

---

RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE  
MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE SCIENTIFIQUE  
CENTRE UNIVERSITAIRE AÏN-TÉMOUCHENT BELHADJ BOUCHAÏB



Institut des Sciences  
Département de Mathématiques et Informatique

## Mémoire

Pour l'obtention du Diplôme de Master en Informatique  
Option : Réseaux et Ingénierie des Données (RID)

Présenté par :

Mr. Mohamed BABOUCHE  
Mr. Fethallah TOUATI

---

# OPTIMISATION DU ROUTAGE DANS LES RÉSEAUX LLNs

---

Encadrant :

Mr. M.A. MESSAOUDI  
Maitre Assistant "A" à CUAT BB

Encadrant :

Mr. A. BENZERBADJ  
Maitre de Conférence "B" à CUAT BB

Soutenu Le 23 Juin 2019

Devant le jury composé de :

---

Président :	Mme. F.BEDDAD (M.A.A)	CUAT BB.
Examineurs :	Mr. M.MEDEDJEL (M.A.A)	CUAT BB.
Encadrant :	Mr. M.A.MESSAOUDI (M.A.A)	CUAT BB.
Co-Encadrant :	Mr. A.BENZERBADJ (M.C.B)	CUAT BB.

---

Année Universitaire 2018-2019

---

# *Remerciements*

*Nous tenons tout d'abord à remercier DIEU le tout puissant et Miséricordieux, qui nous a donné la force et la patience d'accomplir ce modeste travail*

*Un très grand merci à :*

*Nos parents qui nous ont suivis pendant nos études*

*En second lieu nous tenons à remercier notre encadreur Mr MESSAOUDI et notre Co-encadreur Mr BENZERBADJ pour leurs précieux conseils et leurs encouragements durant tout la période de ce projet.*

*Nous remercions les membres de jury qui ont pris de leur temps pour juger ce modeste travail, qu'ils trouvent ici l'expression de notre gratitude et tout notre respect.*

*Nous adressons aussi nos remerciements à tous les professeurs qui nous ont enseignés durant le cursus universitaire afin d'obtenir le diplôme de master*

*Merci à tous les travailleurs de Centre Universitaire Belhadj Bouchaib d'Ain-Temouchent et à tous nos amis avec qui nous avons gardé des souvenirs très agréables des bons moments vécus ensemble*

*En fin, nous remercions notre promotion RID 2018-2019, tout en leurs souhaitons un avenir plein de réussite.*

---

## *Dédicace*

*Je dédie ce mémoire à...*

*À mes chers parents, pour tous leurs sacrifices, leur amour, leur tendresse, leurs prières tout au long de mes études*

*A ma grand soeur Asmaa à qui je souhaite son beaucoup de joie et de bonheur dans sa vie*

*A ma petit frère Housseem à qui je souhaite son la réussite dans sa vie.*

*A tous les membres de ma famille, petits et grands, mes oncles, mes tantes, mes cousins et mes cousines.*

*A tous mes amies et mes camarades ... Merci pour tous ces moments qu'on avait vécus ensemble*

*A mon binôme TOUATI Fethallah .*

BABOUCHE Mohamed.

---

## *Dédicace*

*Je dédie ce mémoire à...*

*Ma mère, qui a œuvré pour ma réussite, de par son amour, son soutien, tous les sacrifices consentis et ses précieux conseils, pour toute son assistance et sa présence dans ma vie, reçois à travers ce travail aussi modeste soit-il, l'expression de mes sentiments et de mon éternelle gratitude.*

*Mon père, qui peut être fier et trouver ici le résultat de longues années de sacrifices et de privations pour m'aider à avancer dans la vie. Puisse Dieu faire en sorte que ce travail porte son fruit. Merci pour les valeurs nobles, l'éducation et le soutien permanent venu de vous.*

*Mes professeurs du CUAT, les professeurs et développeurs à travers le monde auquel j'avais l'honneur d'apprendre auprès d'eux qui doivent voir dans ce travail la fierté d'un savoir bien acquis.*

*Mes amis et Mes camarades de classe qui m'ont pas laissé tout seul pendant toute cette épreuve et qui m'ont aidé jusqu'au bout, et j'espère bien que ce travail les rendra satisfaits de leurs soutiens.*

*Mes petits frères, mes sœurs, mes cousins, cousines et tout les membres de ma grande famille qui étiez toujours a mes cotés a me soutenir et a m'encourager à faire l'impossible pour me mettre en contact avec des professeurs quelque soit au niveau national ou bien international, je souhaite qu'avec ce travail je peux les récompenser.*

*Un remerciement très spéciale pour le cher cousin Omar qui était plus qu'un cousin plus qu'un frère et qui a laissé la vie le mois dernier en laissant derrière lui des espoirs et des belles histoires a propos de cette soutenance sans pouvoir être la en ce moment, merci infiniment mon très cher que dieu te bénisse et te mettre au prêt du prophète Mohamed.*

TOUATI Fethallah.

## Résumé

Ces dernières années la communauté scientifique et industriel avait un intérêt croissant pour les réseaux capteur sans fil RCSF qui mène un nouveau protocole Normalisé en considérant les caractéristiques matériels Des nœuds utilisés .

L'IETF (Mobile Engineering Task Force) a proposé dans la couche réseau le protocole RPL ( de l'acronyme IPV6 Routing Protocole for Low-Power and Lossy Networks ) Comme standard de routage pour les réseaux dont les nœuds sont de types "LLN" (Low-Power and Lossy Networks ). Une faible autonomie énergétique le caractérise et le transmet sur des liens radios dotés d'un taux de perte de données élevé . Le but de notre recherche est d'expertiser le travail de RPL selon deux métriques de routage : ETX et l'énergie. Puis, conseiller un perfectionnement du RPL pour un meilleur appui de la mobilité pour améliorer les performances des LLN's, et surtout la durée de vie du réseau et le nombre de paquets reçus. A cet effet, nous avons combiné deux métriques pour combler les limites des métriques de routage considérées séparément. Les résultats obtenus étaient prometteurs en termes de consommation d'énergie et la durée de vie du réseau et le nombre de paquets reçus.

**Keywords :** Réseaux de capteurs,LLN, RPL, Routage, IPv6,Simulations.

## **Abstract**

In recent years the scientific and industrial community has a growing interest in the wireless sensor networks RWSN which leads a new standardized protocol considering the hardware characteristics of the nodes used.

The IETF has proposed RPL (IPv6 Routing Protocol for Low-Power and Lossy Networks) as the routing standard for networks with "LLN" (Low-Power and Lossy) type nodes. A low energetic autonomy characterizes it and transmits it on radio links with a high rate of data loss.

The purpose of our research is to appraise the work of RPL according to two routing metrics: ETX and energy. Then, advise an improvement of the RPL for a better support of the mobility to improve the performances of LLN's, and especially the lifespan of the network and the number of packets received. For this purpose, we have combined two metrics to fill the limits of the routing metrics considered separately. The results obtained were promising in terms of energy consumption and network lifetime and the number of packets received.

