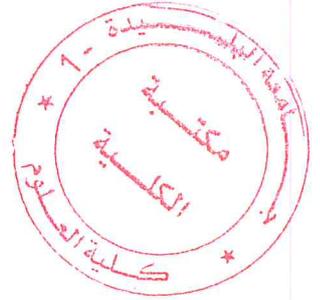


RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE  
MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA  
RECHERCHE SCIENTIFIQUE



UNIVERSITÉ SAAD DAHLEB BLIDA  
FACULTÉ DES SCIENCES  
DÉPARTEMENT INFORMATIQUE

Mémoire de fin d'étude  
Pour l'obtention d'un diplôme de Master en Informatique  
Spécialité : Systèmes Informatiques et Réseaux

Thème

Proposition et Evaluation de Performances d'une nouvelle  
fonction  
objective pour l'équilibrage de charge dans L'Internet des Objets

Réalisé par :

- EL BEY AISSA Sihem
- HASSNAOUI Asmaa

Encadré par :

- Mr. BOUCHAMA Nadir

Organisme d'accueil : Centre de Recherche sur l'Information Scientifique et Technique

Jury :

Président : Mme. ABED Hafida

Maitre Assistant A à USDB 1.

Examineur : Mr. DJEDDAR Afrah

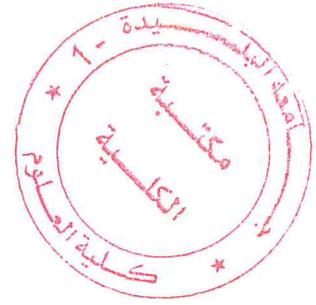
Maitre Conférence B à USDB 1.

Promoteur : Mr. OULD-KHAOUA Mohamed

Professeur à USDB 1.

Date de Soutenance : 27/10/2018

Promotion : 2017/2018



## Résumé

Grâce à l'Internet des Objets (Internet of Things, IoT), des objets peuvent être interconnectés et suivis à distance via une infrastructure réseau existante. Chaque objet connecté est identifiable de façon unique grâce à un système embarqué. L'IoT a des applications dans plusieurs domaines tels que les villes intelligentes, les distributions intelligentes d'électricité (Smart Grids), le suivi de la pollution et de l'environnement, la domotique, etc. La plupart des noeuds dans un réseau IoT ont des capacités réduites en termes de mémoire, puissance de calcul, énergie, etc. L'objectif de l'équilibrage de charge est d'équilibrer le trafic entre les différents noeuds du réseau de telle sorte à améliorer la durée de vie totale. Pour ce faire, on peut tenir compte de plusieurs paramètres tels que le taux d'occupation des buffers dans les noeuds, la capacité de stockage, la bande passante disponible, la charge d'un noeud, etc. L'objectif de ce travail est :

- Etudier les différentes fonctions objectives proposées pour l'équilibrage de charge dans les réseaux IoT ;
- Proposer une nouvelle fonction objective pour l'équilibrage de charge dans l'Internet des Objets ;
- Implémenter le nouveau protocole de routage sur un simulateur ;
- Evaluer les performances du nouveau protocole de routage ;

**Mots clés :** Internet des Objets (IoT), équilibrage de charge, fonction objectives, routage, simulation à événements discrets.

## Abstract

With the Internet of Things (IoT), objects can be interconnected and remotely monitored over an existing network infrastructure. Each connected object is uniquely identifiable by an embedded system. IoT has applications in several areas such as Smart Cities, Smart Grids, Pollution and Environmental Monitoring, Home Automation, etc. Most nodes in an IoT network have reduced capabilities in terms of memory, computing power, energy, and so on. The goal of load balancing is to balance the traffic between the different nodes of the network so as to improve total service life and improve network throughput. To do this, we can take into account several parameters such as the occupancy rate of the buffers in the nodes, the storage capacity, the available bandwidth, the load of a node, etc. The objective of this work is:

- Studying the different objective functions proposed for load balancing
- Propose a new objective function for load balancing in the Internet of Things;
- Implement the new routing protocol on a simulator;
- Evaluate the performance of the new routing protocol;

**Keywords** : Internet of Things (IoT), load balancing, objective functions, routing, discrete event simulation.

## Remerciements

Nous tenons tout d'abord à remercier Dieu le tout puissant, qui nous a donné la force et la patience d'accomplir ce travail. En second lieu, nous tenons à remercier notre encadreur: M. BOUCHAMA Nadir, et notre promoteur: M. OULDKHAOUA Mohamed, de leurs précieux conseils et leurs aides durant toute la période du travail. Nos vifs remerciements vont également aux membres du jury pour l'intérêt qu'ils ont porté à notre recherche en acceptant d'examiner notre travail Et de l'enrichir par leurs propositions.

Nos reconnaissances et gratitude à l'administration et à l'ensemble du corps enseignant de l'Université Saad Dahlab de Blida pour leurs efforts à nous garantir la continuité et labouissement de ce programme de Master.

Enfin, nous tenons également à remercier toutes les personnes qui ont participé de près ou de loin à la réalisation de ce travail.

# Table des matières

<b>Introduction générale</b> . . . . .	12
Contexte général . . . . .	12
<b>1 Etat de l'art sur l'Internet des Objets</b>	<b>14</b>
1.1 La 3ème vague de l'internet : Internet des objets . . . . .	14
1.2 Cycle de vie d'un objet connecté dans l'internet des objets . . . . .	15
1.3 Technologies fondatrices de l'IoT . . . . .	15
1.3.1 Les réseaux de capteurs sans fil . . . . .	17
1.4 Architecture de l'Internet des objets . . . . .	18
1.4.1 La couche perception . . . . .	18
1.4.2 La couche réseau . . . . .	18
1.4.3 La couche application . . . . .	19
1.5 Fonctionnement d'IOT . . . . .	19
1.5.1 Collecter /Actionner . . . . .	19
1.5.2 Communiquer : . . . . .	20
1.5.3 Exécuter : . . . . .	20
1.5.4 Visualiser : . . . . .	20
1.6 Pile protocolaire d'IoT . . . . .	20
1.7 Les applications de l'Internet des objets . . . . .	21
1.8 Les enjeux de l'Internet des objets . . . . .	22
1.8.1 La sécurité : . . . . .	22
1.8.2 L'énergie du capteur : . . . . .	22
1.8.3 Les limitations de ressources : . . . . .	23
1.8.4 L'interopérabilité : . . . . .	23
1.8.5 La virtualisation : . . . . .	23
1.8.6 Le nombre croissant d'objets connectés : . . . . .	23
1.8.7 La mobilité : . . . . .	23
1.8.8 La qualité de service des communications : . . . . .	24
<b>2 Routage et Equilibrage de charge dans l'Internet des objets</b>	<b>25</b>
2.1 Le principe de routage . . . . .	25
2.1.1 Classification des protocoles de routage . . . . .	25
2.1.1.1 Selon le moment d'établissement des routes . . . . .	26
2.1.1.2 Selon la participation des nœuds dans le routage . . . . .	26
2.2 Protocole de routage . . . . .	27
2.2.1 LOAD . . . . .	27
2.2.2 RPL . . . . .	28
2.2.2.1 Les messages de contrôle RPL . . . . .	28
2.2.2.2 La Construction DODAG . . . . .	29

2.3	Problème de l'équilibrage de charge :	30
2.4	Solution existant pour l'équilibrage de charge :	32
2.4.1	LB-RPL	32
2.4.2	CO-RPL	33
2.4.3	ELT	34
2.4.4	QU-RPL	35
2.4.5	ALABAMO	36
2.4.6	Load balancing in OF :LB-OF	36
2.4.7	Power Control-RPL	37
2.4.8	ECRM	37
2.5	comparaison entre différentes solutions	38
<b>3</b>	<b>Proposition d'une nouvelle fonction objective pour équilibrage de charge</b>	<b>42</b>
3.1	Motivation	42
3.2	Diagramme de classe pour le protocole RPL	43
3.3	Fonction objective	45
3.4	Principale métrique proposée	46
3.4.1	Energie consommé :	46
3.4.2	Nombre de voisin	47
3.5	La fonction objective proposé	47
3.5.1	message DIO modifié	51
<b>4</b>	<b>Évaluation des performances</b>	<b>52</b>
4.1	Système d'exploitation :	52
4.1.1	Contiki OS	52
4.1.2	COOJA :	53
4.2	Implémentation LbCCRPL :	54
4.3	Métriques de simulation	56
4.4	Evolution	56
4.5	RÉSULTATS ET DISCUSSION	58
<b>5</b>	<b>Annexes</b>	<b>62</b>
	<b>Introduction générale</b>	<b>62</b>
5.1	Installation de Contiki	62
5.2	Le simulateur Cooja	62
5.3	Démarrage de Cooja	62
5.4	Création d'une simulation	63
5.5	L'interface de simulation	64
5.6	Définition des types de nœuds	65
5.7	Ajout de nœuds (Motes)	65
5.8	Conclusion	66
	<b>Conclusion générale</b>	<b>67</b>

5.4	L'interface créer le type de nœud . . . . .	65
5.5	a. . . . .	66
5.6	b. . . . .	66
5.7	La fenetre du réseau. . . . .	66

# Liste des tableaux

2.1	Table de comparaison entre les solutions existant pour l'équilibrage de charge . . . . .	39
4.1	Paramètres de simulation . . . . .	57

# Abréviations

**6LowWPAN** . IPv6 over Low power Wireless Personal Area Networks

**AODV** . . . . . Ad-hoc On demand Distance Vector

**ALABAMO** . A LoAd BALancing MOdel

**A-LLN** . . . . . Agricultural-Low power and Lossy Networks

**API** . . . . . Application Programming Interface

**CoAP** . . . . . Constrained Application Protocol

**CAOF** . . . . . Context-Aware Objective Function

**CNC** . . . . .Count Nodes Child

**Cyber-OF** . . . . Cyber-Physical objective function

**DIS** . . . . . DODAG Information Solicitation

**DIO** . . . . . DODAG Information Object

**DAO** . . . . . Destination Advertisement Object

**DAO-ACK** . . . Destination Advertisement Object Acknowledgement

**DAG** . . . . . Directed Acyclic Graph

**D2D** . . . . . Device to Device

**DSR** . . . . . Dynamic Source Routing

**DTSN** . . . . . Destination Advertisement Trigger Sequence Number

**DODAG** . . . . . Destination Oriented Directed Acyclic Graph

**EN\_CON** . . . . . Energie Consommé

**ETX** . . . . . Expected Number of Transmissions

**ELT** . . . . . Expected LifeTime

**FMOF** . . . . . Fuzzy Mobility Objective Function

**HC** . . . . . Hop Count

**IP** . . . . . Internet Protocol

**IETF** . . . . . International Engineering Task Force

**IS-IS** . . . . . Intermediate System to Intermediate System

**IoT** . . . . . Internet of Things

**IPv6** . . . . . Internet Protocol version 6

**IEEE** . . . . . Institute of Electrical and Electronics Engineers

**ICMPv6** . . . . . Internet Control Message Protocol version 6

**LLN** . . . . . Low power and Lossy Networks

**LBCC\_RPL** . . Load Balancing Count Child Objective Function

**LBCC\_RPL** . . Load Balancing Count Child RPL

**LR-WPAN** . . . Low Rate-Wireless Personal Area Network

**LQL** . . . . . Link Quality Level

**MP2P** . . . . . Multipoint-to-Point

**MOP** . . . . . Mode of Operation

**MAC** . . . . . Medium Access Control

**MRHOF** . . . . . Minimum Rank with Hysteresis Objective Function

**ND** . . . . . Neighbor Discovery

**OCP** . . . . . Objective Code Point

**OF0** . . . . . Objective Function0

**OF-FL** . . . . . Objective Function based on Fuzzy Logic

**Prf** . . . . . DODAGPreference

**P2MP** . . . . . Point-to-Multipoint

**P2P** . . . . . Point-to-point

**QoS** . . . . . Qualite of Service

**RFID** . . . . . Radio Frequency Identification

**RPL** . . . . . Routing Protocol for Low Power and Lossy Networks

**RREQ** . . . . . Route Request

**RREP** . . . . . Route Reply

**RERR** . . . . . Route Error

**ROLL-WG** . . . . . ROLL Working Group

**RPAL** . . . . . Agricultural Low-Power and Lossy Networks

**RE** . . . . . nergie restante

**RX** . . . . . Receive

**SCAOF** . . . . . Scalable Context-Aware Objective Function

**TCP/IP** . . . . . Transmission Control Protocol/Internet Protocol

---

**TBRPF** . . . . . Topology dissemination Based on Reverse Path Forwarding

**TX** . . . . . Transmit

**UDGM** . . . . . Unit Disk Graph Model

**UDP** . . . . . User Datagram Protocol

**WSN** . . . . . Wireless Sensor Networks

**WPAN** . . . . . Wireless Personal Area Network

**ZRP** . . . . . Zone based hierarchical link stateRouting Protocol

# Introduction générale

## Contexte général

Les réseaux à faible consommation d'énergie et à perte (LLN) sont composés de plusieurs systèmes intégrés de détection qui communiquent et permettent des différentes applications, telles que la construction intelligente, la surveillance de l'habitat et automatisation industrielle . Ces réseaux font partie de l'Internet des objets (IoT), qui sera le principal moteur de la croissance de la productivité . L'internet des objets (IoT) devrait compter des dizaines de milliards d'appareils d'ici la fin de la décennie. Dans la plupart des applications basées sur le réseau LLN, les nœuds de capteurs ont des ressources limitées : mémoire, puissance de traitement et énergie . Compte tenu de ces capacités limitées, les protocoles de routage doivent être bien conçus, car ils ont un impact sur l'utilisation des ressources du nœud et sur la consommation d'énergie . En plus de fournir une livraison fiable et efficace, les algorithmes de routage doivent prendre en compte l'efficacité énergétique afin de prolonger la durée de vie du réseau .

le groupe de travail IETF ROLL a publié un protocole de routage pour les réseaux à faible consommation et à perte (RPL) . Le RPL est devenu le protocole de routage normalisé conforme à IPv6 pour les LLN . Il s'agit d'un protocole de vecteur de distance qui établit des graphes acycliques dirigés vers une destination (DODAG) en fonction de métriques de liens et de nœuds . Un DODAG est une topologie en arborescence qui prend en charge le trafic en aval et en amont et est construit selon une fonction objectif (OF).

Différentes fonctions objectives peuvent être conçues afin de répondre à des critères d'optimisation spécifiques et de satisfaire aux exigences d'une application particulière. Complétant la spécification RPL, le groupe de travail ROLL a également normalisé la fonction objective de classement minimum avec hystérésis (MHROF) . MHROF est conçu pour être étendu avec différentes mesures permettant de calculer le coût d'un chemin. Une métrique par défaut définie par le RFC 6719[], utilisée dans la plupart des implémentations RPL, utilise le nombre de liaisons prévu (ETX) pour calculer le chemin avec un coût minimal . Avec les DODAG créés à l'aide de MHROF avec la métrique ETX, les nœuds ont tendance à sélectionner des parents avec la meilleure qualité de liaison et le plus petit nombre de sauts au nœud racine . Même si la sélection du parent avec la meilleure qualité de lien est un choix optimal local, cela pourrait causer un problème d'équilibrage de charge . Dans des scénarios où la distribution des nœuds et / ou la structure du trafic étant hétérogène, le déséquilibre de la charge de travail devient critique . Les nœuds qui sont plus proches de la racine et / ou qui ont de nombreux enfants avec des liens de bonne qualité recevront et achemineront un grand nombre de paquets qui pourraient rapidement épuiser la durée de vie de leur batterie . Pour résoudre les problèmes ci-dessus, nous proposons une nouvelle fonction objective basée sur MHROF . La nouvelle OF utilise à la fois la consommation d'énergie et le nombre de voisins des nœuds et ETX des liens afin de résoudre le problème de déséquilibre.

---

La proposition s'appelle LBCCOF( Load Balancing Count Children).  
ce mémoire est organisé en 4 chapitres comme suit :

**Chapitre1** : Analyse sous forme d'un état de l'art sur l'internet des objets et ses principales caractéristiques ainsi ses différentes concepts .

**Chapitre2** : Introduit les problématiques liées a l'équilibrage de charge , les travaux annexes des solutions, il décrit également le protocole RPL .

**Chapitre3** : Est consacré à la conception de la nouvelle fonction objective .

**Chapitre4** : Présente l'évaluation de notre solution et discussion des résultats , ainsi les détails de l'implémentations .

# Chapitre 1

## Etat de l'art sur l'Internet des Objets

### Introduction

L'Internet des Objets est une technologie en croissance rapide avec des opportunités et des risques commerciaux . C'est la confluence de réseaux sans fil, internet et informatique . IoT relie les objets physiques comme les véhicules, les batiments et autres dispositifs avec des capteurs intelligents intégrés et permet à ces objets d'échanger et de collecter des données . Les domaines de IoT est les villes intelligentes, la santé en ligne, les réseaux intelligents, le commerce électronique, les transports intelligents, etc...

L'architecture de l'IoT est incorporée aux dernières technologies des protocoles de communication, réseaux de capteur et RFID. Les problèmes de sécurité et de confidentialité de l'IoT sont cruciaux car ils connectent les périphériques de grand nombre .

Dans ce chapitre nous analysons un état de l'art de l'IoT en ce qui concerne les technologies, les protocoles, l'architecture et les domaines d'application.

### 1.1 La 3éme vague de l'internet : Internet des objets

L'Internet of Things (IoT) est un réseau qui relie et combine les objets avec l'Internet, en suivant les protocoles qui assurent leurs communications et échange d'informations à travers une variété de dispositifs [21], Il peut être défini comme une évolution et extension de l'Internet de nos jours pour l'inclusion de tous les objets et les endroits dans notre entourage (réfrigérateurs, thermostat, maisons, véhicules, routes, etc...). Le concept promoteur de l'IoT va nous simplifier la vie, nous faire gagner du temps, décharger notre cerveau de la mémorisation de données logistiques (itinéraires, temps de prise des médicaments, etc.). Ainsi, l'accès ubiquitaire a différents types d'informations permettrait la sophistication du mode de vie et une amélioration significative de la qualité des services dans différents domaines .

En 1999, la désignation Internet des objets a été prononcée pour la toute première fois par Kevin Ashton, dans le cadre de la gestion de la chaine d'approvisionnement [2]. après, en 2000 la société LG annonce son premier réfrigérateur intelligent connecté a Internet . De plus, la technologie RFID (Radio Frequency IDentification) qui est l'une des technologies constitutionnelles de l'IoT, a commencé à être massivement déployée vers les années 2003 et 2004 . D'autre part, une initiative très intéressante a été prise en 2008 : un groupe de recherche appelé IPSo Alliance s'est consacré à promouvoir l'utilisation du protocole IP

---

(Internet Protocol) pour les réseaux d'objets miniatures intelligents [3] . Cependant, au cours de la dernière décennie, la défi

nition a été plus inclusive couvrant une large gamme d'applications comme les transports, les soins de santé, les services publics, etc...

## 1.2 Cycle de vie d'un objet connecté dans l'internet des objets

Dans l'IoT, les objets intelligents passent par trois étapes : la phase préparatoire (bootstrapping), la phase opérationnelle et la phase de maintenance maReference4.

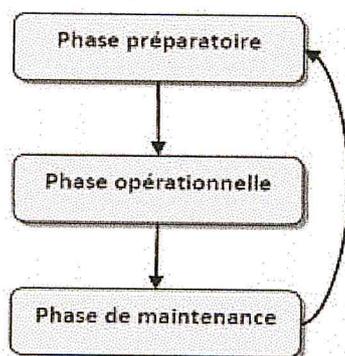


FIGURE 1.1 – Cycle de vie d'un objet

- La phase préparatoire (bootstrapping) : déploiement des objets (capteurs, tags), leur configuration avec les informations nécessaires, par exemple les identificateurs, les clés de sécurité, etc...
- La phase opérationnelle : dans la phase opérationnelle, l'objet connecté se met à réaliser sa mission qui diffère d'une application à une autre .
- La phase de maintenance : effectue des mises à jours, régle les problèmes en faisant d'éventuelles réparations des objets en cas de défaillance par exemple, Il est meme possible de remplacer carrément des objets et redémarrer à nouveau à partir de la phase préparatoire.

## 1.3 Technologies fondatrices de l'IoT

Bien que l'Internet des objets soit une notion relativement nouvelle, les technologies qui la rendent possible existaient depuis quelques années déjà. On parle alors des réseaux de capteurs sans fil et de la technologie d'identification par radio fréquence . Les évolutions observées par les technologies sans fil et le domaine des réseaux de télécommunication d'une part, et l'Internet de l'autre part, ont permis d'ouvrir de nouvelles perspectives pour ces technologies, qui ont pu s'instaurer efficacement dans notre vie quotidienne et qui sont devenues de plus en plus omniprésentes . Ainsi, de nouvelles facilités et de nouveaux modes d'exploitation des services peuvent être envisagés si les capteurs et les marqueurs d'identification intègrent l'Internet . En effet, bien qu'il existe plusieurs technologies utilisées dans le fonctionnement de l'IoT, nous mettons l'accent seulement sur quelques-unes qui sont, selon Han et Zhanghang, les technologies clés de l'IoT . Ces

technologies sont les suivantes : RFID Un système RFID[5] est composé d'un ou plusieurs lecteurs et d'un ensemble d'étiquettes (appelée aussi tags, marqueurs, identifiants ou transpondeurs) a micro-puissances . Les étiquettes sont des dispositifs minuscules équipés d'une puce contenant des informations et une antenne pour la communication radio . Elles sont placées sur les éléments que l'on veut identifier d'une manière unique ou tracer . Les étiquettes peuvent avoir différentes formes et peuvent être passives ou actives : Les étiquettes actives sont équipées d'une batterie, elles diffusent des signaux automatiquement et d'une façon autonome, tandis que les étiquettes passives ne disposent d'aucune source d'énergie et attendent a ce qu'un signal électromagnétique leur arrive et munit de l'énergie pour pouvoir envoyer leurs propres signaux . Les étiquettes passives sont plus déployées que celles qui sont actives car leur usage est beaucoup plus flexible avec un coût nettement réduit . Une autre spécificité pas moins importante dans les étiquettes passives qui est la durée de vie . Par le fait d'être passive, la durée de vie de l'étiquette est importante (elle reste valable tant qu'elle garde son bon état), ce qui n'est pas le cas pour une étiquette active ou la durée de vie est restreinte (s'achève avec l'épuisement de la batterie).

