



INSTITUT UNIVERSITAIRE DE TECHNOLOGIE DE BÉZIERS

MÉMOIRE LICENCE PROFESSIONNELLE DE L'INTERNET DES OBJETS

**COMMENT L'IOT, PEUT-IL PERMETTRE UN SUIVI
DÉTAILLÉ ET UNE POTENTIELLE PRÉSERVATION DE
LA FAUNE ET DE LA FLORE EN MILIEU FORESTIER ?**

Par Buddy-Lee CELDRAN-SCHWARZ : Étudiant en Licence professionnelle de l'Internet des Objets

Tuteur académique : M. SEBASTIEN DRUON

IUT de Béziers 3, place du 14 juillet BP 50438
Béziers cedex Hérault 34505 France





Année universitaire 2019-2020

MÉMOIRE LICENCE PROFESSIONNELLE DE L'INTERNET DES OBJETS

COMMENT L'IOT, PEUT-IL PERMETTRE UN SUIVI DÉTAILLÉ ET UNE
POTENTIELLE PRÉSERVATION DE LA FAUNE ET DE LA FLORE EN
MILIEU FORESTIER ?

Présenté par **Buddy-Lee CELDRAN-SCHWARZ**

Numéro d'étudiant : 21700012

Sous la direction de **Sebastien DRUON**, Enseignant chercheur.

Préface :

Ce mémoire rentre dans le cadre de l'obtention du diplôme de Licence professionnelle de l'Internet des Objets. Il étudiera le fonctionnement de L'IOT dans le suivi et dans la préservation de la faune et de la flore dans le milieu forestier. L'idée de ce mémoire est venue du faite que le sujet du stage initialement prévu, portait sur la smartforest qui consistait à un suivi de l'évolution des forêts, de plus la préservation des forêts est un attendu majeurs avec les nouvelles technologies.

En effet, la préservation de la faune et de la flore est une préoccupation majeure, avec l'arrivé de l'Internet des Objets, il est devenu possible de suivre une évolution et d'agir en conséquence s'il y a nécessité.

Ce mémoire se veut être une contribution devant permettre de mettre en avant les avancés faits pour la préservation de la faune et de la flore, tout en apportant une solution ou une amélioration sur des projets existants.

Difficultés :

Remerciement :

Je tiens à remercier toutes les personnes qui m'ont aidée lors de la rédaction de ce mémoire.

Je voudrais dans un premier temps remercier, mon tuteur pour ce mémoire M.DRUON, enseignant chercheur à l'IUT de Béziers, pour sa patience, sa disponibilité et surtout ses judicieux conseils, qui ont contribué à alimenter ma réflexion.

Je remercie également toute l'équipe pédagogique de l'IUT de Béziers, pour avoir assuré la partie théorique de la formation.

Résumer (à faire en dernier) :

Sommaire

INSTITUT UNIVERSITAIRE DE TECHNOLOGIE DE BÉZIERS.....	1
COMMENT L'IOT, PEUT-IL PERMETTRE UN SUIVI DÉTAILLÉ ET UNE POTENTIELLE PRÉSERVATION DE LA FAUNE ET DE LA FLORE EN MILIEU FORESTIER ?.....	1
COMMENT L'IOT, PEUT-IL PERMETTRE UN SUIVI DÉTAILLÉ ET UNE POTENTIELLE PRÉSERVATION DE LA FAUNE ET DE LA FLORE EN MILIEU FORESTIER ?.....	2
Préface :.....	3
Remerciement :.....	4
Résumer (à faire en dernier) :.....	5
Liste des tableaux et graphiques :.....	7
Liste des abréviations :.....	8
Glossaires :.....	9
1. Introduction	11
2. Internet des objets (à modifier).....	13
2.1 Sa définition.....	13
2.2 Son fonctionnement (à modifier).....	14
2.3 Les protocoles réseaux utilisés (à compléter).....	16
Bluetooth Low Energy (BLE).....	16
ZigBee.....	16
Wi-Fi.....	17
4G (Réseau cellulaire).....	19
3 Le lien avec la faune et la flore.....	20
3.1 Les différents projets existants pour la préservation de la faune et de la flore.....	21
Chapitre 4 : Partie Tech.....	28
Simulation d'une situation de monitoring dans une forêt.....	28
Configuration de l'environnement	29
1.1 Le broker MQTT.....	29
1.2 La base de donnée InfluxDB.....	30
1.3 Télégraf.....	32
1.4 Grafana.....	34
2. Redirection des ports :.....	36
3. Test MQTT.....	37
Alertes :.....	43
Conclusion de la simulation :.....	46
Sources (A bien remplir):.....	46
Bibliographie :.....	47
Citations :.....	48

Liste des tableaux et graphiques :

Index des Figures

Figure 1: Nombre d'objets connectés dans le monde (Cisco Mars 2020).....	11
Figure 2: Fonctionnement objets connectés (ConnectWave).....	14
Figure 3: Cheminement des données (ConnectWave).....	15
Figure 4: Exemple d'architecture réseau ZigBee.....	17
Figure 5: Schéma du principe d'une architecture Wi-fi au niveau domestique.....	19
Figure 6: Architecture réseau du projet SmartForest.....	21
Figure 7: Schéma fonctionnement du projet paru dans IJCTER.....	23
Figure 8: Fonctionnement de la solution (IoTree).....	24
Figure 9: Exemple de l'analyse des sons.....	25
Figure 10: Dispositif Rainforest.....	25
Figure 11: Fonctionnement de la solution chez Rainforest.....	26
Figure 12: Schéma de la simulation.....	28
Figure 13: Paramètres BDD.....	35
Figure 14: Requêtes.....	36
Figure 15: Adresse serveur.....	36
Figure 16: Redirection sur un port spécifique.....	36
Figure 17: Interfaces clé 4G et point d'accès.....	37
Figure 18: Paramètres du point d'accès.....	37
Figure 19: Requête puissance en dBm.....	41
Figure 20: Requête signal Wi-Fi.....	42
Figure 21: Requête niveau de batterie.....	42
Figure 22: Requête niveau de température du dispositif.....	42
Figure 23: Vue globale des différents composants.....	43
Figure 24: WebHook serveur Discord.....	43
Figure 25: Configuration notifications Discord.....	44
Figure 26: Règle pour l'alerte de température.....	44
Figure 27: Paramètre du message.....	45
Figure 28: Message reçu serveur discord.....	45

Index des Tableaux

Tableau 1: Tableau des débits et portées de la technologie Wi-Fi en fonction de la norme.....	17
Tableau 2: Tableau des alertes.....	21
Tableau 3: Niveau de Co2 et impact sur l'Homme (th-industrie).....	22

Liste des abréviations :

IOT : Internet of Things

IDO : Internet des Objets

GSM : Global System for Mobile

GPS : Global Positioning System

M2M : Machine to Machine

Wi-Fi : Wireless Fidelity

4G : LTE/LTE-A

MQTT : MQ Telemetry Transport / Message Queuing Telemetry Transport (IBM)

Glossaires :

Gateways : dispositif permettant de relier deux réseaux distincts présentant une topologie différente.

Routeur : équipement en réseau informatique assurant le routage des paquets. Son rôle est de faire transiter des paquets d'une interface réseau vers une autre.

Objet connecté : objet possédant la capacité d'échanger des données avec d'autres entités physiques ou numériques.

Internet des objets (IDO) : expansion du réseau internet à des objets et/ou des lieux du monde physique. En anglais, on parle d'IoT : Internet of Things.

M2M : machine to machine, échange d'informations entre deux machines sans intervention humaine.

LoraWan : protocole de télécommunication permettant la communication à bas débit, par radio, d'objets à faible consommation électrique communiquant selon la technologie LoRa et connectés à l'Internet via des passerelles, participant ainsi à l'Internet des objets. LoRa est un réseau longue portée à débit compris entre 0,3 et 50Kbps soit un débit plus faible que le 2G.

Zigbee : ZigBee est un protocole de haut niveau permettant la communication d'équipements personnels ou domestiques équipés de petits émetteurs radios à faible consommation. Son débit est de 250 kb/s.

Wi-Fi : ensemble de protocoles de communication sans fil . Un réseau Wi-Fi permet de relier par ondes radio plusieurs appareils informatiques (ordinateur, routeur, smartphone, etc.) au sein d'un réseau informatique afin de permettre la transmission de données entre eux. Les débits peuvent varier en fonction du protocole utilisé (a/b/g/n/ac/ax) on peut avoir des débit compris entre 11 et 1000 Mb/s.

4G : Protocole de 4ème génération des standards de la téléphonie mobile. Ses débits varie en fonction de la catégorie du protocole (4,6,12), on a donc le LTE Cat 4 pour 150 Mbit/s, LTE-A Cat 6 pour 300 Mbit/s, LTE-A Cat 12 pour 600 Mbit/s. Dans ces variations, on a la 4G+ soit LTE-A(Advanced) qui propose des débit nettement supérieurs aux débits 4G classique.

MQTT : Il s'agit d'un protocole de messagerie de publication/abonnement extrêmement simple et léger, conçu pour les appareils à contraintes et les réseaux à faible bande passante, à forte latence ou peu fiables et/ou à faible autonomie.

Dashboard : (Tableau de bord) est un ensemble d'un ou plusieurs panneaux ou graphiques organisés et disposés en une ou plusieurs rangées.

1. Introduction

Depuis plusieurs années, on assiste à une modification de notre environnement du point de vue technologique, car de plus en plus d'objets physiques, sont connectés à l'Internet (montres, lunettes, caméra,...) . Cette innovation technologique a pour nom « l'Internet des objets » (IoT en Anglais), c'est une évolution en marche depuis plus de 20 ans avec la première désignation de ce terme émis par un employé de Procter & Gamble (P&G), Kevin Ashton, qui parlait alors du lien entre la technologie RFID et l'Internet.

Bien que le terme internet des objets a disparu entre 1999 et 2005. C'est en 2003 que le premier objet connecté est commercialisé par la firme Violet. C'est la lampe DAL, équipée de 9 leds qui s'allument en fonction des événements. Puis deux ans plus tard, c'est en 2005 que Violet lance le lapin Nabaztag.

Aujourd'hui, le nombre d'objets connectés augmente de manière exponentielle. D'après l'entreprise Cisco, on se rapprocherait des 22,5 milliards d'objets connectés en 2020 et 29 milliards d'objets connectés prévu pour 2023, comme le montre *la figure 1* du nombre d'objets connectés dans le monde (Mars 2020) :

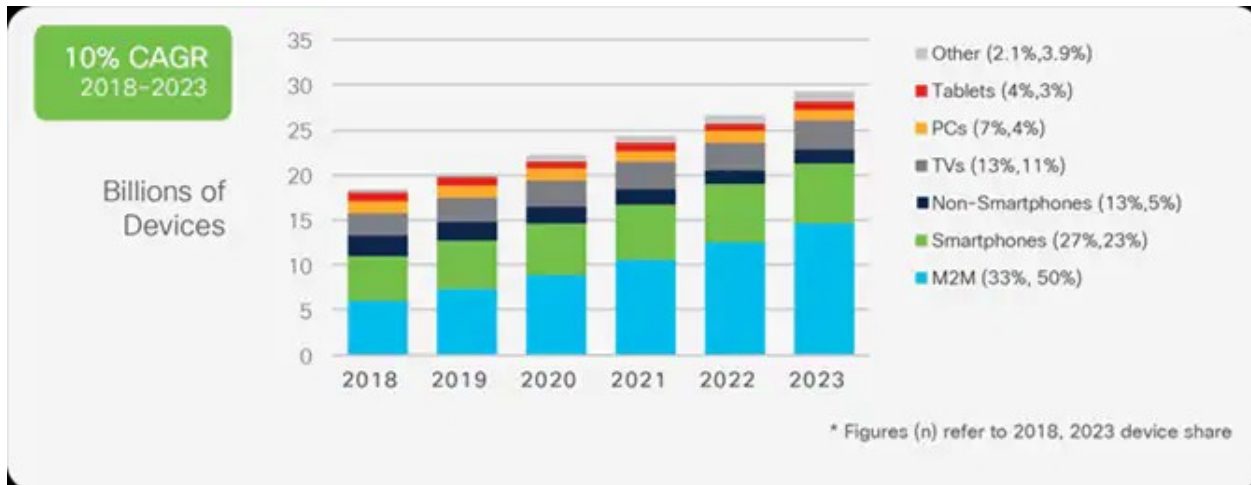


Figure 1: Nombre d'objets connectés dans le monde (Cisco Mars 2020)

Bien que de nombreuses possibilités voient le jour dans plusieurs secteurs, dans ce mémoire nous allons nous intéresser à la préservation de la faune et de la flore dans le domaine forestier, et ce, pour deux raisons.