**Mots clés:** Wireless sensor networks,LLN, RPL, Routing, IPv6, Simulations.

# Table des matières

<b>Remerciements</b>	<b>i</b>
<b>Dédicace</b>	<b>ii</b>
<b>Introduction</b>	<b>1</b>
<b>1 Les réseaux de capteurs sans fil</b>	<b>2</b>
1.1 Introduction . . . . .	3
1.2 Qu'est ce qu'un capteur . . . . .	3
1.2.1 Architecture . . . . .	4
1.3 Définition d'un réseau RCSF . . . . .	6
1.4 Architecture d'un RCSF . . . . .	6
1.5 Topologie des réseaux de capteurs . . . . .	7
1.5.1 Topologie physique . . . . .	7
1.5.1.1 Topologie en étoile . . . . .	7
1.5.1.2 Topologie en maille ou grille (Mesh Network) . . . . .	7
1.5.1.3 Topologie hybride . . . . .	8
1.5.2 Topologie logique . . . . .	8
1.5.2.1 Topologie plate . . . . .	8
1.5.2.2 Topologie hiérarchique . . . . .	8
1.6 Routage et transmission dans les réseaux de capteurs . . . . .	9
1.6.1 Le routage . . . . .	9
1.6.2 La transmission de l'information dans un réseau de capteurs . . . . .	10
1.6.2.1 L'envoi direct . . . . .	10
1.6.2.2 L'envoi par routage multi-sauts . . . . .	10
1.6.3 Routage dans un réseau de capteurs . . . . .	11
1.6.3.1 Taxonomie des protocoles de routages . . . . .	11

1.6.3.2	Protocoles de routage basé sur établissement de route	12
1.6.3.3	Protocoles de routage basé sur la structure réseau . . .	13
1.7	Facteurs et contraintes des RCSF . . . . .	15
1.7.1	Durée de vie du réseau . . . . .	15
1.7.2	Ressources limitées . . . . .	15
1.7.3	Bande passante limitée . . . . .	15
1.7.4	Topologie dynamique . . . . .	15
1.7.5	L'environnement . . . . .	16
1.7.6	Agrégation de donnée . . . . .	16
1.8	L'internet des objets . . . . .	16
1.9	Low-Power and Lossy Networks . . . . .	17
1.9.1	LLN versus RCSF . . . . .	18
1.9.2	IPV6 . . . . .	18
1.9.3	Description de la norme IEEE 802.15.4 . . . . .	19
1.10	6LOWPAN . . . . .	19
1.10.1	Architecture 6LoWPAN . . . . .	19
1.10.2	Fonctionnement de 6LoWPAN . . . . .	20
1.11	Le routage dans les réseau LLNs . . . . .	21
1.11.1	Contraintes de routage dans les réseaux LLN . . . . .	21
1.11.2	Algorithmes de routage dans les réseaux LLN . . . . .	21
1.12	Conclusion . . . . .	22
<b>2</b>	<b>Le protocole de routage RPL</b>	<b>23</b>
2.1	Introduction . . . . .	24
2.2	Historique . . . . .	24
2.3	Définition . . . . .	25
2.4	Fonctionnement du protocole RPL . . . . .	26
2.4.1	Construction de la topologie . . . . .	26
2.4.1.1	Les graphes DAG et DODAG . . . . .	26
2.4.1.2	Le modèle réseau . . . . .	27
2.4.1.3	Messages de contrôle du protocole RPL . . . . .	27
2.4.1.4	Construction du DODAG . . . . .	28
2.4.1.5	Les modes d'opération du protocole RPL . . . . .	31
2.4.1.6	Les paradigmes de communication . . . . .	32
2.4.1.7	Gestion du réseau RPL . . . . .	32
2.4.2	Maintenance de la topologie . . . . .	33
2.4.2.1	Réparation globale: . . . . .	33
2.4.2.2	Réparation locale : . . . . .	33



2.5	RPL et la notion du temps . . . . .	35
2.5.1	Le fonctionnement de l’algorithme Trickle Timer . . . . .	35
2.6	Fonction Objectife . . . . .	37
2.6.1	Objective Function Zéro . . . . .	37
2.6.2	Minimum Rank with Hysteresis Objective Function . . . . .	37
2.7	Les principales métriques . . . . .	38
2.7.1	Control Traffic Overhead . . . . .	38
2.7.2	Lost Packet . . . . .	39
2.7.3	Node Energy . . . . .	39
2.7.4	Hop Count (HC) . . . . .	39
2.7.5	Expected Transmission Count (ETX) . . . . .	39
2.8	Sécurité dans RPL . . . . .	40
2.9	Conclusion . . . . .	41
<b>3</b>	<b>Améliorations des performances de RPL</b>	<b>42</b>
3.1	Introduction . . . . .	43
3.2	Outils de la simulation . . . . .	43
3.2.1	Contiki . . . . .	43
3.2.2	Architecture . . . . .	44
3.2.3	Connectivité . . . . .	44
3.2.3.1	La couche Rime . . . . .	44
3.2.3.2	La couche uIP . . . . .	44
3.2.4	Simulateur Cooja . . . . .	45
3.2.4.1	La fenêtre Timeline . . . . .	45
3.2.4.2	La fenêtre Network . . . . .	45
3.2.4.3	La fenêtre Mote Output . . . . .	45
3.2.4.4	La fenêtre Notes . . . . .	46
3.2.4.5	La fenêtre Simulation control . . . . .	46
3.3	Environnement de développement . . . . .	46
3.4	Evaluation de RPL . . . . .	47
3.5	Métriques de la simulation . . . . .	47
3.6	Amélioration proposée . . . . .	48
3.6.1	Démarche à suivre . . . . .	48
3.7	Résultat de la simulation . . . . .	49
3.7.1	topologie sous forme d’une gril ordonnée . . . . .	50
3.7.2	Simulation 1 . . . . .	51
3.7.3	Simulation 2 . . . . .	52
3.7.4	Simulation 3 . . . . .	53

## TABLE DES MATIÈRES

---

3.7.5	Simulation 4 . . . . .	54
3.7.6	Simulation 5 . . . . .	55
3.7.7	Simulation 6 . . . . .	56
3.8	Simulation d'une défaillance partielle de réseau . . . . .	57
3.9	Conclusion . . . . .	59
	<b>Conclusion</b>	<b>60</b>
	<b>Bibliographie</b>	<b>61</b>