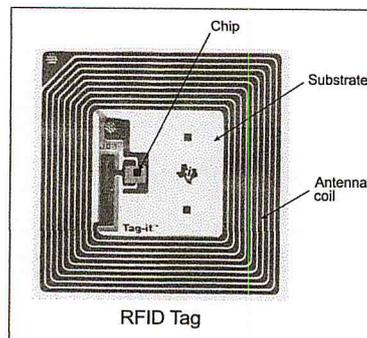


FIGURE 1.2 – étiquette passive

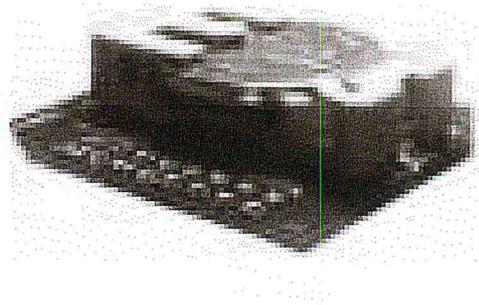


FIGURE 1.3 – étiquette active

Le processus d'identification se réalise à travers un scénario bien déterminé . En effet, le lecteur active les étiquettes qui passent devant lui en leur envoyant un signal électromagnétique puissant . Les étiquettes s'activent et réagissent en répondant par un signal transportant les identités . Contrairement aux systèmes d'identification par codes barre qui exigent que le lecteur et le code barre soient exactement opposés et très proches l'un de l'autre, dans un système RFID, il suffit juste que le lecteur et l'étiquette soient l'un dans la portée de communication de l'autre pour que l'interaction puisse avoir lieu . La portée de communication radio dans un système RFID dépend du type de tag (passif ou

actif) et de la gamme de fréquences utilisées . Par exemple, la portée avec les étiquettes actives est plus importante qu'avec celles qui sont passives . Dans le contexte de l'Internet des objets, les objets intelligents ont besoin d'être identifiés de façon unique, à partir de l'adoption de la technologie RFID s'est avérée nécessaire.

### 1.3.1 Les réseaux de capteurs sans fil

Les RCSF se composent généralement d'un grand nombre de nœuds capteurs minuscules, stationnaires ou mobiles, souvent déployés aléatoirement dans un champ de captage . Ce dernier est généralement un milieu hostile, isolé ou difficile à contrôler ou la mission d'un nœud capteur consiste à chaque fois, de récolter, d'une façon autonome, des informations précises depuis l'environnement de déploiement . Suivant le type du nœud capteur, la donnée captée peut être la température, l'humidité, la pression, la lumière ou autres . les nœuds capteurs dans un RCSF communiquent entre eux via des liens radio pour l'acheminement des données collectées à un nœud considéré comme point de collecte, appelé station de base ou puit . Cette dernière peut être connectée à une machine puissante, appelée gestionnaire des tâches, via Internet ou par satellite . En outre, le réseau peut être configuré de telle sorte que l'utilisateur puisse adresser ses requêtes aux capteurs en précisant l'information requise, et en ciblant les nœuds capteurs qui devraient s'y intéresser.

Les RCSF jouent un rôle très intéressant dans l'Internet des objets . En effet, les cap-

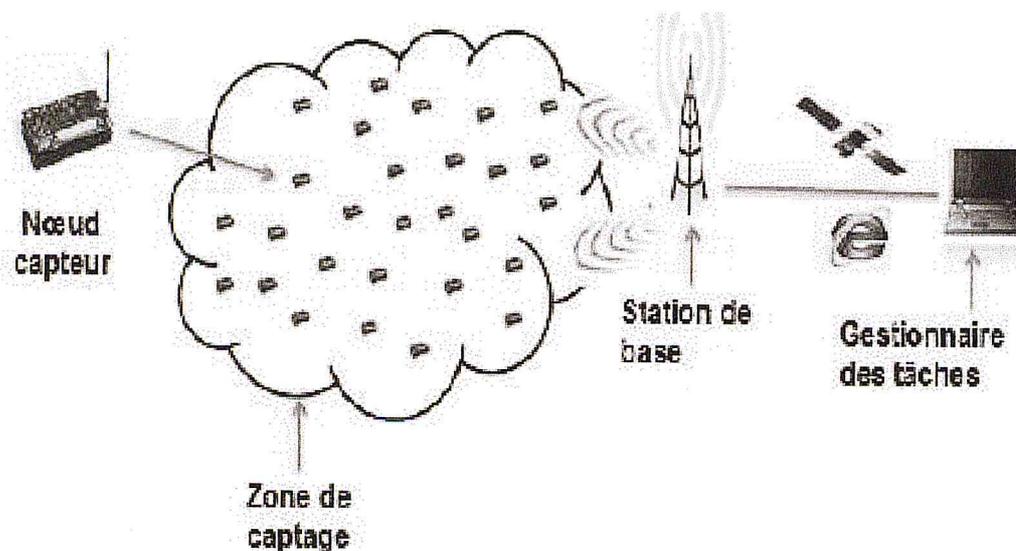


FIGURE 1.4 – Architecture de communication d'un réseau de capteur sans fi

1

teurs permettent la représentation des caractéristiques dynamiques (température, humidité, pression, mouvements, . . .) des objets et des endroits du monde réel dans le monde virtuel représenté par le réseau Internet global . Ainsi, avec l'incorporation des réseaux de capteurs dans l'Internet, Les capteurs deviennent des serveurs (fournisseurs de services) dans ce que l'on désigne par le web des objets (dit WoT pour Web of Things)maReference6.

Ainsi, les services (applications) des RCSF se rajoutent à l'ensemble des services et applications de l'Internet de futur qui réunira une variété de réseaux fortement hétérogènes (que ça soit sur le plan matériel ou logiciel), soumis à des contraintes différentes et qui sont déployés pour diverses applications, afin d'en avoir un monde réel très sophistiqué.

## 1.4 Architecture de l'Internet des objets

De point de vue architectural, on peut dire que l'Internet des objets est organisée en trois couches principales <sup>maReference7</sup> : la couche de perception de données, la couche réseau et troisièmement la couche application . La figure ci-dessous illustre telle organisation.

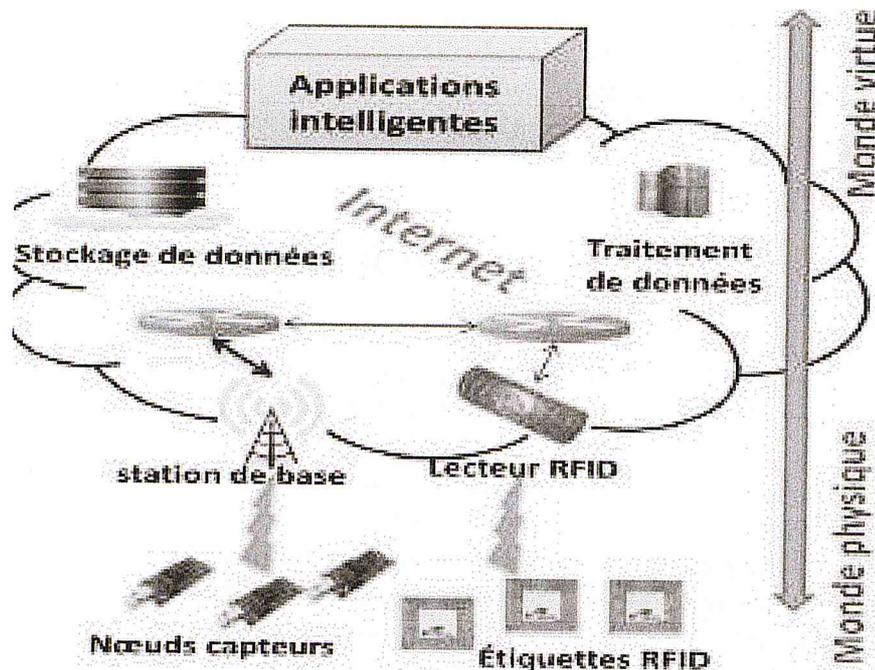


FIGURE 1.5 – Architecture de l'internet des objet

### 1.4.1 La couche perception

La couche perception, au niveau bas dans la hiérarchie, est responsable de la capture de données, ainsi que leur identification dans leur environnement . Cette couche comprend ainsi le matériel nécessaire pour parvenir à la collection de données contextuelles des objets connectés, à savoir les capteurs, les étiquettes RFID, caméras, GPS (Global Positioning System), etc...

### 1.4.2 La couche réseau

Cette couche se charge de la transmission fiable des données générées dans la couche perception ainsi que l'assurance de la connectivité inter-objets connectés et entre objets intelligents et les autres hotes de l'Internet . D'autre part, il est prévu que les données issues de la couche perception soient énormes car le nombre d'objets connectés à Internet

---

### 1.5.2 Communiquer :

C'est l'étape qui permet l'envoi des données depuis le réseau local vers le cloud. On parle essentiellement des protocoles pour transporter la donnée et on peut en distinguer deux modèles : Le modèle Publish / Subscribe avec des protocoles de type MQTT et le modèle REST avec des protocoles comme HTTP ou encore CoAP.

### 1.5.3 Exécuter :

C'est l'étape de stockage et de traitement de la donnée. A cette étape on parle souvent de " Plate-forme IoT " qui est souvent une solution cloud capable de connecter plusieurs objets connectés, stocker leurs données, les traiter, les analyser et les exposer a travers différentes applications . Les plateformes IoT permettent aussi de faire communiquer des objets hétérogènes. Ces plateformes se multiplient de nos jours (Amazon, Google, Microsoft, etc...) et on parle même de guerre des plateformes IoT.

### 1.5.4 Visualiser :

C'est l'étape qui permet d'exposer les services des objets connectés à travers différentes applications dédiées . Un utilisateur, à travers une application mobile, peut par exemple communiquer avec ses objets en consultant leurs données ou en envoyant des actions vers ses objets.

## 1.6 Pile protocolaire d'IoT

Le protocole est un ensemble de règles et de règlements utilisés pour communication dans des réseaux identiques ou différents. les protocoles tels que HTTP et TCP / IP ne peuvent pas être utilisés pour l'IoT en raison de l'existence de dispositifs intelligents et d'autres contraintes.

Pour la communication machine à machine, les protocoles MQTT (Transport de télémétrie de file d'attente de messages) et CoAP (Constraint Application Protocol) sont utilisés [maReference8](#). MQTT protocole inclut les fonctionnalités de publication/abonnement modèle de message, le transport de messagerie .CoAP est un web protocole de transfert pour les nœuds contraints et contraint réseaux.

une pile de protocole avec les protocoles existants peuvent être conçus comme le montre la Fig.4[12]. la partie à droite montre la pile de protocoles pour l'environnement traditionnel, et la partie à gauche montre la pile de protocoles pour les contraintes environnement . Le protocole CoAP (The Constrained Application Protocol) à été conçu par l'IETF pour une utilisation avec des réseaux à faible puissance et contraintes, et il est un bon choix pour les appareils fonctionnant sur des batteries ou la récolte d'énergie .

Le protocole RPL [maReference9](#) est un protocole de routage IPv6 destiné aux réseaux 6LoWPAN dans l'Internet des objets. il forme une topologie dynamique et optimisé avec l'évitement des boucles et la considération des paramètres de qualité de service pour l'acheminement des datagrammes IPv6 depuis et vers les noeuds capteurs . Avec UDP (User Datagram Protocol), les applications peuvent envoyer des messages vers d'autres hotes sur un réseau IP sans communication préalable pour con

naitre des canaux de transmission spéciaux ou des chemins de données . 6LowPAN est un protocole réseau qui dé

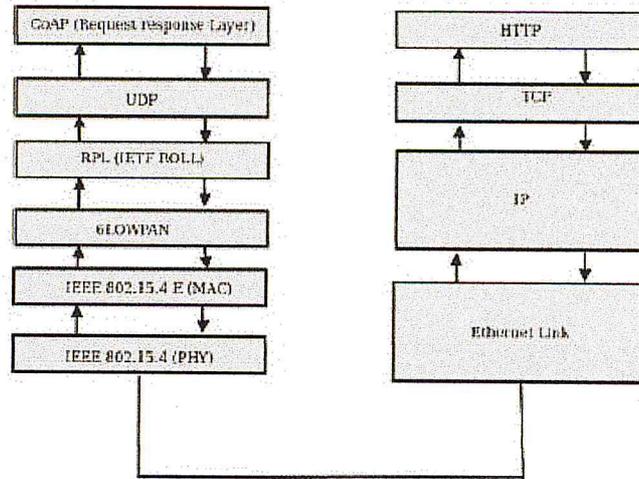


FIGURE 1.6 – pile protocolaire

nit des mécanismes d'encapsulation et de compression d'en-tête qui permettent d'envoyer et de recevoir des paquets IPv6 sur des réseaux locaux sans fil (LR-WPAN) à faible débit. IEEE 802.15.4 E MAC (The Medium Access Control) permet la transmission de trames MAC à l'aide de la couche physique [8], elle offre aussi une interface de gestion et gère lui-même l'accès à la couche physique. Elle contrôle également la validation des trames IEEE 802.15.4 PHY. La couche physique fournit le service de transmission de données, ainsi que l'interface avec l'entité de gestion de couche physique, qui offre l'accès à chaque fonction de gestion de couche et maintient une base de données d'informations sur des réseaux personnels associés.

Comme IoT se compose d'énormes nombres d'appareils, la gestion du réseau devient difficile. Pour activer la bonne gestion du réseau, les protocoles comme LNMP, SNMP sont utilisés [9]. LNMP est réseaux de gestion de réseau PAN 6LoWPAN, SNMP est Simple Network Management Protocol est un protocole utilisé pour contrôler et gérer les périphériques réseau IP. SNMP peut être utilisé pour divers périphériques tels que les routeurs, commutateurs, postes de travail, serveurs, etc...

## 1.7 Les applications de l'Internet des objets

L'Internet des objets ce n'est qu'un immense ensemble d'objets intelligents interconnectés et connectés à Internet mais c'est également et plus considérablement, les applications qui sont en fait la raison d'être de cette nouvelle vague de connectivité sur internet. L'existence des objets intelligents avec de nouvelles possibilités de communications automatiques et intelligentes vont sensiblement améliorer le mode de vie des gens ainsi que la qualité de services dans divers domaines à travers des degrés élevés d'autonomie et d'intelligence.

Nous pouvons affirmer que l'Internet peut être connecté à n'importe quel objet. Ainsi, les domaines d'applications de l'IoT sont multiples. On cite, à titre d'exemples, l'industrie, la santé, l'éducation et la recherche. Cependant, il sera possible dans le futur de trouver le concept de l'IoT n'importe où, n'importe quand et à la disposition de tout le monde.

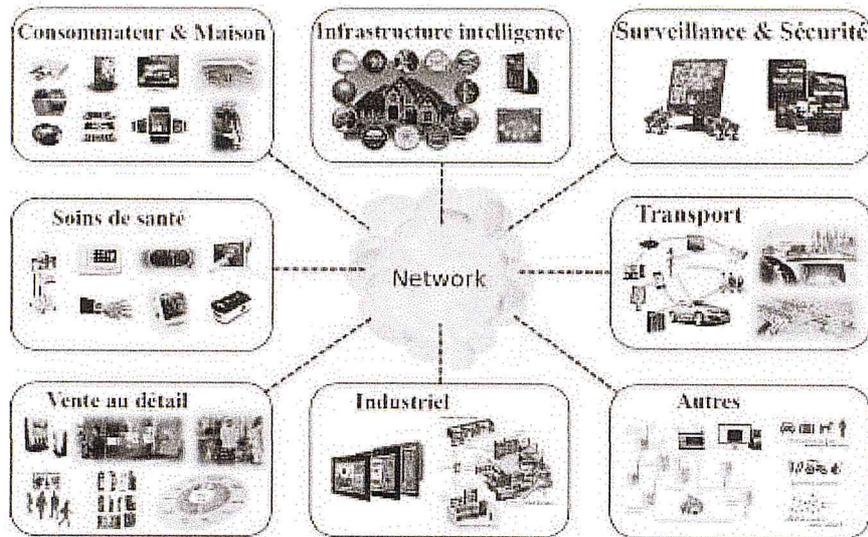


FIGURE 1.7 – Application d'IOT

## 1.8 Les enjeux de l'Internet des objets

Bien que l'Internet des objets soit un concept qui est à la fois avantageux et prometteur, et qui pourra apporter des solutions efficaces des problèmes du suivi et de télésurveillance dans différents domaines. En contrepartie, l'IoT soulève certaines questions décisives, étroitement liées à sa maturité et son acceptabilité. On cite ci-dessous les enjeux les plus marquants :

### 1.8.1 La sécurité :

la sécurité des personnes, des communications, des données, des services, des réseaux et des équipements était et continue à être un problème sévère observé par l'internet courant. aujourd'hui avec la naissance de l'IoT, l'amplitude du problème va prendre un autre ordre de gravité. Des milliers d'objets contraints connectés en permanence à internet et intégrés dans toute sorte d'objets dans notre vie quotidienne, vont porter le risque d'être ciblés par les menaces classiques de l'Internet. Il est même possible que de nouvelles générations d'attaques apparaissent. Donc, les objets intelligents dans l'IoT, la transmission et le stockage de leurs données sur Internet devraient être sécurisés. D'autre part, l'IoT peut lui-même menacer la sécurité des individus ou des institutions. L'armée chinoise proscrit les officiers et les soldats de porter des objets connectés (comme les montres et les lunettes connectées à Internet) et considère leur utilisation comme une violation de la réglementation sur le secret dans les casernes [10].

### 1.8.2 L'énergie du capteur :

Pour IoT, les choses seront actives participants dans tous les aspects comme les affaires, l'information et autres processus. Pour activer l'IoT pour les environnements intelligents plus d'attention doit être donnée sur les exigences des capteurs. Ceux-ci peuvent être mis en œuvre à travers des réseaux de capteurs à grande échelle. Donner une alimentation continue et fiable aux nœuds de capteurs est une préoccupation majeure. Pour déployer l'IoT

---

avec succès [11], la puissance besoin d'être fourni pour une période de temps prolongée . Divers technologies comme les cellules solaires, les générateurs thermiques, et la rectification des signaux radio doivent être déployés pour connecter des capteurs situés dans des endroits éloignés et distants .

### **1.8.3 Les limitations de ressources :**

les capteurs et les tags RFID sont très limités en ressources de calculs, de stockage mémoire et d'énergie . A cet effet, les solutions (protocoles de communications ou de sécurité, technologies de transmission, etc...) destinées à l'Internet des objets doivent prendre en considération telles contraintes et limitations .

### **1.8.4 L'interopérabilité :**

c'est parmi les plus grands manque de la réalisation de l'Internet des objets . L'interopérabilité c'est en réalité, la cohabitation des dispositifs, des systèmes et des mécanismes disjoints et la possibilité de les faire coopérer et interagir en toute exhibilité . Une tendance récente tend vers la standardisation et l'uni

cation des systèmes et protocoles opérationnels dans l'IoT et de les présenter en open source . Ceci afin de faciliter la collaboration entre objets connectés, ainsi que le couplage avec les entités externes se trouvant sur Internet.

### **1.8.5 La virtualisation :**

plusieurs capteurs connectés peuvent représenter un seul capteur virtuel qui rapporte une mesure virtuelle résultant de l'agrégation de plusieurs états secondaires . Par exemple un capteur virtuel qui nous dit si l'état de santé du patient est bon ou non . Cette information n'est qu'une combinaison de plusieurs informations fournies par plusieurs capteurs médicaux réels incorporés dans le corps du patient . Ainsi, un modèle générique de virtualisation des objets connectés a l'IoT, nommé VoT (Virtualization of Things) [12] permet une représentation abstraite des objets et l'accumulation des données qui en proviennent, depuis différents endroits, pour faciliter leur contrôle .

### **1.8.6 Le nombre croissant d'objets connectés :**

Il est prévu que le nombre d'objets intelligents qui vont peupler l'Internet du futur franchira les millions, voir les milliards. Avec cela, l'adoption de nouveaux mécanismes qui supportent efficacement l'évolutivité continue dans le nombre d'objets connectés, est vivement recommandée [13].

### **1.8.7 La mobilité :**

un nombre immense d'objets connectés à Internet en tant que partie de l'Internet des objets, seront le plus souvent mobiles . De ce fait, des solutions exibles de gestion de la

---

mobilité doivent être mises en place pour permettre à tels objets d'accomplir leurs missions efficacement indépendamment de la fréquence et la vitesse de la mobilité.

### 1.8.8 La qualité de service des communications :

[14] suivant l'application qu'elle soit critique ou non, les communications inter objets connectés dans l'IoT et entre ces derniers et les hotes ordinaires de l'internet, peuvent exiger ou non un minimum de qualité de service en termes de delais, debits, fiabilité, etc...

## Conclusion

Dans son ensemble, l'IoT change la façon dont nous vivons . Le déploiement d'IoT est en cours. Comme innovant et créatif produits seront conçus et présentés chaque jour, de nombreux des défis comme la collecte d'énergie, les problèmes d'alimentation, le réveil retards, et l'identification de la technologie viendra a l'image . Ces défis doivent être affrontés et résolus soigneusement .

Dans ce chapitre, nous avons analysé les dernières tendances dans l'architecture IoT, protocoles de communication, problèmes de sécurité, applications qui ont beaucoup de portée pour accueillir le développement futur.

Dans le chapitre qui suit nous entamons le routage dans l'Iot, les problèmes d'équilibrage de charge ainsi que ses solutions .

# Chapitre 2

## Routage et Equilibrage de charge dans l'Internet des objets

### Introduction

Le routage est le mécanisme par lequel des chemins sont sélectionnés dans un réseau pour acheminer les données d'un expéditeur jusqu'à un ou plusieurs destinataires. Dans les réseaux de faible puissance et de perte, le routage qui existe dans internet n'est pas adapté à ce réseau .Les nœuds fonctionnent comme routeur pour envoyer un paquet sinon elles fonctionnent comme station pour émission ou réception d'un paquet. Dans ce chapitre, nous allons aborder les routage et l'équilibrage de charge dans l'IoT, pour cela nous présentons les principes de routage, ainsi que les problèmes des routage dans IoT, les problèmes d'équilibrage de charge et les solutions qui existent pour optimiser équilibrage de charge dans IoT, qui est l'objectif de notre travail.

