



REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE  
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique  
Université Mohamed Khider – BISKRA  
Faculté des Sciences Exactes, des Sciences de la Nature et de la Vie  
**Département d'informatique**

N° d'ordre : GLSD 6 /M2/2019

## Mémoire

Présenté pour obtenir le diplôme de master académique en

# Informatique

Parcours : **Génie Logiciel et Système Distribué**

---

# Modélisation et vérification d'un protocole de communication pour l'internet des objets

---

Par :

**Dhahoua Oumaima**

Soutenu le 07 juillet 2019, devant le jury composé de :

Bennaoui Hammadi

Prof

Président

Djaber khaled

M.A.A

Rapporteur

Hmidi Zohra

M.A.A

Examineur

## **Dédicaces**

### *À nos très chers parents*

*Nous vous devons ce que nous sommes aujourd'hui, grâce à votre amour, votre patience et vos innombrables sacrifices.*

*Que ce modeste travail, soit pour vous une petite compensation et reconnaissance pour tout ce que vous avez fait.*

*Que Dieu, vous préserve et vous procure santé et longue vie afin que nous puissions à notre tour vous combler.*

### *À nos très chers frères et sœurs*

*Aucune dédicace ne pourrait exprimer assez profondément ce que nous ressentons envers vous.*

*Nous vous dirons tout simplement, un grand merci, nous vous aimons.*

### *À nos très chers ami(e)s*

*En témoignage de l'amitié sincère qui nous lie et les bons moments passés ensemble.*

*Nous vous dédions ce travail en vous souhaitons un avenir radieux et plein de réussites.*

## *Remerciement*

*Avant tout, notre sincère louange à ALLAH le tout puissant qui nous a donné la fois, la volonté, la sante et la patience afin d'accomplir ce m´mémoire.*

*Nous remercions les plus distingués à Mr. djaber khaled qui nous a honorés par son encadrement, par sa présence toujours avec nous, pour sa direction, son orientation, sa modestie, ses conseils et toutes ces remarques constructives pour le bon déroulement de notre projet.*

*Nous remercions beaucoup l'ensemble des enseignements du d´département d'informatique pour la formation qu'ils nous ont assurée tout le long de notre cursus universitaire, Enfin, nous adressons aussi un grand merci `à tous les personnels du d´département d'informatique.*

## Résumé

L'Internet des Objets (IoT) rend les objets qui nous entourent intelligents en leur offrant la faculté de communiquer entre eux ou avec le nuage (cloud). L'Internet des objets (IoT) est l'interconnexion de périphériques physiques, de véhicules et d'autres objets qui consiste en un système embarqué avec capteurs/actionneurs et un réseau permettant de collecter et d'échanger des données. L'IOT permet aux objets d'être détectés et / ou contrôlés à distance sur l'infrastructure réseau existante, créant des opportunités de plus grande intégration du monde physique dans des systèmes informatiques, et aboutir à l'amélioration de la précision, de l'efficacité et des avantages économiques. L'IoT est un système en rapide augmentation et une technologie prometteuse qui devient de plus en plus présente dans nos vies quotidiennes. En outre, la technologie est un exemple de la classe plus générale de cyber-physique systèmes, qui englobe également des technologies telles que les réseaux intelligents, les maisons intelligentes et Villes intelligentes. Compte tenu du développement rapide des technologies IoT et de l'augmentation significative du nombre d'appareils connectés, vue l'ensemble des objectifs, l'architecture, les défis, les applications. et le marché du système IoT et les protocoles.

Dans ce rapport, nous avons introduit le protocole RPL, afin d'étudier ces performance et de garantir son bon fonctionnement ,pour atteindre cet objectif ,nous avons utilisé UPAAL pour modéliser ce protocole et vérifier quelques propriétés nécessaire pour un bon fonctionnement,les résultats obtenus sont par la suite compares avec ceux obtenus par le simulateur Cooja ,afin de valider notre travail.

Mot clé :

Internet des objet(IoT), le protocole RPL ,cooja, Upaal ,analyse de performance ,protocole de routage.

# Table des matières :

## Chapitre 1 : généralité sur l'internet des objets

1.1. Introduction.....	1
1.2. Définition d'internet des objets.....	3
1.3 Définition d'un objet connecté.....	3
1.4. Types d'objets.....	3
1.5. Infrastructure de communication.....	4
1.6. L'évolution d'IOT et son impact dans le monde.....	5
1.7. Domaine d'application d'IOT.....	6
1.7.1. Les bâtiments intelligents :.....	7
1.7.2. L'aérospatiale et l'aviation :.....	7
1.7.3. La santé et les technologies médicales (Health care).....	8
1.7.4. L'agriculture.....	8
1.8. Architecture de l'internet des objets.....	9
1.8.1. Le domaine du réseau d'objets.....	9
1.8.2. Le domaine du réseau cœur d'accès.....	10
1.8.3. Le domaine des applications M2M et applications clientes.....	10
1.9. Les normes et les standards pour l'IOT.....	10
1.9.1. Les technologies de courte portée.....	11
1.9.2. Les technologies de longue portée.....	12
1.10. Les avantages de l'IOT.....	13
1.11. Désavantages de l'IOT.....	14
1.12. Les protocoles d'IOT.....	15
1.12.1. Le protocole IEEE 802.15.4.....	15
1.12.2. Le protocole MQTT.....	16
1.12.3. Le Protocole CARP.....	17
1.12.4. Le Protocole RPL.....	18
1.12.5. Le protocole CORPL.....	18
1.13 Conclusion.....	19

## Chapitre 2 : le protocole de routage RPL

2.1. Introduction.....	20
2.2. Définition de protocole RPL.....	20
2.3. La topologie du RPL.....	21
2.3.1. Structure DAG et DODAG.....	21
2.3.2. Identifiant RPL.....	22
2.3.3. Fonction objective.....	25
2.4. Le modèle réseau.....	26
2.5. Message de contrôle RPL.....	26
2.5.1. DIO (DODAG Information Object).....	26

2.5.2.DIS (DODAG Information Sollicitation).....	27
2.5.3.DAO (DODAG Advertisement Object) .....	27
2.5.4.DAO-Ack (DAO-Acknowledgment).....	27
2.6.Algorithme Trickle .....	28
2.7.Construction de DODAG .....	30
2.8.Maintenance de la topologie.....	31
2.8.1.Réparation globale.....	31
2.8.2.Réparation locale .....	31
2.9.Les modes de fonctionnements du protocole RPL .....	32
2.9.1.Mode non storing .....	32
2.9.2.Mode stroing.....	34
2.10.Les paradigmes de communication .....	35
2.11.Travaux d'optimisation de RPL .....	36
2.11.1.Optimisation de la fonction objective .....	36
2.11.2.Optimisation de la topologie logique.....	37
2.11.3.Optimisation de Trickle Timer.....	38
2.12.Conclusion .....	38

### **Chapitre 3:modélisation de protocole de routage RPL**

3.1.Introduction .....	39
3.2.Présentation du Contiki/ Cooja .....	39
3.3.Simulateur Cooja.....	40
3.4.L'interface de simulateur Cooja.....	40
3.5. Les différents outils nécessaires.....	41
3.6. Téléchargement et installation de Contiki.....	42
3.7.Démarrage de l'application .....	42
3.8.Les paramètres estimés .....	44
3.9.Paramètre de simulation .....	46
3.10. Le rôle du système Cooja / Contiki dans notre projet .....	46
3.11. Avantage de la simulation.....	46
3.12. Limité de la simulation (Puissance et performance du simulateur).....	46
3.13. L'outil UPPAAL.....	46
3.14. Syntaxe d'UPPAAL .....	47
3.15. Description globale de l'outil UPPAAL .....	51
3.16. La modélisation du protocole RPL .....	51
3.17. Conclusion .....	61

### **Chapitre 4 : vérification et analyse de performance**

4.1. Introduction .....	62
4.2. La simulation de RPL dans UPAAL .....	62

4.3. La vérification RPL dans UPAAL :.....	<b>64</b>
4.4. Résultat de l'analyse de performances de RPL dans UPAAL.....	<b>65</b>
4.5. Comparaison des résultats avec le simulateur Cooja.....	<b>69</b>
4.6. Discussion.....	<b>71</b>
4.7. Conclusion.....	<b>71</b>

## LISTE DES FIGURES :

<b>Figure1.1</b> Jonction entre le monde physique et le monde numérique.....	1
<b>Figure1.2</b> : L'IOT connecte des objets en utilisant des capteurs et Internet.....	2
<b>Figure1.3</b> : une nouvelle dimension pour l'ioT.....	2
<b>Figure 1.4</b> : Application et systèmes connectés.....	4
<b>Figure1.5</b> : Infrastructure réseau communément mise en œuvre pour les objets connectés	5
<b>Figure 1.6</b> : la prévision « cisco» sur les objets connectés par personne.....	6
<b>Figure 1.7</b> : Les bâtiments intelligents.....	7
<b>Figure 1.8</b> : l'impacte de l'ido sur la santé.....	8
<b>Figure 1.9</b> : les domaines d'application de l'ido.....	9
<b>Figure 1.10</b> : Architecture de L'internet des Objets.....	10
<b>Figure 1.11</b> : la technologie wifi.....	11
<b>Figure 1.12</b> : la technologie Bluetooth.....	11
<b>Figure 1.13</b> : la technologie ZigBee.....	12
<b>Figure 1.14</b> : la technologie SigFox et LoRa.....	13
<b>Figure 1.15</b> : la topologie en étoile.....	16
<b>Figure 1.16</b> : topologie paire à paire.....	16
<b>Figure 1.17</b> : Schéma de fonctionnement de MQTT.....	17
<b>Figure 2.1</b> : la structure DAG et DODAG.....	21
<b>Figure2.2</b> : deux instances sur un seul DODAG physique.....	22
<b>Figure 2.3</b> : représentation du rang.....	23
<b>Figure2.4</b> : DODAGs flottants et mis à la terre.....	24
<b>Figure 2.5</b> : les messages du contrôle dans le protocole RPL.....	27
<b>Figure 2.6</b> :l'algorithme trickle.....	29
<b>Figure 2.7</b> : construction de DODAG.....	31

<b>Figure 2.8</b> : la 1ere étape pour construire la table de routage source.....	32
<b>Figure 2.9</b> : la table de routage.....	33
<b>Figure 2.10</b> : en-tête de routesource.....	33
<b>Figure 2.11</b> : mode storing et non storing dans le protocole RPL.....	34
<b>Figure 2.12</b> : Flux de trafic pris en charge par RPL.....	36
<b>Figure3.1</b> : capture d'écran de l'interface du simulateur COOJA.....	41
<b>Figure 3.2</b> : Création d'un MOTE.....	43
<b>Figure 3.3</b> : utilisation du PowerTracker.....	44
<b>Figure3.4</b> : utilisation de l'outil "Mot output".....	45
<b>Figure3.5</b> : l'interface graphique de l'outil UPPAAL.....	47
<b>Figure 3.6</b> : partie déclaration dans UPAAL.....	48
<b>Figure3.7</b> : interface de location.....	49
<b>Figure3.8</b> : l'interface de transition.....	50
<b>Figure3.9</b> : le model du de root.....	54
<b>Figure3.10</b> : le model de nœud dans le cas Non__Connect.....	56
<b>Figure 3.11</b> : le le model de nœud dans le cas Non__Connect.....	57
<b>Figure 3.12</b> : le modèle nœud dans le cas « Connect » et le cas de « vielle ».....	58
<b>Figure 3.13</b> : le modèle de nœud dans le cas « Connect ».....	59
<b>Figure 3.14</b> : le modèle de nœud dans le cas « Connect ».....	60
<b>Figure 4.1</b> :l'initialisation des nœuds.....	62
<b>Figure 4.2</b> : le cas de veille.....	62
<b>Figure 4.3</b> : envoyer le message DIO par le root.....	63
<b>Figure 4.4</b> : envoyer le message DAO par le nœud N_1.....	63
<b>Figure 4.5</b> : envoyer le message DAO_ACK par le Root.....	63
<b>Figure 4.6</b> : envoyer le message_DATA par le nœud N_1.....	64

<b>Figure 4.7:</b> cas d'annulation un nœud.....	64
<b>Figure 4.8:</b> vérifier l'initialisation des nœuds.....	65
<b>Figure 4.9:</b> vérifier la connectivité des nœuds.....	66
<b>Figure 4.10:</b> vérifier la vielle des nœuds.....	67
<b>Figure 4.11 :</b> énergie Vs nombre de nœuds dans UPAAL.....	67
<b>Figure 4.12 :</b> TX vs nombre de nœud dans UPPAL.....	68
<b>Figure 4.13 :</b> RX vs nombre de nœud dans UPPAL.....	68
<b>Figure 4.14 :</b> comparaison de la consommation d'énergie vs nombre ..... de nœud	69
<b>Figure 4.15 :</b> comparaison de la Transmission vs nombre de nœud.....	70
<b>Figure 4.16 :</b> comparaison de la réception vs nombre de nœud.....	70

## **Glossaire des acronymes :**

<b>IOT</b>	Internet Of Things
<b>Cooja</b>	Contiki Os Java Simulator
<b>6lowPAN</b>	IPv6 over Low Power Wireless Personal Area Networks
<b>DODAG</b>	Destination Oriented Directed Acyclic Graph
<b>L'IETF</b>	Internet Engineering Task Force
<b>ROLL</b>	Routing Over Low-power and Lossy Network
<b>ETX</b>	Expected Transmission
<b>LLN</b>	Low Power and Lossy Network
<b>OF</b>	Objective Function
<b>LBR</b>	Low Power and Lossy Border Router
<b>PFI</b>	Packet Forwarding Indication
<b>HO</b>	Hop Count

## **Introduction générale**

L'Internet des objets, ou Internet of Things (IoT) en anglais, transformera l'ensemble de la société, y compris nous-mêmes. À première vue, cette affirmation peut paraître exagérée, mais pensez à l'impact qu'a déjà eu Internet sur l'enseignement, les communications, les entreprises, la science, les organismes publics et les hommes. Internet est sans nul doute l'une des inventions les plus importantes et les plus significatives de toute l'histoire de l'humanité. Dites-vous maintenant que l'IoT représente la prochaine évolution d'Internet et permettra d'améliorer considérablement sa capacité à rassembler, à analyser et à restituer des données que nous pourrons ensuite transformer en informations, en connaissances et enfin en savoir. Dans ce contexte, l'importance de l'IoT paraît évidente. Des projets IoT déjà en cours promettent de combler les écarts de richesse, d'améliorer la distribution des ressources mondiales aux populations défavorisées et de nous aider à comprendre notre planète, ce qui nous permettra d'adopter un comportement plus proactif au lieu de simplement réagir aux événements. Toutefois, plusieurs obstacles menacent de ralentir le développement de l'IoT, notamment la transition vers le protocole IPv6, la mise en place de normes communes et le développement de sources d'énergie pour des millions, voire des milliards de minuscules capteurs. Heureusement, grâce aux efforts conjugués des entreprises, des administrations publiques, des organismes de normalisation et des universités, l'IoT poursuivra sa progression, le bon fonctionnement d'un réseau IoT est fortement lié au bon fonctionnement du protocole de communication, c'est pour ça ,nous avons choisi de modéliser et vérifier le protocole RPL en utilisant les deux outils UPAAL et cooja ,afin de comparer les résultats et valider le modèle [1].

### **Organisation du mémoire :**

Ce mémoire est organisé en quatre chapitres :

Le premier chapitre sera consacré à la présentation de l'Internet des objets, ainsi que l'introduction de quelques notions fondamentales sur l'IOT.

Dans le deuxième chapitre, nous présentons le protocole de communication RPL (IPv6 Routing Protocol for Low-Power and Lossy Networks) le plus utilisé dans le domaine de l'IOT.

Dans le troisième, nous montrons comment ajouter des modifications sur la modélisation du protocole RPL dans le l'outil de modélisation UPPAL et le simulateur Cooja.

Dans le quatrième chapitre, nous présentons une comparaison des résultats obtenus, et les observations sur ces résultats.

# **Chapitre 1 :**

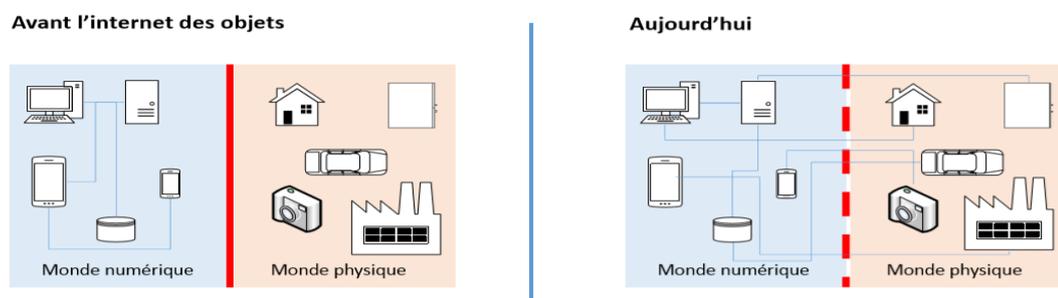
## **Généralité sur l'internet des objets**

## 1.1.Introduction [2] :

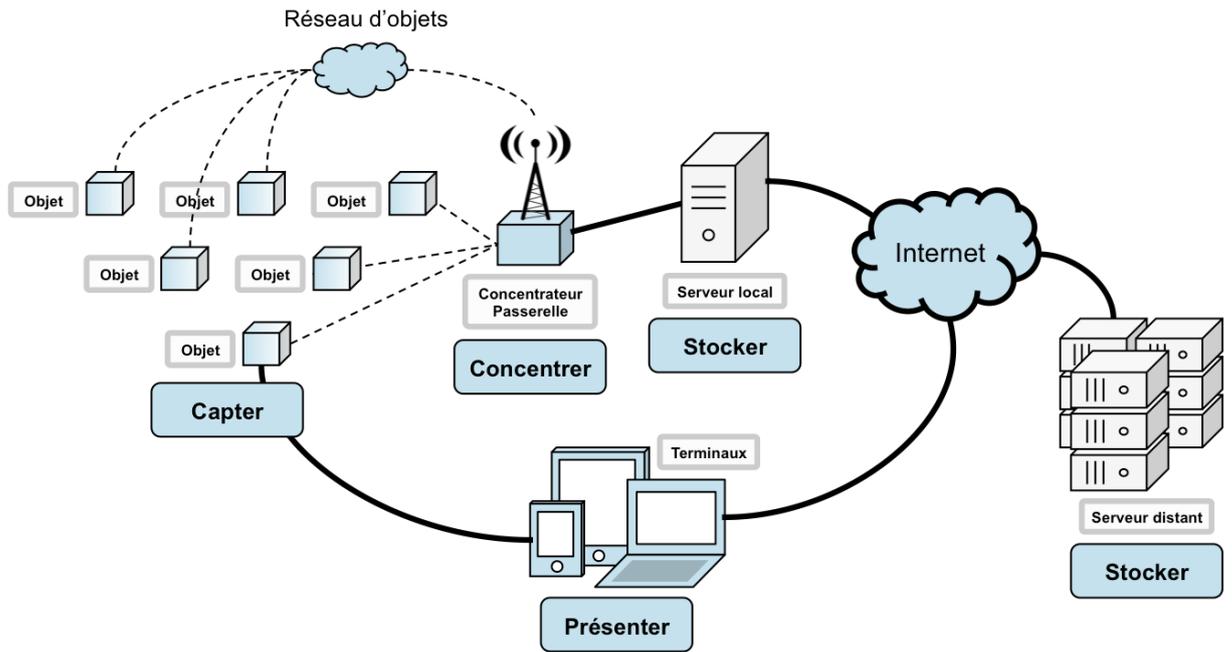
Aujourd'hui, Internet se transforme progressivement en un Hyper Réseau, comme un réseau formé par des multitudes de connexions entre des Artefacts (physiques, documentaires), des acteurs (biologiques, algorithmiques), des écritures et des concepts (linked data, metadata, ontologies), appelé «Internet of Things (IoT) Internet des objets (IdO)», connectant des milliards d'êtres humains, mais aussi des milliards d'objets. Il devient l'outil le plus puissant jamais inventé par l'homme pour créer, modifier, et partager les informations. Cette transformation montre l'évolution du réseau d'internet : d'un réseau des calculateurs vers à un réseau d'ordinateurs personnels, et puis vers un réseau nomade intégrant les technologies des communications. Les développements des technologies Machine-toMachine (M2M) pour le contrôle de machine à distance et aussi l'apparition dans l'année 2000 d'IP (Internet Protocole) sur les réseaux mobiles cellulaires ont accéléré l'évolution de M2M vers l'IOT.

## 1.2. Définition d'internet des objets :

Le CERP-ido «cluster des projets européens de recherche sur l'internet des objets» définit l'internet des objet comme une infrastructure dynamique d'un réseau globale .ce réseau globale a des capacités d'auto configuration basée sur des standards et des protocoles de communication interopérable .dans ce réseau les objet physiques et virtuels ont des identités ,des attributs physiques ,des personnalistes virtuelles et des interfaces intelligentes , et ils sont intégrés au réseau du façon transparente, Ce réseau crée une passerelle entre le monde physique et le monde virtuel, la «figure1.1 » montre la jonction entre le monde physique et le monde numérique .

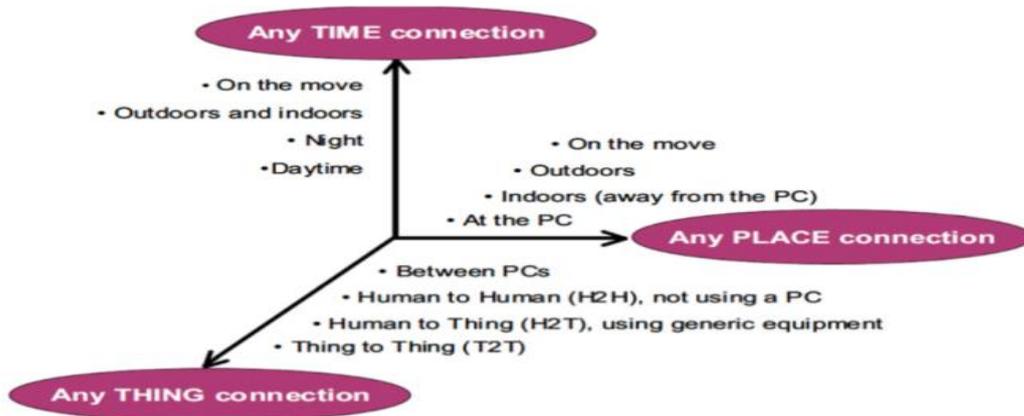


**Figure 1.1** : Jonction entre le monde physique et le monde numérique [2].



**Figure 1.2 :** L'IOT connecte des objets en utilisant des capteurs et Internet [41].

Cette définition montre les deux aspects d'IOT : temporel et spatiale qui permet aux personnes de se connecter de n'importe où à n'importe quel moment à travers des objets connectés (Smartphone, capteur, vidéosurveillance) (figure 1.3).



**Figure 1.3 :** une nouvelle dimension pour l'IOT [2].

### 1.3. Définition d'un objet connecté [3]:

Un objet connecté est un objet physique équipé de capteurs ou d'une puce qui lui permettent de transcender son usage initial pour proposer de nouveaux services. Il s'agit d'un matériel électronique capable de communiquer avec un ordinateur, un smartphone ou une tablette via un réseau sans fil (Wi-Fi, Bluetooth, réseaux de téléphonie mobile, réseau radio à longue portée de type Sigfox ou LoRa, etc.), qui le relie à Internet ou à un réseau local.

On distingue communément deux grands groupes d'objets connectés :

- les objets destinés à **la collecte et l'analyse de données**, dont la mission principale est de collecter et transmettre des informations ;
- les objets qui répondent à une logique de **contrôle-commande** et permettent de déclencher une action à distance.

Les capteurs installés sur ces objets connectés sont plus ou moins intelligents, selon qu'ils intègrent ou non eux-mêmes des algorithmes d'analyse de données, et qu'ils soient pour certains auto-adaptatifs.

Sortie en 2003, la lampe DAL connectée en Wi-Fi est le premier objet connecté de l'histoire. Selon ses créateurs, elle s'est vendue à une cinquantaine d'exemplaires. Grâce à ses 9 LED, elle pouvait s'allumer de différentes couleurs en fonction de différents événements liés à la météo, la situation boursière, la pollution, les alertes Google ou encore des « envois de messages de couleurs » par SMS ou courriel .

### 1.4. Types d'objets [21]:

- Electroniques : comme les véhicules connectés en 4G pour optimiser les performances.
- Electriques : Tout ce qui est de la domotique, allumage à distance etc...
- Non électriques : vêtements, animaux ...
- Capteurs environnementaux.

Les objets sont des dispositifs matériels connectés à Internet et entre eux. Ils détectent et recueillent plus de données, deviennent sensibles au contexte.