# Table des figures

1.1	Les fonctions principales d'un capteur [1]	3
1.2	Architecture d'un capteur [2]	5
1.3	Architecture RCSF [1]	7
1.4	Envoi direct [5]	10
1.5	Envoie par routage [5]	11
1.6	Classification des protocoles de routage	12
1.7	L'internet des objets[8].	17
1.8	Architecture 6LoWPAN [12].	20
2.1	Organisation de l'environnement RPL[15].	26
2.2	les graphes DAG et DODAG[15].	26
2.3	Etape 1, diffusion des DIO par la racine (DAG root)[15].	29
2.4	Etape 2, choix du root comme noeuds parent[15]	29
2.5	Etape 3, pour suite de la construction du DODAG[15].	30
2.6	Le fonctionnement en «non-storing mode»[18]	31
2.7	Le fonctionnement en «storing mode»[18]	31
2.8	Exemple de réparation locale[15][15].	34
3.1	Architecture de Contiki[22].	44
3.2	Pile protocolaire de Contiki [22].	45
3.3	Fenêtres de simulations Cooja [22].	46
3.4	Topologie: (a) 9 Noeuds , (b) 16 Noeuds	50
3.5	Topologie: (a) 25 Noeuds , (b) 36 Noeuds	50
3.6	Topologie 49 Noeuds	50
3.7	Energie consommée par un réseau avec et sans sink mobile (MRHOF–Energie)	51
3.8	Energie consommée par un réseau avec et sans sink mobile (MRHOF–ETX)	52

## TABLE DES FIGURES

---

3.9	Energie consommée par un réseau avec et sans sink mobile (MRHOF-Energie+ETX) . . . . .	53
3.10	Energie consommée par des réseaux avec sinks fixes (MRHOF-Energie-ETX-Energie+ETX) . . . . .	54
3.11	Energie consommée par des réseaux avec sinks mobiles (MRHOF-Energie-ETX-Energie+ETX) . . . . .	55
3.12	Les PDR des réseaux avec sinks fixes (MRHOF-Energie-ETX-Energie+ETX)	56
3.13	Les PDR des réseaux avec sinks mobiles (MRHOF-Energie-ETX-Energie+ETX) . . . . .	56
3.14	Topologie initiale (réseau fonctionnel) . . . . .	57
3.15	DODAG de la topologie initiale . . . . .	57
3.16	Topologie après le déplacement du sink . . . . .	58
3.17	DODAG après le déplacement du sink . . . . .	58
3.18	Analyse des résultats . . . . .	59

# Liste des tableaux

3.1	Caractéristiques de la machine utilisée pour les simulations . . . . .	47
3.2	Paramètres de simulation . . . . .	47

# Introduction Générale

De nos jours, les progrès technologiques permettent d'envisager la connexion des objets du quotidien à l'Internet. L'omniprésence programmée de ces objets implique la fabrication d'objets bon marché (peu d'autonomie énergétique, peu de mémoire de stockage et peu de puissance de calcul). L'ensemble connecté de ces objets est appelé réseau LLN (Low power and Lossy Network).

Dans ce contexte, le protocole de routage RPL (IPv6 Routing Protocol for Low Power and Lossy Networks) est conçu an de prendre en charge les exigences spécifiques de ces réseaux. Toutefois le protocole RPL est peu adapté à la mobilité des nœuds (topologie dynamique).

Notre objectif principal de ce projet de fin d'études est de proposer une amélioration du protocole RPL pour une meilleure prise en charge de la mobilité afin d'améliorer les performances des LLN's, notamment la durée de vie du réseau et le nombre de paquets reçus ,

Ce mémoire comporte trois chapitres :

Dans le premier chapitre, nous présentons les concepts de base des réseaux de capteurs sans fil en mettant l'accent sur les réseaux LLN ou réseaux à faible puissance et fort taux de perte, leurs caractéristiques et leurs modes de fonctionnement. Nous finirons ce chapitre par la présentation des contraintes de routage dans les réseaux LLN.

Le deuxième chapitre, nous nous focaliserons sur le protocole de routage RPL en présentant les différents mécanismes mis en œuvre dans RPL, ainsi que son mode de fonctionnement (les métriques de routage).

Nous proposons dans le dernier chapitre une mise en œuvre de notre démarche, cette dernière est basée sur l'intégration d'un sink mobile dans un réseau LLN en utilisant le protocole RPL avec variation des métriques de routage. En dernier lieu, nous analysons les résultats des simulations menés dans Contiki.

Nous terminons ce mémoire par une conclusion générale dans laquelle nous précisons les apports de notre démarche et les perspectives à cette dernière.

# Chapitre 1

## Les réseaux de capteurs sans fil

### Sommaire

---

1.1	Introduction . . . . .	3
1.2	Qu'est ce qu'un capteur . . . . .	3
1.3	Définition d'un réseau RCSF . . . . .	6
1.4	Architecture d'un RCSF . . . . .	6
1.5	Topologie des réseaux de capteurs . . . . .	7
1.6	Routage et transmission dans les réseaux de capteurs . . . . .	9
1.7	Facteurs et contraintes des RCSF . . . . .	15
1.8	L'internet des objets . . . . .	16
1.9	Low-Power and Lossy Networks . . . . .	17
1.10	6LOWPAN . . . . .	19
1.11	Le routage dans les réseaux LLNs . . . . .	21
1.12	Conclusion . . . . .	22

---

## 1.1 Introduction

Aujourd'hui, les réseaux de capteurs sans fil sont de plus en plus populaires du fait de leur facilité de déploiement et leur utilisation. Ces réseaux jouent un rôle primordial dans plusieurs domaines tels que le domaine militaire, le domaine médical, etc... Ils offrent des solutions ouvertes pour ces domaines d'applications. Cependant, leurs contraintes en termes d'énergie, de calcul et de communication ont rendu la conception des protocoles de routage un grand défi. Dans ce chapitre, nous présentons les réseaux de capteurs sans fil, leurs architectures et leurs topologies, ainsi les Facteurs et contraintes de RCSF.

## 1.2 Qu'est ce qu'un capteur

Un capteur est un dispositif destiné à mesurer une donnée environnementale comme la température, ou même capturer une image par exemple, et de la transformer en information utilisable puis de la transmettre à une unité de traitement de façon analogique ou numérique. Ces capteurs, non autonomes doivent donc être connectés à un appareil capable d'en interpréter la mesure, puis, selon l'usage souhaité permettre l'utilisation. Chaque capteur assure trois fonctions principales : la collecte, le traitement et la communication de l'information vers un ou plusieurs points de collecte appelés station de base.[1]



FIGURE 1.1 – Les fonctions principales d'un capteur [1]



### 1.2.1 Architecture

Suivant le type d'application : il existe plusieurs types de capteurs : les capteurs de température, d'humidité, de pression, etc. malgré cette diversité apparente, ils restent dotés d'une architecture matérielle similaire. Un capteur est composé principalement d'une unité de : capture, traitement, communication et énergie. Des composants additionnels peuvent être ajoutés selon le domaine d'application, comme par exemple un système de localisation de l'environnement tel qu'un GPS (Global Positioning System), d'un système de mobilité mais aussi parfois d'un générateur d'énergie (exemple : cellule solaire). [2]

**Unité de capture ("Sensing unit") :** Est composée de deux sous-unités, un dispositif de capture physique qui prélève l'information de l'environnement local et un convertisseur analogique/numérique appelé ADC ("Analog to Digital Converter"). Le capteur est responsable de fournir des signaux analogiques. Et l'ADC transforme ces signaux en un signal numérique compréhensible par l'unité de traitement.

