République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieure et de la Recherche Scientifique

Centre Universitaire Belhadj Bouchaib d'Aïn-Témouchent

Institut des Technologies

Département de Génie Électrique

Mémoire

Pour l'obtention du Diplôme de Master en Télécommunications

Option : Réseaux et Télécommunications

Présenté par :

Melle. Narimane DAOUD

Melle. Delel Abir LOUATI

Acces aux données d'un Réseau de Capteurs Sans Fil supportant 6LowPAN, en utilisant le protocole MQTT

Encadrant :

M. Ali BENZEBADJ

Maître de Conférences Classe "B", C.U.B.B.A.T.

Soutenu en 2019

Devant le jury composé de :

Président :	M. Chems-Eddine BEMMOUSSAT(MCB)	C.U.B.B.A.T.
Examinateurs :	M. Sihem SOUIKI(MAB)	C.U.B.B.A.T.
Encadrant :	M. Ali BENZERBADJ (MCB)	C.U.B.B.A.T.
Co-encadrant :	M. Mejdi KADDOUR (Professeur)	Université Oran 1 Ahmed BEN BELLA.

Remerciements

En premier lieu et avant tout, toute la gratitude à Allah. Nous remercions Allah le tout-puissant, le miséricordieux, de nous avoir appris ce que nous ignorons, de nous avoir donné la santé, la volonté et le courage et tout ce qui nous était nécessaire, pour l'achèvement de nos études et de réaliser ce modeste travail.

On tient tout d'abord à remercier notre encadrant M. Ali BENZERBADJ, Enseignant-Chercheur au Centre Universitaire BELHADJ Bouchaïb, Ain Témouchent, de nous avoir proposé ce sujet, ses précieux conseils, sa confiance en nous, ses orientations précises, sa patience et son aide durant toute la période de travail.

Nos vifs remerciements vont également aux membres du jury M. Chems-Eddine BEMMOUSSAT et M. SOUIKI, pour l'intérêt qu'ils portent à notre travail en acceptant de l'examiner et de l'enrichir par leurs propositions.

Nous adressons également nos sincères remerciements à M. Mustapha SADOK, Enseignant-Chercheur à l'institut INTTIC pour son aide, sa gentillesse, ses conseils et ses encouragements.

Nous remercions vivement tous nos enseignants du département Génie Electrique pour leurs efforts au cours de nos études universitaires.
Nous n'oublions pas de remercier nos collègues de la promotion 'RT-2019'.

Enfin, on remercie tous ceux qui ont contribué de loin ou de près à la réalisation de ce travail.

DAOUD & LOUATI.

Dédicace

Je dédie ce mémoire à mes chers parents, pour leur amour, leur soutien, leur encouragement et leur sacrifice, et pour tout ce qu'ils ont fait pour que je puisse arriver à ce stade.

À Ma chère mère Ghalima BENDIDANI, qui m'a entouré d'amour, qui a sacrifié son bonheur pour me voir heureuse et qui m'a encouragé durant toutes mes études, que dieu la garde.

À mon cher père Abdeldjebbar, qui est toujours disponible pour nous, et prêt à nous aider, rien ne pourra remplacer mon père, que dieu le garde et le protège.

À mes très chères sœurs Yousra et Feriel, qui ont cru en moi et qui m'ont redonné courage et sourire. Je vous dis merci pour vos efforts et votre soutien. Je vous adore.

À Mohammed El Amine MILIANI, merci pour ton amour, ton encouragement sans limite et pour m'avoir fait rire souvent. Merci d'avoir partagé ma joie et ma peine durant ces cinq longues années. Ce travail t'est dédié.

À ma grand-mère Meriem DALAA, et à toute ma famille.

À Abir, mon binôme Pour tout ce qu'elle a fait pour la réussite de ce travail

Et enfin un dédicace spécial pour mes Enseignants et pour la promotion RT 2019.

DAOUD Narimane.

Dédicace

Je rends grâce à Dieu de m'avoir donné le courage et la volonté d'avoir pu terminer mes études.

Je dédie ce modeste travail :

A la mémoire de mon père Madani, décédé trop tôt.
Tout l'encre du monde ne pourrait suffire pour exprimer mes sentiments envers vous très cher papa.
Vous avez toujours été mon école de patience, de confiance et surtout d'espoir et d'amour.
Vous êtes et vous resterez pour moi ma référence, la lumière qui illumine mon chemin.
Puisse Dieu, le tout puissant, l'avoir en sa miséricorde.

A ma chère mère Soraya.
Source inépuisable de tendresse, de patience et de sacrifice.
Aucune dédicace très chère maman, ne pourrait exprimer la profondeur des sentiments que j'éprouve pour vous.
Vous m'avez aidé et soutenu pendant de nombreuses années avec à chaque fois une attention renouvelée. J'espère ne jamais vous décevoir, ni trahir votre confiance.
Puisse Dieu tout puissant, vous préserver et vous accorder santé et longue vie.

A mon frère Mohamed Iheb et ma sœur Doha Zahra pour leur soutien moral et encouragement durant mes études.

A tous mes amis les plus sincères surtout M. Djawida, S. Inayat Allah, B. Khadidja et mon binôme Narimane qui m'a aidé à réaliser ce travail.

Et enfin un dédicace spécial pour mes Enseignants et pour la promotion RT 2019 et à toute personne qui me connait.

LOUATI Delel Abir.

Table des matières

Table des figures

Liste des tableaux

Liste des abréviations

CoAP Constrained Application Protocol

IdO Internet des Objets

MQTT Message Queuing Telemetry Transport

M2M Machine to Machine

RCSF Réseau de Capteurs Sans Fil

RFID Radio Frequency Identification

RPL Routing Protocol for Lossy Networks

SLIP Serial Line Internet Protocol

TCP Transfert Control Protocol

UDP transport User Datagramm Protcol

6LBR 6LoWPAN Border Router

6LoWPAN IPv6 over Low Power Wirelesse Personal Area Networks

Introduction Générale

L'Internet a beaucoup impacté notre mode de vie ces dernières années et continue de le faire, en particulier avec le nombre croissant d'objets / dispositifs que nous utilisons et qui sont capables de se connecter à Internet. Aujourd'hui, l'Internet des objets (IdO), ou Internet of Things (IoT) en Anglais, permet de connecter à Internet tout appareil (téléphones portables, véhicules, appareils ménagers et autres dispositifs portables avec capteurs ou/et actionneurs) afin que ces objets puissent échanger des données entre eux sur un réseau. Il est intéressant de noter qu'il existe une différence fondamentale entre l'Internet des objets et Internet, c'est l'absence du rôle humain.

Les RCSFs sont considérés comme l'une des technologies clés pour la mise en œuvre des architectures IdO. Ils sont au cœur des applications de l'Internet des objets. Ce type de réseaux sont déployés dans une zone d'intérêt pour collecter les informations sur leur environnement. Les nœuds capteurs communiquent entre eux par des liaisons sans fil, généralement en mode multi-sauts. La volonté de disposer de nœuds capteurs (aussi appelés motes en Anglais) en très grand nombre implique la nécessité d'en réduire les coûts de fabrication. On obtient ainsi, des équipements peu robustes au niveau matériel : autonomie énergétique limitée, faible débit de données, mémoire de capacité réduite et puissance de calcul faible. Pour toutes ces raisons, la mise en réseau de ces objets est désignée sous le sigle LLN (Low-Power and Lossy Networks).

Un RCSF peut être connecté à l'Internet à travers des objets robustes constitués de passerelles qui assurent la communication entre les deux infrastructures, à savoir Internet et le Réseau de Capteurs Sans Fil. Les nœuds capteurs de l'IdO communiquent avec internet via le routeur frontière 6LoWPAN (6LBR). Le 6LBR est en effet une solution prête au déploiement pour les routeurs de bordure, basée sur Contiki OS. Cette solution fonctionne sur la plate-forme intégrée Raspberry Pi qui agit comme un routeur de bordure qui va se charger de transformer et router les

trames d'un monde à l'autre (par ex. IEEE 802.15.4 <-> IPv6). Les nœuds d'extrémité IdO peuvent être identifiés uniquement avec une adresse IPv6. Les capteurs collectent des données en temps réel et les envoient à l'application requise (Cloud, Broker Message Queuing Telemetry Transport -MQTT-).

6LoWPAN est l'acronyme de (IPv6 over Low Power Wireless Personal Area Networks). La pile 6LoWPAN est implémentée et maintenue par IETF [International Engineering Task Force] pour fournir des itinéraires de communication entre les nœuds IoT à faible puissance et Ethernet (ou WiFi). La couche d'adaptation 6LoWPAN a été définie entre la couche liaison de données et la couche réseau du modèle OSI. Les principales fonctionnalités de cette couche sont la compression d'en-tête des paquets IPv6, la fragmentation et le réassemblage des paquets.

L'IdO a besoin de la capacité de travailler avec des périphériques de faible puissance pour réussir, et MQTT répond à tous les besoins de l'IdO en ce sens. MQTT est un protocole de messagerie, extrêmement léger, basé sur le principe de publication/abonnement. Il permet à deux équipements distants de communiquer via des messages de manière asynchrone avec une faible bande passante. Presque toutes les plates-formes IdO prennent en charge le protocole MQTT pour envoyer et recevoir des données à partir d'objets intelligents.

L'objectif principal de ce travail est d'accéder aux données d'un Réseau de Capteurs Sans Fil via 6LoWPAN à l'aide du protocole MQTT. Pour cela, nous avons réalisé une plate-forme constitué d'une carte Raspberry Pi 3, deux nœuds Telos B et un smartphone.

Le manuscrit s'articule autour de trois chapitres en plus d'une introduction générale et d'une conclusion générale.

Le premier Chapitre est consacré à la description des concepts de l'internet des objets et des Réseaux de Capteurs Sans Fil (RCSF). Le deuxième chapitre est dédié à la présentation du protocole 6LoWPAN qui est un protocole conçu pour l'interconnexion d'un réseau 802.15.4 (par ex., un RCSF) et un réseau IPv6. Le troisième Chapitre présente les outils matériels et logiciels que nous avons utilisés, ainsi que toutes les étapes nécessaires à la mise en œuvre de notre projet.

Chapitre 1

Internet Des Objets (IdO)

1.1 Introduction

En 1999, Le terme “Internet of Things” (Internet des Objets en français -IdO-) a été inventé par le britannique Kevin Ashton. L’IdO rend les objets autour de nous intelligents en leur offrant la possibilité de communiquer entre eux grâce aux technologies de l’information et de la communication. L’IdO est un réseau global d’objets interconnectés hétérogènes (capteurs, actionneurs, des objets RFID (Radio-Frequency IDentification), téléphones mobiles, etc.). Ces objets ont leurs propre identité numérique, à savoir une adresse IPv6.

Les Réseaux de Capteurs Sans Fil (RCSFs), qui sont considérés comme étant une passerelle entre le monde physique et le monde numérique, constituent un élément essentiel de l’Internet des Objets. Ils permettent de collecter les paramètres physiques de l’environnement de déploiement et les transmettre à une station de base appellée aussi nœud puits ou sink. Leur faible coût et leur facilité de déploiement les rendent attractifs pour différentes applications dans différents domaines (par ex., militaire, environnement, agriculture, santé, domotique etc.).

Ce chapitre est consacré à l’introduction de l’Internet des Objets et des Réseaux de Capteurs Sans Fil.

1.2 L'Internet des Objets (IdO)

L'Internet est devenu en quelques années le vecteur principal de diffusion de l'information. Il s'est imposé dans de nombreux domaines comme une infrastructure essentielle pour les individus, les entreprises et les institutions. Il se transforme progressivement en un réseau étendu, appelé Internet des Objets (IdO) , en Anglais Internet of Things (IoT), reliant plusieurs milliards d'êtres humains mais aussi des dizaines de milliards d'objets [Nem15]

Figure 1.1 – Internet of Things (IoT, en Français IdO)[mf17].

L'Internet des objets est aujourd'hui présent dans les équipements, les capteurs et les données que la plupart des entreprises exploitent. Notons que chaque objet connecté dispose d'une adresse IP [BF].

La Figure 1.2 montre l'aspect spatio-temporel de l'IdO, qui permet aux personnes de se connecter de n'importe où et à n'importe quel moment à travers des objets connectés (Smartphone, tablettes, capteurs, caméras de vidéosurveillance, etc.).

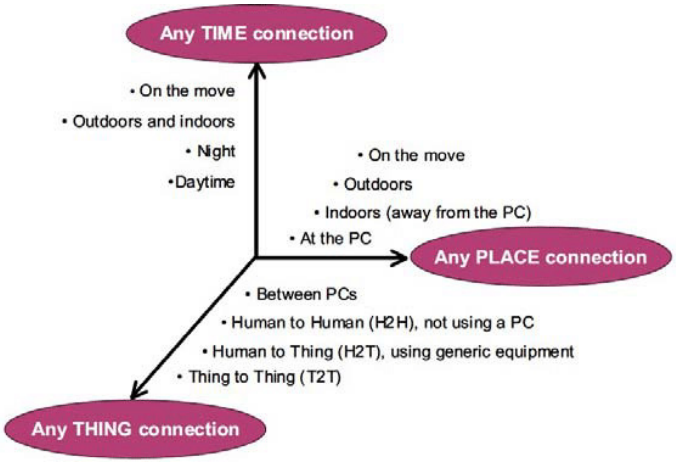


Figure 1.2 – Une nouvelle dimension pour l'IdO.

1.2.1 Objet connecté

Est un dispositif permettant de collecter, stocker, transmettre et traiter des données issues du monde physique (voir Figure 1.3). Il s'agit d'une source de données, identifiés et identifiables de façon unique et ayant un lien direct ou indirect avec Internet [GP11].

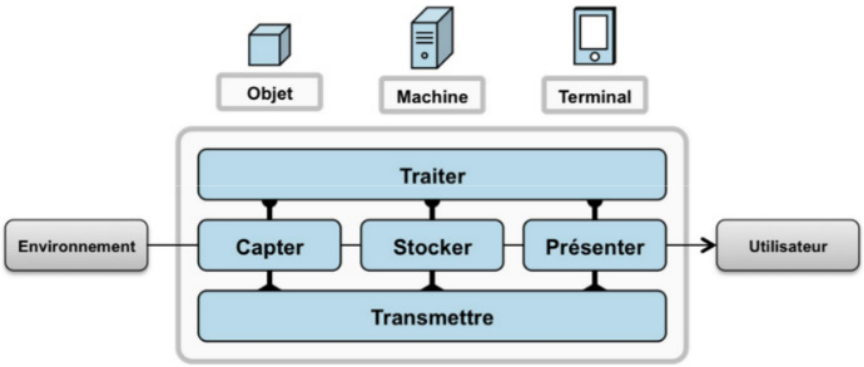


Figure 1.3 – Collecte de données [GP11].

On distingue deux types d'objets :

— Les objets passifs : ils utilisent généralement un tag (puce RFID, code barre 2D). Ils embarquent une faible capacité de stockage (de l'ordre du kilo-octet) leur permettant d'assurer un rôle d'identification. Ils peuvent parfois, dans le cas d'une puce RFID, embarquer un capteur (température, humidité) et être réinscriptibles [GP11].

— Les objets actifs : ils peuvent être équipés de plusieurs capteurs, d'une plus grande capacité de stockage et être dotés d'une capacité de traitement ou encore être en mesure de communiquer sur un réseau. [GP11].

1.2.2 L'IdO aujourd'hui et son impact dans le monde

Les premiers objets connectés n'apparaissent que dans les années 1990. Il s'agit de grille-pain, machines à café ou autres objets du quotidien. En 2000, le fabricant coréen LG est le premier industriel à parler sérieusement d'un appareil électroménager relié à internet. Les années 2000 verront les premières expérimentations d'appareils connectés à Internet. Ils l'utilisent notamment pour consulter des informations de manière automatique.

En 2003, la population mondiale s'élevait à environ 6,3 milliards d'individus et 500 millions d'appareils étaient connectés à Internet. Le résultat de la division du nombre d'appareils par la population mondiale (0,08) montre qu'il y avait moins

d'appareils connectés par personne. Selon la définition de Cisco IBSG, l'IdO n'existait pas encore en 2003 car le nombre d'objets connectés était faible.

En raison de l'explosion des Smartphones et des tablettes, le nombre d'appareils connectés à Internet a atteint 12,5 milliards en 2010, alors que la population mondiale était de 6,8 milliards. C'est ainsi que le nombre d'appareils connectés par personne est devenu supérieur à 1 (plus précisément 1,84) pour la première fois de l'histoire. En affinant ces chiffres, Cisco IBSG a situé l'apparition de l'IdO entre 2008 et 2009 (voir Figure 1.4).

En ce qui concerne l'avenir, Cisco IBSG estime que 50 milliards d'appareils seront connectés à Internet d'ici à 2020. Il est important de noter que ces estimations ne tiennent pas comptedes progrès rapides d'Internet ni des avancées technologiques, mais reposent sur les faits avérés à l'heure actuelle [Eva11].

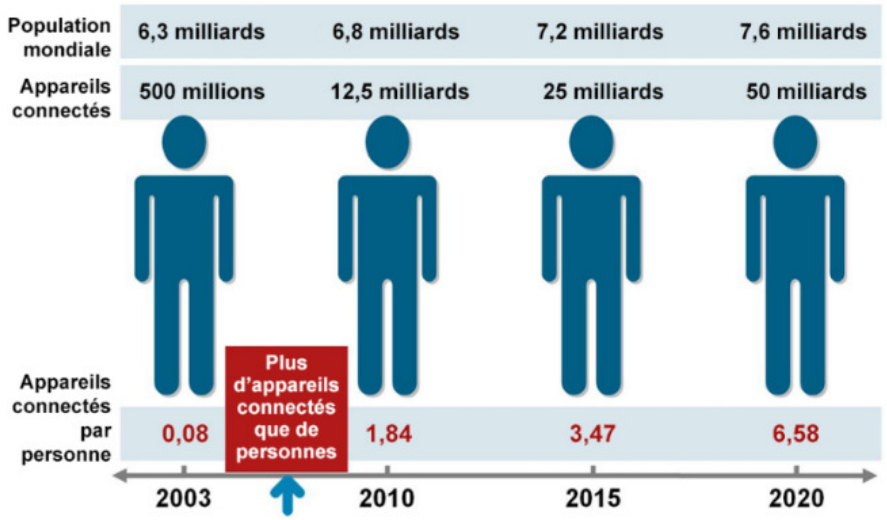


Figure 1.4 – L'Internet des objets est apparu entre 2008 et 2009. [Eva11].

1.3 Les domaines d'application de l'IdO

L'Internet des objets sera un réseau ubiquitaire et atteindra divers secteurs de notre vie quotidienne. Parmi les domaines d'applications de l'IdO, nous citons [TB+17] :

- **Villes intélligentes :** surveillance du trafic des véhicules pour optimiser les itinéraires de conduite et par conséquent décongestionner le trafic.
- **Eau intelligente :** surveillance de l'eau potable (surveiller la qualité de l'eau du robinet dans les villes), détection de fuite chimique dans les rivières .

— **Les compteurs intelligents :** surveillance des niveaux d'eau, d'huile et de gaz dans les réservoirs de stockage et les citernes.
— **Logistique :** localisation des objets, à savoir recherche d'objets individuels dans de grandes surfaces, les entrepôts ou dans les ports.
— **Contrôle industriel :** application machine-à-machine (M2M) : auto-diagnostic de la machine et contrôle des actifs, surveillance et contrôle de la température à l'intérieur des frigos industriels et médicaux contenant des marchandises sensibles.
— **Agriculture intelligente :** améliorer la qualité de l'agriculture en supervisant l'environnement des cultures, optimiser l'eau d'irrigation et planifier les travaux agricoles.
— **Domotique :** surveillance de la consommation d'énergie et de l'approvisionnement en eau afin d'économiser les coûts et les ressources.
— **Santé :** aide à la détection des chutes des personnes âgées ou handicapées vivant indépendamment.

Figure 1.5 – Divers domaines d'applications de l'IdO [AMB+17].

1.4 Fonctionnement de l'IdO

1.4.1 Technologies de l'IdO

L'Internet des objets (IdO) permet l'interconnexion de différents objets intelligents via l'Internet. Son fonctionnement est basé sur l'utilisation de plusieurs technologies. Nous citons, entre autres ; [AMB+17] :

— **Radio Frequency Identification (RFID) :** le terme RFID englobe toutes les technologies qui utilisent les ondes radio pour identifier automatiquement des objets ou des personnes. C'est une technologie qui permet de mémoriser et de récupérer des informations à distance grâce à une étiquette qui émet des ondes radio. Il s'agit d'une méthode utilisée pour transférer les données des étiquettes à des objets, ou pour identifier les objets à distance. L'étiquette contient des informations stockées électroniquement pouvant être lues à distance. Les étiquettes peuvent avoir différentes formes (Figure 1.6) et peuvent être passives ou actives.

— **Réseau de Capteurs Sans Fil (RCSF) :** c'est un ensemble de nœuds qui communiquent sans fil et qui sont organisés en un réseau coopératif. Chaque nœud possède une capacité de traitement et peut contenir différents types de mémoires, un émetteur-récepteur RF et une source d'alimentation, comme il peut inclure divers types de capteurs et des actionneurs.

— **Machine to Machine (M2M) :** c'est l'association des technologies de l'information et de la communication avec des objets intelligents dans le but de donner à ces derniers les moyens d'interagir sans intervention humaine avec le système d'information d'une organisation ou d'une entreprise.

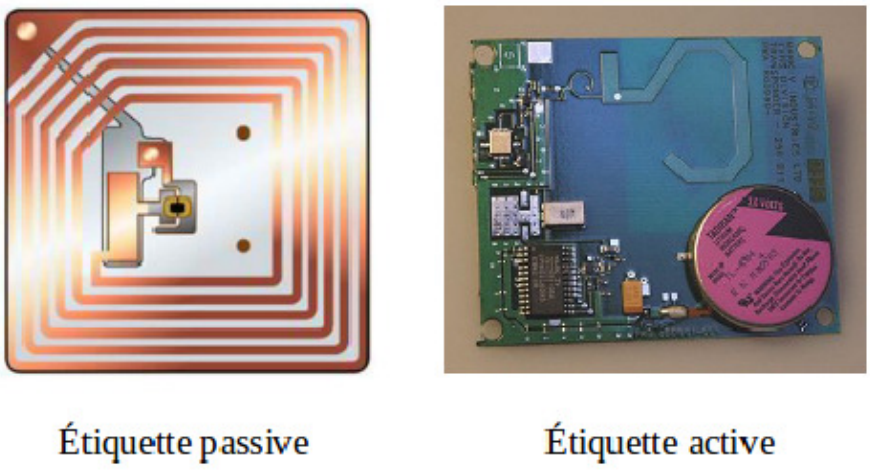


Figure 1.6 – Les étiquettes RFID [DÇYG16].

1.4.2 Protocoles de fonctionnement de l'IdO

La mise en application à une large échelle du concept d'IdO apparait largement tributaire d'une standardisation de la communication entre Objets dite M2M [AMB+17].

- Au niveau de la couche de Liaison, le standard IEEE 802.15.4 est plus adapté que l'Ethernet aux environnements industriels difficiles.
- Au niveau réseau, le standard IPv6 over Low Power Wirelesse Personal Area Networks (6LoWPAN) a réussi à adapter le protocole IPv6 aux communications sans fil entre nœuds à très faible consommation.
- Au niveau routage, l'IETF a publié en 2011 le standard Routing Protocol for Lossy Networks (RPL) .
- Au niveau de la couche application, les protocoles les plus connus sont le protocole Constrained Application Protocol (CoAP) qui utilise le protocole de transport User Datagramm Protcol (UDP), et le protocole Message Queuing Telemetry Transport (MQTT) qui utilise le protocole de transport Transfert Control Protocol (TCP).

1. **CoAP (Constrained Application Protocol) :** est un protocole de la couche d'application pour les applications IdO. Il définit un protocole de transfert Web basé sur les fonctionnalités HTTP, et est lié à UDP (et non TCP) par défaut qui le rend plus approprié pour les applications IdO. En outre, CoAP modifie certaines fonctionnalités HTTP pour répondre aux exigences de l'IdO telles que la faible consommation d'énergie et le fonctionnement en présence de liens à perte [AMB+17].
2. **MQTT (Message Queue Telemetry Transport) :** représente un protocole de messagerie idéal pour les communications IdO et M2M. MQTT utilise le modéle de publication/souscription pour offrir une flexibilité de transition et une simplicité d'implémentation. Il convient aux périphériques à ressources limitées qui utilisent des liens peu fiables ou à faible bande passante. MQTT est construit en haut du protocole TCP. Il se compose de trois composants ; abonnés, éditeurs et courtiers. De nombreuses applications utilisent MQTT telles que les soins de santé, la surveillance, le compteur d'énergie et la notification de Facebook [AMB+17].

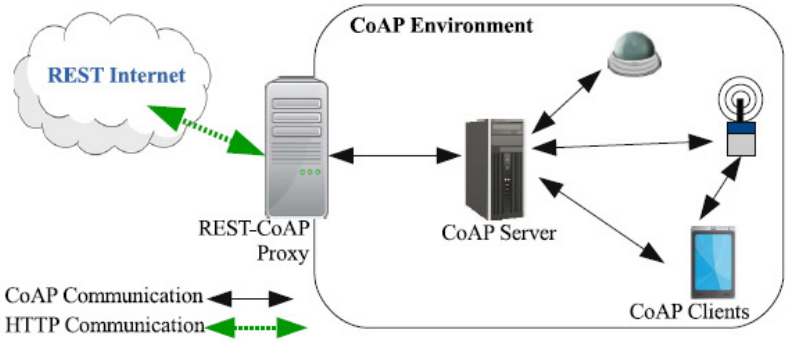


Figure 1.7 – Fonctionnement du protocole CoAP [AFGM+15].

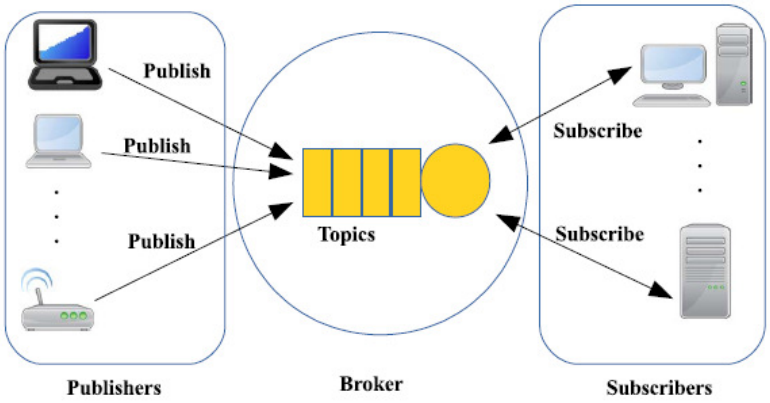


Figure 1.8 – Fonctionnement du Protocole MQTT [AFGM+15].

1.5 Les Réseaux de Capteurs Sans Fil (RCSFs)

Les progrès récents dans la technologie des Systèmes de Microélectro-Mécaniques (MEMS) et dans le marché des réseaux sans fil, ont permis la mise au point de nœuds capteurs multifonctionnels et économiques afin d'offrir des solutions intéressantes pour le traitement des données environnementales complexes. La mise en réseau de ces nœuds capteurs donne lieu à un Réseau de Capteurs Sans Fil (RCSF), en Anglais Wireless Sensor Network (WSN) [ASSC02b].

1.5.1 Les nœuds capteurs

Un nœud capteur sans fil (dit "mote" en Anglais) est un dispositif électronique de taille réduite avec des ressources très limitées, qui permet de transformer une grandeur physique (température, luminosité, humidité, etc.) en une donnée utilisable par un système d'information.

Chaque nœud capteur a la capacité de communiquer avec d'autres nœuds capteurs via une liaison sans fil [Wik15].

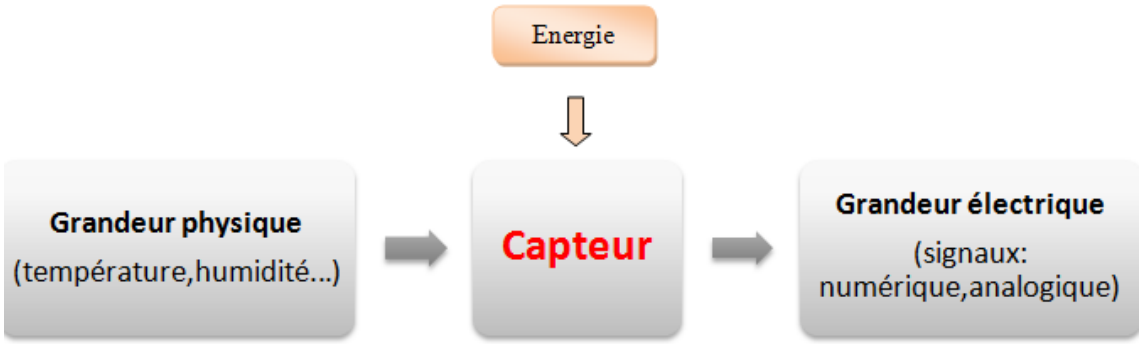


Figure 1.9 – Chaîne de collecte des données.

Les composants d'un nœud capteur

Un nœud capteur est composé de quatre composants de base, illustrés dans la Figure 1.10 : une unité de capture, une unité de traitement, une unité transmission et une unité d'alimentation (énergie). Un nœud capteur peut également avoir des composants supplémentaires dépendants de l'application, tels qu'un système de localisation, un générateur de puissance et un mobilisateur [ASSC02b].

- **Unité de capture (Sensing unit) :** elle est généralement composée de deux sous-unités, les capteurs et les convertisseurs analogique-numérique (appelé CAN). Les signaux analogiques produits par les capteurs basés sur le phénomène observé sont convertis en signaux numériques par le CAN, puis transmis à l'unité de traitement [ASSC02b].
- **Unité de traitement (Processing unit) :** est composée de deux interfaces, une interface avec l'unité de capture et une autre avec l'unité de transmission. Cette unité est également composée d'un processeur qui est la partie la plus importante d'un nœud capteur, et une petite unité de stockage. Elle gère les procédures permettant au nœud capteur de collaborer avec les autres nœuds pour effectuer les tâches de capture, et stocker les données collectées [ASSC02b].
- **Unité de transmission (Transceiver unit) :** est composée d'un module radio (émetteur/récepteur). Elle est responsable de toutes les émissions et réceptions de données sur un medium sans fil [ASSC02b].
- **Unité d'énergie (Power unit) :** C'est un élément important dans l'architecture d'un nœud capteur, puisqu'elle permet de garder le fonctionnement du nœud capteur au sein d'un réseau pour une durée limitée. Cette unité se présente généralement sous la forme de batteries afin de fournir une énergie pour toutes les autres unités [ASSC02b].

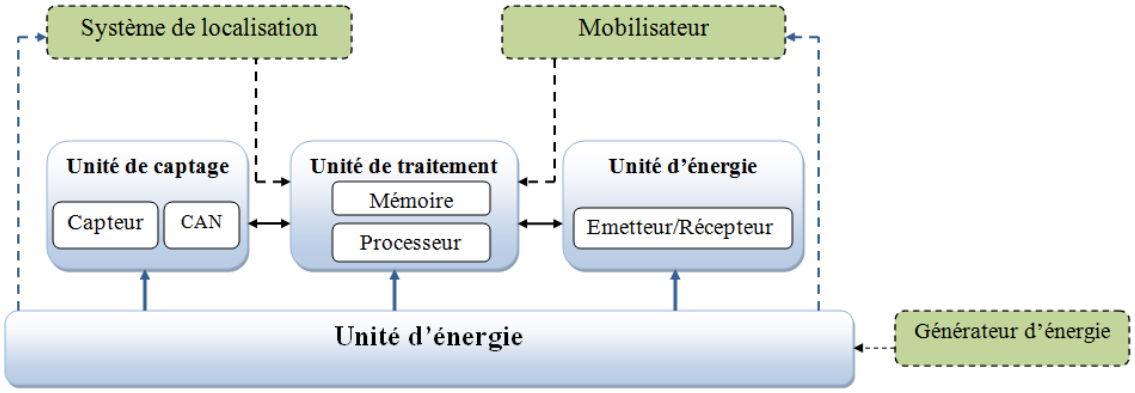


FIGURE 1.10 – Composants d'un nœud capteur.

