comportements réseaux entre end-devices.

4.2.2 Formats de messages

Rapidement, nous avons identifié le besoin de structurer le format des données applicatives codées au niveau de l'objet et décodées coté serveur. Il convient de compresser au maximum les données pour économiser le poids du message radio fortement contraint en LoRa. Progressivement, d'une structure JSON qui utilise énormément de caractères de signalisation, nous avons évolué vers un format TLV (Type, Length, Value) pour finalement converger vers le format Cayenne LPP [24] optimisant la taille des données en fonctions du type de capteur. De nombreuses librairies pour les end-devices supportent ce format de données. LoRa Server implémente un codec qui décode automatique ce format.

4.2.3 Sécurisation entre le serveur et les gateways

Le protocole de communication standard entre les gateways et le serveur ne spécifie pas d'authentification au niveau applicatif. Pour restreindre l'accès au serveur, il faut donc ajouter des règles

de pare-feux autorisant les adresses IP des gateways. Dans le cas d'un déploiement sur une infrastructure de type campus, nous avons dans un premier temps déployé un VLAN transversal sur les différents sites pour raccorder les gateways au serveur. Ce VLAN ne pouvant pas arriver sur tous les campus, nous avons alors utilisé un tunnel OpenVPN [25] de niveau 2. Finalement, en basculant du mode de transport IP/UDP non authentifié vers MQTT TLS, nous avons résolu l'ensemble de nos contraintes. Les connexions sont possibles derrière n'importe quelle adresse publique ou privée, IPv4 et même IPv6.

4.2.4 Mise à jour distantes des gateways

Nous avons été confrontés au problème de mises à jour logicielles distantes des gateways qui n'ont pas été toutes déployées en même temps et de plus avec des versions logicielles différentes. Les outils classiques de déploiement de code ou de mise à jour ne garantissent pas forcément l'homogénéité du parc. Pour résoudre cette contrainte, nous avons intégré à notre infrastructure le service Mender [26] un outil de management de firmware pour OS embarqués. Sur une architecture de type client/serveur, le serveur de mise à jour Mender pousse les nouvelles images sur l'agent installé sur la gateway. Le disque de la gateway est découpé en deux partitions système A et B, chaque mise à jour réécrit totalement la partition non-active, puis redémarre dessus si le transfert a été effectué avec succès. En cas d'échec, il revient sur l'image précédente. Lors de son intégration, nous avons migré d'une image Linux Raspbian [27] vers Linux Yocto [28] un prérequis à l'époque pour Mender. Nous sommes passé d'un firmware d'une taille de 500 Mo avec Raspbian à 100 Mo grâce à Yocto, accélérant de ce fait le déploiement du serveur vers les gateways.

4.2.5 Interopérabilité et résilience des offres externalisées

Au niveau de l'Eurométropole, le constat a été fait que les plateformes d'exploitation des données des objets connectés proposées par les acteurs du marché sont aujourd'hui intégrées dans des offres métier incluant les capteurs, l'abonnement au réseau IoT bas-débit et la plateforme de supervision, sans possibilité de dissocier chacune de ces composantes.

Ce type d'architecture est problématique dans la mesure où les applications métier fonctionnent de fait en silo et sont difficilement interfaçables. De plus l'usage d'un réseau LoRa spécifique comme celui mis en oeuvre dans cette expérimentation n'est pas toujours possible. Enfin, lorsque le marché avec le fournisseur arrive à échéance, les capteurs peuvent devenir inutilisables du fait que leur paramétrage n'est pas ouvert et qu'ils ne sont compatibles qu'avec la solution du fournisseur original.

L'expérimentation a permis de mettre en évidence ces problématiques et d'anticiper les marchés futurs en incluant des clauses restrictives obligeant les fournisseurs de solution à proposer des systèmes ouverts.

4.3 Investissements financiers

Pour Inetlab, le cout global estimé d'investissement matériel est d'environ 12 k€ pour 10 gateways et 60 end-devices. Dans ce calcul, nous n'incluons pas l'hébergement des serveurs.