**Unité de traitement ("Processing unit") :** C'est l'unité principale du capteur, généralement un processeur couplé à une mémoire vive. Son rôle est de contrôler le bon fonctionnement des autres unités. Sur certains capteurs elle peut embarquer un système d'exploitation pour faire fonctionner le capteur. Elle peut aussi être couplée à une unité de stockage, qui servira par exemple à y enregistrer les informations transmises par l'unité de capture.

**Unité de communication ("Transceiver unit") :** Elle effectue toutes les émissions et réceptions des données sur un médium « sans-fil ». Elle peut être de type optique (comme dans les noeuds Smart Dust), ou de type radiofréquence. Les communications de type optique sont robustes vis-à-vis des interférences électriques. Néanmoins, ne pouvant pas établir de liaisons à travers des obstacles, elles présentent l'inconvénient d'exiger une ligne de vue permanente entre les entités communicantes. Les unités de transmission de type radiofréquence comprennent des circuits de modulation, démodulation, filtrage et multiplexage ; ceci implique une augmentation de la complexité et du coût de production du micro-capteur.

Concevoir des unités de transmission de type radiofréquence avec une faible consommation d'énergie est un défi car pour qu'un noeud ait une portée de communication

suffisamment grande, il est nécessaire d'utiliser un signal assez puissant et donc une énergie consommée importante. L'alternative consistant à utiliser de longues antennes n'est pas possible à cause de la taille réduite des micro-capteurs.

**Unité d'énergie ("Power unit") :** Un noeud capteur est muni d'une ressource énergétique (une batterie). Étant donné sa petite taille, cette ressource énergétique est limitée. Ceci fait souvent de l'énergie la ressource la plus précieuse d'un capteur, car elle influe directement sur sa durée de vie.[2]

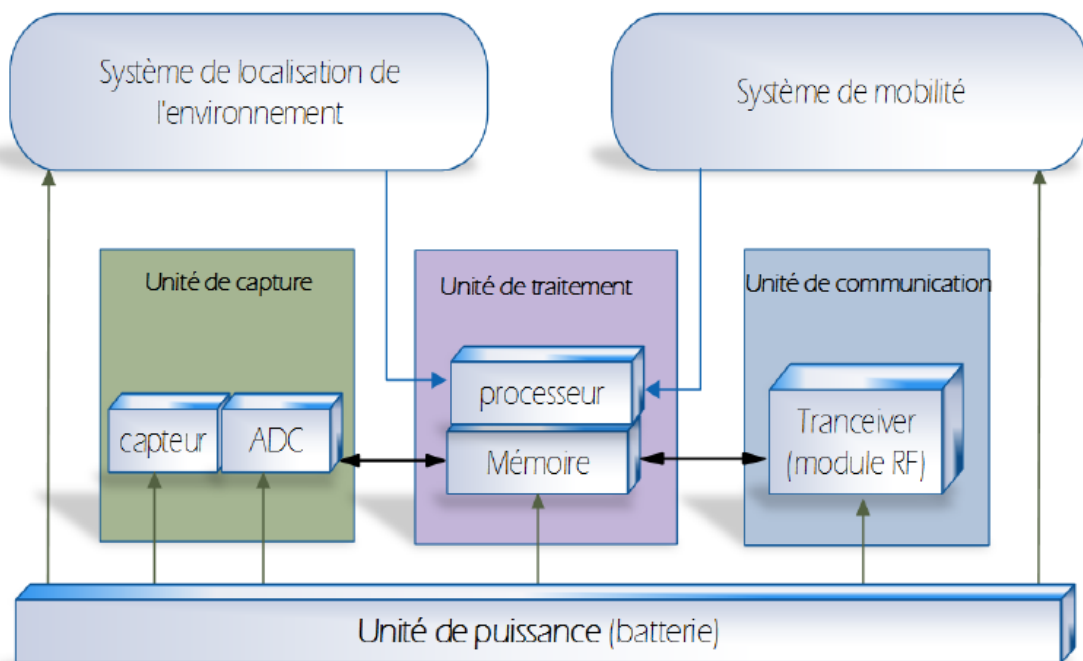


FIGURE 1.2 – Architecture d'un capteur [2]

**Système de localisation de l'environnement :** Les tâches de détection et les techniques de routage ont besoin de connaître souvent la localisation géographique d'un noeud. Ainsi, un noeud peut être équipé d'un système de localisation géographique. Ce système peut se composer d'un module de GPS pour un noeud de haut niveau ou bien d'un module de software qui implémente des algorithmes de localisation qui fournissent les informations sur l'emplacement du noeud par des calculs distribués.

**Système de mobilité :** La mobilité est parfois nécessaire pour permettre à un noeud de se déplacer pour accomplir ses tâches. Le support de mobilité exige des ressources énergétiques étendues qui devraient être fournies efficacement. Le système de mobilité peut, également, opérer dans l'interaction étroite avec l'unité de détection et le processeur pour contrôler les mouvements du noeud.

### 1.3 Définition d'un réseau RCSF

Un Réseau de Capteurs Sans Fil (RCSF) ou Wireless Sensor Network (WSN) est un réseau informatique composé de petits dispositifs autonomes, fixés ou dispersés aléatoirement dans une zone d'intérêt, utilisant des capteurs coopérant pour surveiller des conditions environnementales ou physiques, comme la température, le son, les vibrations, la pression, le mouvement, etc. Puisque les réseaux de capteurs sans fil peuvent être déployés dans des terrains inaccessibles, la position des noeuds capteurs ne peut être prédéterminée. En conséquence, un système de localisation est requis afin de fournir les informations de position aux noeuds. Parmi les domaines d'application on trouve la santé, le domaine militaire, et de la sécurité.[1]

### 1.4 Architecture d'un RCSF

Un RCSF est composé d'un ensemble de noeuds capteurs. Ces noeuds capteurs sont organisés en champs « sensor fields » (voir figure suivante). Chacun de ces noeuds a la capacité de collecter des données et de les transférer au nœud passerelle (dit "sink" en anglais ou puits) par l'intermédiaire d'une architecture multi-sauts. Le puits transmet ensuite ces données par Internet ou par satellite à l'ordinateur central «Gestionnaire de tâches» pour analyser ces données et prendre des décisions.[1]