Les plates-formes existantes des nœuds capteurs

De nos jours, la technologie des capteurs sans fil a fait des avancées considérables. Les modules deviennent de plus en plus petits et les durées de vie prévues augmentent. Le marché de cette technologie a été ouvert à l'industrie par le fournisseur le plus connu, à savoir Crossbow Inc., avec ses capteurs Mica2 et MicaZ et autres (voir Tableau 1.1) [Nab12].

	Unité de captage	Unité de traitement			Unité de transmission			Unité d'énergie	
		Processeur	RAM	Mémoire Flash	Module radio	Bande de fréquence	Débit de transmission		
Telos B (crossbow)	lumière température humidité	TI MSP430	10Ko	48Ko	Chipcon CC2420 IEEE802.15.4	2.4GHz	250Kbps	2X AA batteries	
Mica2 (crossbow)	lumière température accélération Magnétique pression…etc.	Atmel ATmega 128L	4Ko	128Ko	Chipcon CC100	315, 433MHz Ou 868/916MHz	38.4Kbps	2X AA batteries	
Micaz (crossbow)	lumière température accélération Magnétique pression…etc.	Atmel ATmega 128L	4Ko	128Ko	Chipcon CC2420 IEEE802.15.4	2,4GHz	250Kbps	2X AA batteries	
Tmote sky (Moteiv)	Lumière température humidité	TI MSP430	10Ko	48Ko	Chipcon CC2420 IEEE802.15.4	2,4GHz	250Kbps	2X AA batteries	

TABLE 1.1 – Exemples de nœuds capteurs existants sur le marché.

1.5.2 Définition des RCSFs

Les Réseaux de Capteurs Sans Fil (RCSFs) sont un cas particulier des réseaux Ad-hoc. Ils sont composés d'un grand nombre de nœuds capteurs multifonctionnels, à faible coût, déployés dans une zône d'intérêt. Ces nœuds capteurs sont capables de détecter les grandeurs physiques de l'environnement et de communiquer les informations collectées via des transmissions sans fil, à une station de base (sink), et ce à travers généralement une communications multi-sauts [ASSC02b].

1.6 Architecture des Réseaux de Capteurs Sans Fil

1.6.1 Architecture de communication

Un RCSF est constitué d'un ensemble de nœuds capteurs permettant de capturer des événements, traiter et de transmettre les informations recueillies à un nœud puits (sink), qui a son tour les envoie à un centre de traitement. Les nœuds capteurs ainsi que le nœud sink peuvent être fixes ou mobiles dans une zône de capture [ASSC02b].

- **Nœuds capteurs :** leur type, leur architecture et leur prositon géographique dépendent de l'exigence plication en question. Leur énergie est souvent limitée puisqu'ils sont alimentés par des batteries [Sah11].
- **Nœud Sink :** c'est un nœud particulier du réseau. Il est chargé de la collecte des données issues des différents nœuds du réseau. Il doit être toujours actif puisque l'arrivée des informations est aléatoire. C'est pourquoi son énergie doit être illimitée. Dans un RCSF plus ou moins large et à charge un peu élevée, on peut trouver deux ou plusieurs sinks, et ce pour alléger la charge du réseau [Sah11].
- **Centre de traitement des données :** c'est le centre vers lequel, les données collectées par le sink, sont envoyées (voir Figure 1.11) [Sah11].

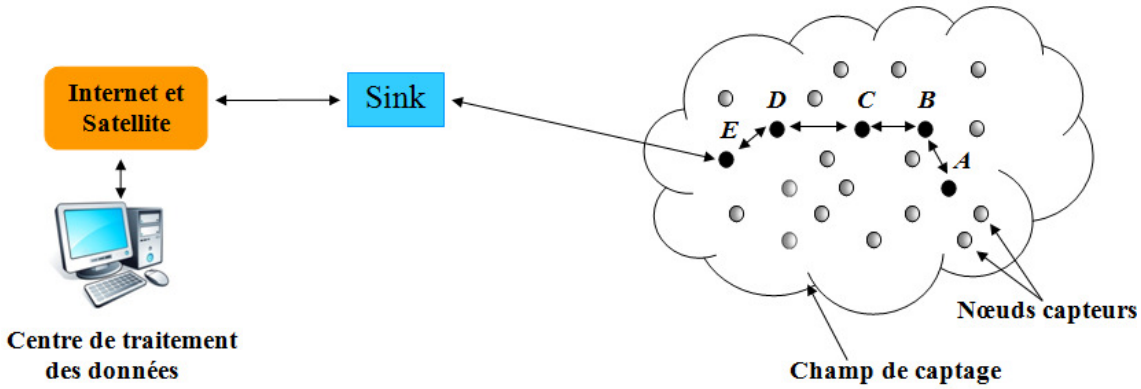


Figure 1.11 – Architecture de communication d'un RCSF [ASSC02b].

1.6.2 Architecture protocolaire

Comme illustrée dans Figure 1.12, la couche protocolaire des RCSFs comprend cinq couches, à savoir la couche application, la couche de transport, la couche réseaux, la couche liaison de données et la couche physique, qui ont les mêmes fonctions que celles du modèle OSI [ASSC02a]

De plus, cette pile possède trois plans (niveaux) de gestion, à savoir le plan de gestion des tâches qui permet de bien d'affecter les taches aux nœuds capteurs, le plan de gestion de mobilité qui permet de garder une image sur la localisation des nœuds pendant la phase de routage, et le plan de gestion d'énergie qui permet de conserver le maximum d'énergie. Notons, qu'Il existe une interaction entre les différentes couches et les différents niveaux.

Les couches de la pile protocolaire

- **La couche physique :** La couche PHY fournit une interface entre la couche MAC et le canal radio physique. Cette couche est responsable de [Rot12] :
  - La transmission et la réception des données sur le médium radio en utilisant une fréquence définie et un mécanisme de modulation.
  - Activation et désactivation de l'émetteur-récepteur radio
  - Mesurer la puissance du signal reçu et le comparer avec le niveau du bruit radio ambiant. Ce processus est utilisé par le mécanisme Clear Channel Assesment (CCA) pour déterminer si le canal est utilisé ou non.
  - L'allumage et la mise en veille de la puce radio en fonction des besoins de la couche MAC.
  - La détection d'énergie et indication de la qualité du signal.
- **La couche liaison :** Elle est composée de deux sous couches. La sous couche Logical Link Control -LLC- fournit une interface entre la couche liaison et la couche réseau. La sous couche Media Access Control -MAC-, contrôle l'accès au canal sans fil [ASSC02a].
- **La couche réseau :** La couche réseau s'occupe du routage des données. Plusieurs nouveaux algorithmes de routage, à efficacité energétique, ont été conçus spécialement pour les RCSFs [ASSC02a].
- **La couche transport :** Cette couche est chargée du transport des données, de leur découpage en paquets, du contrôle de flux, de la conservation de l'ordre des paquets et de la gestion des éventuelles erreurs de transmission [ASSC02a].
- **La couche application :** Cette couche assure l'interface avec les applications. Il s'agit donc du niveau le plus proche des utilisateurs, [ASSC02a].

Les plans de gestion de la pile protocolaire

- **Le niveau de gestion d'énergie :** Les fonctions intégrées à ce niveau consistent à gérer l'énergie consommée par les capteurs. Un nœud capteur peut par exemple éteindre son interface de réception dès qu'il reçoit un mes-

sage d'un nœud voisin afin d'éviter la réception des messages dupliqués. De plus, quand un nœud possède un niveau d'énergie faible, il peut diffuser un message aux autres nœuds capteurs pour ne pas participer aux tâches de routage, et conserver l'énergie restante aux fonctionnalités de capture, par exemple [ASSC02a].

— **Le niveau de gestion de mobilité :** Ce niveau détecte et enregistre tous les mouvements des nœuds capteurs, de manière à leur permettre de garder continuellement une route vers l'utilisateur final, et maintenir une image récente sur les nœuds voisins. Cette image est nécessaire pour pouvoir équilibrer l'exécution des tâches et la consommation d'énergie [ASSC02a].

— **Le niveau de gestion des tâches :** Lors d'une opération de capture dans une région donnée, les nœuds composant le réseau ne doivent pas obligatoirement travailler avec le même rythme. Cela dépend essentiellement de la nature du capteur, son niveau d'énergie et la région dans laquelle il a été déployé. Pour cela, le niveau de gestion des tâches assure l'équilibrage et la distribution des tâches sur les différents nœuds du réseau afin d'assurer un travail coopératif et efficace en matière de consommation d'énergie, et par conséquent, prolonger la durée de vie du réseau [ASSC02a].

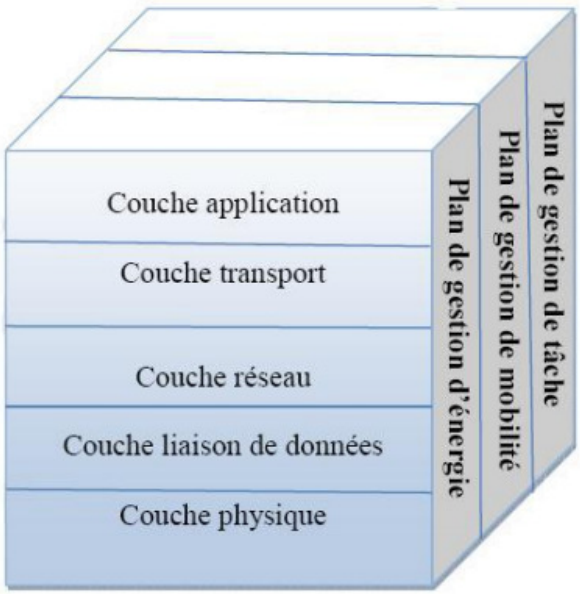


Figure 1.12 – Architecture protocolaire d'un RCSF [ASSC02a].

1.7 Domaines d'application des Réseaux de Capteurs Sans Fil

1.7.1 Domaine militaire

Les applications militaires ont constituées la locomotive de la recherche pour les RCSFs. Ce type de réseau est en effet très intéressant pour les militaires car il offre plusieurs avantages : utiliser un réseau de capteurs dans un endroit stratégique ou difficile d'accès, Surveiller toutes les activités des ennemies, Analyser le terrain avant d'envoyer des troupes [Nab12].

1.7.2 Domaine de La surveillance environnementale

l'aspect miniature des nœuds capteurs permet de les déployer dans des zône difficile d'accès ou carrémment inaccessibles à l'homme, comme par exemple les volcans, les terrains de conflit et les champs de bataille. On peut aussi utiliser les RCSFs pour la surveillance du degré de maturité des récoltes (raisin), la mesure de la qualité de l'eau ou de l'air [Nab12].

1.7.3 Domaine industriel

Les applications militaires ont constituées la locomotive de la recherche pour les RCSFs. Ce type de réseau est en effet très intéressant pour les militaires car il offre plusieurs avantages. Il s'agit d'un réseau qui s'installe rapidement, dynamiquement et sans aucune infrastructure. Ainsi, il offre un atout de taille pour surveiller les mouvements de l'ennemi, communiquer à bas coût entre les unités avec une logistique peu compliquée [Nab12].

1.7.4 Domaines urbain et domotique

Dans le milieu urbain, les nœuds capteurs sont déjà utilisés pour la localisation des bus, pour des tickets électroniques et pour la sécurité. Une des applications est la surveillance du trafic routier avec les réseaux de capteurs déployés sur les autoroutes. De plus, les maisons, les bâtiments, les bureaux équipés de capteurs intelligents permettent de construire des systèmes pervasifs où l'information est omniprésente [Nab12].

1.7.5 Domaine médical

La recherche sur l'usage des capteurs intelligents dans le domaine médical inclut les moyens d'hospitalisation à domicile, l'intégration des micro-capteurs dans le corps (par ex., construire un Body Area Network -BAN-) et la gestion des urgences. Parmi les applications les plus utiles, on cite la télésurveillance des signes vitaux et des niveaux d'activité à domicile des personnes âgées ou handicapées ainsi que le contrôle à distance des données physiologiques [Nab12].

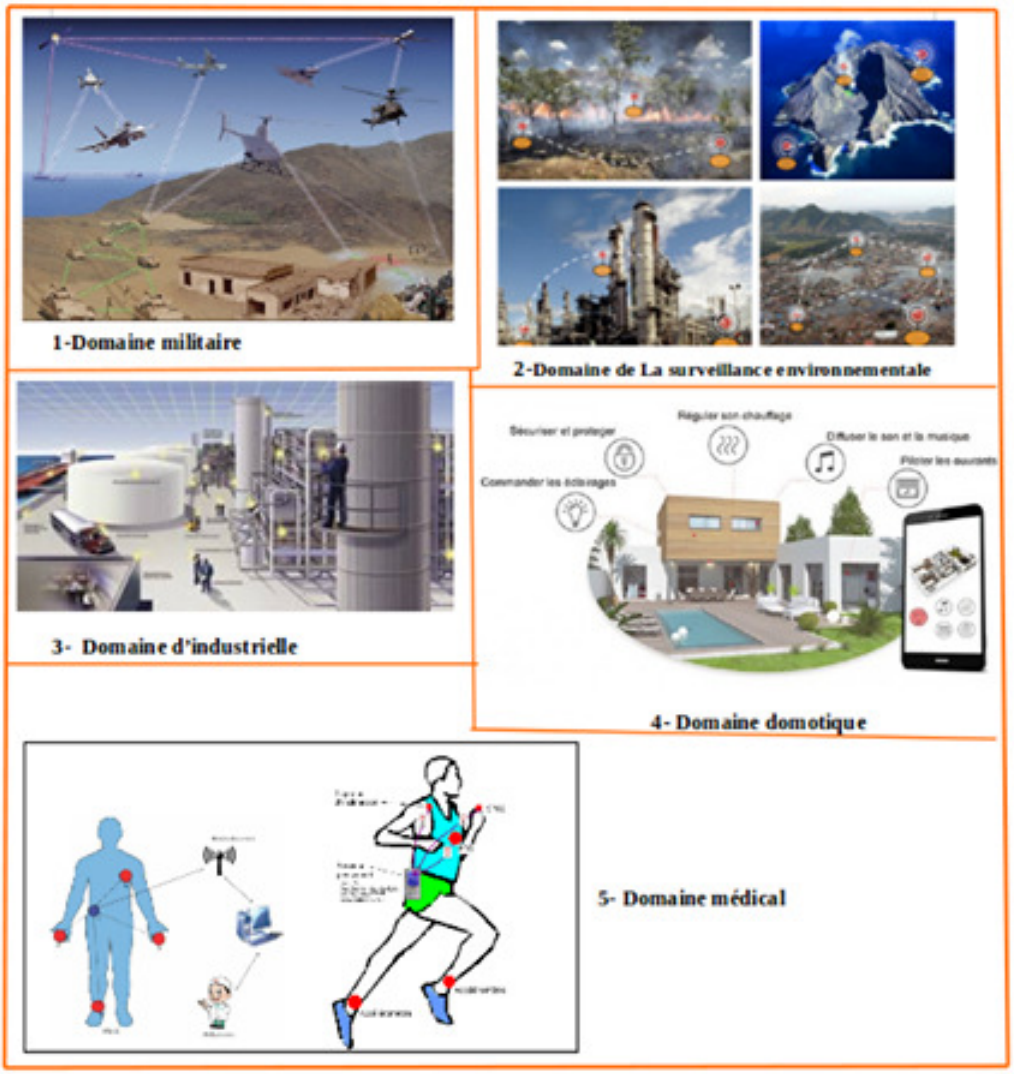


Figure 1.13 – Domaines d'application des RCSFs.

1.8 Conclusion

Dans ce chapitre, nous avons présenté quelques concepts sur l'internet des objets, ces domaines d'application et son fonctionnement. Ensuite nous avons décrit d'une façon générale le fonctionnement des RCSFs, leur architecture et leurs domaines d'application.

Dans le chapitre suivant nous abordons le protocole 6LoWPAN (IPv6 over Low-power Wireless Personal Area Networks) qui est un protocole conçu pour l'interconnexion d'un réseau 802.15.4 et un réseau IPv6.

Chapitre 2

Réseaux 6LoWPAN

2.1 Introduction

L'interconnexion des Réseaux de Capteurs Sans Fil (RCSFs) avec Internet constitue un avantage considérable en termes de collecte de données. L'accès à ce type de réseaux à partir d'internet a été rendu possible grâce à l'arrivée du protocole Ipv6. Ce dernier permet en effet l'affectation d'une adresse IP unique à chaque nœud, ce qui rend un nœud capteur accessible depuis internet.

La résolution du problème d'interconnexion des RCSFs avec le monde IP, nécessite le développement d'une couche permettant une adaptation fluide entre les exigences de la couche réseau IPv6 et les capacités de la couche liaison 802.15.4. Cette couche d'adaptation s'appelle Ipv6 over Low power Wireless Personal Area Networks (6LoWPAN), que nous allons décrire en détail dans ce Chapitre.

2.2 Standard 802.15.4

2.2.1 description de la norme IEEE 802.15.4/ZigBee

L'institut des ingénieurs électriciens et électroniciens (ou l'IEEE) soutient de nombreux groupes de travail pour développer et maintenir des normes de communications sans fil et filaires. Nous citons à titre d'exemple la norme 802.3 pour Ethernet et 802.11 pour les LAN sans fil, en Anglais Wireless Local Area Networks (WLAN). Le groupe de normes 802.15 spécifie une variété de réseaux personnels sans fil, en Anglais Wireless Personal Area Networks (WPAN) tel que ZigBee qui s'appuie sur la norme IEEE 802.15.4 qui est très utilisée dans les protocoles de communication des RCSFs. Nous rappelons que ces derniers sont caractérisés par un débits de données trop faible, une basse consommation d'energie et une faible portée [AC17]. La spécification ZigBee propose une pile protocolaire propriétaire et légère.

La norme IEEE 802.15.4 est la norme fondamentale qui définit la fonctionnalité de la couche physique (PHY) et de la couche de contrôle d'accès au médium (MAC) pour les réseaux personnels à faible puissance, en Anglais Low power Wireless Personal Area Networks (LoWPAN). Elle est caractérisée par [BF] :

- 3 bandes de fréquences : 868 MHz (Europe), 915 MHz (USA) et 2450 MHz (mondial).
- Débits de données trop faible 250 kb/s, 100 kb/s, 40 kb/s et 20 kb/s.
- Distance maximale entre deux nœuds : 100 mètres.
- Supporte les topologies en étoile, paire à paire et hiérarchique.
- Utilise le protocole (Carrier Sense Muliple Access / Collision Avoidance (CSMA/CA) pour l'accès partagé au canal de transmission.
- Permet l'allocation de tranches de temps garanties (GTS) pour les applications nécessitant des garanties de délais déterministes.
- Basse consommation énergétique.

2.2.2 Principe de fontionnement

La norme 802.15.4 définit au niveau de la couche MAC trois types de dispositifs qui utilisent tous le même canal physique pour communiquer :

- Le premier dispositif est le coordinateur qui gère les fonctions de haut niveau du réseau comme l'authentification et la sécurité. Il existe un seul coordinateur pour tout le réseau. Il s'agit dans la plupart du temps du nœud puits (appelé la racine).
- Les dispositifs à fonction complète (Full Function Devices), appelés FFD,

sont des dispositifs dotés de toutes les fonctions. Ils peuvent par conséquent jouer le rôle d'un coordinateur ou d'un dispositif relié à un capteur. Ils aident des dispositifs plus restreints en fournissant des fonctions telles que la coordination du réseau, l'acheminement de paquets ou l'interaction avec d'autres types de réseaux. Les FFD sont les dispositifs les plus utilisés dans les réseaux de capteurs.

— Les dispositifs à fonction réduite (Reduced Function Device), appelés RFD, possèdent des fonctions limitées, contrairement aux FFD. Les RFD ne peuvent être que des nœuds feuilles (en anglais End Devices) dans l'arborescence construite car les nœuds RFD ne communiquent pas entre eux, ils ne peuvent communiquer qu'avec des FFD afin d'acheminer leur trafic [MB16].

Selon les besoins des applications, ces dispositifs peuvent être déployés selon différentes topologies. Dans la topologie en étoile, tous les dispositifs sont regroupés à un saut autour du coordinateur de réseau et toutes les communications passent par le coordinateur. Au sein de la topologie paire à paire, le coordinateur du réseau peut ne pas être directement accessible à tous les équipements. La dernière topologie proposée par la norme est nommée cluster tree. Cette topologie est constituée de groupes d'équipements (clusters) reliés entre eux et permet d'étendre la couverture du réseau [Rot12]. La Figure 2.1 illustre ces différentes topologies.

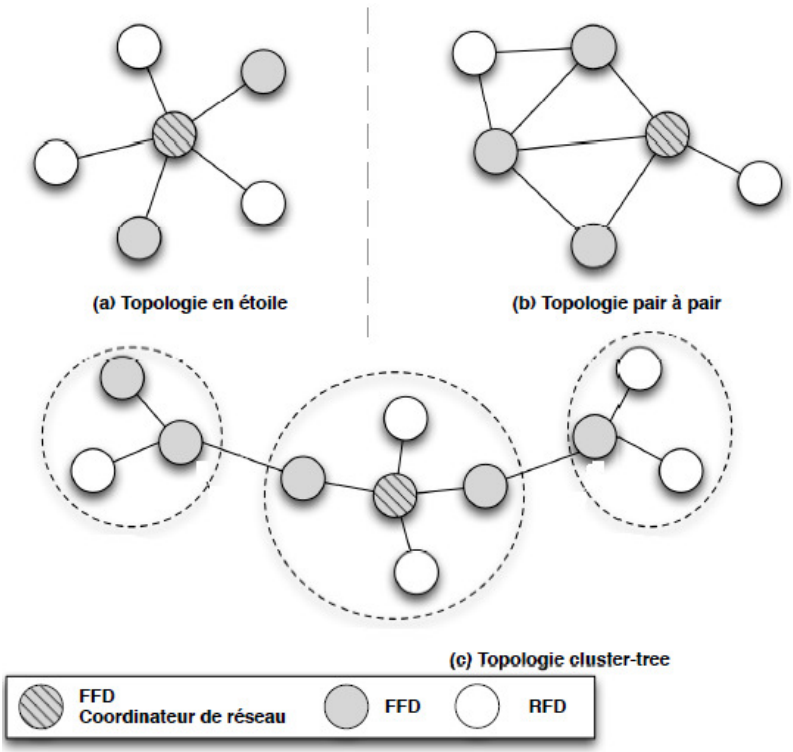


Figure 2.1 – Exemple de topologies [Rot12].

2.2.3 Format de la trame 802.15.4

La norme IEEE 802.15.4 est implémentée sur deux couches, la couche physique (PHY) et la couche contrôle d'accès au médium (MAC) tel qu'il est illustré dans les Figures 2.2 et 2.3. Chaque couche est responsable d'une partie de la norme et fournit des services aux autres couches. Nous rappellons que les deux couches en question sont décrites dans 1.6.2.

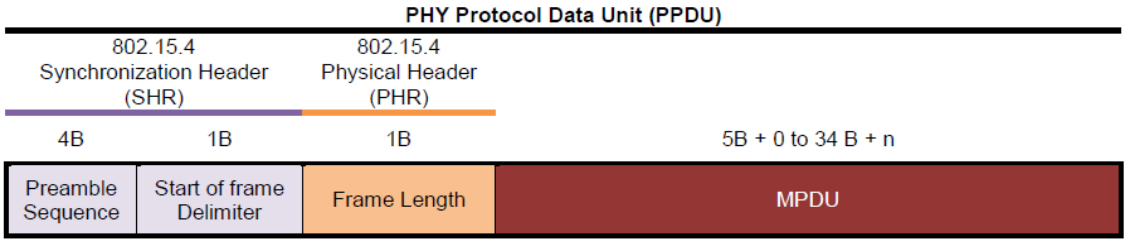


Figure 2.2 – Format d'une trame IEEE 802.15.4 (couche physique) [NR14].

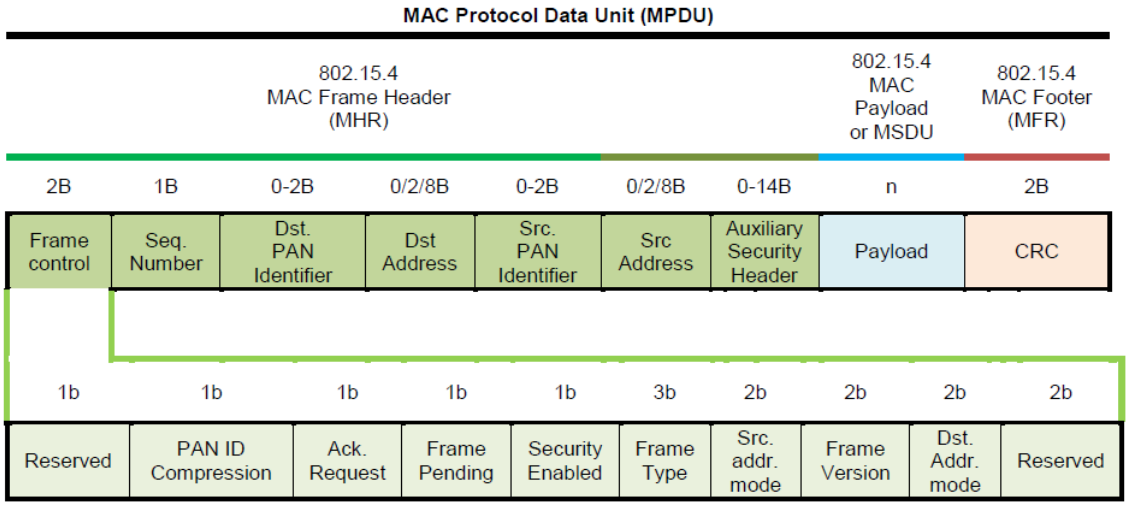


Figure 2.3 – Format d'une trame IEEE 802.15.4 (couche MAC) [NR14].

— B : Byte (Octet).
— b : bit.
— n : numbers (nombres).

2.3 Pourquoi IPv6 ?

2.3.1 Internet Protocol version 4

Adresse IPv4 : C'est la première version d'IP et la plus utilisée actuellement, sur 32 bits soit 4 octets. Elle est généralement représentée en notation décimale avec

quatre nombres, chacun compris entre 0 et 255, séparés par des points (voir exemple illustré dans la Figure 2.4) [BF].

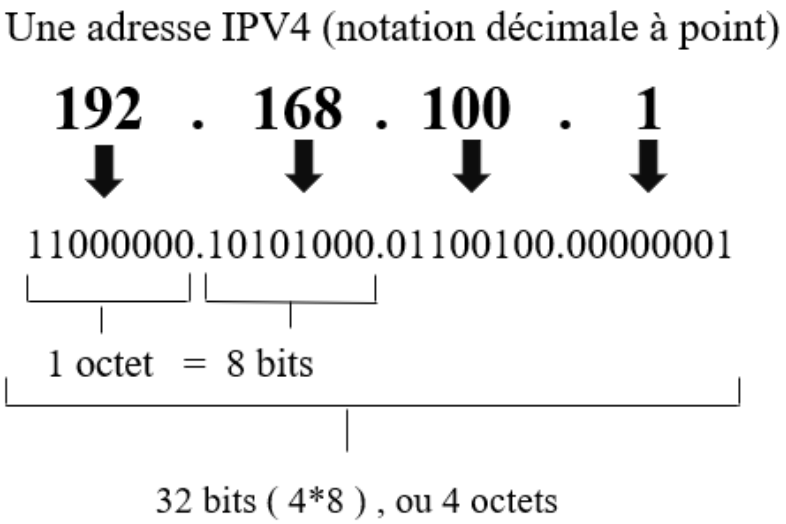


Figure 2.4 – Exemple d'une adresse IPv4.

Les problèmes posés par IPv4 [BF] :

— La taille de l'Internet double tous les 6 mois.
— La saturation de l'espace d'adressage.
— Explosion de la taille des tables de routage.
— Qualité de service insuffisante pour les nouvelles applications multimédia.
— Sécurité insuffisante (ex : chiffrement).

2.3.2 Passage de l'IPv4 à l'IPv6

Avec la croissance actuelle d'Internet et la volonté d'y connecter les objets de notre quotidien (l'Internet des Objets), le nombre d'adresses IPv4 disponibles s'épuise très rapidement. Par rapport à IPv4, IPv6 multiplie par 4 la taille des adresses, passant ainsi de 32 bits à 128 bits. Cela permettra d'obtenir $3.4 * 10^{28}$ adresses et de s'affranchir ainsi du mécanisme Network Address Translation (NAT) qui regroupe derrière une adresse IP publique plusieurs équipements mais de ce fait complexifie le fonctionnement d'applications.

Le passage au protocole IPv6 a permis de simplifier le format de l'en-tête IPv6 (retrait des champs qui ne sont plus utilisés). Il dispose désormais d'une taille fixe : les options d'IPv4 sont devenues des extensions pour IPv6 [Rot12].

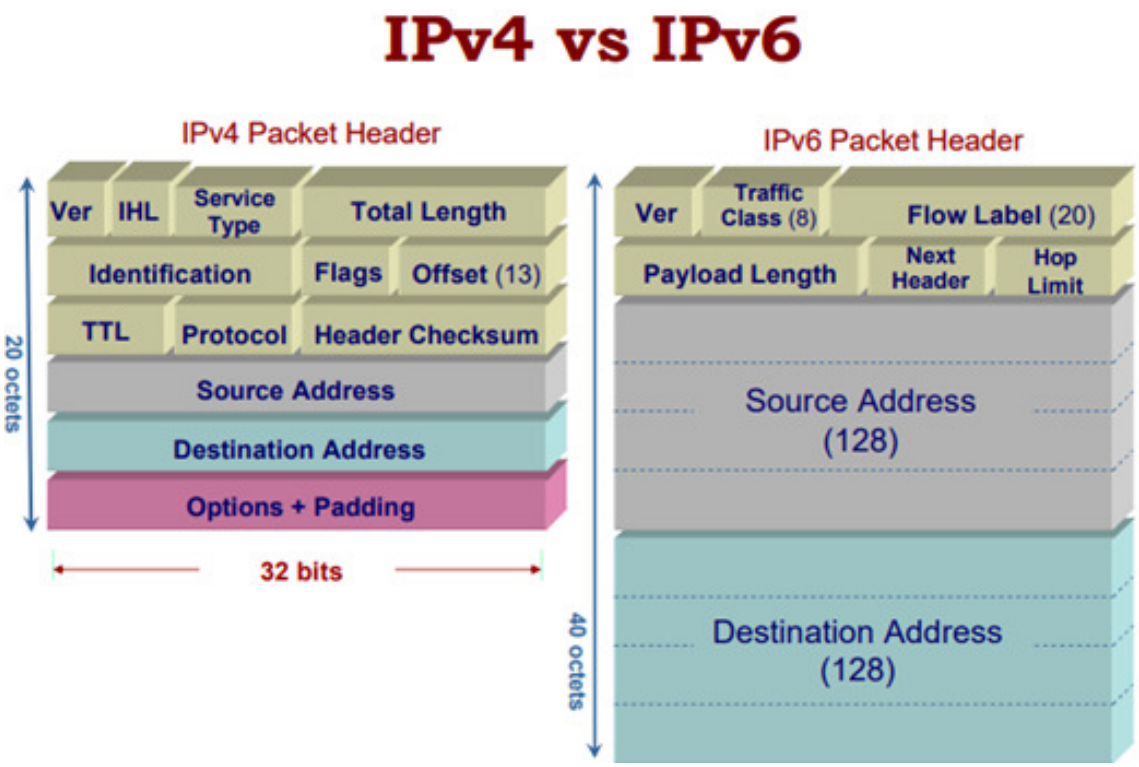


Figure 2.5 – IPv4 vs IPv6 [BF].

2.3.3 Le Protocole IPv6

IPv6, en Anglais “Internet Protocol version 6”, est le successeur du protocole IPv4 (quoi il y a eu la version 5 -IPv5- qui n’a pas vu le jour). Il est considéré comme le moteur de l’IdO. Il permet en effet d’affecter une adresse IPv6 unique à chaque objet connecté.

Ce nouveau protocole a été standardisé pour atteindre les objectifs suivants :

— Augmenter l’espace d’adressage (de 32 bits à 128 bits).

— Eliminer les problèmes d’IPv4 déjà cités.

— Prendre en charge les problèmes de la mobilité et de la sécurité.