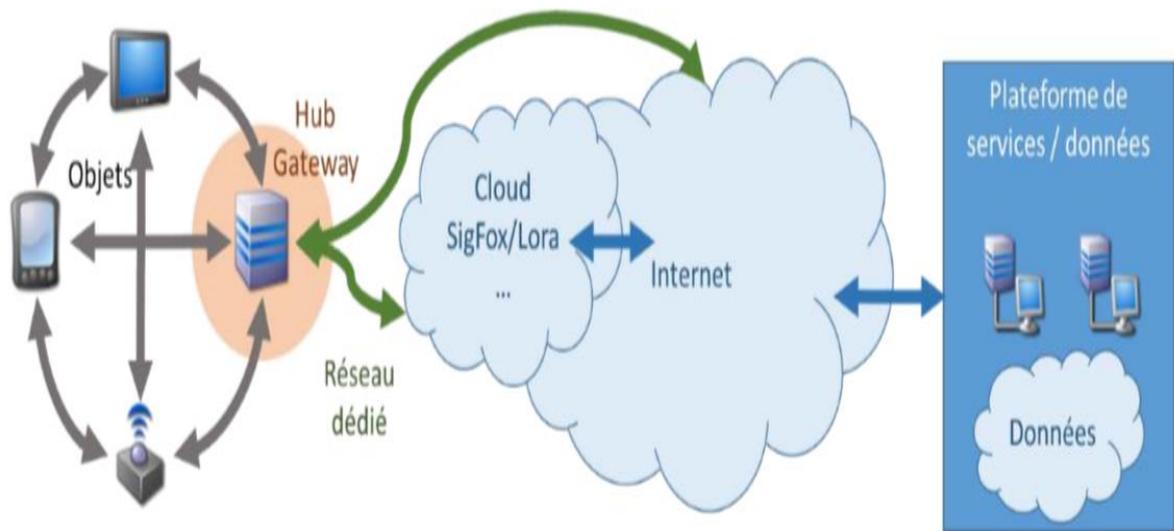
## **1.5. Infrastructure de communication [23]:**

Dans leur majorité, les objets ne se connectent pas directement à internet mais via une passerelle ou un hub numérique. L'infrastructure réseau la plus souvent mise en œuvre peut se représenter comme montrer la figure 1.5 :

Les objets connectés communiquent entre eux, ou avec les serveurs de traitement, prioritairement par le réseau Internet. Néanmoins, les types de liaison peuvent différer suivant les situations et les environnements : des objets peuvent partager directement entre eux des informations sur leur environnement pour interagir sans échanger avec des serveurs ; ou bien peuvent avoir besoin d'agréger les données avant de les transmettre à un serveur de traitement. Ils peuvent aussi être dans des zones à faible couverture avec un débit faible ou très éloignés d'un point d'accès réseau et dans ce cas utilisé un réseau dédié.

Dans l'infrastructure de communication, les éléments à prendre en compte concernent :

- Le type de communication (courte ou longue portée)
- La couverture du réseau
- La consommation énergétique de l'objet
- Le volume de données transmises
- La fréquence de captation
- La fréquence de transmission
- Le prix des capteurs/émetteurs.



**Figure 1.5:** Infrastructure réseau communément mise en œuvre pour les objets connectés[23].

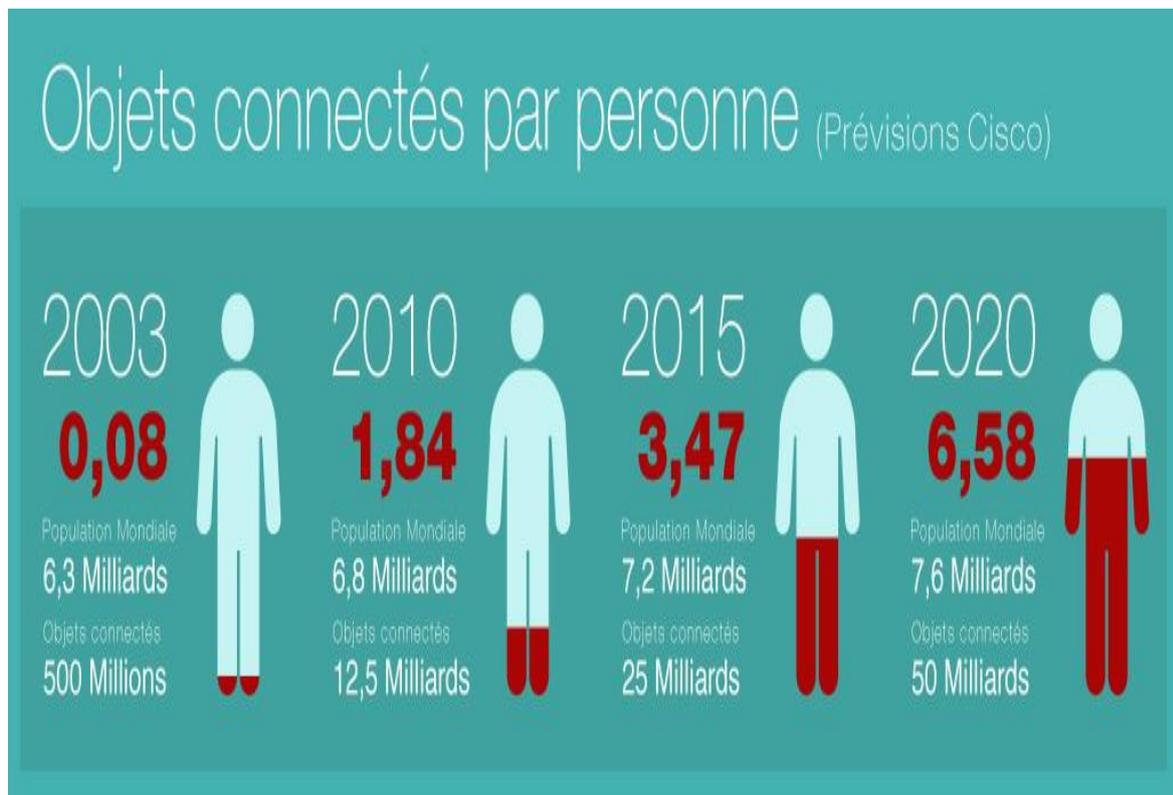
### 1.6.L'évolution d'IOT et son impact dans le monde [1] :

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'un appareil connecté par personne. Selon la définition de Cisco IBSG, l'IoT n'existait pas encore en 2003 car le nombre d'objets connectés était relativement 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 (1,84 pour être exact) pour la première fois de l'histoire.

En ce qui concerne l'avenir, Cisco IBSG estime que 25 milliards d'appareils seront connectés à Internet d'ici à 2015 et 50 milliards, d'ici à 2020. Il est important de noter que ces estimations ne tiennent pas compte des progrès rapides d'Internet ni des avancées technologiques, mais reposent uniquement sur les faits avérés à l'heure actuelle. En outre, le nombre d'appareils connectés par personne peut sembler faible, mais il ne faut pas oublier que le calcul porte sur l'ensemble de la population mondiale, dont une grande partie n'est pas encore connectée à Internet. Si l'on se base uniquement sur la population disposant d'une connexion à Internet, le nombre d'appareils connectés augmente

considérablement. Par exemple, nous savons qu'environ 2 milliards de personnes utilisent actuellement Internet. Si l'on utilise ce chiffre, le nombre d'appareils connectés par personne passe à 6,25 en 2010, au lieu des 1,84 précédemment indiqués.

La figure suivante montre l'augmentation des objets connectés par rapport le monde humain.



**Figure 1.6** : la prévision « cisco » sur les objets connectés par personne[1]

### **1.7. Domaine d'application d'IOT [4] :**

L'IOT permettra le développement de plusieurs applications intelligentes à l'avenir qui toucheront essentiellement:

- l'aérospatiale et l'aviation;
- l'automobile
- les télécommunications
- les bâtiments intelligents
- la santé et les technologies médicales

### 1.7.1. Les bâtiments intelligents :

Le secteur du bâtiment est, parmi les secteurs économiques, le plus consommateur en énergie. Des solutions comme les compteurs intelligents deviennent de plus en plus populaires pour mesurer la consommation d'énergie et la transmettre, par téléphone ou par courant porteur en ligne (CPL) au gestionnaire des données de comptage. Toujours dans une logique d'objets interconnectés, ce type de solution peut être combiné à d'autres capteurs (température, humidité) afin de disposer d'informations générales et précises sur les bâtiments formant un environnement intelligent et économe. La valeur ajoutée se trouve dans l'acquisition des données en temps réel permettant de s'adapter aux facteurs environnementaux et de détecter immédiatement les situations dangereuses. De plus, ces systèmes accompagneront et assisteront les personnes âgées à domicile dans une société confrontée au vieillissement de la population.



Figure 1.7 : Les bâtiments intelligents[3].

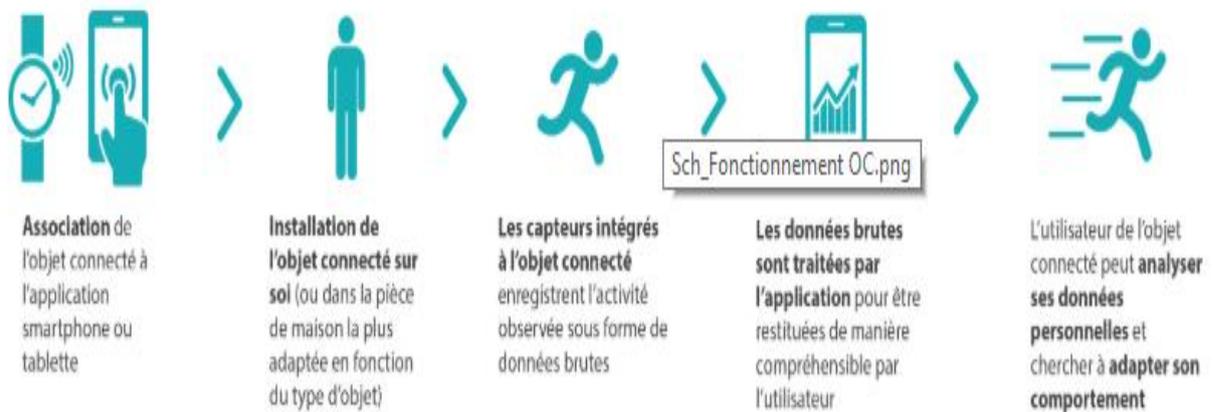
### 1.7.2. L'aérospatiale et l'aviation :

L'Internet des Objets va renforcer la sécurité des produits et des services en les protégeant de la contrefaçon. Un problème auquel est confrontée l'aviation ; 28 incidents ont été provoqués aux Etats-Unis par des éléments contrefaits non conformes aux normes de sécurité. Ces incidents pourraient être évités en introduisant un «pedigree électronique» qui tracerait le cycle de vie des éléments critiques de l'avion, de leur fabrication jusqu'à

leur utilisation. Ceci est réalisable en couplant la technologie RFID à une base de données dynamique. Cette base de données pourra être couplée à d'autres éléments de l'avion comme différents capteurs (pression, température...) et systèmes de sécurité.

### 1.7.3. La santé et les technologies médicales (Health care) :

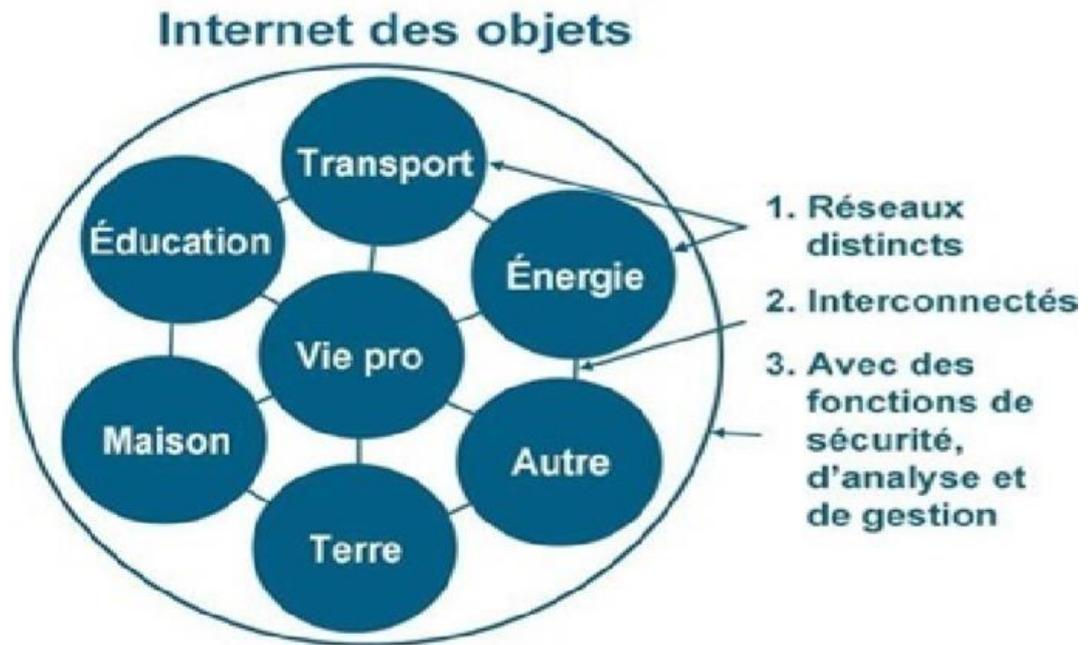
pour faire face au surpoids, détection d'anomalies dans le corps (fièvre, hypertension, battement cardiaque trop lent ou trop élevé..) en se servant de capteurs adaptés afin d'envoyer une alerte à un professionnel de la santé lorsqu'un certain seuil est dépassé. Nous pourrions même savoir si la personne qui le porte est tombée et ne parvient pas à se relever [20].



**Figure 1.8** : l'impacte de l'IOT sur la santé [20].

### 1.7.4. L'agriculture :

L'IdO permettra une meilleure aide à la décision en agriculture. L'IdO servira non seulement à optimiser l'eau d'irrigation, mais aussi, cette technologie peut être utilisée pour lutter contre la pollution (l'air et les eaux) et améliorer la qualité de l'environnement en général [6].



**Figure 1.9** : les domaines d'application de l'IOT [1].

## **1.8. Architecture de l'internet des objets [5] :**

Les racines de l'IDO remontent aux technologies M2M (machine à machine) pour le contrôle des processus à distance. L'IDO qui est aujourd'hui un mélange de plusieurs technologies telles que la RFID, NFC, les capteurs et actionneurs sans fil, le M2M, l'ultrabande ou 3/4G, IPv6, et RPL nécessite la définition d'une architecture et des standards afin de faciliter son développement dans le futur. L'ETSI propose une architecture découpée en trois domaines distincts, le domaine du réseau d'objets, le domaine du réseau cœur d'accès et le domaine des applications M2M et applications clients et organisée en trois couches principales : la couche de perception de données, la couche réseau et enfin la couche application.

### **1.8.1. Le domaine du réseau d'objets :**

Est la couche physique, qui contient des capteurs permettant de détecter et de collecter des informations sur l'environnement.

Dans ce domaine nous trouvons les différentes technologies d'interconnexion des objets M2M, RFID, Bluetooth, IETF6L low PAN, IETF RPL et des passerelles vers les réseaux cœur de transport.

### 1.8.2. Le domaine du réseau cœur d'accès [6] :

La couche réseau est responsable de la connexion à d'autres objets intelligents, périphériques réseau et serveurs. Ses fonctionnalités sont également utilisées pour la transmission et le traitement de données de capteurs.

Dans ce domaine nous trouverons les différentes technologies de réseaux de transport et d'accès comme XDSL, WIMAX, WLAN, 3/4G, etc.

### 1.8.3. Le domaine des applications M2M et applications clientes :

La couche application est chargée de fournir des services spécifiques à l'application à l'utilisateur. Diverses applications dans lesquelles l'Internet des objets peut être déployé, par exemple, les maisons intelligentes, les villes intelligentes et la santé intelligente.

Ce domaine est composé de plateformes M2M, les Middlewares et API des applications M2M, processus métiers exploitant l'IDO, etc.[6].

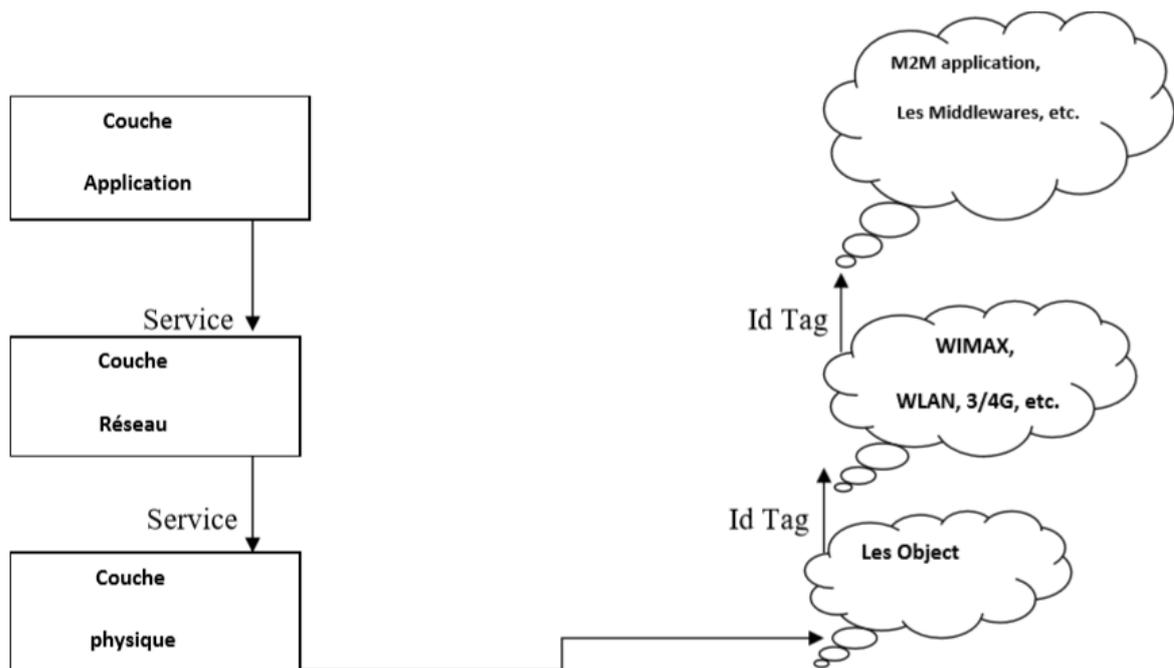


Figure 1.10 : Architecture de L'internet des Objets[6].

## 1.9. Les normes et les standards pour l'IOT [19.a]:

On distingue plusieurs technologies de communication sans fil utilisées par les objets communicants.

### 1.9.1. Les technologies de courte portée :

- **Wifi :**

La première est la technologie WiFi (correspondant à la famille des standards IEEE 802.11) qui équipe aujourd'hui tous les nouveaux smartphones, et est utilisée principalement par les réseaux locaux sans fil. Le Wifi permet l'accès à Internet avec des vitesses de transfert atteignant facilement des dizaines de méga bits par seconde. Il est caractérisé par une consommation relativement importante, et donc une autonomie faible pour les équipements qui l'embarquent.



**Figure 1.11:** la technologie wifi

- **Bluetooth :**

La deuxième technologie est celle du Bluetooth, qui équipe également la plupart des terminaux intelligents. Un grand nombre d'objets communicants l'adopteront aussi. Elle est définie par le groupement d'intérêt Bluetooth, pour les réseaux sans fil personnel. Dans ses versions de base, Bluetooth se caractérise par une communication à faible portée et un débit beaucoup plus faible que le WiFi, de l'ordre de quelques centaines de kilo bits par seconde. Néanmoins dans sa dernière version 4.2, qui met l'accent sur les objets communicants, on promet des débits 2 fois et demi plus rapides que les versions précédentes, tout en ayant une faible consommation d'énergie. Des objets tels que les montres connectées, équipées de cette dernière version, seront capables de dialoguer directement avec un routeur, afin d'accéder à Internet en utilisant le protocole 6LowPan d'IPv6, sans dépasser par un smartphone.



**Figure 1.12 :** la technologie Bluetooth.

- **Zigbee :**

La troisième technologie se nomme Zigbee. Elle est basée sur le standard IEEE 802.15.4 pour les couches physique et liaison. Elle est définie par la Zigbee Alliance, et a été spécialement conçue pour des équipements à faible consommation d'énergie, avec un très bas débit n'excédant pas 250 kilo bits par seconde, et une taille de paquets ne dépassant pas 127 octets. Zigbee intègre un protocole de routage mesh, permettant une connectivité au-delà de la portée radio, en utilisant les nœuds intermédiaires comme relais et en utilisant un plan d'adressage qui lui est propre. Cela dit, une version plus récente, Zigbee IP, supporte désormais les standards 6LowPan d'IPv6, ouvrant le champ de l'interopérabilité avec le reste du monde.



**Figure 1.13 :** la technologie ZigBee.

### **1.9.2. Les technologies de longue portée [7] :**

Pendant plusieurs années, les communications de courte portée comme ZigBee, WiFi, RFID, NFC et autres étaient les seules technologies et standards considérés dans les choix de technologies de communications.

- **Sigfox et lora :**

Depuis trois ans, une technologie portée par une startup française, SigFox, nommée Ultra Narrow Band va révolutionner les technologies de communication pour l'Internet des Objets.

Sigfox et LoRa communiquent avec 25mW de puissance, le premier à 60km, le second à 15km. Ils utilisent des modes déconnectés où la consommation n'est nécessaire que lorsque l'objet a des données à transmettre.

Solution adaptées aux lieux isolés mais pas seulement Reposant sur une infrastructure réseau publique (ou privée dans le cas de LoRaWan) il est possible de disposer les objets dans des lieux isolés. Communication électrique que l'ordre de 25mAh, débit de 100b/s à 6Kb/s

Sans besoin de configuration ou câblage Autonome et configurés pour accrocher un réseau publique ou privé dédié ils ne nécessitent pas d'étape d'installation sur site : déploiement rapide et à faible couts. Non accessibles par un réseau classique ils sont peu exposés aux attaques.



**Figure 1.14** : la technologie SigFox et LoRa[42].

### **1.10. Les avantages de l'IOT [22] :**

Voici quelques avantages de l'IoT:

- **Données:**

Plus les informations ne sont nombreuses, plus il est facile de prendre la bonne décision. Savoir ce qu'il faut obtenir de l'épicerie pendant que vous êtes dehors, sans avoir à vérifier par vous-même, non seulement vous fait gagner du temps, mais c'est également pratique.

- **Suivi:**

Les ordinateurs gardent une trace à la fois de la qualité et de la viabilité des choses à la maison. Connaître la date de péremption des produits avant de les consommer améliore la sécurité et la qualité de vie. En outre, vous ne manquerez jamais de rien lorsque vous en aurez besoin au dernier moment.

- **Temps:**

Le temps économisé en surveillance et le nombre de déplacements effectués seraient énormes.

- **Argent :**

L'aspect financier est le meilleur avantage. Cette technologie pourrait remplacer les humains responsables de la surveillance et de la maintenance des fournitures.

## **1.11. Désavantages de l'IOT :**

Voici quelques inconvénients de l'IoT:

- **Compatibilité:**

À l'heure actuelle, il n'existe pas de norme pour l'étiquetage et la surveillance avec des capteurs. Un concept uniforme tel que l'USB ou le Bluetooth est requis, ce qui ne devrait pas être si difficile à faire.

- **Complexité:**

Il existe plusieurs possibilités d'échec avec des systèmes complexes. Par exemple, vous et votre conjoint pouvez recevoir des messages indiquant que le lait est fini et que vous pouvez tous les deux acheter le même produit. Cela vous laisse le double de la quantité requise. Ou bien il y a un bug logiciel qui oblige l'imprimante à commander de l'encre plusieurs fois lorsqu'elle nécessite une seule cartouche.

- **Confidentialité / Sécurité:**

La confidentialité est un gros problème avec l'IdO. Toutes les données doivent être cryptées afin que les informations relatives à votre situation financière ou à la quantité de lait que vous consommez ne soient pas connues du public sur le lieu de travail ou avec vos amis.

- **Sécurité:**

Le logiciel peut être piraté et vos informations personnelles mal utilisées. Les possibilités sont infinies. Le fait que votre ordonnance soit modifiée ou que les détails de votre compte soient piratés peut vous mettre en danger. Par conséquent, tous les risques pour la sécurité deviennent la responsabilité du consommateur.

## **1.12. Les protocoles d'IOT :**

– 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 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 RPL.

– Au niveau de la couche application le protocole CoAP qui tente d'adapter HTTP, beaucoup trop gourmand aux contraintes des communications entre nœuds à faible consommation.

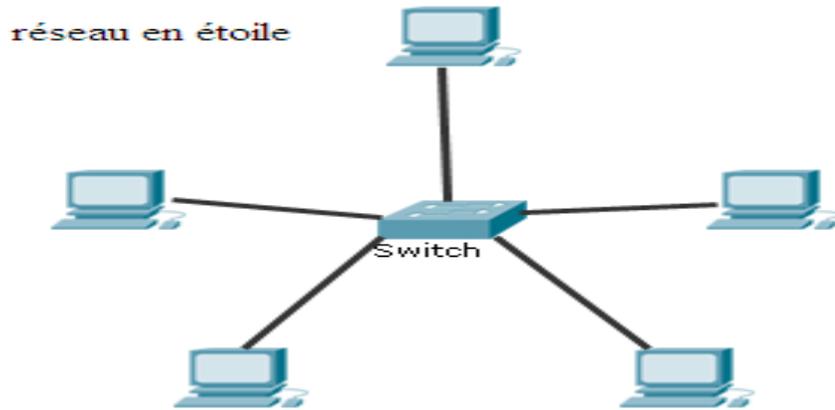
### **1.12.1. Le protocole IEEE 802.15.4 :**

IEEE 802.15.4 est une norme conçue pour répondre aux besoins des applications de réseau sans fil à faible consommation d'énergie, à faible débit de données et à faible coût. Il a été appliqué dans des domaines tels que la surveillance industrielle, les soins de santé personnels et l'automatisation résidentielle. À mesure que la demande de caractéristiques en temps réel augmente, la prise en charge de la qualité de service (QoS) devient un problème important dans ces domaines [8].