### 2.1 Le principe de routage

[4] Le routage peut être défini comme étant l'ensemble des règles appliquées au format et à la signification des trames, paquets, ou messages échangés entre les nœuds. l'objectif principal d'un protocole de routage et de trouver et maintenir le chemin optimal multi saut pour une communication quelconque, le sens du mot optimal dépend de la nature du réseau, la nature des utilisateurs, on distingue plusieurs points de vue :

- chemin optimal qui consomme le minimum de ressources (bande ,mémoire, processeur,énergie, etc... ).
  - chemin optimal qui est le plus court chemin (distance).
  - chemin optimal qui est le plus sûr.
  - chemin qui assure la meilleure qualité de service (QoS).
- cet objectif peut être atteint par les points suivants :
- Éviter les boucles de routage.
  - Réduire le nombre et la taille des messages de contrôle.
  - Réduire et simplifier les trames .

#### 2.1.1 Classification des protocoles de routage

Il existe plusieurs critères pour classifier les protocoles de routage

---

### 2.1.1.1 Selon le moment d'établissement des routes

#### 1. protocoles de routage proactifs :

Le principe de cette classe est que les routes sont préparées à l'avance, les protocoles de routage proactifs utilisent une ou plusieurs tables de routage dans chaque nœud, ils essaient de maintenir les meilleurs chemins existants vers toutes les destinations possibles au niveau de chaque nœud au réseau, pour ce faire, ils utilisent l'échange régulier des messages de contrôle pour mettre à jour les tables de routage vers toute destination atteignable depuis celui-ci . Cette approche permet de disposer d'une route vers chaque destination immédiatement au moment où un paquet doit être envoyé . les tables des routages sont modifiées à chaque changement de la topologie du réseau . Cette approche offre un gain de temps lorsqu'une route est demandée. mais elle peut être coûteuse en termes de bande passante à cause de l'émission régulière des messages de contrôle surtout dans un réseaux à grand nombre de nœuds.

#### 2. Les protocoles de routage réactifs :

Le principe de cette classe est que les routes sont établies à la demande .Ces protocoles se basent sur la découverte et le maintien des routes. suite à un besoin d'une procédure de découverte globale de route est lancée . Ce processus s'arrête une fois la route trouvée ou toutes les possibilités sont examinées. Dès que la communication est établie .cette route maintenu jusqu'à ce que la destination devienne inaccessible ou jusqu'à ce que la route ne soit plus désirée. Ce type de routage minimise l'échange des messages de contrôle qui libère la bande passante . Cependant il est lent à cause de la recherche des chemin .ce qui peut dégrader les performances des applications interactives.

#### 3. protocoles de routage hybrides

Cette classe combine les deux concepts proactif et réactif ,afin de profiter de leurs avantages ,et limiter leurs inconvénient . Le principe est que chaque nœud utilise un protocole de routage proactif localement (3 sauts en général ) et un protocole de routage réactif.

### 2.1.1.2 Selon la participation des nœuds dans le routage

1. **Les protocoles uniformes** Dans cette classe de protocoles, les nœuds ont le même niveau et la même fonctionnalités en point de vue de la responsabilité de participer dans le routage . La décision qu'un nœud peut router des paquets dépend de sa position. Ainsi que chaque nœud envoie et reçoit des messages de contrôle de routage. Donc la charge est répartie de façon uniforme dans le réseau. Ce type de routage est simple et efficace dans les réseaux de petites tailles, par contre dans le cas des réseaux denses ou de grande dimension, le volume d'information de routage devenant important.
2. **Les protocoles non uniformes** Le but de ce type de protocole est de limiter la complexité du routage en réduisant le nombre de nœuds qui contribuent à la détermination des routes. Dans les nœuds n'ont pas la même responsabilité dans le routage. Cette classe de protocole est composée de deux familles :
  - Protocole sélection des voisins : Chaque nœud sélectionne un sous-ensemble de ses voisins qui prend un rôle distinct dans l'opération de routage .Chaque nœud fait sa sélection tout seul (indépendamment des autres).

- Protocole à partitionnement(hiérarchique) :les nœuds négocient un partitionnement sur la topologie, cette opération est effectuée de manière distribuée car il n'y a pas de nœud qui fait office de gestionnaire de topologie .Généralement, les noeud sont partitionnés en groupes (cludtersdont l'appartenance d'un noeud varie suivant les changements de connectivité. Quelques nœuds sont élus pour prendre un certain rôle dans le routage (leader du groupe, ou bien passerelle entre deux groupes [Gateway]).

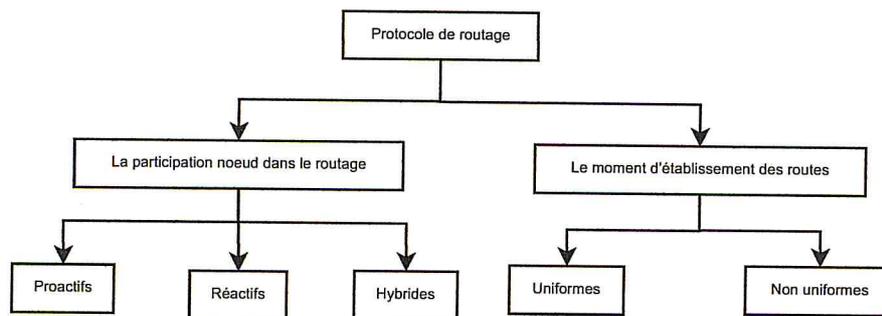


FIGURE 2.1 – Diagramme de classification des protocoles de routage

## 2.2 Protocole de routage

[12]Il existe plusieurs protocole de routage dans IoT mais aujourd'hui est utiliser les deux protocoles qu'on va voir :

### 2.2.1 LOAD

Est un service simplifié à la demande protocole de routage réactif basé sur AODV pour 6LoWPAN qui peut être utilisé sur les réseaux LLN . LOAD est défini pour fonctionner sur le dessus de la couche d'adaptation au lieu de la couche de transport. Les nœuds directement accessibles sont considérés comme des voisins . Lorsqu'un nœud doit envoyer un message à un autre nœud qui n'est pas son voisin , il diffuse un message RREQ (Route Request). Si un nœud reçoit un message RREQ et a la destination dans la table de routage, il répond par une réponse de route (RREP) sinon il transmet le RREQ. LOAD présente des simplifications et des fonctionnalités supplémentaires en ce qui concerne AODV. Une partie des extensions du protocole sont la modularité (possibilité d'ajouter des attributs arbitraires), le support d'adresses courtes et le support de diverses métriques. LOAD utilise les opérations de protocole de base d'AODV, y compris Route Discovery et Route Maintenance mais sous une forme simplifiée : pendant la découverte d'itinéraire, les messages RREQ sont inondés par le réseau et seul le nœud avec l'adresse dans le RREQ répondra avec un RREP unicast. La maintenance de la route est effectuée lorsqu'un itinéraire activement utilisé échoue. Si un paquet ne peut pas être livré, un message RERR est généré, envoyé en monodiffusion le long de la route vers la source du paquet de données.

Comparaison entre AODV et LOAD

1. LOAD est une méthode de routage des messages entre appareil mobiles. LOAD simplifie le comportement du protocole en interdisant aux périphériques intermé-

---

diaires de répondre avec une route-réponse, même s'ils ont un itinéraire actif vers la destination prévue, éliminant ainsi le besoin de numéros de séquence de destination.

2. Où dans AODV, dans le cas où un périphérique détecte une rupture de lien, ce périphérique tentera de transmettre le message d'erreur d'itinéraire à tous les voisins qui l'ont récemment utilisé comme saut suivant sur un chemin vers la destination du paquet non distribué, LOAD désactive cela, éliminant ainsi la nécessité d'un dispositif pour maintenir une liste de précurseurs. D'autres différences mineures incluent la simplification du format de paquet, la prise en charge des adresses IPv6 compressées, etc...

LOAD n'impose aucun rôle spécifique sur des périphériques spécifiques, notamment, n'a aucun contrôleur ou racine avec des responsabilités spécifiques pour le fonctionnement du réseau. Ainsi, le modèle de trafic par défaut pris en charge par LOAD est le trafic point-to-point bidirectionnel. Le seul sacrifice que fait LOAD en ce qui concerne le trafic de données, en simplifiant de AODV, est qu'il suppose qu'une destination donnée est typiquement en communication avec une seule source à un moment donné - d'où la suppression de la liste des précurseurs.

## 2.2.2 RPL

est un protocole de routage à vecteur de distance pour les LLNs, a émergé comme la solution standard de facto. RPL est une solution de routage basée sur IPv6 proposée par l'IETF (Internet Engineering Task Force) et projetée pour les réseaux LLN. La construction est basée sur le rang d'un nœud, qui représente sa distance relative à la racine, utilisant une fonction d'objectif et un ensemble de métriques et de contraintes. La fonction d'objectif se base sur une combinaison de métriques et de contraintes pour calculer le « meilleur » chemin acceptable. Il pourrait y avoir plusieurs fonctions d'objectifs considérées par le même nœud ou le même réseau. Par exemple, plusieurs DODAGs peuvent être utilisés avec l'objectif de (1) Trouver les chemins avec les meilleures valeurs de ETX (métriques) et d'éviter les liens non-cryptés (contrainte) ou (2) Trouver le chemin offrant la plus faible latence (métrique) tout en évitant les nœuds fonctionnant sur batterie (contrainte). La fonction d'objectif ne doit pas nécessairement préciser les métriques/contraintes, mais elle dicte des règles qui cadrent la formation du DODAG (par exemple, le nombre de parents secondaires, l'utilisation de l'équilibrage de charge, etc...).

### 2.2.2.1 Les messages de contrôle RPL

Il existe 4 principaux messages de contrôle RPL :

- **DODAG Information Solicitation (DIS)** : il est utilisé pour solliciter un DODAG Information Object (DIO) à partir d'un nœud RPL. Le DIS peut être utilisé pour sonder les nœuds voisins dans les DODAG adjacents.
- **DODAG Information Object (DIO)** : il est publié par la racine DODAG pour construire un nouveau DAG puis envoyé en multidiffusion via la structure DODAG. Le message DIO contient des informations réseau pertinentes qui permettent à un nœud de découvrir une instance RPL, d'apprendre ses paramètres de configuration, de sélectionner un ensemble parent DODAG et de conserver le DODAG.
- **Destination Advertisement Object (DAO)** : il est utilisé pour propager des informations de route inverse pour enregistrer les nœuds visités le long du chemin ascendant. Les messages DAO sont envoyés par chaque nœud, autre que la racine

---

DODAG, pour remplir les tables de routage avec les préfixes de leurs enfants et pour annoncer leurs adresses et préfixes à leurs parents.

- **Acknowledgement d'objet de publicité de destination (DAO-ACK)** : il est envoyé en tant que paquet unicast par un DAO destinataire (un DAO parent ou une racine DODAG) en réponse à un message DAO unicast.

### 2.2.2.2 La Construction DODAG

La construction DODAG est basée sur le processus Neighbor Discovery (ND), qui se résume en deux opérations principales :

- La transmission des messages de contrôle DIO émis par la racine DODAG pour construire des routes dans la direction descendante depuis le noeud racine vers le noeud client)
- Diffusion de messages de contrôle DAO émis par les noeuds clients et envoyés à la racine DODAG pour construire des routes dans la direction ascendante.

Afin de construire un nouveau DODAG, la racine DODAG diffuse un message DIO pour annoncer son DODAGID, son rang des informations pour permettre aux noeuds de déterminer leurs positions dans le DODAG, et la fonction Objective . Ce message sera reçu par un noeud client qui peut être un noeud disposé à rejoindre ou un noeud déjà joint. Lorsqu'un noeud disposé à rejoindre, le DODAG reçoit le message DIO, il :

1. Ajoute l'adresse de l'émetteur DIO à sa liste parente,
2. Calcule son rang selon la Fonction Objective, de sorte que le rang du noeud est supérieur à celui de chacun de ses parents,
3. Et transmet le message DIO avec les informations de rang actualisées. Le noeud client choisit le parent le plus préféré parmi la liste de ses parents comme noeud par défaut par lequel le trafic entrant est renvoyé.

Lorsqu'un noeud déjà associé à DODAG reçoit un autre message DIO, il peut procéder de trois manières différentes :

- De rejeter le message DIO selon certains critères spécifiés par RPL,
- Traiter le message DIO pour maintenir son emplacement dans un Existant DODAG
- Ou améliorer son emplacement.

En obtenant un rang inférieur dans le DODAG basé sur la fonction objectif et le coût du chemin. Chaque fois qu'un noeud change du rang, il doit jeter tous les noeuds de la liste des parents dont les rangs sont plus petits que le nouveau rang calculé pour éviter les boucles de routage. L'organigramme présenté à la figure ci-dessous résume le fonctionnement d'un routeur dans un DODAG.

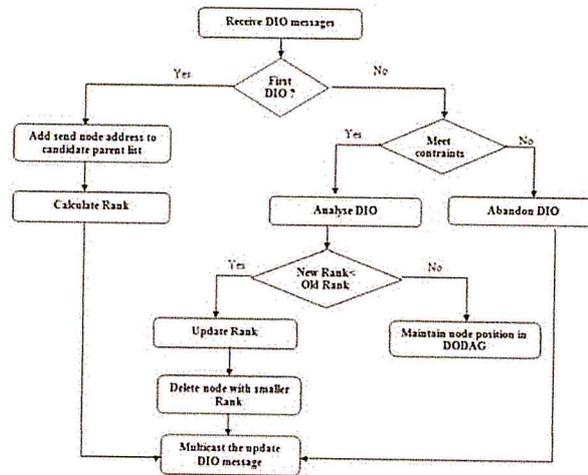


FIGURE 2.2 – L'opération dans un routeur dans DODAG

Après la construction du DODAG, chaque nœud client aurait un itinéraire par défaut par lequel il pourrait transmettre son trafic vers la destination de la racine DODAG . Évidemment, l'itinéraire par défaut est formé par le parent le plus préféré de chaque nœud.

### 2.3 Problème de l'équilibrage de charge :

[17][24] un nœud qui a une grande capacité de transfert peut être sollicité dans le routage des paquets par plusieurs nœuds. Cela peut surcharger ce nœud par rapport aux autres ce qui diminue ses performances, et peut donner des résultats inversés dans le routage, une métrique doit tenir en compte la capacité des nœuds sans négliger leur charge, qui affecte sur le réseau complet. on a classifié les métriques de routage qui sont utilisées par les nœuds pour déterminer leur chemin en deux classes :

#### Etat du nœud

- **état du nœud et attributs (NSA) :** est utilisé pour fournir des informations sur les caractéristiques de nœud, y compris l'attribut d'agrégation de données et la charge de travail de nœud. Certaines applications peuvent utiliser l'attribut de nœud d'agrégation dans leur décision de routage de manière à minimiser la quantité de trafic sur le réseau, ce qui augmente potentiellement sa durée de vie dans les environnements fonctionnant sur batterie . La charge de travail de nœud peut être une métrique utile à prendre en compte lors du calcul du chemin . La charge de travail du nœud peut être définie en cas de surcharge du processeur, de manque de mémoire ou de toute autre condition liée au nœud.
- **Objet énergétique de Nœud :** C'est la capacité de batterie de chaque nœud. L'énergie du nœud est utilisée lorsqu'un nœud doit considérer le niveau d'énergie de ses voisins avant de choisir parmi eux . Lorsque cette métrique est utilisée, nous devons regarder le type de fourniture, nous pouvons trouver trois états : sur batterie, alimenté et piègeur . par exemple , pour le nœud sur batterie, l'estimation de l'énergie (EE) représente le pourcentage de batterie restée dans

un nœud, elle est calculée comme suit :

$$EE = \frac{Power_{ist}}{Power_{max}} \quad (2.1)$$

- **Nombre de saute** : L'objet Comptage de bonds (HP) est utilisé pour indiquer le nombre de nœuds traversés le long du chemin . L'objet HP peut être utilisé comme une contrainte ou une métrique.  
Lorsqu'il est utilisé comme contrainte, le nœud racine indique le nombre maximal de sauts qu'un chemin peut traverser . Lorsque ce nombre est atteint, aucun autre nœud ne peut rejoindre ce chemin.  
Lorsqu'il est utilisé en tant que mesure, chaque nœud visité incrémente simplement le champ Nombre de bonds.

## Etat du lien

- **Thought object** : L'objet de débit est utilisé pour signaler le débit de liaison lorsqu'il est utilisé en tant que mesure, et il peut être utilisé en tant que mesure additive ou pour indiquer un maximum ou un minimum .
- **latence** : L'objet de latence est utilisé pour signaler la latence du chemin . Similaire au débit, la latence peut être utilisée comme une métrique ou une contrainte. Lorsqu'elle est utilisé en tant que métrique, l'objet de latence exprime la latence totale (mesure additive) et la latence maximale ou minimale le long du chemin . Lorsqu'elle est utilisée comme une contrainte, la latence peut être utilisée pour exclure les liens qui fournissent une plus grande latence que les valeurs prédéfinies .
- **Fiabilité du lien** : présenter par fiabilité du niveau de qualité des liens(LQL) et nombre de transmissions attendues(ETX).  
L'objet LQL (Link Quality Level) est utilisé pour quantifier la fiabilité du lien en utilisant une valeur discrète, de 0 à 7, où 0 indique que le niveau de qualité de liaison est inconnu et 1 indique le niveau de qualité de liaison le plus élevé. ETX est une métrique de lien, représente le nombre moyen de fois qu'un nœud doit transmettre le message pour terminer la transmission . En sommant tout le ETX le long du chemin du parent candidat, une métrique de chemin basée sur ETX est créée, donnant le nombre attendu de transmissions qu'un nœud doit faire pour envoyer un paquet au récepteur . Pour calculer ETX, le taux de réception de paquet (PRR) doit être calculé :

$$PRR = \frac{Nombre\_paquets\_reus}{Nombre\_paquets\_envoys} \quad (2.2)$$

Un ETX entre deux nœuds est calculé en tenant compte des deux côtés, dans l'équation suivante PRR\_A est le PRR calculé à partir de A sur A-B et PRR\_B est le PRR calculé à partir de B sur A-B :

$$ETX = \frac{1}{PRR_A * PRR_B} \quad (2.3)$$

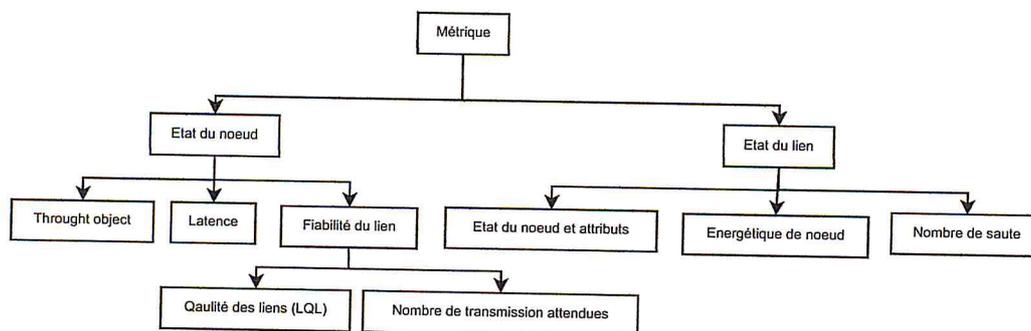


FIGURE 2.3 – Diagramme de classification des métriques

RPL est conçu pour les LLN et effectue le routage de manière distribuée . Cependant, la fonction de balance de charge critique est manquante dans RPL . Sans équilibrage de charge, les trafics de données inégaux, ainsi que la distribution non uniforme des nœuds de capteurs dans les LLN à grande échelle peuvent entraîner un déséquilibre de charge significatif pour les nœuds de capteurs qui ont plus de voisins que d'autres. Par conséquent, l'épuisement de l'énergie de ces nœuds de capteurs est beaucoup plus rapide que ceux avec une charge de travail légère . Cela entraînera des trous et des trous dans tout le réseau et rendra le réseau déconnecté.

De plus, les métriques de routage recommandées et couramment utilisées avec le protocole RPL, par exemple le nombre de transmissions attendues (ETX) et le rapport de livraison de paquets, se concentrent uniquement sur la qualité de communication par paire entre deux nœuds . Étant donné que la communication sans fil peut être facilement perturbée par les transmissions des nœuds voisins, même un lien parfait entre deux nœuds peut ne pas fonctionner correctement dans des conditions de trafic intense .

## 2.4 Solution existant pour l'équilibrage de charge :

### 2.4.1 LB-RPL

[18] Ce protocole designé pour optimiser l'équilibrage de charge de travail entre les nœuds, il répartit la charge de travail entre tous les nœuds par une technique distribuée et non intrusive et fournie pour réaliser la signalisation et la détection de déséquilibre de charge de travail automatique ; La distribution de la charge de travail et les conditions de communication sont conjointement considérées pour sélectionner les chemins d'acheminement de données optimaux pour maximiser le débit de distribution de paquets . Le protocole LB-RPL modifie la procédure de construction DODAG dans RPL en incorporant les deux fonctionnalités suivantes :

#### 1. détection et signalisation de déséquilibre de la charge de travail :

Pour effectuer la détection et la signalisation du déséquilibre de la charge de travail, le nœud ne transmettra pas immédiatement le nouveau DIO avec son rang. Au lieu de cela, il démarrera un temporisateur proportionnel à sa charge de travail au cours de la période précédente et transmettra le message DIO après l'expiration du délai. La même procédure se poursuit à chaque niveau croissant de DODAG jusqu'à ce qu'un message DIO avec un numéro de version plus récent atteigne les nœuds de papier . La détermination de la valeur de synchronisation appropriée est essentielle pour la détection du déséquilibre de charge et de la signalisation . Pour évaluer la

valeur de la minuterie, ils ont utilisé la méthode suivante :

$$T_i = T_0 * \text{Compteur d'utilisation du tampon} \quad (2.4)$$

Lorsqu'un nœud reçoit plusieurs copies du même message DIO provenant de différents nœuds inférieurs, il forme son ensemble parent comme décrit dans RPL. Selon l'heure à laquelle ces messages DIO sont reçus de ces nœuds parents, l'ordre de priorité de ces parents est déterminé en conséquence.