— Favoriser l’émergence d’un monde connecté.

Adresses IPv6

Le protocole IPv6 fait passer la taille d’une adresse de 32 bits (la version 4) à 128 bits. le nombre d’adresses disponibles passe par conséquent de 2^{32} à 2^{128}. Une adresses IPv6 est représentée en notation hexadécimale. Elle se compose de 8 mots de 16 bits, séparés par une colonne “ :”.

Par exemple :

2001 :0FB8 :0000 :0000 :210F :A9D0 :0C61 :8001

Dans un mot, il n’est pas nécessaire d’écrire les zéros placés en tête :

2001 :0FB8 :0000 :0000 :210F :A9D0 :0C61 :8001

L'adresse précédente peut alors s'écrire :

2001 :FB8 :0 :0 :210F :A9D0 :C61 :8001

En outre plusieurs mots nuls consécutifs peuvent être remplacés par " : :".
L'adresse précédente peut s'écrire comme suit :

2001 :FB8 : :210F :A9D0 :C61 :8001

L'abréviation " : :" ne peut apparaître qu'une fois, au plus, dans une adresse IPv6, sinon le nombre de groupes remplis de zéros compressés serait ambigu.

Une adresse IPv6 selon les procédures CIDR (Classless Inter-Domain Routing) sont divisées en deux parties une partie hôte et une partie réseau [Iva17]. Les 64 premiers bits de l'adresse IPv6 (préfixe) servent généralement à l'adresse de sous-réseau, tandis que les 64 bits suivants identifient l'hôte à l'intérieur du sous-réseau. Un préfixe IPv6 est donc représenté par la notation suivante :

adresse-IPv6/longueur-du-préfixe-en-bits.

Les formes représentées par " : :" sont autorisées :

2001 :0DB8 :76FF :A510 :0000 :0000 :0000 :0000/64

2001 :DB8 :76FF :A510 :0 :0 :0 :0/64

2001 :0DB8 :76FF :A510 : :/64

La Figure 2.6 montre un exemple d'adresse IPv6 :

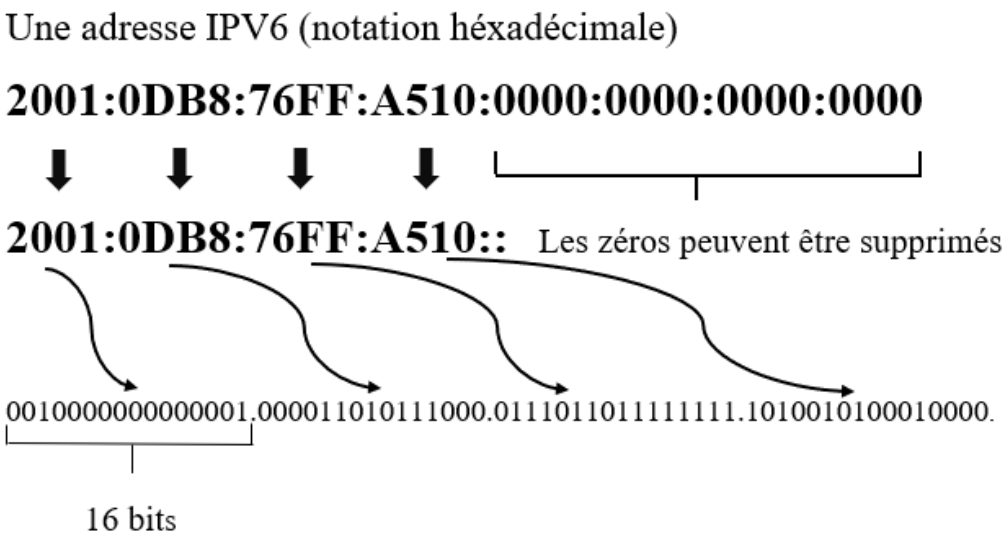


Figure 2.6 – Exemple d'une adresse IPv6.

Type d'adresses IPv6

Une adresse IPv6 peut être divisée en trois catégories : unicast, multicast et anycast [Iva17]. Le type d'adresse définit la cardinalité de la communication, c-à-d., à combien de destinataires doit être remis le paquet.

- **Unicast :** C'est le type le plus simple. Une adresse Unicast désigne une interface unique. Les paquets envoyés à une adresse IPv6 de type unicast sont livrés à une seule destination. Les adresses IPv6 Unicast peuvent être subdivisées en adresses Link-Local, Site/Unique Local et Global.
  - **Adresse Link-Local (FE80 : :/10) :** accessible uniquement dans le sous-réseau local. Equivalent à une adresse Automatic Private IP Addressing (APIPA) en IPv4.
  - **Site Local (FEC0 : :/10) :** Equivalente à une adresse privée en IPv4. Toutefois, elle n'est pas unique.
  - **Unique Local (FD00 : :/8 ou FC00 : :/7) :** Equivalente à une adresse privée en IPv4.
  - **Global Unicast (2000 : :/3) :** identiques aux adresses publiques IPv4. Elles sont uniques au monde et dont utilisées pour envoyer un paquet d'un sous-réseau à n'importe quelle destination sur Internet.
- **Multicast :** Une adresse multicast est équivalente à une adresse de type broadcast (on notera que le mode broadcast n'existe pas en IPv6. C'est en effet un cas particuleir du multicast). Elle désigne un groupe d'interfaces qui en général appartiennent à des nœuds différents pouvant être situés n'importe où dans l'Internet. Les adresses de ce type commencent toujours par le préfixe

FF00 : :/8. Les paquets en mode multicast sont envoyés d'un hôte unique à plusieurs destinataires.

— **Anycast :** Anycast est un nouveau type d'adressage. Les adresses Anycast sont utilisées pour transmettre des paquets de données d'une source à la destination la plus proche parmis un groupe d'interfaces. Elle n'a pas de préfixe spécial.

La figure 2.7 ci-dessous montre une représentation graphique des méthodes de routage unicast, anycast et multicast.

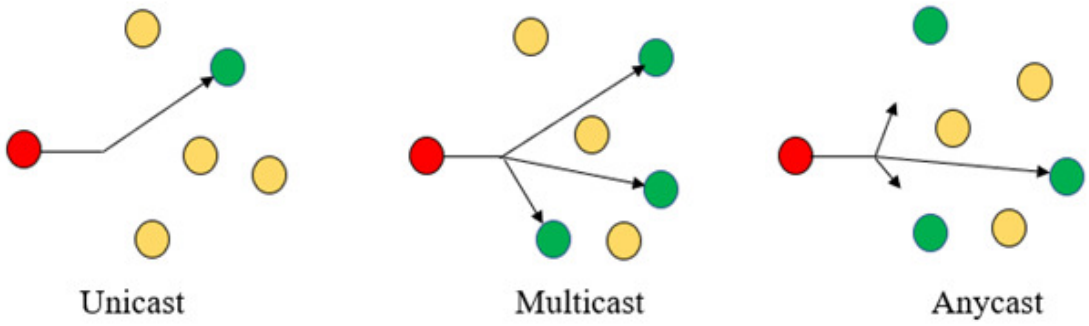


Figure 2.7 – Les méthodes de routage : unicast, anycast et multicast.

L'en-tête IPV6

Pour mieux comprendre comment 6LoWPAN utilise IPv6, quelques détails concernant l'en-tête IPv6 doivent être étudiés. Un paquet IPv6 comprends deux parties : l'en-tête et la charge utile, avec un Maximum Transmission Unit (MTU) de 1280 octets. L'en-tête des paquets IPv6 est simple (moins de champs par rapport à l'en-tête du protocole IPv4) et la taille de l'en-tête IPv6 est fixée à 40 octets, c'est donc facile pour le processus de traitement.

Les champs de l'en-tête sont [Iva17] :

— Le champ Version (4 bits) : version du protocole IP utilisée et vaut toujours 6.

— Le champ classe de trafic (8 bits) : identifie le type de contenu encapsulé dans le paquet IPv6.

— Etiquette de flux (20 bits) : pour faire de la qualite de service et affecter le traitement des paquets dans les routeurs.

— Longueur de la charge utile (16 bits) : Taille en octet des données transportées a partir de ce champ.

— En-tête suivant (8 bits) : indique la longueur des données suivantes.

— Nombre de sauts (8 bits) : définit le nombre maximum de sauts que l'en-tête doit traverser, allant d'un nœud de réseau à un autre.
— Adresses source et de destination : adresse longue de 128 bits de la source et de la destination du paquet.
— Une en-tête d'Extensions ou la charge des données utiles.

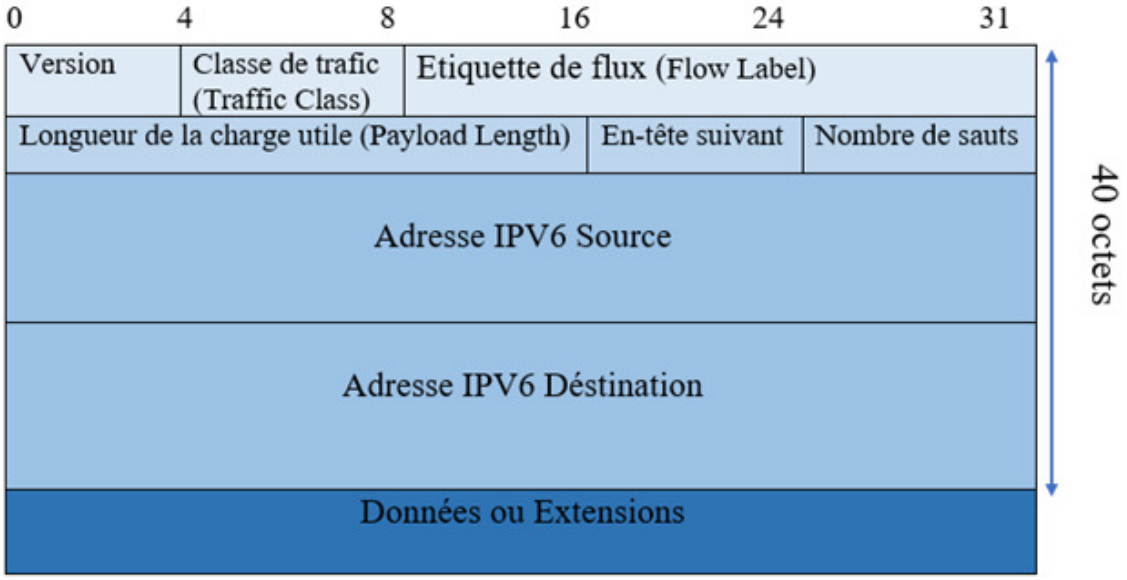


Figure 2.8 – En-tête IPv6.

2.4 Les réseaux 6LoWPAN

6LoWPAN, (IPv6 over Low-power Wireless Personal Area Networks en anglais), est le nom d'un groupe de travail créé par l'IETF [Wik19], afin de développer un réseau de communication simple, à bas coût pour des systèmes embarqués à puissance limitée.

6LoWPAN est apparu pour résoudre un problème d'interconnexion de réseaux de capteurs sans fil avec IPv6 dans le but de relier n'importe quel appareil à l'internet. C'est une technologie, ou une couche d'adaptation, qui permet aux paquets IPv6 d'être transportés d'une façon efficace dans les couches de liaison définies par la norme IEEE 802.15.4, ce qui nécessite des mécanismes de compression et de fragmentation de données pour rendre cette adaptation possible.

2.4.1 Architecture des réseaux 6LoWPAN

Un réseau 6LoWPAN est constitué de nœuds partageant le même préfixe IPv6 et de routeur de bordure qui s'occupe de la gestion de la compression et la fragmentation d'en-têtes IPv6 et qui gère ces trois actions :

— L'échange de données entre les périphériques 6LoWPAN et Internet (ou un autre réseau IPv6).

— L'échange local de données entre les périphériques du 6LoWPAN.

— La génération et la maintenance du sous-réseau radio (le réseau 6LoWPAN).

On distingue trois familles d'architecture dans un réseau 6LoWPAN qui sont illustrées dans la Figure 2.9.

— Un réseau 6LoWPAN simple avec un seul routeur de bordure (Edge router) connecté à d'autres réseaux IP.

— Un réseau 6LoWPAN étendu avec plusieurs routeurs de bordure reliés par une dorsale.

— Un réseau ad-hoc 6LoWPAN non connecté.

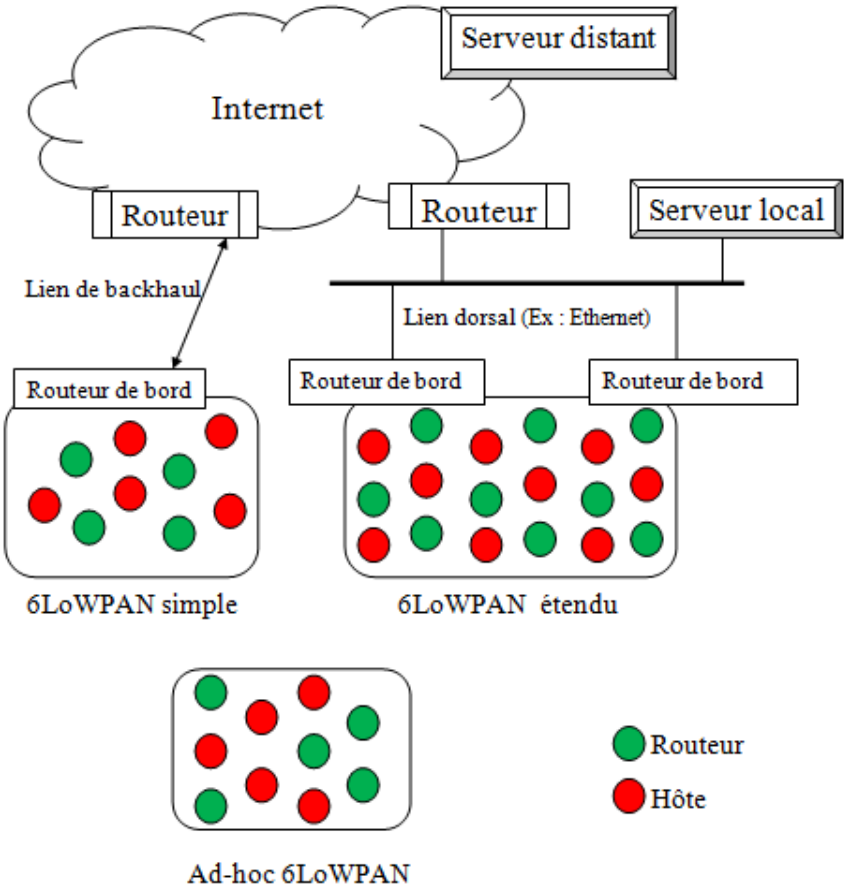


Figure 2.9 – Architecture des réseaux 6LoWPAN[SB11].

2.4.2 Les piles de protocoles IPv6 et 6LoWPAN

La Figure 2.10 illustre la pile de protocoles IPv6 et 6LoWPAN. Une pile de protocoles IPv6 simple avec 6LoWPAN (également appelée pile de protocole 6LoWPAN) est presque identique à une pile IP normale avec les différences suivantes :

Le 6LoWPAN ne supporte que l'IPv6, pour lequel une petite couche d'adaptation (appelée couche d'adaptation LoWPAN) a été définie pour optimiser IPv6 par rapport à IEEE 802.15.4 et des couches de liaison. Au-dessus de la couche d'adaptation, il existe une couche réseau qui assure le routage des paquets. Elle utilise un protocole de réseau IP (Internet Protocole), pour fournir des adresses aux nœuds. La communication entre ces nœuds est sous la responsabilité de la couche de transport. Le protocole de transport le plus couramment utilisé avec 6LoWPAN est le protocole User Datagram Protocol (UDP), qui peut également être compressé au format LoWPAN. Le protocole de contrôle de transmission Transmission Control Protocol (TCP) n'est pas couramment utilisé avec 6LoWPAN pour des raisons de performances, d'efficacité et de complexité. Le protocole de messages de contrôles Internet v6 Internet Control Message Protocol (ICMPv6 est utilisé pour la messagerie de contrôle, par exemple les messages Internet Control Message Protocol (ICMP) echo, ICMP destination inaccessible et Neighbor Discovery. Les protocoles d'application sont souvent spécifiques à l'application et au format binaire, bien que de plus en plus de protocoles d'application standard deviennent disponibles [SB11].

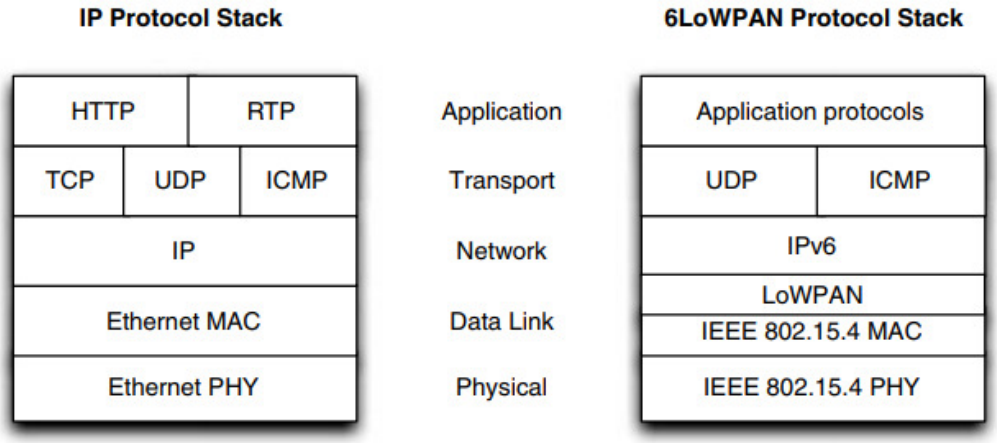


Figure 2.10 – Comparaison des piles de protocoles IP et 6LoWPAN [SB11].

L'adaptation entre le format IPv6 complet et le format LoWPAN est effectuée par les routeurs situés à la frontière des ilots 6LoWPAN, appelés routeurs de bordure. Cette transformation est transparente, efficace et sans état dans les deux sens.

L'adaptation LoWPAN dans un routeur de bordure est généralement effectuée dans le cadre du pilote d'interface réseau 6LoWPAN et est généralement transpa-

rente pour la pile de protocoles IPv6 elle-même. La Figure 2.11 illustre une réalisation d'un routeur de bordue prenant en charge 6LoWPAN. A l'intérieur de LoWPAN, les hôtes et les routeurs n'ont pas besoin de travailler avec des formats d'en-tête IPv6 ou UDP complets à un moment donné, car tous les champs compressés sont connus implicitement par chaque nœud [SB11].

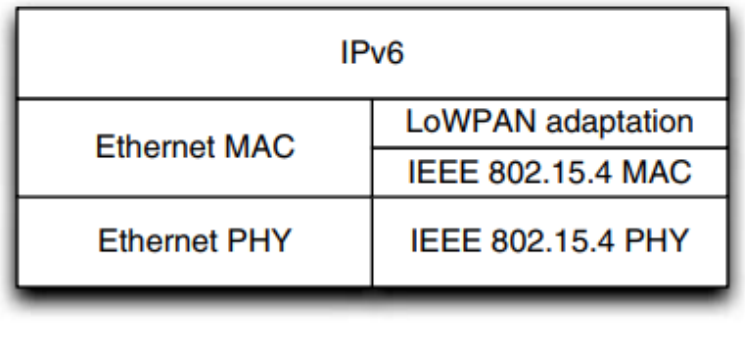


Figure 2.11 – Routeur de bordue IPv6 avec prise en charge 6LoWPAN [SB11].

2.4.3 Couche d'adaptation 6LoWPAN

6LoWPAN est une couche d'adaptation de niveau "2.5" située entre la couche réseau et la couche liaison de données (Figure 2.12) chargée de réduire la taille des paquets IPv6 en utilisant la fragmentation et la compression.

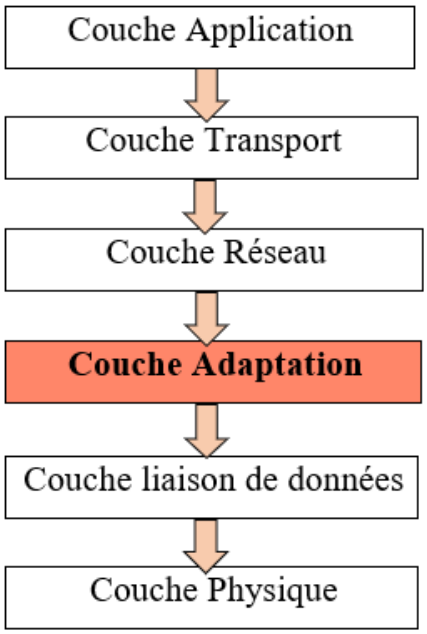


Figure 2.12 – La pile de protocoles de 6LoWPAN.

Les scénarios de compréssion 6LoWPAN

On peut distinguer trois scénarios de communication possibles afin d'illustrer l'intérêt d'un tel mécanisme [AC17] :

- Premier scénario : La communication entre deux périphériques à l'intérieur du même réseau 6LoWPAN, en utilisant des adresses locales de liaison (link-local addresses), l'en-tête IPv6 peut être compressée à seulement 2 octets. C'est le scénario du meilleur cas.
- Second scénario : La communication destinée à un périphérique en dehors du réseau 6LoWPAN et le préfixe pour le réseau externe est connu, où l'en-tête IPv6 peut être compressé à 12 octets.
- Troisième scénario : Similaire au second scénario, mais cette fois sans connaître le préfixe du périphérique externe, ce qui donne un en-tête IPv6 de 20 octets scénario du pire cas.

On constate que même dans le pire cas, on arrive quand même à réduire la taille de l'en-tête IPv6 de moitié. Si malgré ce mécanisme la taille du datagramme reste supérieure à celui des trames, un autre mécanisme est proposé celui de la fragmentation.

2.5 Routage dans 6LoWPAN

Pour acheminer les paquets vers leur destination, les réseaux 6LoWPAN utilisent un mode de routage. A la différence des réseaux traditionnels dans lequel le routage est effectué uniquement au niveau 3, il peut être effectué à deux niveaux différents. Le routage dans le 6LoWPAN est réalisé selon deux modes : mesh-under et route-over.

2.5.1 Mush-under

Dans le Mesh-under, la couche réseau n'effectue aucun routage IP à l'intérieur d'un LoWPAN. La couche d'adaptation exécute le relayage de noeud en noeud sans solliciter la couche IP et achemine les paquets vers la destination sur plusieurs sauts radio donc la décision de routage se fait au niveau de la couche d'adaptation. Si un paquet IP est fragmenté par la couche d'adaptation les fragments sont livrés au saut suivant par routage maillé. Différents fragments d'un paquet IP peuvent passer par différents chemins et ils sont rassemblés à la destination. Dans ce cas, le paquet IPv6 n'est reconstitué que sur l'équipement destinataire. Si tous les fragments sont atteints avec succès, alors la couche d'adaptation du noeud de destination rassemble tous les fragments et crée un paquet IP. Dans le cas où un fragment quelconque

manque, tous les fragments pour ce paquet IP sont retransmis vers la destination [AC17].

2.5.2 Route-over

Dans le mode route-over, toutes les décisions de routage sont prises dans la couche réseau où chaque noeud agit en tant que routeur IP. Lorsque'un paquet IP est fragmenté par la couche d'adaptation, les fragments sont envoyés au saut suivant sur la base des informations de la table de routage. Si tous les fragments sont reçus avec succès, la couche d'adaptation crée un paquet IP à partir de fragments et l'envoie à la couche réseau. Si le paquet est destiné à lui-même, la couche réseau envoie le paquet IP à la couche de transport, sinon envoie le paquet au saut suivant sur la base des informations de la table de routage. Dans ce mode de routage, les paquets sont reconstitué sur chaque équipement intermédiaire afin de prendre la décision de routage s'il y a un ou plusieurs fragments manquants, tous les fragments sont retransmis à une distance de houblon. Route-over est plus efficace dans des conditions dégradées (perte de paquets) [AC17].

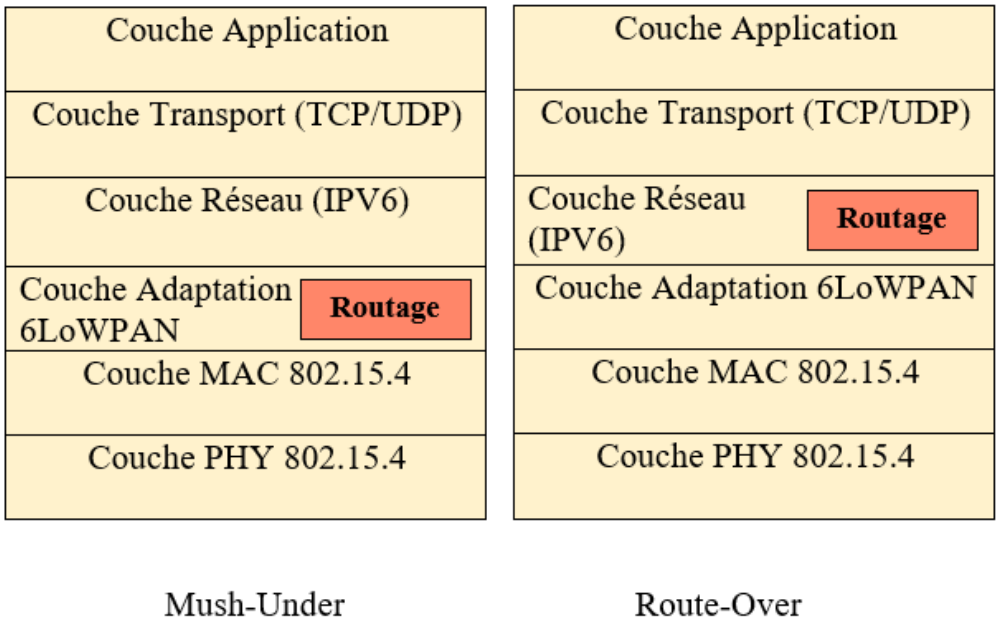


Figure 2.13 – Schéma de routage 6LoWPAN.

2.6 Les protocoles de routage

De nombreux protocoles de routage ont été développés par la communauté 6LoWPAN, tel que LOAD, DYMO-LOW, HI-LOW. Mais aujourd'hui, il n'y a qu'un protocole qui est le plus utilisé pour de larges déploiements, à savoir Routing Protocol for low power and Lossy Networks (RPL).

2.6.1 Protocole RPL

RPL (Routing Protocol for low power and Lossy network) est un protocole de routage à vecteur de distance basé sur IPv6, et est le plus utilisé dans les réseaux 6LoWPAN. Il a été développé par le groupe de travail IETF ROLL (Internet Engineerg Task Force) (Routing Over Low-power and Lossy) pour répondre aux besoins des réseaux à faible consommation d'énergie et à perte. Le protocole RPL est implémenté au niveau routage (route over). Il est donc défini au niveau de la couche réseau dans l'architecture IP [MB16]. La topologie RPL est organisée sous la forme d'un graphe acyclique dirigé appelé DAG (Directed Acyclic Graph) composé d'un ou plusieurs DAGs acycliques orientés vers la destination qu'on appelle DODAG (Direction-Oriented DAG). Chaque racine du réseau possède donc un DODAG qui est le collecteur de données du graphe (le sink).

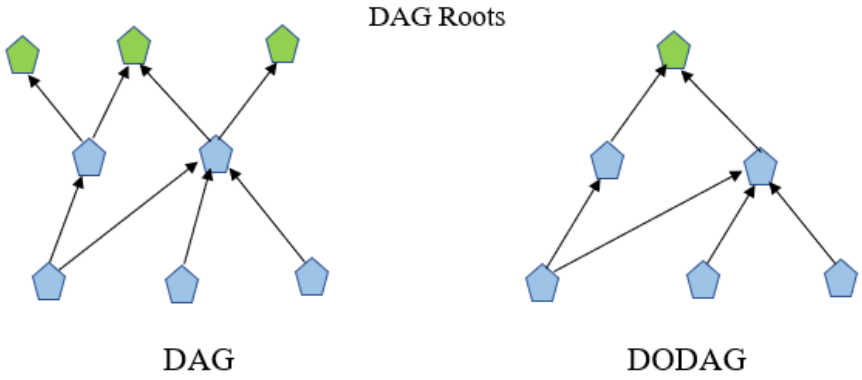


Figure 2.14 – graphe acyclique dirigé (DAG) et DAGs acycliques orientés vers la destination (DODAG).

L'un des principaux avantages du protocole RPL est qu'il supporte plusieurs types de trafics réseau : le trafic multi-points à point, le trafic point à multi-points et le trafic point à point. Le protocole RPL répond ainsi aux exigences du transfert d'informations dans les réseaux de capteurs d'aujourd'hui [MB16].

2.7 Conclusion

Dans ce deuxième chapitre nous avons présenté la technologie 6loWPAN, la solution idéale pour les applications de l'internet des objets. Nous avons d'abord commencé par une description générale de la norme IEEE 802.15.4, son fonctionnement et le format de ses trames. Ensuite, nous avons détaillé le concept des adresses IPv4 et IPv6. La fin du Chapitre a été consacrée à la description des réseaux 6loWPAN et leurs propres protocoles de routage qui ont des avantages et des inconvénients selon l'application. Dans le prochain Chapitre, nous allons décrire la réalisation de notre plate-forme matérielle qui permet de connecter un RCSF à Internet.

Chapitre 3

Réalisation de la plate-forme d'échange RCSF-Internet

3.1 Introduction

Les nœuds capteurs de l'IdO communiquent avec internet via le routeur frontière 6LowPAN Border Router (6LBR). Il s'agit d'un routeur IPv6 à part entière, interconnectant deux sous-réseaux IPv6, à savoir le sous-réseau RCSF et le sous-réseau Ethernet ou WIFI.

Ce chapitre comprend deux parties. Dans la première partie, nous décrivons le 6LBR et ses modes de fonctionement, le protocole Message Queue Telemetry Transport (MQTT) et le protocole Serial Line Internet Protocol (SLIP). Dans la deuxième partie, nous présentons tout d'abord l'environnement de travail ainsi que les matériel utilisé pour réaliser notre plate-forme d'échange entre un RCSF et un broker MQTT installé sur une carte Raspberry Pi 3 . Ensuite, nous expliquons les étapes d'installation d'implémentation des différents outils et programmes sur le matériel utilisé. Nous clôturons le chapitre par la description de la simulation de notre plate-forme pratique, et ce raison d'un problème rencontré lors de l'installation du 6LBR sur la RPi.

3.2 Le protocole Message Queue Telemetry Transport (MQTT)

Pour répondre à la problématique du nombre grandissant d'objets connectés qui sont entrain de faire leur apparition sur internet ces dernières années (selon une étude Gartner : c'est près de 26 milliards d'objets connectés qui seront sur Internet d'ici 2020), l'internet des objets s'est doté d'un nouveau standard, à savoir le protocole Message Queuing Telemetry Transport **(MQTT)** [Bon15].

MQTT a été inventé en 1999 par des chercheurs d'IBM qui cherchaient à avoir un protocole de messagerie léger pour faire communiquer des machines dans un environnement où les déconnexions sont fréquentes, à l'image des déconnexions des nœuds capteurs dans un RCSF et où il faut quand meme pouvoir recevoir les informations lors de la reconnexion [Kas17]. MQTT est un protocole léger, facile à comprendre, très souple et sécurisé. C'est un service de messagerie TCP/IP (pas plus de 256 Mo par message). Il est idéal pour répondre aux besoins suivants [Bon15] :

- Particulièrement adapté pour utiliser une très faible bande passante.
- Idéal pour l'utilisation sur les réseaux sans fil.
- Faible consommateur en énergie.
- Très rapide, il permet un temps de réponse supérieur aux autres standards du web actuel.
- Permet une forte fiabilité si nécessaire.
- Nécessite peu de resources en termes de processeur et de mémoire.

Principe de fonctionnement

Contrairement au principe du client/serveur utilisé sur le Web (HTTP), MQTT utilise celui de la publication/souscription : plusieurs clients se connectent à un serveur unique (appelé broker) pour soit publier des informations, soit souscrire à leur réception. Les "publisher" et "subscriber" sont des "clients" [Kas17].

Les messages sont envoyés par les publishers sur un canal appelé "topic" à un "broker". Ces messages peuvent être lus par les subscribers (abonnés).

Les topics (ou canaux d'informations) peuvent avoir une hiérarchie qui permet de sélectionner précisemment les informations auquelles on veut s'abonner [Bon15].