Les protocoles IEEE 802.15.4 supportent deux types de topologies [7] :

- **Les topologies en étoile :**

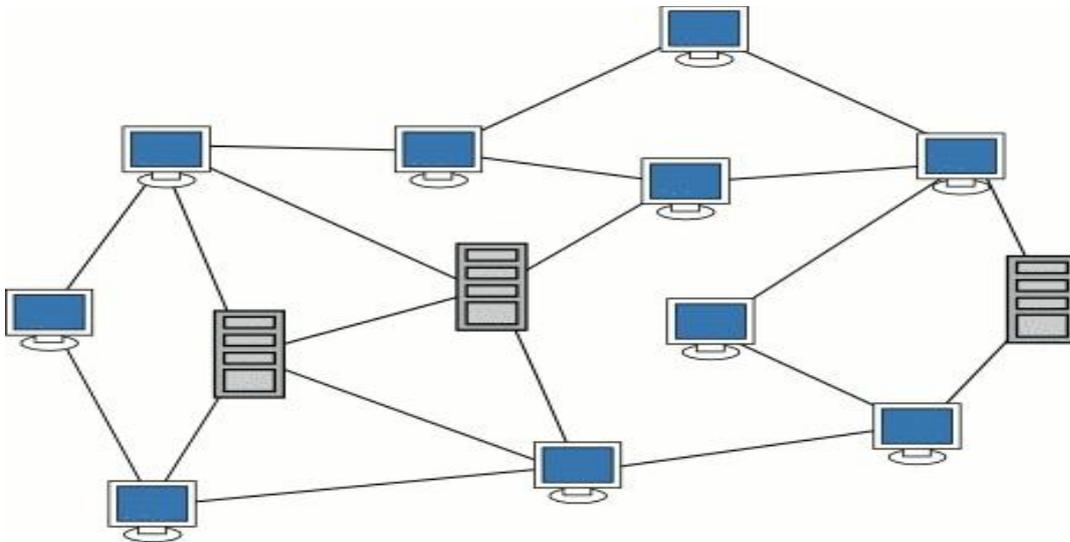
Les stations communiquent uniquement avec un nœud central, qui est le coordinateur PAN qui est la seule entité à pouvoir accepter de nouvelles connexions dans le réseau. La taille du réseau est limitée à la portée du coordinateur.



**Figure 1.15** : la topologie en étoile [49]

- **La topologie peere to peere :**

Les coordinateurs fils du coordinateur du PAN permettent a d'autre entités de s'associer au réseau a travers eux .Cela permet la formation d'un réseau plus complexe et plus large comme les réseaux maillés.



**Figure 1.16** : topologie peer to peer [49].

### 1.12.2. Le protocole MQTT [10] :

MQTT (Message Queuing Telemetry Transport). Ceci nous amène au protocole MQTT (Message Queuing Telemetry Transport). Ce protocole part de la constatation que le polling d'un nombre de plus en plus grand d'équipements connectés conduit à un gaspillage de bande passante en véhiculant des informations sans intérêt. C'est un protocole de type Pub/Sub (Publish/Subscribe) dans lequel les données émises par les équipements de terrain sont transférées vers un serveur MQTT (un « broker »), qui les trie, les stocke et les publie en les mettant à la disposition de clients intéressés. Ces clients peuvent être des systèmes scadas ou n'importe quelle autre entité intéressée par certaines données publiées par le broker. Les clients s'abonnent auprès du broker à un service en sélectionnant les types de données qui les intéressent (figure 1.17).

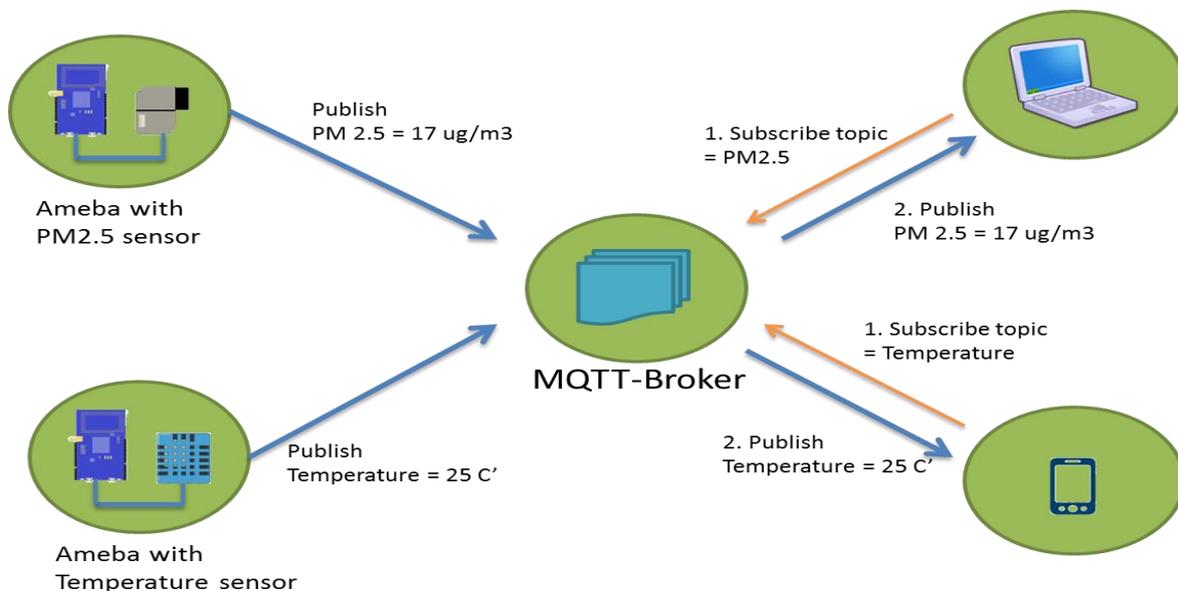


Figure 1.17: Schéma de fonctionnement de MQTT [10].

### 1.12.3. Le Protocole CARP [24] :

Le protocole CARP (Channel-Aware Routing Protocol) est un protocole de routage distribué conçu pour la communication sous-marine. Il peut être utilisé pour l'Ido en raison de ses paquets légers. Il prend en compte la qualité de la liaison, qui est calculée sur la base des transmissions de données historiques réussies recueillies à partir de capteurs voisins, pour sélectionner les nœuds de transmission. Il existe deux scénarios: l'initialisation du réseau et le transfert des données. Lors de l'initialisation du réseau, un

paquet HELLO est diffusé à partir du récepteur vers tous les autres nœuds des réseaux. Lors de la transmission des données, le paquet est acheminé du capteur vers le récepteur, et ce, étape par étape. Chaque saut suivant est déterminé indépendamment. Le principal problème de CARP est qu'il ne prend pas en charge la réutilisation des données précédemment collectées. En d'autres termes, si l'application requiert des données de capteur uniquement lorsque celles-ci changent de manière significative, le transfert de données CARP n'est pas avantageux pour cette application spécifique. Une amélioration de CARP a été effectuée dans E-CARP en permettant au nœud collecteur de sauvegarder les données sensorielles précédemment reçues. Lorsque de nouvelles données sont nécessaires, E-CARP envoie un paquet Ping auquel sont répliquées les données des nœuds de capteurs. Ainsi, E-CARP réduit considérablement les frais de communication

#### **1.12.4. Le Protocole RPL :**

Le groupe de travail IETF (Internet Engineering Task Force en Anglais) sur le routage sur les réseaux à faible consommation d'énergie et à perte a récemment proposé le protocole de routage IPv6 pour les réseaux à faible consommation d'énergie et à perte, c'est-à-dire le protocole RPL. Il a été conçu pour répondre aux exigences typiques des réseaux de capteurs sans fil. Compte tenu de sa pertinence dans les milieux industriels et scientifiques, ce protocole et ses règles sont détaillées dans le deuxième chapitre.

#### **1.12.5. Le protocole CORPL :**

CORPL, ou RPL cognitif, est une extension de RPL conçue pour les réseaux cognitifs et utilisant la génération de topologie DODAG, mais avec deux nouvelles modifications apportées à RPL. CORPL utilise le transfert opportuniste pour transférer le paquet en choisissant plusieurs redirecteurs (ensemble de redirecteurs) et des coordonnées entre les nœuds pour choisir le meilleur saut suivant pour le transfert du paquet. DODAG est construit de la même manière que RPL. Chaque nœud conserve un ensemble de transfert au lieu de son parent uniquement et met à jour son voisin avec ses modifications à l'aide de messages DIO. Sur la base des informations mises à jour, chaque nœud met à jour de manière dynamique les priorités de ses voisins afin de construire le groupe de redirecteurs.

### **1.13. Conclusion :**

Ce chapitre décrit une vision technique de l'internet des objets en fonction des phases qu'elle a franchies aussi que les types des objets et la description de l'architecture internet des objets et les technologies de communication des objets connectés ,ainsi que les différentes protocoles ,le protocole RPL fera l'objet du chapitre suivant.

# **Chapitre 2 :**

## **Protocole de routage RPL**

## **1.1.Introduction :**

Dans ce chapitre, nous allons présenter avec plus de détails le fonctionnement du protocole de routage RPL. Ce protocole a été développé pour des systèmes à ressources limitées et pour les systèmes qui se caractérisent par l'instabilité des connexions entre les nœuds à cause de présence d'obstacles. Par ailleurs, RPL instaure à chaque nœud un rang qui correspond à une distance virtuelle de ce dernier au nœud puits ceci est dans le but d'éviter des boucles de routage.

## **2.2.Définition de protocole RPL [11] :**

Le protocole de routage pour réseaux à faible consommation et à perte RPL (Routing Protocol for Low power and lossy networks-LLNS) est un protocole de routage IPv6, optimisé en particulier pour les réseaux sans fil, et conçu par l'IETF(Internet Engineering Task Force) sur un réseau à faible consommation et avec perte ROLL(Routing Over Low-power and Lossy Network) comme norme proposée. Les nœuds de capteur d'un réseau WSN sont capables de traiter, de collecter des informations sensorielles et de communiquer avec d'autres nœuds connectés au réseau.

RPL sélectionne le meilleur chemin en fonction de la fonction objectif (OF) ou des métriques de chemin. La force du RPL dépend du choix du meilleur OF, ce qui crée un chemin optimal entre un nœud source et un nœud collecteur.

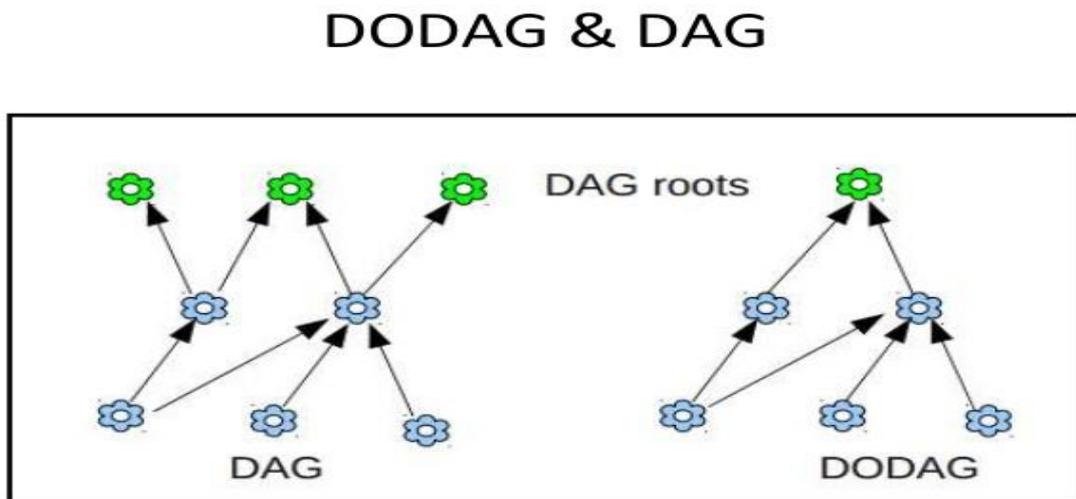
RPL construit un DODAG (Destination Oriented Directe Acyclic Graph) pour l'acheminement des données vers la station de base. Le DODAG construit permet à chaque nœud du DODAG de transmettre les données qu'il a récolté jusqu'au DODAG root (racine). Chaque nœud dans le DODAG sélectionne un parent selon une métrique de routage donnée et une fonction objective. Les données récoltées sont acheminées de fils à parent jusqu'à la racine.

## 2.3.La topologie du RPL :

### 2.3.1. Structure DAG et DODAG [12] :

RPL est un protocole de routage comprenant un algorithme de structuration du réseau. La topologie créée par RPL est un graphe orienté acyclique DAG (Directed Acyclic Graph). Le DAG définit une structure arborescente spécifiant les routes par défaut entre les nœuds du LLN (low power and lossy network). Cependant, une structure de DAG est plus qu'un arbre typique dans le sens où un nœud peut être associé à plusieurs parents dans le DAG. L'objectif de cette topologie est d'offrir un routage efficace et fiable de n'importe quel point du réseau vers la racine du DAG.

Le protocole RPL organise les nœuds sous forme de DODAG (DAG Destination Oriented) c'est-à-dire un graphe orienté dirigé vers une destination qui est la racine du réseau (seul un parent est autorisé) comme illustré dans la figure 11 chaque DODAG a un numéro de version (figure 2.1).



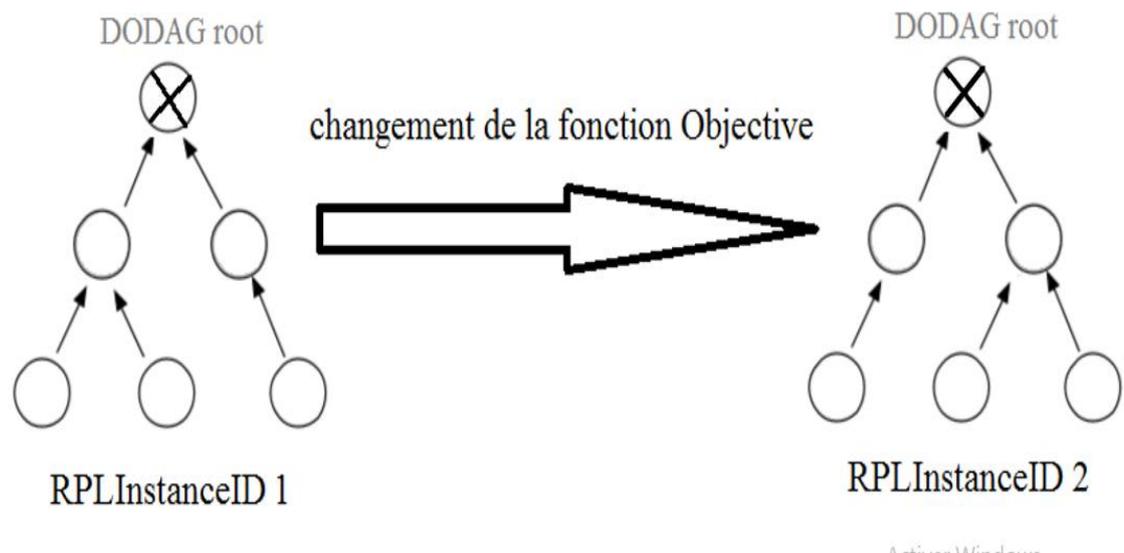
**Figure 2.1** : la structure DAG et DODAG [43].

### 2.3.2. Identifiant RPL [14]:

RPL utilise quatre valeurs pour identifier et gérer une topologie:

- **RPLInstanceID :**

Un RPLInstanceID identifie un ensemble d'un ou plusieurs DAG orientés destination (DODAG). Un réseau peut avoir plusieurs identificateurs d'instance RPL, chacun définissant un ensemble de DODAG pouvant être optimisés par des différentes fonctions objectives et / ou applications. L'ensemble des DODAG identifiés par un RPLInstanceID est appelé une instance RPL. Tous les DODAG de la même instance RPL utilise la même OF.



**Figure2.2 :** deux instances sur un seul DODAG physique [14].

- **DODAGID :**

La portée d'un DODAGID est une combinaison de RPLInstanceID et DODAGID uniquement identifie un seul DODAG dans le réseau. Une instance RPL peut avoir plusieurs DODAG, chacun ayant un DODAGID unique.

- **Version DODAG:**

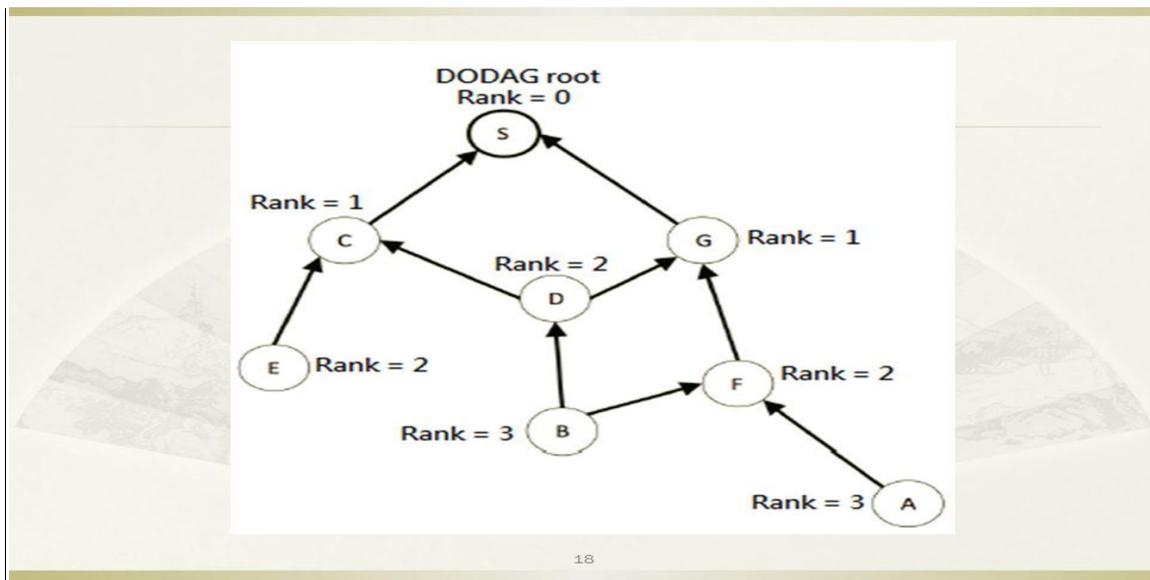
Une version DODAG est une itération spécifique ("version") d'un DODAG avec un DODAGID donné.

- **DODAGVersionNumber :**

Un DODAG est parfois reconstruit à partir de la racine DODAG, en incrémentant le DODAGVersionNumber. La combinaison de RPLInstanceID, DODAGID et DODAGVersionNumber identifie de manière unique une version DODAG.

- **Rank :**

La portée ou Rank définit les positions des nœuds par rapport à la racine DODAG comme illustrer la figure 2.3.



**Figure 2.3** : représentation du rang[42].

- **Objectif (Goal) :**

L'objectif est un objectif spécifique à l'application défini hors du champ de RPL. Tout nœud qui enraine un DODAG sera besoin de savoir sur cet objectif pour décider si oui ou non l'objectif peut être satisfait.

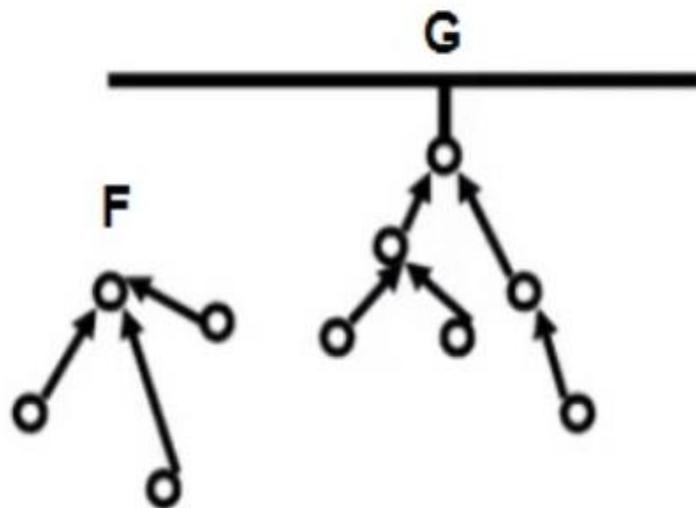
Un objectif typique est de construire le DODAG selon une fonction objectif spécifique et de garder connectivité à un ensemble d'hôtes (par exemple, pour utiliser un objectif Fonction qui minimise une métrique et est connectée à une spécifique base de données pour stocker les données collectées).

- **Mise à la terre(Grounded) :**

Un DODAG est mis à la terre lorsque la racine DODAG peut satisfaire l'Objectif.

- **Flottante(Floating) :**

Un DODAG flotte s'il n'est pas mis à la terre. Un flottant DODAG ne devrait pas avoir les propriétés requises pour satisfaire l'objectif. Il peut toutefois fournir une connectivité à autres nœuds au sein du DODAG.



**Figure2.4** : DODAGs flottants et mis à la terre[45]

- **DODAG parent:**

Un parent d'un nœud au sein d'un DODAG est l'un des successeurs immédiats du nœud sur un chemin menant au DODAG racine. Le rang d'un parent DODAG est inférieur à celui du nœud.

- **Sous-DODAG:**

Le sous-DODAG d'un nœud est l'ensemble des autres nœuds dont Les chemins d'accès à la racine DODAG passent par ce nœud. Nœuds dans le, Les sous-DODAG d'un nœud ont un rang supérieur à celui-ci.

### **2.3.3. Fonction objective :**

La fonction Objective (OF) définit comment les nœuds RPL sélectionnent et optimisent les itinéraires dans d'une instance RPL. L'OF est identifié par un point de code objectif (OCP) dans l'option de configuration DIO. Un OF définit comment les nœuds traduisent une ou plusieurs métriques et contraintes, pour les buts de, calculer le rang de chaque nœud dans le DODAG, définit également comment les nœuds sélectionnent les parents.

Le groupe de travail ROLL a défini deux fonctions objectives dans le RPL originale cette séparation destinée à permettre à la RPL d'être adaptée pour répondre aux critères d'optimisation différents, d'applications et de conceptions de réseaux : [26] [27]

- OF0 : Objective Function Zero Ici, la métrique d'acheminement adoptée est le nombre de sauts. OF0 est conçu pour permettre l'interopérabilité entre les implémentations différent de RPL. [26]

- MRHOF (Minimum Rank Hysteresis Objective Function) : La métrique utilisée par MRHOF est déterminée dans le conteneur de métrique du DIO. Le plus souvent c'est le nombre de retransmission attendue (ETX) qui a utilisé avec une hystérésis pour éviter les différences minimales de rang. Cette métrique permet à RPL de trouver les chemins stables à partir des nœuds vers une racine. En l'absence d'une métrique dans le conteneur métrique DIO, MRHOF utilise par défaut ETX [29].

## 2.4. Le modèle réseau :

RPL définit trois types de nœuds [25] :

- **DODAG root ou Low power and lossy Border Router (LBR)** : Il se réfère à la racine d'un DODAG qui représente un point de collecte dans le réseau et a la capacité de construire un DAG. Le LBR agit également comme une passerelle (routeur edge) entre l'internet et le LLN, il peut avoir des racines multiples configurées dans le réseau.

- **Nœud Routeur (ou statefull node)** : Il se réfère à un périphérique qui peut transmettre et génère du trafic des voisins. Un tel routeur n'a pas la capacité de créer un nouveau DAG.

- **Nœud host (ou stateless node)** : Il se réfère à un périphérique terminal capable de générer des données, mais qui n'est pas capable de router le trafic.

## 2.5. Message de contrôle RPL [13]:

### 2.5.1. DIO (DODAG Information Object) :

Ce message est utilisé pour la création des routes ascendantes. Il se propage dans le réseau en sens inverse, et transporte de nombreux paramètres nécessaires à la mise en place de la topologie. Ces paramètres sont fournis uniquement par la racine RPL, les autres nœuds ne servant que de relais. La dissémination des DIO se fait par défaut en multidiffusion, mais en cas de sollicitation spécifique d'un nœud, la monodiffusion est utilisée.

Le DIO est prévu pour transporter de nombreuses options, toutefois certaines informations sont obligatoires. Parmi celles-ci on a: RPLInstanceID qui identifie de façon unique l'instance RPL dans le réseau, le DODAGID qui caractérise la racine (très souvent l'une de ses adresses IPv6), le numéro de version du DODAG permettant de savoir si les informations reçues sont «fraîches» ou obsolètes et le rang du nœud émetteur dans le DODAG qui renseigne sur sa position dans l'arbre construit sur le réseau.

### 2.5.2. DIS (DODAG Information Solicitation) :

Elle est utilisée par un nœud qui souhaite rejoindre la topologie (envoi en multidiffusion) ou réclamant des informations de configuration plus récentes (envoi en monodiffusion). Dans tous les cas, un nœud recevant un DIS répond à l'initiateur par une monodiffusion de paquet DIO.