Premièrement, l'Internet des objets va révolutionner la manière de surveiller les forêts et la manière dont les personnes pourront agir en conséquence. Ne serait-ce que pour la surveillance en cas d'incendies ou d'inondations.

Bien que les promesses soient très grandes, de nombreux challenges sont à surmonter pour pouvoir voir se réaliser les bénéfices espérés, notamment avec la gestion des dispositifs (réseaux, alimentation, ...).

Deuxièmement, même si l'internet des objets est une technologie qui séduit de plus en plus, il ne faut pas oublier, qu'il faut comprendre son fonctionnement et donc il faut se préparer à l'utiliser.

Le problème des nouvelles technologies est un problème assez connu :

Il faut impérativement s'informer et se positionner face à un nouveau phénomène ; cependant, il n'existe pas à notre connaissance d'outil simple pour la mise en place de solutions afin de surveiller la faune et la flore dans les forêts. Dans ce mémoire, nous tenterons d'établir une solution afin de palier à ce problème. Nous allons donc nous concentrer sur la mise en place de solutions possibles quelle que soit la grandeur du secteur à surveiller.

De manière concrète, nous allons mettre en place de façon théorique et de manière méthodologique, une solution applicable par certaines personnes ayant travaillé un minimum dans les réseaux et télécommunications. Nous chercherons le type de matériels à utiliser, l'alimentation et la mise en place d'une architecture réseau pour transmettre les informations souhaitées.

Dans ce mémoire, nous suivrons la logique suivante :

- Premièrement, nous ferons un état de ce qui existe en terme de projets et nous étudierons ces solutions afin de comprendre le fonctionnement d'une surveillance dans le milieu forestier (Chapitre 2 et 3).
- Deuxièmement, avec toutes les informations récoltées au travers des solutions existantes nous essaierons de créer la nôtre, avec la mise en place d'un réseau, du choix des paramètres à prendre en compte pour la surveillance de la faune et de la flore (programmation de capteurs) et de la gestion de l'énergie (Chapitre 4).

2. Internet des objets (à modifier)

Avant de commencer à référencer tous les projets en rapport à la préservation de la faune et de la flore dans le milieu forestier, il est nécessaire de comprendre le concept d'IOT, pour cela, on partira sur une explication en 3 étapes : sa définition, comment fonctionne cette technologie et les protocoles réseaux utilisés pour l'IOT.

2.1 Sa définition

Pour reprendre la citation de M. Han and H. Zhang l'internet des objets serait considéré comme « un réseau qui relie et combine les objets avec l'Internet, en suivant les protocoles qui assurent leurs communications et échange d'informations à travers une variété de dispositifs. » [1]

Pour approfondir cette citation, on peut expliquer que les objets connectés possèdent leurs propre « identité numérique » et qu'ils sont capables de communiquer entre eux. Ce réseau d'objets connectés crée une passerelle entre un monde physique et un monde numérique. Dans certains cas, on parle aussi d'interaction numérique-physique. L'IoT commence ainsi dans le monde physique avec les capteurs qui récupèrent des informations, ces informations sont ensuite transmises grâce à différents protocoles de communication (LoraWan, Zigbee,...), les données sont ensuite traitées et stockées pour être analysées et exploitées par l'utilisateur.

On retrouve cette technologie principalement dans ces 4 grands domaines :

L'industrie : La surveillance et la réduction des dépenses énergétiques, la gestion des alertes,...

La ville : Avec la gestion des places de parking, avec des parkings connectés, la gestion en eau et électricité de la ville.

L'environnement : La surveillance des espaces pleins airs, la gestion des arrosages automatiques

La santé : Suivi des patients et des machines connectés.

2.2 Son fonctionnement (à modifier)

Comme décrit dans la partie précédente, un objet connecté permet la récupération d'informations via des capteurs et permet de transmettre ces informations via l'interconnexion des objets sur le réseau, ainsi son fonctionnement se déroule de la manière suivante :

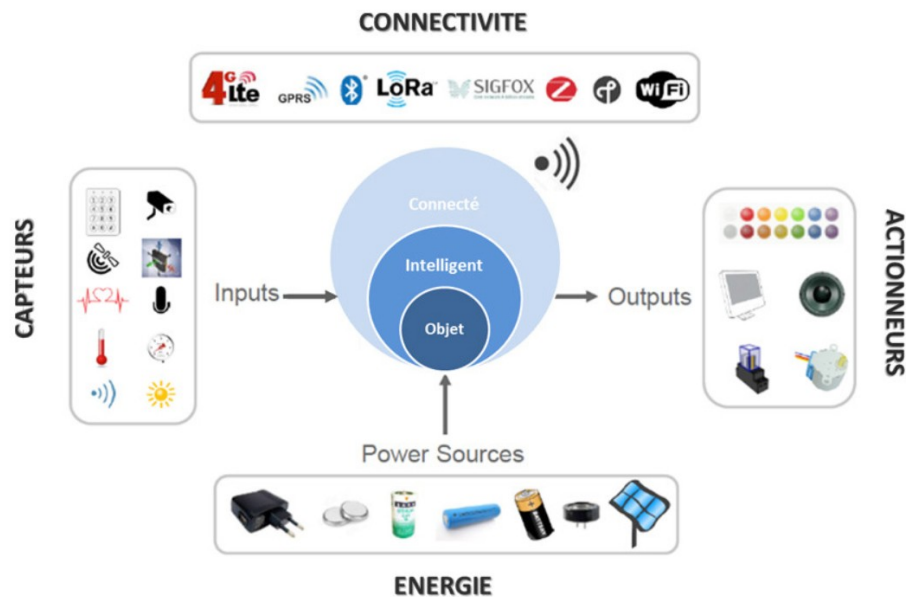


Figure 2: Fonctionnement objets connectés (ConnectWave)

L'objet possède un ou des capteurs qui interagissent avec le monde physique, puis ces informations sont traitées et envoyées via différents protocoles de communication comme LoraWan, Zibgee, Wi-Fi, 4G, ... Bien sûr un objet connectés à besoin d'être alimenté soit par une batterie, soit par une pile bouton dans certains cas. Une fois les données reçues, certains objets peuvent actionner des composants (valve, écran,...)

Ensuite, une fois le traitement effectué (envoi des données,...), on passe sur une plus grande échelles avec d'autres objets qui envoient aussi des données en parallèle :

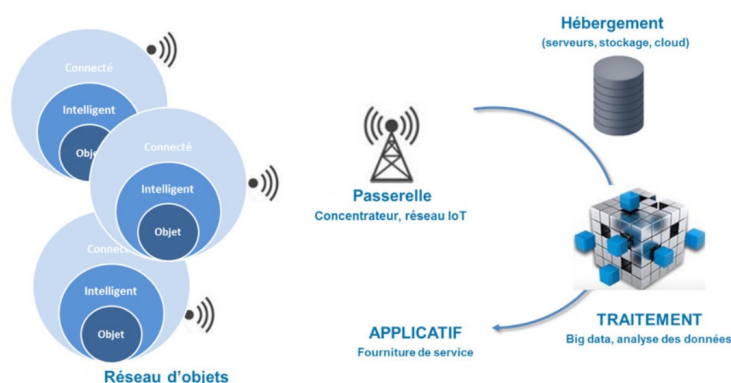
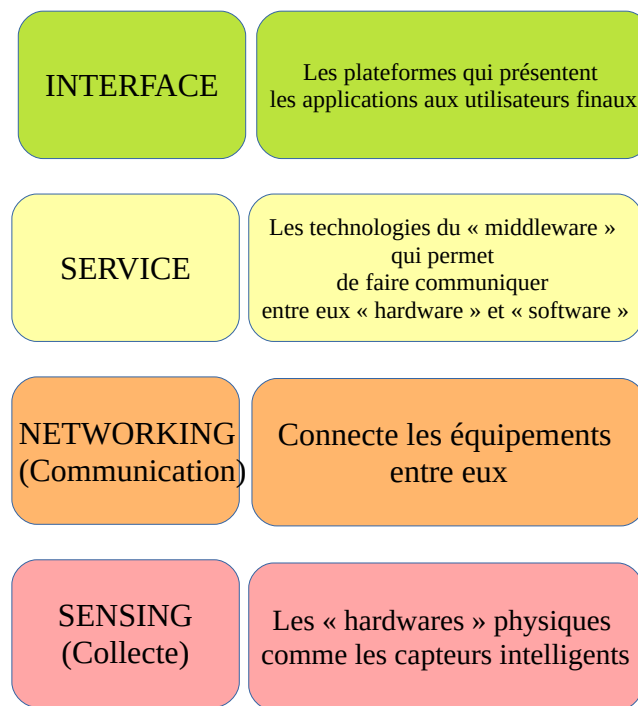


Figure 3: Cheminement des données (ConnectWave)

Chaque objet envoie ces données vers un ou des serveurs en passant par une passerelle qui redirigera ensuite les données vers le serveur adéquat. Ensuite, une analyse des données reçues est effectuée. Ces données peuvent être utilisées pour des utilisateurs (Application et service) ou alors pour de la supervision ou sécurisation.

On peut alors en ressortir 4 couches en rapport à cette technologie :



En ce qui concerne le choix du protocole à utiliser, on peut le choisir en premier lieu en fonction du besoin, si ce sont des données importante alors privilégié des protocoles qui ont un taux de transfert rapide comme le Wi-Fi ou la connexion cellulaire 4G, si au contraire se sont des données de faible taille alors , ZigBee ou encore le Bluetooth Low Energy (BLE) peuvent être utilisés.

On peut aussi choisir le protocole en fonction de la superficie à couvrir dans notre cas, il peut s'agir d'une petite forêt ou d'une réserve naturelle comme celle de la « Réserve Naturelle Nationale de la Forêt de Massane » proche de Banyuls sur Mer. Dans ces cas-là, on peut utiliser si ce sont des petites superficie des réseaux à courte portée comme le Wifi, ZigBee, ou encore le Bluetooth Low Energy, car ils permettent de transférer des données sur de faibles distances. S'il s'agit de grandes distances, on pourra se focaliser sur des réseaux longue portée avec une faible consommation comme Sigfox, LoRa ou encore les technologies cellulaires (GSM, 2G, 3G, 4G...).