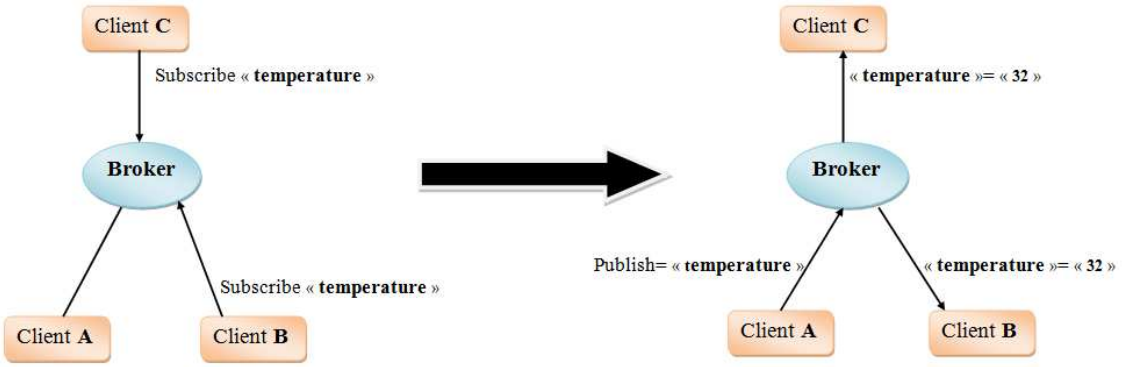


Figure 3.1 – Principe de fonctionnement du protocole MQTT.

— Les trois clients établissent une connexion TCP avec le broker. Les clients B et C souscrivent au topic **temperature**.
— Le Client A publie sur le topic **temperature** une valeur de 32 degrés. Le broker propage le message à tous les clients ayant préalablement souscrit au topic t**emperature**.

Les topics

Les clients de souscription MQTT s'enregistrent auprés du broker sur des topics, des sortes de chemins d'accès à une ressource. Ils demandent ainsi à être notifiés lorsque quelqu'un publie sur ces topics [Kas17].

Cela peut être un topic de température par exemple : /sensor/1/temperature.

On peut souscrire à un ensemble de topics en utilisant des wildcards # ou +.

Par exemple, si un client publie sur les topics /sensor/1/temperature et /sensor/1/humidity, un autre client peut écouter ces deux topics à la fois : /sensor/1/#.

Si plusieurs clients publient leurs températures et humidités en intercalant leur numéro de client sur leur topic, un autre client peut écouter toutes les températures ainsi : /sensor/+/temperature. Il recevra alors les températures du client 1 (/sensor/1/temperature), du client 2 (/sensor/2/temperature), etc.

Qualité de service (QoS)

MQTT intègre en natif la notion de QoS. En effet, le publisher a la possibilité de définir la qualité de son message.
Trois niveaux sont possibles [Bon15] :

— Un message de **QoS niveau 0** "At most once " sera délivré tout au plus une fois. Ce qui signifie que le message est envoyé sans garantie de réception, le broker n'informe pas l'expéditeur qu'il a reçu le message.

— Un message de **QoS niveau 1** "At least once" sera livré au moins une fois. Le client transmettra plusieurs fois s'il le faut jusqu'à ce que le Broker lui confirme qu'il a été transmis sur le réseau.
— Un message de **QoS niveau 2** "exactly once " sera obligatoirement sauvegardé par l'émetteur qui le transmettra toujours tant que le récepteur ne confirme pas son envoi sur le réseau. La principale différence étant que l'émetteur utilise une phase de reconnaissance plus sophistiquée avec le broker pour éviter une duplication des messages (plus lent mais plus sûr).

3.3 Le protocole SLIP

Serial Line Internet Protocol (SLIP) est un protocole de liaison internet en série. Il est considéré comme un protocole simple d'encapsulation des datagrammes IP sur des liaisons series. Il définit une séquence de caractères qui encapsulent des paquets IP sur une ligne série. Il n'assure que la delimitation des trames, ne fournissant aucun contrôle d'erreur ou d'adresse [Fra19].

La transmission de données avec SLIP est trés simple. Ce protocole envoie une trame composée uniquement des données à envoyer suivies des caractères spéciaux pour indiquer qu'une série d'octets devraient être regroupés pour former un datagramme.

3.4 Routeur de bordure (6LBR)

La passerelle peut être implémentée comme un routeur de bordure assurant la liaison entre le RCSF et Internet. Il connecte deux technologies de liaisons de données différentes : typiquement, l'IEEE 802.15.4 du côté RCSF et la technologie Ethernet ou l'IEEE 802.11 du côté Internet. Une passerelle simple mais puissante a été développée récemment, permettant de faire transiter les communications du domaine filaire (Ethernet) au RCSF (6LoWPAN), et vice versa. Nous nous en servons comme base dans la construction de notre plate-forme d'interconnexion dynamique entre le RCSF et Internet. Cette passerelle est désignée sous le nom de 6LoWPAN Border Router (6LBR) et elle peut s'exécuter sur un matériel embarqué ou toute plate-forme disposant d'un système d'exploitation Linux. Trois principaux modes de fonctionnement sont possibles pour le 6LBR. Ils sont détaillés ci-dessous [Kam17].

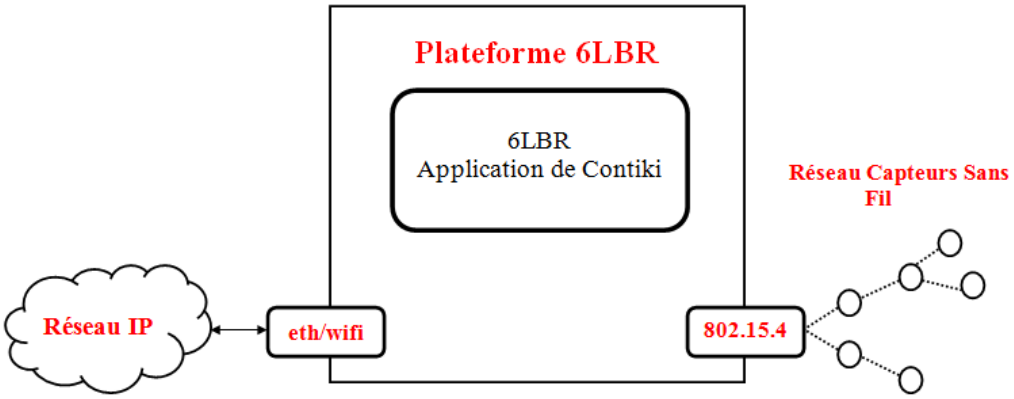


Figure 3.2 – 6LoWPAN Border Router (6LBR).

3.4.1 Mode routeur

Dans le mode routeur, les sous-réseaux Ethernet et WPAN sont logiquement séparés, le 6LBR fournissant la route entre les deux. Chaque interface a un préfixe réseau distinct, comme on peut le voir sur la Figure 3.3, où celui-ci est : a.a.a.a : :/64 sur Ethernet et b.b.b.b : :/64 pour le réseau WPAN. Les deux domaines de diffusion sont parfaitement isolés l'un de l'autre. Pour assurer la communication entre les deux domaines, les nœuds Ethernet devront être configurés pour acheminer le trafic destiné au RCSF à travers le 6LBR. Du côté du RCSF, c'est la racine RPL qui sera chargée de router le trafic destiné au réseau Ethernet via le 6LBR [Kam17].

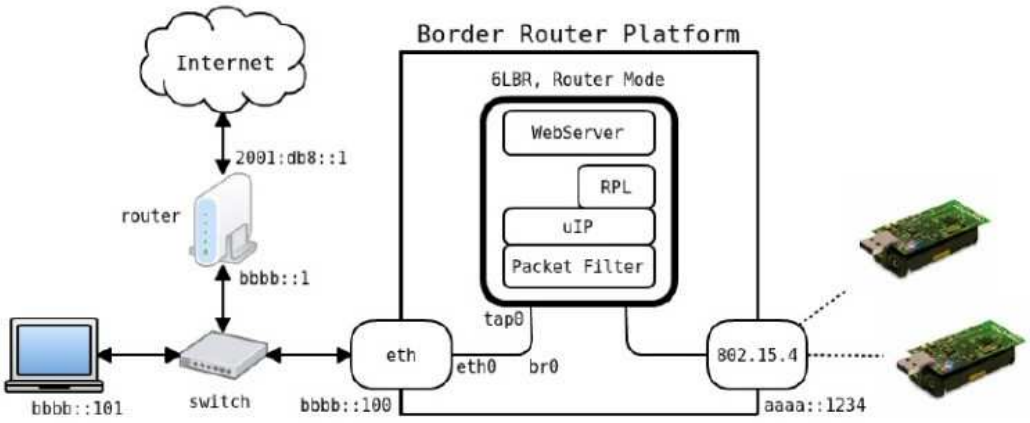


Figure 3.3 – Mode de fonctionnement routeur du 6LBR [Kam17].

3.4.2 Mode pont transparent

Ce mode assure une fonction de commutation de base permettant ainsi à plusieurs RCSF d'être agrégés en DODAG global, la racine RPL pouvant être exté-

rieure au RCSF. Toutes les trames de multidiffusion ou destinées à une interface IEEE 802.15.4 spécifique sont commutées de l'interface Ethernet vers le RCSF. Inversement, celles destinées à un hôte Ethernet spécifique ou celles constituant une adresse de multidiffusion sont commutées du RCSF par le 6LBR vers le segment Ethernet. L'ensemble du réseau constitue un seul domaine de diffusion et tous les nœuds partagent le même préfixe réseau (comme indiqué sur la Figure 3.4). Le 6LBR dispose de sa propre adresse physique dans chacun des segments réseaux et traduit les adresses physiques/trames Ethernet en adresses/trames IEEE 802.15.4 et vice versa [Kam17].

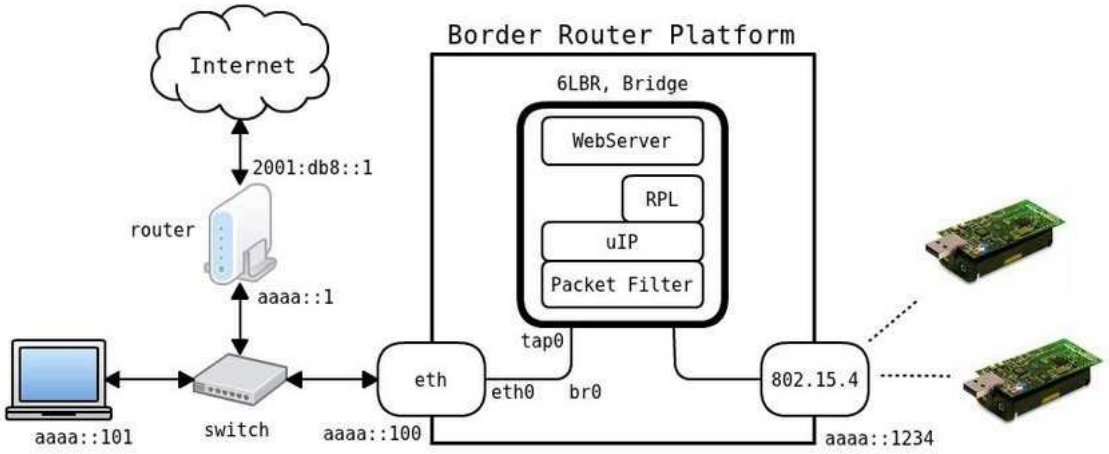


Figure 3.4 – Mode de fonctionnement Pont Transparent du 6LBR [Kam17].

3.4.3 Mode pont intelligent

Dans ce mode, 6LBR va agir comme un pont intelligent permettant d'interconnecter un réseau standard basé sur IPv6 avec un réseau de capteurs sans fil en maillage basé sur 6LoWPAN. Les nœuds du réseau de capteurs sont vus comme s'ils appartenaient au réseau IPv6 préexistant. C'est le mode à privilégier si on souhaite intégrer nos capteurs dans un réseau IPv6 [Kam17].

3.5 Environnement de travail et outils de dévlopement

3.5.1 Outils matériels

Il existe plusieurs modèles de capteurs commercialisés dans le marché. Les plus connus sont TelosB, MICAx. Dans notre projet, nous nous intéressons au modèle

TelosB montré dans la Figure 3.5.

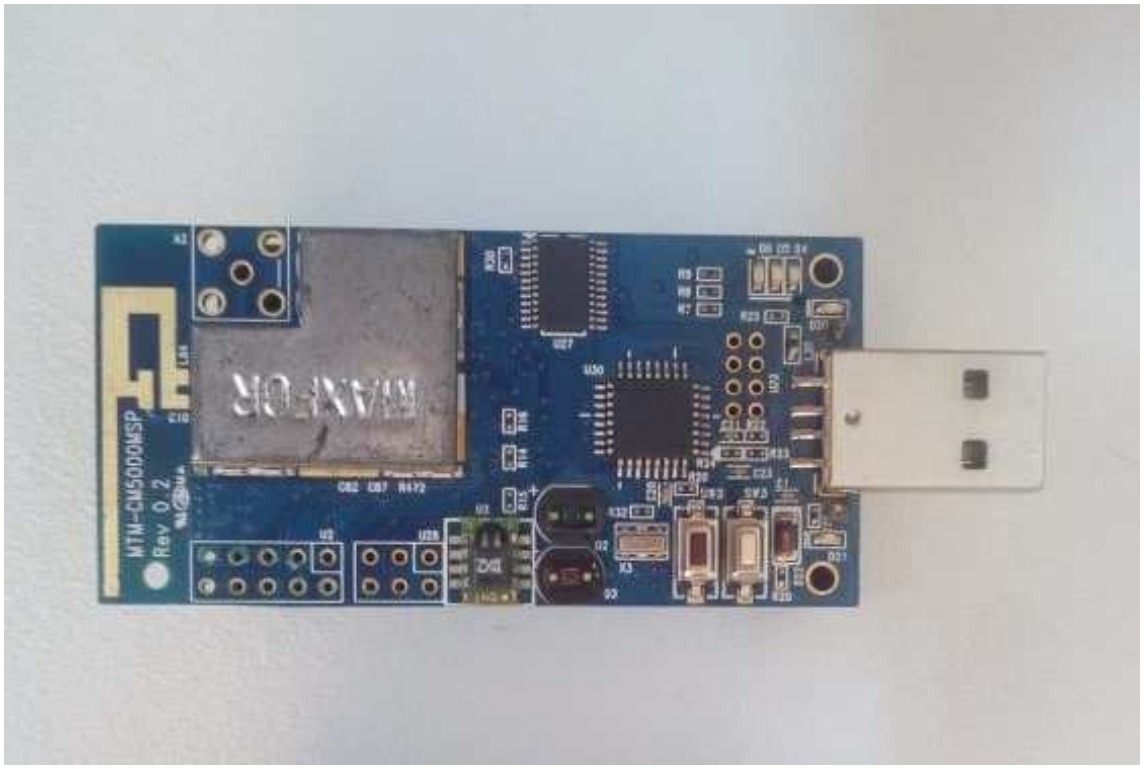


Figure 3.5 – Nœuds Capteurs TelosB.

TelosB

Les capteurs TelosB sont des nœuds de capteurs sans fil à très basse consommation, destinés aux applications IdO. Ils sont conformes à la norme IEEE 802.15.4 et sont composés principalement d'un processeur, une mémoire, un émetteur/récepteur radio, un ensemble de capteurs, et une batterie externe contenant deux piles AA. Le TelosB offre de nombreuses fonctionnalités, notamment :

- Emetteur-récepteur CC2420 RF conforme à IEEE 802.15.4.
- 2.4 à 2.4835 GHz, bande ISM compatible dans le monde entier.
- Débit de données de 250 kb/s.
- Antenne intégrée.
- Microcontrôleur TI MSP430 8 MHz avec 10 Ko de RAM.
- Faible consommation de courant.
- 1 Mo de mémoire flash externe pour l'enregistrement de données.
- Programmation et collecte de données via USB.
- Ensemble de capteurs comprenant un capteur intégré de lumière, de température et d'humidité.

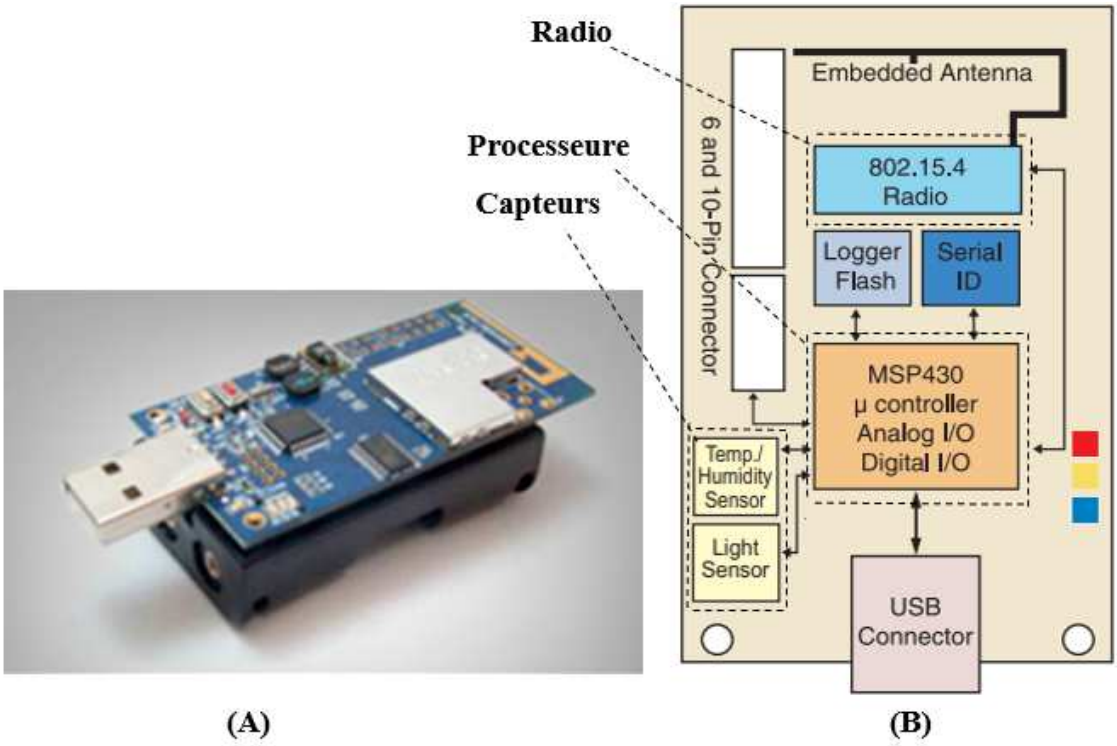


Figure 3.6 – (A) TelosB (B) Diagramme.

Raspberry Pi

Raspberry Pi [RPi] est basée sur l'architecture ARM. Cette carte gagne en popularité en raison de ses caractéristiques uniques et de ses capacités de haut niveau en termes de performances et de rapidité.

Le Raspberry Pi 3 Model B est la troisième génération de Raspberry Pi. Ce puissant ordinateur monocarte de taille réduite pouvant se connecter à un téléviseur, à un clavier et disposant d'une connectivité WiFi et Bluetooth. La version 3 est basée sur un processeur ARM Cortex-A53 64 bits quatre cœurs à 1,2 GHz (environ 10x plus rapide que la Pi 1 et 50% plus performante que le modèle Pi 2), et possède 1 GB de mémoire RAM [ele19].

Le modèle Raspberry Pi 3 B dispose d'une connectivité WiFi et Bluetooth, possède 4 ports USB, un port micro-SD, un connecteur d'E/S 40 broches et un port HDMI. Il peut effectuer des tâches d'un PC de bureau (feuilles de calcul, traitement de texte, jeux).

Caractéristiques [ele19] :

— CPU : Un processeur ARM Cortex-A53 64 bits quatre cœurs à 1,2 GHz.
— Mémoire : 1 Go de RAM LPDDR2.
— WiFi : 2,4 GHz, 802.11 b/g/n (Broadcom BCM43438).
— Bluetooth 4.1, 802.15 (Broadcom BCM43438).

— Alimentation : 5 Vcc/maxi 2.5 A via prise micro-USB (intensité max si toutes les fonctions sont utilisées).
— 4 ports USB2.0 avec une sortie jusqu'à 1.2A.
— En-tête GPIO étendu à 40 broches.
— Sortie vidéo / audio via un connecteur 3,5 mm 4 broches, HDMI, caméra CSI ou LCD brut (DSI).
— Stockage : Support pour cartes micro-SD.
— Port Ethernet 10/100 BaseT (RJ45).

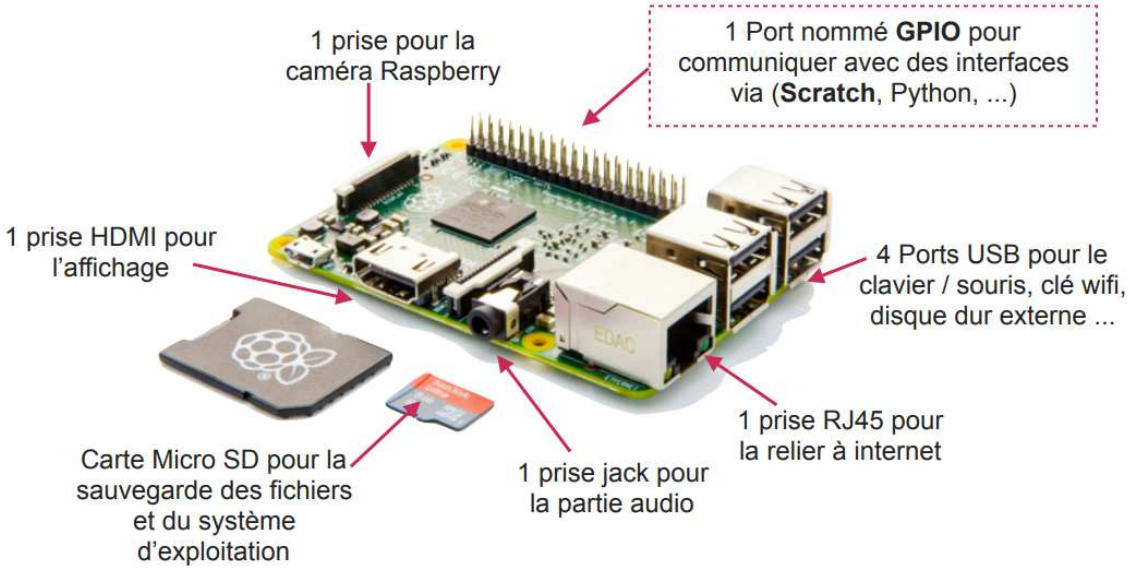


Figure 3.7 – Les composants de la carte raspberry [ele19].

3.5.2 Outils Logiciels

Contiki

Il existe plusieurs systèmes d'exploitation pour nœuds capteurs tel que Contiki OS, TinyOS, Mantis et autre, chaque système d'exploitation a ses propres fonctionnalités.

Contiki est un système d'exploitation open source léger et flexible pour l'internet des objets. Il est utilisé, d'une façon intense ses dernières années dans les RCSFs. Il a été en effet conçu pour les nœuds capteurs embarqués qui disposent de ressources limitées.

Contiki propose les principales caractéristiques et fonctionnalités d'un système d'exploitation en favorisant une consommation énergétique et une empreinte mémoire minimales.

Les applications du système contiki sont implémentées en langage de programmation C. Contiki se compose d'un noyau basé sur les événements, sur lequel les programmes d'application peuvent être chargés et déchargés dynamiquement au moment de l'exécution.

Le simulateur Cooja

Cooja est un environnement de simulation conçu pour les RCSFs, qui accompagne le système d'exploitation Contiki OS. Le simulateur est implémenté en Java mais permet de rédiger une application en C pour les nœuds capteurs. Dans Cooja, toutes les interactions avec les nœuds simulés sont effectuées via des plugins comme Simulation Visualizer, Timeline et Radio Logger. Il stocke la simulation dans un fichier xml avec l'extension 'csc' (configuration de simulation Cooja). Ce fichier contient des informations sur l'environnement de simulation, les plugins, les nœuds et leurs positions etc. Cooja est un outil très utile pour le développement de Contiki car il permet aux développeurs de tester leur code et leurs systèmes bien avant de le téléverser dans le matériel cible. La Figure 3.8 présente l'interface de simulateur cooja [AC].

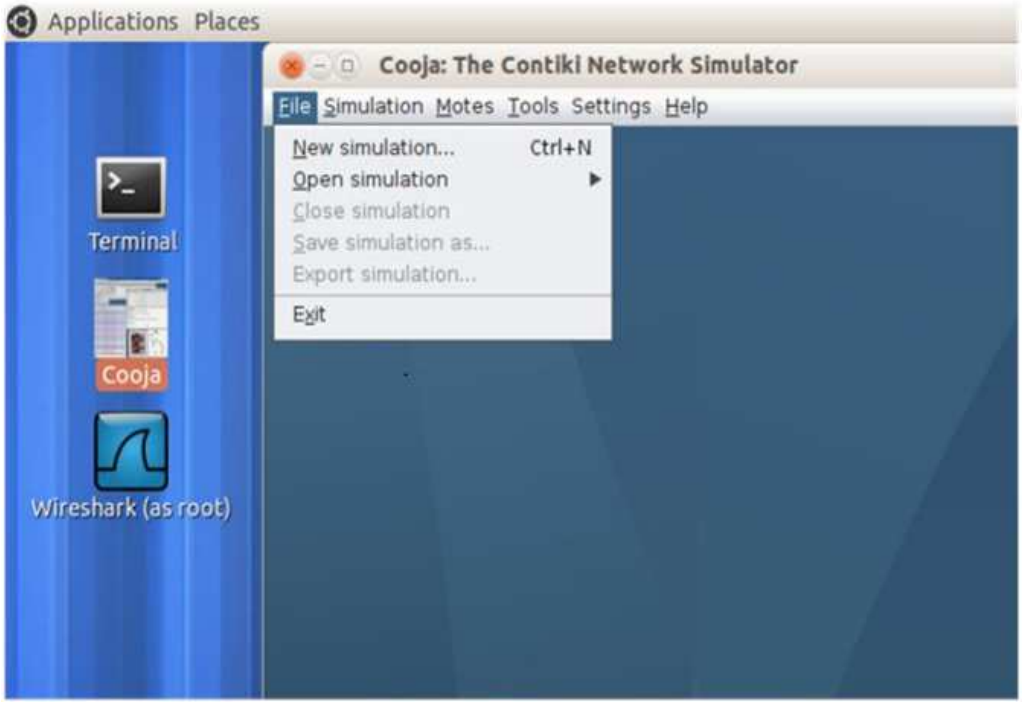


Figure 3.8 – Interface du simulateur Cooja.

3.6 Implémentation

3.6.1 Le protocole Message Queuing Telemetry Transport (MQTT)

Le protocole MQTT utilise un périphérique en tant que courtier "Broker", qui agit en tant que maître sur le réseau et acheminera les messages vers les périphériques appropriés. D'autres appareils vont publier et / ou s'abonner à certains types de messages, également appelés "Topics". Lorsqu'un appareil publie un message sur un topic, le broker le transfère à tous les appareils abonnés à ce topic. Avec cette architecture, la quantité de données transférées et les ressources nécessaires sur les périphériques sont réduites, pour les raisons suivantes :

- Les appareils doivent seulement connaître le broker. Si un appareil doit envoyer un message à 10 autres appareils, il n'a pas besoin de connaître les détails de ces 10 appareils. Il suffit de publier un message et le broker se chargera de la distribution.
- Les appareils ne reçoivent pas les messages auxquels ils ne sont pas abonnés. C'est bien mieux qu'une architecture de base "tout diffusion" par exemple.
- Le contenu des messages est minimal. Au moins, un message ne contient qu'un topic. Si davantage d'informations sont nécessaires, le message peut inclure tout type de données (texte, images, etc.).

Broker Mosquitto

Mosquitto est un broker open-source (courtier) pour le protocole MQTT soutenu par la fondation Eclipse.

Installation de Mosquitto sur une carte Raspberry Pi

L'objectif est d'utiliser Raspberry Pi (RPi) en tant que courtier(broker) MQTT dans une application IdO. Nous allons installer Mosquitto, le configurer et le tester pour nous assurer que tout a été correctement configuré (voir l'annexe B).

3.6.2 Procédure de mise en œuvre d'une RPi comme 6LBR

Nous tenons à rappeler que le 6LBR est une solution prête au déploiement pour les routeurs frontaliers, basée sur Contiki OS. Il fonctionne comme un routeur autonome sur les plates-formes matérielles intégrées open source telles que Raspberry Pi et BeagleBone ou dans un processus sur n'importe quelle distribution Linux.

6LBR est capable de gérer le traitement des demandes simultanées provenant des nœuds connectés et de gérer l'encombrement des paquets entre les interfaces sans fil IEEE 802.15.4 et IPv6. RPL est le protocole de routage destiné à prendre en charge la communication entre les nœuds 6LBR et d'extrémité.

Aussi, comme nous l'avons mentionné dans la Section 3.4, 6LBR fonctionne en trois modes, à savoir le mode routeur, le mode pont transparent et le mode pont intelligent. En ce qui nous concerne, nous avons utilisé le mode routeur. Rappelons que dans ce mode, le 6LBR agit comme un routeur IPv6 à part entière, interconnectant deux sous-réseaux IPv6. Le sous-réseau RCFS est géré par le protocole RPL et le sous-réseau Ethernet ou Wi-Fi est géré par IPv6 Neighbor Discovery Protocol (NDP). Ce mode fonctionne plus comme une passerelle entre Ethernet / Wi-Fi et le 6LoWPAN RPL.

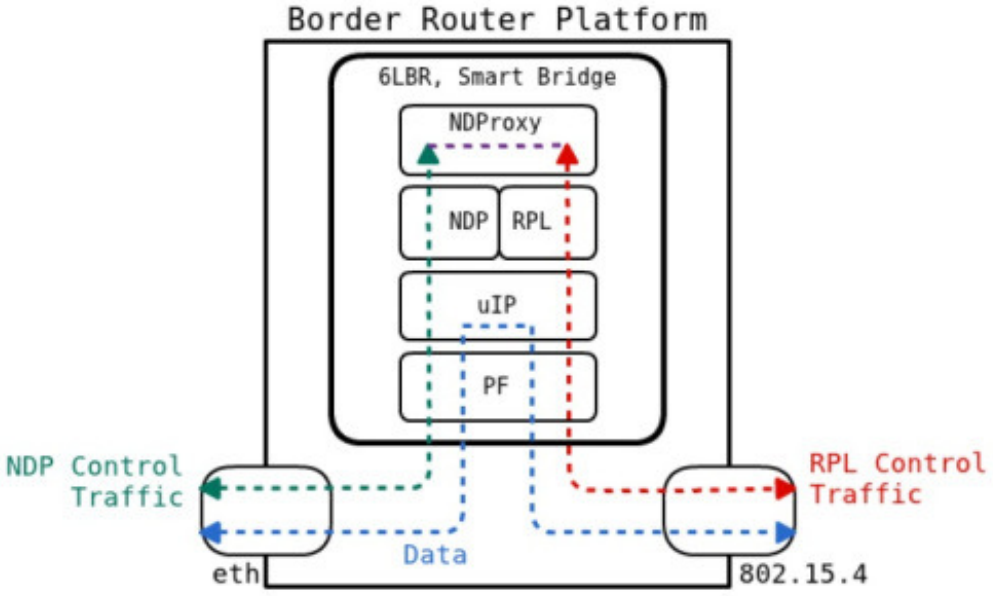


Figure 3.9 – Mode routeur.

L'installation de 6LBR sur Raspberry Pi

Nous allons utiliser une carte Raspberry Pi 3 comme plate-forme de routeur frontière et l'étendre avec une interface TelosB (sky) pour l'interface radio 802.15.4.

Les étapes nécessaires pour que la RPi soit un 6LBR, sont montrées dans l'annexe B.

3.6.3 SLIP radio

La radio SLIP est utilisée pour que le routeur de bordure, qui est la carte Raspberry Pi, puisse communiquer avec le réseau 802.15.4. SLIP-radio crée un tunnel pour la transmission sans fil des données provenant des terminaux IPv6 vers le 6LBR.