4.3.1 Objets connectés

Les end-devices LoRaWAN que nous avons expérimentés ont été assemblés et programmés sur mesure. Leur coût moyen est de 100 €, il inclut :

- le microcontrôleur équipé de la puce LoRa (50 €) ;
- le(s) capteurs ou/et actionneur(s) (20 €) ;
- une batterie de type LiPo (10 €) ;
- une antenne (10 €) ;
- un boitier (10 €).

Pour un passage en exploitation, les fabricants proposent des end-devices spécialisés prêt à l'usage. Il suffit alors de les enregistrer sur le serveur et de les activer.

4.3.2 Gateway

Le coût moyen de nos gateways indoor est relativement bas (environ 210 €). Leur composition est la suivante :

- une carte Raspberry Pi 3 (40 €) ;
- un concentrateur LoRa iC880A-SPI (119 €) ;
- un PoE splitter ou une alimentation secteur (20 €) ;
- un boitier (20 €) ;
- une antenne simple (10 €).

Les gateways outdoor testées ont un prix qui grimpe à environ 1000 € du fait d'un boîtier renforcé résistant aux intempéries, de supports de fixations, de l'antenne omnidirectionnelle longue portée, des câbles coaxiaux renforcés.

4.3.3 Serveur

Voici l'ensemble des ressources matérielles attribuée à notre solution LoRaWAN hébergée dans un cloud OpenNebula [29]orchestrant des hyperviseurs KVM :

- VM LoRa Server (4 CPU, 8 Go de RAM) ;
- VM Mosquitto frontal utilisateurs (4 CPU, 4Go de RAM) ;
- VM Elasticsearch (4 CPU, 16Go de RAM) ;
- VM Grafana (2 CPU, 4Go de RAM) ;
- VM Mender (2 CPU, 4GB de RAM).

4.3.4 Infrastructure réseau et câblage

La mise en réseau d'une gateway LoRaWAN est sensiblement identique à celle d'un point d'accès Wi-Fi. Nous avons utilisés majoritairement des commutateurs PoE coûtant en moyenne 1000 €. Coté câblage, le coût moyen à la prise pour le déploiement sur le toit des bâtiments est d'environ 1000 €.

4.4 Évolutions de l'infrastructure

4.4.1 Haute-disponibilité du serveur

Dans le cadre de la phase d'expérimentation, nous n'avons pas déployé de services critiques nécessitant une haute-disponibilité. Afin de rendre le serveur LoRaWAN de notre solution totalement redondant voici les points à considérer :

- le clustering du démon LoRa Server sur plusieurs serveurs distincts synchronisés via le protocole gRPC et un load balancer ;
- le clustering de la base PostgreSQL pour la persistance des ressources enregistrées ;
- le clustering de Redis pour la mémoire partagées entre les instances LoRa Server ;
- le clustering du serveur MQTT Mosquitto.

4.4.2 Géolocalisation par les gateways

Dès lors qu'un end-device est capté par au moins trois gateways, il est possible d'utiliser un algorithme de triangulation basé sur le temps de réception afin de localiser sa position. Pour arriver à ce but, les gateways doivent posséder une horloge synchronisée à la nanoseconde. Dans notre cas, une modification matérielle consistant à ajouter une RTC ainsi qu'un GPS est nécessaire.

4.4.3 Mise à jour OTA des end-devices

Dès lors que les end-devices sont déployés sur le terrain, il devient fastidieux de modifier leur firmware pour corriger un bug ou améliorer l'application. Cette opération implique généralement une intervention manuelle sur le terrain pour reprogrammer chaque end-device. LoRa Server propose pour certains end-devices (Pycom) de déployer Over The Air (OTA) un nouveau firmware fragmentés en plusieurs messages downlinks. Une autre stratégie pour les end-devices possédant plusieurs technologies radio est de déclencher via LoRaWAN une mise à jour téléchargée via le Wi-Fi plus rapide en bande passante mais plus coûteux en énergie.