### 2.5.3. DAO (DODAG Advertisement Object) :

Ce message est utilisé uniquement en cas de nécessité de disposer des routes descendantes (pour le trafic point à point par exemple). Autrement il peut être désactivé pour économiser les ressources (mémoire et bande passante). Il est envoyé à la racine RPL quand le mode de fonctionnement est non conservation (non-storing mode) ou au(x) parent(s) en mode conservation (storing mode).

Le DAO quant à lui transporte les préfixes réseaux permettant de rendre accessible par monodiffusion les nœuds situés en profondeur de l'arbre. Contrairement aux autres messages de contrôle RPL, il est le seul toujours envoyé en monodiffusion et nécessite un acquittement du destinataire.

### 2.5.4. DAO-Ack (DAO-Acknowledgment) :

Il est utilisé par la destination en réponse à un message DAO reçu pour acquitter ce dernier. En cas de non réception du DAO-Ack par la source, celle-ci peut réémettre le DAO initial.

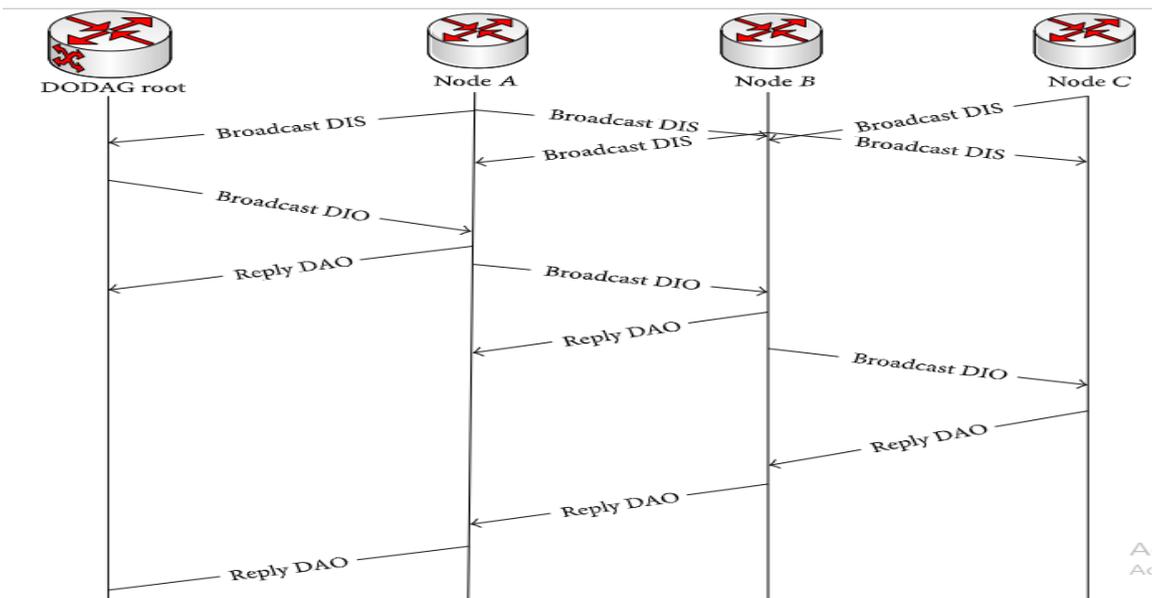


Figure 2.5 : les messages du contrôle dans le protocole RPL[13].

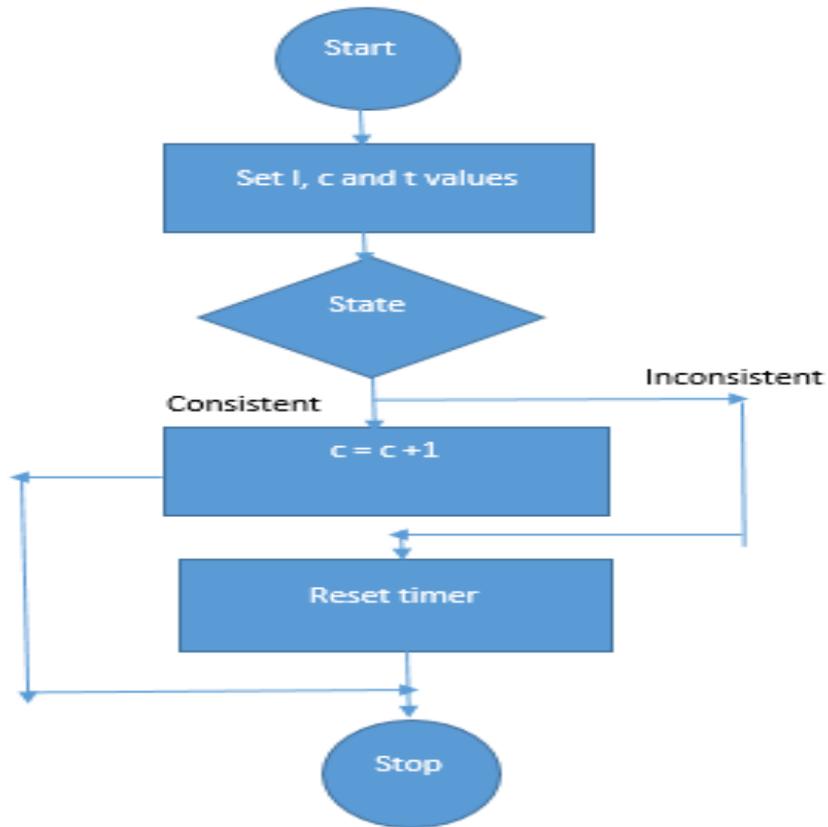
## 2.6. Algorithme Trickle :

RPL est le protocole de routage principalement utilisé pour les réseaux LLN. Il contient beaucoup compte-gouttes qui est principalement utilisé pour contrôler la transmission des messages de contrôle et économiser de l'énergie.

Il y a 3 paramètres dans l'algorithme de Trickle. IMIN, IMAX et  $k$ , où IMIN est la taille de l'intervalle minimal, IMAX est le nombre de doublons d'IMIN et  $k$  est la constante de redondance. Il y a 3 variables I qui définissent la taille de l'intervalle actuel, T qui indique l'heure actuelle dans l'intervalle d'intervalle et C qui est une variable de compteur.

Avant de démarrer la minuterie, la valeur IMIN est définie dans la plage  $[I / 2, I]$ . La variable compteur C permet de suivre l'état du système, qu'il soit incohérent ou cohérent. S'il est cohérent, le compteur est incrémenté, sinon sa réinitialisation. Cette valeur c doit toujours être inférieure à k (appelée constante redondante). Le filet est réinitialisé sur IMIN à l'expiration du délai sinon il est doublé. Il faut prendre soin de vérifier si le nouvel intervalle actuel se situe dans la plage ci-dessus. Les nœuds peuvent même être conçus pour supprimer l'incohérence lorsque cela est nécessaire. Ceci n'est pas recommandé, mais pour quelques applications, il peut être nécessaire de désactiver la suppression en définissant  $k = 0$ .

L'organigramme présenté ci-dessous décrit brièvement le fonctionnement de l'algorithme de ruissellement (figure 2.6).



**Figure 2.6** :l'algorithme trickle[15].

Les minuteries sont utilisées dans diverses applications. La plupart des applications peuvent avoir besoin de vérifier périodiquement si un nœud contient de nouvelles informations. Cela se fait à l'aide d'un numéro de version. Si une incohérence est détectée, cela signifie que le nœud ne dispose pas de nouvelles informations et doit mettre à jour ses informations. Un exemple de protocole utilisant le compte-gouttes est le protocole RPL. RPL dispose de messages DIO et DAO qu'il utilise pour envoyer et recevoir des messages. Le message DIO provient du récepteur tandis que le DAO provient des nœuds. Ce minuteur permet de garder une trace des messages envoyés par le récepteur lorsqu'il dit aux nœuds «Je suis le récepteur». En RPL, les DIO sont envoyées périodiquement et l'intervalle de temps est augmenté après chaque transmission. Cela se fait à l'aide d'un minuteur [15]

## 2.7. Construction de DODAG :

Le processus de construction de la structure DODAG commence à la racine ou LBR (LoWPAN Border Router) qui est généralement le nœud de collecte de données (le puits ou un actionneur). Il pourrait y avoir des racines multiples configurées dans le réseau. Le protocole de routage RPL spécifie un ensemble de nouveaux messages de contrôle ICMPv6 pour échanger des informations liées à la construction de la structure DODAG.

La racine commence la diffusion des informations concernant la structure en utilisant le message DIO (DODAG Information Object). Les nœuds à portée de communication de la racine recevront et traiteront ce message DIO, puis ils rendront une décision (joindre la structure ou pas) fondée sur certaines règles (selon la fonction objectif, les caractéristiques du DAG et le coût du chemin annoncé). Une fois que le nœud s'est joint à la structure, il a une route vers la racine de la structure DODAG.

La racine du DODAG est appelée le parent du nœud. Le nœud calcule son rang dans le graphe, qui représente la position du nœud dans la structure DODAG. Si ce nœud est configuré pour agir comme un routeur dans le réseau il commence à diffuser à son tour dans son voisinage les nouvelles informations de la structure qu'il vient de rejoindre à travers des messages DIOs. Si le nœud n'est pas configuré pour être un routeur alors il rejoint tout simplement la structure DODAG et n'envoie pas de message DIO.

Les nœuds voisins recevant cette annonce vont répéter ce processus de sélection de parent, d'ajout d'itinéraire et d'annonce des nouvelles informations concernant la structure DODAG à l'aide des messages DIOs . Ce processus continue jusqu'à couvrir tous les nœuds du réseau.

Chaque nœud de la structure DODAG a une entrée de routage vers son parent (ou plusieurs parents selon la fonction objectif) à travers lequel ce nœud peut atteindre la racine de la structure DODAG. En outre, chaque nœud dans le graphe a un rang qui représente la position relative de ce nœud par rapport à la racine de la structure DODAG. La notion de rang est utilisée par RPL à des fins diverses, y compris l'évitement des boucles.

Les différentes étapes du processus de construction graphique sont représentées dans la figure 2.7 où les messages DAO (Destination Advertisement Object) visent à

maintenir les routes descendantes et ne sont utilisés que pour des applications nécessitant des trafics de type point à multipoint et point à point [16].

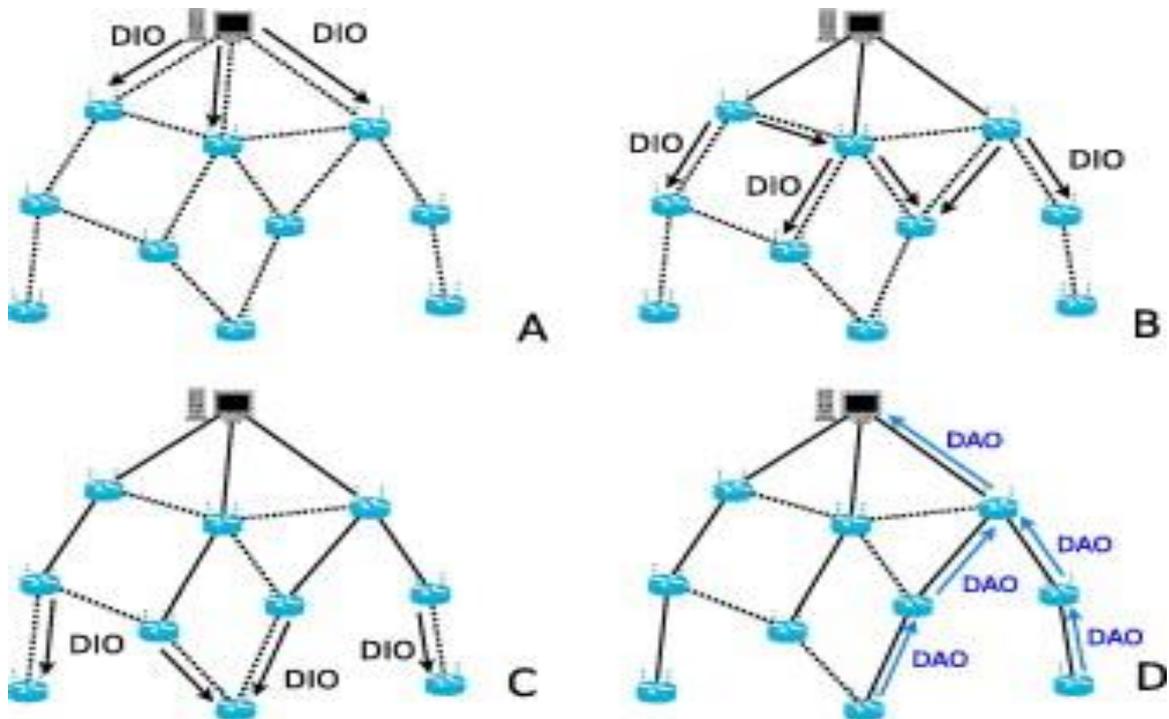


Figure 2.7 : construction de DODAG[16].

## 2.8. Maintenance de la topologie [26] :

En cas de rupture d'un lien, le DODAG peut être réparé de deux façons :

### 2.8.1. Réparation globale :

La racine initie une reconstruction complète du DODAG avec des messages DIO, des numéros de séquence sont utilisés pour différencier les DODAG ancien et nouveau, réparation coûteuse en trafic.

### 2.8.2. Réparation locale :

Recherche d'un nouveau parent par le nœud affecté, le DODAG n'est plus optimal, seule une réparation globale permettra une nouvelle optimisation.

## 2.9. Les modes de fonctionnements du protocole RPL :

### 2.9.1. Mode non storing [17]:

AD6LoWPAN Stack implémente RPL en mode sans stockage. Dans ce mode, les entrées de la table de routage ne sont gérées que sur un routeur frontière comme illustre la figure suivant.



Figure 2.8 : la 1ere étape pour construire la table de routage source[17].

Ici, R1 tente de rejoindre le routeur frontière (BR). Lorsque R1 envoie un DAO (Destination Information Object), le BR fait une entrée de R1 dans la table de routage source avec l'adresse du saut suivant définie comme non valide. Cela indique que le R1 est un enfant direct et ne nécessite aucun routage entre les deux.

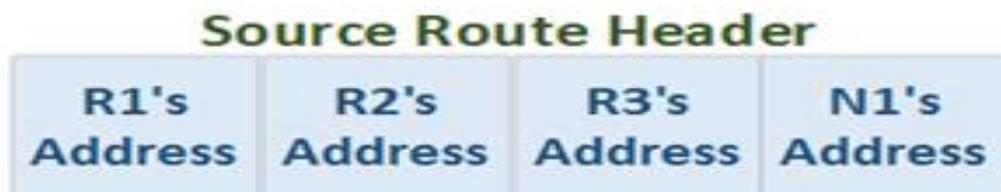
Lorsque R2 rejoint le réseau et choisit R1 comme routeur par défaut, il envoie un DAO (Destination Information Object) à son parent (R1), qui le transmet à son tour au routeur frontière (BR). Maintenant, ce BR fait une entrée pour le R2 dans la table de routage source avec la prochaine adresse de saut définie comme R1. Cela indique que pour atteindre R2, le paquet doit être routé via R1.



**Figure 2.9** : la table de routage [17]

AD6LoWPAN implémente un mode de fonctionnement sans stockage (routé entièrement à la source). Ainsi, lorsque BR souhaite envoyer un paquet à N1, il crée un en-tête de routage avec tous les sauts nécessaires pour atteindre N1. Ceci est réalisé comme suit:

Lorsque le BR souhaite transmettre un paquet à N1, il recherche l'adresse IP de ce dernier dans la table de routage source. S'il trouve l'adresse IP du N1 et l'ajoute dans l'en-tête de routage. Si l'adresse de saut suivant pour l'entrée de route source de N1 n'est pas invalide, l'adresse de saut suivant est ajoutée dans l'en-tête de routage. Dans notre exemple, le prochain saut est R3. Le routeur frontière recherche alors l'itinéraire pour atteindre R3. Cette opération est effectuée jusqu'à ce qu'une adresse immédiatement trouvée entre le routeur enfant et le routeur frontière soit trouvée (avec une adresse de saut suivant non valide).



**Figure 2.10** : en-tête de route source[17].

### 2.9.2. Mode stroing [18] :

En mode de stockage, un nœud conserve une liste complète des entrées de routage pour les nœuds de son sous-DODAG. Lorsqu'un nœud reçoit un paquet, il le transmet au saut suivant s'il trouve la destination dans la liste. S'il ne trouve pas la destination dans la liste, il transfère le paquet au parent préféré. À l'heure actuelle, RPL ne prend pas en charge le fonctionnement en mode mixte dans lequel certains nœuds s'exécutent en mode sans stockage et d'autres en mode de stockage. Si tous les nœuds du réseau s'exécutent en mode sans stockage, les nœuds qui ne sont pas la racine n'ont pas besoin de stocker d'informations de routage sur le réseau

Si tous les nœuds du réseau fonctionnent uniquement en mode de stockage, chaque nœud stockera les informations de routage relatives à son sous-DODAG. Avec un grand réseau, l'exécution du mode de stockage nécessitera uniquement des nœuds (en particulier ceux situés à la racine) pour fournir beaucoup de mémoire pour stocker la table de routage. Les nœuds situés autour de la racine peuvent ne pas se permettre de disposer de ressources mémoire aussi importantes (figure 2.11).

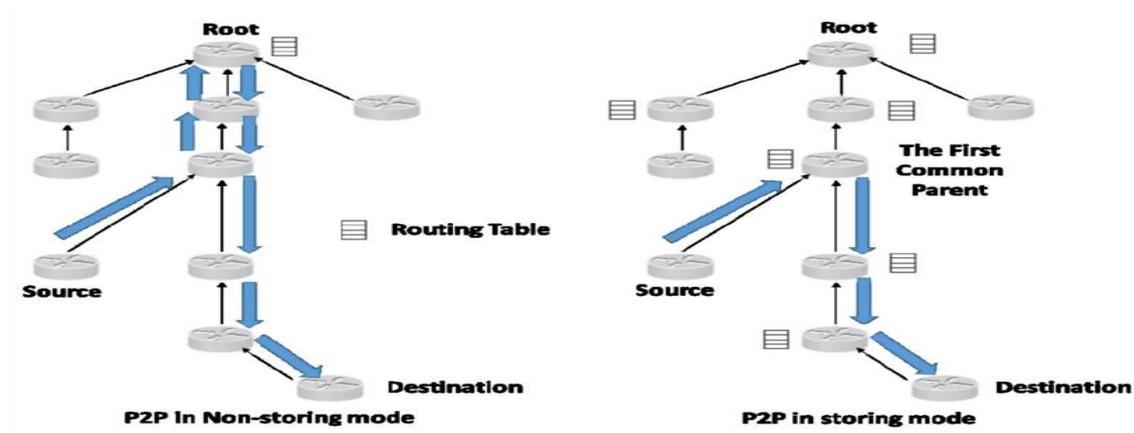


Figure 2.11 : mode storing et non storing dans le protocole RPL[18].

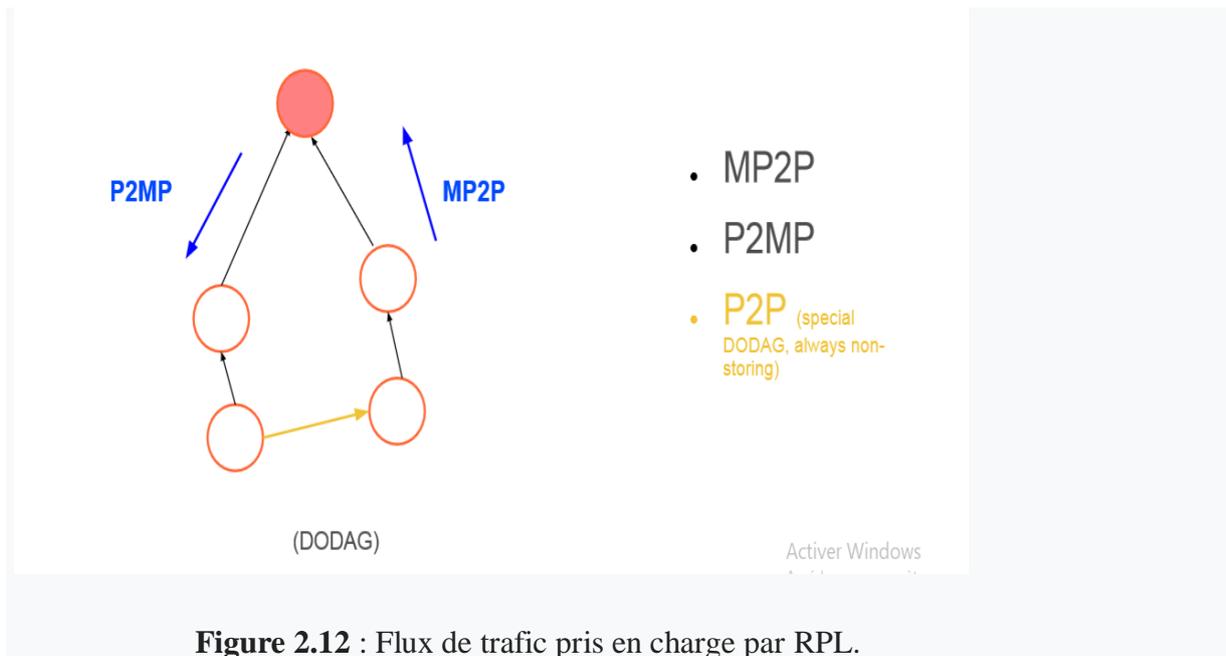
## 2.10. Les paradigmes de communication :

RPL prend en charge trois paradigmes de communication (MP2P), (P2MP) et (P2P). Dans ce qui suit, nous détaillons le fonctionnement de ces modèles de communication.

**Multipoint à point (MP2P) :** RPL a été conçu principalement pour optimiser le type de flux de trafic multipoint à point (MP2P), cette communication MP2P a été fournie par la construction des routes à partir de chaque nœud vers la racine DODAG à l'aide de DIO du parent préféré d'un nœud il s'agit de « Routes ascendantes » (Upward Routes) Les destinations des flux MP2P sont des nœuds désignés qui ont une certaine importance pour l'application, tels que la fourniture de connectivité à l'Internet ou au réseau IP privé principal.

**Point à multipoint (P2PM) :** Il s'agit de Routes descendantes (Downward Routes) RPL prend en charge le trafic P2MP il utilise un mécanisme de publicité de destination qui prévoit des itinéraires descendants de la racine vers d'autres nœuds (préfixes, adresses ou groupes de multidiffusion) [26]. Par exemple les messages DIO, P2PM est le modèle de trafic requis par plusieurs applications LLN. [26][27]

**Point à point (P2P) :** Pour le trafic P2P, la construction des routes ça dépend de mode de fonctionnement du protocole RPL. Si le cas de mode Non-Storing mode le paquet dirige vers une racine, ensuite la racine effectuera le routage vers la destination, si le cas de mode Storing mode, le paquet s'écoule vers la racine jusqu'à ce qu'il atteigne un ancêtre qui a une route connue vers la destination. Cet ancêtre commun peut être la racine DODAG. Dans d'autres cas, il peut s'agir d'un nœud plus proche de la source ou de la destination. [27] [28]



## 2.11. Travaux d'optimisation de RPL :

Même s'il est proposé pour répondre aux contraintes de routage dans les réseaux LLN. Plusieurs travaux ont montré que RPL souffre de certains défauts.

Dans cette partie, nous allons examiner une partie de ces travaux d'amélioration de RPL que nous avons classé en plusieurs catégories :

- (A) optimisation de la fonction objective,
- (B) optimisation de la topologie logique le nombre de chemins choisis et
- (C) Optimisation de la fonction Trickle Timer.

### 2.11.1. Optimisation de la fonction objective :

- **Choix des métriques**

Selon une seule métrique Par défaut, une seule métrique est utilisée dans les fonctions objectives définies par ROLL que cela soit le nombre de saut ou l'ETX. D'autres travaux proposent néanmoins d'utiliser de nouvelles métriques visant à améliorer un aspect de façon plus spécifique. Les auteurs dans proposent de s'appuyer sur le délai de transmission moyen ou (Averaged Delay - AVG DEL) qui vise à réduire le délai entre les nœuds, tout en supposant que les nœuds fonctionnent sur des cycles de sommeil et des cycles de réveil différents et ont des cycles de travail très faibles (moins de 1%).

L'inconvénient avec cette approche c'est qu'elle rajoute plus de données à un DIO déjà large, ce qui augmente le risque de fragmentation [30].

- **Selon plusieurs métriques**

Dans cette catégorie d'approche plusieurs métriques sont utilisées en même temps pour calculer le rang des nœuds. Le problème qui se pose est donc de trouver comment combiner ces métriques. Plusieurs solutions ont été proposées dans la littérature.