Pour compiler et flasher la SLIP-radio, nous avons utilisé une application fournie en exemple dans Contiki. Dans le répertoire Contiki nous avons suivi les étapes suivantes :

cd examples/ipv6/slip-radio
make TARGET=sky slip-radio.upload

Une fois la compilation terminée, sur un nœud capteur, nous flashons le programme SLIP-radio et nous connectons le nœud capteur TelosB à la carte Raspberry Pi 3 via un port Universal Serial Bus (USB).

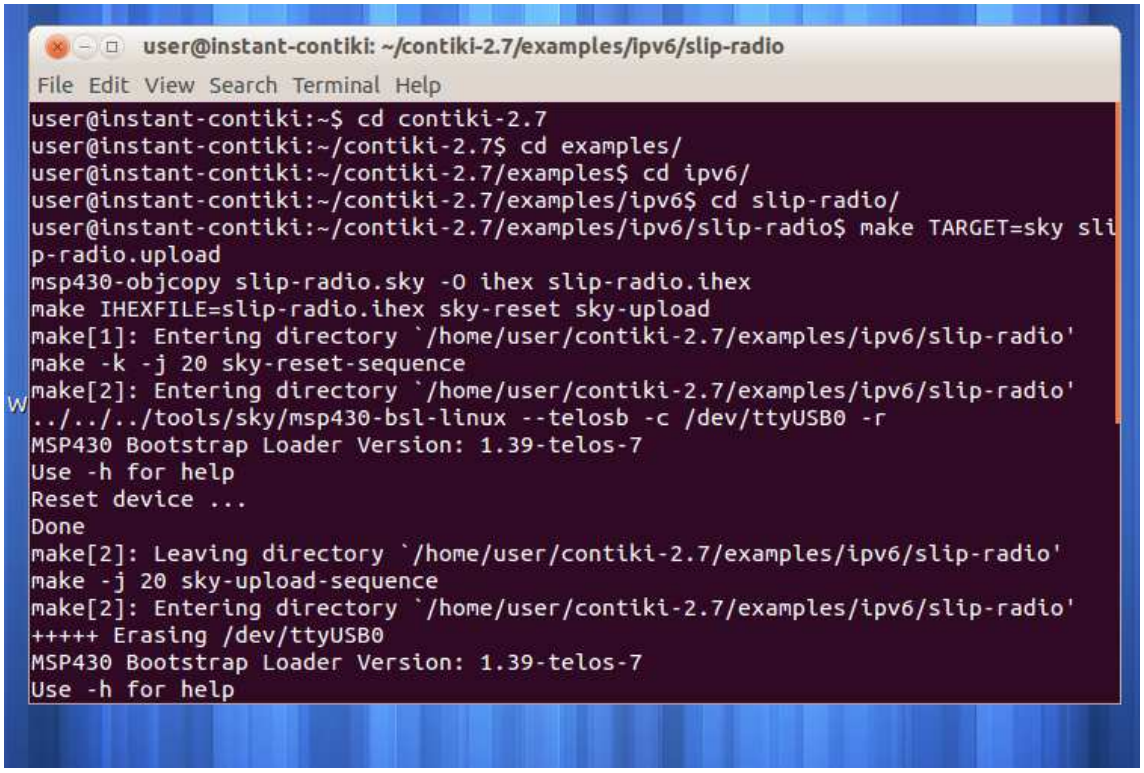


Figure 3.10 – Capture d'écran montrenat le téléversement du programme SLIP-radio dans le nœud TelosB.

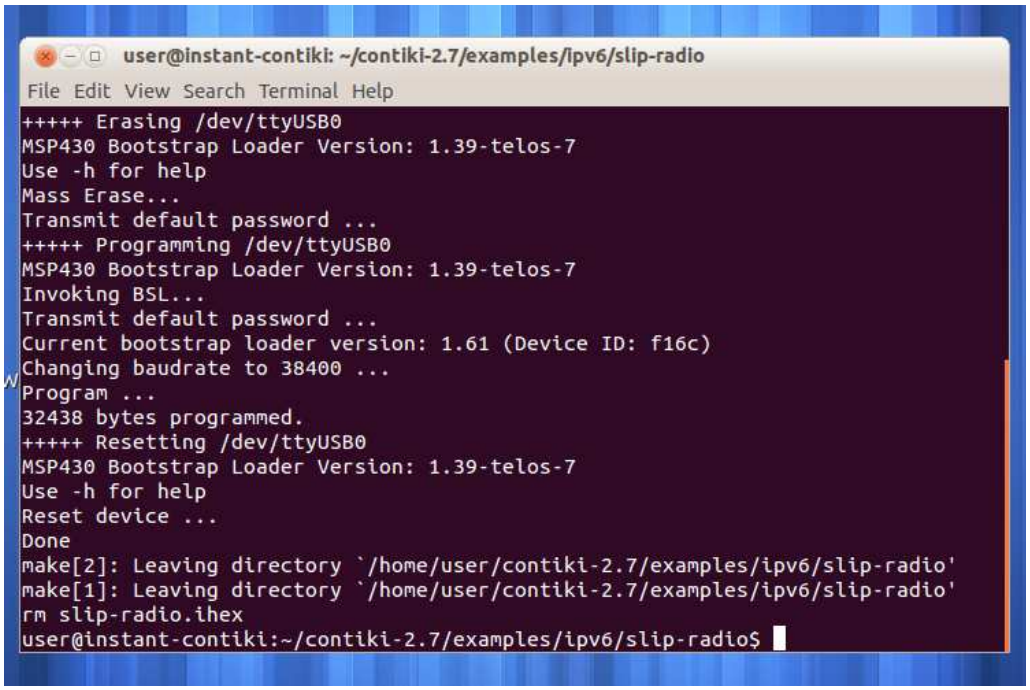


Figure 3.11 – Capture d'écran montrant le téléversement du programme SLIP-radio dans le nœud TelosB.

Ensuite, nous effectuons une vérification et nous exécutons cette commande pour nous assurer que le nœud est connecté sur le port ttyUSB de la Raspberry Pi 3 :

dmesg | grep ttyUSB

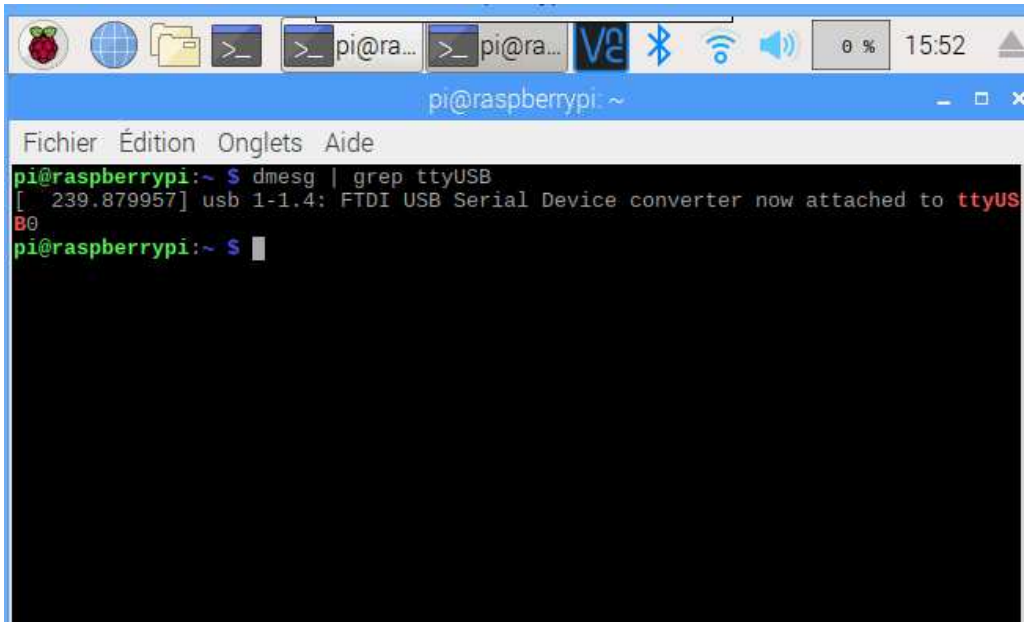


Figure 3.12 – la connexion de nœud le port ttyUSB de la RPI

3.6.4 Sky websense

Nous avons programmé les nœuds avec l'application websense, qui permet la collecte et la transmission de données à la SLIP-radio. Nous avons compilé et uploader l'application websense sur les nœuds capteurs, en procédant de la façon suivante :

cd examples/ipv6/sky-websense
make TARGET=sky sky-websense.upload

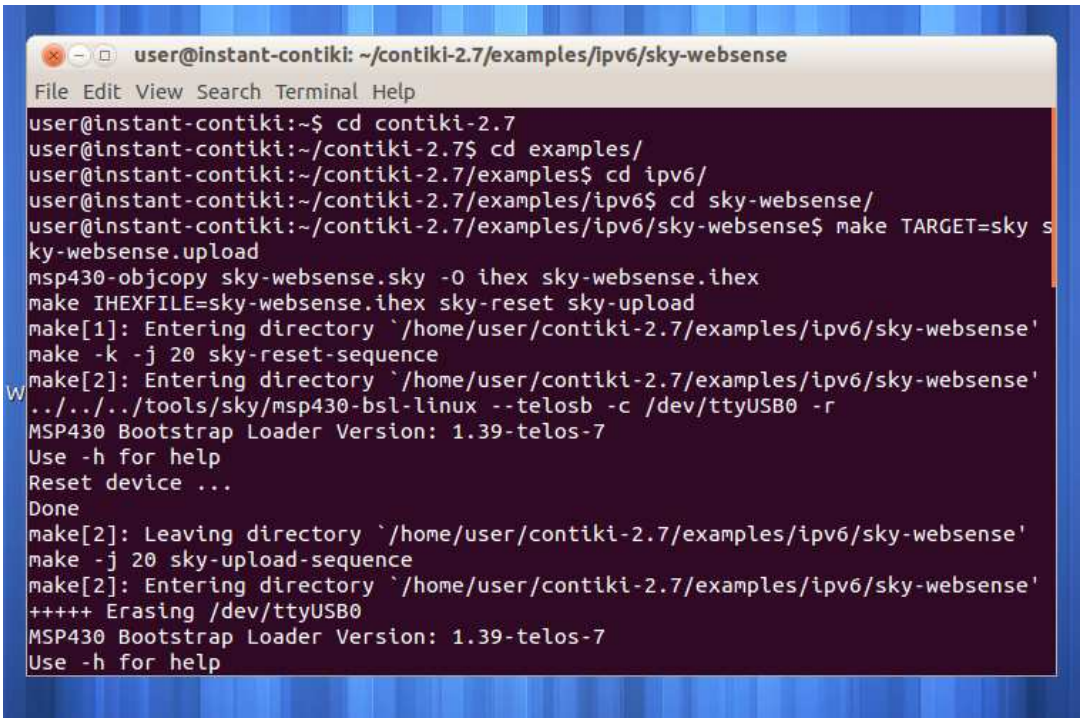
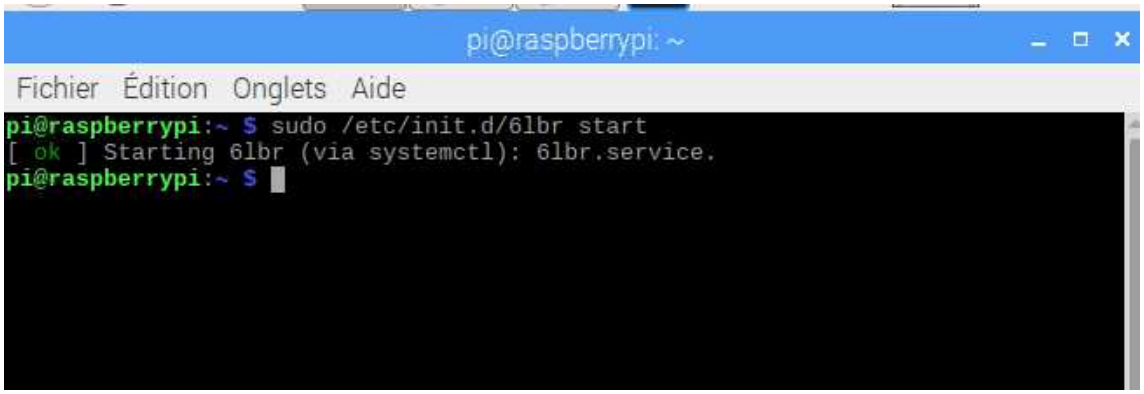


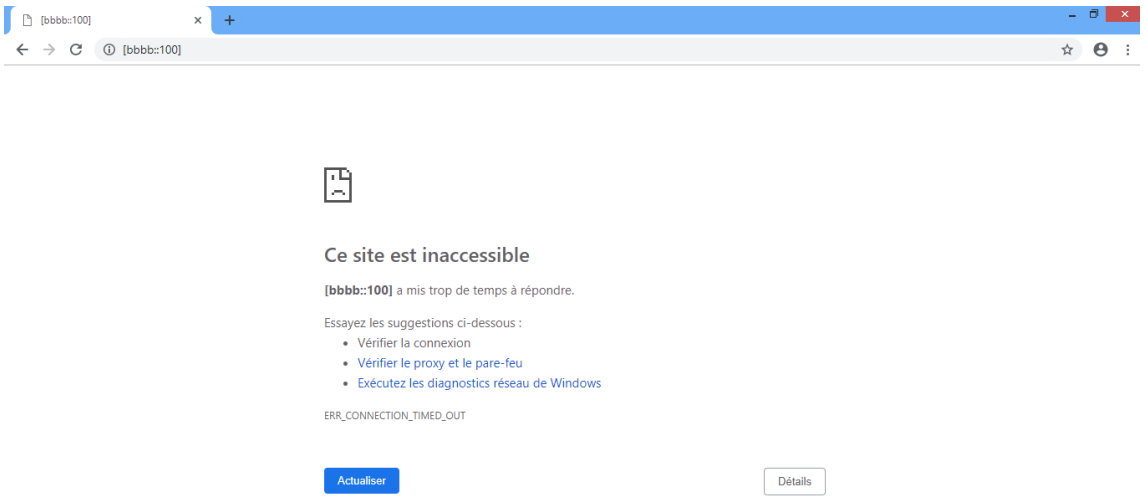
Figure 3.13 – Capture d'écran montrant le téléversement du programme sky-websense dans le nœud TelosB.

3.7 Difficultés rencontrées lors de la réalisation de la plate-forme matérielle

A la fin de l'installation de la solution 6LBR sur la RPi, et en tapant la commande "sudo /etc/init.d/6lbr start" pour tester le bon fonctionnement de celui-ci, le message d'erreur suivant s'affiche :

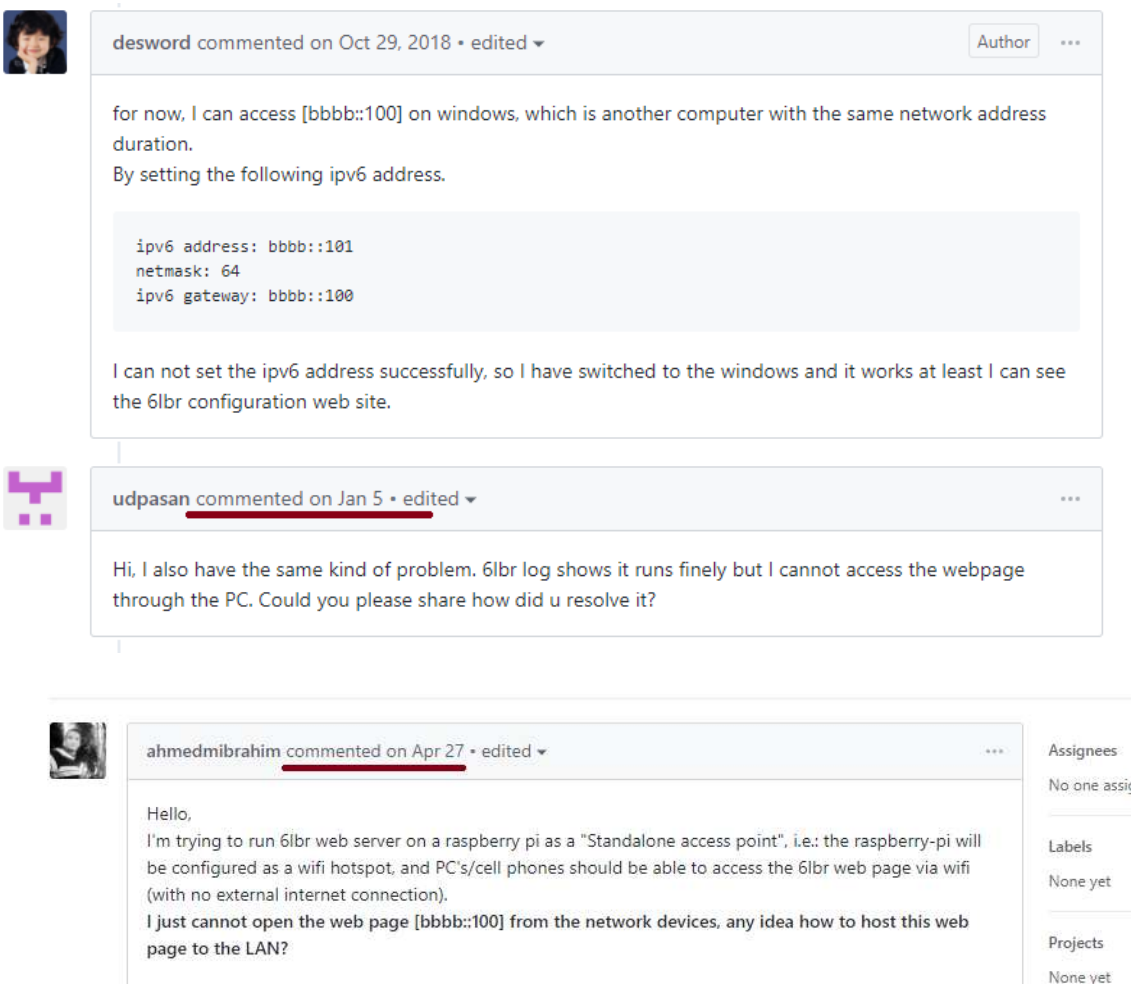
Donc le probléme rencontré c'était de ne pas pouvoir accéder à la page dans le navigateur en entrant l'adresse ipv6 "bbbb : :100" on nous a affiché un message "ce site est inaccessible".





Nous avons également essayé d'installer 6lbr sur la machine linux (les étapes sont détaillées dans l'annexe A). Malheureusement, nous avons rencontré le même problème (toujours le site en question est toujours inaccessible).

Nous tenons à signaler que nous ne sommes pas les seuls à avoir fait face à ce problème. Ceci est confirmé par les différentes questions postées sur des forums sur le net, comme peuvent le témoigner les Figures suivantes :



Vu ces difficultés rencontrées qui nous ont coûté un temps énorme en leur cherchant une solution, nous avons refléchi à une solution de rechange qui consiste à utiliser un nœud, en l'occurrence node MCU (espressif systems) dôté d'une interface WIFI. L'avantage de ce nœud, équipé d'un capteur de température, est qu'il va nous permettre de transférer les températures capturées vers le broker Mosquitto installé sur la Raspberry Pi , directement via son interface WIFI. La communication est donc entre l'interface WIFI du node MCU et l'interface WIFI de la Raspberry, sans qu'on ait besoin d'installer un routeur frontière (6LBR) sur la raspberry Pi.

3.8 description de la solution de rechange proposée

Dans cette application on a un nœud capteur (node MCU), dôté d'une interface WIFI et d'un capteur de température . Ce nœud va capturer la température du monde physique et la publier dans le broker Mosquitto installé sur la Raspberry Pi. D'autre part, une application Mosquitto client installée sur un smartphone va souscrire au topic température. Si la valeur de la témpérature réçue est différente de 37 donc il faut générer une alarme sinon l'application attend une nouvelle valeur.

Les Figures 3.14 et 3.18 illustrent respectivement l'organigramme et les étapes de l'exécution de notre application

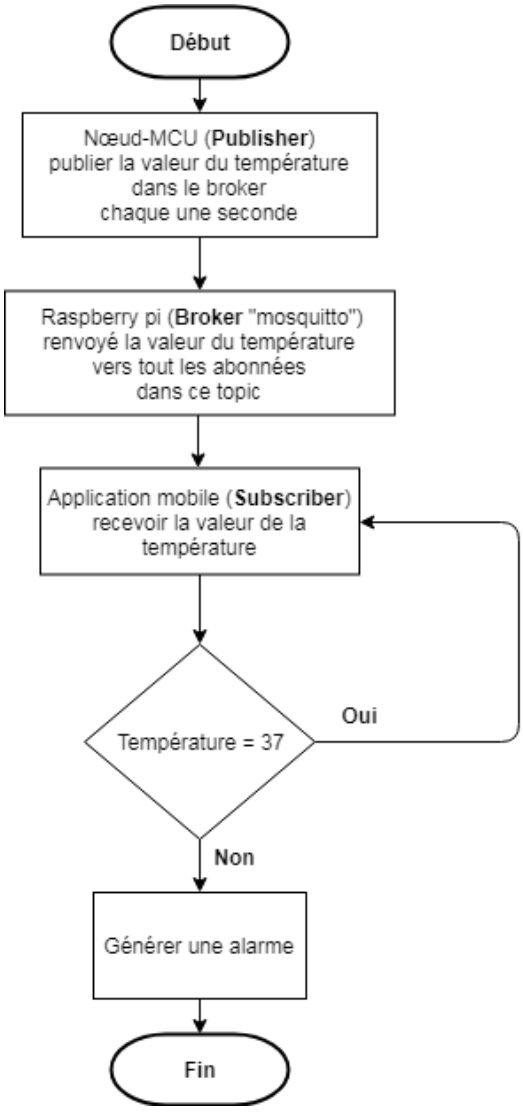


Figure 3.14 – Organigramme de l'application.

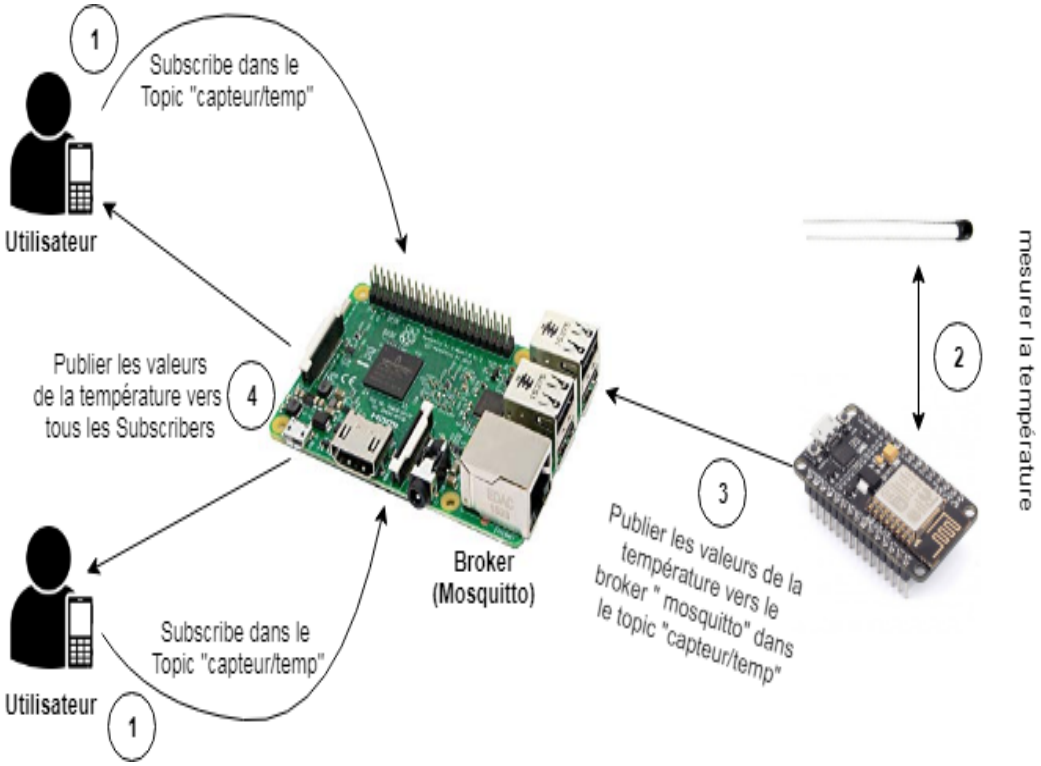


Figure 3.15 – Architecture de l'application

Interface de l'application mobile "MQTT TMP" :

Authentification :

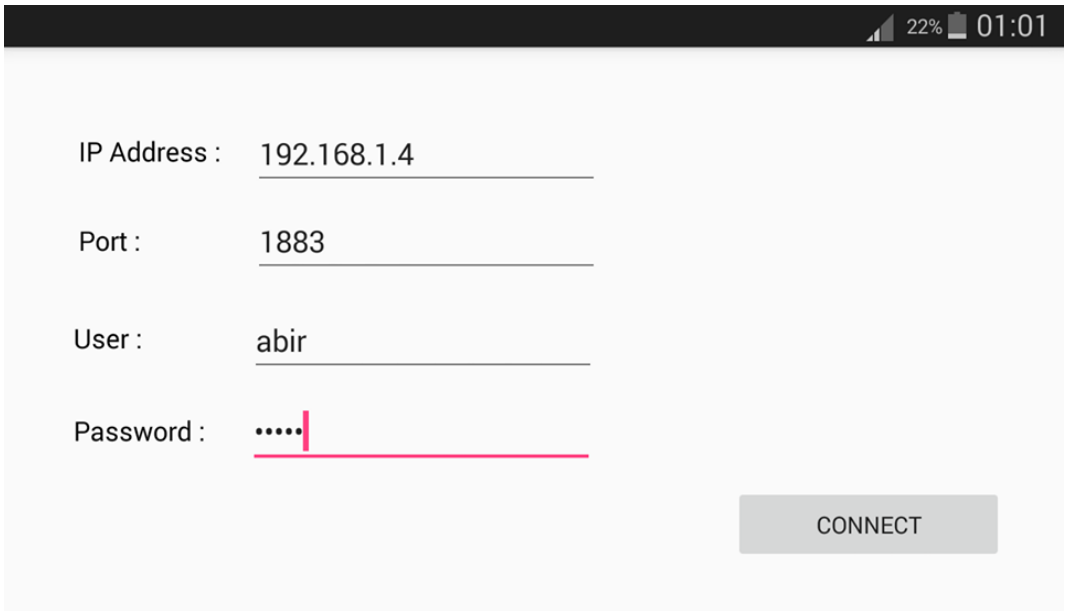


Figure 3.16 – Authentification

Graphe de changement de la température capturé :

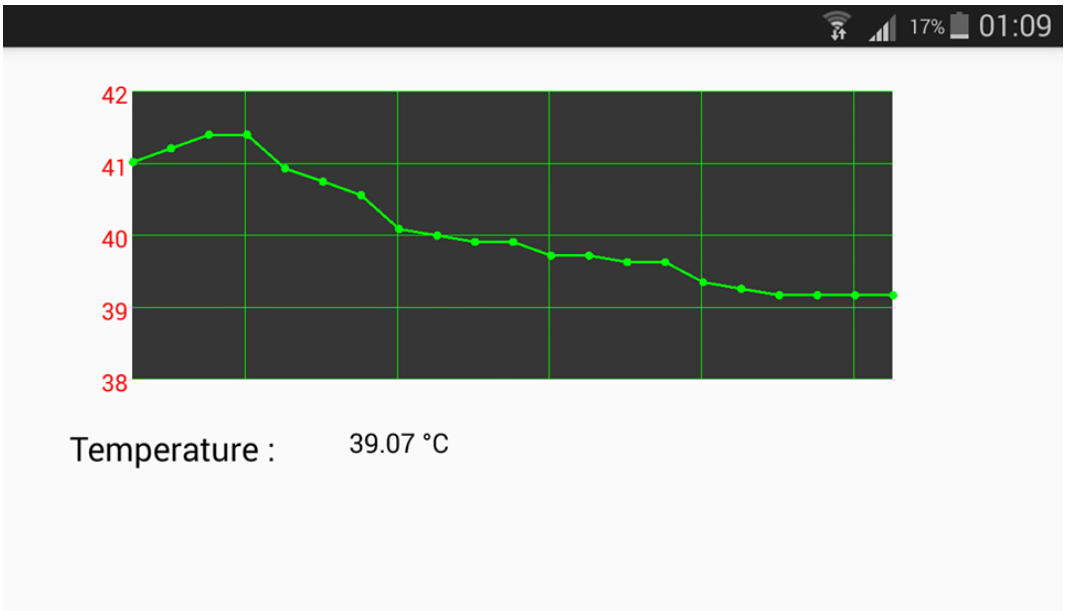


Figure 3.17 – Graphe de changement de la température capturé

Génération d'une alarme :

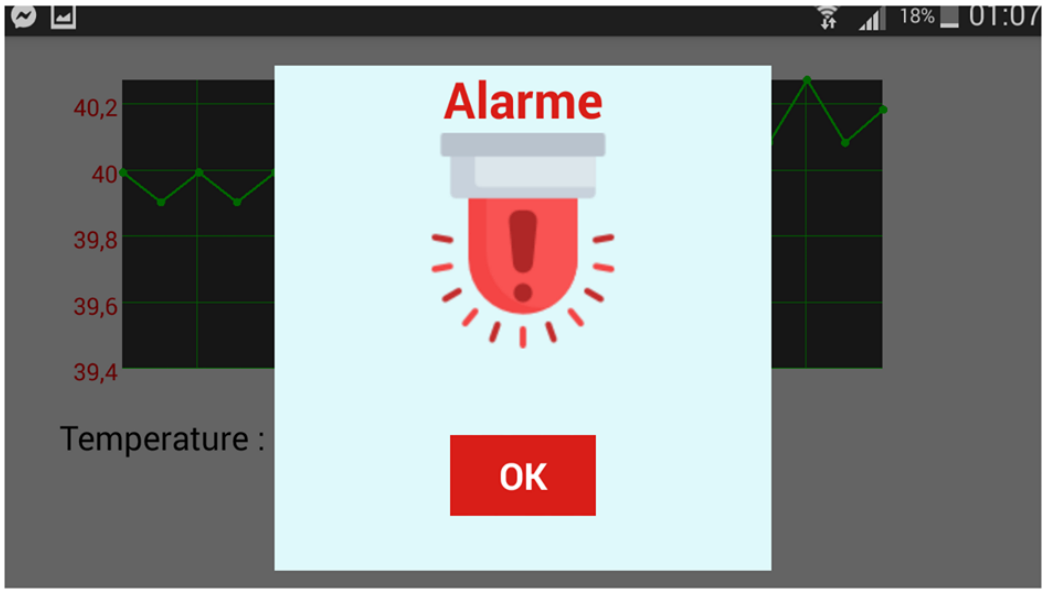


Figure 3.18 – Génération d'une alarme

3.9 Simulation de Border Router

Comme signalé dans la Section précédente, nous allons simuler le fonctionnement de notre plate-forme réelle, et ce à défaut d'une solution à notre problème recontré.

La simulation va se faire en utilisant Contiki/Cooja. Pour demarrer ce dernier, nous allons ouvrir une terminal et nous procédons comme suit :

cd contiki/tools/cooja
Ant run

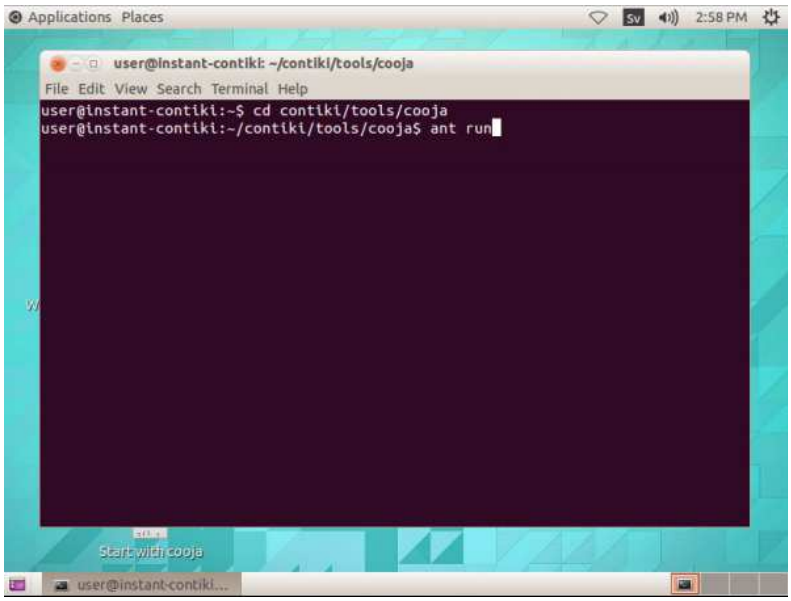


Figure 3.19 – Lancement du simulateur Cooja.