5 Conclusion

Après trois années d'expérimentations, nous avons tiré divers enseignements sur l'exploitation d'un réseau LoRaWAN à l'échelle d'une agglomération. Avec un investissement financier relativement faible de quelques dizaines de milliers d'euros, il est possible d'obtenir une très bonne couverture radio autorisant l'utilisation d'un objet connecté en tout point de la ville. L'approche d'un réseau multi-partenaires a été bénéfique pour la densification du réseau et la redondance de la couverture radio. Cette approche permet également des économies d'échelle en partageant le cout de l'infrastructure. Les solutions techniques actuelles autorisent ce type de déploiement. Toutefois, nous devons encore améliorer l'approche peering opérateur afin de respecter le temps d'émission limité des équipements LoRaWAN.

Concernant nos scénarios d'usage, nous avons pu traiter avec succès des cas variés liés aux villes intelligentes comme par exemple l'éclairage urbain, le suivi de personnes en mobilité. En particulier dans le cas de manipulations scientifiques sur le terrain, nous sommes passés d'une collecte manuelle et ponctuelle des données à une remontée en temps réel, libérant ainsi énormément de temps humain pour d'autres tâches plus intéressantes.

A l'échelle d'une collectivité ou d'une université, il devient intéressant de posséder sa propre infrastructure LoRaWAN afin de faire baisser le cout du déploiement IoT lié aux applications métiers en évitant le piège d'offres verticales segmentées. Arrivé à ce stade de convergence, le prochain défi est l'analyse et le recoupement des différentes sources de données pour exploiter toute la valeur de l'infrastructure sous-jacente.

Bibliographie

[1] Plateforme Inetlab (Internet Network Technologies Lab) ; https://inetlab.icube.unistra.fr/.

[2] ICube (UMR7357), le laboratoire des sciences de l'ingénieur, de l'informatique et de l'imagerie ; https://icube.unistra.fr.

[3] Eurométropole de Strasbourg ; https://www.strasbourg.eu.

[4] Strataggem ; https://www.strataggem.com/.

[5] Wi-Fi ; http://www.ieee802.org/11/.

[6] Zigbee ; https://zigbee.org.

[7] IEEE Standard for Low-Rate Wireless Personal Area Networks (LR-WPANs). IEEE Std 802.15.4-2015 (Revision of IEEE Std 802.15.4-2011), April 2016.

[8] Bluetooth ; https://www.bluetooth.com.

[9] NB IoT ; https://en.wikipedia.org/wiki/Narrowband_IoT.

[10] Sigfox ; https://www.sigfox.com.

[11] LoRa ; https://www.semtech.com/lora.

[12]LoRaWAN Specification ; https://www.lora-alliance.org/lorawan-for-developers.
[13]LoRa Alliance ; https://lora-alliance.org/.
[14]The Things Network (TTN) ; https://www.thethingsnetwork.org.
[15]LoRa Server, open-source LoRaWAN network-server ; https://www.loraserver.io.
[16]MQTT, Message Queuing Telemetry Transport ; http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/mqtt-v3.1.1.html.
[17]Eclipse Mosquitto, An open source MQTT broker ; https://www.mosquitto.org.
[18]TagoIO ; https://tago.io.
[19]Elastic ; https://www.elastic.co/fr/.
[20]Grafana, The open observability platform ; https://grafana.com.
[21]ThingsBoard, Open-source IoT Platform ; https://thingsboard.io.
[22]Lora network packet forwarder project ; https://github.com/Lora-net/packet_forwarder.
[23]LoRa Gateway Bridge ; https://www.loraserver.io/lora-gateway-bridge/overview/.
[24]Cayenne Low Power Payload ; https://developers.mydevices.com/cayenne/docs/lora/#lora-cayenne-low-power-payload.
[25]OpenVPN ; https://openvpn.net.
[26]Mender ; https://mender.io.
[27]Raspbian ; https://www.raspbian.org.
[28]Yocto ; https://www.yoctoproject.org.
[29]OpenNebula ; https://opennebula.org.