## 2. charge de transfert de données équilibré :

Contrairement à RPL, où un seul nœud de capteur dans l'ensemble parent est sélectionné comme parent préféré selon une métrique unique, dans LB-RPL, les  $k$  premiers nœuds parents sont tous considérés comme un prochain saut potentiel pour le transfert de données. La probabilité pour le nœud  $i$  de transmettre le paquet de données à un nœud parent particulier  $j$  est calculée comme suit :

$$f_{ij} = \frac{(1 - p_{ij}^c)}{\sum_{j=1}^k (1 - p_{ij}^c)} \quad (2.5)$$

### 2.4.2 CO-RPL

[8] pour les WSN mobiles à faible puissance et à perte, l'architecture de corona est utilisée pour la localisation de nœuds mobiles. Cette architecture facilite la recherche rapide de parents alternatifs comme le prochain saut. L'architecture de Corona repose sur le concept simple de division de la zone de réseau en couronnes. Une couronne est définie comme une région circulaire avec un certain rayon centré sur la racine DAG.

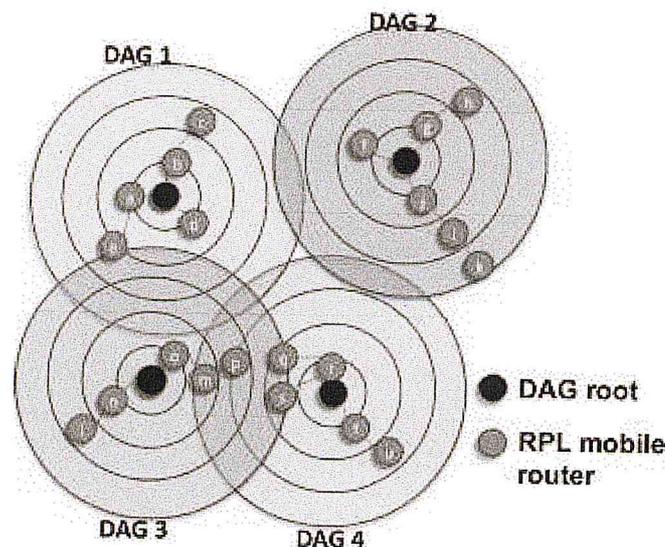


FIGURE 2.4 – Architecture réseau composée de quatre DODAG avec des racines DAG statiques

Le comportement du routeur mobile Chaque routeur RPL écoute les DIO des nœuds voisins. S'il ne reçoit aucun DIO donc il n'a pas de voisins et il sera isolé du reste du réseau. Le nœud sera alors en mode inactif et continuera à envoyer des messages DIS

jusqu'à ce qu'il reçoit une DIO pour rejoindre un DAG (lorsqu'il se trouve à portée d'un nœud). Lorsqu'un routeur RPL reçoit le premier DIO, il rejoint le DAG, calcule son rang et son CID, et diffuse le message DIO mis à jour à ses voisins. S'il reçoit plusieurs DIO (avec plus d'un ID corona), le nœud sélectionne un parent. Il le fait en fonction de l'ID corona minimum et de la meilleure qualité annoncée dans les DIO reçus.

Un problème qui doit être résolu est lorsqu'un nœud change de position sans que son identifiant C soit changé (il est toujours proche de la même racine DAG). Cela ne déclenche pas la découverte d'un voisin et nécessite une attention particulière. Traiter chaque fois que la table des voisins change. Cette modification est suffisante car même si un nœud mobile conserve le même identifiant corona, la liste de ses voisins changera nécessairement avec sa position modifiée. Si l'un des ID C des nœuds voisins est modifié ou si un nouveau voisin est détecté, le nœud déclenche une découverte de voisin immédiat.

### 2.4.3 ELT

[13] La métrique de durée de vie prévue (ELT) c'est combien de temps un capteur doit-il vivre avant que son batterie soit déchargée et montre comment l'estimer pour chaque nœud. Un nœud N calcule son ELT de la manière suivante :

1. estimer le trafic total que N doit transmettre en tenant compte à la fois du trafic qu'il génère et du trafic entrant de ses enfants :

$$T_{total}(N) = T_{gen}(N) + \sum_{i \in Children(N)} T_{total}(i) \quad (2.6)$$

2. multiplier le trafic que N doit transmettre par la moyenne nombre de retransmissions, donné par la fiabilité du lien à son parent préféré ( $ETX(N, P_N)$ ) : plus on retransmet des missions nécessaires, plus l'énergie est consommée :

$$ELT(N) = T_{Total}(N) * ETX(N, P_N) \quad (2.7)$$

3. calculer le taux d'occupation du milieu en prenant en compte le débit auquel les données sont envoyées :

$$ELT(N) = \frac{T_{Total}(N) * ETX(N, P_N)}{DATARATE} \quad (2.8)$$

4. calculer l'énergie dépensée pour transmettre tout le trafic par multipliant le taux d'occupation par la transmission puissance de sa radio :

$$ELT(N) = \frac{T_{Total}(N) * ETX(N, P_N)}{DATARATE} * P_{TX}(N) \quad (2.9)$$

5. enfin, calculer la durée de vie restante de N comme le rapport entre son énergie résiduelle et l'énergie dépensée pour transmettre son trafic :

$$ELT(N) = \frac{E_{res}}{\frac{T_{Total}(N) * ETX(N, P_N)}{DATARATE} * P_{TX}(N)} \quad (2.10)$$

## 2.4.4 QU-RPL

[19] un mécanisme léger basé sur le taux d'utilisation de la file d'attente, pour l'équilibrage de la charge afin de prendre en charge un trafic important.

1. le metrique ETX ne reflète pas la congestion qui provoque la perte de réseau pour résoudre ce problème ,cette solution ajoute une nouvelle valeur QU . QU c'est un compte qui calculer l'espace resté dans la queue ,elle est calculé comme suit :

$$Q_u = \frac{\text{depaquet dans la file d'attente d'un noeud } k}{\text{taille totale de la file d'attente d'un noeud } k} \quad (2.11)$$

2. **propagation la congestion :** Dans QU-RPL, chaque nœud distribue ses informations QU à ses nœuds voisins . Il y a plusieurs façons de mettre en œuvre ceci dans la norme RPL la plus simple est consisté à modifier uniquement le caractère OF et à redéfinir RANK pour contenir la valeur QU en même temps que le RANK précédemment défini (c'est-à-dire le nombre de sauts). l'équation suivant calculé le nouveau RANK :

$$RANK\_QU(k) = \beta(h(k) + 1) + (\beta - 1)Q(k) \quad (2.12)$$

3. Problème de sélection d'un nouveau parent : L'utilisation de la nouvelle métrique de routage et de la propagation QU dans QU-RPL créer un nouveau défi appelé effet d'élevage comme le montre la figure 3.5 :

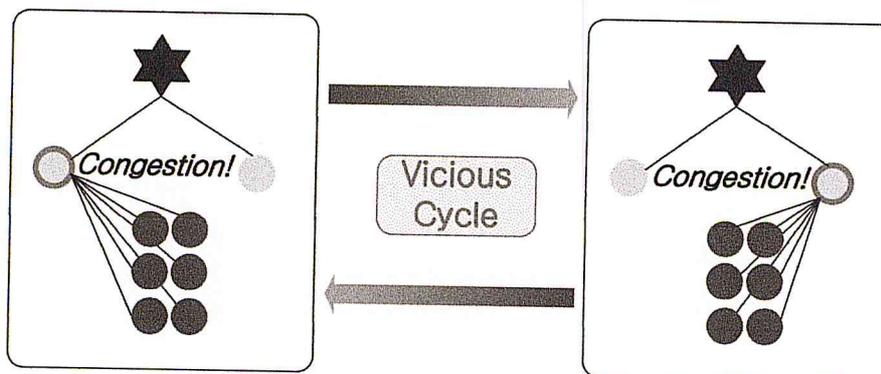


FIGURE 2.5 – L'opération dans un routeur dans DODAG

Le problème est que les nœuds enfants peuvent avoir découvert la congestion dans le nœud parent et cela peut changer le parent en même temps, ce qui entraîne une nouvelle congestion sur le bon nœud 1-hop . De cette façon, un cercle vicieux peut être créé, où un groupe de nœuds répète indéfiniment des changements de parent sans signification et sans réaliser l'équilibrage de charge. pour résoudre ce problème cette condition doit être remplie :

$$\mu > \gamma \quad (2.13)$$

où  $\mu$  est l'indicateur d'encombrement qui indique l'encombrement du trafic autour du nœud  $k$ , et  $\gamma$  est une valeur de seuil pour décider quand effectuer l'équilibrage de charge . Si cette condition (14) est satisfaite, le noeud  $k$  doit changer son noeud parent en considérant l'équilibrage de charge. Sinon, si seule la condition de stabilité

est satisfaite, le nœud k dans QU-RPL change son nœud parent de la même manière que dans le RPL par défaut . En d'autres termes, QU-RPL a le même mécanisme de changement parent que RPL lorsqu'un nœud ne subit pas d'encombrement.

#### 2.4.5 ALABAMO

[15] Cette OF utilise à la fois le profil de trafic des nœuds et l'ETX des liaisons pour résoudre le problème de déséquilibre . La proposition est appelée AL-ABAMO (A LoAd BALancing MOdel pour RPL) ALABAMO est un modèle flexible qui utilise deux paramètres d'entrée et améliore la tâche de choix de parent préféré dans RPL visant à l'extension de la durée de vie du réseau. [] L'implémentation d'ALABAMO est une fonction objective basée sur le rang minimum avec fonction d'hystérésis (MHROF). Dans cette solution, chaque nœud diffusait le nombre de paquets transmis au cours du dernier intervalle de mesure, ce qui inclut les paquets générés et transmis, ces informations sont intégrées dans les messages DIO . Pour choisir le parent préféré, les nœuds considèrent les métriques de qualité de chemin standard, telles que ETX, et le nombre de paquets que chacun de ses parents candidats a déjà envoyé.

ALABAMO utilise un mécanisme d'hystérésis similaire à celui utilisé dans MHROF pour éviter les changements instables pendant les fluctuations rapides . Nous définissons deux constantes auxiliaires pour fournir une flexibilité lors de la pondération des paramètres dans notre fonction objective.

#### 2.4.6 Load balancing in OF :LB-OF

[9] LB-OF Stimule la durée de vie de réseau LB-OF équilibrera le trafic de donnée en prenant en compte le nombre d'enfants pour chaque parent candidat. Dans cette solution les messages de contrôle DIO a été modifié du côté de parent et DIA par rapport les nœuds enfant.

1. **Une nouvelle technique d'utilisation pour l'DIO modifiée** : le nœud DIO reçu du nœud enfant est compté par le nœud parent préféré en utilisant un tampon spécial (ensemble) créé. à cet effet, le DIO modifié contient l'adresse IP du parent préféré choisi. Ainsi, pour chaque DIO reçu, le nœud correspond à sa propre adresse IP avec l'adresse IP parent préférée qui est insérée dans le message DIO, puis incrémente le nombre d'enfants défini par un pour ce nœud s'il y a correspondance.
2. **Nouvelle métrique proposée pour la sélection des parents** : cette solution modifie le message de contrôle DIO, les informations nouvelles ajoutées sont a 2 octets nommés nombre de nœuds enfants(child node count CNC), Le format du corps d'objet CNC est le suivant :
  - **'P' flag** : l'adresse de l'adresse parent . si elle est définie sur 1, indique qu'il y a un champ d'adresse parent dans l'objet CNC.
  - **CNC** : sur 8 bit, représentant le nombre de nœuds enfants.
  - **MAX-CNC** : elle est sur 8 bit, c'est le nombre maximum d'enfants qu'un nœud peut contenir .
  - **parent adresse** : 128 bits, ipv6 Adresse du nœud parent . Ce champ n'est présent que lorsque le drapeau 'p' est mis à 1.

Dans le mode de stockage, DAO peut être utilisé pour l'enregistrement des nœuds enfants alors que no-pathDAO peut être utilisé pour le dés-enregistrement, et cela permet de compter le nombre de nœuds enfants . Ainsi, pour minimiser la charge

de trafic, le champ d'adresse parent dans le L'objet CNC ne doit pas être présent dans le stockage.

Dans le mode sans stockage, NS / NA pourrait être une option pour le compte de nœud enfant. Lorsque l'indicateur p est activé, l'adresse parent dans l'objet CNC devrait être recherchée pour le comptage des nœuds enfants.

Lorsque cette métrique CNC est utilisée, l'informatique RNK reflète la capacité de chaque nœud à contenir plus de nœuds enfants. De plus, un nouveau calcul RANK a été effectué

$$RANK = \frac{CNC}{MAYCNC} * 255 \quad (2.14)$$

un nœud avec RANK plus petit à une grande priorité pour accepter de nouveaux nœuds enfants, un nœud avec  $RANK = 255$  ne devrait plus contenir de nouveaux nœuds enfants.

### 2.4.7 Power Control-RPL

[16] PC-RPL qui vise à obtenir une meilleure bande passante par rapport à RPL . PC-RPL s'attaque aux problèmes de déséquilibre de charge et de terminal cachés dans un réseau multi-sauts en contrôlant la topologie de routage via la puissance de transmission et le contrôle de seuil RSSI . Sur la base de ces idées, PC-RPL emploie un nouveau mécanisme d'élection des parents qui utilise des seuils RSSI adaptatifs et une valeur RSSI de référence d'un nœud candidat parent, en plus des règles par défaut dans RPL (c'est-à-dire, distance de saut et sélection parentale ETX). PC-RPL contrôle ces RSSI thresholds de manière adaptative pour atténuer les problèmes de terminal cachés et réaliser l'équilibrage de charge. De plus, le PC-RPL minimise la puissance d'émission de données en fonction des résultats de transmission RSSI et de transmission (succès ou échec), qui réduisent les terminaux et encombrant les liaisons sans compromettre la fiabilité.

### 2.4.8 ECRM

[23] Energy and congestion-Aware Routing for Smart Grid (ECRM) un mécanisme de sélection de nœud parent dynamique dans RPL pour la mesure intelligente dans les réseaux AMI, en tenant compte à la fois de l'énergie résiduelle et de l'utilisation des files d'attente.

Lorsque la topologie est construite, chaque nœud commencera périodiquement à diffuser des messages DIO. Ces paquets contiennent des informations sur leur rang, le facteur QU, l'état de l'énergie, la valeur ETX et l'identité de DODAG. Lorsqu'un nœud reçoit des messages DIO provenant de nœuds voisins, il génère un ensemble de nœuds de candidats parents avec un nombre de sauts de 3 sauts et un ETX minimum de 3 sauts vers le routeur de bordure. ETX prédit le débit pour les courtes routes de 1, 2 ou 3 sauts . A partir de tous les nœuds du jeu de candidats, le nœud choisira un nœud parent ayant une énergie résiduelle et QU avec un certain niveau et un rang le plus bas et ETX vers le routeur frontière . Chaque nœud modifie son nœud parent actuel lorsque les informations sur les candidats parents ont été modifiées.

dans cette solution il propose une métrique de routage,  $R_{pro}(pk)$ , pour sélectionner le meilleur parent alternatif, qui est défini comme suit :

$$R_{pro}(pk) = Rank(pk) + ETX(k, pk) + \alpha Q(pk) + \beta Eres(pk) \quad (2.15)$$

où  $Rank(pk)$  est la valeur de rang du nœud parent,  $ETX(k, pk)$  montre la valeur ETX du nœud.  $Q(pk)$  est le QU sur le nœud parent  $pk$ , QU il est déjà calculer par l'équation (2.3).  $\alpha$  est un coefficient qui contrôle le poids de QU,  $\beta$  est le poids donné à  $E_{res}(pk)$  et  $E_{res}(pk)$  est l'énergie résiduelle du nœud parent  $pk$  a été calculé comme suit :

$$E_{res} = InitialEnergy - E_{cons} \quad (2.16)$$

le processus du schéma proposé est présenté dans la figure 3.6 .

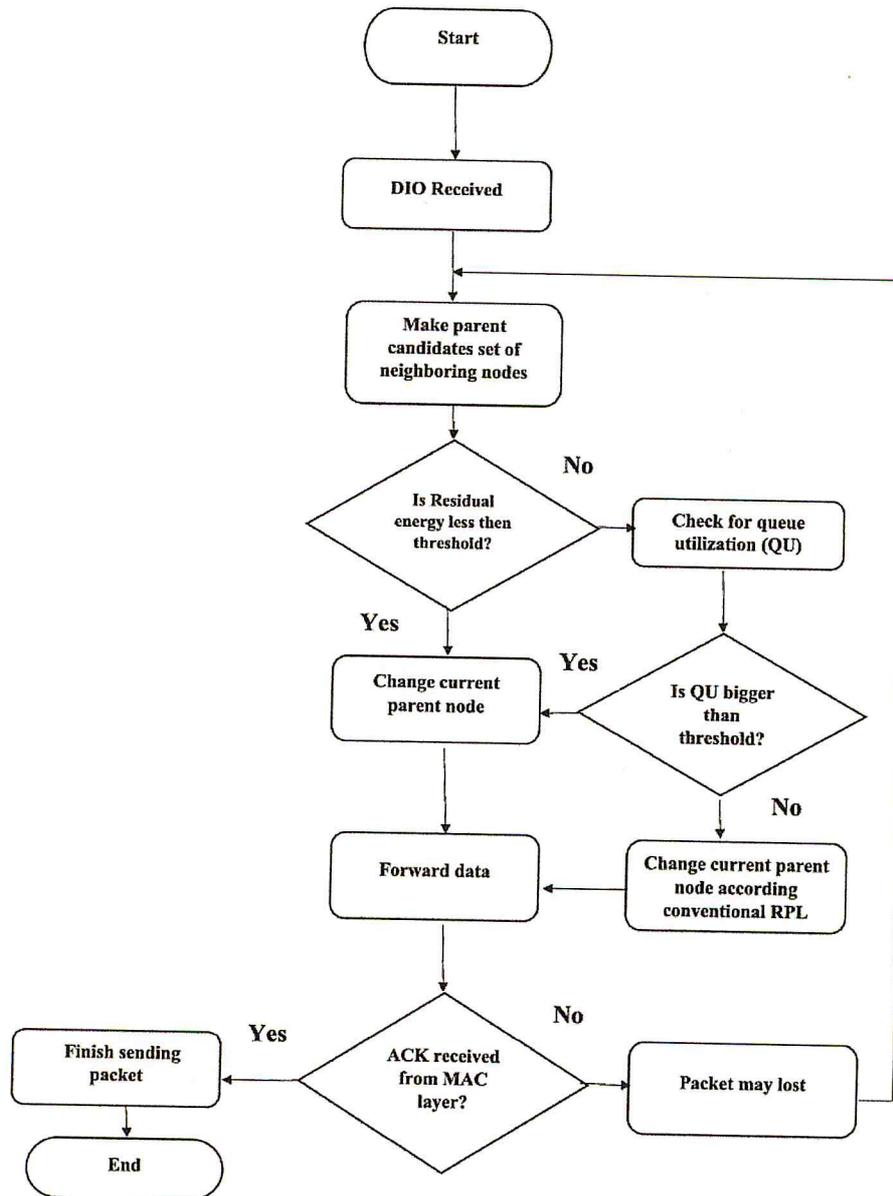


FIGURE 2.6 – Processus du schéma proposé de ECRM .

## 2.5 comparaison entre différentes solutions

TABLE 2.1 – Table de comparaison entre les solutions existant pour l'équilibrage de charge

Proposition	Métrique	Date	Implémentation	application	avantage	inconvénients
LB-RPL	file d'attente, Délai	2013	NS2		Réduit le congestion, limité le perte des paquets	
CO-RPL	delai end to end, énergie	2014	MWSN	réseau de capteur comme application de collecte de donnée du capteur utilisent des robots mobiles, utiliser le réseau de capteur pour surveiller les signer des patients par transmis aux leur médecine	sélectionner meilleur parent, implémenter un mécanisme de réduction de chemin d'échelle de noeud, trouver des parent alternatif	consomme plus d'énergie avec l'augmentation du débit de transmission de paquet, n'offre pas une bonne performance en terme de délai moyen
ELT	reliability, delay, energy consumption	2014	simulated using WSNnet		*Réduire le nombre de changements parents, Minimise la consommation d'énergie	n'offre pas une bonne performance en termes de délai moyen de bout en bout et de taux de livraison de paquets
QU-RPL	Rank, ETX, fils attendre	2015	contikiOS	Nombreuses application	simple, Résoudre le congestion, Sélection le meilleur parent par rapport a buffet	nécessite une distance de saut plus grande, la fréquence de changement de parent de QU-RPL augmente avec la puissance de transmission

ALABAMO(A LoAd Balancing MOdel for RPL)	the traffic profile of the nodes( le profil de trafic des noeuds), ETX	2016		application de collecte de données multi-hop, comme la surveillance environnementale	compatible avec RPL et permettant un acheminement équilibré et respectueux de la circulation, Améliorer la durée de vie du réseau	
LB-OF	Rank	2017	contikiOS	cooja	équilibre le nombre des enfants dans les parents	
PC-RPL(power control rpl)	RSSI, ETX, cout hop .	2017	TinyOS		sélectionner, minimiser la transmission des messages de contrôle, Augmenter le succès de livraison des paquets, minimiser le nombre de paquets perdu	il y a perd des paquet lorsque le valeur de seuil et puissance d'émission étaient unifiées (défaut) parmi les noeuds
ECRM(Energy and congestion-Aware Routing for Smart Grid)	Énergie, file d'attente .	2017	contiki OS	Smart Grid	éviter les boucles de routage, éviter les incohérences dans le DODAG, minimiser la consommation d'énergie qui augmente la durée de vie de réseau, minimiser le perdant des packs	Augmenter la consommation d'énergie et fils l'attendre lorsque le fréquence des messages d'application très rapide

---

## Conclusion

Dans ce chapitre nous avons parlé sur les principes et classification des protocoles de routage et ainsi présenté les deux protocoles de routage les plus utilisés aujourd'hui : RPL et AODV et nous avons détaillé la méthode de joindre entre nœud enfant et nœud parent.