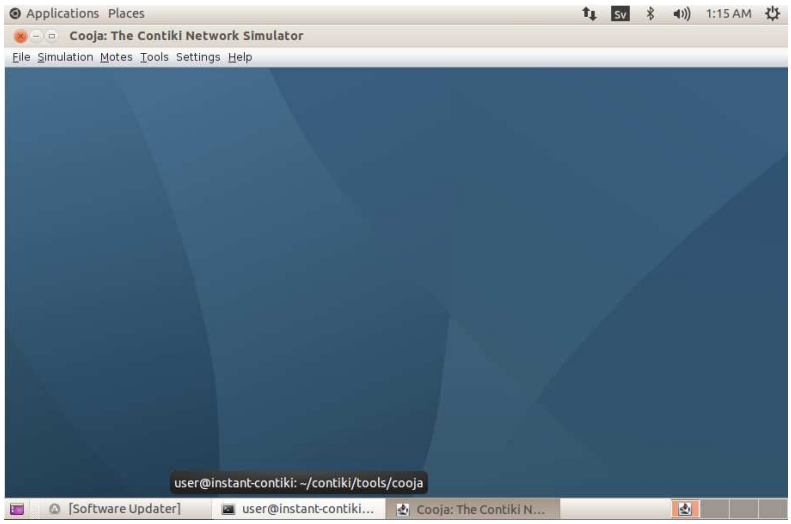


Figure 3.20 – Interface du simulateur Cooja.

Les étapes de la simulations sont énumérées ci-dessous :

1. Création d'une nouvelle simulation en cliquant sur le menu Fichier, puis sur nouvelle simulation.

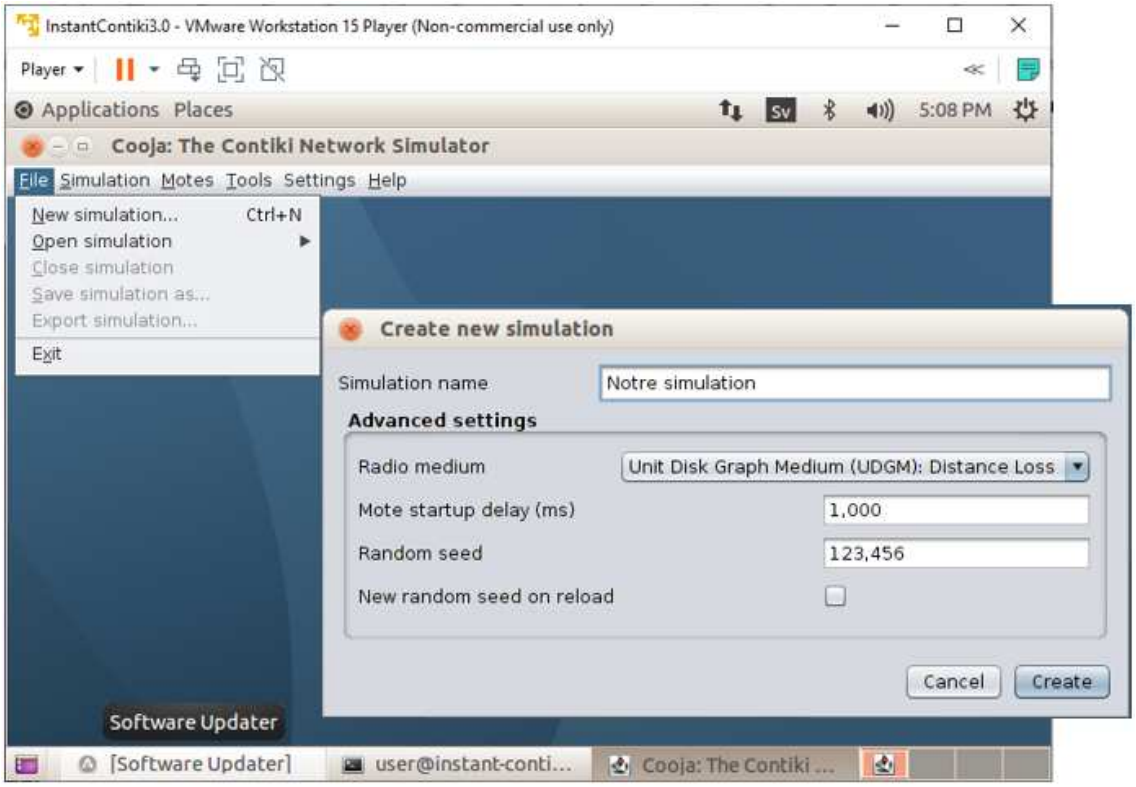


Figure 3.21 – Creation d'une nouvelle simulation.

2. Création du réseau en ajoutant des nøuds capteurs ("Sky mote") :
   - Dans le menu Motes, cliquer sur nouveau type de mote et sélectionner Sky mote.
   - Dans la boîte de dialogue Create Mote Type ouverte par Cooja, nous allons garder le nom de notre type de Mote proposé par défaut et nous allons cliquer sur le bouton Parcourir pour choisir notre application Contiki située dans le répertoire /home/user/Contiki-3.0/examples/ipv6/rpl-border-router. Il s'agit du fichier border-router.c. Le nœud capteur sur leqquel est installé cette application va fonctionner en tant que routeur frontière.

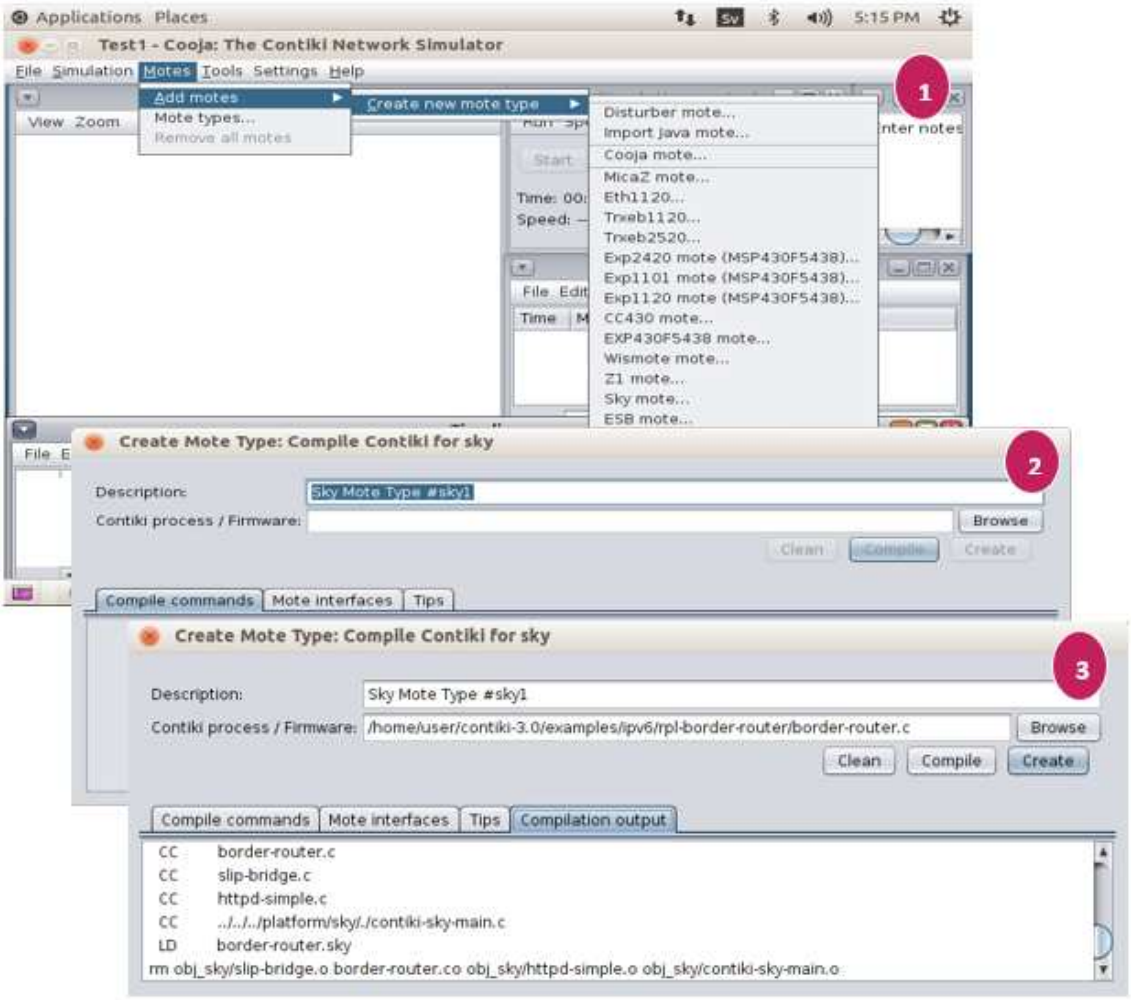


Figure 3.22 – Création des nœuds (capteurs).

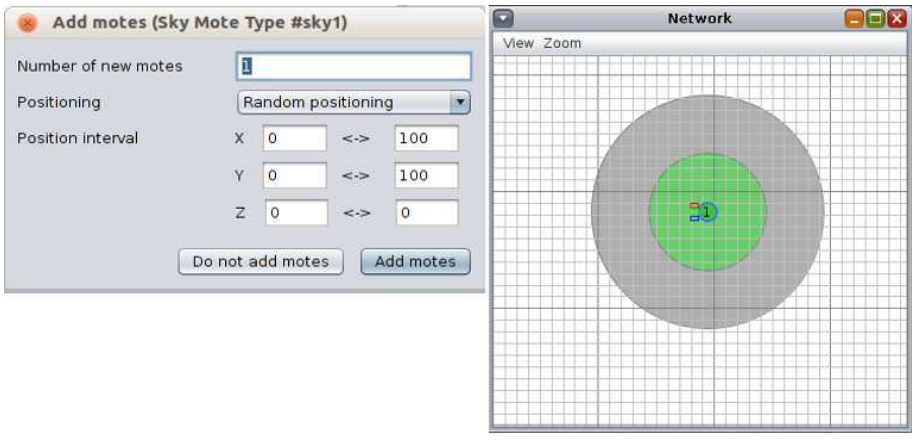


Figure 3.23 – Nombre de motes ajouté.

— Pour créer d'autre nœuds capteurs de type sky-websense, on passe par les mêmes étapes précédentes, mais au lieu de charger le fichier border-router.c, on chargera sur chaque nœud ajouté le fichier sky-websense.c

qui est dans le répertoire /home/user/Contiki-3.0/examples/ipv6/sky-websense. On ajoute 5 nœuds comme il est montré dans la Figure suivante.

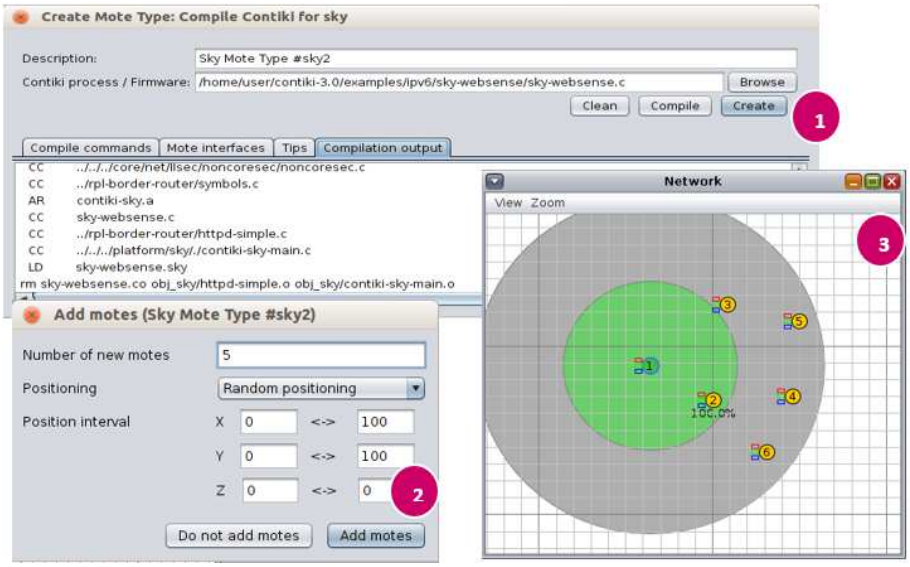


Figure 3.24 – Ajout des nœuds de type Sky websense.

— Allumer le routeur : Clic droit dessus. Une fenêtre "Serial socket server" va apparaitre avec une option de démarrage (start). Cliquer dessus.
Ensuite, nous lancons la simulation à partir de la fenêtre 'Simulation control'.

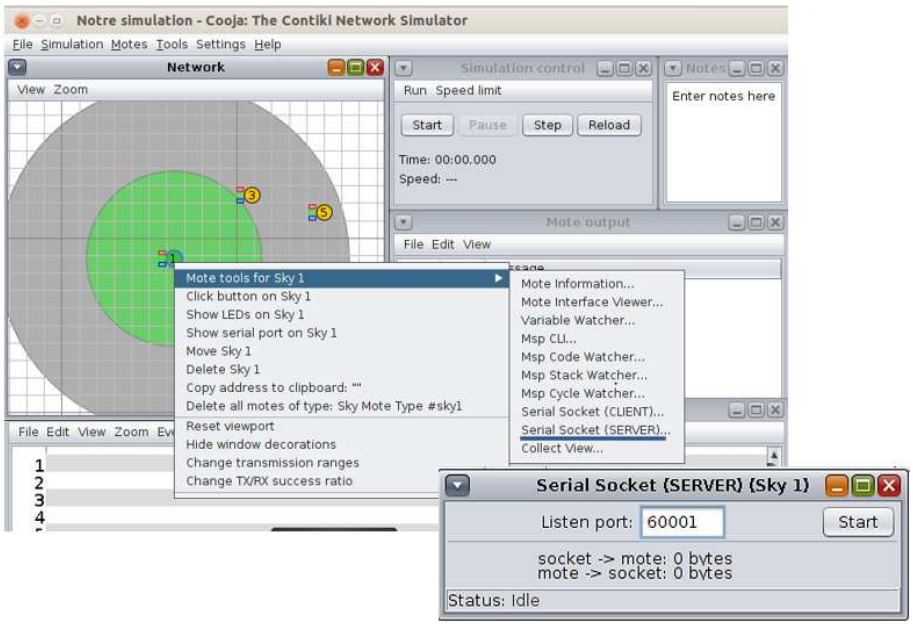


Figure 3.25 – Démarrer le routeur.

— Démarrer le routeur à partir du terminal afin que nous puissions obtenir les adresses IPv6 du routeur. Dans un nouveau terminal, nous devons exécuter la commande "make connect-router-cojaa". Cela lancera un programme nommé tunslip6 qui configure une interface sur la pile IP Linux et connecte cette interface via un socket au nœud routeur de bordure dans Cooja.

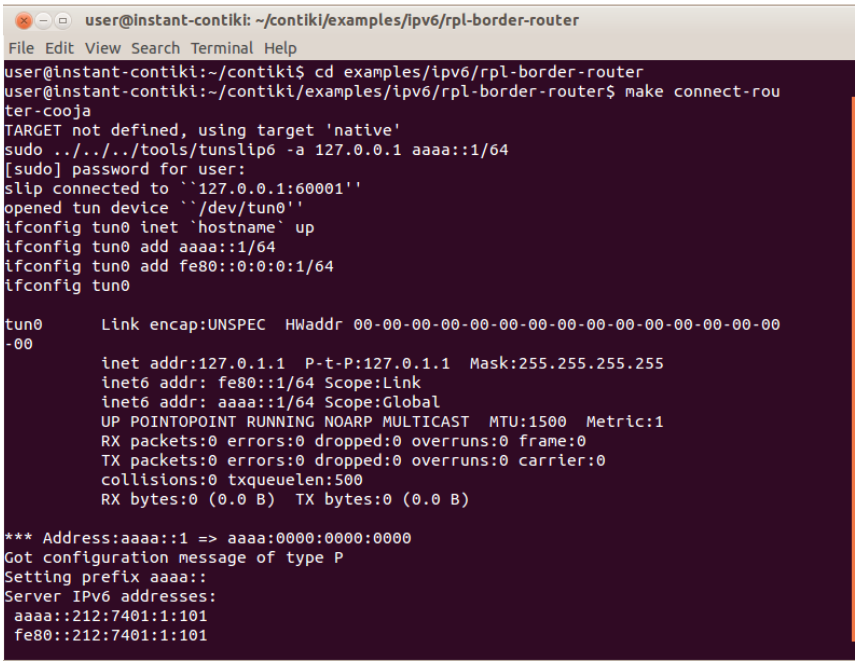


Figure 3.26 – Obtenir les adresses IPv6 du border-router.

— Les Figures suivantes montrent les paquets échangés entre les motes.

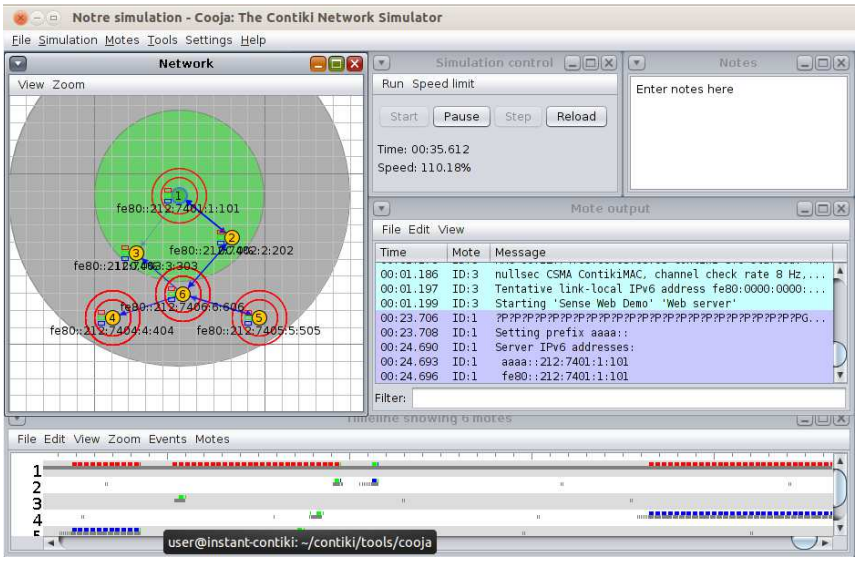


Figure 3.27 – Lancement de la simulation.

— Vérifier la connectivité avec le nœud border-router. Ouvrir un autre terminal puis taper la commande :

ping6 aaaa : :212 :7401 :1 :101.

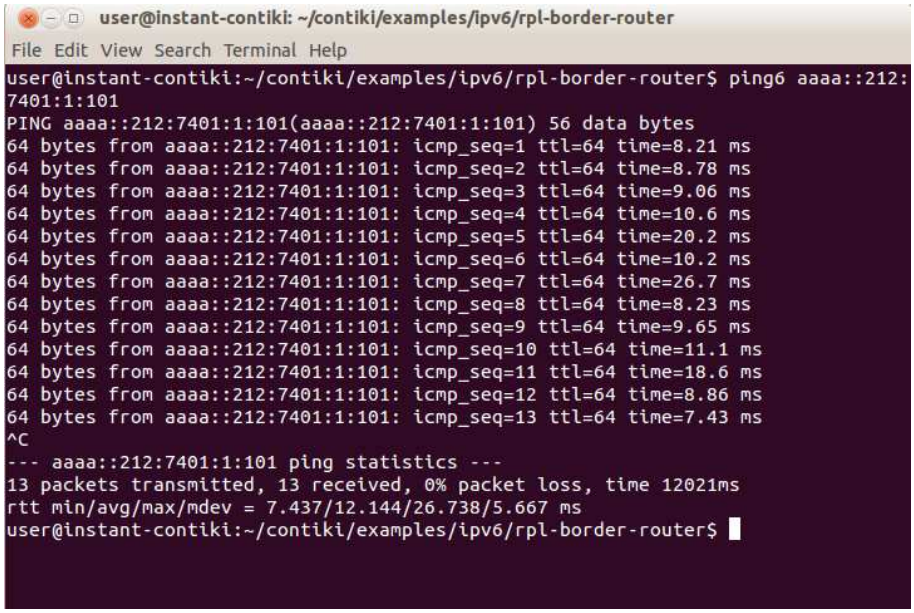


Figure 3.28 – Test du ping du nœud border-router.

La connexion a été établie avec succès.

— Confirmer que les motes utilisent 6LoWPAN : cliquer sur "Tools" puis sur "Radio messages", puis cochet '6LoWPAN Analyzer PCAP'.

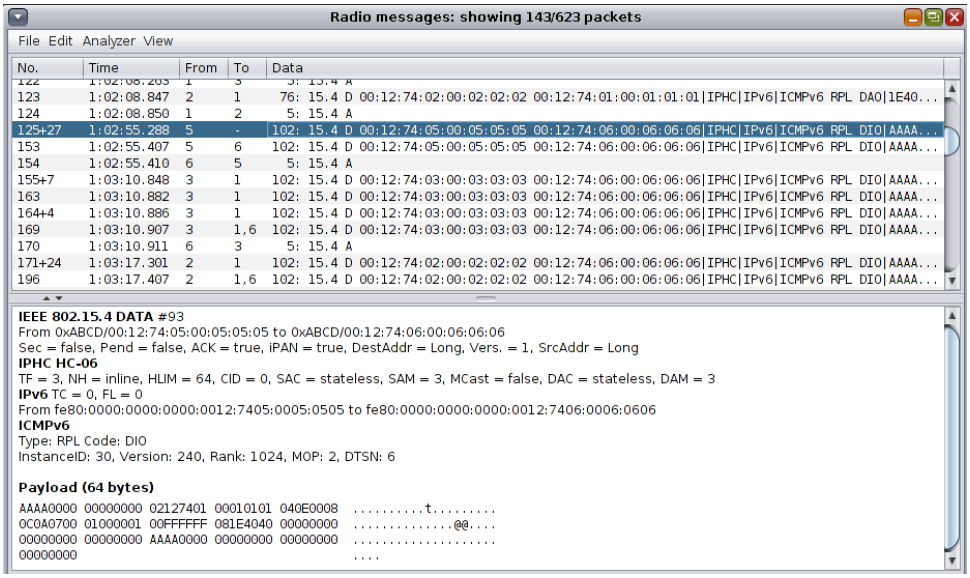


Figure 3.29 – Les nœud sont connectés au réseau 6LoWPAN

Nous pouvons constater que le mote numéro 5 utilise la norme IEEE

802.15.4, et il envoie des trame IPv6 au mote numéro 6. Donc les motes utilisent le réseau 6LoWPAN.

— Ouvrir le navigateur Mozilla FireFox et taper l'adresse du border-router dans la bare de recherche (http ://[aaaa : :212 :7401 :1 :101]/) pour voir les adresses des voisins et les adresses de routes, et ce afin de nous confirmer que le border-router fonctionne maintenant correctement.

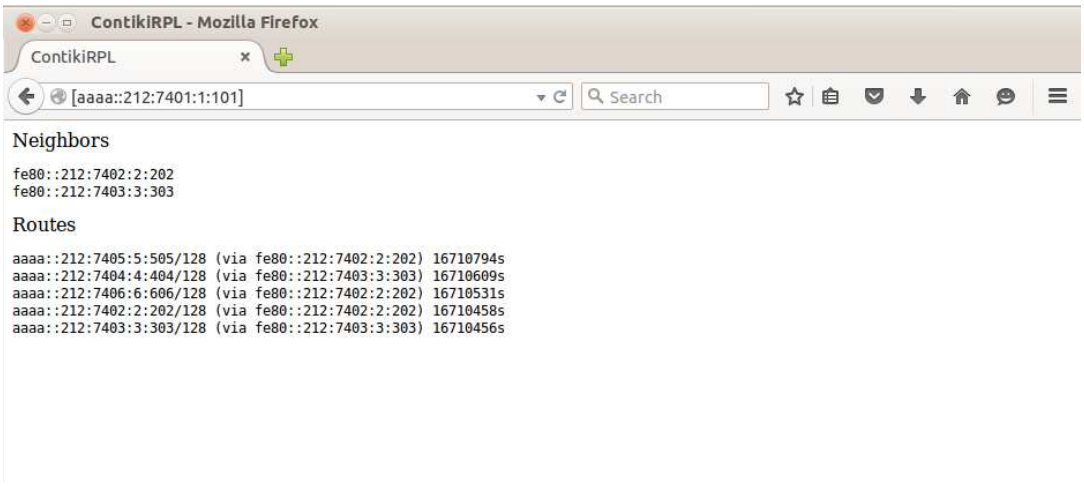


Figure 3.30 – Les adresses des nœuds voisins et les adresses de routes

— Nous accédons aux autres nœuds du réseau via leurs adresses IPv6, afin de pouvoir constater les informations détectées (température, lumière et humidité).

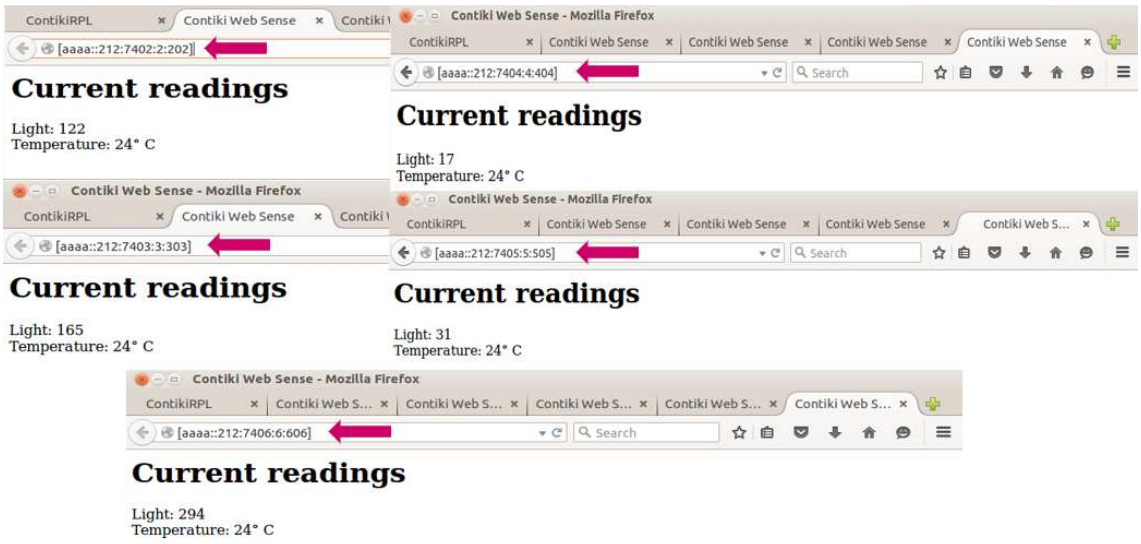


Figure 3.31 – Les informations détectés par les nœuds capteurs (température, lumière et humidité).

3.10 Conclusion

Dans ce chapitre nous avons abordé la partie mise en œuvre de notre travail, en décrivant les étapes de réalisation de notre plate-forme d'échange entre un RCSF et un broker MQTT installé sur une carte Raspberry Pi 3, et les étapes de simulation de cette même plate-forme. Nous notons que nous avons recouru à la simulation en raison d'un problème rencontré dans le fonctionnement du 6LBR installé sur la carte Raspberry Pi 3.

Conclusion Générale

Dans le cadre de notre projet de fin d'études, nous avions le choix entre de nombreux sujets. Plusieurs sujets proposés nous intéressaient. Mais en choisissant l'Internet des Objets, nous voulions nous distinguer et en savoir plus, notamment comment la mettre en œuvre, comme c'est la technologie la plus récente dans le monde.

Après avoir présenté le concept de l'Internet des Objets, ses domaines d'application et son fonctionnement, nous avons donné un aperçu général sur les Réseaux Capteurs Sans Fil, leurs architectures et leurs domaines d'applications, ainsi que des notions pour l'intégration des RCSFs dans l'Internet des objets. Ensuite, nous nous sommes consacrées à la description de la technologie 6LoWPAN. La dernière partie du travail a été dédiée à la présentation des outils matériels et logiciels utilisés et à toutes les étapes de la réalisation de notre plate-forme matérielle permettant un échange de données entre un Réseau de Capteurs Sans Fil (RCSF) constitué de deux nœuds capteurs TelosB et une carte raspberry Pi 3 qui joue le rôle d'un border router. Les données des nœuds capteurs sont en effet publiés sur un broker Mosquitto installé sur la carte Raspberry. Un smartphone jouant le rôle de souscripteur aux données publiées, fait aussi partie de la plate-forme.

Comme perspective de ce travail, nous projettons d'essayer d'autres solutions permettant de déployer un border router (autre que 6LBR) ainsi que d'essayer d'autres protocoles d'accès aux données des nœuds capteurs, autre que le MQTT. Nous pourrions aussi essayer de réduire la latence de livraison des données aux clients, surtout dans le cas ou la plate-forme est utilisée dans le domaine de la surveillance où les alarmes doivent parvenir à temps aux concernés.

Bibliographie

[AC] HANANE AMIMER and FADIA CHEKROUN. *Etude de l'authentification d'une source de diffusion dans le contexte de l'Internet des Objets.* PhD thesis, 14-01-2018.

[AC17] Younes Ait Chabane. Optimisation du routage dans un contexte 6lowpan, article. 2017.

[AFGM+15] Ala Al-Fuqaha, Mohsen Guizani, Mehdi Mohammadi, Mohammed Aledhari, and Moussa Ayyash. Internet of things : A survey on enabling technologies, protocols, and applications. *IEEE communications surveys & tutorials, Article*, 17(4) :2347–2376, 2015.

[AMB+17] Raouf Achour, Naima Makhloufi, Abdellah Boukerram, et al. *Authentificaiton Dans L'iot.* PhD thesis, Université abderrahmane mira béjaia, 2017.

[ASSC02a] Ian F Akyildiz, Weilian Su, Yogesh Sankarasubramaniam, and Erdal Cayirci. A survey on sensor networks. *IEEE Communications magazine, Article*, 40(8) :102–114, 2002.

[ASSC02b] Ian F Akyildiz, Weilian Su, Yogesh Sankarasubramaniam, and Erdal Cayirci. Wireless sensor networks : a survey. *Computer networks, Article*, 38(4) :393–422, 2002.

[BF] Chahrazed BAGHDADLI and Zineb FEROUANI. *DEVELOPPEMENT D'UNE APPLICATION DEPLOYEE SUR UN RESEAU DE CAPTEURS SANS FIL SUPPORTANT 6LOWPAN.* PhD thesis.

[Bon15] Sébastien Bonnet. Mqtt : Un protocole dédié pour l'iot. http://www.digitaldimension.solutions/blog/avis-d-experts/2015/02/mqtt-un-protocole-dedie-pour-liot/, Février 2015.

[DÇYG16] H Doğan, MF Çağlar, M Yavuz, and MA Gözel. Use of radio frequency identification systems on animal monitoring. *SDU International Journal of Technological Sciences, Article*, 8(2) :38–53, 2016.

[ele19] Orbit electronic. Cartes mini-pc : Raspberry pi 3 model b. http://www.orbit-dz.com/produit/scanner-automobile-2/cartes-mini-pc/raspberry-pi-3-detail, Avril 2019.

[Eva11] Dave Evans. L'internet des objets comment l'évolution actuelle d'internet transforme-t-elle le monde. *Livre Blanc, Édition : Cisco IBSG, États-Unis*, 2011.

[Fra19] Anthony Fradera. Slip : Serial line ip. https://www.guill.net/index.php?cat=3&pro=3&wan=3, Avril 2019.

[GP11] Nicolas Colomer Guillaume Plouin. Modèles d'architectures de l'internet des objets. https://blog.octo.com/modeles-architectures-internet-des-objets/, Septembre 2011.

[Iva17] Dmitri Ivanov. *Synchronization of measurement and data transfer in 6LoWPAN network*. PhD thesis, Tallinn University of Technology, 2017.