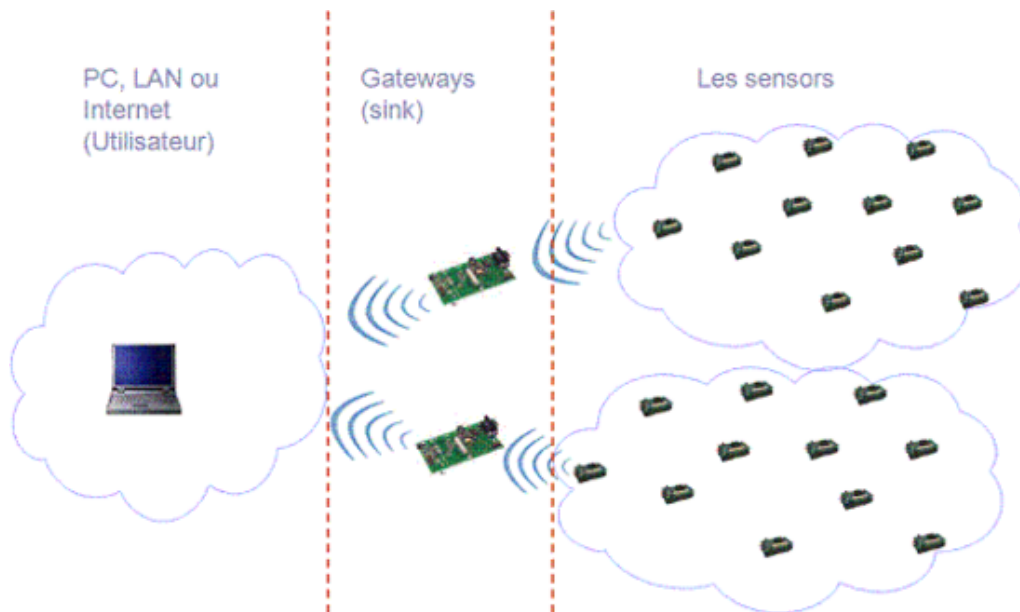


FIGURE 1.3 – Architecture RCSF [1]

## 1.5 Topologie des réseaux de capteurs

Un réseau de capteurs sans fil est composé d'un ensemble de noeuds capteurs et des Gateway qui s'occupent de collecter les données des capteurs et de les transmettre à l'utilisateur via l'internet ou le satellite, il existe plusieurs topologies pour les réseaux de capteurs :[2].

### 1.5.1 Topologie physique

#### 1.5.1.1 Topologie en étoile

La topologie en étoile est un système uni-saut. Tous les noeuds envoient et reçoivent seulement des données avec la station de base. Cette topologie est simple et elle demande une faible consommation d'énergie, mais la station de base est vulnérable et la distance entre les noeuds et la station est limitée.[2].

#### 1.5.1.2 Topologie en maille ou grille (Mesh Network)

Dans ce cas (dit « communication multi-sauts »), tout noeud peut échanger avec n'importe quel autre noeud du réseau (s'il est à portée de transmission). Un noeud voulant transmettre un message à un autre noeud hors de sa portée de transmission, peut utiliser un noeud intermédiaire pour envoyer son message au noeud destinataire. Cette topologie a plus de possibilités de passer à l'échelle du réseau, avec redondance et tolérance aux fautes, mais elle demande une consommation d'énergie

plus importante.

Une latence est créée par le passage des messages des noeuds par plusieurs autres avant d'arriver à la station de base.[2].

### 1.5.1.3 Topologie hybride

La topologie hybride est un mélange des deux topologies ci-dessus. Les stations de base forment une topologie en toile et les noeuds autour d'elles sont en topologie étoile. Elle assure la minimisation d'énergie dans les réseaux de capteurs.[2].

## 1.5.2 Topologie logique

La topologie détermine l'organisation logique des capteurs dans le réseau. Il existe deux principales topologies dans les protocoles de routage pour les WSNs.

### 1.5.2.1 Topologie plate

Un réseau de capteurs sans fil plat est un réseau homogène, où tous les noeuds sont identiques en termes de batterie et de complexité du matériel et ont le même rôle, excepté le noeud puits qui joue le rôle d'une passerelle et qui est responsable de la transmission de l'information collectée à l'utilisateur final. Selon le service et le type de capteurs, une densité de capteurs élevée (plusieurs noeuds capteurs/m) ainsi qu'une communication multi-sauts peut être nécessaire pour l'architecture plate. Ce type de solution permet une grande tolérance aux pannes, cependant, elle souffre d'une faible scalabilité. En effet, si tous les noeuds opèrent de la même façon et d'une manière distribuée, on aura un grand nombre de messages de contrôle nécessaires pour le bon fonctionnement du réseau [3].

### 1.5.2.2 Topologie hiérarchique

Afin d'augmenter la scalabilité du système, les topologies hiérarchiques ont été introduites en divisant les noeuds en plusieurs niveaux de responsabilité. L'une des méthodes la plus employée est le Clustering, avec laquelle le réseau est partitionné en groupes appelés clusters. Un cluster est constitué d'un chef (ClusterHead) et de ses membres. Suivant l'application, les membres peuvent être des voisins directs ou indirects du ClusterHead.[3]

Avec une approche hiérarchique, il est plus facile d'intégrer un mécanisme d'agrégation au système : les noeuds membres transmettent leurs données vers le ClusterHead, qui va par la suite agréger ces lectures afin de transmettre le résumé à la

station de base. Pour augmenter la tolérance aux pannes, la sélection des ClusterHead doit être dynamique afin d'éviter la présence des membres isolés.

L'inconvénient de la hiérarchisation est la surcharge des ClusterHeads, induisant à un déséquilibre de la consommation d'énergie sur le réseau [4].

## 1.6 Routage et transmission dans les réseaux de capteurs

### 1.6.1 Le routage

Le routage est un mécanisme pour l'acheminement des paquets de données d'un émetteur vers un ou plusieurs destinataires, à travers des routeurs et des protocoles de routage, les routeurs sont des machines qui relient deux ou plusieurs réseaux afin de commuter des paquets d'une interface vers une autre selon la destination des paquets, les protocoles de routage définissent la manière de fonctionnement des routeurs selon la propriété de protocole, les protocoles de routages sont classés en plusieurs grande classe selon des différent critères.

Les protocoles de routage servent à deux fonctions, la première est la construction avec la maintenance des routes pour certaines destinations, la deuxième consiste en l'acheminement des données sur ces routes.

Le plus grand défi des protocoles de routage est de trouver à un instant donné le meilleur chemin entre deux stations, c'est-à-dire la suite des noeuds pouvant acheminer les données le plus efficacement possible [5]. Cependant, les ressources du réseau peuvent changer à tout moment, la difficulté est donc de s'adapter à ces changements afin de maintenir la communication, d'une part, et de conserver un routage efficace d'autre part.

Il y'a deux type de routage, le routage statique consiste à faire les mises à jour des informations de routage de façon manuelle à chaque modification de la topologie réseau et le routage dynamique où les mises à jour des informations de routage sont faites de façon automatique à chaque modification de topologique réseau. [6]

## 1.6.2 La transmission de l'information dans un réseau de capteurs

La transmission de l'information dans un réseau peut se faire de deux façons :

### 1.6.2.1 L'envoi direct

Chaque noeud est en lien étroit avec l'unité de collecte, et aucun intermédiaire ne peut s'interposer dans cette relation directe privilégiée comme le montre la figure 1.4.