Les auteurs dans s'appuient sur une approche de combinaison additive et pondérée. Dans ce cas les métriques sont normalisées et des nombres réels positifs sont utilisés comme facteurs de multiplication pour ajuster les poids relatifs des métriques de routage en fonction de l'application. Ainsi, ils ont proposé de combiner deux métriques : le nombre de saut (Hop Count ou HC) et une métrique d'indicateur de confiance appelée Packet Forwarding Indication ou PFI qui permet de tester la confiance qu'un nœud peut avoir en son parent en calculant la probabilité de transfert des paquets. Cette métrique suit le concept d'ETX mais capture la fiabilité du nœud plutôt que la fiabilité du lien. En utilisant PFI, un nœud peut identifier et exclure des nœuds malicieux et parer ainsi aux attaques du trou noir ou du trou gris (black- and grey-hole attacks) [31]

Choix de la fonction objective Une autre approche proposée dans s'appuie sur les principes de la logique floue pour combiner plusieurs métriques. Une nouvelle fonction objective OF-FL a également été mise en place. Les métriques considérées dans cette approche sont le délai de bout-en-bout, le nombre de saut, la qualité du lien et l'énergie du nœud. La logique floue est utilisée pour évaluer le meilleur voisin comme étant le parent préféré. Les études de simulation sont réalisées sur une implémentation Contiki / Cooja de RPL[32].

#### **4.11.2. Optimisation de la topologie logique :**

Cette stratégie s'appuie sur le constat que comme il s'agit de réseaux à faible puissance et à perte, il serait intéressant non pas de considérer un chemin unique vers la racine mais plusieurs. De nombreux travaux de recherche ce sont intéressés à cette approche où une myriade de bénéfices a été trouvée tels que la fiabilité, la tolérance aux pannes ou encore l'évitement de la congestion par l'équilibrage de charge efficace. Ainsi, les auteurs proposent dans un RPL optimisé appelé ORPL qui diminue la latence de routage en utilisant une approche multipath. Ils utilisent la métrique du cycle de service attendu (EDC), qui est revendiquée comme l'équivalent multi chemin de ETX, il utilise

également un minuteur Trickle pour diffuser les mises à jour de routage, l'inconvénient de ce protocole et qu'il produit une surcharge significative de trafic de contrôle [33] [34].

### **2.11.3. Optimisation de Trickle Timer :**

qui est une sorte de minuterie visant à propager les messages de contrôle tout en essayant de les réduire est l'un des composants phare du protocole RPL. L'une des catégories d'approches pour optimiser RPL s'intéresse justement à Tricke timer. En effet, alors que la RFC 6206 qui décrit le principe de Trickle et son fonctionnement avertit les chercheurs/programmeurs sur la faible probabilité de d'améliorer Trickle, de nombreux travaux d'optimisation de RPL se sont intéressés à cet algorithme. Parmi ces travaux on peut citer Etricke pour Enhanced-Trickle décrit dans où les auteurs proposent un nouvel algorithme [35]

appelé E-Trickle qui n'a pas de période d'écoute. Au lieu de réinitialiser  $c$ , le compteur de cohérence, au début de l'intervalle, il réinitialise  $c$  à un moment choisi de manière aléatoire pour éliminer l'effet cumulatif du problème d'écoute courte. Les auteurs dans ont également proposé une version optimisée de Trickle (appelée opt-Trickle) qui, lorsqu'un nouvel intervalle commence, choisit les valeurs de  $t$  en fonction de l'état actuel. Si l'intervalle est réinitialisé, il choisit  $t$  de 0 à  $I_{min}$ , et s'il a été configuré ou démarré à partir d'un intervalle expiré, il choisit  $t$  de la moitié de l'intervalle à l'intervalle entier  $[I/2, I]$  [36].

### **2.12. Conclusion :**

Ce chapitre a été consacré à la description détaillée du protocole RPL. après l'avoir défini ,on a expliqué la procédure de construction du DODAG ,la fonction objective ,les types des messages de contrôles utilisées(DIO,DAO et DIS) ainsi que le fonctionnement de l'algorithme trickle .a la fin de ce chapitre ,nous avons exposé les deux mode de fonctionnement(Non-storting et storting) et les travaux d'optimisation du protocole RPL.

# **Chapitre 3 :**

## **Modélisation du protocole de routage Rpl**

## **1.1.Introduction :**

Le but de notre travail est de modéliser, vérifier et analyser les performances du protocole de routage RPL “Routing Protocol for Low-Power and Lossy Networks (RPL)” utilisé dans l’internet des objets. Notre travail est donc basé sur trois grandes étapes qui sont :

1. La modélisation : En utilisant les automates temporisés, afin de capter l’aspect temporel du protocole “RPL” ainsi que l’aspect fonctionnel.
2. La vérification : En utilisant la technique “Model Checking” en utilisant le model checker de l’outil UPPAAL.
3. Analyse des performances du protocole RPL avec l’outil COOJA et la simulation de l’outils UPPAAL..

Dans ce chapitre nous allons présenter la technique, le modèle et les outils nécessaires pour atteindre l’objectif fixé.

## **3.2. Présentation du Contiki/ Cooja [39] :**

Contiki est un système d’exploitation open source, hautement portable et multitâches pour les systèmes embarqués en réseau et les réseaux de capteurs sans fil à mémoire efficace.

Contiki est conçu pour les microcontrôleurs avec de petites quantités de mémoire. Contiki fournit une communication IP, à la fois pour IPv4 et IPv6. Contiki et sa pile uIPv6 sont certifiés IPv6 Ready Phase 1 et ont donc le droit d’utiliser le logo argenté IPv6 Ready. De nombreux mécanismes et idées clés de Contiki ont été largement adoptés dans l’industrie. Aujourd’hui il est utilisé par des centaines d’entreprises dans des systèmes tels que les navires de fret, les satellites et les équipements de forage pétrolier. Contiki et uIP sont reconnus par l’outil d’analyse de réseau nmap populaire. Les protothreads Contiki ont été utilisés dans de nombreux systèmes embarqués différents, allant des décodeurs de télévision numérique aux capteurs de vibration sans fil.

Contiki a présenté l'idée d'utiliser la communication IP dans les réseaux de réseaux de capteurs de faible puissance. Cela a ensuite conduit à une norme IETF et l'IPSO Alliance, une alliance internationale de l'industrie. Le système de construction Contiki est conçu pour faciliter la compilation des applications Contiki soit vers une plate-forme matérielle, soit vers une plateforme de simulation en fournissant simplement différents paramètres à la makecommande, sans avoir à modifier les fichiers makefiles ou modifier le code de l'application.

### **3.3. Simulateur Cooja [39] :**

Cooja est un simulateur/émulateur de réseau de capteurs appelé *motes* pour Contiki. Il permet de simuler les connexions réseaux et d'interagir avec les capteurs Le code exécuté par les nœuds est le même chargé sur des capteurs ou des nœuds physiques Les capteurs supportés par Cooja sont : Exp5483, z1, wismote, micaz, sky, jcreate, sentilla-usb, esb. Les motes peuvent être émulsés à haut niveau (niveau hardware) Résultat : long mais permet une inspection précise du comportement du système Les motes peuvent être aussi émulsés à niveau moins détaillé Résultat : une connexion rapide et permet la simulation d'un réseau plus large. L'interface du simulateur est composée de plusieurs plugins présents sous formes de fenêtres.

### **3.4.L'interface de simulateur Cooja [40] :**

Il existe 5 zones dans cette interface :

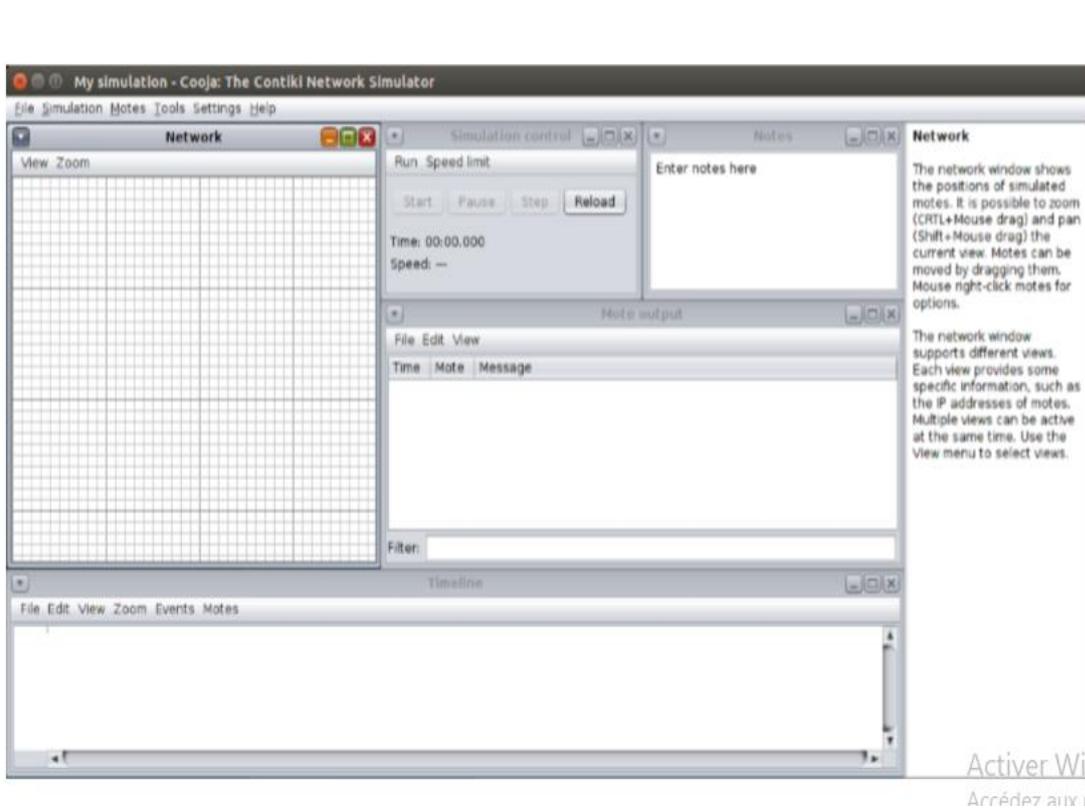
- La première zone porte le nom de "Network" : Cette zone permet de visualiser chaque nœud du réseau et de visualiser leur état (identifiant, adresse, LED, etc.) A l'initialisation de la simulation, cette zone est vide et il faut lui ajouter des nœuds.

- La seconde zone porte le nom de "Timeline" : Il s'agit d'une frise chronologique dans laquelle chaque les messages et les événements sont affichés. Elle permet par exemple de visualiser les envois de données entre les nœuds. Les messages à afficher doit être indiqué dans le code du nœud Claude Duvallat Simulation de réseaux de capteurs sans fils 11/29.

- La troisième zone porte le nom de "Mote Output" : Cette zone permet d'afficher toutes les sorties des différentes interfaces des nœuds. On peut disposer d'une fenêtre "Mote Output" différente pour chaque nœud.

- La quatrième zone porte le nom de "Control" : Cette zone est utilisée pour contrôler la simulation : démarrer, recharger, ou exécuter pas à pas. • Le temps d'exécution ainsi que la vitesse y sont affichés.

- La cinquième zone porte le nom de "Notes" : Il s'agit d'un espace permettant de prendre des notes sur la simulation en cours, dans l'optique de l'enregistrement de la simulation.



**Figure3.1** : capture d'écran de l'interface du simulateur COOJA [40].

### 3.5. Les différents outils nécessaires [40] :

#### Ant :

- Ant est un logiciel créé par la fondation Apache qui vise à automatiser les opérations répétitives du développement de logiciel telles que la compilation, la génération de documents (Javadoc) ou l'archivage au format JAR, à l'instar des logiciels Make

- Pour installer ce premier paquet, il faudra utiliser la commande suivante dans un terminal :

```
sudo apt-get install ant
```

- **Le compilateur MSP430 :**

Cooja simule les communications réseau en utilisant l'émulateur MSPSim pour émuler finement (au niveau des instructions) l'exécution d'un programme sur une plateforme basé sur un processeur MSP430

- Il nous faudra donc un compilateur adapté à cette architecture.
- La commande suivante permet son installation :

```
sudo apt-get install gcc-msp430
```

- **Le JDK :**

Apache Ant étant écrit en Java, il a besoin d'une machine virtuelle (JVM : Java Virtual Machine) pour fonctionner.

- Nous installerons donc Open-JDK (Java Development Kit) pour pouvoir l'utiliser.

```
sudo apt-get install openjdk-8-jdk-headless
```

### **3.6. Téléchargement et installation de Contiki :**

Contiki 2.7 peut être téléchargé à l'adresse suivante :  
<https://sourceforge.net/projects/contiki/files/Contiki/Contiki%202.7/contiki2.7.zip/download>

- Il faut ensuite décompresser le fichier obtenu dans le répertoire Home de votre utilisateur :

```
unzip contiki-2.7.zip
```

### **3.6. Démarrage de l'application :**

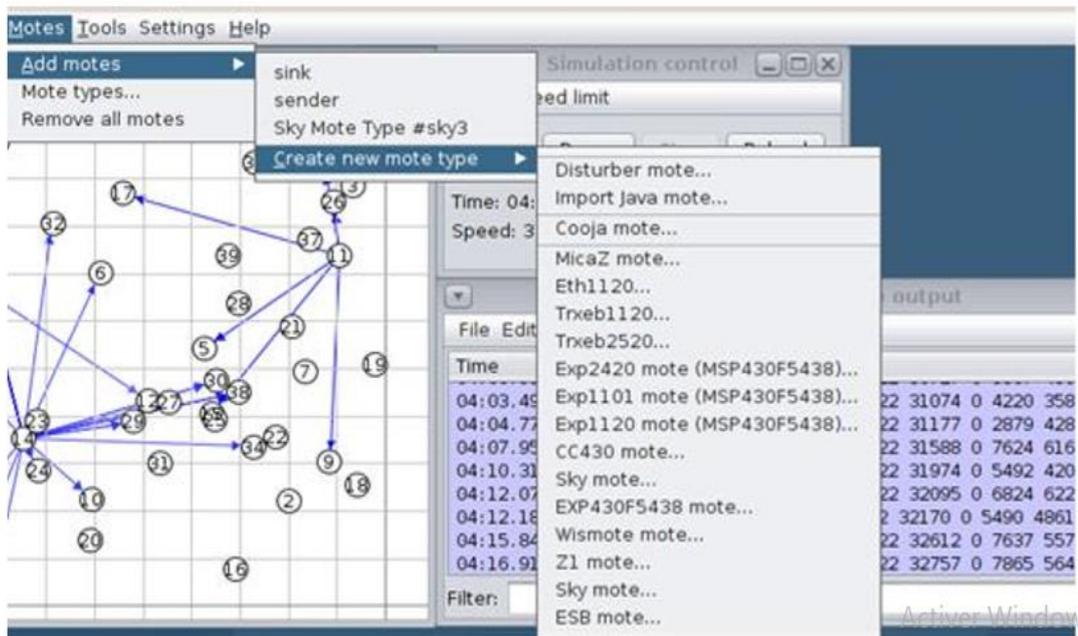
- Pour démarrer l'application, il suffit de lancer la commande suivante :

```
ant run
```

- Puis, on va créer une nouvelle simulation à partir du menu File / New simulation

- **Ajouter un nœud racine [39] :**

Motes → Add Motes → Create New Motes Type → Choisir le type de mote.



**Figure 3.2 :** Création d'un MOTE[39].

On ajoute une mote de type « Skymote », on implémente le code source udpsink qui est implémenté dans ... (ajouter le chemin), on a essayé de compiler le code pour voir s'il génère des erreurs, on peut faire Clear pour générer le code de nouveau parce que si le code est déjà compilé on remarque que le simulateur ne fait pas la compilation du code et surtout si on le modifie, par la suite on ajoute le nombre de nœuds souhaité, dans notre cas on a ajouté un seul nœud puisqu'il représente la racine.

- **Ajouter des nœuds Sender :**

On ajoute une mote de type « Skymote » (même type avec le nœud de la racine), puis on ajoute le code source udpsender de rplcollect, on a suivi les mêmes étapes comme on a fait pour la création du nœud de la racine, mais cette fois un nombre de nœuds différents pour qu'on puisse faire une interprétation pour étudier la consommation énergétique et le nombre de messages envoyés et le nombre de messages reçus.

### 3.7. Les paramètres estimés :

- **L'énergie :**

Nous étudions ici la consommation énergétique et les performances des objets par rapport à l'énergie utilisée et consommée tout au long de la simulation.

Pour faire cela nous avons un outil qui PowerTracker, permettre de connaître le taux et la durée d'utilisation des fonctions radio des objets.



Figure 3.3: utilisation du PowerTracker.

- **Transmission (TX) et réception (RX) :**

Les envois et la réception des paquets se font grâce au routage RPL qui permet entre autre de relayer les paquets entre capteurs .ce routage se fait grâce aux différents paquets RPL que l'on peut citer :

Message d'information DIO

Message d'information DAO

Message d'information DIS

Ce routage permet a tous les équipements de connaître l'ensemble des routes disponibles pour acheminer un paquet d'un a un autre .

On peut calculer le nombre total des messages envoyés/réceptionnés ,grâce a l'outil : "Mote output" du simulateur Cooja.

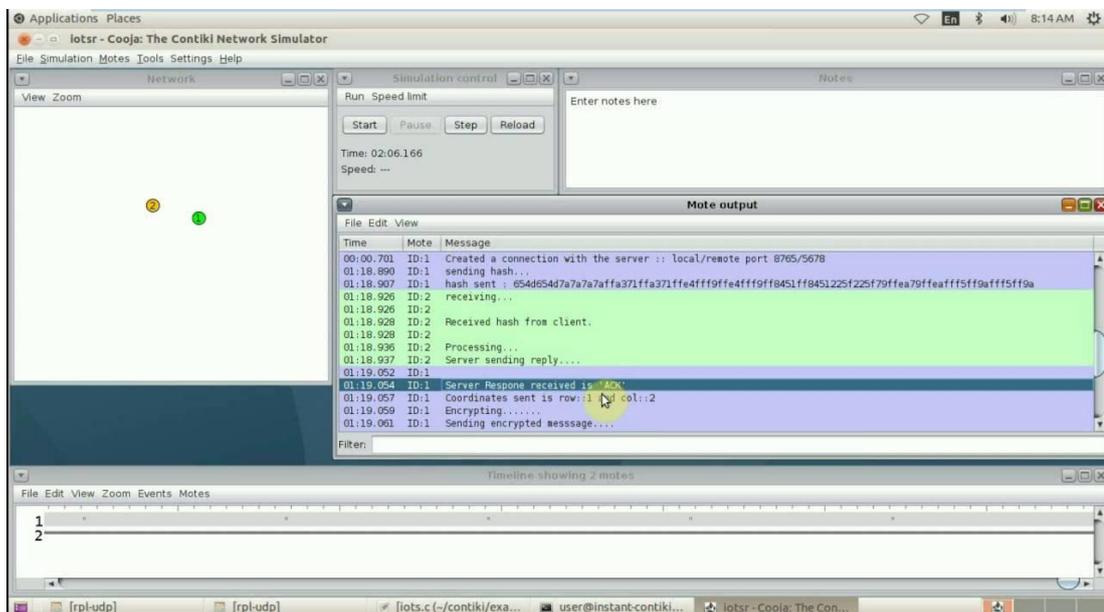


Figure3.4 : utilisation de l'outil "Mot output."

### 3.7.Paramètre de simulation :

Paramètres	Valeurs
Simulateur	Cooja
Temps	5 min
Nombre des nœuds	5, 15, 20, 25,30
Type des nœuds	SKymote
TX rang	32m
RX rang	32m
Consommation énergétique	CPU 1.8 mw
	Ecouter 20.0mw
	Transmettre 17.7 mw

Tableau3.1 : Paramètre de simulation.

### **3.8.Le rôle du système Cooja / Contiki dans notre projet :**

Pour implémenter notre solution proposée dans le contexte de surveillance des composants de réseau nous avons utilisé un Environnement de développement simplifié Instant Contiki sous la forme d'une machine virtuelle VMWARE qui contient tout le code Source de Contiki et toutes ses fonctionnalités à l'aide d'un simulateur de réseau adapté à nos besoins COOJA. Ce système nous permet de construire, implémenter et tester le protocole de surveillance des composants du réseau en utilisant IPv6 Routing Protocol for Low-Power and Lossy Networks (RPL) et les résultats de simulation sera afficher dans le quatrième chapitre.

### **3.9.Avantage de la simulation [39] :**

- Un faible cout.
- Une grande flexibilité
- Un contrôle total de la plate-forme expérimentale.

### **3.10.Limité de la simulation (Puissance et performance du simulateur):**

- On ne peut pas exécuter n'importe quelle taille d'expérience
- Le temps d'exécution d'une expérience peut s'avérer être très long
- Modèles de trafic utilisé : on ne peut pas aisément injecter des traces de trafic réelles dans l'expérience.

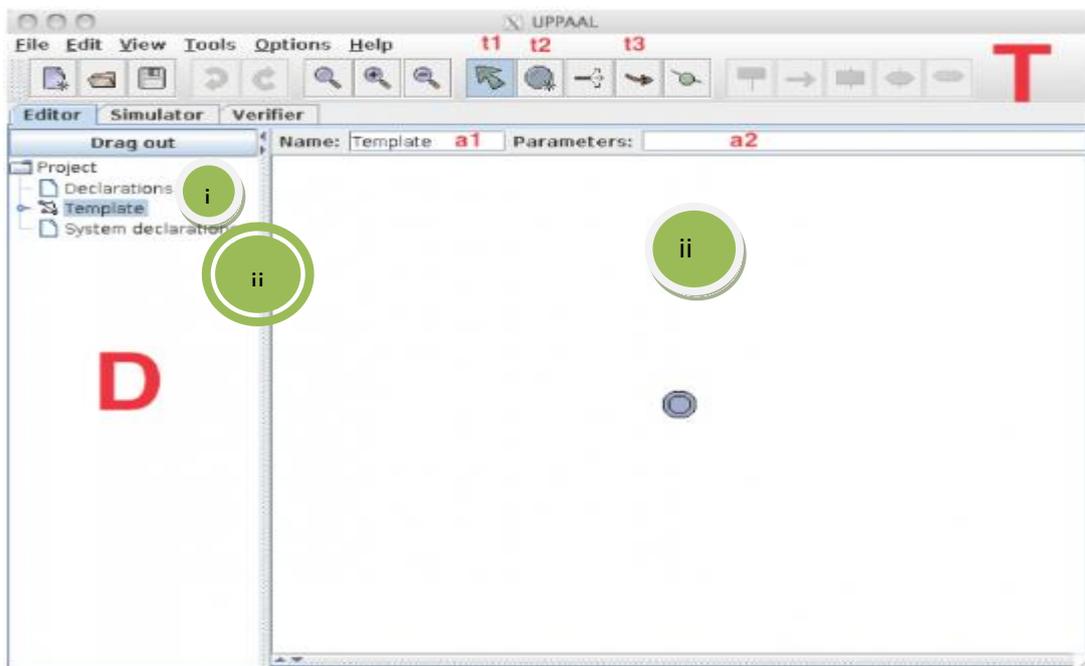
### **3.11.L'outil UPPAAL :**

Uppaal est une boîte à outils pour la vérification des systèmes temps réel développés conjointement par les universités d'Uppsala et d'Aalborg. Il a été appliqué avec succès dans des études de cas allant des protocoles de communication aux applications multimédias. L'outil est conçu pour vérifier les systèmes pouvant être modélisés comme des réseaux d'automates temporisés étendus avec des variables entières, des types de données structurés, des fonctions définies par l'utilisateur et la synchronisation de canaux. La première version d'Uppaal a été publiée en 1995. Depuis lors, il a connu un développement constant. Les expériences et les améliorations comprennent des structures de données, une réduction partielle de la commande, une version distribuée d'Uppaal, une accessibilité guidée et à un coût minimal des travaux sur les graphiques UML Statecharts ect ,Uppaal a également généré des doctorats associés. Il comporte une interface utilisateur Java et un moteur de vérification écrit en C ++. Il est disponible gratuitement sur <http://www.uppaal.com/> [21].

### 3.12. Syntaxe d'UPPAAL [41]:

Un modèle UPPAAL est basé sur les automates temporisés. La description d'un modèle se compose de trois parties :

- i) les déclarations globales et locales,
- ii) les modèles d'automates, et
- iii) la définition du système.



**Figure3.5** : l'interface graphique de l'outil UPPAAL.

- **Déclarations :**

Les déclarations peuvent être locales ou globales. UPPAAL permet de déclarer des horloges (type clock), des entiers (type int), des booléens (type bool), des canaux de synchronisation (type chan), des tableaux, des structures de données (type struct), et de définir des types de données (typedef).