[Kam17] Patrick Olivier Kamgueu. *Configuration dynamique et routage pour l'internet des objets*. PhD thesis, Université de Lorraine, 2017.

[Kas17] Christophe Kassabji. Le protocole mqtt dans l'iot. https://blog.groupe-sii.com/le-protocole-mqtt-dans-liot/, Novembre 2017.

[MB16] Fadwa Moussaoui and Fadwa Boubekeur. *Les arbres couvrants de la théorie à la pratique. Algorithmes auto-stabilisants et réseaux de capteurs*. PhD thesis, Université Pierre et Marie Curie-Paris VI, 2016.

[mf17] Jean michel franco. Objets connectés et gdpr, comment sécuriser le tout-connecté ? https://fr.blog.businessdecision.com/bigdata/2017/04/objets-connectes-gdpr-tout-connecte/, Avril 2017.

[Nab12] Mme LABRAOUI Nabila. La sécurité dans les réseaux sans fil ad hoc. *Doctorat Spécialité :"Informatique", A L'université de TLEMCEN Faculté des sciences, Article*, 2012.

[Nem15] Mehdi Nemri. Demain l'internet des objets. *France Stratégie, Note d'analyse, Article*, 22(01) :2015, 2015.

[NR14] Alex Nasarre Ramírez. Implementació d'internet de les coses mitjançant raspberry pi. Master's thesis, Universitat Politècnica de Catalunya, 2014.

[Rot12] Damien Roth. *Gestion de la mobilité dans les réseaux de capteurs sans fil.* PhD thesis, Université de Strasbourg, 2012.

[Sah11] Belkheyr Sahraoui. *La Géo-localisation dans les Réseaux de Capteurs sans Fil.* PhD thesis, 2011.

[SB11] Zach Shelby and Carsten Bormann. *6LoWPAN : The wireless embedded Internet*, volume 43. John Wiley & Sons, livre, 2011.

[TB+17] Youva Tizzaoui, S Berrah, et al. *Internet des Objets «IoT» Application.* PhD thesis, Université Abderrahmane Mira, 2017.

[Wik15] Source Wikipedia. sensor node. https://en.wikipedia.org/wiki/Sensor_node, Février 2015.

[Wik19] Source Wikipedia. 6lowpan. https://fr.wikipedia.org/wiki/6LoWPAN, Mars 2019.

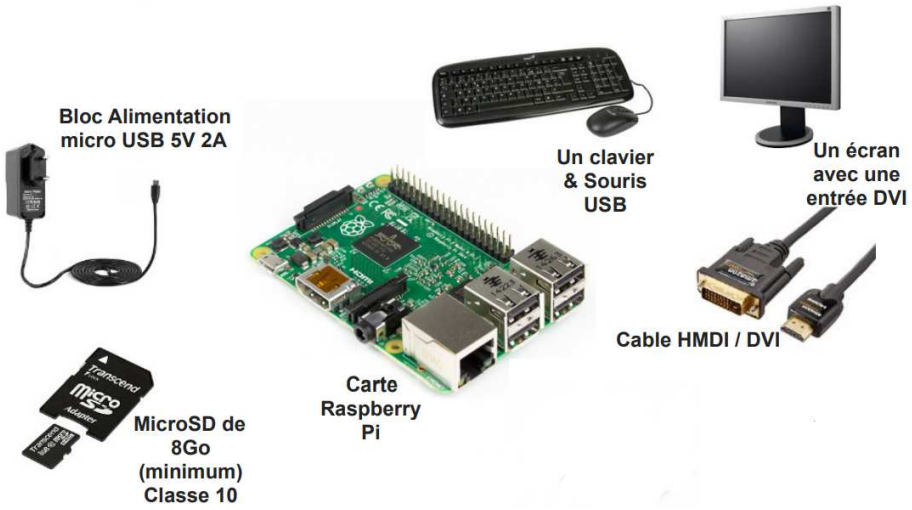
Annexes

Annexe A

Les installations sur la carte Raspberry Pi

A.1 Le système d'exploitation Raspbian

a) Préparer les outils necessaires pour se connecter à la raspberry Pi3 :

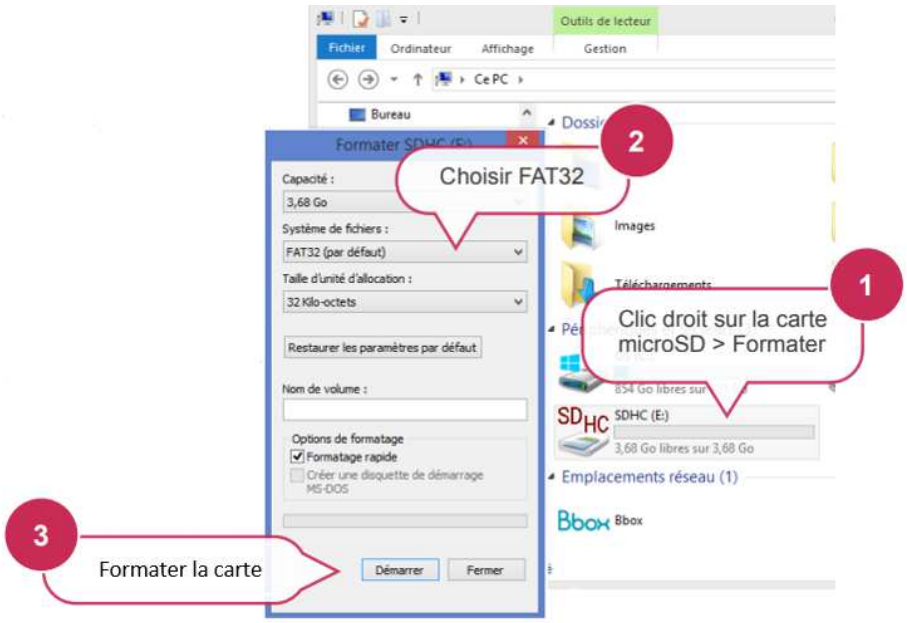


— Une alimentation : Pour se connecter à une prise d'alimentation, le Raspberry Pi dispose d'un port micro USB (identique à celui de nombreux téléphones mobiles au moins 2,5 A).
— Une carte micro SD : pour stocker tous les fichiers et le système d'exploitation Raspbian (d'une capacité d'au moins 8 Go).
— Un clavier et une souris : Pour commencer à utiliser Raspberry.
— Un écran de télévision ou d'ordinateur : Pour afficher l'environnement de bureau Raspbian.

b) Préparation du système d'éxploitation :

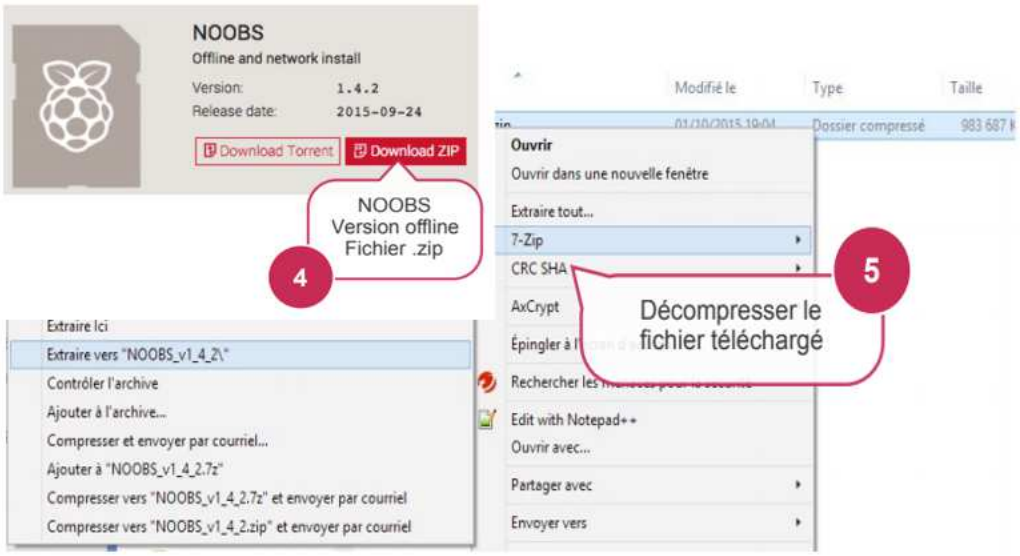
— Formater en FAT32 une carte microSD (8Go minimum) qui sera dédiée au Raspbian. Pour cela utiliser directement l'utilitaire de disque de windows.

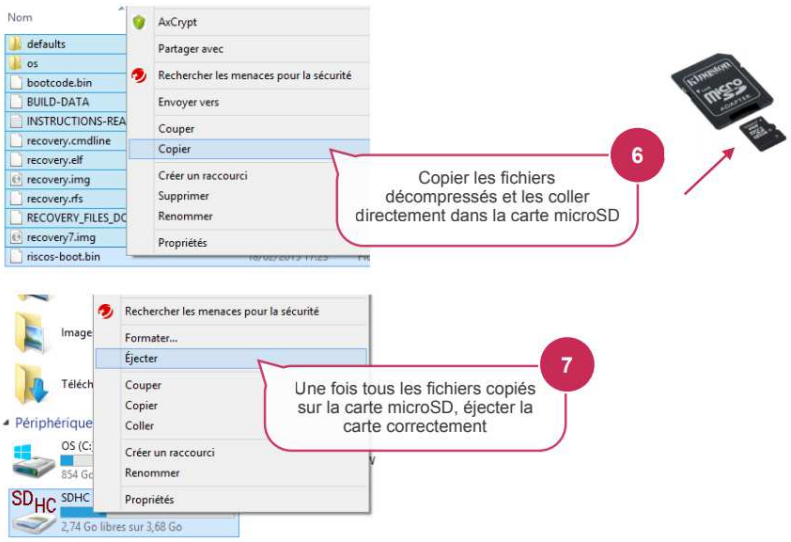
Clic droit sur le carte microSD, puis formater (FAT32).



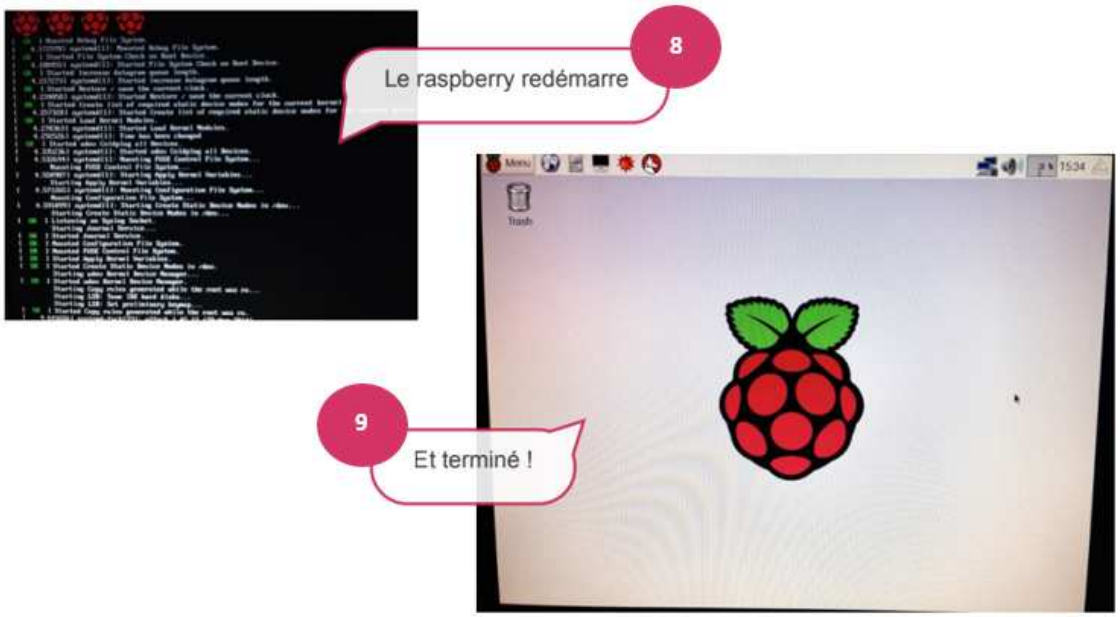
— Accéder au site officiel de la Raspberry Pi, rubrique Download. https ://www.raspberrypi.org/downloads/. Ensuite, télécharger la version NOOBS (version offline) qui est le moyen le plus simple d'installer Raspbian sur votre carte SD.

— Une fois le téléchargement terminé, décompresser le fichier .zip, puis copier les fichiers obtenus sur la carte microSD.





— Brancher le clavier, la souris, l'écran et insérer la carte microSD précédemment préparée. Une fois tout est bien connecté, alimenter la carte, le système démarre immédiatement.



— Au démarrage, des framboises apparaitront dans le coin supérieur gauche de votre écran.

c) Mise à jour du système d'éxploitation :

Ouvrir le terminal et taper les commandes suivantes pour faire la mise à jour : sudo apt-get update.

A.2 L'installation de 6LBR sur Raspberry Pi

Les étapes suivantes sont nécessaires pour que RPi 3 soit 6LBR.

Installation à partir des sources :

Les dépendances :

Sudo apt-get install git build-essential libncurses5-dev

Téléchargement de 6lbr :

git clone https ://github.com/cetic/6lbr

cd 6lbr

git submodule update –init –recursive

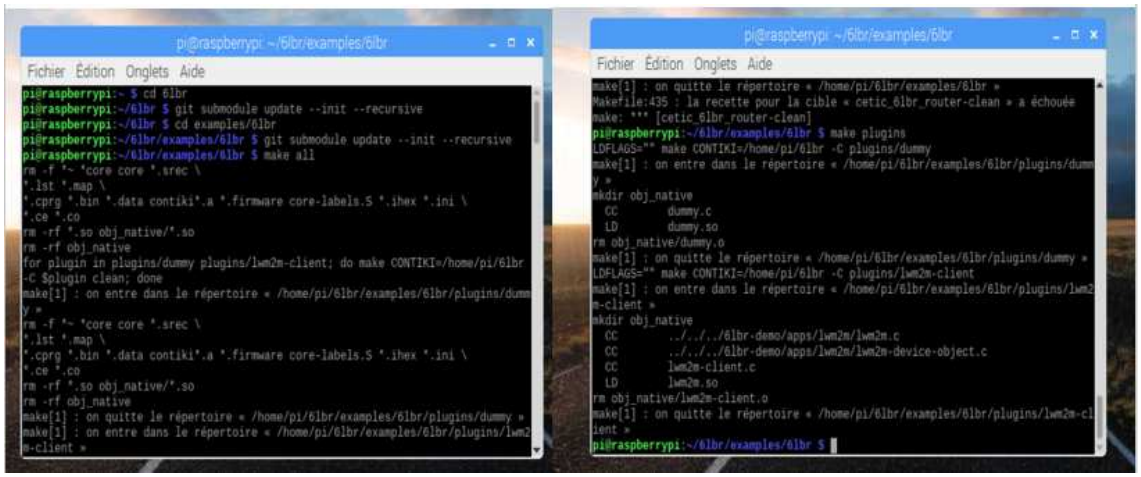
Construire les paquets :

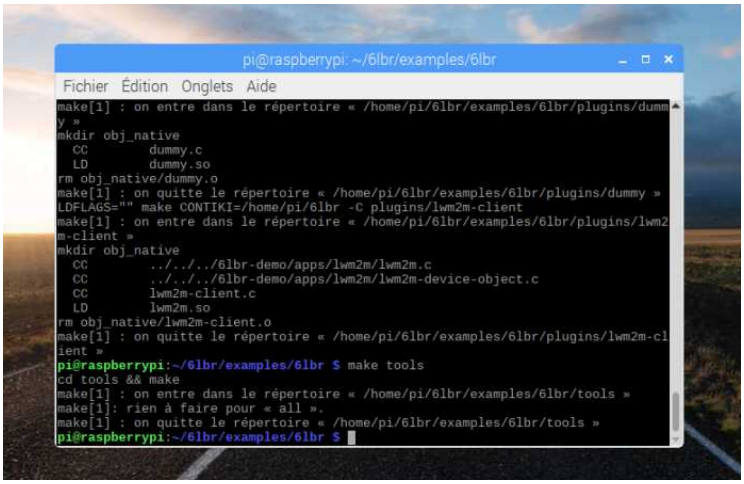
Aller au répertoire Source 6LBR, puis au dossier example 6LBR et exécuter les commandes suivantes :

make all

make plugins

make tools





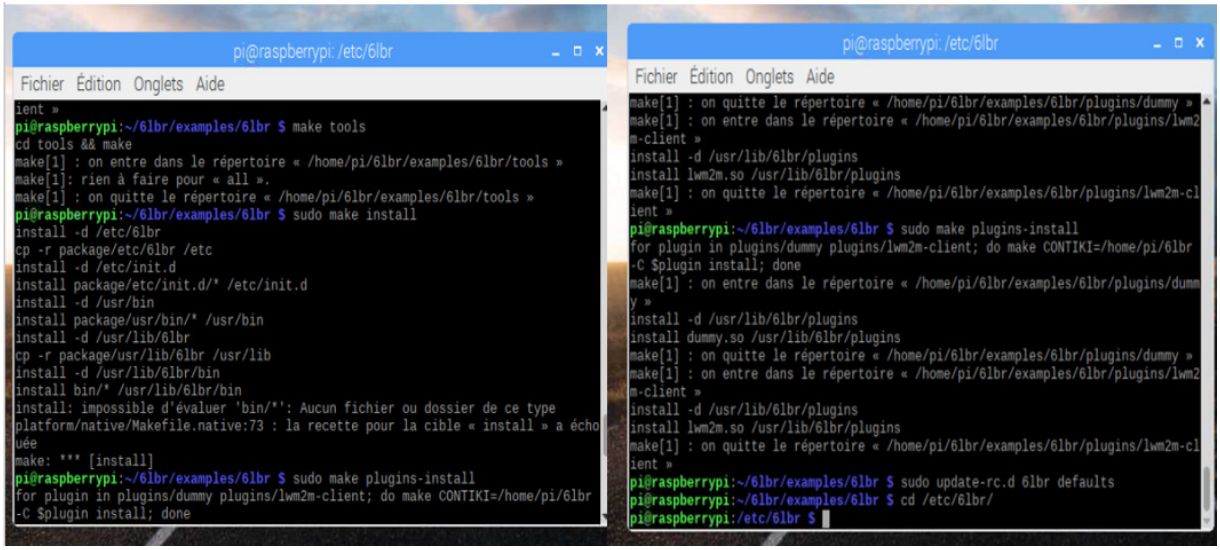
Installation de 6LBR :

Pour installer 6LBR, nous exécutons les commandes suivantes en tant qu'utilisateur root de la machine raspberry dans le répertoire example 6LBR.

make install

make plugins-install

update-rc.d 6lbr defaults



Configuration de 6LBR :

Maintenant que 6LBR est déjà installé, l'étape suivante consiste à configurer l'application 6LBR. Cette configuration est nécessaire pour appeler le bon exécutable 6LBR, pour cela, 6LBR utilise le fichier 6lbr.conf placé dans /etc/6lbr :

sudo touch /etc/6lbr/6lbr.conf

sudo nano /etc/6lbr/6lbr.conf

le fichier de configuration est le suivant :

RAW_ETH=1

BRIDGE=0

DEV_BRIDGE=br0

DEV_TAP=tap0

DEV_ETH=eth0

RAW_ETH_FCS=0

DEV_RADIO=/dev/ttyUSB0

BAUDRATE=115200

LOG_LEVEL=3

L'explication des paramètres de configuration :

MODE : est utilisé pour sélectionner le mode d'exécution de 6LBR, tel qu'un pont intelligent, routeur ou pont transparent.

RAW_ETH : sélectionne le mode de l'interface Ethernet :

0 : mode TAP , l'interface côté Ethernet utilise une interface pseudo- TAP.

1 : mode RAW , l'interface Ethernet est une interface Ethernet existante.

BRIDGE :

0 : aucun pont d'interface effectué

1 : TAP Bridge

DEV_BRIDGE : interface de pont à créer avec le nom.

DEV_TAP : périphérique TAP à utiliser en mode TAP.

DEV_ETH : interface Ethernet à utiliser en mode Ethernet brut

RAW_ETH_FCS : Indique si les trames Ethernet brutes contiennent le champ FCS .

DEV_RADIO : numéro de port du périphérique slip radio

BAUDRATE : est le débit en bauds à utiliser pour communiquer avec la radio SLIP.

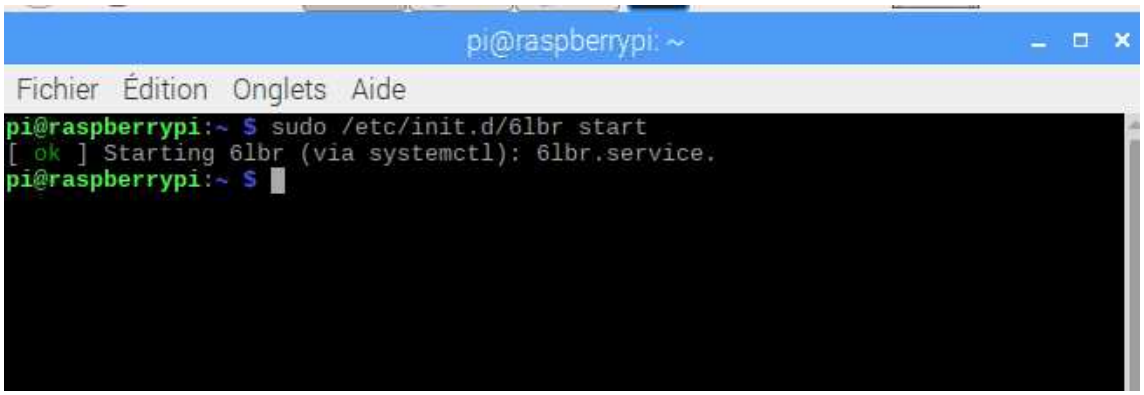
LOG_LEVEL : est un nombre (1 à 12) permettant de spécifier la verbosité du système de journalisation (log system).

Démarrage de 6lbr :

Nous avons démarré 6LBR avec les commandes suivantes :

service 6lbr start

sudo /etc/init.d/6lbr start

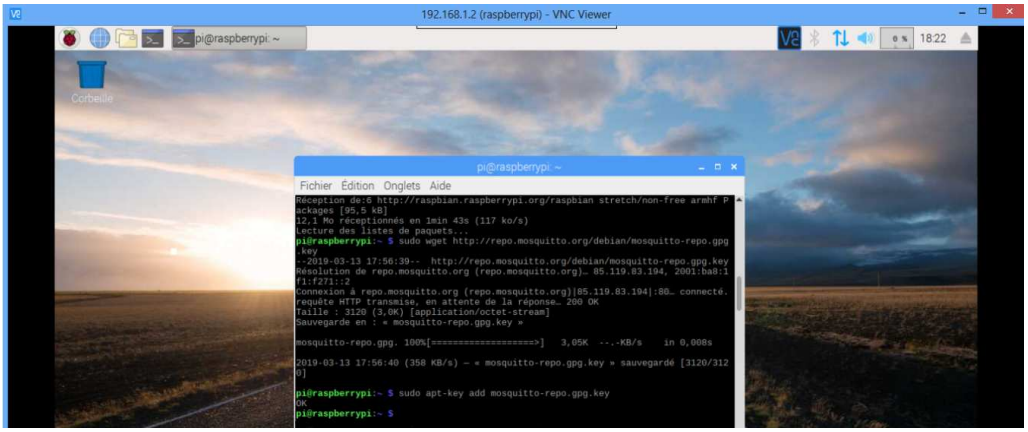


A.3 L'installation de Mosquitto

Mettre à jour la clé de signature pour apt-get

— Avant d'installer Mosquitto via apt-get, nous devons mettre à jour la clé de signature. Tout d'abord, nous avons téléchargé la clé :
sudo wget http ://repo.mosquitto.org/debian/mosquitto-repo.gpg.key

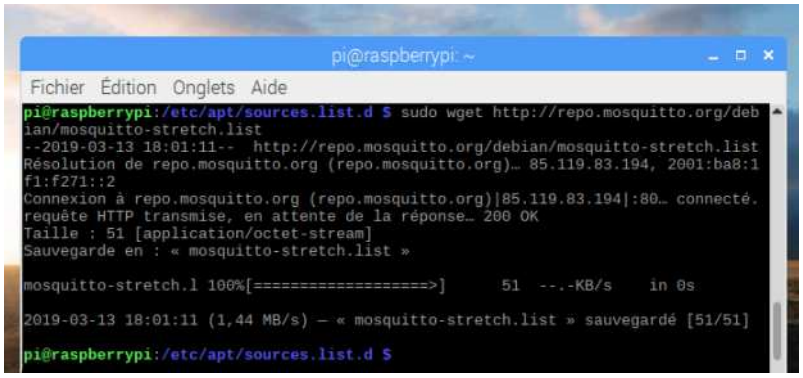
— Ensuite, nous avons ajouté cette clé à apt-get :
sudo apt-key add mosquitto-repo.gpg.key



Nous ajoutons le référentiel Mosquitto à apt-get

— Mosquitto n'est pas disponible dans les dépôts de base Rasbian. Pour ajouter le référentiel, nous avons commencé par aller dans le dossier contenant les listes de référentiels pour apt-get :
cd /etc/apt/sources.list.d/

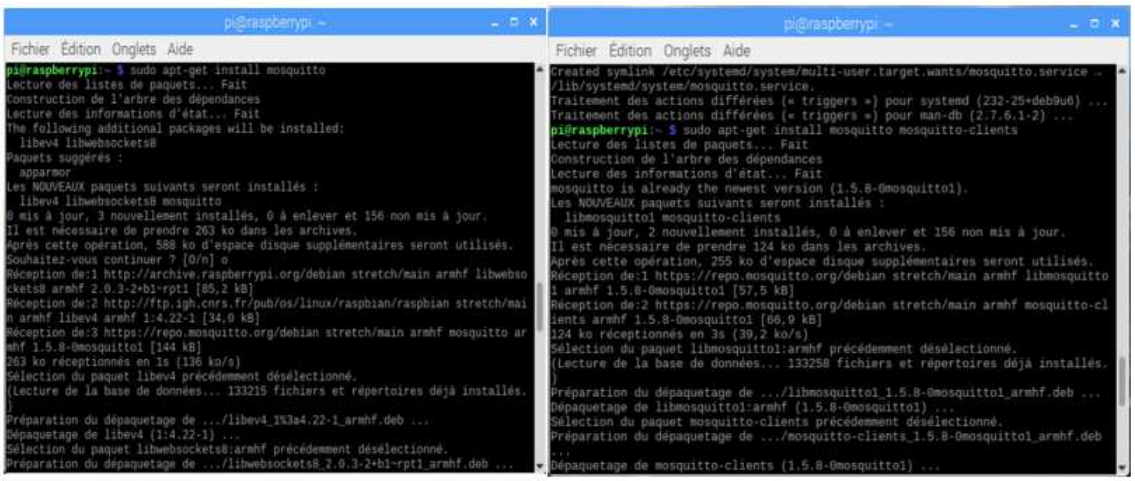
— Ensuite, nous avons téléchargé le fichier de liste de référentiels pour Mosquitto :
sudo wget http ://repo.mosquitto.org/debian/mosquitto-stretch.list



Installation de Mosquitto

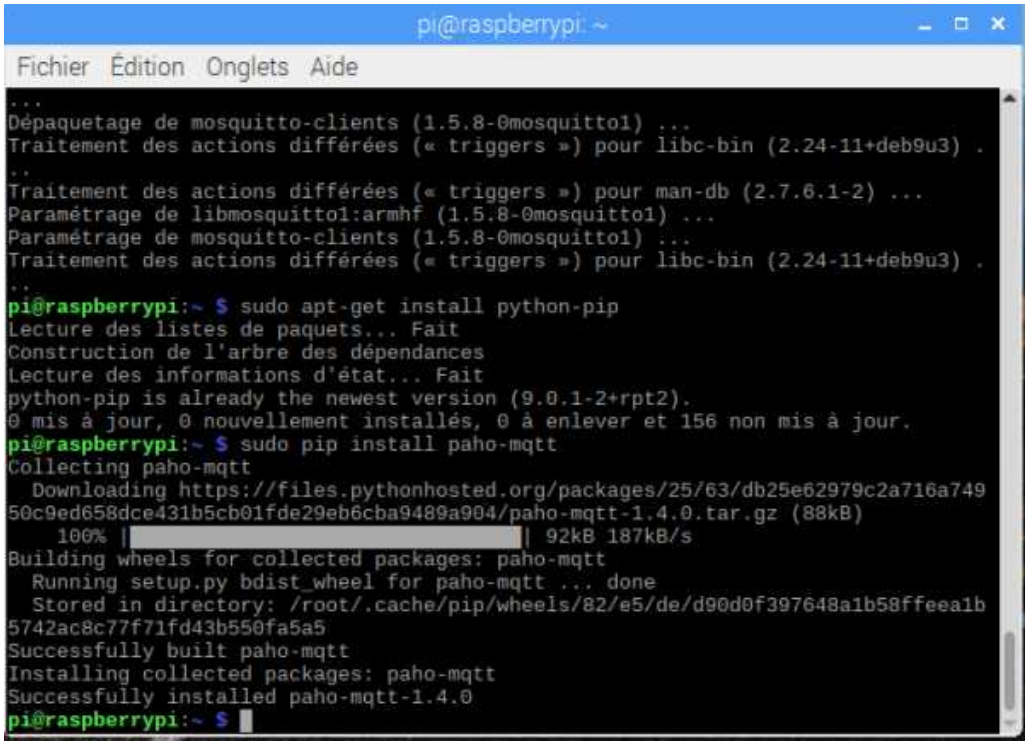
— Tout d'abord, nous mettons à jour les listes de sources d'apt-get :
sudo apt-get update

— A ce stade Nous pouvons installer le paquet [mosquitto] , qui est le broker MQTT, et les [mosquitto-clients] , que nous utiliserons plus tard pour tester l'installation.
sudo apt-get install mosquitto
sudo apt-get install mosquitto mosquitto-clients



— pour installer le client paho-mqtt, nous devons d'abord installé python-pip :
sudo apt-get install python-pip

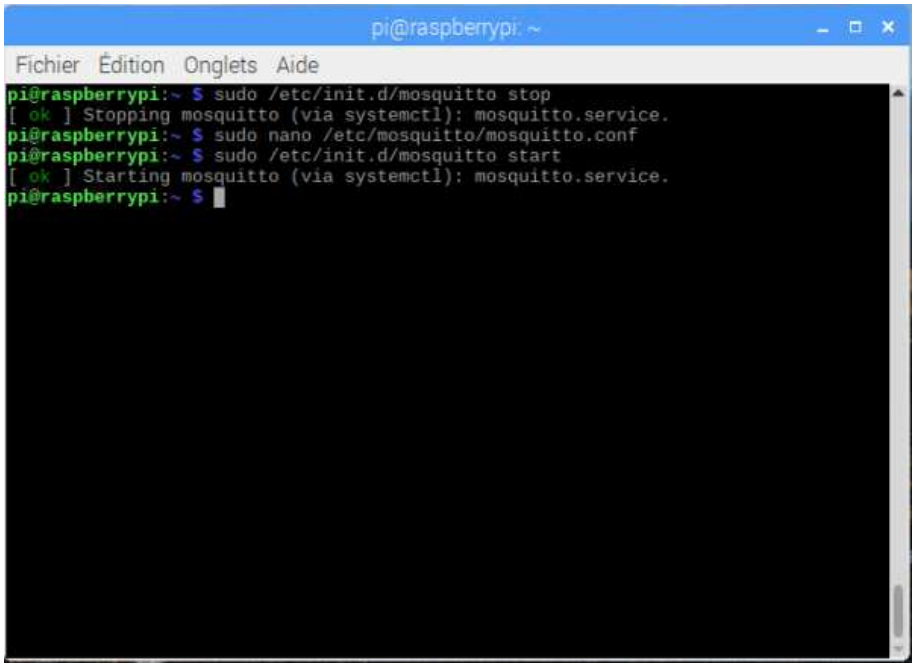
— Puis nous avons installé paho-mqtt en utilisant pip :
sudo pip install paho-mqtt



Configuration de Mosquitto

— Le fichier de configuration de Mosquitto se trouve dans [/etc/mosquitto/mosquitto.conf] :
sudo /etc/init.d/mosquito stop
sudo nano/etc/mosquito/mosquito.conf
sudo /etcinit.d/mosquito start

— La configuration de Mosquitto devrait être suffisante pour commencer à expérimenter. nous avons modifié pour enregistrer chaque type de message de journal dans le fichier journal :
Place your local configuration in /etc/mosquitto/conf.d/
#
A full description of the configuration file is at
/usr/share/doc/mosquitto/examples/mosquitto.conf.example
pid_file /var/run/mosquitto.pi
persistence true
persistence_location /var/lib/mosquitto/
Save all log in file
log_dest file /var/log/mosquitto/mosquitto.log
log_type all
log_timestamp true
include_dir /etc/mosquitto/conf.d



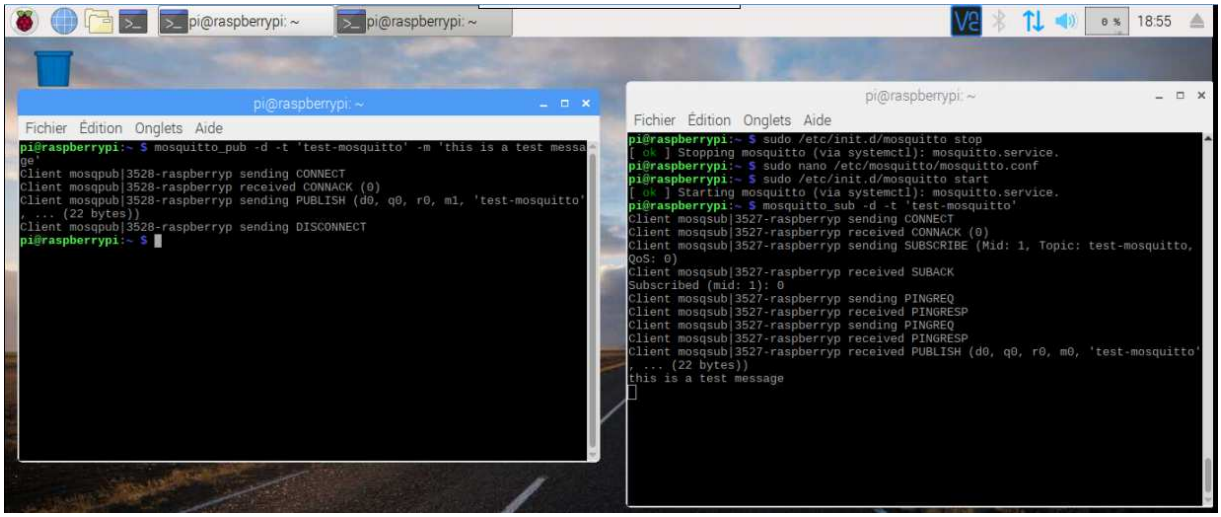
Test Mosquitto

tester notre installation. Nous allons utiliser 2 terminaux. L'un s'abonnera au topic "test-mosquitto", l'autre publiera un message à ce topic. Le test réussira si le message envoyé par l'éditeur est enregistré sur le terminal d'abonné.