2.3 Les protocoles réseaux utilisés (à compléter)

L'établissement d'une connexion consiste à résoudre le principal problème, à savoir établir une méthode de communication, et les différentes méthodes utilisées sont influencées par les contraintes imposées aux appareils connectés au réseau. L'une des contraintes est celle de l'autonomie, et cette contrainte est généralement la plus importante.

Si l'autonomie est assez faible ou limitée, il faut se tourner vers des protocoles dont les débits de transferts sont faibles et dont la consommation est limitée.

Bluetooth Low Energy (BLE)

La technologie Bluetooth Low Energy est une nouvelle implémentation qui n'est pas directement compatible avec un dispositif possédant la norme Bluetooth classique. Elle a été conçue pour des besoins énergétiques faibles et un taux d'échange de données moins fréquent. Un dispositif BLE communique avec un débit correspondant à 1 Mbit/s pour une consommation d'énergie 10 fois inférieur par rapport à un dispositif Bluetooth classique (<15 mA). La portée du BLE est de 50 m.

Le Bluetooth Low Energy cherche donc à s'adresser à des appareils à faible puissance de calcul et dont l'autonomie est une contrainte majeure.

ZigBee

ZigBee est un protocole de communication sans-fil à courte portée et à faible consommation énergétique. Basé sur la norme 802.15.4 qui est un protocole de communication. ZigBee utilise les fréquences 868 Mhz, 915 Mhz et 2,4 Ghz pour établir une connexion. Son débit varie en fonction de la fréquence utilisée (20 kbit/s pour la fréquence 868 Mhz et 250 kbits/s pour la fréquence en 2,4 Ghz).

La portée est assez courte, jusqu'à 100 mètres, mais il peut parcourir de longues distances en passant à travers un réseau maillé avec d'autres appareils ZigBee.

En ce qui concerne l'implémentation réseau, on retrouve 3 topologies différentes (maillée, en étoile et en arbre).

Dans une architecture réseau qui utilise cette technologie, on retrouve 3 modules :

Zigbee Coordinator (ZC) : c'est le module qui permet d'initialiser le réseau, une fois le réseau initialisé, il se comporte comme un routeur.

ZigBee Routeur (ZR) : C'est un module optionnel, cependant s'il est présent il participe au routage des messages.

Zigbee End Device (ZED) : Dispositif qui a pour fonction de communiquer avec son coordinateur.
Il est facultatif et ne peut pas participer au routage des messages.

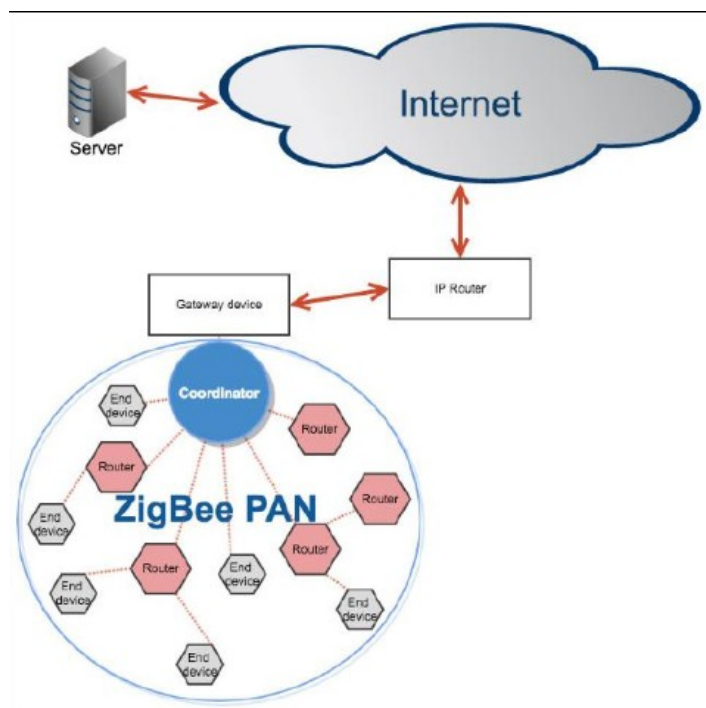


Figure 4: Exemple d'architecture réseau ZigBee

Si l'autonomie n'est pas un problème, on peut donc se tourner vers des protocoles plus énergivores dont le débit de transferts sera plus important.

Wi-Fi

Cette technologie est un ensemble de protocoles de communication sans fil . Un réseau Wi-Fi permet de relier par ondes radio plusieurs appareils informatiques (ordinateur, routeur, smartphone, etc.) au sein d'un réseau informatique afin de permettre la transmission de données entre eux. Le Wi-Fi offre des débits de données très rapides, mais nécessite plus de puissance pour supporter les communications constantes en aller et retour. Les débits peuvent varier en fonction du protocole utilisé (a/b/g/n/ac/ax) on peut avoir des débits compris entre 11 et 1000 Mb/s. Cette technologie est aussi utilisée pour l'internet des objets, car elle permet d'établir une connexion vers « L'internet ».

Sa portée dépend de la norme utilisée :

Standard	Bande de fréquence	Débit	Portée
WiFi a (802.11a)	5 GHz	54 Mbit/s	10 m
WiFi B (802.11b)	2.4 GHz	11 Mbit/s	140 m
WiFi G (802.11g)	2.4 GHz	54 Mbit/s	140 m
WiFi N (802.11n)	2.4 GHz / 5 GHz	450 Mbit/s	250 m

Tableau 1: Tableau des débits et portées de la technologie Wi-Fi en fonction de la norme

Le schéma de fonctionnement est presque identique entre les différents protocoles, c'est-à-dire qu'il s'agit d'un terminal qui va se connecter sur un point d'accès et ce point d'accès est lui même connecté à une passerelle. Dans certains cas comme celui du wi-fi au niveau domestique en général, la « box » ou encore la « box internet » permet d'être à la fois un point d'accès et une passerelle. Cette particularité est due au fait que la box est un matériel qui intègre un routeur, un point d'accès, un switch et un modem. La box est donc une passerelle entre le LAN-réseau local privé et WAN réseau public-internet.

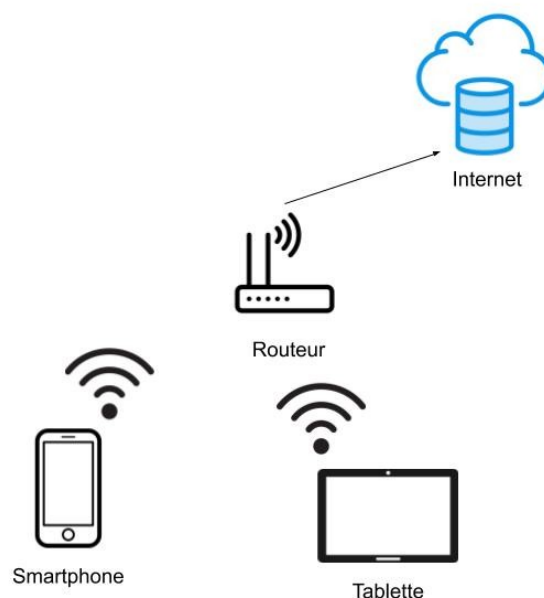


Figure 5: Schéma du principe d'une architecture Wi-fi au niveau domestique

4G (Réseau cellulaire)

Protocole de 4ème génération des standards de la téléphonie mobile. Ses débits varient en fonction de la catégorie du protocole (4,6,12), on a donc pour la catégorie LTE Cat 4 on a 150 Mbit/s, LTE-A Cat 6 pour 300 Mbit/s, LTE-A Cat 12 pour 600 Mbit/s. Dans ces variations, on observe deux types de 4G, nous avons la 4G et la 4G+ soit LTE-A(Advanced) qui propose des débit nettement supérieurs aux débits 4G classique.

Une catégorie a été conçu pour l'internet des objets, il s'agit de la catégorie 0, cette catégorie a un débit de données plus faibles, généralement plafonnés à 1 Mbit/s maximum.

3 Le lien avec la faune et la flore

Sur la période 2007 à 2018, en France métropolitaine, on dénombre en moyenne par an 4 040 feux qui ravagent approximativement 11 117 ha de forêt. La plupart des feux ont eu lieu dans les zones méditerranéennes pas moins de 6 698 ha ont déjà brûlés sur cette période. La plupart de ces incendies ont été provoqués à cause des conditions météorologiques particulières que l'on retrouve sur les côtes méditerranéenne. Les facteurs principaux sont les fortes températures, des vents forts et une sécheresse importante. Bien que les événements naturels surviennent, l'Homme a aussi sa part de responsabilité dans ces incendies, L'activité humaine est une des causes de déclenchement d'incendies dû à une activité économique (chantiers de BTP, activités agricoles...) ou bien d'une activité du quotidien (mégots de cigarettes, barbecues ou feux de camps). Ces incendies sont dû à des imprudences et à des comportements dangereux, aussi bien de touristes que de riverains.

Le problème étant que lorsque qu'un feu est déclaré par une personne, le feu est déjà présent et fait des ravages, c'est là que l'Internet des objets peut intervenir sur ce genre de cas. Des projets comme IoTrees et SmartForest ont vu le jour.

3.1 Les différents projets existants pour la préservation de la faune et de la flore

Le projet SmartForest vise à développer des applications pour les propriétaires de parcs forestiers, pour la surveillance en temps réel de leur propriété, grâce à un réseau de capteurs peu coûteux. Le but est d'anticiper les conditions environnementales propices à l'apparition d'incendies et les détecter dès le début. Une fois, les conditions propices à l'apparition d'un incendie, une alerte est envoyée afin de prévenir les gardes forestiers ou l'organisme qui s'occupe de la forêt pour qu'ils agissent de manière rapide. Les capteurs utilisés sont des capteurs de température, d'humidité et de mesure de niveaux de Co2 et Co.

L'architecture réseau d'un tel projet se compose de la manière suivante :

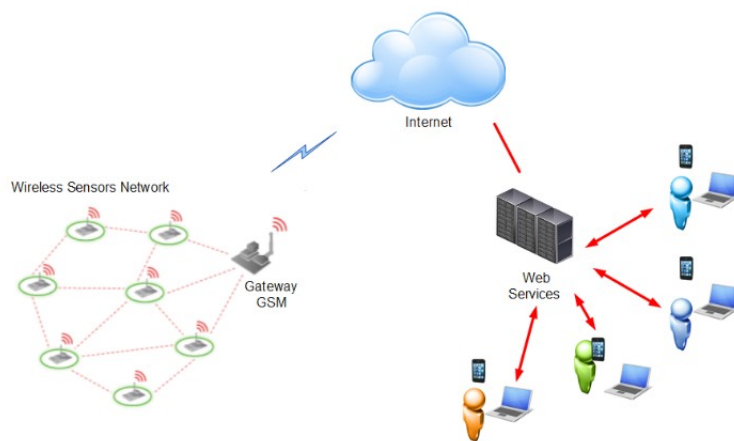


Figure 6: Architecture réseau du projet SmartForest

Les dispositifs sont connectés à une passerelle GSM, afin de faire transiter les informations vers les serveurs, comme il s'agit de données dont la taille est assez petite, un réseau GSM est suffisant. De plus la portée du signal peut varier entre 1 et 30 km. Le gain quant à lui doit être assez élevé car les arbres présents dans une forêt peuvent « interférer » avec le signal. Si le signal n'est pas assez « puissant » et que le dispositif est connecté dessus, mais qu'il n'arrive pas à garder une connexion stable alors on aura une surconsommation d'énergie dû à des tentatives répétées de connexion et de recherche de signal.