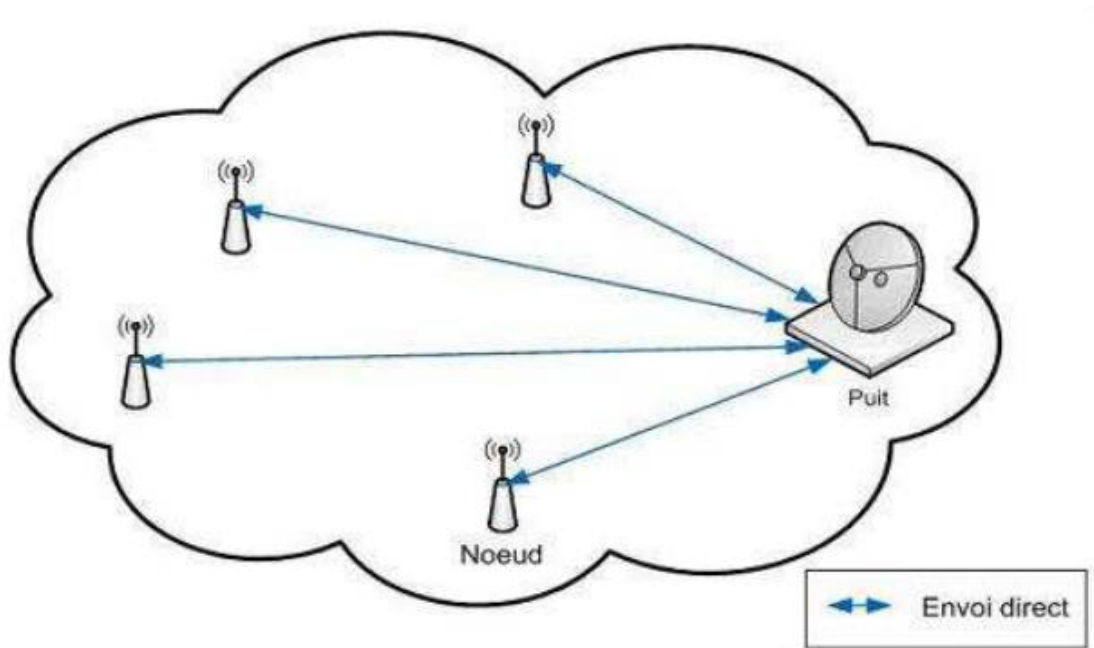


FIGURE 1.4 – Envoi direct [5]

### 1.6.2.2 L'envoi par routage multi-sauts

Lorsque les noeuds sont distants de la station de base, l'envoi direct n'est pas possible car la portée des capteurs est limitée et toute communication distante directe peut épuiser rapidement la batterie d'un capteur. Pour se remédier à ce problème, il faut passer par des noeuds relais pour acheminer l'information à la station de base comme le montre la figure 1.5.

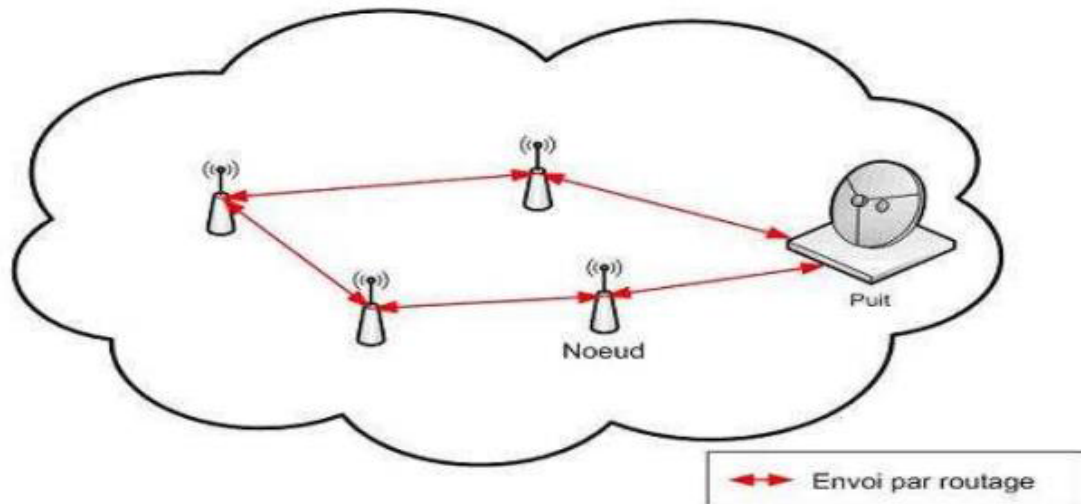


FIGURE 1.5 – Envoie par routage [5]

### 1.6.3 Routage dans un réseau de capteurs

Les noeuds de capteurs communiquent en utilisant un protocole de routage qui définit le processus de déplacement des paquets à travers le réseau d'un noeud à un autre. Le protocole choisi doit tenir compte des caractéristiques du réseau. Il devrait fournir l'évolutivité, la fiabilité, faible surcharge, etc.

#### 1.6.3.1 Taxonomie des protocoles de routages

Récemment, les protocoles de routage pour les RCSF ont été largement étudiés, et différentes études ont été publiées. Les méthodes employées peuvent être classifiées suivant plusieurs critères comme illustré sur la figure suivante :



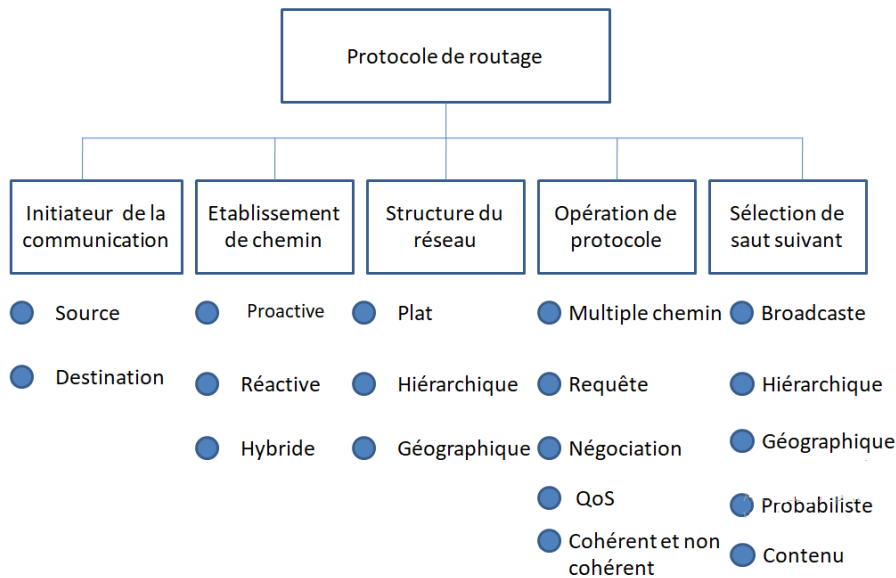


FIGURE 1.6 – Classification des protocoles de routage

### 1.6.3.2 Protocoles de routage basé sur établissement de route

Les chemins de routage peuvent être établis dans l'un des trois façons, à savoir proactif, réactif ou hybride. Protocoles proactive calculer toutes les routes avant qu'ils ne soient vraiment nécessaires et les stocker dans une table de routage sur chaque noeud. Protocoles réactifs calculer des itinéraires seulement quand ils sont nécessaire. Protocoles hybrides utilisent une combinaison de ces deux idées.[5]