— Pour démarrer l'abonné, nous avons exécuté la commande [mosquitto_sub] dans le premier terminal. L'option [-d] est utilisée pour activer le mode débogage, et [-t] nous permet de spécifier le topic :
mosquitto_sub -d -t 'test-mosquitto'

— Pour publier un message, nous avons utilisé la commande [mosquitto_pub] avec les mêmes options et nous ajoutons un message avec [-m] :
mosquitto_pub -d -t 'test-mosquitto' -m 'This is a test message'

— Nous avons réussi de voir le message de test dans le premier terminal ce qui signifie que l'abonné a reçu le message de test



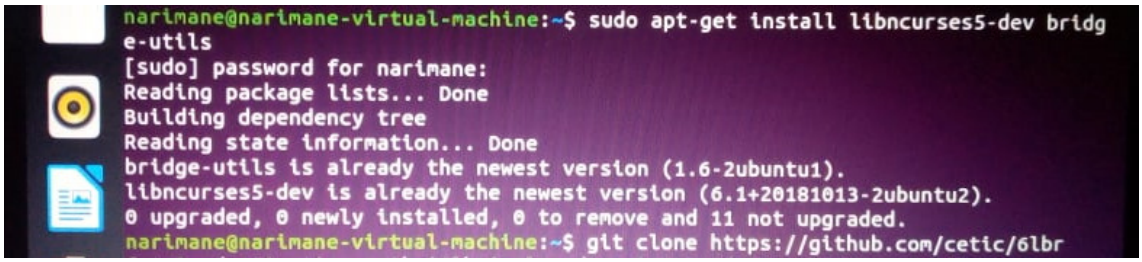
A.4 Installation de 6LBR sur une machine linux

6LBR peut être exécuté sur un hôte Linux.

Nous avons installé VMware workstation Player qui est un outil de virtualisation pour l'exécution de plusieurs systèmes d'exploitation en tant que machines virtuelles sur un seul ordinateur (Linux ou Windows). et nous avons installé Ubuntu (version 19.0) sur VMware workstation Player. Après avoir installé Ubuntu, nous avons commencé à installer l'application 6lbr sur ubuntu.

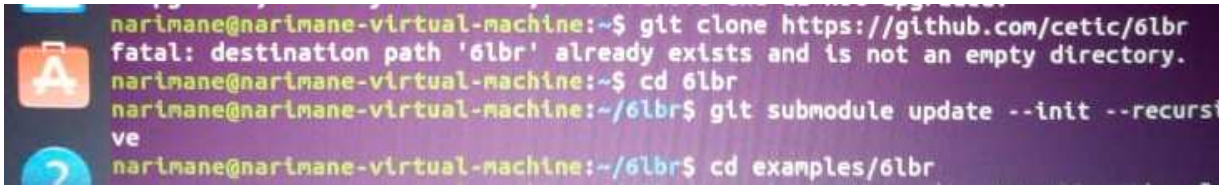
Les étapes de l'installation :

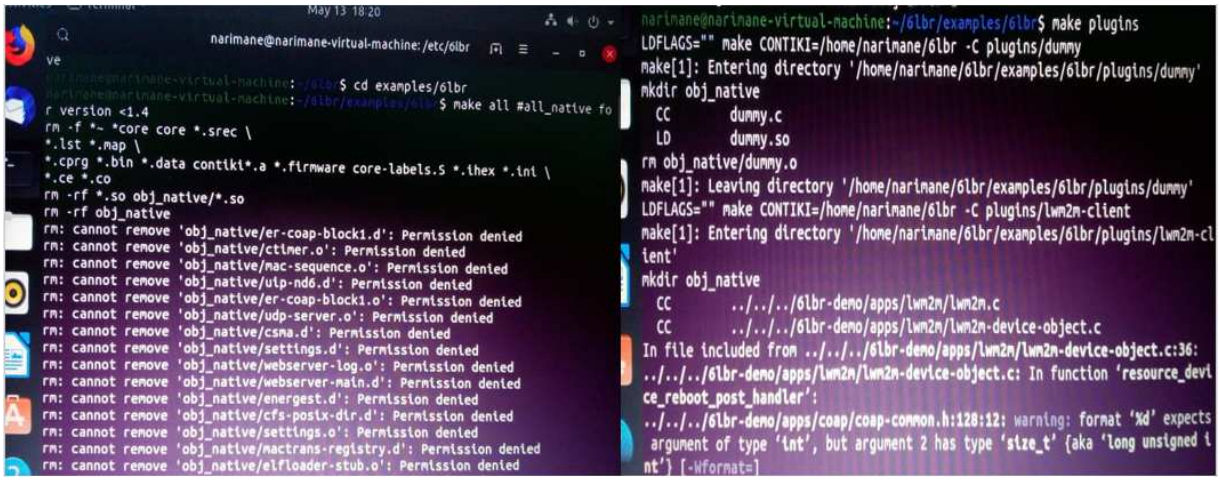
— Nous avons installé les conditions préalables de l'installation :
apt-get install libncurses5-dev bridge-utils

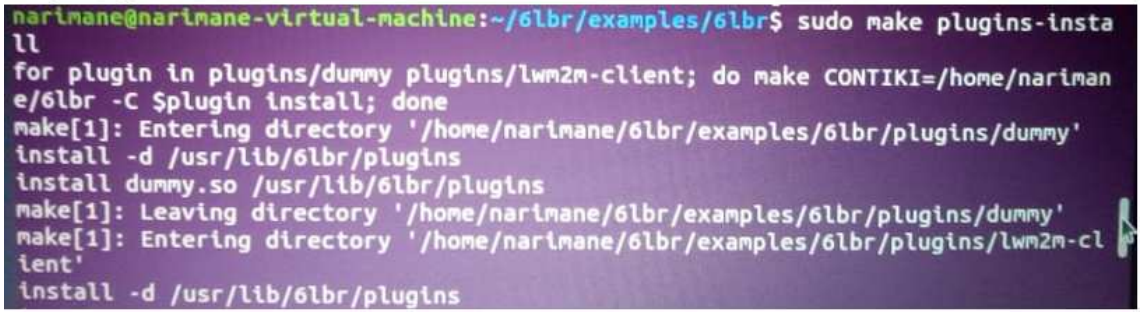


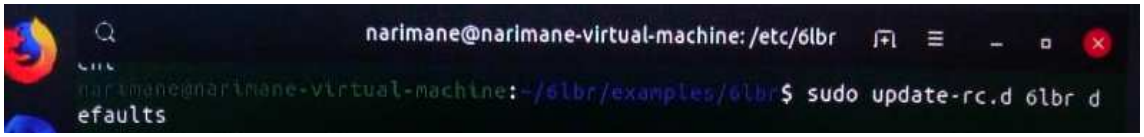
— Construction et installation de 6LBR :
Nous avons obtenu une copie de 6LBR du site github, et nous l'avons compilée et installée :
git clone https ://github.com/cetic/6lbr
cd 6lbr
git submodule update –init –recursive
cd examples/6lbr
make all #all_native for version < 1.4

make plugins
make tools
sudo make install
sudo make plugins-install
sudo update-rc.d 6lbr defaults





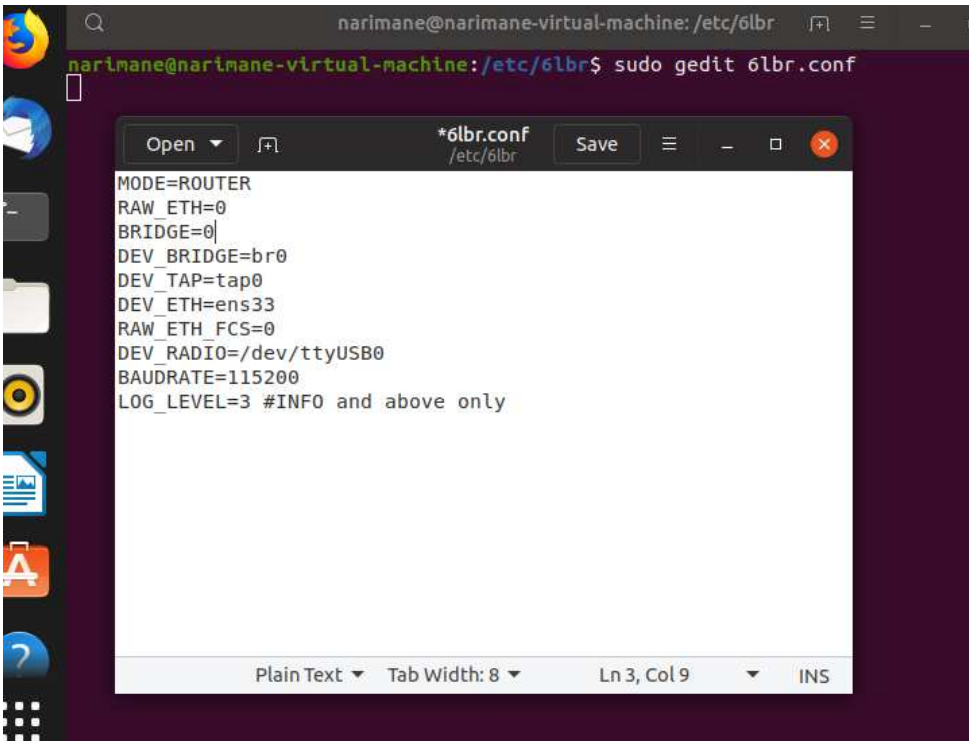




Configuration de 6LBR :

— Nous avons crée un fichier de configuration /etc/6lbr/6lbr.conf avec le contenu suivant :

RAW_ETH=0
BRIDGE=0
DEV_BRIDGE=br0
DEV_TAP=tap0
DEV_ETH=eth0
RAW_ETH_FCS=0
DEV_RADIO=/dev/ttyUSB0
BAUDRATE=115200
LOG_LEVEL=3



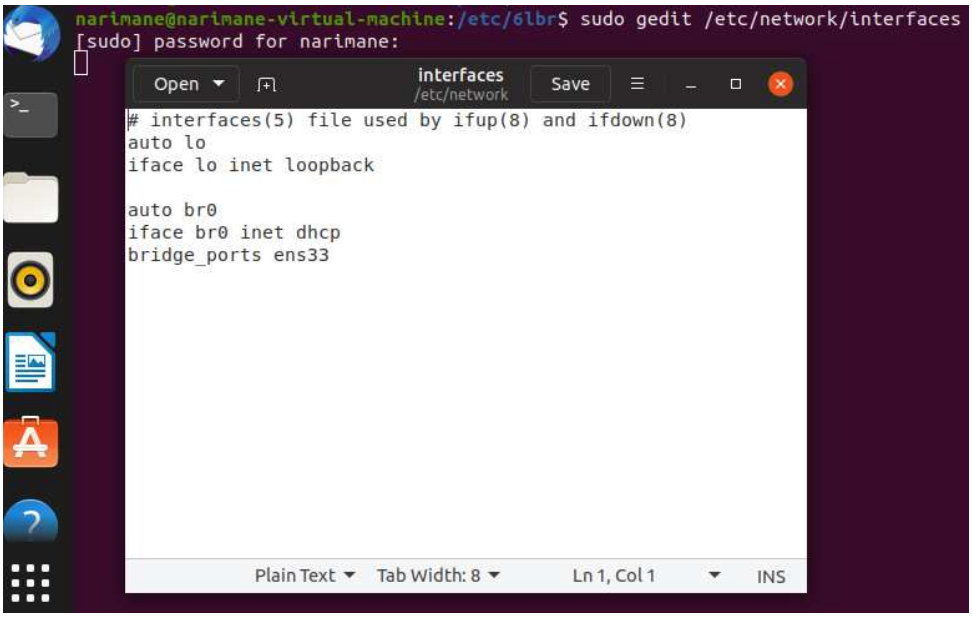
Dans cette étape, slip radio est connecté au Raspberry Pi via USB au port / dev / ttyUSB0 à 115200 bps, et il faut que nous redémarrons 6lbr en procédant comme suit :

sudo service 6lbr restart

la commande suivantes permet de vérifier si 6lbr fonctionne correctement :

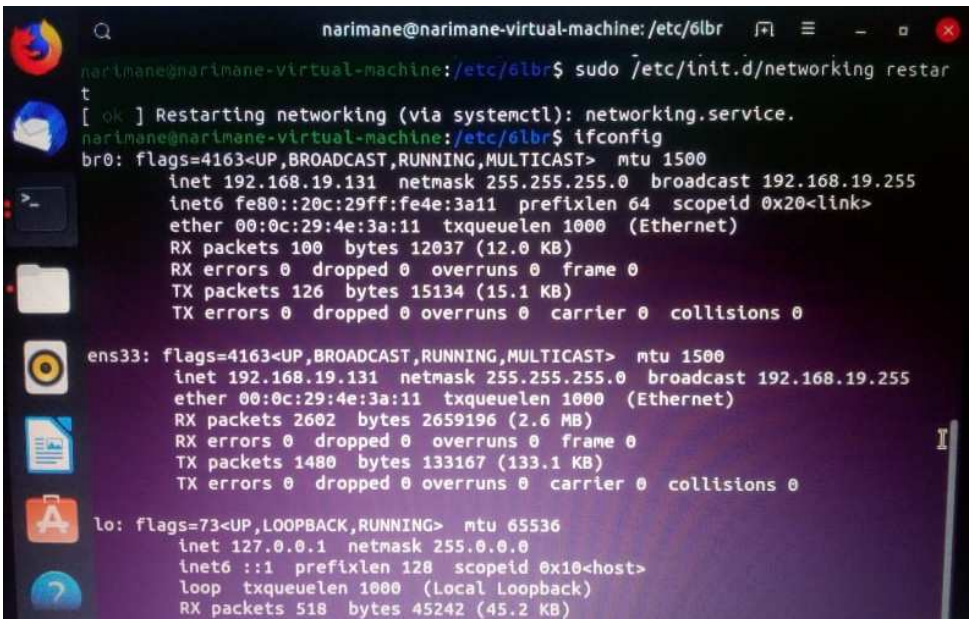
sudo service 6lbr status

— la configuration des interfaces dans le dossier (/ etc / network / interfaces) :
sudo gedit /etc/network/interfaces
le contenu dans le dossier est comme suit :
auto lo
iface lo inet loopback
auto br0
iface br0 inet dhcp
bridge_ports ens33



— Nous redémarrons l'interface réseau en tapant cette commande :
sudo /etc/init.d/networking restart

— Nous affichons les informations des interfaces réseau :
Ifconfig



Annexe B

Installations des outils utilisée

B.1 La machine virtuelle

La machine virtuelle est un environnement d'applications ou de systèmes d'exploitation qui exécute un ou plusieurs systèmes d'exploitation sur le même ordinateur sans redémarrer.

Dans notre projet, nous allons utiliser VMware Player Worstation 15.0 comme une application de virtualisation recommandée par la communauté Contiki pour exécuter l'environnement de développement "instant Contiki". Nous allons installer VMware Workstation Player sous Microsoft Windows. Il est aussi possible de l'installer sous lunix.

VMWare Player est un outil gratuit pour des usages non commerciaux et facile à installer. Le lien de téléchargement sur le site official est suivant :

https ://my.vmware.com/en/web/vmware/free#desktop_end_user_computing/vmware_workstation_player/15_0.

B.2 Instant Contiki :

Est un outil téléchargable gratuitement. Il se décline en deux version : " instant Contiki2.7 " et "Instant Contiki 3 ". Le lien de téléchargement sur le site officiel est le suivant :

https ://sourceforge.net/projects/contiki/files/Instant%20Contiki/ .

Après l'installation de VMWare player et la décompression de " Instant Contiki", ouvrir le répertoire ou on a décompressé "l'instant Contiki", puis lancer l'installation de l'instant Contiki en cliquant sur le fichier "Instant_Contiki_Ubuntu_12.04_32-bit.vmx".

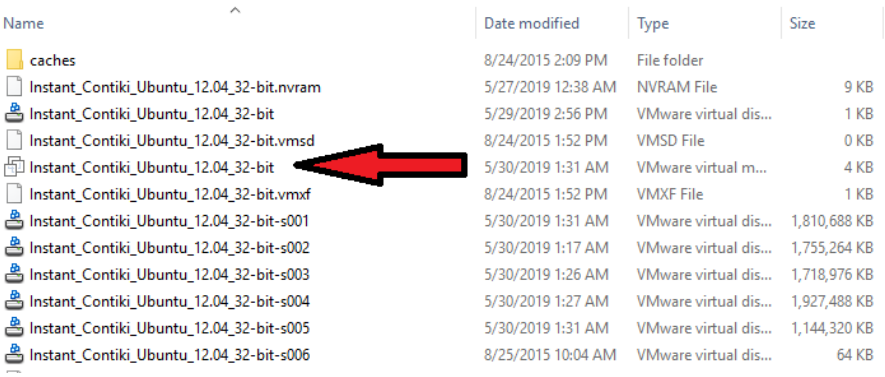


Figure B.1 – Installer InstantContiki2.7.

Après la fin de l'installation en lançant VMWare player, puis lancer la machine virtuelle intitulée "InstantContiki2.7 ", apparaît à gauche de la fenêtre dans la liste des machines Virtuelle disponibles.

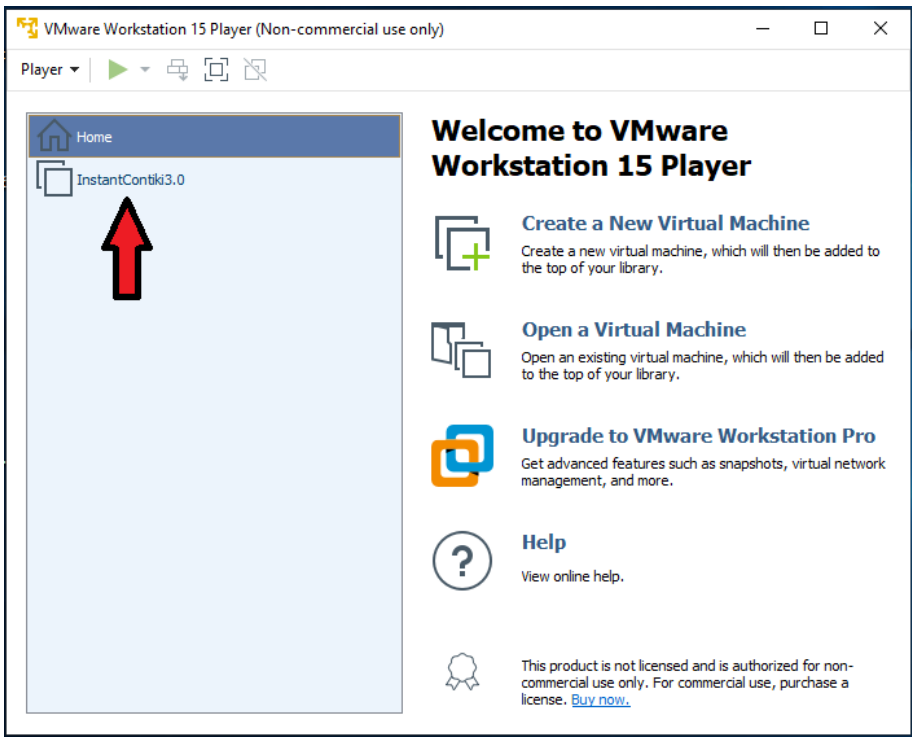
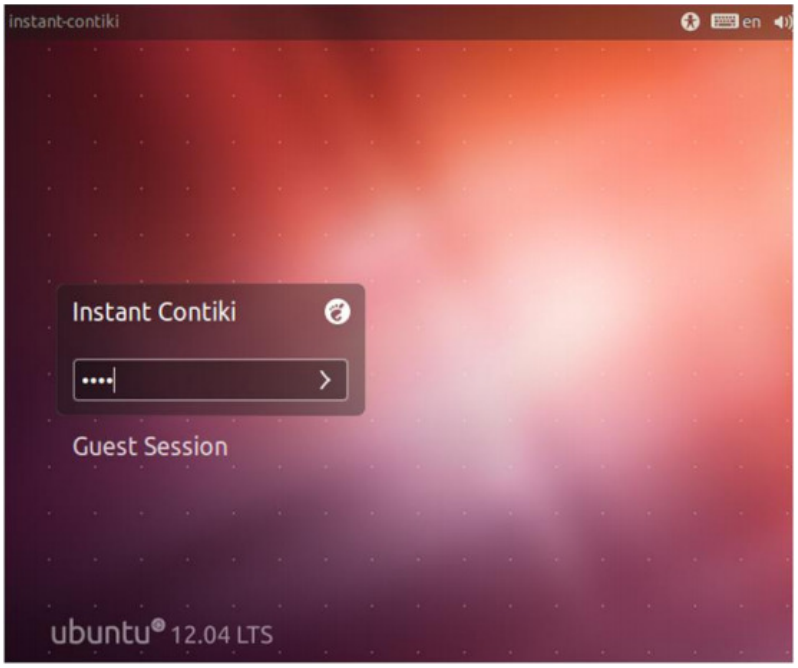


Figure B.2 – Lancer Instant Contiki.

et aprés la fin de l'installation connectez-vous à " Instant Contiki ". Le mot de passe est «user». Et enfin le bureau "d'instant Contiki " s'affiche.



Lancer l'emulateur Cooja :

1. Soit directement à partir de l'icône qui se trouve sur le bureau Ubuntu virtuel.
2. Soit en ouvrant un terminal puis :

a) user@instant-contiki :~$ cd contiki/tools/cooja

b) ant run (La commande ant run permet de lancer Cooja).

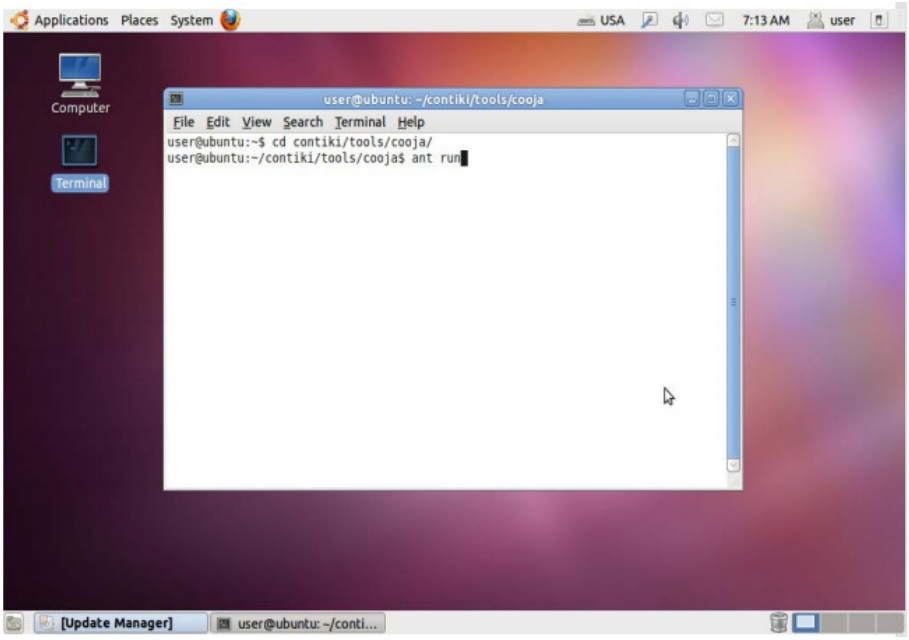


Figure B.3 – L'interface du lancement de cooja.

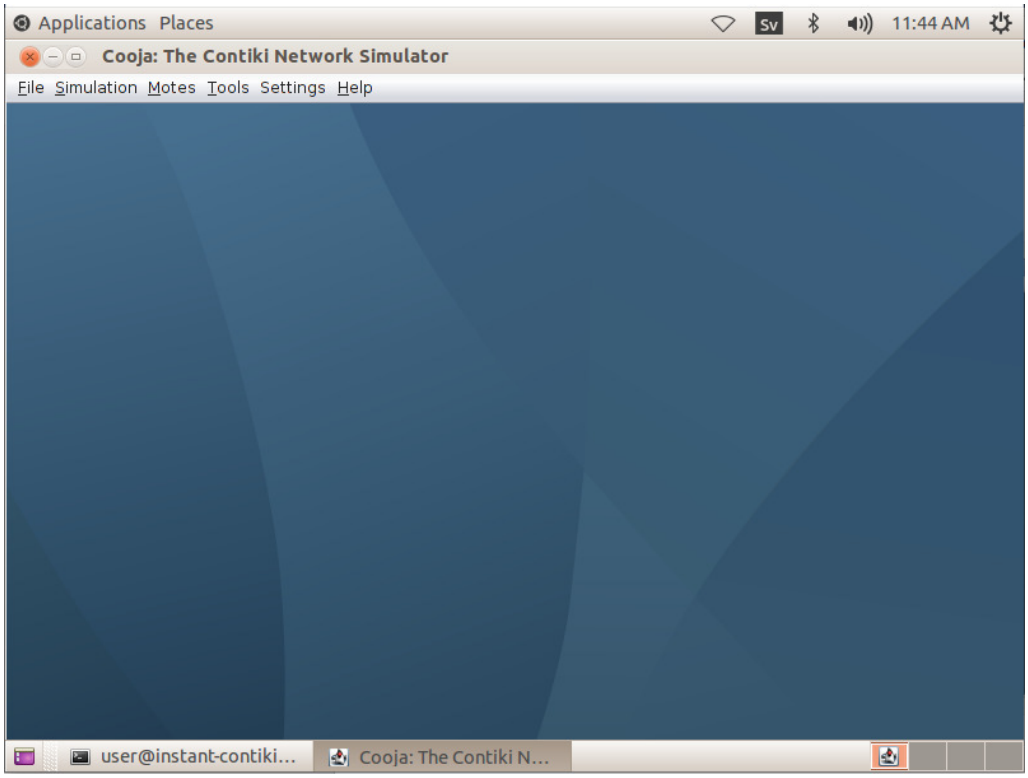


Figure B.4 – L'interface du simulateur de cooja.

Résumé

Le fonctionnement de l'Internet des objets est basé sur deux concepts essentiels, à savoir le protocole IPv6 et la présence d'une couche d'adaptation de paquets (par ex., IPv6 over Low power Wireless Personal Area Networks -6LoWPAN-). Le protocole IPv6 est indispensable en raison de l'explosion du nombre d'objets connectés dans le monde ces dernières années. Il permet d'identifier chaque objet connecté avec une adresse unique. Quant à la couche d'adaptation, son rôle est d'adapter le format et la taille des paquets dans les deux sens, IEEE 802.15.4 <-> IPv6, dans le cas par exemple de l'exploitation à distance des données collectées par un Réseau de Capteurs Sans Fil (RCSF). 6LoWPAN Border Router (6LBR) est une solution permettant de compresser les grandes trames de paquets IPv6 afin de les rendre compatibles aux trames IEEE 802.15.4. Il s'agit d'une passerelle entre les nœuds utilisant la technologie IEEE 802.15.4 et Internet.

L'objectif principal de ce mémoire est d'accéder aux données des nœuds d'un RCSF supportant 6LowPAN, en utilisant le protocole de messagerie Message Queuing Telemetry Transport (MQTT) qui représente un protocole de messagerie idéal pour les communications IdO, et qui est basé sur le principe souscription / publication. A cet effet, nous avons réalisé une plate-forme matérielle constituée d'une carte Raspberry Pi 3, d'un RCSF constitué de deux nœuds capteurs TelosB et d'un smartphone. L'échange de données entre la Raspberry Pi 3 et le RCSF se fait grâce à la solution 6LBR installée sur la carte Raspberry. Les données collectées par les nœuds capteurs sont publiées dans le broker Mosquitto installé sur la Raspberry. Le smartphone joue le rôle du souscripteur aux données publiées par les TelosB. Il recevra ces données via Internet.

Mots clès : Réseaux de Capteurs Sans Fil, Internet Des Objets, 6LBR, IPv6, 6LoWPAN, MQTT, IEEE 802.15.4.

Abstract

The operation of the Internet of Things is based on two essential concepts, namely the IPv6 protocol and the presence of a packet adaptation layer (eg, 6LowPAN). The IPv6 protocol is indispensable because of the explosion of the number of connected objects in the world lately. It identifies each connected object with a unique address. As for the adaptation layer, its role is to adapt the packet size in both directions, IEEE 802.15.4 <-> IPv6, in the case for example of the remote exploitation of the data collected by a network of Wireless Sensors (WSN). Indeed, the size of IEEE 802.15.4 packets is much smaller than the size of IPv6 packets. 6LoWPAN Border Router (6LBR) is a solution for compressing large frames of IPv6 packets to make them compatible with IEEE 802.15.4 frames. It is a gateway or bridge between nodes using IEEE 802.15.4 technology and the Internet.

The main purpose of this memory is to access data from the nodes of a RCSF supporting 6LowPAN, using the Message Queuing Telemetry Transport (MQTT) messaging protocol which represents an ideal messaging protocol for IoT communication, and which is based on the principle subscription / publication. To this end, we have created a hardware platform consisting of a Raspberry Pi 3 card, a RCSF consisting of two TelosB sensor nodes and a smartphone. The data exchange between the Raspberry Pi 3 and the RCSF is done thanks to the 6LBR solution installed on the Raspberry card. The data collected by the sensor nodes are published in the Mosquitto broker installed on the Raspberry. The smartphone plays the role of the subscriber to the data published by TelosB. He will receive this data via the Internet.

Keywords: Network of Wireless Sensors, Internet of, 6LBR, IPv6, 6LoWPAN, MQTT, IEEE 802.15.4 .