Une fois que les informations sont envoyées vers la passerelle, ces données sont envoyées vers les serveurs qui sont ensuite analysées et affiche l'alerte sur les dispositifs des utilisateurs.

En ce qui concerne l'alimentation des dispositifs, ils sont alimentés par batterie, mais aussi par panneau solaire afin de permettre une certaine autonomie sans que l'Homme ne soit dans l'obligation d'aller vérifier les niveaux des dispositifs.

Un projet semblable à SmartForest a vu le jour, il s'agit d'un projet qui est apparu dans le « International Journal of Current Trends in Engineering & Research (IJCTER) » dans le 3ème volume, numéro 05 datant de Mai 2017.

Ce projet est aussi sur la prévention d'incendies dans le milieu forestier, il utilise plusieurs capteurs à faible coût afin d'alerter l'utilisateur. Ces capteurs sont des capteurs de température LM35 ou encore le DHT11. On a aussi des photorésistances, des capteurs ultra son et des capteurs d'humidité, dont un mesure l'humidité de l'air et l'autre du sol.

BASIC SYSTEM ARCHITECTURE OF THE IoT ENABLED SYSTEM:-

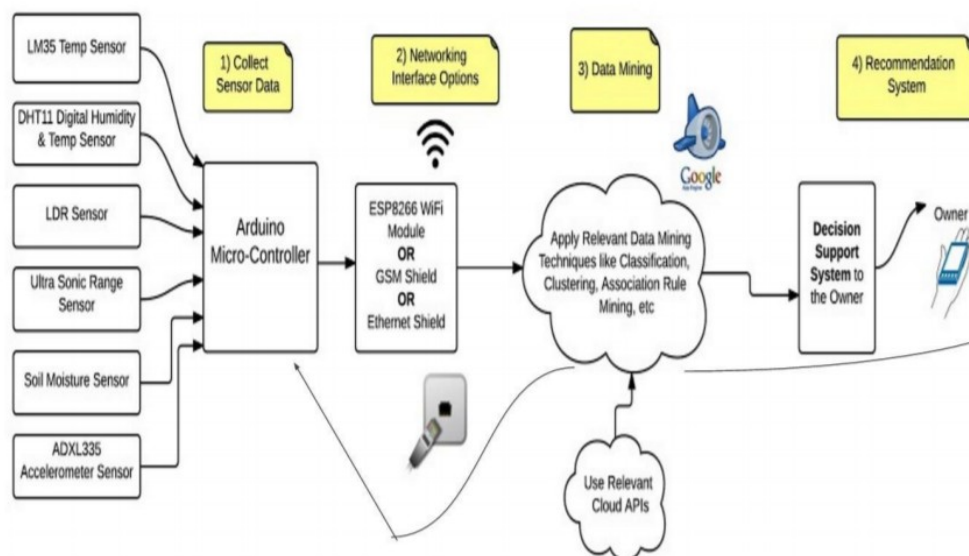


Figure 7: Schéma fonctionnement du projet paru dans IJCTER

Pour la détection et l'envoi d'alerte, ces dispositifs peuvent se baser sur des conditions particulières comme l'exemple ci-dessous qui est un tableau qui est apparu dans un document durant l'événement « The Eleventh International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies » UBICOMM 2017 et qui représente les différents facteurs d'alerte :

Variable	Alert: level 3	Alert: level 2	Alert: level 1
Temperature	$\geq 30^{\circ}\text{C}$	$\geq 37^{\circ}\text{C}$	$\geq 40^{\circ}\text{C}$
Humidity	$\leq 30\%$	$\leq 20\%$	$\leq 10\%$
CO2	$\geq 350\text{ ppm}$	$\geq 2000\text{ ppm}$	$\geq 5000\text{ ppm}$
CO	$\geq 10\text{ ppm}$	$\geq 25\text{ ppm}$	$\geq 50\text{ ppm}$

Tableau 2: Tableau des alertes

Pour donner plus d'informations ne serait-ce que pour le taux de Co2 un taux normal de Co2 est compris entre 380 et 480 ppm au-delà de 5 000 ppm cela devient dangereux pour l'Homme.

<u>Concentration</u>	<u>Effet sur l'homme - Seuil</u>
380 - 480 ppm	Taux normal de l'atmosphère
600 - 800 ppm	Taux correct en lieux fermés
1000 - 1100 ppm	Taux tolérable en lieux fermés
5000 ppm	Limite haute pour 8h d'exposition
6000 - 30000 ppm	Exposition très courte
3 à 8 %	Augmentation fréquence respiratoire et cardiaque
Au-delà de 10 %	Nausée, vomissement, évanouissement
Au-delà de 20 %	Evanouissement rapide, décès

Tableau 3: Niveau de Co2 et impact sur l'Homme (th-industrie)

D'autres projets comme le projet IoTrees vise à développer une surveillance en utilisant dendrométrie, pour mesurer le diamètre des arbres et envoyer les informations de mesure de manière transparente à la plateforme qui gère le projet (ForestHQ) où l'utilisateur peut interagir avec les données forestières pour estimer la croissance de la forêt et surveiller la santé des cultures. Cette solution a pour but de réduire le coût de la collecte de données et permettre d'effectuer des mesures forestières plus fréquentes, améliorant ainsi la surveillance durable des ressources forestières. Les nouvelles informations reçues permettront de prendre des décisions plus efficaces sur la gestion de la forêt. Les décisions seront plus opportunes et plus faciles à prendre, par exemple en ce qui concerne le moment de la récolte et celui de l'application de la lutte contre la végétation ou les parasites. Les futures prévisions de la croissance des forêts seront plus précises et plus fiables grâce à des mesures plus fréquentes.

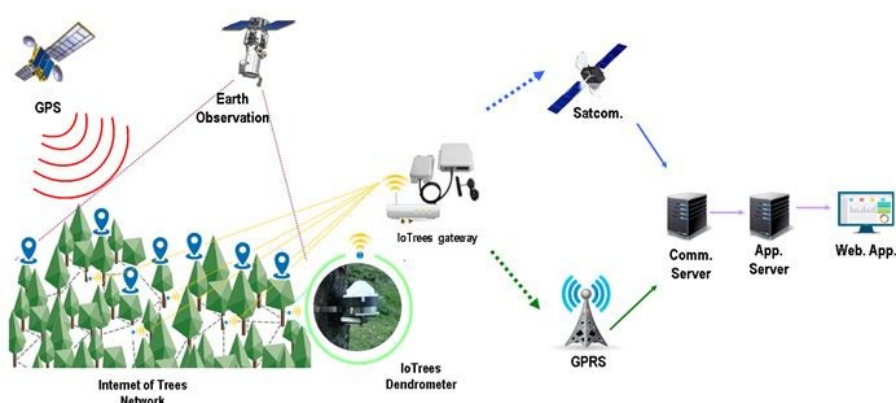


Figure 8: Fonctionnement de la solution (IoTree)

Le projet Rainforest est un projet qui se situe cette fois-ci dans la forêt tropical. Ce projet a pour but de protéger la forêt de déforestations illégales, au travers d'une analyse de son. Les ingénieurs qui gèrent ce projet réutilisent des téléphones mobiles recyclés, ces systèmes sont rendus étanches et possèdent une autonomie importante. Ces dispositifs sont ensuite placés en forêt. Ces dispositifs

envoient en temps réel le son émis par la forêt. Les différentes fréquences sonores reçu sont ensuite analysées et les alertes sont envoyées directement auprès des organismes de surveillance des forêts et de la lutte contre la déforestation criminelle.

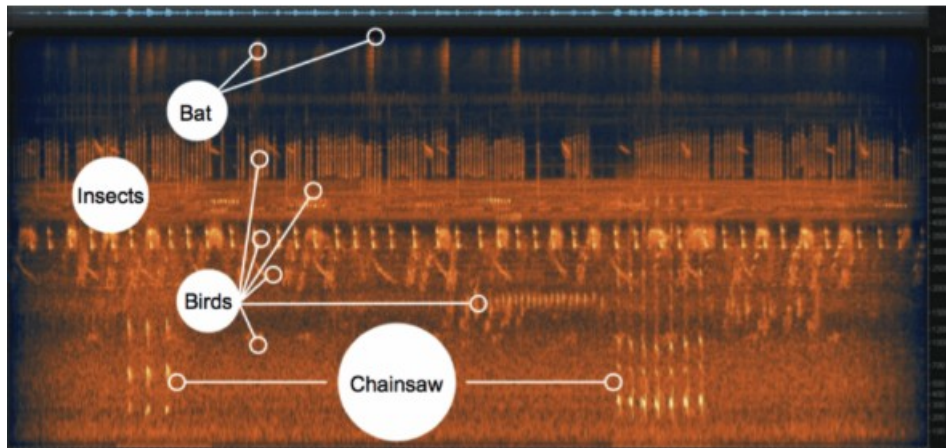


Figure 9: Exemple de l'analyse des sons

Chaque son est analysé et permet de déterminer s'il s'agit ou non d'un animal.

S'il s'agit du bruit d'une tronçonneuse, une alerte est automatiquement envoyée vers les équipes d'intervention afin de stopper l'activité en cours, bien sûr quand un appareil envoi une alerte il envoie ses informations (nom, position,...) afin que l'intervention soit rapide.

Le dispositif permettant de recueillir les sons et de les envoyés vers le cloud de rainforest ressemble à cela :



Figure 10: Dispositif Rainforest

Ce dispositif est alimenté par des panneaux solaires accrochés tout autour du téléphone, ils ont disposé plusieurs bandes de panneaux solaires afin de permettre à l'appareil d'être rechargé tout au long de la journée grâce à une disposition permettant de capter les rayons émis par le soleil quel que

soit le moment de la journée. Le dispositif utilise sa batterie la nuit et est rechargé la journée, cela permet une certaine autonomie sans que l'homme ait besoin d'intervenir de manière régulière.

Le schéma ci-dessous montre le fonctionnement du projet :

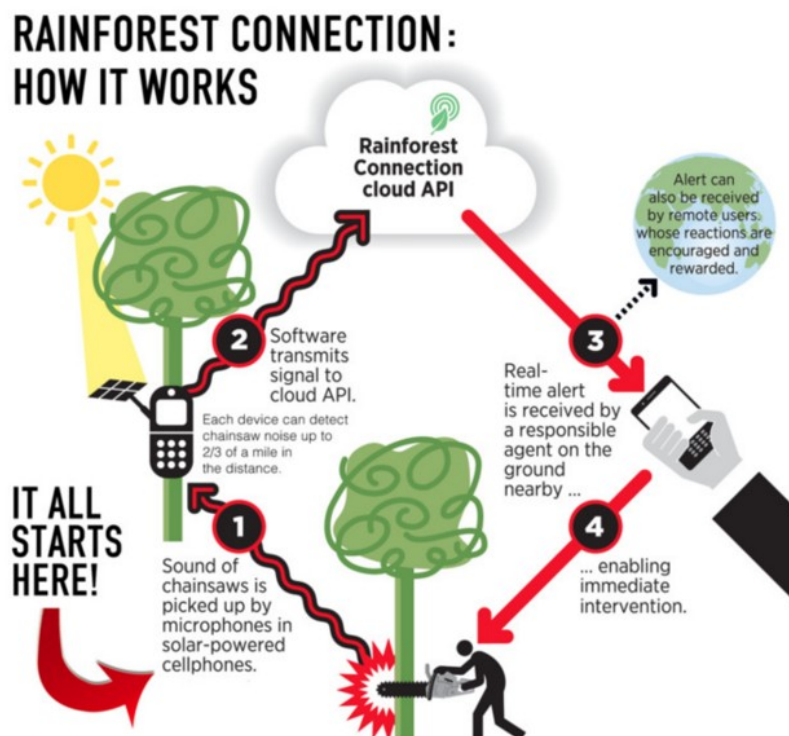


Figure 11: Fonctionnement de la solution chez Rainforest

De plus, ce projet, en plus de gérer la flore, s'occupe aussi de braconnage d'animaux, le but est de faire en sorte que les braconniers ne puissent plus régner librement sur les forêts tropicales humides du monde. Les capteurs qu'ils ont développés permettant d'aider leurs partenaires à reconnaître les schémas d'activité liés au braconnage, notamment les alertes concernant les camions, les voitures et les motos utilisées par les braconniers dans les principales zones protégées. Pour reprendre un de leurs exemples, en Afrique, ils ont pu démontrer, que la protection et la surveillance d'une route clé utilisée par les braconniers pouvaient permettre de protéger une grande partie de la forêt tropicale.