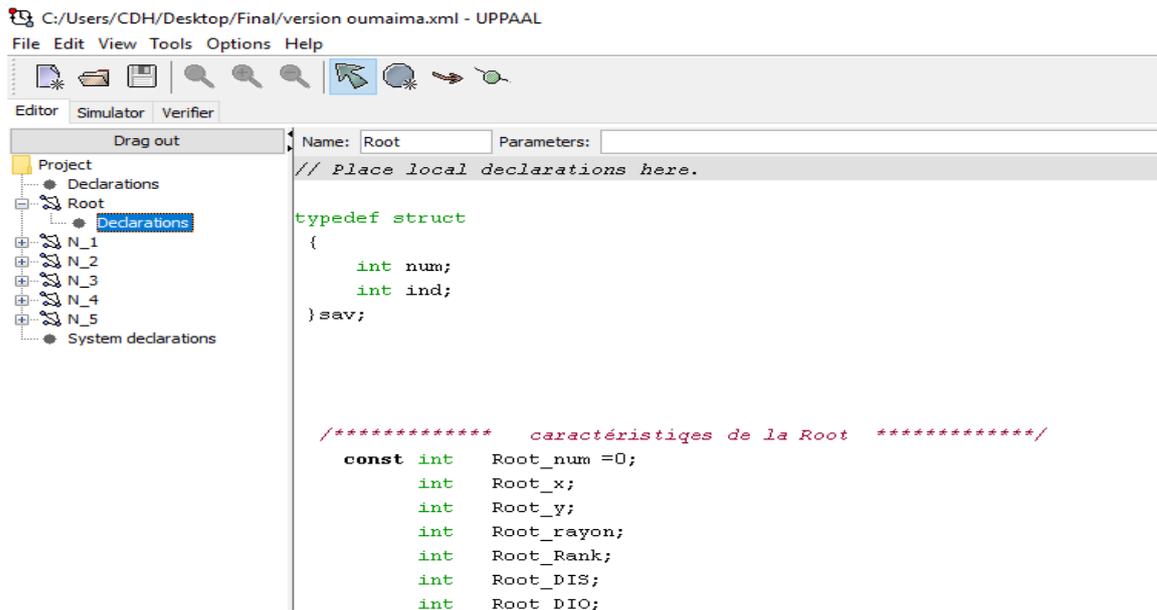
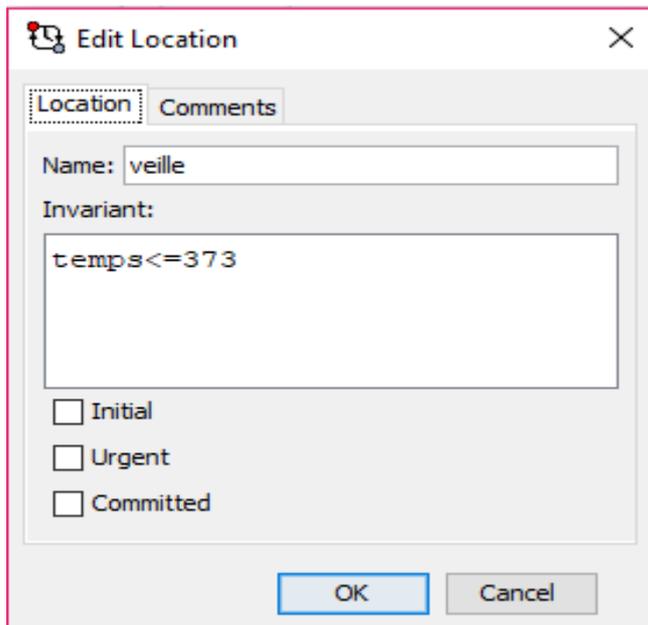


Figure 3.6 : partie déclaration dans UPPAAL.

- **Locations :**

Un automate temporisé est considéré comme un graphe à états. Les locations représentent les états dans ce graphe. Elles se communiquent via les transitions. Un identifiant optionnel peut être associé à chaque location afin de l'identifier. Elles peuvent être étiquetées par des invariants. Un invariant peut être une expression composée d'un ensemble de contraintes sur des horloges, une différence entre des horloges ou des expressions booléennes ne faisant pas intervenir des horloges. UPPAAL distingue trois types de locations :

- **Location initiale** : Chaque automate doit avoir une location initiale représentée par un double cercle.
- **Location urgente** : La location urgente ne permet pas de faire passer le temps quand le processus s'y trouve. Sémantiquement, une location urgente est équivalente à une location dont l'invariant est  $x \leq 0$  ou  $x$  est une horloge remise à zéro sur toutes les transitions entrantes de la location en question.
- **Location engagée** : La location engagée ne permet pas l'écoulement du temps quand un processus s'y trouve. Elles sont utiles pour assurer l'atomicité de la synchronisation entre trois processus ou plus. Elles sont plus prioritaires que les locations urgentes.



**Figure3.7** : interface de location.

- **Transitions :**

Les locations sont connectées entre elles par les transitions (edges). Ces dernières sont étiquetées par des sélections, des gardes, des synchronisations et des mises à jour des variables et des fonctions.

**-Sélection** : Cette option permet d'associer, d'une façon indéterminée, à une variable donnée une valeur dans un intervalle bien déterminé. Les gardes, la synchronisation et les mises à jour peuvent être spécifiées en fonction de la valeur sélectionnée.

**-Garde** : Une transition est franchie si et seulement si la garde spécifiée est satisfaite. Les gardes expriment des contraintes d'horloges ou de données nécessaires pour franchir une transition.

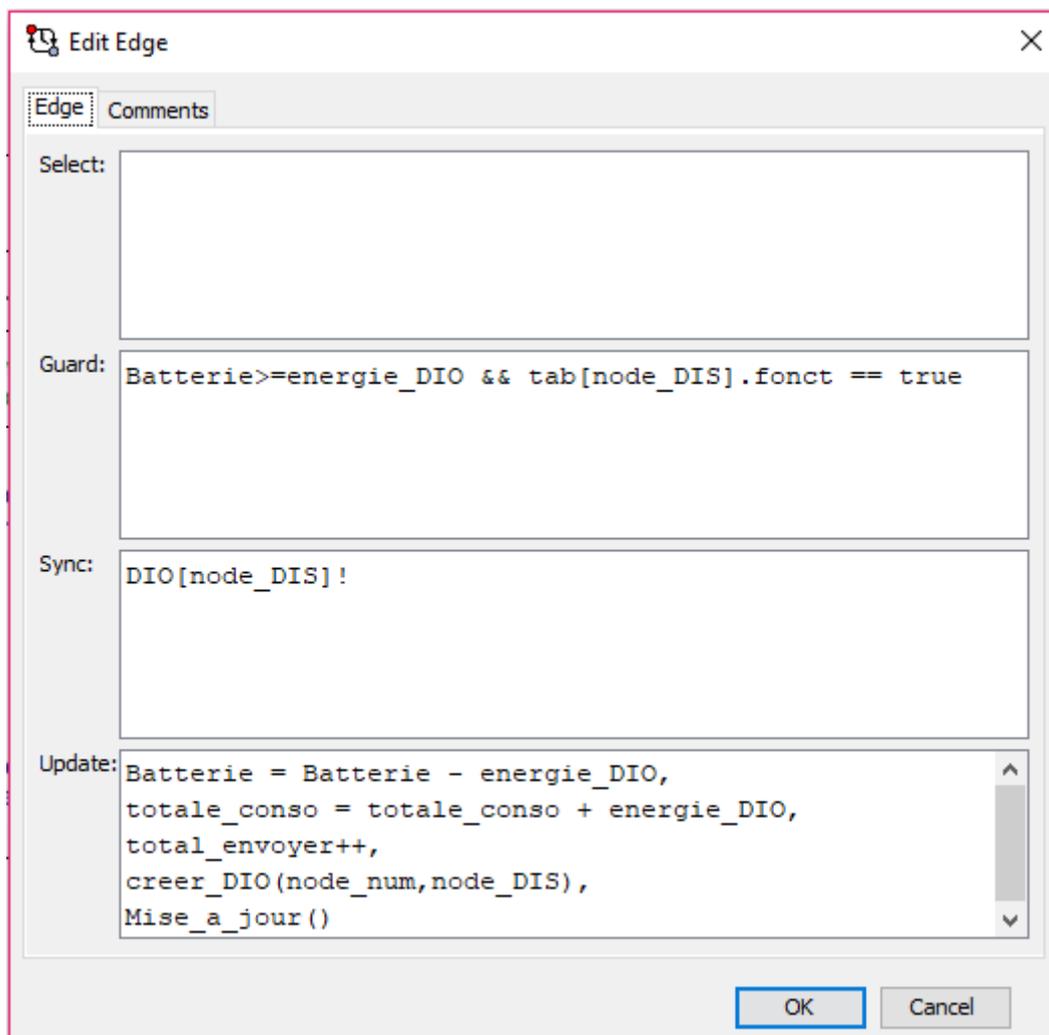
**-Synchronisation** : Les différents automates dans un modèle UPPAAL peuvent se synchroniser via des canaux de synchronisation permettant l'envoi et la réception d'un message. Deux transitions dans deux processus différents peuvent se synchroniser si et seulement si les gardes des deux transitions sont satisfaites et elles admettent deux étiquettes de synchronisation  $a1 !$  (envoi) et  $a2 ?$  (réception) où  $a1$  et  $a2$  représentent le même canal de synchronisation. Quand deux processus se synchronisent, les deux transitions sont franchies au même temps. Les mises à jour de la transition portant

l'étiquette a! se font avant celles de la transition portant l'étiquette a! ?. Il existe deux types de canaux de synchronisation :

i) Les canaux de synchronisation binaires qui permettent de synchroniser deux processus via un seul canal de synchronisation a (au moyen de deux actions a! et a?) et

ii) Les canaux de synchronisation de diffusion (broadcast) qui permettent de synchroniser un processus avec plusieurs autres processus. Une transition avec une étiquette de synchronisation a!, avec "a" est un canal broadcast, peut synchroniser avec chaque transition ayant une étiquette de synchronisation a ? et dont les gardes sont satisfaites.

**-Mise à jour** : L'exécution de la mise à jour sur une transition permet de changer l'état du système. Une mise à jour est une liste d'expressions séparées par des virgules et exécutées dans un ordre séquentiel. L'opérateur d'assignement est " = ".



**Figure3.8** : l'interface de transition.

### 3.13. Description globale de l'outil UPPAAL :

Uppal se compose de trois parties principales [43]:

- **Un langage de description** : est un langage de commande surveillé non déterministe avec des types de donnée (par exemple tableaux, délimitée entiers, etc..) il sert de modélisation ou de conception de langage pour d'écrire le comportement du système Comme des réseaux d'automates étendus avec horloges et des données variables.
- **Le simulateur** : est un outil de validation qui permet l'examen de possibles exécutions dynamiques d'un système au début de la conception (ou de modélisation). Il offre ainsi un moyen peu coûteux de détection des défauts avant vérification par le vérificateur de modèle exhaustif qui couvre le comportement dynamique du système.
- **Le modèle vérificateur** : peut vérifier l'accessibilité et l'invariabilité par l'exploration des propriétés de l'état espace d'un système, c'est-à-dire l'accessibilité de l'analyse en termes symboliques représentés par des contraintes.

Un autre élément important pour l'efficacité est l'application d'une technique qui réduit symboliquement la vérification des problèmes à celle de la manipulation efficace et la résolution de contrainte.

Pour faciliter la modélisation et le débogage, le vérificateur du modèle Uppaal génère automatiquement une trace de diagnostic qui explique pourquoi la propriété est (ou n'est pas) satisfaite par un modèle du système.

### 3.14. La modélisation du protocole RPL :

Nous proposons des modification sur le model formel conforme au fonctionnement de protocole RPL (expliquer dans le deuxième chapitre),les modification touche que les nœuds "sender "pour améliorer et plus préciser les résultats précédents obtenu .

On a deux model le première représente le nœud « Root » et le deuxième représente les nœuds « sender (s)»

### 3.15. La déclaration générale :

On a la structure générale des nœuds comme suite :

**const int nb\_noeud** = nombre des nœuds.

**int num** : numero du nœud.

**int x, int y** : les coordonnee de nœud.

**int rayon** : rayon de nœud pour calculer la zone de transmission

**int Rank** : le rang du nœud

**int DIS** : le numero du nœud qui a envoyer le message DIS.

**int DIO** : le numero du nœud qui a envoyer le message DIO

**int DAO** : le numero du nœud qui a envoyer le message DAO

**int DAO\_ACK** : le numero du nœud qui a envoyer le message DAO\_ACK

**int DATA** : le numéro du nœud qui a envoyer le message DATA

**bool connecte** : variable pour vérifier la connexion du nœud au réseau.

**bool fonct** : variable pour vérifier le fonctionnement du nœud au réseau.

**noeud tab[nb\_noeud]** :table de voisinage.

**clock temps** : horloge définit la durée de l'état veille.

### 3.16. Les méthodes :

Voici quelque méthode déclarée dans les nœuds :

void donne\_rank() :une fonction pour donner le rang du nœud.

void creer\_DATA() :créer un message DATA.

void supp() : fonction pour suppression du reseau.

Void Mise\_a\_jour() :une fonction pour savoir quels messages ont atteint le nœud.

### 3.17. La modélisation de nœud « root » :

Lors du démarrage du Root à partir l'état « démarre », le root a deux possibilités soit il se connecte (il passe a l'état « Connecte»), ou il reste déconnecte (et il passe a l'état «Non\_connec »).

Etat « non connecter » : lorsque le root est déconnecter diffuse un message DIO pour connaitre ses voisins.la reconnaissance des voisin se fait par la methode Voidprep\_diffuse\_Root().

Il change son état vers l'état connecté « etat connect »tout au changement la valeur de la variable Bool Root\_Connect de « false ».

Etat « connect » : quand il est connecté a un reseau, il peut effectuer trois fonction :

- Recevoir le message DATA et connaitre sa destination en utilisant la fonction « Void rec\_data ()», et exécute la fonction « Void routage () »pour trouver le meilleur chemin vers la destination, puis il revient a l'état « connecter ».

- Recevoir le message DIS des nœuds qui viennent de se connecter au réseau ou qui viennent de rejoindre le réseau après une déconnexion, et envoyez-leur un message DIO, puis recevez un message DAO et envoyer un message DAO\_ACK pour se connecter.

- Recevoir le message DAO et envoyer un message DAO\_ACK.

Dans le modèle a des transitions étiquetées par :

- Des gardes par exemple : « Batterie  $\geq$  energie\_noactive » avant qu'une émission ou réception se fait, energie doit être considérée pour savoir si le taux de la batterie est suffisait pour une telle operation .

- Synchronisation : lors de l'envoi ou la réception des messages comme DAO[Root\_num] ?, DIO[Root\_DIS] !.

- Mise à jour :

Batterie=Batterie -energie \_noactiv,

Total\_cono=totale\_conso+energie\_noactiv ,

la figure suivante illustre la modelisation du « Root » :

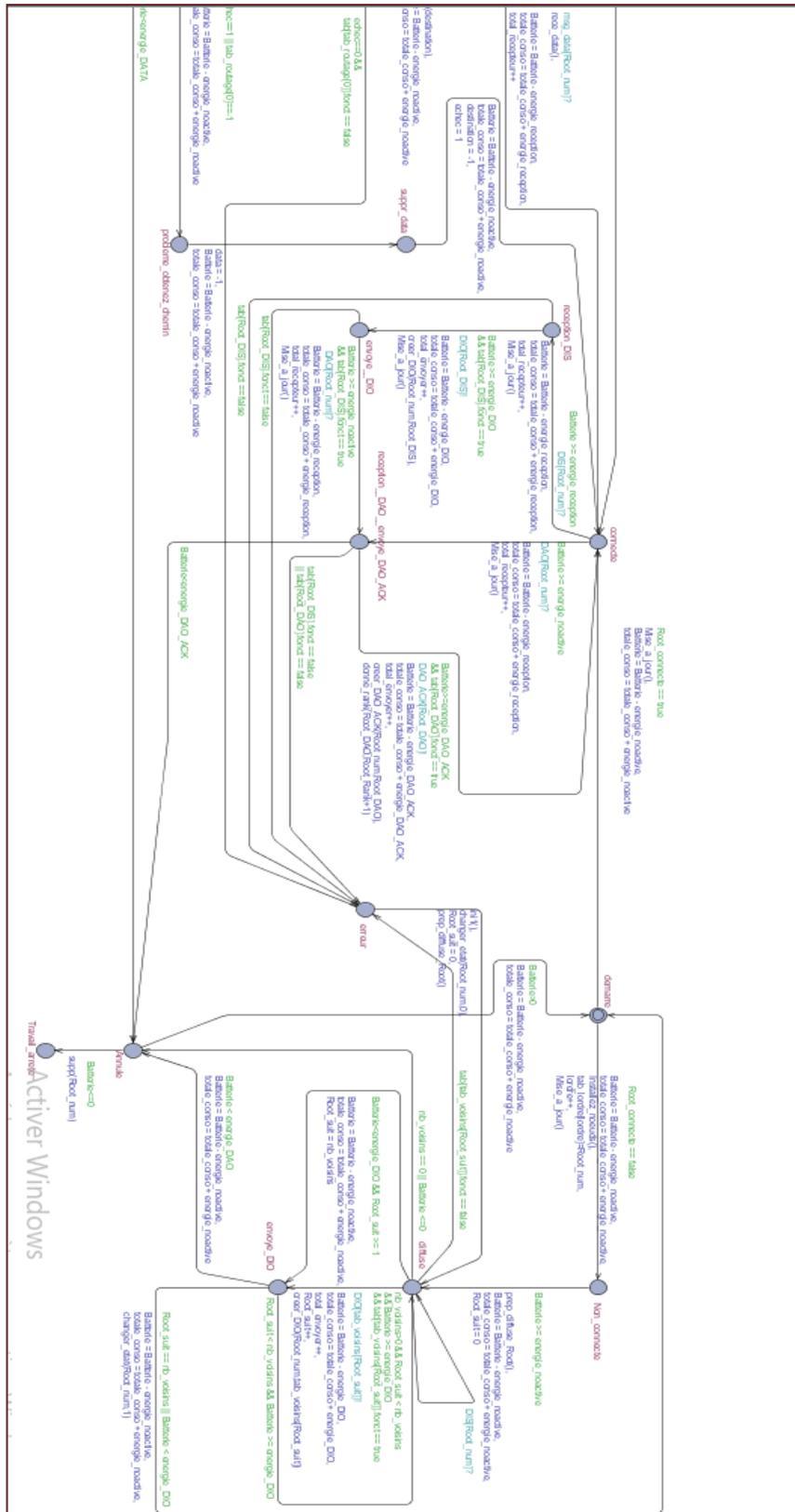


Figure3.9 : le model du de root.

### 3.18. La modélisation du nœud :

Lors du démarrage d'un nœud à partir l'état « démarre » ce nœud doit passer par l'état « non\_connecte » pour recevoir message DIO de ses père .une fois ce message est réceptionné, le nœud envoie un acquittement ,et diffuse lui-même un message DIO a ses enfants et il passe à l'état « connect »

Et pour minimiser la consommation d'énergie. Les nœuds (Matériel) tombe en veille en cas de non activité .nous avons modélisé ça par l'ajout d'un état « veille » tel que l'invariant déclaré dans l'état « veille » "**temps**<273" selon le type de radio « CC1200 » .si le nœud reste inactif, c'est-à-dire , il ne recoit aucun message de ces types DIO, DIS, DAO\_ACK.

Nous avons ajouté un horloge « temps » initialiser par 0 à partir de l'état « démarre » quand le nœud pas connecté selon la garde "**nœud\_connect**= false" et l'horloge **temps** est inférieure strictement à 273 le nœud passe à l'état « veille »,et pour réveiller le nœuds attend jusqu'à l'horloge **temps** <=373 pour retourner à l'état « démarre » puisque quand est veille considéré comme n'est pas connecter pour ce la il doit retourne à l'état « démarre » pour connecter sur le réseau à nouveau ,cette changement d'état exigent des mise à jour sur la consommation de la batterie et l'horloge de temps et la batterie de nœuds en fin la mise à jour de la table de voisinage et la tables des parents .

- **La modélisation du nœud dans le cas « non connecter »**

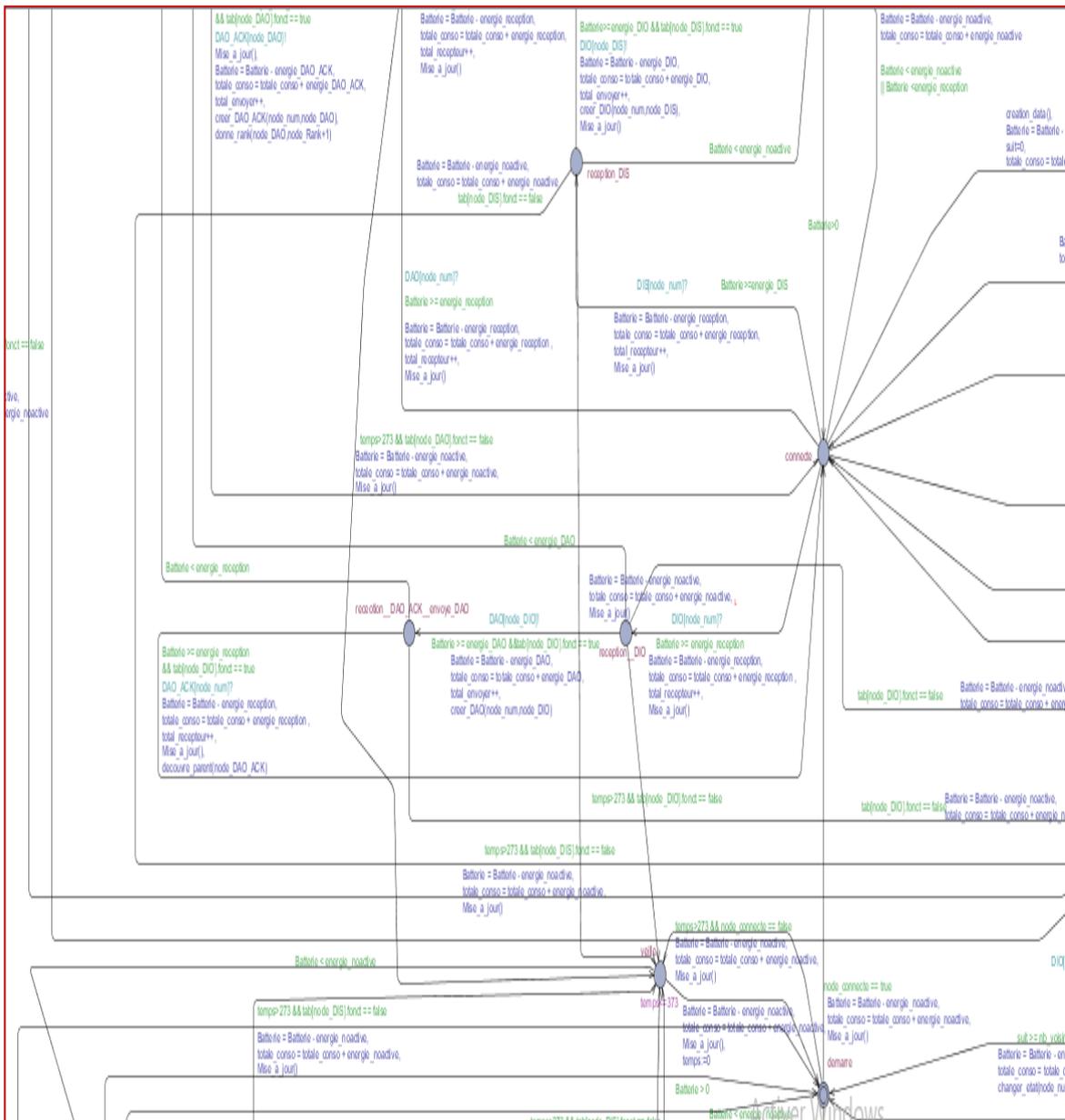
Cette figure représente le nœud dans le cas non connecté et les possibilités de réception message DIS ou bien l'état de diffusion « diffuse ».



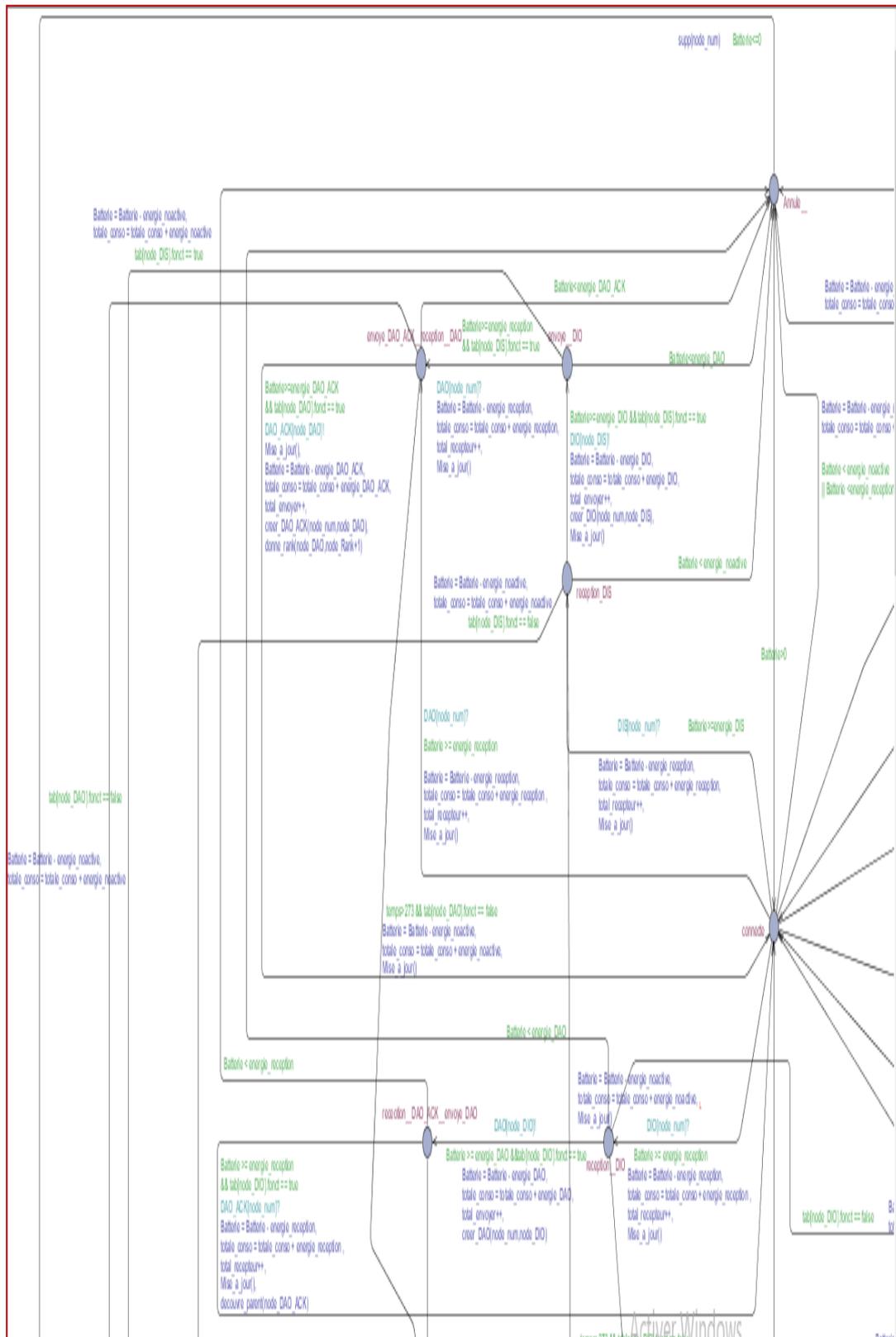
Figure 3.11: le modèle nœud dans le cas Non\_\_Connect

- La modélisation dans le cas « connecter » :

La figure 3.12 représente le nœud dans le cas connecté et les états d'échange des message (DIS ,DIO... ) avec la possibilité de tombe en veille .