Notre travail est basé sur l'équilibrage des charge dans l'IoT où nous avons choisi le protocole RPL pour discuter des problèmes l'équilibrage de charge et nous avons présenté quelques solutions qui déjà existent pour réaliser l'équilibrage de charge dans l'IoT .

Dans le chapitre suivant, nous proposons une solution pour réaliser l'équilibre de charge qui manque dans le protocole de routage RPL.

## Chapitre 3

# Proposition d'une nouvelle fonction objective pour équilibrage de charge

### Introduction

Il est conclu que les OF actuels conduisent à la construction d'une topologie qui souffre d'un trafic de charge déséquilibré dans les nœuds à goulot d'étranglement, en particulier pour les nœuds du premier saut (c'est-à-dire à partir de la racine) . Par conséquent, ce problème a un impact crucial sur la durée de vie de ces types de nœuds et aussi l'épuisement de la batterie de ce nœud parent surchargé peut affecter négativement la fiabilité du réseau . Ce problème difficile est toujours un problème ouvert et c'est la raison pour laquelle on a tenté de le résoudre, pour ce faire, nous proposons une nouvelle fonction objective qui équilibre le nombre de nœuds enfants pour les nœuds surchargés afin de maximiser la durée de vie de réseau dans RPL.

Ce chapitre peut être résumée comme suit. Tout d'abord, nous avons commencé par un diagramme de classe montrant les classe de protocole RPL et les relation entre eux ainsi que les fonctions ajouter . ensuite, nous avons détailler les métriques de routage utilisé tout en expliquant les méthodes de calcule et les modifications nécessaire ajoutées à la structure RPL . et L'idée principale de ce travail est de combiner ses métriques de routage des nœuds qui sont l'énergie consommé(EN\_CON) et le nombre de voisin(NBV) avec une métrique de liaison qui est le nombre de transmissions prévu (ETX)et avec le mécanisme de fonctionnement basé sur la recherche de routes assurant l'équilibrage de bout en bout, Cette combinaison permet de résoudre certaines limitations dues à l'utilisation de métriques uniques.

### 3.1 Motivation

La motivation peut être illustrée à l'aide d'un exemple présenté à la figure 3.1, La figure montre un exemple de configuration de réseau et explique comment le routage DAG est construit sur la base des valeurs de rank et ETX, les nœuds C et D ont les nœuds A et B comme meilleurs nœuds candidats parents et éventuellement, les deux nœuds choisissent le nœud B comme meilleur nœud parent.

Supposons que le nœud C choisisse le nœud B comme meilleur nœud parent au début, puis que le nœud D le fasse plus tard. Par conséquent, les nœuds C et D transmettent éventuellement leurs propres données ou transmettent leurs données de nœuds enfants

au nœud B . Toutefois, si l'énergie consommé du nœud B est très supérieure à celle du nœud A, les communications entre les nœuds C / D et B vont être déconnecté en raison de l'épuisement de l'énergie du nœud B. En conséquence, les nœuds C et D démarreront le processus pour rechercher et définir le meilleur nœud parent, ce qui entraînera une surcharge de réseau indésirable . Au lieu de choisir le nœud B, le nœud A pourrait être le meilleur choix pour le meilleur nœud parent. Ainsi, le réseau ne peut pas déclencher de mécanisme de réparation local ou global pour la reconfiguration de la topologie et par conséquent, les performances du réseau seraient meilleures. Outre le problème d'énergie consommé, le nombre de voisin du meilleur nœud parent peut affecter les performances du réseau. Par exemple, étant donné que le nœud B est le nœud parent des nœuds C et D, le nœud B a plus de voisin que le nœud A. Si plusieurs nœuds du réseau sélectionnent le nœud B comme nœud parent, le nœud B sera saturée et le réseau sera non équilibre . Par conséquent, si l'autre nœud (dans l'exemple, le nœud A) ayant le statut de moins de voisin était sélectionné en tant que nœud parent, le surcharge de réseau aurait pu être évitée . En résumé, comme nous l'avons décrit ci-dessus avec la figure 3.1, afin d'améliorer les performances du réseau, l'état de l'énergie consommé et le nombre des nœuds voisins des nœuds candidats parents doivent être pris en compte afin de choisir le meilleur nœud parent . Par conséquent, l'objectif de schéma proposé vise à fournir un moyen d'améliorer les performances du réseau en prenant en compte les deux facteurs (énergie consommé et le nombre de voisin des noeuds ).

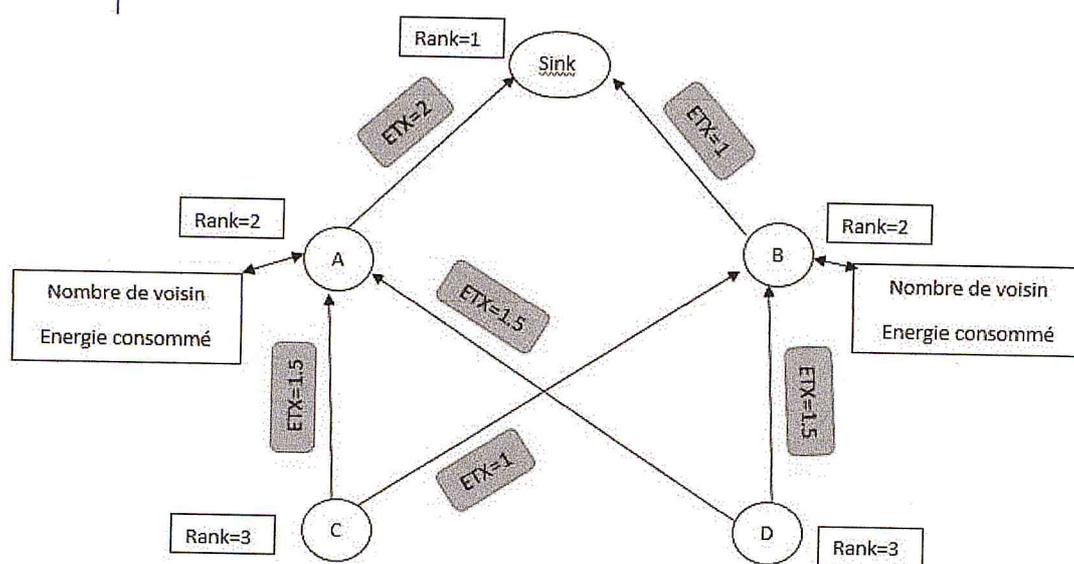


FIGURE 3.1 – Un exemple de construction DODAG et de sélection de nœud parent

### 3.2 Diagramme de classe pour le protocole RPL

Un réseau peut exécuter plusieurs instances de RPL simultanément, Chacune de ces instances peut servir des contraintes ou des critères de performance différents et potentiellement antagonistes . RPL utilise quatre valeurs pour identifier et gérer une topologie, Le

premier est un RPLInstanceID. Un RPLInstanceID identifie un ensemble de un ou plusieurs DAG orientés destination (DODAG), Le second est un DODAGID. Une instance RPL peut avoir plusieurs DODAG, chacun ayant un DODAGID unique ,Le troisième est un DODAGVersionNumber. Un DODAG est parfois reconstruit à partir de la racine DODAG, en incrémentant le DODAGVersionNumber, Le quatrième est le rang, Rang établit une commande partielle sur une version DODAG, définissant positions de nœud individuelles par rapport à la racine DODAG.

La fonction objectif (OF) définit la manière dont les nœuds RPL sélectionnent et optimisent itinéraires dans une instance RPL a l'aide de la classe ICMPV6, elle est identifiée par un objectif Point de code (OCP) et elle définit comment les nœuds traduisent une ou plusieurs métriques (ETX ,ENERGYTOTAL, ...etc) et contraintes, et une valeur appelée Rank qui se rapproche de la distance du nœud à une racine DODAG. Chaque message de contrôle ICMPV6 RPL (DIS ,DAO,DIO) à un champ RPLInstanceID et il est identifié par un code qui précise le type de message de contrôle, la classe de message DIS contient les attribut suivants :Drapeaux , Réservé. la classe de message DIO contient les attributs suivants : **RPLInstanceID** : Est une information défini par la racine DODAG qui indique l'ID de l'instance RPL dont le DODAG fait partie.

**Grounded (G)** : Est un flag indiquant si le DODAG actuel satisfait à l'objectif défini par l'application.

**Mode of Opération(MOP)** : Identifie le mode de fonctionnement de l'instance RPL définie par la racine DODAG.

**DODAGPreference (Prf)** : Un entier qui définit comment la racine de ce DODAG est préférable à d'autres racines DODAG dans l'instance.

**Version Number** : indique le numero de version d'un DODAG qui est généralement incrémenté à chaque mise à jour d'informations .

**Destination Advertisement Trigger Sequence Number (DTSN)** : est un indicateur qui est utilisé pour maintenir des itinéraires descendants.

**Rank** : un champ qui spécifie le rang du nœud envoyant le message DIO.

**Flags** : Champ non utilisé réservé aux flags. Le champ doit être initialisé à zero par l'expéditeur et doit être ignorée par le récepteur.

**Reserved** : Champ non utilisé, Le champ doit être initialisé a zéro par l'expéditeur et DOIT être ignore par le récepteur.

**DODAGID** :Une adresse IPv6 définie par une racine DODAG qui identifie de manière unique un DODAG . Le DODAGID doit être une adresse IPv6 appartenant à la racine DODAG.

la classe DAO contient les attribut suivants : **RPLInstanceID** : Un champ indiquant l'instance de topologie associée au DODAG.

**K** : Indique que le destinataire doit retourner un DAO-ACK.

**D** :Indique que le champ DODAGID est présent, Ce drapeau doit être défini quand un RPLInstanceID local est utilisé.

**Flags** : champ Flags est réservés aux flags.

**Reserved** : Un champ non utilisé de 8 bits.

**DAOSequence** : incrémenté à chaque message DAO unique à partir d'un nœud et echo dans le message DAO-ACK.

Un nouveau nœud peut rejoindre un réseau déjà formé en diffusant un message DIS pour solliciter en réponse un message DIO qui contient des informations sur le DODAG

. une fois qu'il choisit son parent il envoie un message DAO a son parent préféré en attendant une accusé de réception dao-ack.

la figure 3.2 montre le diagramme de classe de du scénario ci-dessus :

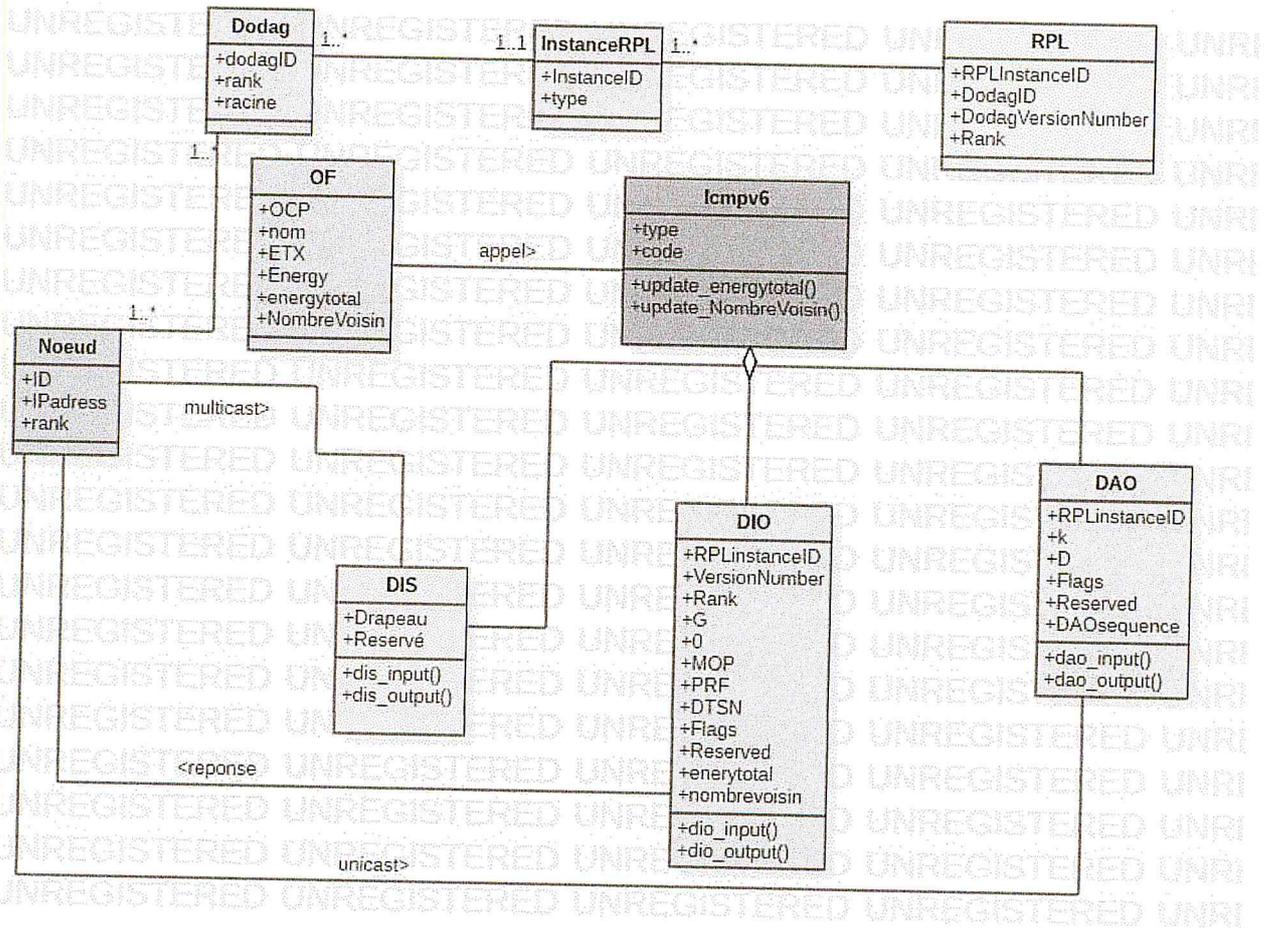


FIGURE 3.2 – Diagramme de classe de protocole RPL

### 3.3 Fonction objective

Une fonction objectif (OF) définit la manière dont un nœud RPL sélectionne et optimise les itinéraires au sein d'une instance RPL en fonction des objets d'information disponibles. Prenons un réseau physique constitué de plusieurs liaisons de différentes qualités, telles que le débit, la latence, et de nœuds de différentes qualités, tels que, ceux qui fonctionnent sur batterie ou sur secteur . Si le réseau achemine différents types de trafic, il peut s'avérer utile de le transporter en fonction de différents OF qui optimisent différentes métriques ou satisfont à des contraintes . Ainsi, le OF sert à orienter le trafic sur différents trajets en fonction des besoins . Ces exigences sont en fait codées dans une logique de programmation, appelée OF, et utilisées par RPL lors des opérations de routages. Le processus de construction DODAG de RPL est principalement déterminé par la fonction objectif (OF) et les métriques de chemin . Une OF définit la manière dont un nœud calcule son rang, basé sur un ou plusieurs métriques . Un rang représente la distance relative à la racine DODAG . De plus, Il décrit également comment un nœud doit choisir

son parent préféré. L'une des spécifications de RPL est donc qu'il crée une route vers la racine en fonction de la fonction Objectif . Pour cette raison, l'OF est considérée comme le facteur clé pour déterminer dans le réseau, le parent préféré d'un nœud . Un nœud peut avoir plus d'un parent en particulier dans un réseau avec une énorme densité . Pour cela, la fonction Objectif essaie de choisir quel parent est adapté pour un nœud que d'autres. En outre, le choix de parent par la fonction objectif est basé sur un ou plusieurs critères spécifiques qui sont des métriques . Ces métriques peuvent être spécifiées par le concepteur en fonction de ses besoins . Jusqu'à présent, le groupe de travail ROLL a spécifié deux fonctions objectives. La première est l'objective Fonction Zéro (OF0), sur laquelle le critère de sélection du meilleur parent est le nombre minimum de sauts (Hop Count). Le deuxième est le minimum rank avec fonction d'objectif d'hystérésis (MRHOF). Contrairement à OF0, le MRHOF sélectionne les itinéraires en fonction de la métrique de comptage de transmission attendue (ETX). La valeur minimale de cette métrique détermine donc l'itinéraire optimal pour remonter vers la racine.

### 3.4 Principale métrique proposée

Le choix des métriques est donc essentiel car ces dernières ont le rôle le plus important dans la prise de décision de cette fonction objectif . nous avons choisit trois métrique pour améliorer l'équilibrage de charge qui sont l'énergie consommé par le nœud, nombre de voisin de nœud et ETX(nombre de transmissions prévu) qu'on l'a expliqué au chapitre2 . Dans cette section, nous décrivons tous les métriques que nous avons étudiées dans cette étude.

#### 3.4.1 Energie consommé :

EN\_CON est l'énergie des nœuds dépensés lors de l'échange d'informations dans le réseau . La consommation d'énergie d'un nœud est calculée en fonction de la puissance consommée lors de la transmission et de la réception . De plus, d'autres paramètres sont pris en compte pour la calculé. L'OF peut alors sélectionner l'itinéraire vers la racine en fonction de la consommation énergétique de ses parents, chaque nœuds enfants choisit le parent qui a consommé moins d'énergie que d'autre nœud parents condidats . Cette métrique est donc la principale à prendre en compte dans le contexte d'une application efficace au niveau énergétique. l'énergie consommé est une métrique du nœud La formule utilisée pour la calculer est la suivante :

$$EN\_CON = Transmit*19,5 + Listen*21,5 + CPU\_time*1,8 + LPM*0,0545 * \left( \frac{3V}{32768} \right) \quad (3.1)$$

Où : CPU : représente la consommation d'énergie au cours de la mode pleine puissance.

LPM : représente la consommation d'énergie en mode basse consommation.

Transmit correspond aux opérations de transmission, alors que Listen est destiné aux opérations d'écoute.

La consommation d'énergie est l'un des enjeux majeurs des réseaux de l'IoT . c'est une préoccupation majeure en raison de l'approvisionnement en énergie limité des nœuds . Par conséquent, les protocoles de routage sont nécessaires pour maximiser la forme de communication et de calcul économe en énergie afin de prolonger la durée de vie de la batterie.

### 3.4.2 Nombre de voisin

être le parent préféré de plus d'enfants signifie plus de frais généraux et une charge déséquilibrée, ce qui conduit à drainer sa propre énergie beaucoup plus rapidement que les autres parents candidats et aussi peut entraîner une distribution inégale du trafic de données . Pour résoudre ce problème, nous proposons une nouvelle métrique qui est nombre de voisin(NBV).

Lorsqu'un nœud veut rejoindre un réseau il envoie un message DIS en multidiffusion pour demander des informations sur le DODAG, en d'autres termes il demande un message DIO , dans ce message DIS il incrémente ses nombre de voisin , dans notre travail on a récupérer ce totale des voisins et on la injecter dans le message DIO comme ça dès qu'un nœud reçoit le message DIO d'un parent candidat il sera ses nombre de voisin de ce dernière .comme le montre la figure 3.3 le nœud M a deux parents candidat (G,C) quand il reçoit le message DIO de ses deux parent candidat il compare ses nombre de voisin , on vois que le nœud C possédé trois voisin (I,A,F) tandis que G a les nœuds (B,D,K,E,H) comme nœuds voisins donc il va choisit le nœuds C comme parent préféré.

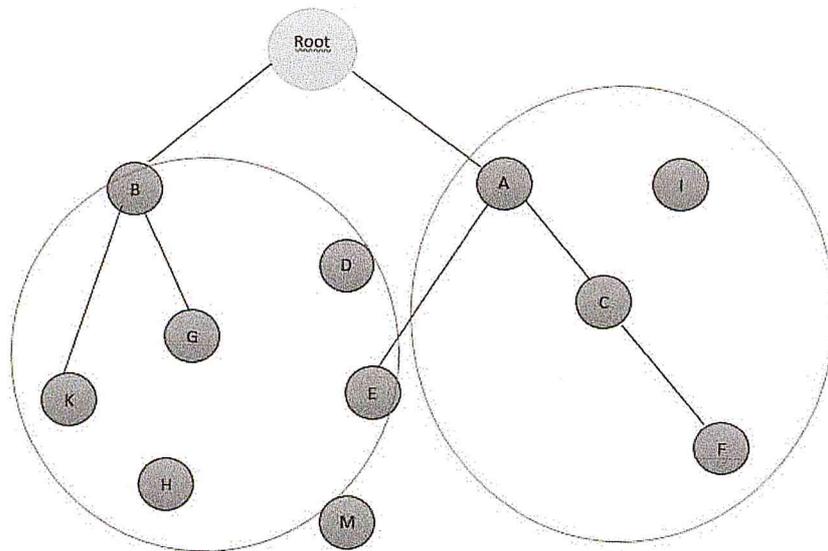


FIGURE 3.3 – Les voisins d'un nœud

Plus précisément, le parent ayant le moins nombre de voisins sera élu comme parent préféré. À cette fin, l'équilibre sera atteint en réduisant le nombre d'enfants du nœud surchargé.

La probabilité attendue qu'une transmission soit reçue avec succès et acquittée est  $df$  \* dr. Un expéditeur retransmettra un paquet qui n'a pas été acquitté avec succès.

### 3.5 La fonction objective proposé

RPL est un protocole de routage vectoriel de distance proactif conçu pour les LLN . Il construit un DODAG en utilisant une certaine fonction objective qui répond aux exigences de l'application. Pour résumer, RPL s'appuie sur un message de contrôle DODAG Information Object (DIO) pour créer le DODAG. Ainsi, le point de départ commence

lorsque le nœud racine (racine) diffuse le message DIO aux nœuds voisins comme indiqué sur la figure 3.4 (a). Dès que le nœud le plus proche reçoit le message, il peut décider de rejoindre ce DODAG . Si le nœud décide de se joindre, il ajoute l'expéditeur à la liste des parents candidats.