Leurs dispositifs ont permis aux équipes d'intervention d'agir à des moments et des jours clé de chaque mois où les activités de braconnage étaient statistiquement très élevées.

Bien que la flore soit un domaine où l'Internet des Objets peut se développer, il faut aussi prendre en compte que la faune est aussi touchée par cette avancée technologique. On retrouve des objets connectés notamment dans des colliers avec traqueur GPS intégré à l'intérieur du collier. Cela permet aux soigneurs et aux chercheurs de suivre des animaux sans avoir besoin d'intervenir. Cela permet d'évaluer leurs lieux de vie et leurs habitudes, et ainsi intervenir au cas où des braconniers voudraient agir, et donc perturber les habitudes des animaux et l'envoi des données effectué par le collier.

Généralement ce genre de dispositifs est utilisé dans les pays où le taux de braconnage est élevé, pour donner un exemple, au Kenya, la police a installé des micropuces dans les cornes des rhinocéros pour suivre leurs mouvements et ainsi augmenter les chances de poursuivre les braconniers. Au Zimbabwe, le projet Rapid (Real-Time Anti-Poaching Intelligence Device) de l'International Humane Society est déployé sur les rhinocéros avec des capteurs de fréquence cardiaque, des trackers GPS ainsi que des caméras vidéo. En cas de rencontre avec un braconnier, le rythme cardiaque du rhinocéros augmente en raison du stress. Celui-ci déclenche une alarme et transmet ses coordonnées GPS à un centre de contrôle. Le centre de contrôle active alors la caméra du rhinocéros afin d'obtenir une vérification visuelle de ce qui se passe. Ils envoient ensuite un hélicoptère qui arrive sur place avant que les braconniers puissent s'échapper ou récolter les cornes.

Chapitre 4 : Partie Tech.

Dans ce mémoire, nous allons pas seulement parler des projets existants, nous allons aussi tenter de concevoir une solution possible même s'il s'agit d'une solution théorique.

Le projet autour de cette solution serait de combiner les technologies utilisées pour la supervision de la faune et de la flore et de les faire communiquer entre elles.

Nous allons donc dans une première partie essayer de faire communiquer un dispositif et un serveur dans lequel le dispositif transmettra des informations vers le serveur qui lui affichera les informations.

Simulation d'une situation de monitoring dans une forêt

Dans cette partie, nous essaierons de faire une maquette d'une simulation sur une situation de monitoring pour une forêt, pour ce faire nous utiliserons un ordinateur portable qui servira de dispositif pour la surveillance, nous utiliserons un autre ordinateur avec une clé 4G qui sera connecté à Internet et servira de point d'accès pour le dispositif.

Nous ferons aussi la configuration d'un point d'accès et d'une passerelle qui permettra la redirection des informations vers le réseau domestique où se trouve le serveur.

Le schéma de la simulation sera présenté de la façon suivante :

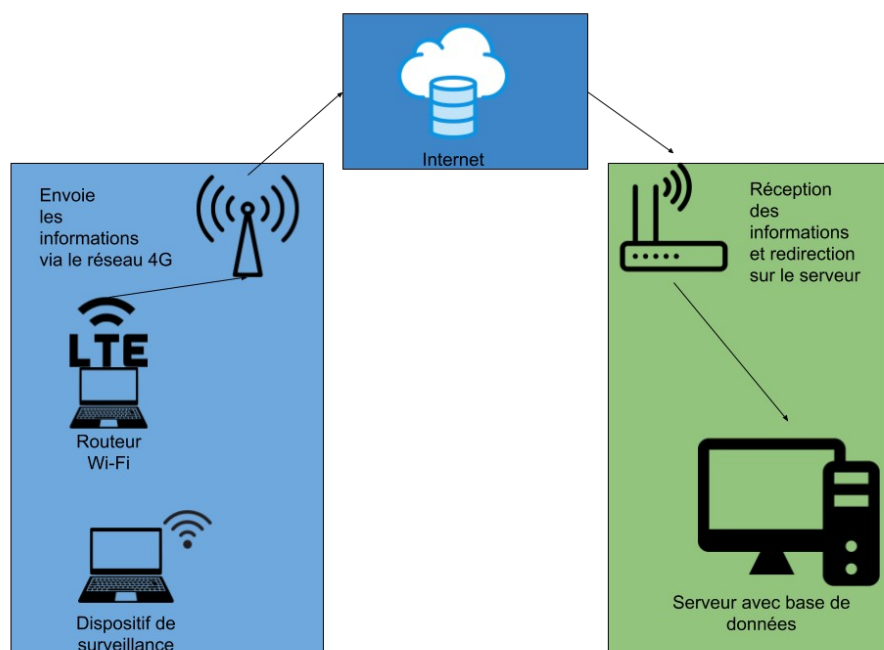


Figure 12: Schéma de la simulation

L'ordinateur qui servira de dispositif de surveillance, enverra des informations « essentiels » via MQTT, il enverra sa température, son niveau de charge pour la batterie, son signal wi-fi (en pourcentage) et son niveau de puissance (en dBm). Pour la transmission des informations on mettra en place un environnement MQTT, pour cela il nous faut mettre en place différents services.

Pour cela on utilisera les programmes suivants :

- Broker MQTT : Mosquitto
- Base de données (InfluxDB)
- Visualiser les données MQTT sur serveur web : Grafana

Configuration de l'environnement

1.1 Le broker MQTT

Pour le Broker qui est la base de l'environnement pour l'envoi des données, on utilisera Mosquitto, celui-ci sera installé sur une VM qui servira de serveur ainsi que notre base de donnée et le serveur web.

Installation de Mosquitto :

Pour installer le paquet on passe par la commande apt , une fois installé on se déplace dans le dossier / etc /mosquitto , pour aller chercher le fichier de configuration mosquitto.conf, afin de configurer certains paramètres :

```
#Partie configurée
log_type debug
#on affichera les logs
allow_anonymous false
#on refuse les connexions anonymes
password_file /etc/mosquitto/user.txt
#on crée un fichier qui contiendra des utilisateurs et mots de passes afin de permettre la connexion
au broker
```

Pour « sécuriser » le broker, on désactive les connexions anonymes, on fait en sorte de pouvoir récupérer les logs en cas de problèmes, puis on lui indique que l'on souhaite avoir une liste d'utilisateurs qui pourront se connecter au broker seulement s'ils sont dans ce fichier texte.

Ensuite on utilise la commande `mosquitto_passwd` afin de prendre en compte le fichier `user.txt` et on lui attribue un utilisateur ici celui-ci sera `buddy` :

```
mosquitto_passwd -c /etc/mosquitto/user.txt buddy
Password:
Reenter password:
```

On définit un utilisateur afin de permettre l'envoi des données.

1.2 La base de donnée InfluxDB

Pour sauvegarder les messages envoyés sur le broker et donc garder un historique, on configure une base de donnée. Elle recevra les messages via Telegraf, qui sera considéré comme client mqtt sur le topic souhaité. Pour cela on utilisera InfluxDB.

Pour installer influxDB, on procède de la manière suivante :

```
curl -sL https://repos.influxdata.com/influxdb.key | sudo apt-key add -
echo "deb https://repos.influxdata.com/ubuntu stretch stable" | sudo tee
/etc/apt/sources.list.d/influxdb.list
apt update
apt install influxdb
E: Sub-process /usr/bin/dpkg returned an error code (1)
systemctl unmask influxdb.service
Removed /etc/systemd/system/influxdb.service.
service influxdb start
influx
```

Afin de configurer notre base de données nous devons modifier son fichier de configuration qui se trouve dans le répertoire `/etc/influxDB`.

On modifie le fichier .conf et on recherche la partie qui concerne la connexion via HTTP :

```
[http]
# Determines whether HTTP endpoint is enabled.
enabled = true

# The bind address used by the HTTP service.
bind-address = ":8086"

# Determines whether user authentication is enabled over HTTP/HTTPS.
auth-enabled = true

# Determines whether detailed write logging is enabled.
write-tracing = false
# Determines whether HTTP request logging is enabled.
log-enabled = true
# Determines whether the pprof endpoint is enabled. This endpoint is used for
# troubleshooting and monitoring.
pprof-enabled = true

# Enables a pprof endpoint that binds to localhost:6060 immediately on startup.
# This is only needed to debug startup issues.
debug-pprof-enabled = false

# Determines whether HTTPS is enabled.
https-enabled = true

# The SSL certificate to use when HTTPS is enabled.
https-certificate = "/etc/ssl/influxdb.pem"
```

Le but de cette manœuvre est d'activer le support HTTP, sur le port 8086, avec l'authentification pour que la base de données ne reçoive que les messages d'utilisateurs enregistrés.

Pour finir la configuration, il nous faut définir un utilisateur avec les privilèges, sur la base de données. On lance influx pour afficher le CLI afin de créer un utilisateur admin :

```
influx
Connected to http://localhost:8086 version 1.7.9
InfluxDB shell version: 1.7.9
> CREATE USER admin WITH PASSWORD 'influx' WITH ALL PRIVILEGES
```

L'utilisateur admin a été créé avec tous les privilèges.

Une fois l'utilisateur créé, il nous faut créer notre base de données, pour ce faire on crée la base de donnée iot :

```
CREATE DATABASE iot
```

On vérifie les valeurs dans la base de donnée iot :

```
> use iot
Using database iot
> SHOW MEASUREMENTS
name: measurements
name
----
cpu
disk
diskio
kernel
mem
processes
swap
system
```

De ce côté la base de données est prête.

1.3 Télégraf

Pour faire le lien entre la base de donnée et le broker on va utiliser Telegraf qui va traduire les informations pour la base de données.