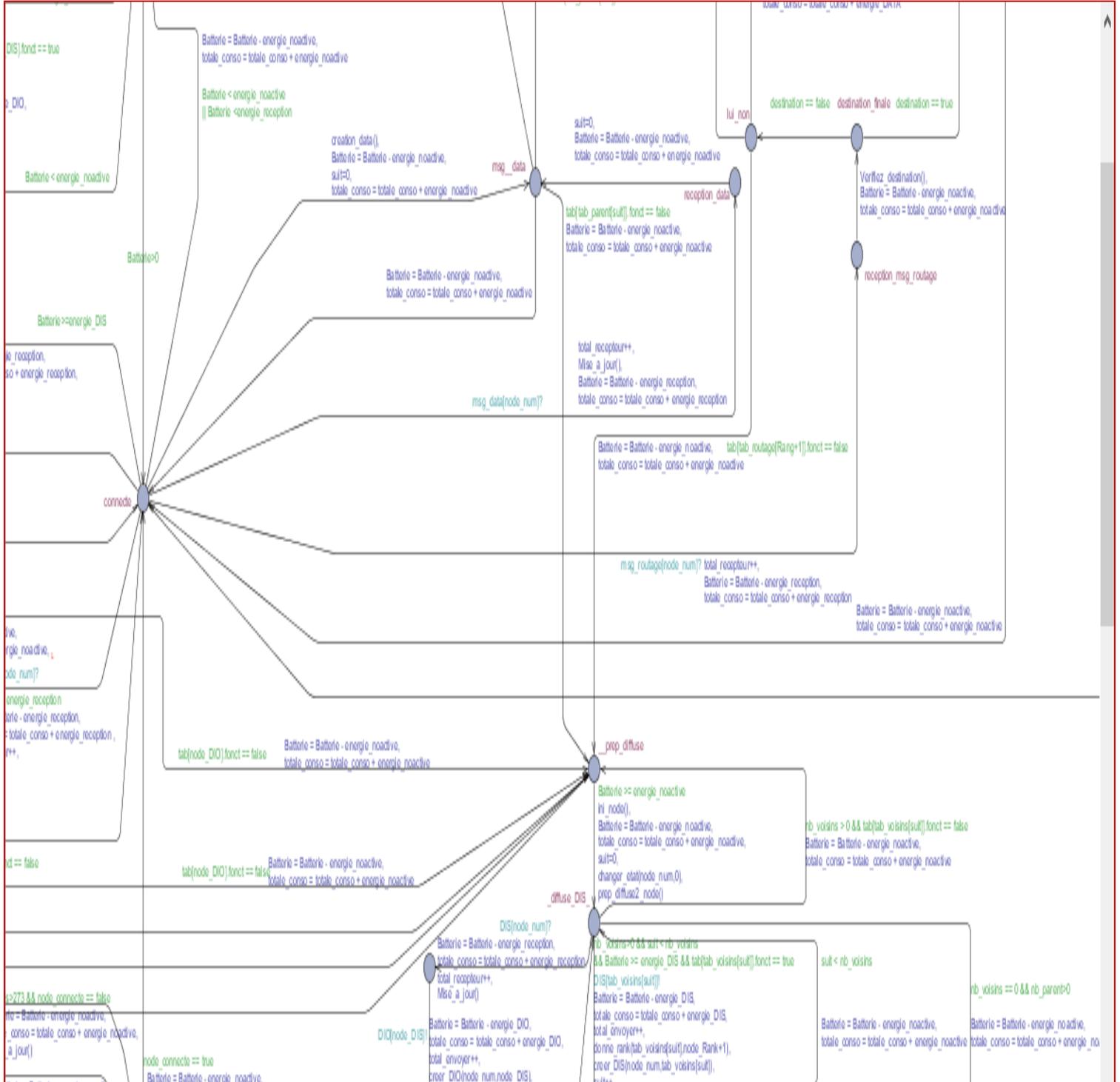
**Figure 3.12** : le modèle nœud dans le cas « Connect » et le cas de « vielle ».



La figure 3.13 représente le nœud dans le cas connecté et l'état quand annuler le nœud.

**Figure 3.13** : le modèle de nœud dans le cas « Connect ».

la figure suivante représente le nœud dans l'état connecté et la possibilité de réception



un message data ou message de routage.

**Figure 3.14** : le modèle de nœud dans le cas « Connect ».

### **3.16.Conclusion :**

Ce chapitre a été consacré pour présenter la modélisation de protocole dans les deux outils « Cooja et Uppaal ». Nous avons réussi à modéliser la station de base (Root) ainsi que les noeuds. Cette modélisation prend en charge les fonctionnalités de base de chacun des noeuds. Ces fonctionnalités se résument dans l'envoi et la réception des intérêts, des événements et des renforcements positifs. Cette modélisation nous a aidé à mieux comprendre le fonctionnement de protocole de routage pour l'internet des objets « RPL ».

# **Chapitre 4 :**

## **Vérification et analyse de performance**

## 1.1.Introduction :

Le but de ce chapitre est de valider notre modélisation faite par les outils UPPAAL et Cooja. Cette validation sera faite après une comparaison entre les résultats obtenus par les deux outils. Cette comparaison est le fruit de l'extraction de valeurs obtenues en utilisant des métriques spécifiques pour ces réseaux. Nous terminons ce chapitre par la discussion des résultats obtenus.

## 4.2.La simulation de RPL dans UPAAL :

- **Initialisation :**

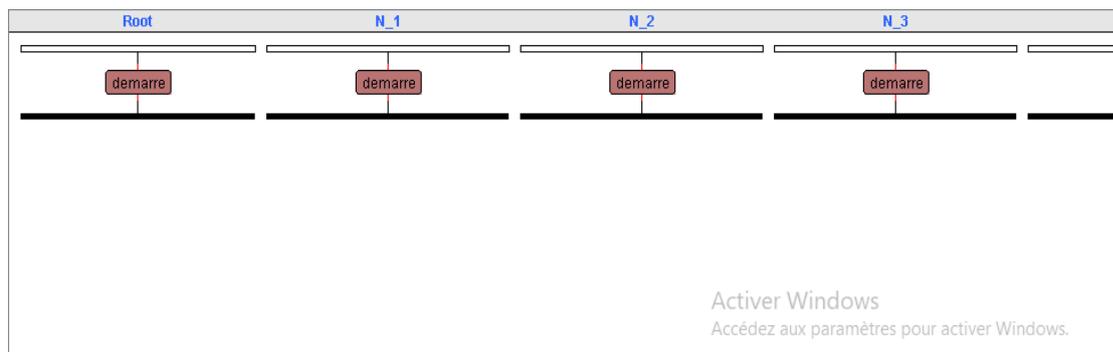


Figure 4.1 : l'initialisation des nœuds.

- **Cas de veille :**

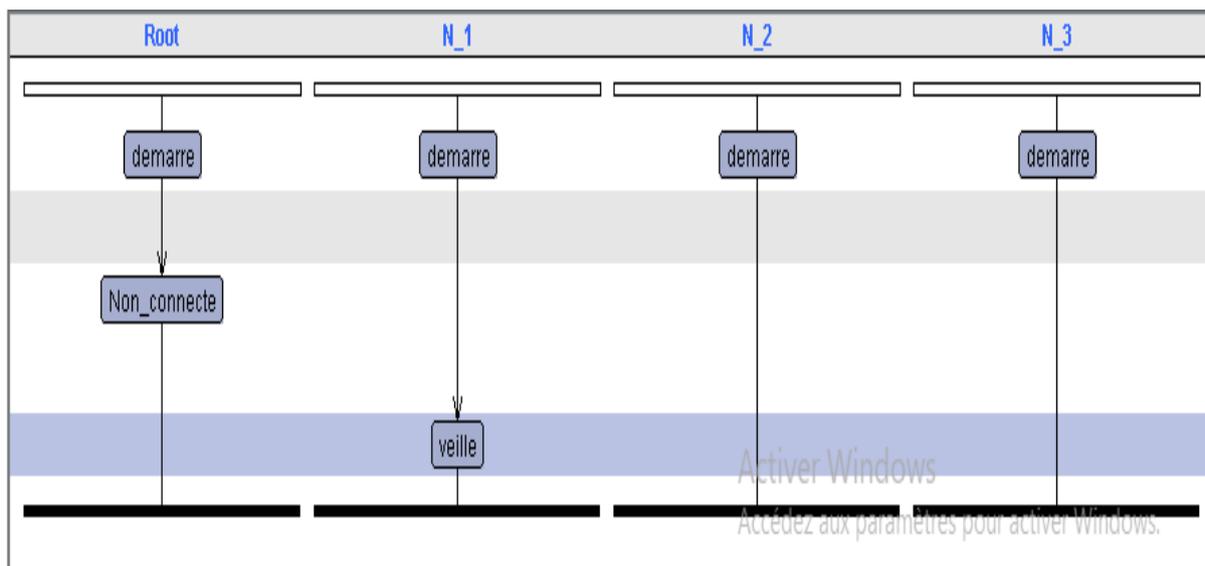


Figure 4.2 : le cas de veille.

- Cas d'envoyer message DIO :

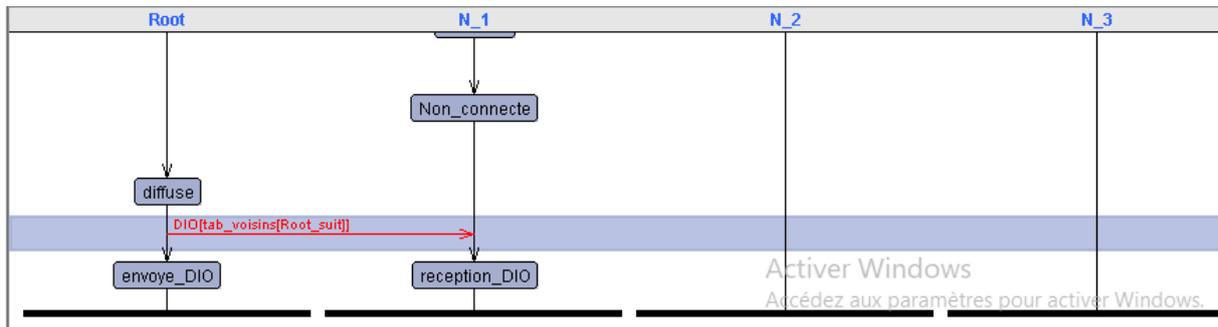


Figure 4.3 : envoyer le message DIO par le root.

- Cas d'envoyer message DAO :

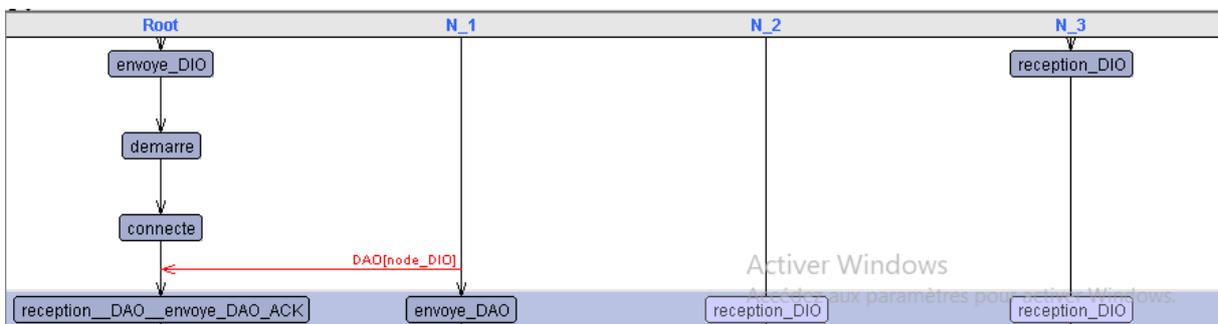


Figure 4.4 : envoyer le message DAO par le nœud N\_1.

- Cas d'envoyer message DAO\_ACK :

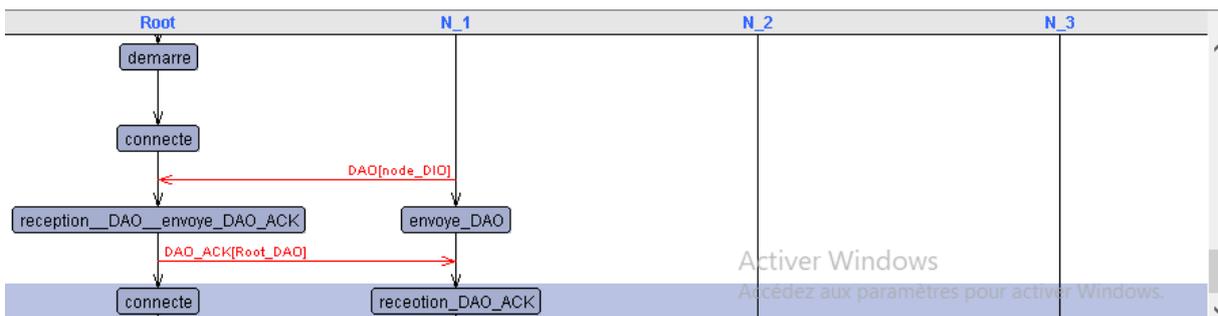


Figure 4.5 : envoyer le message DAO\_ACK par le Root.

- Cas d'envoyer message\_DATA:

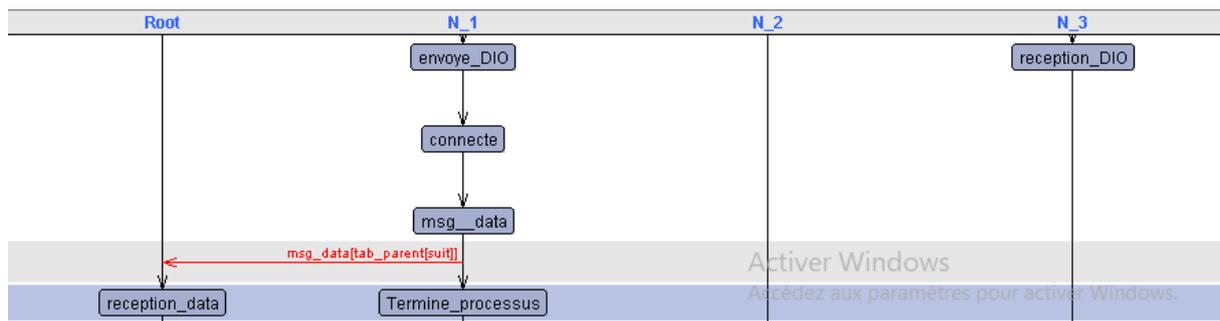


Figure 4.6 : envoyer le message\_DATA par le nœud N\_1.

- Cas d'annuler le nœud :

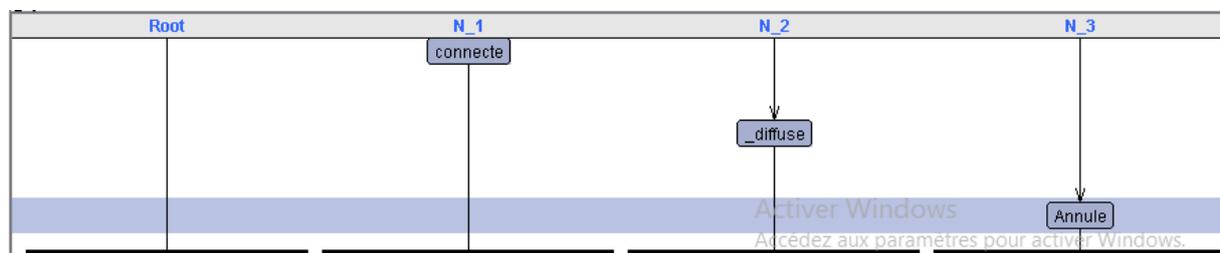


Figure 4.7: cas d'annulation un nœud.

### 4.3.La vérification RPL dans UPAAL :

L'objectif principal de l'usage de l'UPPAAL c'est de vérifier certaines propriétés dans le système spécifié. Nous avons établi les propriétés suivantes qu'on va essayer de les vérifier dans cette section.

Exemple des propriétés a vérifié :

- L'initialisation de Root et les nœuds (du le nœud numero 1 jusq'a 5) :

E<> Root.demarre.

E<> N\_1.demarre.

E<> N\_2.demarre.

E<> N\_3.demarre.

E<>N\_4.demarre.

E<>N\_5.demarre.

- La connectivité :

$A \leftrightarrow \text{Root}.\text{Root\_connecte} == \text{false} \text{ imply } N_3.\text{node\_connecte} == \text{true}$

- La vielle des nœuds : on n'eglier le Root pisque nous considérons que ce dernier a une source d'énergie.

$E \leftrightarrow N_1.\text{demarre}.$

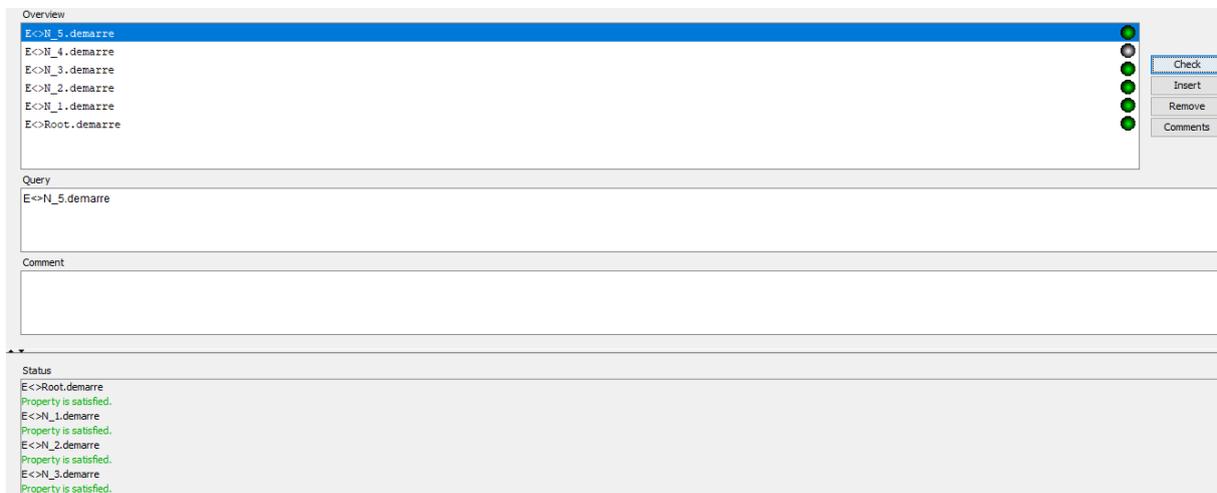
$E \leftrightarrow N_2.\text{demarre}.$

$E \leftrightarrow N_3.\text{demarre}.$

$E \leftrightarrow N_4.\text{demarre}.$

$E \leftrightarrow N_5.\text{demarre}.$

- **L'initialisation :**



**Figure 4.8:** vérifie L'initialisation des nœuds.

- **La connectivite :**

Le Root n'est pas connectée au réseau et le Nœud 4 est connecte.

Query
A<>Root.Root_connecte==false imply N_3.node_connecte==true
Comment

Status
Established direct connection to local server. (Academic) UPPAAL version 4.0.13 (rev. 4577), September 2010 -- server. Killing server! Disconnected. A<>Root.Root_connecte==false imply N_4.node_connecte==true Established direct connection to local server. (Academic) UPPAAL version 4.0.13 (rev. 4577), September 2010 -- server. <b>Property is not satisfied.</b> A<>Root.Root_connecte==false imply N_3.node_connecte==true <b>Property is not satisfied.</b> A<>Root.Root_connecte==false imply N_3.node_connecte==true <b>Property is not satisfied.</b>

**Figure 4.9:** vérifie la connectivite des nœuds.

- **La ville :**

Query
E<>N_1.veille
Comment

Status
(Academic) UPPAAL version 4.0.13 (rev. 4577), September 2010 -- server. Disconnected. Established direct connection to local server. (Academic) UPPAAL version 4.0.13 (rev. 4577), September 2010 -- server. E<>N_1.demarre <b>Property is satisfied.</b> E<>N_1.veille Disconnected. E<>N_1.veille Established direct connection to local server. (Academic) UPPAAL version 4.0.13 (rev. 4577), September 2010 -- server. <b>Property is satisfied.</b>

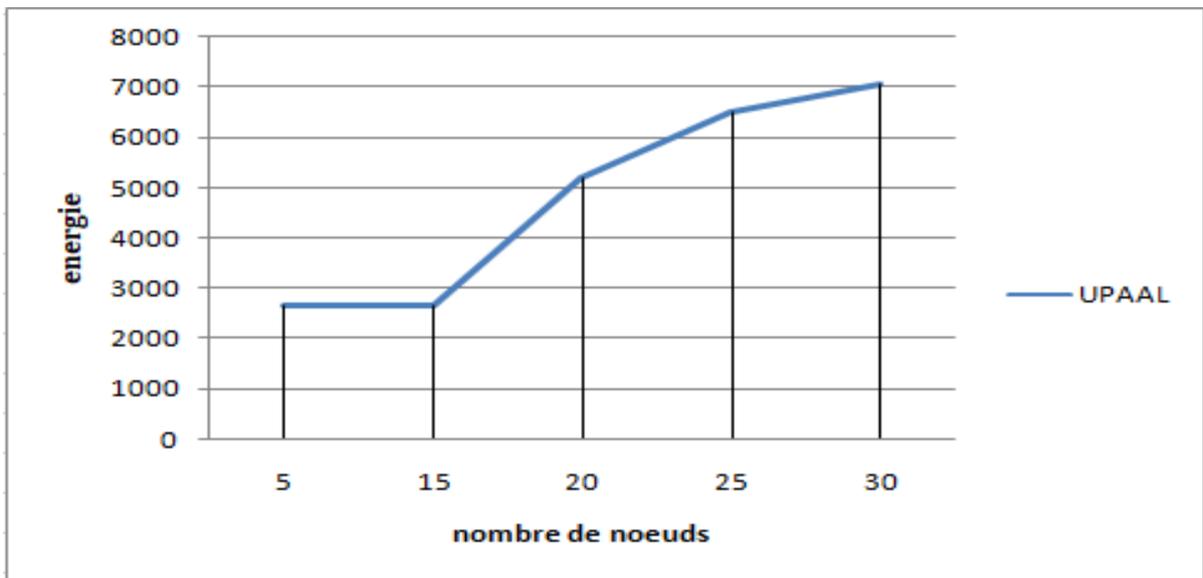
**Figure4.10 :** vérifier la veille des nœuds.

#### 4.4. Résultat de l'analyse de performances de RPL dans UPAAL:

Il existe plusieurs métriques pour analyser les performances d'un RCSF, par ces métriques nous avons utilisé les suivantes :

- Consommation totale d'énergie.
- Nombre de messages envoyés par tous les nœuds (Transmission TX).
- Nombre de message reçu par tous les nœuds (réception RX).

Consommation d'énergie vs nombre de nœud :



**Figure 4.11** : énergie Vs nombre de nœuds dans UPAAL.

-La figure 4.10 représente les valeurs que nous avons eu après la simulation paramétré en fonction de nombre de nœud, On observe que dans l'intervalle de 5 a 15 nœuds l'énergie totale consommé est stable, mais a partir le nombre 20 nous remarquons l'augment considérablement d'énergie.