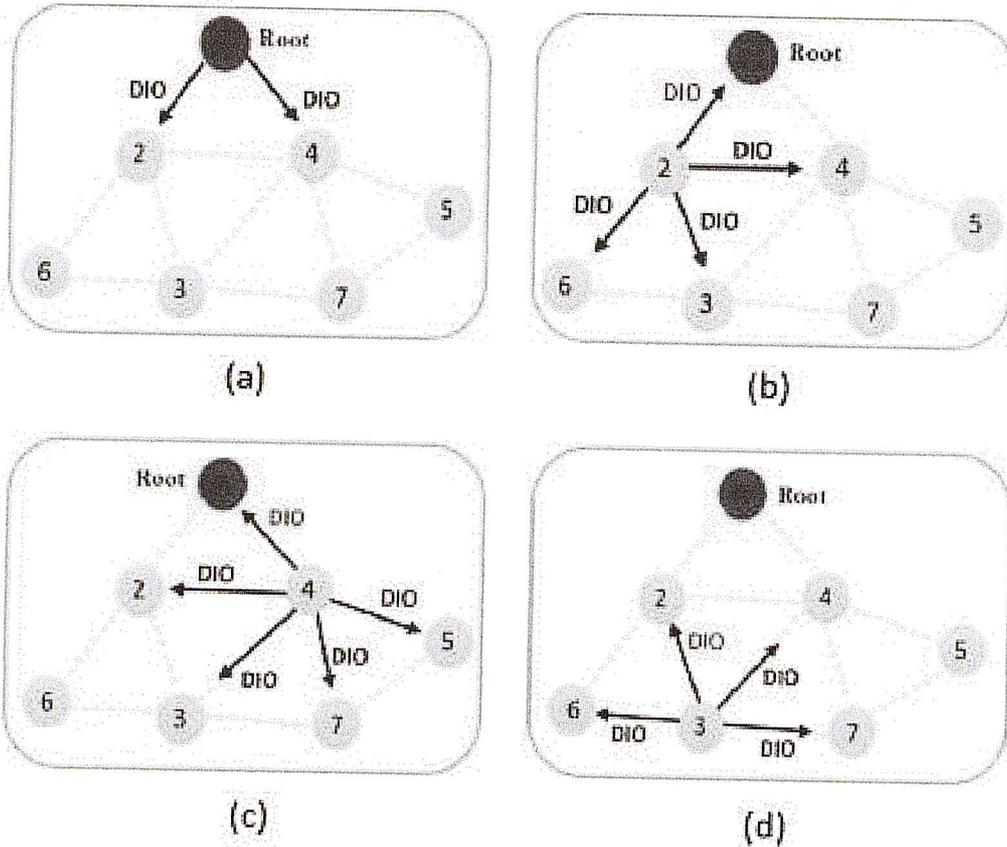


FIGURE 3.4 – La séquence de DIO dans la construction DoDAG dans RPL

Ensuite, le parent préféré, c'est-à-dire le saut suivant à la racine, sera choisi en fonction du rang de cette liste pour recevoir tous les trafics du nœud enfant. après, il calcule son propre classement en fonction de la fonction objective sélectionnée. La fonction objective sélectionnée définit la manière de calculer le rang . Dans les fonctions d'objectif standard le rang est calculé soit en fonction de nombre de transmissions prévu (ETX) avec la fonction objective MRHOF qui choisit le parent avec une meilleure qualité de lien , soit en fonction de nombre de saut avec OF0 qui minimise le nombre de saut vers la racine, mais les deux fonction objective donne un réseau non équilibré, pour résoudre ce problème nous proposons une nouvelle fonction objective qu'on a nommé LBCC Load Balancing Count Children ,elle équilibre le nombre de nœuds enfants et évite le problème de surcharge et assurer la maximisation de la durée de vie.

LBCC est basé sur la fonction objective MRHOF, elle prend en compte les deux metrique qu'on a étudié qui sont l'énergie consommé par le noeud ainsi que le nombre des voisins de noeud, elle combine ses deux métrique avec la métrique ETX donc le rang sera calculé comme suit :

$$Rank(N) = Rank(PN) + ETX(N, PN) + \alpha * NBV(PN) + \beta * EN\_CON(PN) \quad (3.2)$$

où  $Rank(PN)$  est la valeur de classement d'échelle du nœud parent PN,  $ETX(N, PN)$  affiche la valeur ETX du nœud N au nœud parent PN.  $NBV(PN)$  est le nombre de voisin de nœud parent PN.  $\alpha$  est un coefficient qui contrôle le poids donné au nombre de voisin.  $EN\_CON(PN)$  est l'énergie consommé du nœud parent PN et  $\beta$  correspond au poids attribué à  $EN\_CON(PN)$ ,  $\alpha$  et  $\beta$  sont des paramètres de conception.

Par conséquent, le nœud parent PN sera remplacé par le meilleur nœud parent alternatif PC si la condition suivante est remplie.

$$Rank(PC) < Rank(PN) \quad (3.3)$$

dans ce cas le parent ayant le moins nombre de voisin et la moindre consommation d'énergie sera élu comme parent préféré, donc la majorité des enfants choisissent un parent préféré en fonction du rang inférieur et à sûrement un nombre moins élevé d'enfants.

Après cela, le nœud propage son propre DIO avec toutes les informations mises à jour à ses voisins, comme indiqué sur la Figure 3.4 (b) et la même chose avec les autres nœuds de la figure 3.4 (c)(d). Le processus du schéma proposé est présenté à la Figure 3.5 sous forme de diagramme de flux et la description détaillée du processus est donnée comme suit :

**étape 1 :** Chaque nœud reconnaît ses nœuds voisins par des messages DIO reçus périodiquement de leur part, y compris des informations de rang, ETX, Energyconsommé et le nombre de voisins des nœuds.

**étape 2 :** s'il s'agit d'un DIO reçu pour la première fois, il ajoute l'expéditeur à la liste des parents.

**étape 3 :** Sinon, le nœud parent actuel sera modifié selon le rang inférieur basé sur LBCC.

**étape 4 :** le nœud améliore sa position et recalcule son propre Rank.

**étape 5 :** Après le processus de sélection du nœud parent, les données seront transmises en multidiffusion aux nœuds voisins.

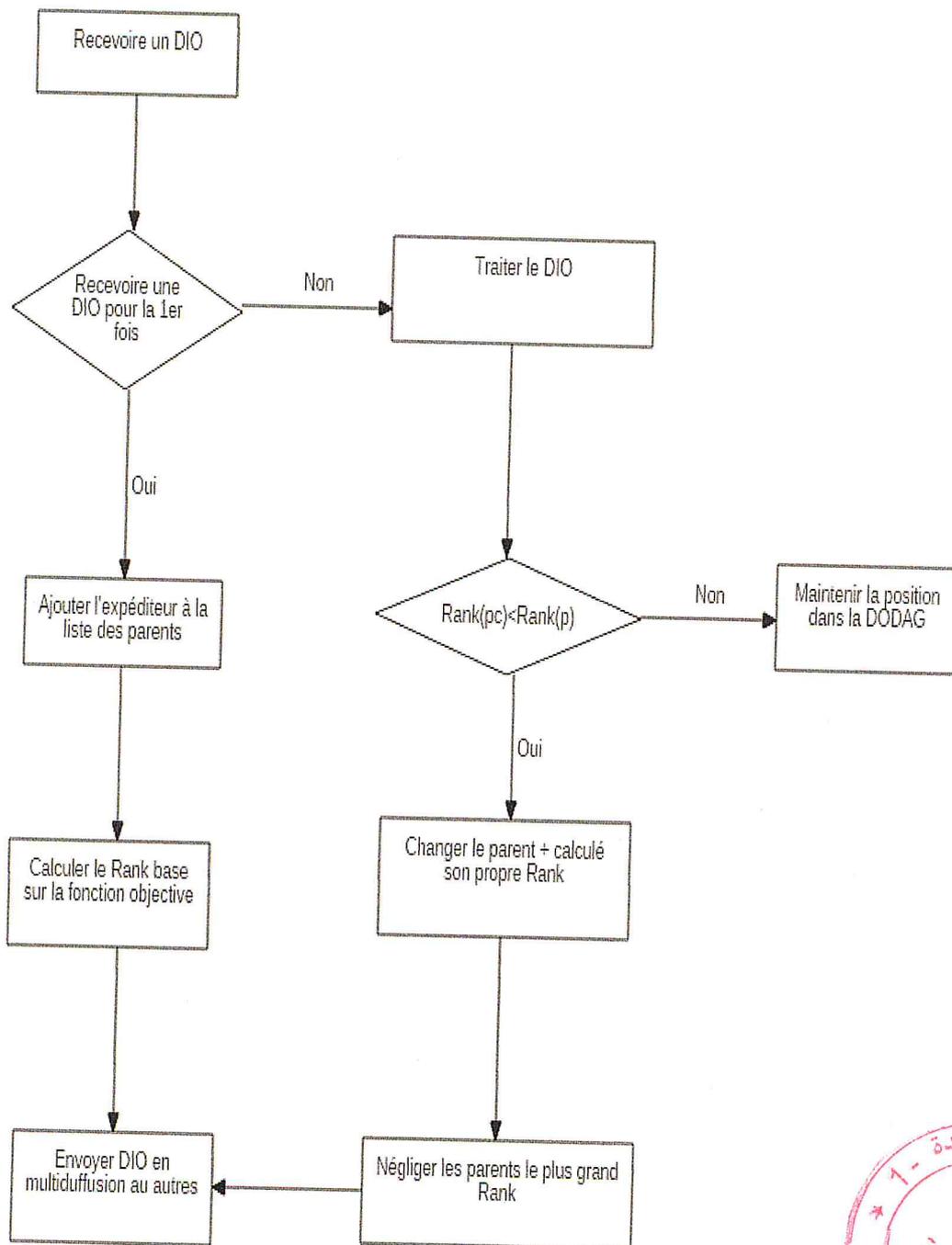
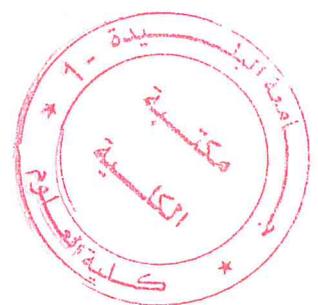


FIGURE 3.5 – Processus du shema proposé



### 3.5.1 message DIO modifié

Pour transporter des informations sur l'énergie consommé et nombre de voisin des noeuds , nous avons exploité le champ Options du paquet DIO conventionnel, comme indiqué sur la Fig3.6. ces champs nous permet de connaitre l'énergie consommé par le noeud parent candidat qui est sur 32 bits et le nombre de voisin qu'il possède chaque noeud qui est aussi sur 32 bits .

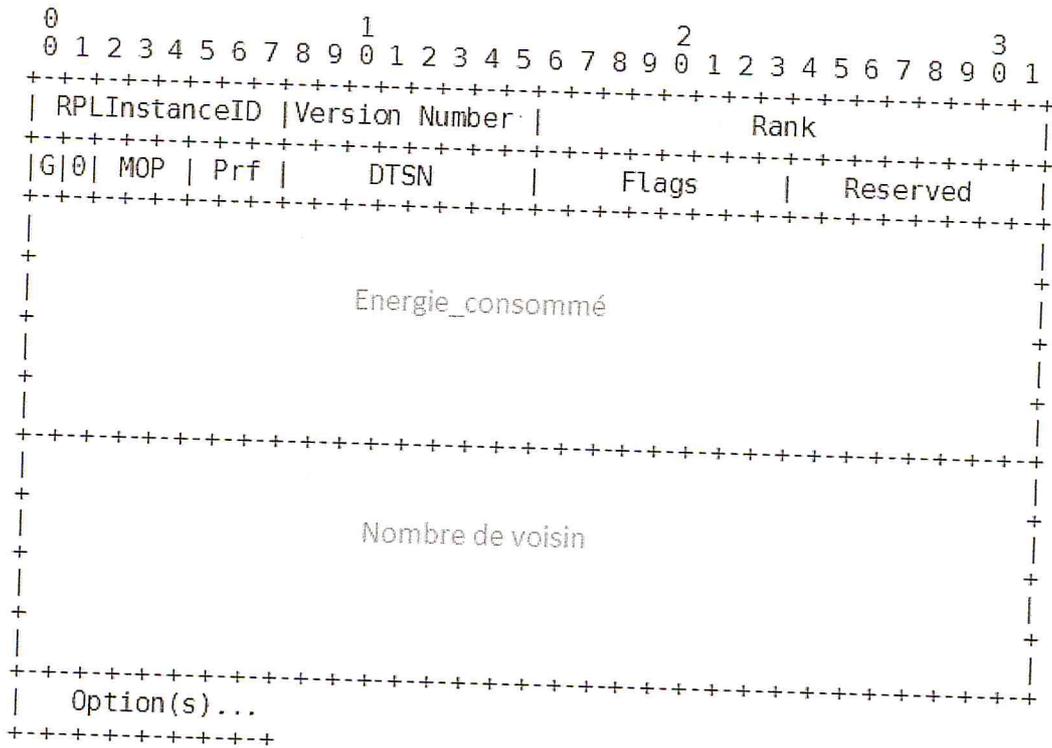


FIGURE 3.6 – message DIO modifié

## Conclusion

Dans ce chapitre nous avons proposé un mécanisme de sélection de noeud parent dynamique dans RPL pour l'équilibrage de charge prenant en compte à la fois l'énergie consommé par le noeud pour éviter l'épuisement de la batterie et le nombre de voisin pour éviter un parent ayant beaucoup de voisins ainsi le nombre de transmissions prévu (ETX), nous avons détaillé également le principe de fonctionnement du nouveau protocole.

Le prochain chapitre sera consacré aux détails de l'implémentation et l'évaluation de performances de la fonction objective LBCCOF sous le simulateur Cooja.

# Chapitre 4

## Évaluation des performances

### Introduction

Dans ce chapitre nous avons implémenter le protocole LBCC\_RPL qui assure l'équilibrage de charge dans le but d'améliorer les performances des application par rapport a RPL standard sous simulateur Cooja et système d'exploitation Contiki. a fin de valider et évaluer LBCC\_RPL, nous présentons d'abord les différents fichier concernés par les modifications . En suite, nous décrivons les modifications nécessaire du protocole RPL pour l'ajouts des nouvelles métriques qui sont énergie consommé et nombre des voisin, et la combinaison qui est la nouvelle fonction objective LBCCOF . aussi nous présentons les mesures de simulation qu'on a utilisé pour évalué ses performances . à la fin nous montrons les résultats de performance de cette nouvelle fonction objective en comparaisant avec la fonction objective OF0.

### 4.1 Système d'exploitation :

#### 4.1.1 Contiki OS

Contiki OS est un système d'exploitation open source léger conçu pour l'Internet des Objets . Il a été développé à l'Institut suédois des sciences de l'informatique par Adam Dunkels et écrit dans la langue de programmation C . Contiki est un système d'exploitation hautement portable et il a déjà été porté sur plusieurs plates-formes fonctionnant sur différents types de processeurs . La plupart des plates-formes utilisent le processeur Texas Instruments MSP-430 ainsi que la série de microcontrôleurs Atmel ATmega. Le principal avantage de Contiki est qu'il fonctionne sur un concept qui se situe entre le multi-threading et la programmation événementielle, cela permet aux processus de partager le même contexte d'exécution et donc d'améliorer l'utilisation de la mémoire et de l'énergie . C'est le concept des Protothreads.

Contiki prend en charge les implémentations de pile IPv6 et IPv4, ainsi que les normes sans fil peu avancées comme 6lowpan, RPL, CoAP ou encore la pile Rime . Ils agit d'une pile de communication légère pour les réseaux de capteurs et possède des couches plus petites que les piles traditionnelles.

Ce sont des couches simples qui ont de petits en-têtes (seulement quelques octets). Rime prend également en charge la réutilisation du code et le but principal de ce protocole est de simplifier la mise en œuvre des réseaux de capteurs.

Contiki est utilisé dans de nombreux systèmes tels que les compteurs électriques, la

surveillance de systèmes industriels, les feux de signalisation, les systèmes d'alarme et la domotique, la surveillance des rayonnements électromagnétique etc... La dernière version de Contiki OS est Contiki 3.0 (publié le 26.08.2015), cela fait donc quelque temps qu'il n'a pas évolué comparé aux possibilités du secteur.

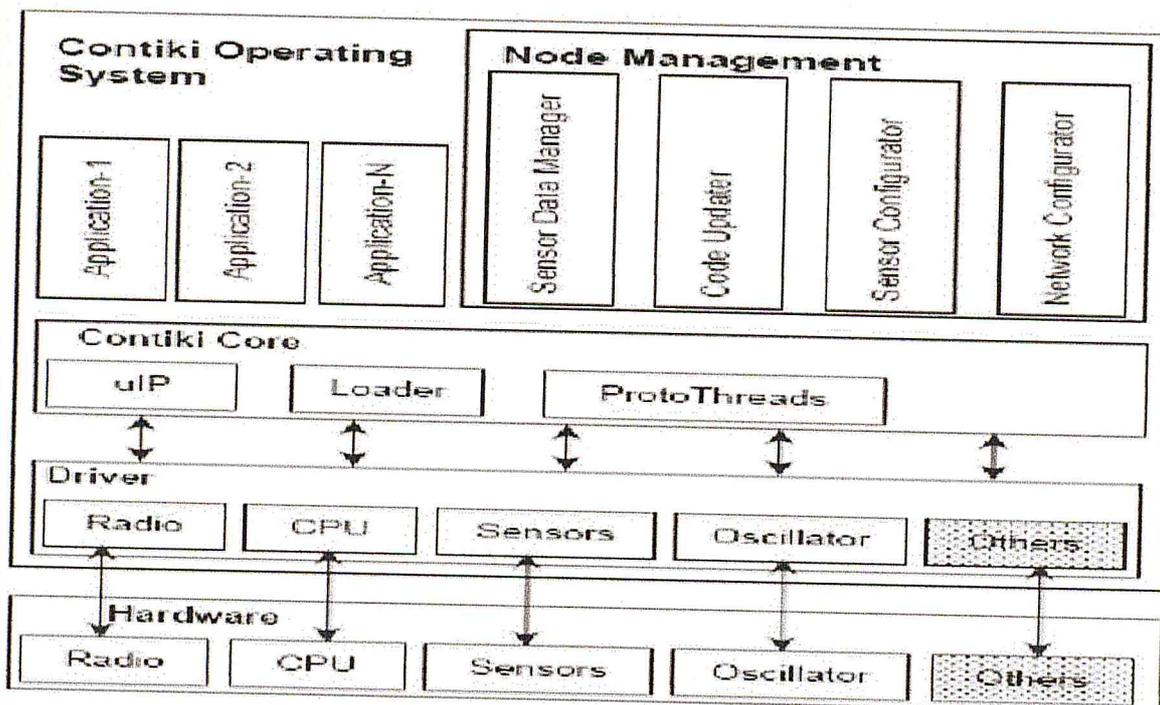


FIGURE 4.1 – Architecture de Contiki

#### 4.1.2 COOJA :

Cooja est un outil de Contiki, c'est un simulateur de réseaux de capteurs . C'est une application Java avec une interface graphique (GUI basée sur la trousse à outils Swing standard de Java). Cooja prend en charge la simulation du support radio et l'intégration avec les outils externes pour fournir des fonctionnalités supplémentaires à l'application. Il peut simuler de grands et petits réseaux de différents capteurs qu'on appelle nœuds sur lesquels on peut charger un système d'exploitation et des applications . Ces capteurs peuvent être émulés à des niveaux plus ou moins complexes. Cooja comporte deux logiciels émulateurs : Avrora et MSPSim . Cooja utilise Avrora, pour l'émulation des périphériques Atmel AVRbased et MSPSim pour l'émulation de périphériques TI MSP430 . La plupart des plates-formes ont des microcontrôleurs MSP430. C'est la raison pour laquelle MSPSim est le logiciel le plus utilisé pour la simulation de réseaux de capteurs sans fil . Cooja peut imiter plusieurs plates-formes comme : TelosB / SkyMote, Zolertia Z1 mote, Wismote, ESB, MicaZ mote . C'est un outil très utile pour le développement et le débogage d'applications Contiki OS. Il permet aux développeurs de tester leur code et leurs systèmes avant de l'exécuter sur le matériel cible réel, d'estimer les consommations d'énergie des nœuds dans les simulations ou de voir les transmissions radio et les réceptions.