On récupère la clé pour le dépôt, ainsi que le dépôt :

```
wget -qO- https://repos.influxdata.com/influxdb.key | sudo apt-key add -
source /etc/os-release
test $VERSION_ID = "10" && echo "deb https://repos.influxdata.com/debian buster stable" | sudo
tee /etc/apt/sources.list.d/influxdb.list
apt update
apt install telegraf
```

On modifie telegraf.conf pour prendre en compte le serveur MQTT.

```
#####  
#                               #  
#####  
[[outputs.influxdb]]  
  urls = ["http://127.0.0.1:8086"]  
  database = "iot"  
## HTTP Basic Auth  
  username = "admin"  
  password = "influx"
```

```
[[inputs.mqtt_consumer]]  
servers = ["tcp://127.0.0.1:1883"]  
  topics = ["iot/msg"]  
  username = "buddy"  
  password = "test"  
  data_format = "influx"
```

On lance le service et on effectue des envoies via mqtt sur un pc distant ou en local :

```
mosquitto_pub -h 192.168.1.40 -t iot/msg -m "iot,host=buddy value=42" -u buddy -P test
```

Cette requête va se connecter au serveur pour publier dans le topics iot/msg dont le contenu du message est : la table iot, la colonne host= buddy, colonne value=42

Après plusieurs tests, sur la base de données et via mqtt on obtient le résultat suivant via la commande SHOW MEASUREMENTS dans influxDB :

```
> SHOW MEASUREMENTS  
name: measurements  
name  
----  
cpu  
disk  
diskio  
iot  
kernel  
mem  
message  
processes  
swap  
system
```

On a bien la table iot qui a été créée à la réception du message.

On peut vérifier la table avec la requête SELECT :

```
> SELECT * FROM iot
name: iot
time          host  topic  value
----          -
1575117850893600318 buddy iot/msg 42
1575118597839564563 buddy iot/msg 69
```

On a bien le nom associé host avec le topic ainsi que la valeur qui avait été envoyée.

1.4 Grafana

Afin de visualiser les données transmises à la base de données, nous devons installer un service capable de lire ces données et d'afficher un rendu clair. Grafana permettant de faire cela et étant facile à configurer et à utiliser, c'est celui-ci que nous allons installer.

Pour installer Grafana plusieurs étapes sont nécessaires tout d'abord il faut ajouter le dépôt sur la machine :

```
sudo add-apt-repository "deb https://packages.grafana.com/oss/deb
stable main"
```

Puis on récupère la clé pour installer grafana :

```
wget -q -O - https://packages.grafana.com/gpg.key | sudo apt-key
add
```

On met à jour les dépôts et on installe Grafana.

On lance le serveur avec les fichiers de configuration par défaut et on vérifie son statut :

```
service grafana-server start
service grafana-server status
• grafana-server.service - Grafana instance
   Loaded: loaded (/usr/lib/systemd/system/grafana-server.service; disabled; vendor preset:
   enabled)
   Active: active (running) since Thu 2020-06-25 10:52:04 UTC; 1s ago
```

On se connecte sur la page web de Grafana avec l'utilisateur et le mot de passe par défaut soit « admin/admin », ensuite on modifie le mot de passe admin.

On ajoute une nouvelle base de données source et on lui attribue une base InfluxDB et on lui donne les paramètres suivants :

The image shows the Grafana configuration interface for a new data source named 'InfluxDB'. The 'Name' field is set to 'InfluxDB' and the 'Default' toggle is turned on. The 'HTTP' section shows the 'URL' as 'http://localhost:8086', 'Access' as 'Server (default)', and a 'Whitelisted Cookies' section with an 'Add' button. The 'Auth' section has several options: 'Basic auth' (disabled), 'With Credentials' (disabled), 'TLS Client Auth' (disabled), 'With CA Cert' (disabled), 'Skip TLS Verify' (disabled), and 'Forward OAuth Identity' (disabled). The 'Custom HTTP Headers' section has an 'Add header' button. The 'InfluxDB Details' section shows 'Database' as 'iot', 'User' as 'admin', 'Password' as 'configured' (with a 'Reset' button), and 'HTTP Method' as 'GET'.

Field	Value
Name	InfluxDB
Default	On
HTTP	
URL	http://localhost:8086
Access	Server (default)
Whitelisted Cookies	Add Name
Auth	
Basic auth	Off
With Credentials	Off
TLS Client Auth	Off
With CA Cert	Off
Skip TLS Verify	Off
Forward OAuth Identity	Off
Custom HTTP Headers	
+ Add header	
InfluxDB Details	
Database	iot
User	admin
Password	configured
HTTP Method	GET

Figure 13: Paramètres BDD

Une fois configuré, il faut créer un dashboard, on utilisera le dashboard de base. Pour la partie Query (Requêtes) qui va récupérer les informations de la base de donnée, on l'écrit de la façon suivante :

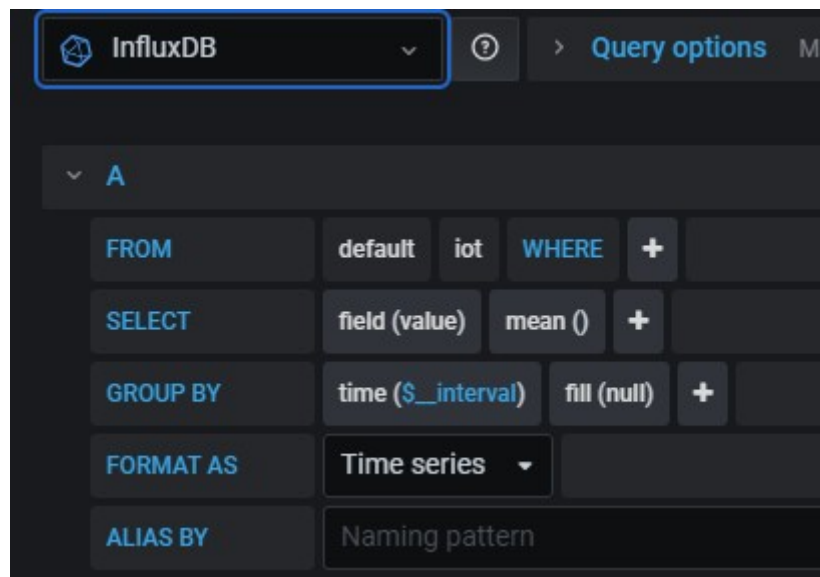


Figure 14: Requêtes

2. Redirection des ports :

Pour cette partie on va configurer la redirection des ports afin que les informations soient redirigées vers le serveur. On attribue au serveur une adresse IP fixe :

```
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 08:00:27:fc:36:1f brd ff:ff:ff:ff:ff:ff
    inet 192.168.0.48/24 brd 192.168.0.255 scope global dynamic eth0
        valid_lft forever preferred_lft 16486 sec
```

Figure 15: Adresse serveur

Une fois l'adresse attribué, il faut procéder à la redirection, sur la « box », il faut se rendre dans la section gestion des ports et affecter un port :

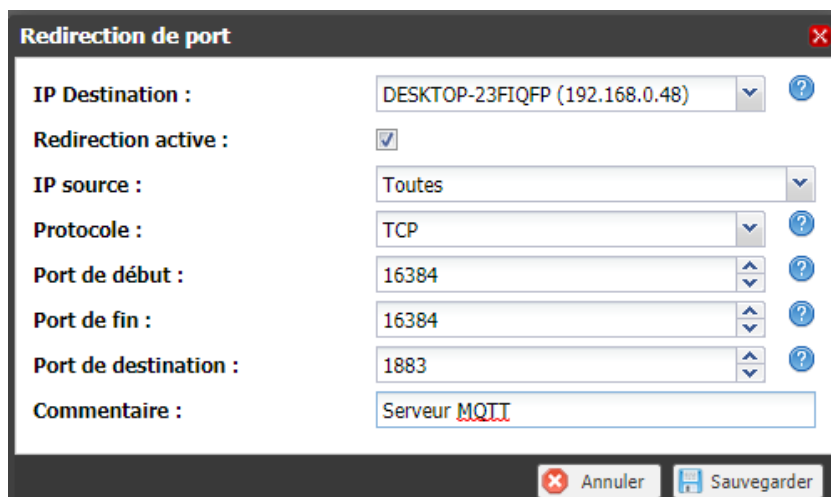


Figure 16: Redirection sur un port spécifique

En ce qui concerne la redirection des ports, il n'y a rien de plus à faire.

3. Test MQTT

Pour cette partie, nous allons faire des test d'envois de messages MQTT :

Dans un premier temps, on connecte le dispositif au point d'accès qui se nomme AP 4G :

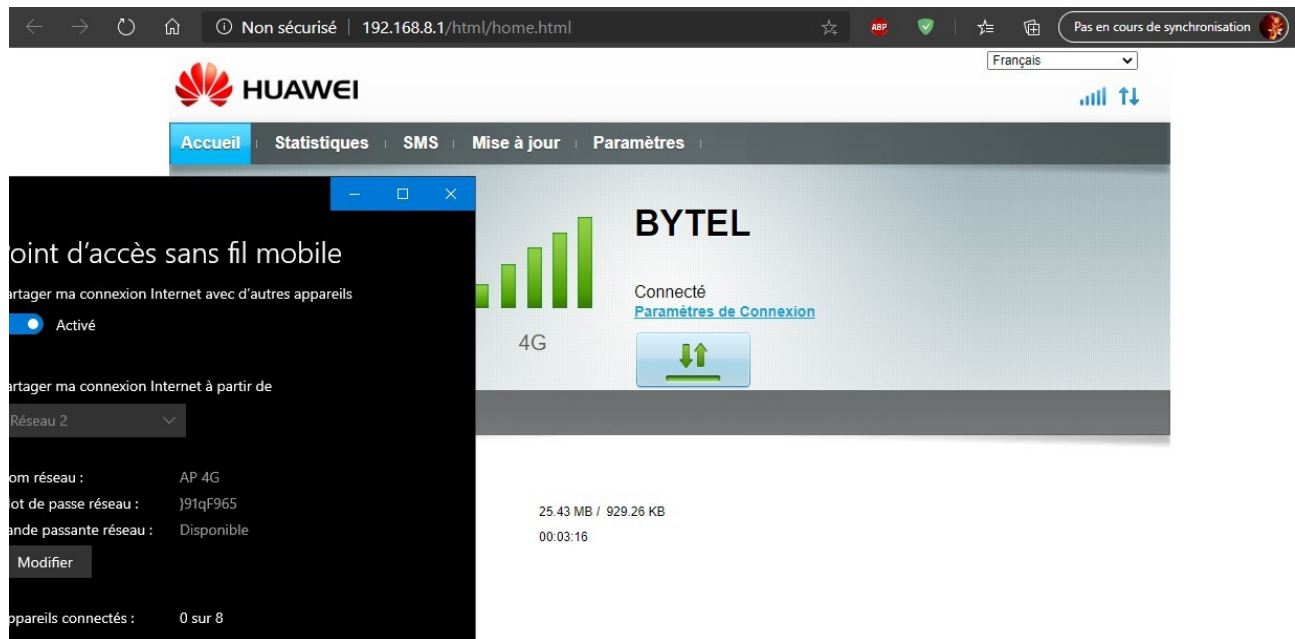


Figure 17: Interfaces clé 4G et point d'accès

Nom réseau :AP 4G

Mot de passe réseau :}91qF965

Bande passante réseau :Disponible

Modifier

Appareils connectés :1 sur 8

Nom de l'appareil	Adresse IP	Adresse physique (MAC)
buddy-G5-5590	192.168.137.9	a4:c3:f0:36:eb:c4

Figure 18: Paramètres du point d'accès

Sur le dispositif on vérifie son adresse IP :

```
3: wlo1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group
default qlen 1000
link/ether a4:c3:f0:36:eb:c4 brd ff:ff:ff:ff:ff:ff
inet 192.168.137.9/24 brd 192.168.137.255 scope global dynamic noprefixroute wlo1
    valid_lft 604728sec preferred_lft 604728sec
inet6 fe80::4a71:7319:57f9:ebbd/64 scope link noprefixroute
    valid_lft forever preferred_lft forever
```

la commande iwconfig nous donne des information sur la carte réseau sans-fil et notamment le nom du point d'accès auquel le dispositif s'est connecté :

```
wlo1 IEEE 802.11 ESSID:"AP 4G"
    Mode:Managed Frequency:2.462 GHz Access Point: 86:C6:3B:B4:FF:61
    Bit Rate=72.2 Mb/s Tx-Power=22 dBm
    Retry short limit:7 RTS thr:off Fragment thr:off
    Power Management:on
    Link Quality=70/70 Signal level=-33 dBm
    Rx invalid nwid:0 Rx invalid crypt:0 Rx invalid frag:0
    Tx excessive retries:21 Invalid misc:1 Missed beacon:0
```

Pour « prouver » que l'on se trouve sur une connexion différente du serveur, on effectue un curl d'un site web qui nous donne notre adresse IP publique :

```
curl ifconfig.me/ip
176.149.89.77
```

Une fois les étapes de connexion effectuées, le plus simple est d'envoyer un message pour vérifier la connexion, on effectue la requête :

```
mosquitto_pub -h 88.127.243.12 -p 16384 -t iot/msg -m "iot,host=buddy value=54" -u buddy -P
test
```

On définit l'adresse publique du réseau où se trouve le serveur ainsi que le port à utiliser.