Transmission vs nombre de nœud :

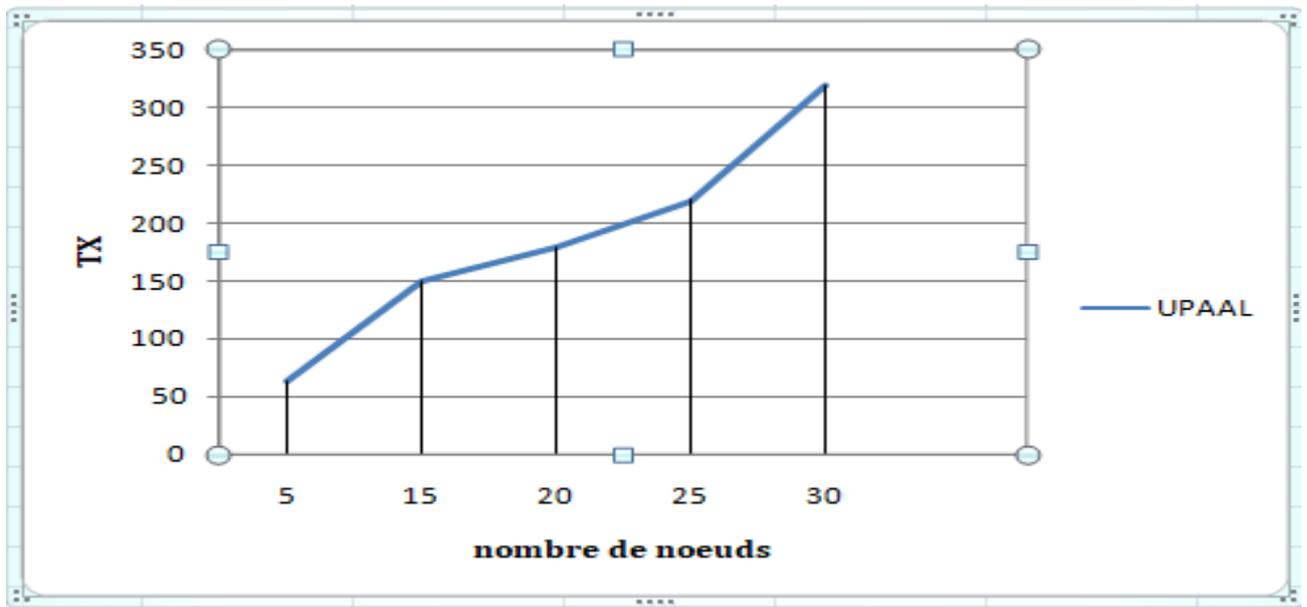


Figure 4.12 : TX vs nombre de nœud dans UPPAL.

Réception vs nombre de nœud :

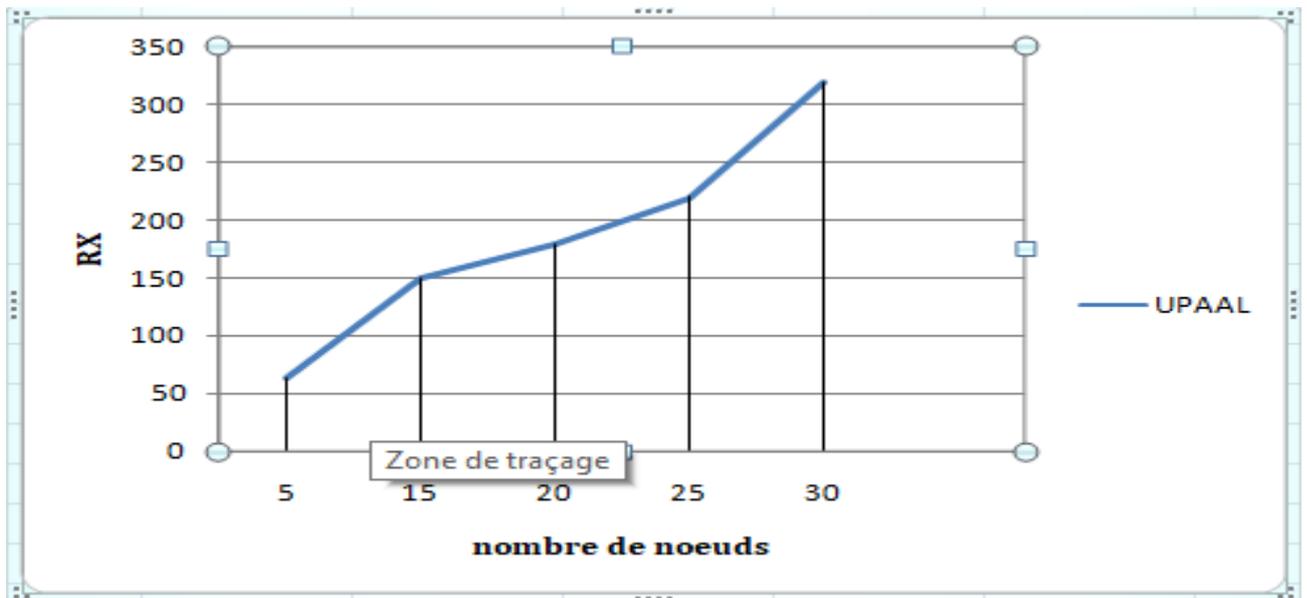


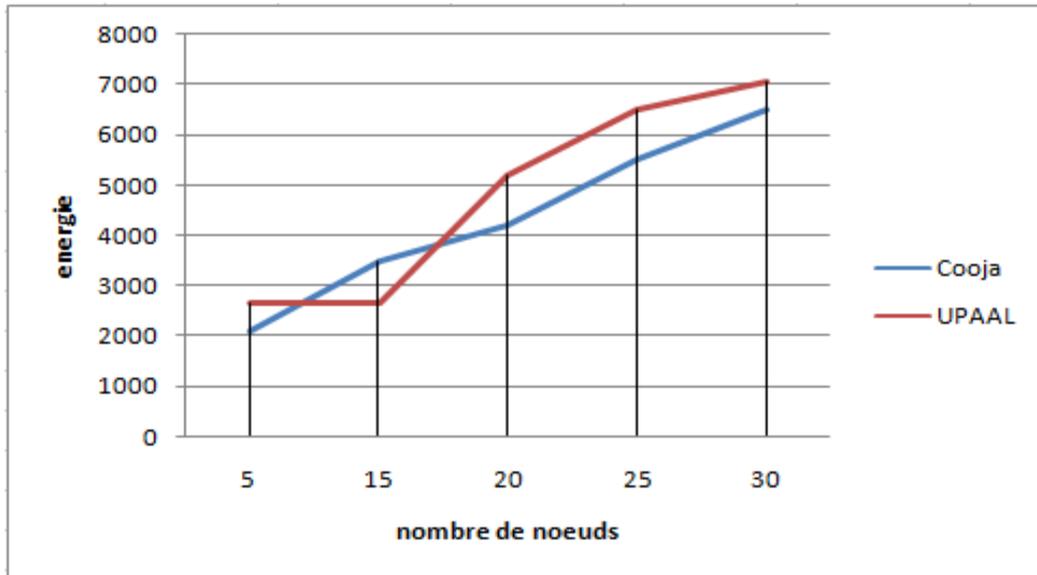
Figure 4.13 : TX vs nombre de nœud dans UPPAL.

Les figures 4.11 et 4.12 représente l'échange des messages de notre réseau en fonction de nombre des nœuds, on remarque que le trafic de réseau augmente (RX/TX) quand le nombre de nœud augmente mais partir le nombre 25 les transmissions et les réceptions augmentent considérablement.

#### 4.5. Comparaison des résultats avec le simulateur Cooja :

Après la modélisation et simulation de protocole RPL nous comparons les résultats obtenus d'UPPAAL avec les résultats de COOJA pour valider notre modélisation

La consommation d'énergie :



**Figure 4.14 :** comparaison de la consommation d'énergie vs nombre de nœud.

En observant la figure 4.13, nous constatons que les courbes presque sont dans le même plage des valeurs ,il est claire qu'il y a une similitude entre les deux courbes dans l'intervalle de 5 a 17 nœuds a partir cette nombre de nœud avec un petite différence ,après cette nombre de nœud ,nous observons que la différence augmente entre les deux outils jusqu'a le nombre 30 j'observe que la différence est démunie entre les deux courbe cette phénomène se refer a le comportement des nœuds en plus de leur nombre.

### La transmission :

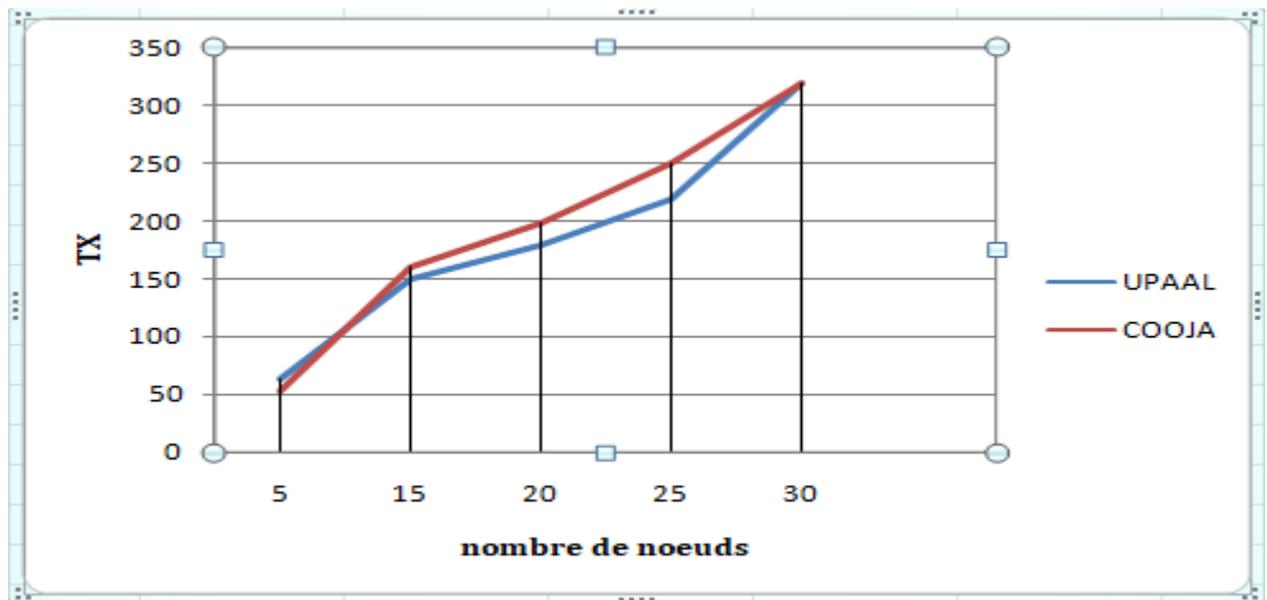


Figure 4.15 : comparaison de la Transmission vs nombre de nœud.

### La réception :

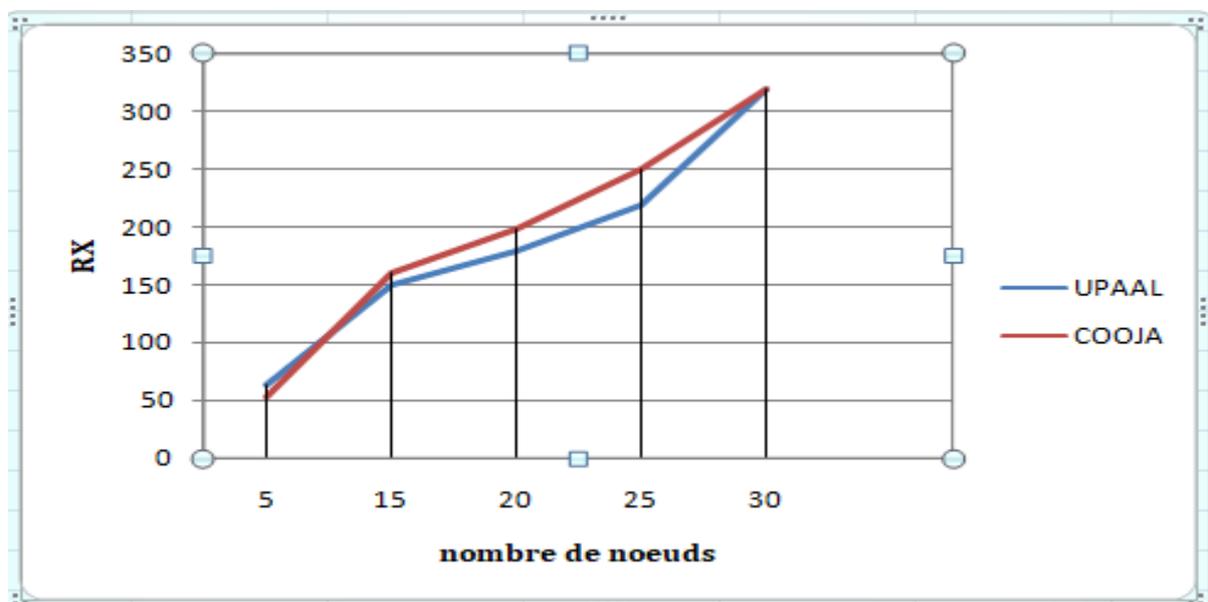


Figure 4.16 : comparaison de la réception vs nombre de nœud.

On explique les deux figure (4.14 et 4.15) en même temps car le nombre de message émis égal au nombre de messages reçu les deux figure représente le nombre de réception et transmission en fonction du nombre de nœud je remarque une similitude entre les deux trace puisque on prend en considération la veille de nœud dans notre modélisation sur UPPAL.

#### **4.5.Discussion :**

Après la comparaison des résultats obtenus dans les deux outils UPPAL et COOJA, nous pouvons conclure que la modification sur le modèle du protocole RPL sous UPPAAL est plus au moins valide car ce dernier est facile pour la modélisation puisque utiliser le langage plus proche au langage C et utiliser les automates temporisés et la vérification des propriétés dans UPPAL plus facile par une seule expression logique par rapport à Cooja nécessite de maîtriser plusieurs langages et des fichiers pour compléter la procédure de vérification .

#### **4.6.Conclusion :**

Ce chapitre a été consacré pour présenter notre travail .Nous avons vérifié quelques propriétés après la modification du protocole en utilisant le Model Checker de l'outil UPPAAL. Nous avons pu simuler le mode de fonctionnement du protocole avec l'outil de simulation Cooja et le simulateur UPPAAL. Nous avons poussé au maximum de ces limites de fonction du côté consommation d'énergie et aussi du côté d'échange des messages de transmission et réceptions ce qui nous a permis de tracer des courbes et de déduire les capacités de ce protocole. Nous avons comparé les résultats pour valider notre modélisation et nous avons les discutés pour tirer des conclusions sur notre travail et comment l'améliorer.

## **Conclusion générale :**

Comme on le voit, la technologie IoT est de nos jours très fondée et nous pensons que dans les prochaines années à venir, la plupart des objets de la vie quotidienne seront connectés entre eux et à Internet.

Le protocole de routage RPL est motivé par la nécessité de soutenir l'infrastructure telle que le domaine agricole et les applications de réseau domestique.

En dépit de certains inconvénients, comme le manque de documentations pratiques et facilité de prise en main de Cooja Ce rapport présente une application du protocole RPL comme une étude de cas d'un réseau IoT. Nous avons étudié quelques concepts du RPL ,et la simulation. Nous avons également essayé d'apprendre et d'acquérir plus d'expérience de l'utilisation de les outils Cooja et UPAAL, et avons proposé de nouvelles analyse.

Et enfin, nous espérons pouvoir continuer de cette manière pour l'étude du monitoring des nœuds.

### **Ce travail nous a aidés a :**

- Comprendre des aspects précise sur l'internet des objets et les défis de conception qui leurs sont attachés.
- Acquérir des connaissances plus au moins avancées sur la modélisation et la simulation par les outils UPPAAL er Cooja.
- Mieux comprendre le fonctionnement du protocole RPL a travers des simulations.
- Vérifier des propriétés attendues de ce protocole.
- Analyser des performances de protocole.

### **Nous allons essayer d'améliorer ce travail de la manière suivante :**

- Prend en charge le changement de la topologie pour des cas spécifiques (des nœuds qui se déplacent).
- Utiliser un modèle statistique sur UPPAL-SMC pour mieux analyser les performances de ce protocole.
- Optimiser le modèle si possible.

- [1] Dave Evans, " L'Internet des objets Comment l'évolution actuelle d'Internet transforme-t-elle le monde ?" Cisco Internet Business Solutions Group (IBSG), Avril 2011.
- [2] Imad Saleh<sup>1</sup>, " Les enjeux et les défis de l'Internet des Objets (IdO) Issues and Challenges of the Internet of Things (IoT)" , Laboratoire Paragraphe Université Paris 8, [imad.saleh@univ-paris8.fr](mailto:imad.saleh@univ-paris8.fr)
- [3] <http://www.smartgrids-cre.fr/index.php?p=objets-connectes-definition>.
- [4] "Les Cahiers du CiTC -eurarFid Collection CiTC, analyse et perspectives de l'internet des objets", Juillet 2011.
- [5] Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash. "Internet of things : A survey on enabling technologies, protocols, and applications. IEEE Communications Surveys & Tutorials", 17(4) :2347–2376, 2015.
- [6] Y.ait mouhoub, F.Bouchebbah . " Propotion d'un modèle de confiance pour l'internet des Objets", Université A/MIRA de Bejaia , 2015 .
- [7] Paul Pinault @disk\_91 , "Introduction aux objets connectés" , <https://www.disk91.com> ZZ promo 2000 / F1.
- [8] Helder D. Silva<sup>1</sup>, Pedro Macedo<sup>1</sup> and Luis A. Rocha , " Article An Enhanced Reservation-Based MAC Protocol for IEEE 802.15.4 Networks " José A. Afonso<sup>1,\*</sup>,<sup>2</sup> Department of Industrial Electronics, University of Minho; E-Mails: dsilva@dei.uminho.pt (H.D.S.);, Received: 27 January 2011; in revised form: 24 March 2011 / Accepted: 28 March 2011 / Published: 30 March 2011.
- [9] Jose A.Gutierrez, Edgar H.Callaway, Raymond L.Barrett , "Low –Rate Wireless per-sonal Area Networks :Enabling Wireless Sensors with IEEE 802.15.4" in ,IEEE press,2003.
- [10] Jean-Pierre Hauet Membre émérite de la SEE, " L'Internet des objets Deux technologies clés : les réseaux de communication et les protocoles", Rédacteur en chef de la REE, Article extrait de REE 2016-5.
- [11] "Analysis of Routing Protocol for Low-Power and Lossy Networks in IoT Real Time Applications " G. Gautham Krishnaa, G. Krishnaa, Dr. N. Bhalajiaa Department of Information Technology, SSN College of Engineering, Kalavakkam, Chennai 603 110.
- [12] sergine Makhtar Dioum, "Evaluation de l'impact du, mécanisme du « dutycyc» sur la qualité de service," Université de lorraine ,France.
- [13] Patrick Olivier KAMGUEU, THESE EN COTUTELLE : Configuration Dynamique et Routage pour l'Internet des Objets", pour l'obtention du Doctorat de l'Université de Lorraine (France) et de l'Université de Yaounde 1 (Cameroun), présentée et soutenue publiquement le 18 Décembre 2017.
- [14] T. Winter, Ed, A. Brand Sigma Designs, Arch Rock Corporation, R. Kelsey Ember Corporation,

P. Levis Stanford University, K. Pister Dust Networks, R. Struik Struik Security Consultancy, JP. Vasseur Cisco Systems, R. Alexander Cooper Power Systems," Internet Engineering Task Force (IETF), March 2012.

[15] [https://anrg.usc.edu/contiki/index.php/Trickle\\_library](https://anrg.usc.edu/contiki/index.php/Trickle_library).

[16] B.Romdhani, "Exploitation de l'hétérogénéité des réseaux de capteurs et d'actionneurs dans la conception des protocoles d'auto-organisation et de routage", Thèse de doctorat, en informatique, Institut National des Sciences Appliquées de Lyon , juillet 2012.

[17] FAQ:"RPL Non-Storing mode in AD6LoWPAN",  
<https://ez.analog.com/wirelessensor/networks/ad6lowpan/w/documents/557/faq-rpl-non-storing-mode-in-ad6lowpan>.

[18] "MERPL:A More Memory-efficient Storing Mode in RPL", Wei Gan Zhiqiang Shi Limin Sun SKLOIS Institute of Information Engineering, CAS Beijing, China ganwei,shizhiqiang,sunlimin@iie.ac.cn, Dan Ionescu SITE University of Ottawa Ottawa, Canada [ionescu@site.uottawa.ca](mailto:ionescu@site.uottawa.ca).

[19] A Tutorial on Uppaal 4.0 Updated November 28, 2006 Gerd Behrmann, Alexandre David, and Kim G. Larsen Department of Computer Science, Aalborg University, Denmark {behrmann,adavid,kg1}@cs.auc.dk.

[19.a] fun Mooc," Objets communicants & Internet des Objets "

[20] Micheal j.McGrath et Cliodhna Ni Scanaill , sensor technologies :health care : wireless and environmental applications ,Apress open, 336 Pages 26 Décembre 2013

[21] Mémoire Présenté A L'UNIVERSITÉ DE TLEMCEN FACULTÉ DE TECHNOLOGIE DÉPARTEMENT DE TELECOMMUNICATION Présenté par : TALEB Omar MANKOURI Abdelkrim Sur le thème "Programmation de la sécurité Internet des Objet".

[22]The advantages and disadvantages of Internet Of Things , prateek saxena ,15 juin 2016

[23] Un 360° pour bien les comprendre, Décembre 2016.

[24] Networking Protocols and Standards for Internet of Things , Tara Salman (A paper written under the guidance of Prof. Raj Jain)

[25] Olfa Gaddour, Anis Koubâa, "RPL in a nutshell, Polytechnic" Institute of Porto, 02.14.2013.

[26] Aishwarya Parasuram, David Culler Ed, Randy Katz Ed, "An Analysis of the RPL Routing Standard for Low Power and Lossy Networks" University of California at Berkeley, May 14, 2016.

- [27] RFC 6550, T. Winter et al, "IPv6 Routing Protocol for Low-Power and Lossy Networks", March 2012.
- [28] Mr. Tall HAMADOUN, "Réseau maillé sans fil à basse consommation énergétique", Mémoire de Stage de fin d'études Master Informatique, institut de la Francophonie Lyon 1 .
- [29] RFC 6719, O. Gnawali, P. Levis, "The Minimum Rank with Hysteresis Objective Function", September 2012
- [30] Pietro Gonizzi, Riccardo Monica, Gianluigi Ferrari, "Design and Evaluation of a DelayEfficient RPL Routing Metric", Université of Parma Italie, 2012 .
- [31] P. Karkazis, et al, RPL modeling in J-Sim platform, Ninth International Conference on Networked Sensing Systems, 2012.
- [32] Olfa Gaddour, Anis Koubâa, Nouha Baccour, Mohamed Abid, OF-FL : QoS-Aware Fuzzy Logic Objective Function for the RPL Routing Protocol, 2014.
- [33] Simon Duquennoy, Olaf Landsiede, Thiemo Voigt, Let the Tree Bloom : Scalable Opportunistic Routing with ORPL, Sweden, 11–15, 2013
- [34] Olaf Landsiedel, Euhanna Ghadimi, Simon Duquennoy, Mikael Johansson, Low Power Low Delay : Opportunistic Routing meets Duty Cycling, Sweden, 16–20, 2012
- [35] Baraq Ghaleb, Ahmed Al-Dubai, Elias Ekonomou, E-Trickle: Enhanced Trickle Algorithm for Low-Power and Lossy Networks, Oct.2015
- [36] Badis Djamaa and Mark Richardson. "Optimizing the Trickle Algorithm." IEEE Communications Letters 19.5 (2015): 819-822.
- [37] Master Ingénierie logiciel pour la société numérique , "Simulation dans Internet of Things", Réalisé Par : Abdouramane Mahamane Idrissa Amadou Abdoulaye Sy Bechir ARBI Gaby Gadigbe Kossi .
- [38] "SIMULATION D'UN RÉSEAU DE CAPTEURS SANS FILS AVEC COOJA "SFAX, 24-26 NOVEMBRE 2017 WORKSHOP INTERNET DES OBJETS, Claude Duvallet, Université du Havre UFR Sciences et Techniques.
- [39] Majda Moussa, "Vérification et configuration automatiques de pare-feux par model checking et synthèse de contrôleur", Université de Montréal, 2014.
- [40] Mr Kaid Omar Ilyes, "Modélisation d'un contrôleur de température d'un réacteur d'avion par Uppaal", Mémoire fin d'étude licence en informatique de l'université d'oran 2008-2009.
- [41] <https://blog.octo.com/modeles-architectures-internet-des-objets/>.
- [42] Lagustais, "Supervision Iot : Superviser les objets connectés", décembre 15, 2018
- [43] Guy landry DJATCH SIMO lips, "Routing Protocol for Low-Power and Lossy Networks"

[44] IPv6 Routing Protocol for Low- Power and Lossy Networks Speaker : Chih-Ching Chen  
Advisor : Dr. Ho-Ting Wu 2014/3/24 1.

[45] akifmufti , "What is RPL , and how to Simulate it in Cooja ?" August 18, 2016 .

[46] <https://www.futura-sciences.com/tech/definitions/internet-wi-fi-1648/>

[47] <https://www.silicon.co.uk/mobility/4g/bluetooth-5-1-location-accuracy-241087>

[48] <https://www.we-n.eu/loffre/formation/802-15-4-et-zigbee-lessentiel-pour-developper-un-produit/>

[49] <http://zone-astuces.blogspot.com/2010/07/les-topologies-reseaux.html>