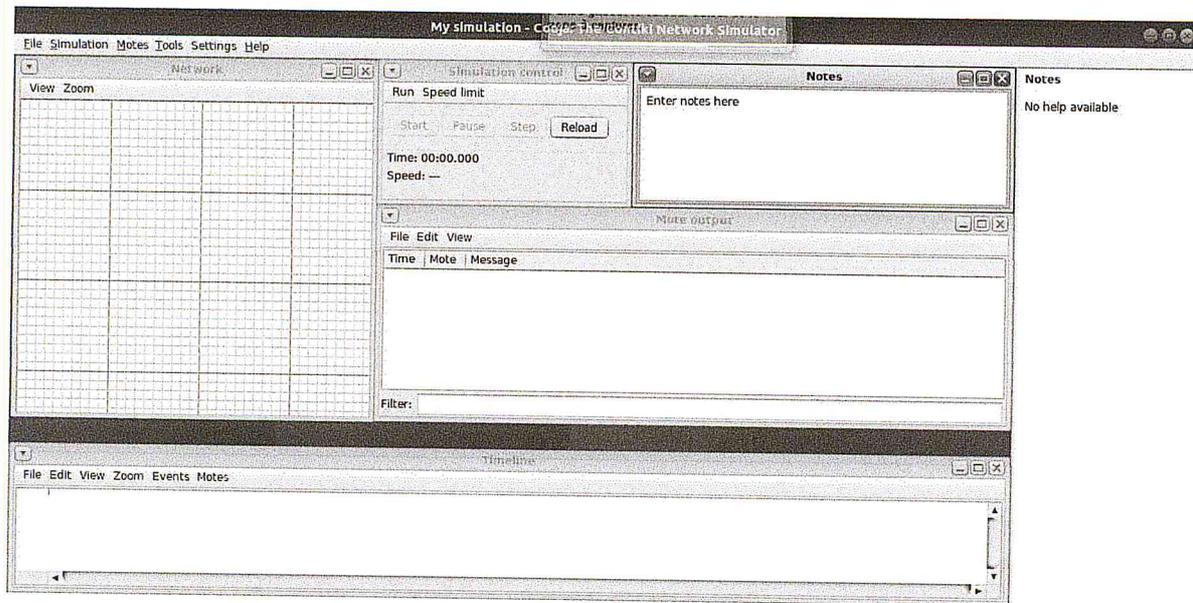


FIGURE 4.2 – L'interface de simulation

## 4.2 Implémentation LbCCRPL :

Contiki OS implémente une version assez stable de RPL. Le code source du routage RPL se trouve dans `/core/net/rpl` c'est dans ce module qu'on va consacrer notre étude car c'est là où on va implémenter notre travail. Pour ce faire, nous avons modifié dans ses fichiers `rpl.h`, `rpl-icmp6.c`, `rpl-mrhof.c`

1. **rpl.h** dans ce fichier nous avons déclaré les deux nouvelles métriques énergie consommé et nombre de voisin qui sont les métriques (9) et (10) comme elles sont présentées dans la figure 4.3 .

```

55 /* DAG Metric Container Object Types, to be confirmed by IANA. */
56 #define RPL_DAG_MC_NONE          0 /* Local identifier for empty MC */
57 #define RPL_DAG_MC_NSA           1 /* Node State and Attributes */
58 #define RPL_DAG_MC_ENERGY        2 /* Node Energy */
59 #define RPL_DAG_MC_HOPOCOUNT    3 /* Hop Count */
60 #define RPL_DAG_MC_THROUGHPUT    4 /* Throughput */
61 #define RPL_DAG_MC_LATENCY       5 /* Latency */
62 #define RPL_DAG_MC_LQL           6 /* Link Quality Level */
63 #define RPL_DAG_MC_ETX           7 /* Expected Transmission Count */
64 #define RPL_DAG_MC_IC            8 /* Link Color */
65 #define RPL_DAG_MC_TOTALENERGY   9 /* Node Total Energy Consumption*/
66 #define RPL_DAG_MC_NEIGHBORS    10 /*neighbors*/
67

```

FIGURE 4.3 – Déclaration des métriques

chaque nœud diffuse ces métriques énergie consommé et nombre des voisins de chemin au autres nœuds, ces deux informations se trouvent dans la métrique container pour cela nous avons ajouté les métriques énergie consommé et nombre des voisins dans le conteneur de métriques pour qu'il soient diffusées et utilisées au lieu ETX.

```

105 union metric_object {
106     struct rpl_metric_object_energy energy;
107     uint16_t etx;
108     uint16_t total_energy; // ajouter l'energie consommé dans la conteneur de métrique
109     uint16_t neighbors; // ajouter nombre de voisin dans la conteneur de métrique
110 } obj;
111 };

```

FIGURE 4.4 – Conteneur des métriques

2. **rpl-icmp6.c** : les messages de contrôle DIS, DIO, DAO sont implémentés dans ce fichier, il existe une fonction de l'envoi qui est `dio_output` responsable d'envoyer et une autre fonction `dio_input` responsable de la réception, pour notre cas nous intéressons d'envoyer des données de l'énergie et nombre de voisins, nous récupère les nouveaux valeurs l'énergie et nombre de voisins qui calculé dans fonction objective `mrhof`, et sauvegarde dans le buffer.

```

515 /*-----*/
516 } else if (instance->mc.type == RPL_DAG_MC_TOTALENERGY) {
517     buffer[pos++] = 2;
518     set16(buffer, pos, instance->mc.obj.total_energy);
519     PRINTF("DIO Output total_energy %u\n", instance->mc.obj.total_energy);
520     pos += 2;
521 }
522 /*-----*/
523 /*-----*/
524 } else if (instance->mc.type == RPL_DAG_MC_NEIGHBORS) {
525     buffer[pos++] = 2;
526     set16(buffer, pos, instance->mc.obj.neighbors);
527     PRINTF("DIO Output neighbors %u\n", instance->mc.obj.neighbors);
528     pos += 2;
529 }
530 /*-----*/

```

FIGURE 4.5 – Energie consommé et nombre de voisin dans `dio_buffer`

dans `dio_inpt` nous récupère les valeurs l'énergie et nombre de voisins qui contient dans le message DIO :

```

341 /*-----*/
342 } else if (dio.mc.type == RPL_DAG_MC_TOTALENERGY) { // si nous trouver par metrique energie consommé
343     dio.mc.obj.total_energy = get16(buffer, 4);
344     PRINTF("RPL: DAG MC: type %u, flags %u, aggr %u, prec %u, length %u, TotalEnergy %u\n",
345         (unsigned)dio.mc.type,
346         (unsigned)dio.mc.flags,
347         (unsigned)dio.mc.aggr,
348         (unsigned)dio.mc.prec,
349         (unsigned)dio.mc.length,
350         (unsigned)dio.mc.obj.total_energy);
351 }
352 /*-----*/
353 } else if (dio.mc.type == RPL_DAG_MC_NEIGHBORS) { // si nous trouver par metrique nombre des voisins
354     PRINTF("RPL: DAG MC: type %u, flags %u, aggr %u, prec %u, length %u, TotalEnergy %u\n",
355         (unsigned)dio.mc.type,
356         (unsigned)dio.mc.flags,
357         (unsigned)dio.mc.aggr,
358         (unsigned)dio.mc.prec,
359         (unsigned)dio.mc.length,
360         (unsigned)dio.mc.obj.neighbors);
361 }
362 /*-----*/

```

FIGURE 4.6 – Energie consommé et nombre de voisin dans `dio_buffer`

3. **rpl-mrhof.c** MRHOF fonctionne avec des métriques qui sont additifs le long d'une route et la métrique utilisée est déterminée par les métriques annoncé par les messages DIO, puisque la nouvelle fonction objective que nous avons proposé est multi-parent c-à-d quand un nœud choisit un parent préféré, il prendre en considération le père du père du candidat. Dans MRHOF, on a implémenté la nouvelle fonction objective qui combine les trois métriques qui sont Nombre des voisin et énergie consommé et ETX. nous choisirons la métrique ETX pour applique OF quand utiliser la fonction `neighbor_linkcall_back` qui calcule la somme de ETX de nœud instance et leur nœud parent.

```

100 |
101 | #if RPL_DAG_MC == RPL_DAG_MC_NONE
102 |     return p->rank + (uint16_t)p->link_metric;
103 |
104 | #elif RPL_DAG_MC == RPL_DAG_MC_ETX
105 |
106 |     return p->mc.obj.etx + p->rank + (uint16_t)p->link_metric; // combineur entre le Rank et ETX
107 |
108 | #elif RPL_DAG_MC == RPL_DAG_MC_ENERGY

```

FIGURE 4.7 – combine entre rank et ETX

Pour diffuser la nouvelle fonction objective nous utilisons le Rank pour cela nous avons modifié la fonction `calculate_rank`, qui calcule le nouveau Rank en fonction de ETX, energie consommé et nombre des voisin.

```

166 |
167 | if(p == NULL) {
168 |     if(base_rank == 0) {
169 |         return INFINITE_RANK;
170 |     }
171 |     rank_increase = RPL_INIT_LINK_METRIC * RPL_DAG_MC_ETX_DIVISOR;
172 | } else {
173 |
174 |     #if RPL_DAG_MC == RPL_DAG_MC_ETX //pour calculer le nouveau Rank
175 |
176 |     rank_increase = Neighbors_BETA * uip_ds6_nbr_num() + Neighbors_ALPHA * get_total_energy_consumption();
177 |

```

FIGURE 4.8 – Modification le Rank

### 4.3 Métriques de simulation

Les métriques sont des paramètres de test du protocole de routage qui permettent de mesurer les performances de celui-ci. Dans notre étude, nous avons pris en compte les métriques suivantes :

**Taux de perte :** représente le rapport entre le nombre de paquets de données perdus et la somme de paquets livrés et le nombre de paquets perdus :

$$\frac{\text{Nombre\_paquet\_perdu}}{\text{Nombre\_paquet\_mis}} \quad (4.1)$$

**Energie :** représente l'énergie consommée par les nœuds parent.

**la charge de noeud :** Total nombre d'enfant de chaque parent par rapport à nombre total des parent

$$\frac{\text{Nombre\_des\_enfants}}{\text{Nombre\_total\_des\_parents}} \quad (4.2)$$

### 4.4 Evolution

Pour évaluer notre protocole, nous avons simulé LBCC\_RPL avec Cooja Simulator, un simulateur largement utilisé dans le domaine de l'IoT développé par Contiki . Nous avons comparer LBCC\_RPL avec ContikiRPL. on a défini OF0 comme OF de contikiRPL, et les paramètres du protocole sont ajustés conformément à ceux de LBCCOF (Tableau 4.1).

TABLE 4.1 – Paramètres de simulation

Paramètre	Valeur
Simulateur	Cooja
Temps de simulation	30 min
Surface	300x300 mètres
Nombre de nœud	1 Sink,30 sender
Fonction objective	RPL, LBCCOF
Modèle de rapport(Ratio model)	UDGM
Type de nœud	Cooja mote
Topologie	Random
Rang de transmission	50 mètres
ETX_SCALE	100
ETX_ALPHA	90

Dans notre scénario, nous avons 30 nœuds avec un routeur frontière, déployés dans une zone de 300 m à 300 m pour une topologie aléatoire . Le routeur frontière agit comme une racine. Le type de nœuds est Sky avec micro-contrôleur MSP430, avec émetteur-récepteur sans fil IEEE 802.15.4 à puce 2,4 GHz, et suit le modèle de consommation d'énergie de l'émetteur-récepteur CC2420. Ils utilisent Contiki OS. comme système d'exploitation et le protocole de communication IEEE 802.15.4 pour communiquer entre eux . UDGM3 de Cooja utilisé avec perte de distance, pour RadioMedium . Le nœud récepteur est situé devant les nœuds, comme indiqué à la Fig. 7. Pour introduire la perte dans le support sans fil, nous avons utilisé le modèle de canal Cooja Unit Disk Graph Medium (*UDGM*), qui applique le facteur de perte relatif aux distances relatives des nœuds. dans le milieu radio . Les chiffres de la figure indiquent le taux de réception des paquets (RX) en unité de pourcentage%, indiquant les pertes du support sans fil, ce qui signifie que la réception des paquets des nœuds augmente à mesure que la distance entre le nœud récepteur et le nœud émetteur diminue . Nous avons utilisé design paramètres  $\alpha$  et  $\beta$  tout au long de notre simulation .  $\alpha$  est le poids donné à NBV et  $\beta$  est le poids donné à l'énergie consommé  $EN_{CON}$  .  $\alpha$  et  $\beta$  devrait être supérieur à 1 pour avoir un effet notable sur la sélection du nœud parent . Sinon, NBV et  $EN_{CON}$  auraient un effet plus petit que ETX.nous donnons les valeurs de  $\alpha = 3$  et  $\beta = 2$  . Nous avons utilisé l'intervalle d'envoi de paquets a 30 s.Nous avons calculé la consommation d'énergie à l'aide des sorties de suivi de l'alimentation[?] , notamment le temps de transmission, le temps d'écoute, le temps CPU et le temps LPM. L'énergie consommée au cours des états de transmission, *d'écoute/rception*, de l'UC et du LPM était respectivement de 0,0017944 mJ, 0,00199 mJ, 0,00016 mJ et 0,000004 mJ. Pour calculer l'énergie consommée par les particules célestes, nous avons ajouté dans le fichier actuel *sys/energest.c* la fonction `get_total_energy_consumption()` pour calculé l'énergie consommé par nœud

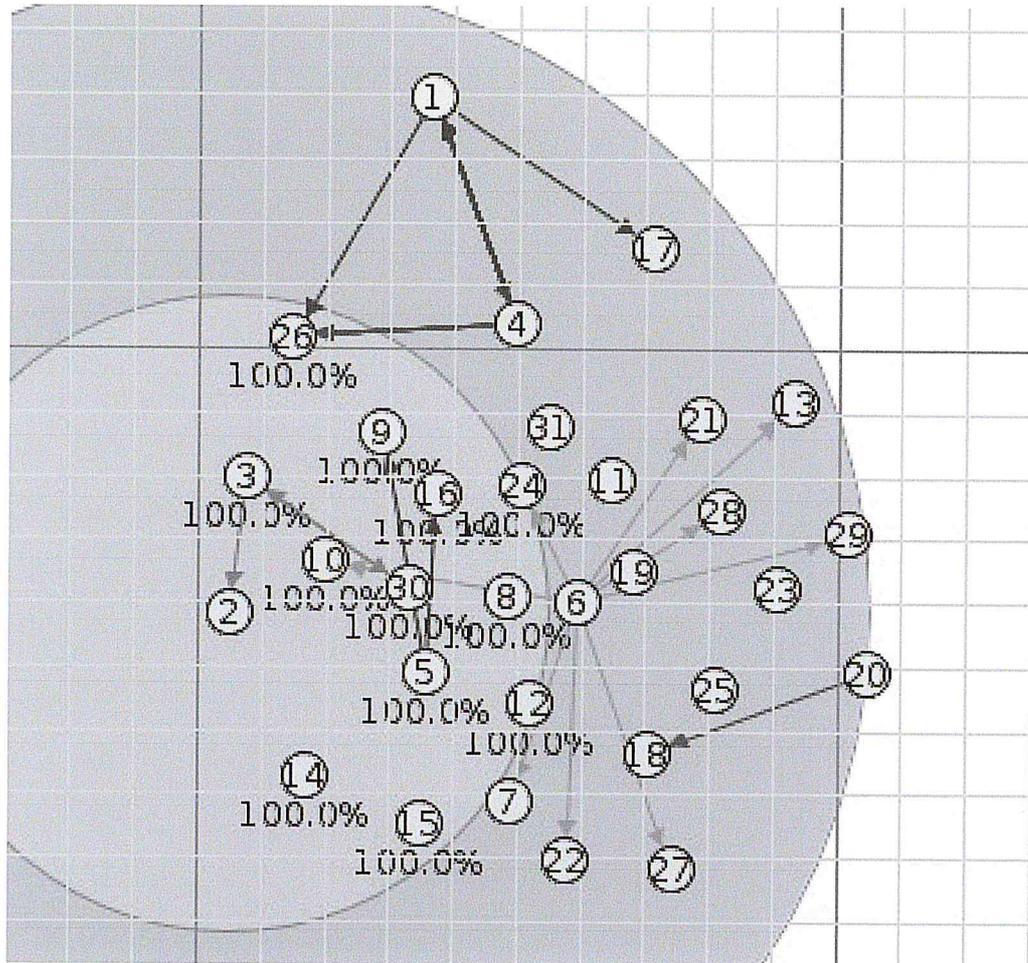


FIGURE 4.9 – Scénario de simulation.

## 4.5 RÉSULTATS ET DISCUSSION

La première comparaison est faite du point de vue de la charge au niveau des parents. Le graphe dans figure 4.10 montre la topologie du réseau en cas d'utilisation de RPL. Le numéro de chaque nœud y est écrit. Dans cette figure, quelques nœuds tels que les nœuds 26 ou 4 doivent gérer un grand nombre de nœuds enfants lors du transfert de paquets, tandis que d'autres ne subissent pas une charge aussi lourde pendant la durée de vie du réseau. Ainsi, le déséquilibre de l'arbre RPL entraîne des encombrements et des pertes. par contre dans LBCC\_RPL protocole, les nœuds qui étaient auparavant sous forte charge, semblent être dans un meilleur état dans LBCC\_RPL . par conséquent, une meilleure construction de DODAG.

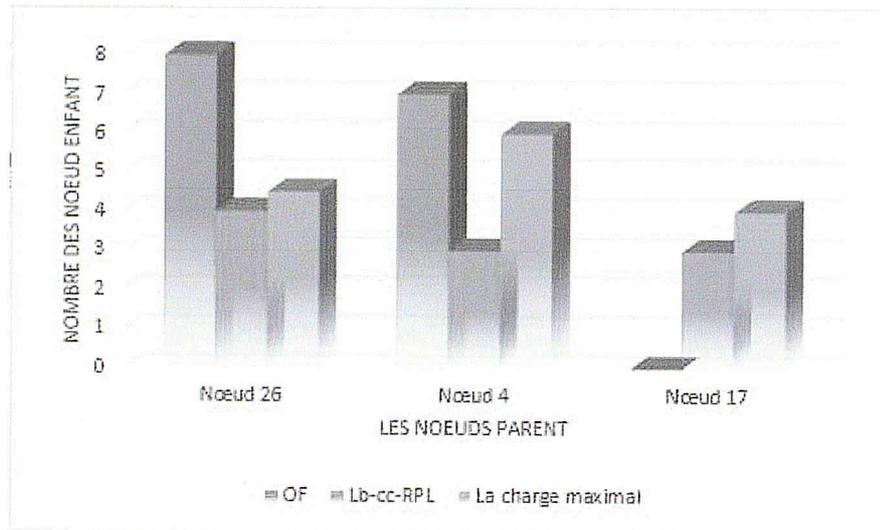


FIGURE 4.10 – la charge des nœuds parents

Nous avons calculée aussi la consommation d'énergie par nœud parent pour LBCC\_RPL et nous l'avons comparée à la fonction objective OF0 du RPL standard . Comme le montre la figure , Le résultat montre que les nœuds parent dans LBCC\_RPL ont consommées mois d'énergie par rapport à OF0 .donc une duré de vie de réseau LBCC\_RPL grande par rapport à OF0 .

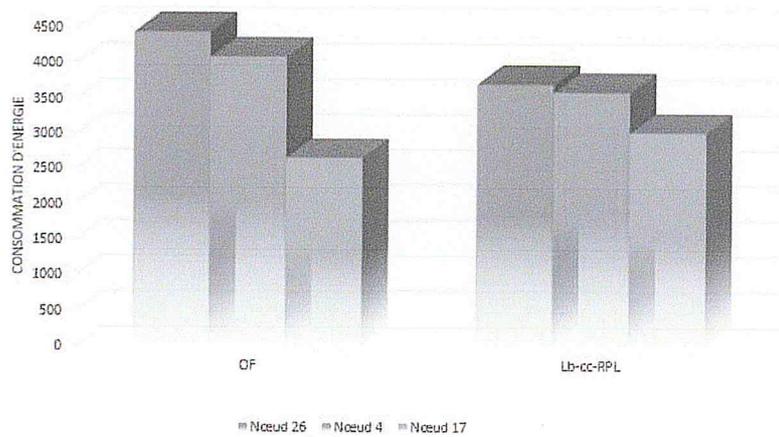


FIGURE 4.11 – La charge maximal vs nombre des nœuds enfant dans un nœud parent.

D'après la figure 12, nous remarquons que la perte des paquet dans OF0 et LBCC\_RPL est presque la même , C'est à cause du changement de parent dans LBCC\_RPL après chaque 2 minute de simulation mais par contre OF0 garde le même parent dans tous le temps de simulation.

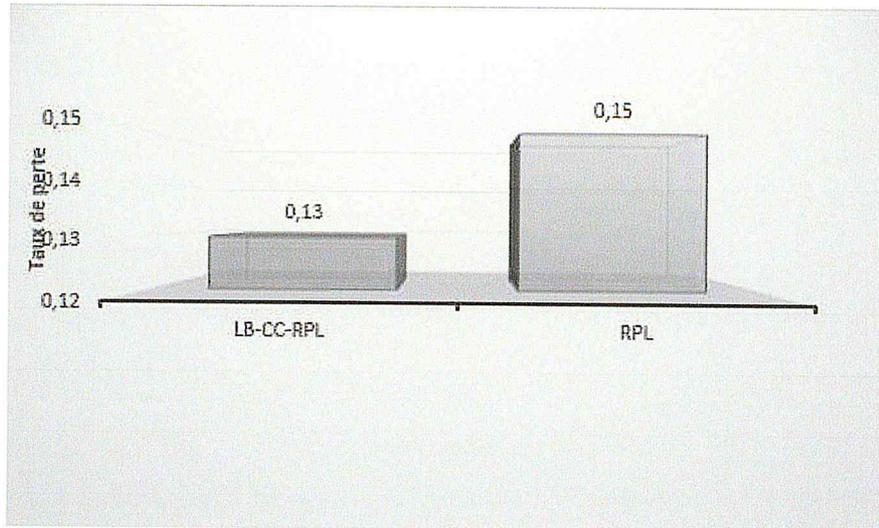


FIGURE 4.12 – La perte des paquets au niveau des nœuds parents

Dans la figure 4.13, Notez que la consommation d'énergie moyenne dans l'oscillateur de protocole LBCC\_RPL est due au changement des parents. Mais dans l'ensemble, on constate une diminution de la consommation par rapport à RPL

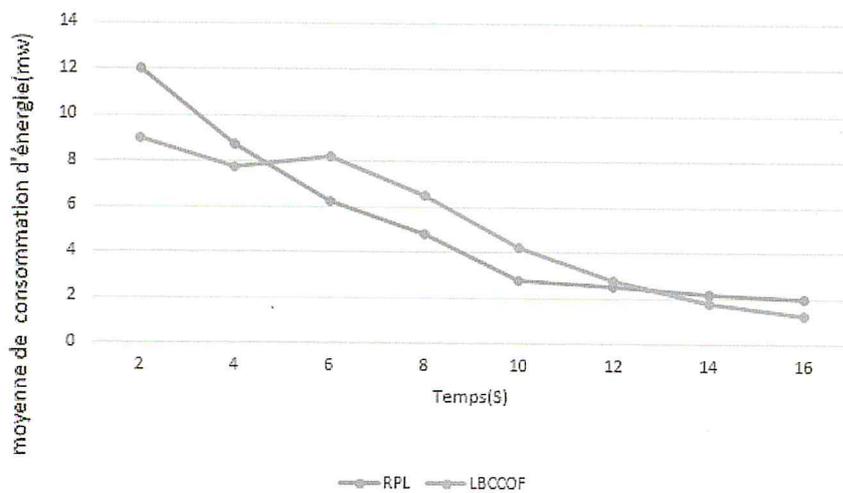


FIGURE 4.13 – La moyenne de consommation d'énergie vs temps(s)

---

## Conclusion

Dans ce chapitre, nous avons présenté l'évaluation du protocole RPL avec une combinaison de trois métriques de routage : ETX et l'énergie consommé par nœud et le nombre des voisin et on a comparés ses résultats avec le RPL standards qui travaille avec OF0

Les résultats obtenus montrent une grande amélioration dans l'énergie consommée par le réseau, le taux de perte de paquets et bon répartition des enfant entre les parents par rapport a RPL ,par conséquent la nouvelle fonction objective améliore le protocole RPL et a réalisé l'équilibrage de charge.