**Protocole proactif :** Ces protocoles de routage essaient de maintenir les meilleurs chemins existants vers toutes les destinations possibles au niveau de chaque noeud du réseau. Les routes sont sauvegardées mêmes si elles ne sont pas utilisées. Chaque noeud du réseau maintient une table de routage pour toutes les destinations indépendamment de l'utilité des routes. Les protocoles proactifs sont adaptés aux applications qui nécessitent un prélèvement périodique des données. Et par conséquent, les capteurs peuvent se mettre en veille pendant les périodes d'inactivité, et n'enclencher leur dispositif de capture qu'à des instants particuliers.[6]

**Protocole réactif :** Ces protocoles (dits aussi, les protocoles de routage à la demande) créent et maintiennent des routes selon les besoins. Lorsque le réseau a besoin d'une route, une procédure de découverte de route est lancée. Ce type de protocoles est pratique pour des applications temps réel où les capteurs doivent ré-

agir immédiatement à des changements soudains des valeurs captées. En effet, un prélèvement périodique des données aurait été inadapté pour ce type de scénarios.[5]

**Protocole hybride :** Ces protocoles combinent les deux idées des protocoles proactifs et réactifs. Ils utilisent un protocole proactif pour apprendre le proche voisinage (par exemple le voisinage à deux ou à trois sauts), ainsi, ils disposent de routes immédiatement dans le voisinage. Au-delà de la zone du voisinage, le protocole hybride fait appel à un protocole réactif pour chercher des routes.

### 1.6.3.3 Protocoles de routage basé sur la structure réseau

Les protocoles sont répartis en fonction de la structure du réseau qui est très important pour l'opération requise. Les protocoles inclus dans cette catégorie sont divisées en trois sous-catégories en fonction de leurs fonctionnalités. Ces protocoles sont :

**Le routage plat :** Lorsque le nombre de noeuds de capteurs sont énorme et que chaque noeud joue même rôle, routage à base plate est nécessaire. Etant donné que le nombre de noeuds de capteurs est très grand par conséquent, il n'est pas possible d'attribuer une identification particulière (Id) à chaque noeud. Cela conduit à l'approche du routage centrée sur les données dans laquelle la station de base envoie la requête à un groupe de noeuds particuliers dans une région et attend la réponse. Des exemples de protocoles de routage à base de plat sont :

- Energy Aware Routing (EAR).
- Directed Diffusion (DD).
- Sequential Assignment Routing (SAR).
- Minimum Cost Forwarding Algorithm (MCFA).
- Sensor Protocols for Information via Negotiation (SPIN).
- Active Query forwarding In sensor network (ACQUIRE).

**Le routage hiérarchique :** Les méthodes de routage hiérarchique ont des avantages spéciaux liés au passage à l'échelle et à l'efficacité dans la communication. Par exemple, elles sont utilisées pour exécuter un routage avec économie d'énergie.[6] Le réseau est divisé en clusters dirigés par un chef de clusters. La création des clusters et l'assignation des tâches spéciales aux têtes de clusters peuvent considérablement renforcer le passage à l'échelle, l'augmentation de la durée de vie et l'efficacité énergétique du système global. Le routage hiérarchique est une manière efficace de réduire la consommation énergétique dans un cluster en exécutant l'agrégation et la fusion de données afin de diminuer le nombre de messages transmis à la station de base. Exemples de protocoles de routage hiérarchique :

- Hierarchical Power-Active Routing (HPAR).
- Threshold sensitive energy efficient sensor network protocol (TEEN).
- Power efficient gathering in sensor information systems.
- Minimum energy communication network (MECN).
- LEACH (Low-energy adaptive clustering hierarchy).

**Le routage géographique (location based routing) :** Dans ce type de routage il est supposé que chaque noeud du réseau connaisse sa position et les positions de ses voisins. Le positionnement du noeud peut être obtenu en utilisant un système de géo-positionnement tel que le GPS (Global Positioning System) ou bien via des algorithmes de positionnement relatif. Le principe général consiste à obliger les noeuds, qui ne sont pas sur le chemin du routage choisi, à entrer en mode sommeil pour conserver l'énergie.[5] Chaque noeud source de données connaît la position du destinataire de ses données de cette façon une estimation de la consommation de l'énergie est réalisée au préalable pour désigner le chemin le plus rentable énergétiquement. On peut citer quelques protocoles appartenant à cette catégorie :

- Geographic Adaptive Fidelity (GAF).
- Geographic and Energy-Aware Routing (GEAR).
- Trajectory-Based Forwarding (TBF).
- Sequential assignment routing (SAR).
- Ad-hoc positioning system (APS).
- Greedy other adaptive face routing (GOAFR).
- Geographic distance routing (GEDIR).

## 1.7 Facteurs et contraintes des RCSF

La conception et la réalisation des réseaux de capteurs sans fil sont influencées par plusieurs paramètres. Ces facteurs servent comme directives pour le développement des algorithmes et protocoles utilisés dans les RCSF.[7].

### 1.7.1 Durée de vie du réseau

C'est l'intervalle de temps qui sépare l'instant de déploiement du réseau de l'instant où l'énergie du premier noeud s'épuise. Selon l'application, la durée de vie exigée pour un réseau peut varier entre quelques heures et plusieurs années.[7].

### 1.7.2 Ressources limitées

En plus de l'énergie, les noeuds capteurs ont aussi une capacité de traitement et de mémoire limitée. En effet, les industriels veulent mettre en oeuvre des capteurs simples, petits et peu coûteux qui peuvent être achetés en masse.[7].

### 1.7.3 Bande passante limitée

Afin de minimiser l'énergie consommée lors de transfert de données entre les noeuds, les capteurs opèrent à bas débit. Typiquement, le débit utilisé est de quelques dizaines de Kb/s. Un débit de transmission réduit n'est pas handicapant pour un réseau de capteurs où les fréquences de transmission ne sont pas importantes.

### 1.7.4 Topologie dynamique

La topologie des réseaux de capteurs peut changer au cours du temps pour les raisons suivantes :[7].

- Les noeuds capteurs peuvent être déployés dans des environnements hostiles (champ de bataille par exemple), la défaillance d'un noeud capteur est, donc très probable.

- Un noeud capteur peut devenir non opérationnel à cause de l'expiration de son énergie.

- Dans certaines applications, les noeuds capteurs et les stations de base sont mobiles.

### 1.7.5 L'environnement

Les capteurs sont souvent déployés en masse dans des endroits tels que des champs de bataille au delà des lignes ennemies, à l'intérieur de grandes machines, au fond d'un océan, dans des champs biologiquement ou chimiquement souillés,... Par conséquent, ils doivent pouvoir fonctionner sans surveillance dans des régions géographiques éloignées[7].