Et on vérifie si le serveur a reçu le message :

```
mosquitto_sub -h localhost -t "iot/msg" -u buddy -P test
```

```
iot,host=buddy value=54
```

Deuxième test :

Cette fois-ci on va élaborer un script qui va envoyer la puissance en dBm du signal wifi auquel on est connecté ainsi que sa puissance en pourcentage, on rajoute aussi la température de l'ordinateur ainsi que son niveau de batterie :

Le script :

```
#!/bin/bash

wifi=$(grep "^s*w" /proc/net/wireless | awk '{print $4}');
wifip=$(grep "^s*w" /proc/net/wireless | awk '{ print int($3 * 100 / 70)}');
battery=$(cat /sys/class/power_supply/BAT0/capacity);
temp=$(sensors | awk '/Core 0/ {print $3}' | sed s/+// | sed s/°// | sed s/C//);

while true ;

do mosquitto_pub -h 88.127.243.12 -p 16384 -t iot/msg -m "wifi2,host=buddy puissanced=$wifi" -u
buddy -P test ;
mosquitto_pub -h 88.127.243.12 -p 16384 -t iot/msg -m "wifi,host=buddy wifi=$wifip" -u buddy -P
test ;
mosquitto_pub -h 88.127.243.12 -p 16384 -t iot/msg -m "bat,host=buddy bat=$battery" -u buddy -
P test ;
mosquitto_pub -h 88.127.243.12 -p 16384 -t iot/msg -m "temp,host=buddy cpu=$temp" -u buddy -
P test ;
done;
```

Chacune des commande ci-dessus va créer une table dans la base de données, afin de différencier les données reçues. Chaque dispositifs aura un nom différents en fonction du secteur et du numéro attribué (ex : SecA42), ce nom sera mis dans host et on pourra trier les informations par la suite.

On test le script, du côté serveur on réceptionne les messages :

```
wifi2,host=buddy puissanced=-47.  
wifi,host=buddy wifi=90  
bat,host=buddy bat=100  
temp,host=buddy cpu=34.0
```

Une fois la réception des messages effectuée, on se dirige vers la base de données afin de vérifier si les messages ont bien été transmis vers celle-ci.

Sur la table puissance (en dBm):

```
> select * from wifi2  
name: wifi2  
time          host puissanced topic  
----          -  
1592571740056193738 buddy -47      iot/msg  
1592571741826097527 buddy -47      iot/msg  
1592571742636649785 buddy -47      iot/msg  
1592571743456343816 buddy -47      iot/msg  
1592571744051337947 buddy -47      iot/msg  
1592571744556447960 buddy -47      iot/msg  
1592571745037457452 buddy -47      iot/msg
```

Sur la table en pourcentage :

```
> select * from wifi  
name: wifi  
time          host topic  wifi  
----          -  
1592571750117268791 buddy iot/msg 90  
1592571750611939548 buddy iot/msg 90  
1592571751097954737 buddy iot/msg 90  
1592571751616812847 buddy iot/msg 90  
1592571752116633057 buddy iot/msg 90  
1592571752591762584 buddy iot/msg 90
```

Sur la table batterie :

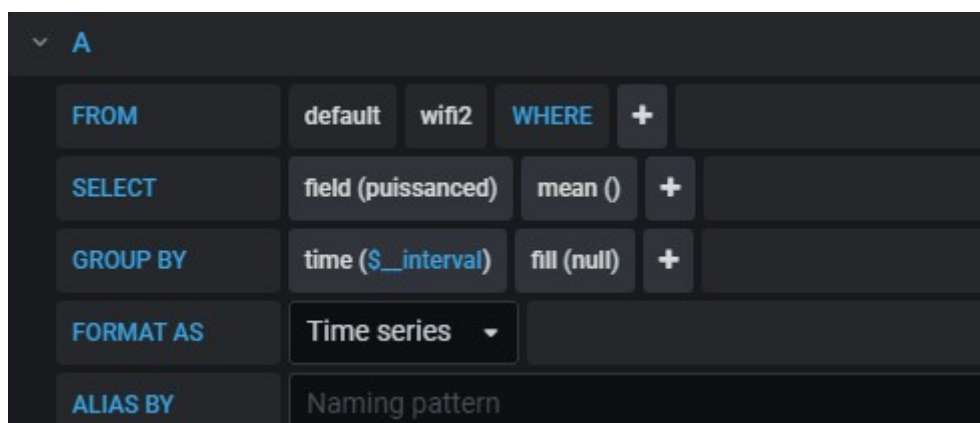
```
> select * from bat
name: bat
time          bat host topic
----          -
1592571750247857057 100 buddy iot/msg
1592571750731875377 100 buddy iot/msg
1592571751226556557 100 buddy iot/msg
1592571751746453906 100 buddy iot/msg
1592571752231690126 100 buddy iot/msg
1592571752722012747 100 buddy iot/msg
```

Sur la table température :

```
> select * from temp
name: temp
time          cpu host topic
----          -
1592571750366771664 34  buddy iot/msg
1592571750856544326 34  buddy iot/msg
1592571751356444821 34  buddy iot/msg
1592571751877276505 34  buddy iot/msg
1592571752351799088 34  buddy iot/msg
1592571752836391624 34  buddy iot/msg
```

On définit ensuite les graphiques sur grafana :

Requête pour la puissance en dBm :



The screenshot shows the Grafana query editor interface. The query is defined as follows:

FROM	default	wifi2	WHERE	+	
SELECT	field (puissanced)	mean ()	+		
GROUP BY	time (\$__interval)	fill (null)	+		
FORMAT AS	Time series				
ALIAS BY	Naming pattern				

Figure 19: Requête puissance en dBm

Requête signal en % :

The screenshot shows a query builder interface with a dark theme. At the top, there is a dropdown menu with a downward arrow and the letter 'A'. Below this, there are five rows of controls, each with a label on the left and a set of buttons on the right. The first row is labeled 'FROM' and contains buttons for 'default', 'wifi', 'WHERE', and a plus sign. The second row is labeled 'SELECT' and contains buttons for 'field (wifi)', 'last ()', and a plus sign. The third row is labeled 'GROUP BY' and contains buttons for 'time (\$__interval)' and 'fill (null)', followed by a plus sign. The fourth row is labeled 'FORMAT AS' and contains a dropdown menu with 'Time series' selected. The fifth row is labeled 'ALIAS BY' and contains a text input field with the placeholder 'Naming pattern'.

Figure 20: Requête signal Wi-Fi

Requête niveau de batterie :

The screenshot shows a query builder interface with a dark theme. At the top, there is a dropdown menu with a downward arrow and the letter 'A'. Below this, there are five rows of controls, each with a label on the left and a set of buttons on the right. The first row is labeled 'FROM' and contains buttons for 'default', 'bat', 'WHERE', and a plus sign. The second row is labeled 'SELECT' and contains buttons for 'field (bat)', 'last ()', and a plus sign. The third row is labeled 'GROUP BY' and contains buttons for 'time (\$__interval)' and 'fill (null)', followed by a plus sign. The fourth row is labeled 'FORMAT AS' and contains a dropdown menu with 'Time series' selected. The fifth row is labeled 'ALIAS BY' and contains a text input field with the placeholder 'Naming pattern'.

Figure 21: Requête niveau de batterie

Requête température :

The screenshot shows a query builder interface with a dark theme. At the top, there is a dropdown menu with a downward arrow and the letter 'A'. Below this, there are five rows of controls, each with a label on the left and a set of buttons on the right. The first row is labeled 'FROM' and contains buttons for 'default', 'temp', 'WHERE', and a plus sign. The second row is labeled 'SELECT' and contains buttons for 'field (cpu)', 'last ()', and a plus sign. The third row is labeled 'GROUP BY' and contains buttons for 'time (\$__interval)' and 'fill (null)', followed by a plus sign. The fourth row is labeled 'FORMAT AS' and contains a dropdown menu with 'Time series' selected. The fifth row is labeled 'ALIAS BY' and contains a text input field with the placeholder 'Naming pattern'.

Figure 22: Requête niveau de température du dispositif

On peut aussi obtenir une vue globale du dispositif :



Figure 23: Vue globale des différents composants

Alertes :

En utilisant grafana nous avons la possibilité de créer des alertes en cas de problèmes, ce qui peut être utile en cas de baisse critique de la batterie ou encore un niveau élevé pour la température.

Pour donner un exemple on va configurer une alerte pour le niveau de température, Grafana a ajouté récemment le support de notifications sur Discord, pour faire fonctionner les notifications, nous devons configurer un paramètre nommé WebHook sur un serveur discord.

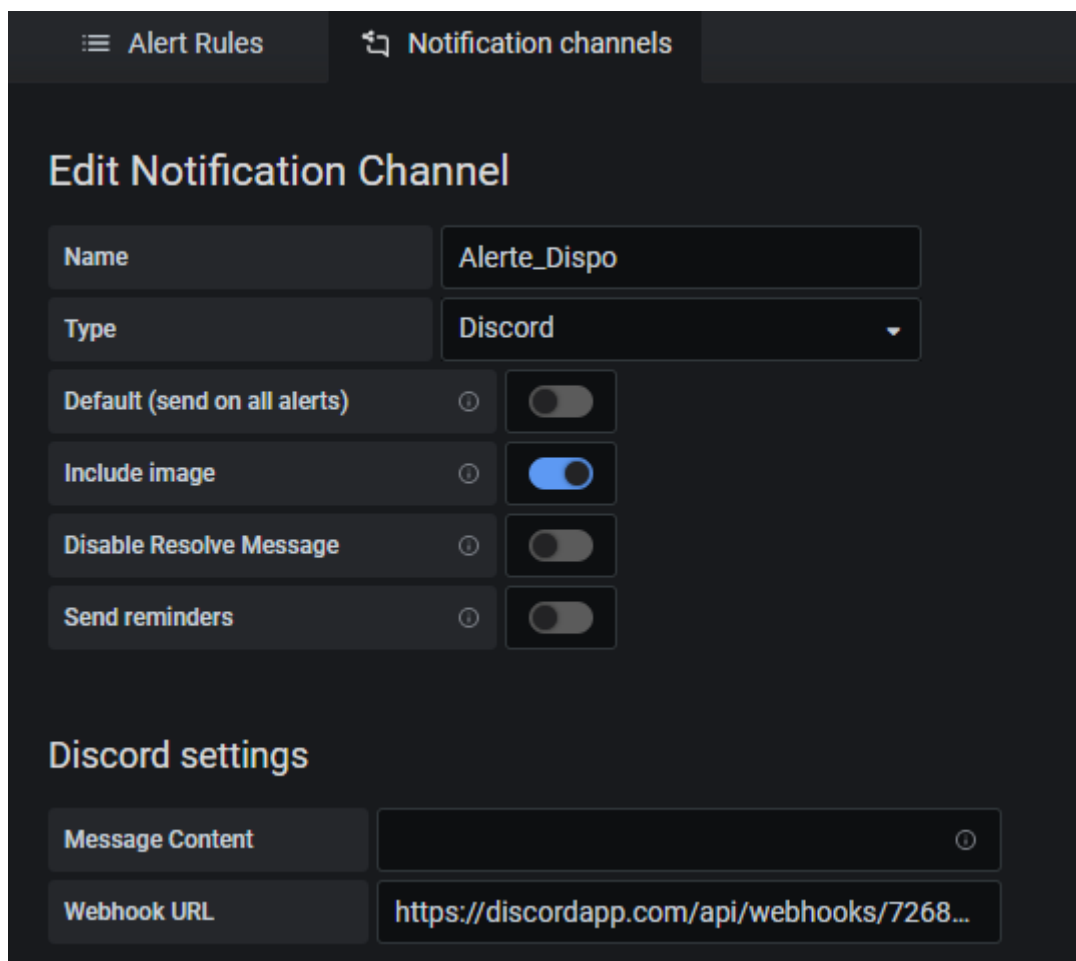
The figure shows a screenshot of the 'MODIFIER LE WEBHOOK' (Edit Webhook) form in Grafana. The form is dark-themed and contains the following fields:

- NOM:** A text input field containing 'Alert_Dispo'.
- SALON:** A dropdown menu showing '#général'.
- ICÔNE DU WEBHOOK:** A section with a text input field containing 'Nous recommandons une image d'au moins 256x256', an 'Uploader une image' button, and a Discord logo icon. Below the icon, it says 'Taille minimum: 128x128'.
- URL DU WEBHOOK:** A text input field containing 'https://discordapp.com/api/webhooks/72684...', with a 'Copier' button to its right.

At the bottom of the form, there is a link: [Besoin d'aide pour la configuration ?](#)

Figure 24: WebHook serveur Discord

Le lien généré par le webhook devrait être retranscrit sur le système d'alerte de Grafana.



The screenshot shows the 'Edit Notification Channel' page in Grafana. At the top, there are tabs for 'Alert Rules' and 'Notification channels'. The main heading is 'Edit Notification Channel'. Below this, there are several configuration options:

- Name:** A text input field containing 'Alerte_Dispo'.
- Type:** A dropdown menu set to 'Discord'.
- Default (send on all alerts):** A toggle switch that is currently turned off.
- Include image:** A toggle switch that is currently turned on.
- Disable Resolve Message:** A toggle switch that is currently turned off.
- Send reminders:** A toggle switch that is currently turned off.

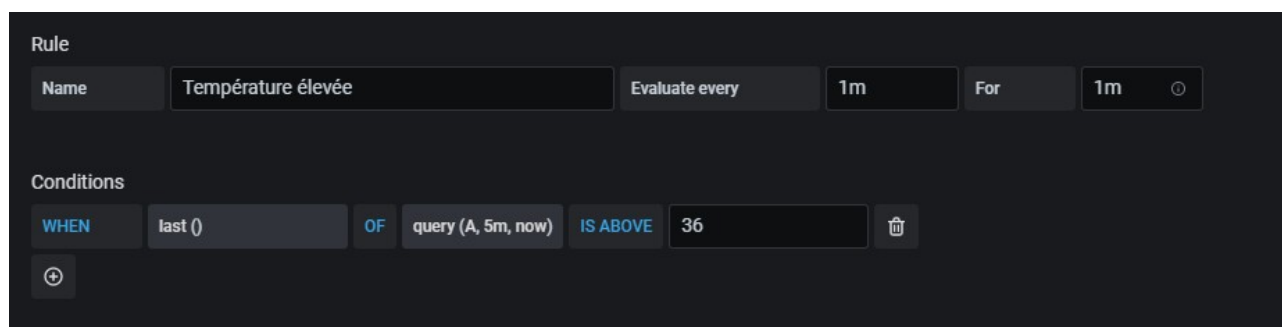
Below these options is a section titled 'Discord settings'. It contains two input fields:

- Message Content:** An empty text input field.
- Webhook URL:** A text input field containing the URL 'https://discordapp.com/api/webhooks/7268...'.

Figure 25: Configuration notifications Discord

Dans la partie notification de grafana, on crée un canal de notification dont les paramètres sont adaptés pour Discord, on lui donne un nom, son type, ainsi que le lien du webhook créé précédemment.

Une fois le canal créé nous devons configurer un dashboard « Graph » car ils prennent en compte le système d'alerte, on lui donne comme requête, la requête du niveau de température, et dans la partie alerte on configure les règles :



The screenshot shows the 'Rule' configuration page in Grafana. It has a 'Rule' section at the top and a 'Conditions' section below it.

Rule section:

- Name:** A text input field containing 'Température élevée'.
- Evaluate every:** A dropdown menu set to '1m'.
- For:** A dropdown menu set to '1m'.

Conditions section:

- WHEN:** A dropdown menu set to 'last ()'.
- OF:** A dropdown menu set to 'query (A, 5m, now)'.
- IS ABOVE:** A text input field containing '36'.

There is a trash icon to the right of the 'IS ABOVE' field and a plus icon at the bottom left of the 'Conditions' section.

Figure 26: Règle pour l'alerte de température

On définit son nom, sa fréquence de vérification, et le moment où l'alerte se déclenche, ici l'alerte se déclenchera dès que la température aura atteint 36 °C.

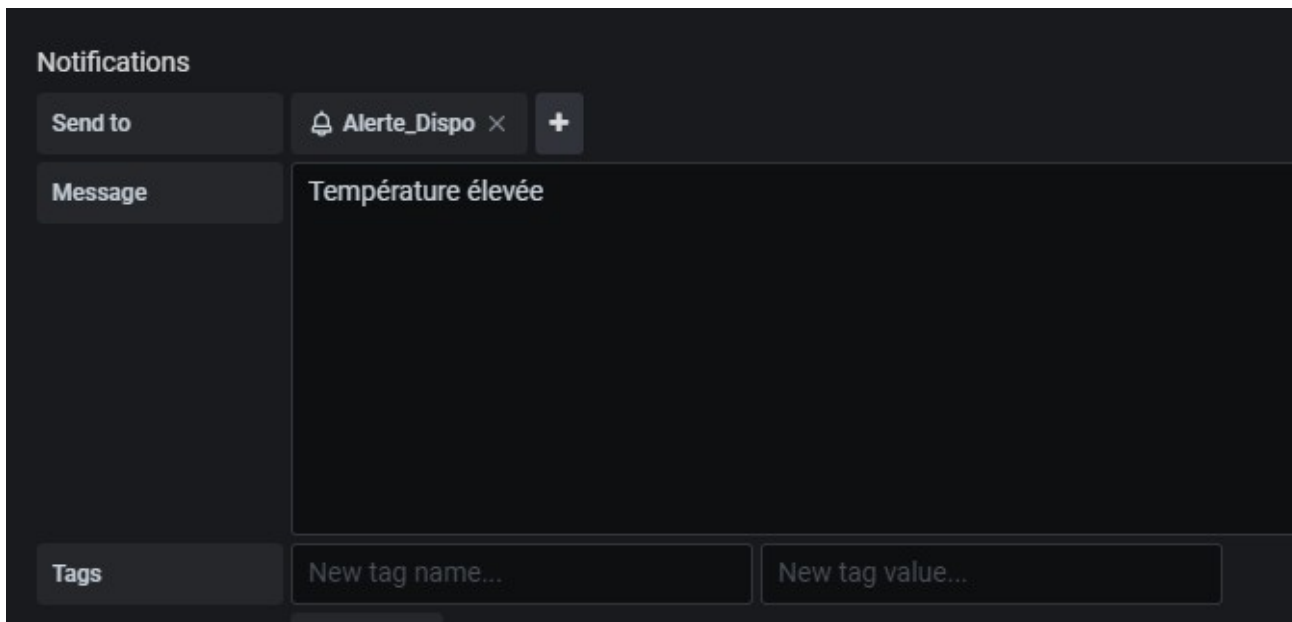


Figure 27: Paramètre du message

Ici on lui attribue le canal de notification créé précédemment avec un message.

Une fois la limite de température atteinte, Grafana envoie une notification sur le serveur :



Figure 28: Message reçu serveur discord

L'utilisateur est donc averti en cas de problème de température.

Conclusion de la simulation :

Cette simulation permet, la visualisation en temps réel, des informations émis par le dispositif qui a été déployé, bien que l'on est utilisé un ordinateur comme dispositif, on pourrait aussi utiliser un Raspberry Pi à la place ou encore un ESP32. En ce qui concerne les requêtes, on pourrait améliorer la répartition des informations, en créant une table par secteur de la forêt, et que l'on pourrait ensuite trier par le nom (« host ») du dispositif. Cela permettrait d'éviter d'avoir trop de tables créée et aussi de rencontrer des problèmes dus à des conflits de nom pour les tables.

De plus le support de notification sur des messageries comme discord permet d'avoir un suivi, même en dehors du lieu de travail, car discord est disponible sur plusieurs plateformes (PC, Smartphones, Montres Connectées,...)

Sources (A bien remplir):

- Figure 1 : <https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.html>
- Figures 2 et 3 : <https://www.connectwave.fr/techno-appli-iot/iot/reseaux-et-infrastructures-iot/>
- <https://www.synox.io/4-choses-a-savoir-sur-linternet-des-objets/>
- <https://azure.microsoft.com/fr-fr/overview/internet-of-things-iot/iot-technology-protocols/>
- Bluetooth (BLE) : <https://blog.groupe-sii.com/le-ble-bluetooth-low-energy/>
- <https://moodle.didex.fr/moodle/course/view.php?id=654>
- <https://moodle.didex.fr/moodle/course/view.php?id=589>
- <https://www.ecologique-solidaire.gouv.fr/prevention-des-feux-foret>
- <http://smartforest.pt/>
- projet de icjiter : <https://pdfs.semanticscholar.org/dc1e/4a751998713d7e51859876ae2ada849d0cee.pdf>
- Schéma IoTree : <https://business.esa.int/projects/iotrees>
- Rainforest : <https://rfcx.org/home>
- Rainforest (vidéo TED) : https://www.youtube.com/watch?time_continue=376&v=xPK2Ch90xWo&feature=emb_logo
- Rainforest schéma : <https://www.networkworld.com/article/3066880/protecting-the-rainforests-with-iot-and-recycled-phones.html>
- Mesure Co2 : <https://th-industrie.com/content/13-mesure-co2>
- <https://theiotmagazine.com/iot-in-the-wild-59c2525d62ef>
- Dashboard : <https://grafana.com/docs/grafana/latest/features/dashboard/dashboards/>

Bibliographie :

- Analytics for the Internet of Things (IoT) **Minteer, Andrew** 2017
- IoT Projects with Bluetooth Low Energy **Bhargava, Madhur** 2017
- IoT: Building Arduino-Based Projects **Waher, Peter Seneviratne, Pradeeka Russell, Brian** 2016

Citations :

[1] M. Han and H. Zhang, "Business intelligence architecture based on internet of things "