# Chapitre 5

## Annexes

### 5.1 Installation de Contiki

Contiki est une source de code ouvert et il marche directement sur les systèmes de Linux. Pour mettre en place contiki sur une machine, il y a deux façons :

- Première façon : télécharger une version Instant Contiki qui est situé dans le site, [http://sourceforge.net/projects/contiki/files/Instant Contiki/](http://sourceforge.net/projects/contiki/files/Instant%20Contiki/). En effet, c'est une version du système de l'exploitation Ubuntu, qui comprend le répertoire de contiki, dedans. Cette façon d'installation est compatible avec une machine de Windows.
- Deuxième façon : utiliser l'outil Git pour synchroniser la dernière version du répertoire de contiki depuis le site <https://github.com/contiki-os/contiki> vers ton ordinateur. Cette façon d'installation est compatible avec une machine de Linux.

### 5.2 Le simulateur Cooja

Le simulateur Cooja est situé dans le dossier `contiki-2.7 / tools / cooja`. Il utilise une combinaison de code Java pour l'interface frontale et les émulateurs spécifiques à la plateforme pour effectuer les simulations.

### 5.3 Démarrage de Cooja

Vous pouvez démarrer le simulateur Cooja avec les commandes suivantes :

1. `cd contiki-2.7/tools/cooja`
2. `ant run`

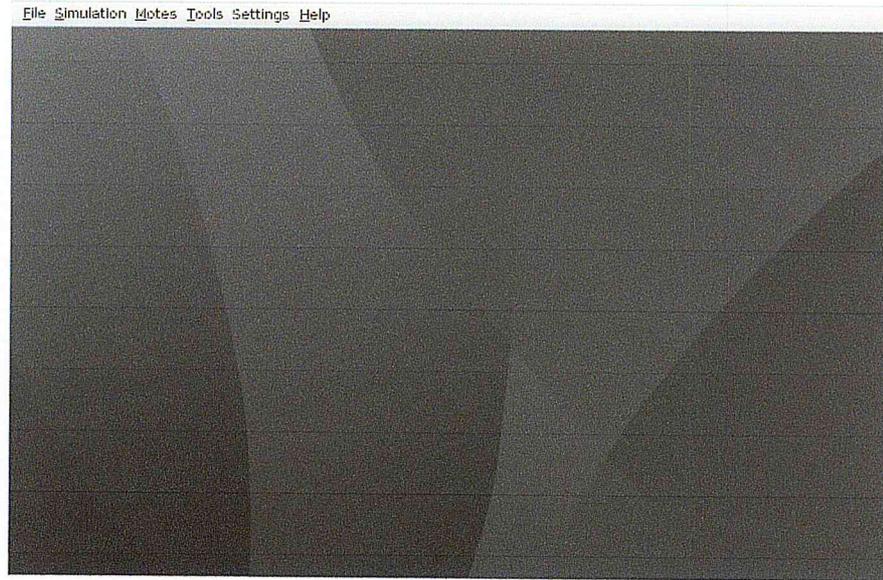


FIGURE 5.1 – L'interface de bureau Cooja

Vous verrez beaucoup de texte passer au moment où l'environnement Cooja est compilé, construit et démarré. Finalement, vous verrez maintenant l'interface de bureau Cooja, comme le montre la figure 5.1.

## 5.4 Création d'une simulation

Tout d'abord, vous devrez créer une nouvelle simulation. Pour ce faire, cliquez sur "Fichier>Nouvelle simulation". Cela présentera une boîte de dialogue, illustrée à la figure 5.2. Donnez à votre simulation un titre plus approprié, puis sélectionnez le support radio

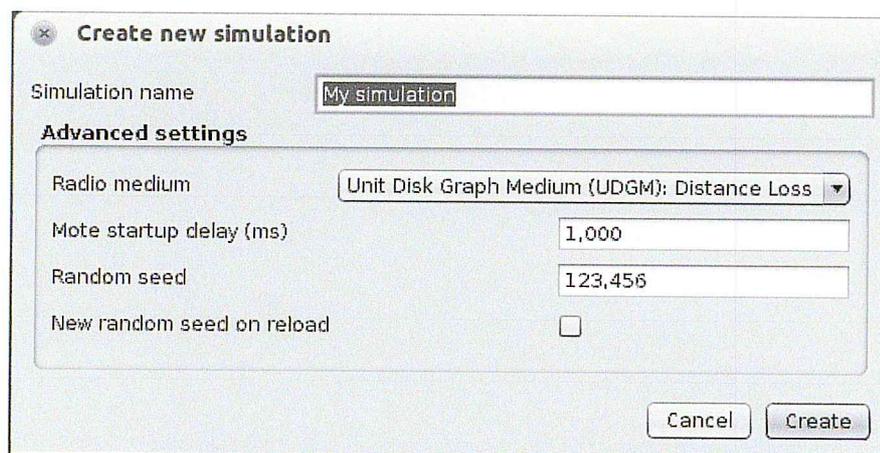


FIGURE 5.2 – Créer une nouvelle interface de simulation.

qui convient le mieux à votre type de simulation. Pour la plupart des simulations, Unit Disk Graph Medium (UDGM) est tout à fait approprié. Désactiver la simulation radio en sélectionnant "No Radio Traffic" ("Pas de trafic radio"), lorsqu'il n'est pas nécessaire, enregistre le temps CPU et rend la simulation plus rapide. Le démarrage aléatoire retarde le démarrage de chaque nœud simulé au hasard, de sorte qu'ils ne commencent pas tout à

la fois exactement. La graine aléatoire (random seed) est une graine pour le générateur de nombres aléatoires. Vous pouvez cocher la case à la fin pour obtenir une graine aléatoire. Lorsque vous êtes prêt, cliquez sur "Créer". Vous allez maintenant voir le bureau Cooja une fois de plus, mais avec l'environnement de simulation présenté, comme le montre la figure 5.3.

## 5.5 L'interface de simulation

L'interface de simulation, illustrée à la figure 5.4, se compose de cinq fenêtres. La

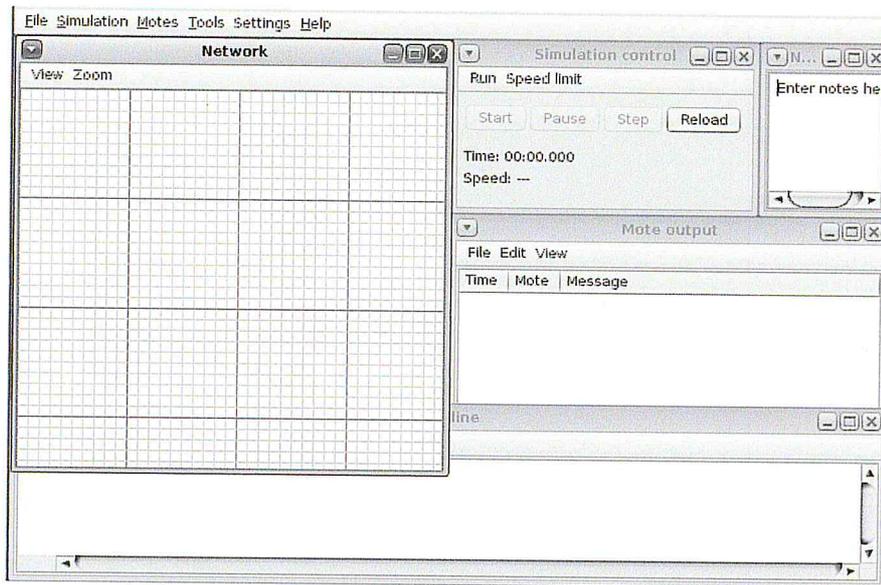


FIGURE 5.3 – L'interface de simulation.

fenêtre Réseau affiche la disposition physique du réseau, c'est-à-dire que vous pourrez placer des nœuds ici et les déplacer, au besoin, afin de former la topologie et la mise en page qui vous intéressent.

- La fenêtre de contrôle de simulation vous permet de démarrer, Arrêtez et rechargez la simulation. Cela vous permet également de contrôler le taux de progression de la simulation.
- La fenêtre Mote Output montre toute sortie série générée par toutes les nœuds, c'est-à-dire la sortie de la commande printf. Vous pouvez filtrer la sortie affichée en fonction de la chaîne que vous entrez dans le champ "Filtre". Par exemple, si vous souhaitez filtrer la sortie de sorte qu'elle affiche uniquement la sortie de nœud 2, vous pouvez entrer "ID :2" dans ce champ.
- La fenêtre Timeline affiche les événements qui se produisent sur chaque nœud au cours de la chronologie de la simulation. Ces événements peuvent être le trafic radio, l'activité LED ou toute autre chose.
- La fenêtre Notes peut être utilisée pour prendre des notes temporaires dans la simulation.

## 5.6 Définition des types de nœuds

La prochaine étape consiste à définir les types de nœuds. Dans Cooja, vous créez une liste de types de nœud, définissez leurs paramètres et attribuez leur code de programme /binaire. Vous alors, en tant que tâche séparée, créez des instances pour simuler. Nous abordons ici la première partie : créer les types de nœuds. Cliquez sur "Motes > Add Motes > Create New Mote Type > cooja mote.". Donnez au nœud une description utile. voir la figure 5.4.

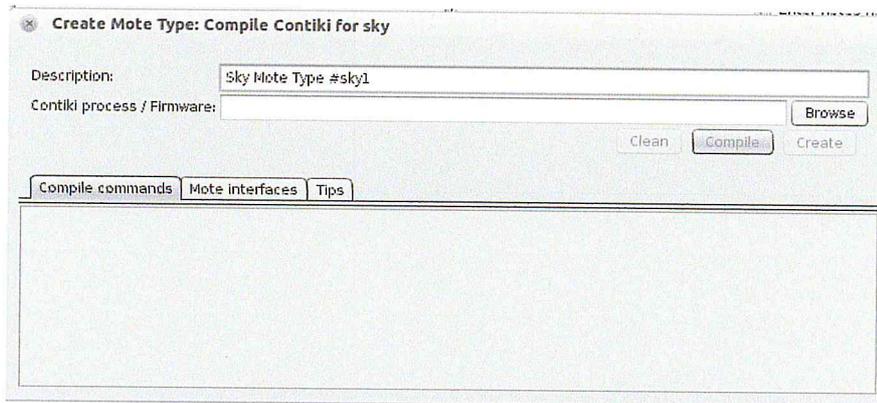


FIGURE 5.4 – L'interface créer le type de nœud

Dans le champ "Contiki Process / Firmware", vous devez spécifier votre fichier source (le fichier .c) ou le fichier binaire (le fichier .sky). Si vous spécifiez le fichier binaire, vous ne verrez aucune instruction de compilation (les fichiers binaires sont le résultat de la compilation - il n'est pas nécessaire de compiler). Si vous spécifiez le code source, le champ de commandes de compilation devient actif et vous pouvez spécifier des instructions spécifiques pour la compilation. pour nous, nous allons utiliser le bouton Browse (Parcourir) pour sélectionner le fichier udp-sink.c, après nous allons cliquer sur Compiler pour pouvoir créer le type de nœud. Vous pouvez voir la sortie / résultat de la compilation dans l'onglet de sortie de compilation. En cas de succès, le bouton Créer est disponible pour créer vos nœuds. Dans le cas, où vous utilisez un binaire pré-compilé (le fichier .sky), le bouton Créer est déjà disponible sans compiler. De meme, vous pouvez créer le client UDP en utilisant le fichier udp-sender.c.

## 5.7 Ajout de nœuds (Motes)

Maintenant que vous avez créé les types de nœuds udp-sink et udp-sender, nous pouvons commencer à ajouter des nœuds physiques à la simulation. Le dialogue "Add motes" vous permet de définir le nombre et la configuration de ces nouvelles actions. Vous pouvez accéder à cela en cliquant sur "Motes > Add Motes". Ajoutez un nœud du type udp-sink et cinq nœuds de type udp-sender par exemple. Vous pouvez utiliser le positionnement aléatoire pour ajouter les nœuds. Une fois que vous faites cela, vous constaterez qu'un total de six nœuds placés au hasard apparaîtront dans la fenêtre Réseau. Un arrangement aléatoire possible peut être vu à la figure 5.7 (a). Parmi ceux-ci, le nœud 1 est le serveur et le reste sont des nœuds qui exécuteront le code du serveur UDP. Vous pouvez interagir avec chacune des nœuds en cliquant et en les faisant glisser. Vous pouvez réorganiser la disposition des nœuds en les faisant glisser autour, de sorte qu'ils formeront une topologie

qui vous intéresse. Lorsque vous cliquez sur un nœud, il affiche également l'environnement radio en couleur verte et grise. Le cercle vert représente la zone dans laquelle le signal a été reçu avec succès par d'autres nœuds et le gris représente la zone où l'interférence radio du nœud actuel. Un exemple de cet environnement radio et une nouvelle mise en page telle que seul un maximum de deux nœuds se situent dans la portée de chaque nœud, peut être vu à la figure 5.7(b).

FIGURE 5.5 – a.

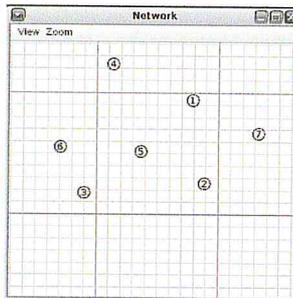


FIGURE 5.6 – b.

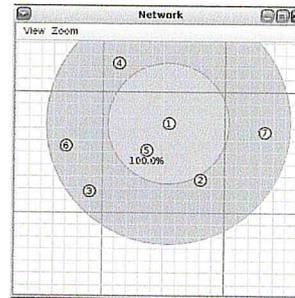


FIGURE 5.7 – La fenêtre du réseau.

## 5.8 Conclusion

Vous savez maintenant comment travailler avec le simulateur Contiki Cooja.

# Conclusion générale

Les réseaux à faible consommation d'énergie et à perte ont un large potentiel et constituent un sujet de recherche innovant ainsi qu'un outil convoité par plusieurs domaines. C'est sans aucun doute, une technologie qui va nous accompagner pour les prochaines années et ainsi faire partie de notre vie quotidienne . Cependant, il y a encore beaucoup de problèmes qui doivent être abordés pour un fonctionnement efficace de ces réseaux dans des applications réelles . Parmi les problèmes fondamentaux et importants dans ces réseaux nous citons la problématique de l'économie de l'énergie qui est une nécessité absolue à laquelle des solutions adéquates doivent être proposées . Ce mémoire est centré sur l'équilibrage de charge des noeuds parents qui est une nécessité absolue à cause de l'épuisement de ces derniers , ce qui cause la perte de réseaux , sachant que la communication entre les noeuds constitue l'opération la plus consommatrice d'énergie donc toute limitation de la communication entre les noeuds sera sûrement bénéfique.

Pour résoudre ce problème de charge dans les noeuds parents, nous avons proposé une stratégie d'optimisation de routage qui est une combinaison des métriques appelé LBCCOF ou une extension de protocole RPL nommé LBCC\_RPL , ces métriques qui sont l'énergie consommée et le nombre de voisin des noeuds, pour l'énergie consommée, les noeuds choisissent un parent préféré selon sa consommation d'énergie , le parent qui a consommé moins d'énergie sera choisi comme parent préféré, et pour la métrique nombre des voisins, le noeud choisit un parent préféré ayant moins de voisins , et ETX qui mesure la qualité de lien entre un noeud et son parent préféré, et on l'a implémenté sous le simulateur cooja, les résultats de simulation montrent un équilibre de charge dans les noeuds d'étranglement et réduisent l'énergie de noeud ainsi que de réseau par rapport à RPL standard. Ce projet nous a permis d'acquérir des connaissances en programmation événementielle. Il nous a aussi fait découvrir le simulateur cooja et son OS contiki. En outre, dans ce projet on a pu bien comprendre le fonctionnement de protocole RPL dédié à les réseaux LLN , mais on a rencontré des difficultés dans l'implémentation comme par exemple une limitation dans la documentation sur le protocole RPL et la simulation sous cooja.

En perspectives, nous proposons de mettre en place les protocoles de RPL modifiés sur des plateformes réelles .

- étudier l'impact de la mobilité sur notre protocole,
- étudier la scalabilité du protocole
- implémenter les mêmes métriques et comparer le protocole avec un protocole réactif tel que LOADng ;
- ajouter d'autres critères d'évaluation de performances,
- etc

# Bibliographie

- [1] *journal RFID*, 22 :97–114.
- [2] Karl Aberer and Manfred Hauswirth. Middleware support for the "internet of things". 2006.
- [3] Luigi Atzori, Antonio Iera, and Giacomo Morabito. The internet of things : A survey. *Computer networks*, 54(15) :2787–2805, 2010.
- [4] Nadjib Badache, Djamel Djenouri, Abdelouahid Derhab, and Tayeb Lemlouma. "les protocoles de routage dans les réseaux mobiles ad hoc. *Revue RIST*, 12(2) :77–112, 2002.
- [5] Yassine Banouar. *Gestion autonome de la QoS au niveau middleware dans l'IoT*. PhD thesis, Université de Toulouse, Université Toulouse III-Paul Sabatier, 2017.
- [6] Edward T Chen. The internet of things : Opportunities, issues, and challenges. In *The Internet of Things in the Modern Business Environment*, pages 167–187. IGI Global, 2017.
- [7] Louis Collins. *Optimisation centralisée et distribuée de la durée de vie dans les réseaux de capteurs sans fil*. PhD thesis, Université Catholique de Louvain.
- [8] Olfa Gaddour, Anis Koubâa, Raghuraman Rangarajan, Omar Cheikhrouhou, Eduardo Tovar, and Mohamed Abid. Co-rpl : Rpl routing for mobile low power wireless sensor networks using corona mechanism. In *Industrial Embedded Systems (SIES), 2014 9th IEEE International Symposium on*, pages 200–209. IEEE, 2014.
- [9] B Ghaleb. Load balancing objective function in rpl draft-qasem-roll-rpl-load-balancing-00. 2017.
- [10] Jorge Granjal, Edmundo Monteiro, and Jorge Sá Silva. Security in the integration of low-power wireless sensor networks with the internet : A survey. *Ad Hoc Networks*, 24 :264–287, 2015.
- [11] Jayavardhana Gubbi, Rajkumar Buyya, Slaven Marusic, and Marimuthu Palaniswami. Internet of things (iot) : A vision, architectural elements, and future directions. *Future generation computer systems*, 29(7) :1645–1660, 2013.
- [12] Ulrich Herberg and Thomas Clausen. A comparative performance study of the routing protocols load and rpl with bi-directional traffic in low-power and lossy networks (lln). In *Proceedings of the 8th ACM Symposium on Performance evaluation of wireless ad hoc, sensor, and ubiquitous networks*, pages 73–80. ACM, 2011.
- [13] Oana Iova, Fabrice Theoleyre, and Thomas Noel. Using multiparent routing in rpl to increase the stability and the lifetime of the network. *Ad Hoc Networks*, 29 :45–62, 2015.
- [14] Nadjah Kara, Djoudi Touazi, et al. *Conception d'un réseau de communication pour une maison intelligente en utilisant la technique d'internet des objets*. PhD thesis, Université A/Mira de Bejaia, 2017.

- 
- [15] Hyung-Sin Kim, Hongchan Kim, Jeongyeup Paek, and Saewoong Bahk. Load balancing under heavy traffic in rpl routing protocol for low power and lossy networks. *IEEE Transactions on Mobile Computing*, 16(4) :964–979, 2017.
- [16] Hyung-Sin Kim, Jeongyeup Paek, David E Culler, and Saewoong Bahk. Do not lose bandwidth : Adaptive transmission power and multihop topology control. In *2017 13th International Conference on Distributed Computing in Sensor Systems (DCOSS)*, pages 99–108. IEEE, 2017.
- [17] Haiyan Liu, Yanyan Liu, Zenggang Xiong, Zhen Yu, and Kun Tian. Study and application of urban flood risk map information management system based on soa. *Journal of Software*, 10(2) :180–190, 2015.
- [18] Xinxin Liu, Jianlin Guo, Ghulam Bhatti, Philip Orlik, and Kieran Parsons. Load balanced routing for low power and lossy networks. In *Wireless Communications and Networking Conference (WCNC), 2013 IEEE*, pages 2238–2243. IEEE, 2013.
- [19] Tarcisio Bruno Oliveira, Pedro Henrique Gomes, Danielo G Gomes, and Bhaskar Krishnamachari. Alabamo : A load balancing model for rpl. In *Proc. Brazil. Symp. Comput. Netw. Distrib. Syst. (SBRC)*, pages 644–657, 2016.
- [20] Mikko Sannikka et al. Palvelunlaatu langattomissa lähiverkoissa. 2009.
- [21] MOCHAMMAD HARIADI Suhartono, MAURIDHI HERY Purnomo, ZHAO BAITING, XIAOFEN JIA, Huanshuang Niu, Zhuoyong Zhang, Yuhong Xiang, Liting Zhao, Fan Yang, A SAKTHIVEL, et al. Plant growth modeling of zinnia elegans jacq using fuzzy mamdani and l system approach with mathematica. *Journal of Theoretical and Applied Information Technology*, 50(1) :1–6, 2013.
- [22] Elisabeth Tovar. Mesurer la justice socio-spatiale : de l’ancien au nouveau monde, promesses et menaces du «big data». *justice spatiale-spatial justice*, 10 :http-www, 2016.
- [23] Rehmat Ullah, Yasir Faheem, and Byung-Seo Kim. Energy and congestion-aware routing metric for smart grid ami networks in smart city. *IEEE Access*, 5 :13799–13810, 2017.
- [24] Jean-Philippe Vasseur, Mijeom Kim, Kris Pister, Nicolas Dejean, and Dominique Barthel. Routing metrics used for path calculation in low-power and lossy networks. Technical report, 2012.
- [25] Mehdi YOUSSEF, David NAVARRO, Patrick SERRAFERO, Maxime FRACHON, and France Brignais. Formalisation du cycle de vie des produits de l’usine du futur étendue : vers l’ingénierie système élargie.