### 1.7.6 Agrégation de donnée

Dans les réseaux de capteurs, les données produites par les noeuds capteurs voisins sont très corrélées spatialement et temporellement. Ceci peut engendrer la réception par la station de base d'informations redondantes. Réduire la quantité d'informations redondantes transmises par les capteurs permet de réduire la consommation d'énergie dans le réseau et ainsi d'améliorer sa durée de vie. L'une des techniques utilisée pour réduire la transmission d'informations redondantes est l'agrégation des données. Avec cette technique, les noeuds intermédiaires agrègent l'information reçue de plusieurs sources. Cette technique est connue aussi sous le nom de fusion de données[7].

## 1.8 L'internet des objets

L'internet des objets (ou IoT) est une technologie qui permet de connecter n'importe quels ensemble d'objets du monde physique entre eux à travers l'internet et /ou des réseaux locaux comme les réseaux de capteurs sans fil (WSN), pas seulement des dispositifs électroniques, mais consiste à intégrer et embarquer des capteurs et systèmes intelligents dans les divers produits [14], pour récupérer les informations et les données (sur leur identité, leur caractéristiques et leur environnement ...). Gartner prévoit en effet que 26 milliards d'objets seront installées en 2020. [8]



FIGURE 1.7 – L'internet des objets[8].

## 1.9 Low-Power and Lossy Networks

Les réseaux de faible puissance et à pertes (LLN) sont ceux dans lesquels les noeuds et leurs interconnexions sont fortement contraints par les ressources. Les noeuds sont généralement limités en termes de puissance de traitement, de batterie et de mémoire, et leurs interconnexions sont caractérisées par des liaisons instables avec des taux de perte élevés et des débits faibles. Les modèles de trafic sont également variés, et peuvent comprendre du point à point (P2P), du point à multipoint (P2MP) ou du multipoint à point (MP2P) [9]. Ils peuvent potentiellement comprendre des milliers de noeuds, ils supportent IPv6 et peuvent s'interconnecter par une variété de technologies de communication, tels que IEEE 802.15.4 ou Wi-Fi à faible consommation. Dans la plupart des cas, les LLN seront utilisés sur des couches de liaison avec des tailles de trame restreintes.

### 1.9.1 LLN versus RCSF

Les réseaux LLN tout comme les réseaux de capteurs sans fil, l'une des principales différences entre eux c'est que les RCSF ne relie pas avec l'internet et ne supportent pas IPv6, l'internet c'est pour contrôler les dispositifs à distance, L'utilisation de IPv6 permet de garantir un espace d'adressage plus large que celui couvert par IPv4, pour la souplesse fournie qui permet l'amélioration des options et pour l'étiquetage des paquets appartenant à des « flux » particuliers.[15]

### 1.9.2 IPV6

Au début des années 1990, il est devenu clair que le développement d'internet allait aboutir à l'épuisement des adresses disponibles fournies par IPv4, ce qui a conduit à l'émergence de IPv6 après des travaux menés au sein de l'IETF au cours des années 1990.

Les changements d'IPv4 à IPv6 tombent principalement dans les catégories :

**Capacités d'adressage étendues :** IPv6 augmente la taille de l'adresse IP de 32 bits à 128 bits, donc l'augmentation de l'espace d'adressage.

**Simplification du format d'en-tête :** Certains champs d'en-tête IPv4 ont été supprimés ou rendus facultatifs pour réduire les coûts et pour limiter le coût de la bande passante de l'en-tête IPv6. Prise en charge de l'amélioration des extensions et des options : Les modifications apportées à la manière dont les options d'en-tête IP ont encodé plus efficacement, des limites moins strictes sur la longueur des options et une plus grande souplesse pour introduire des nouvelles options à l'avenir.

**Capacité d'étiquetage du débit :** Une nouvelle capacité est ajoutée pour permettre l'étiquetage des paquets appartenant à des « flux » particuliers de trafic pour lesquels l'expéditeur requiert un traitement spécial, comme la qualité non service ou « temps réel ».

**Fonctions d'authentification et de confidentialité :** Des extensions pour prendre en charge l'authentification, l'intégrité et la confidentialité des données.

### 1.9.3 Description de la norme IEEE 802.15.4

L'institut des ingénieurs électriciens et électroniciens (ou l'IEEE) soutient de nombreux groupes de travail pour développer et maintenir des normes de communications sans fil et filaires. Par exemple, 802.3 pour les câbles ethernet et 802.11 pour les LAN sans fil (WLAN). Le groupe de normes 802.15 spécifie une variété de réseaux personnels sans fil (WPAN) tel que la catégorie IEEE 802.15.4 qui est très utilisé dans les protocoles de communication des réseaux de capteurs sans fil, caractérisé par des débits de données trop faible de 250 kb/s, 40 kb/s et 20 kb/s, de basse consommation énergétique (inférieure à 0,01 mA en mode veille), et de faible portée jusqu'à 100 m a été développée pour les applications de surveillance et de contrôle à faible débit de données et pour les utilisations à faible consommation d'énergie à longue durée de vie. [10], en raison de sa caractéristique ils ont choisi comme l'un des supports de communication utiliser pour les réseaux LLN.

## 1.10 6LOWPAN

Les réseaux IPv6 personnels sans fil de faible puissance (6LoWPAN) : est un groupe de travail crée par l'IETF afin de permettre l'adoption de IPv6 par les réseaux LLN. En effet, puisque l'adoption de l'IPv6 par les réseaux LLN a fait très vite l'unanimité à cause des milliards d'objets potentiels à connecter à internet, un autre problème s'est posé, celui de la taille des trames très petites définis par les protocoles de communications des couches sous-jacentes. Si on prend pour exemple la norme IEEE802.15.4, la taille des trames est uniquement de 127 octets alors que les paquets IPv6 ont une taille minimale de 1280 octet. Il est donc nécessaire de définir une couche d'adaptation afin de permettre au paquet IPv6 de transiter à travers des trames de taille beaucoup plus réduites. C'est l'IETF qui a défini la couche adaptative 6LoWPAN afin de résoudre cette incompatibilité [11]. 6LoWPAN se base principalement sur deux mécanismes afin de réduire la taille des datagrammes IPv6 à savoir la fragmentation et la compression des entêtes afin de permettre aux paquets IPv6 d'être envoyés ou reçus via les réseaux LLN.

### 1.10.1 Architecture 6LoWPAN

Le réseau 6LoWPAN est composé par des noeuds partagent le même préfixe IPv6 et de routeur de bord qui s'occupe de la gestion de la compression et la fragmentation d'en-têtes IPv6. On [12] distingue trois familles d'architecture dans un réseau 6LoWPAN, réseau **6LoWPAN simple** avec un seul routeur de bord (Edge router)



connecte à d'autre réseau IP, réseau **6LoWPAN étendu** avec plusieurs routeurs de bord relie par une dorsale et réseau **ad hoc 6LoWPAN** non connecté.[12]

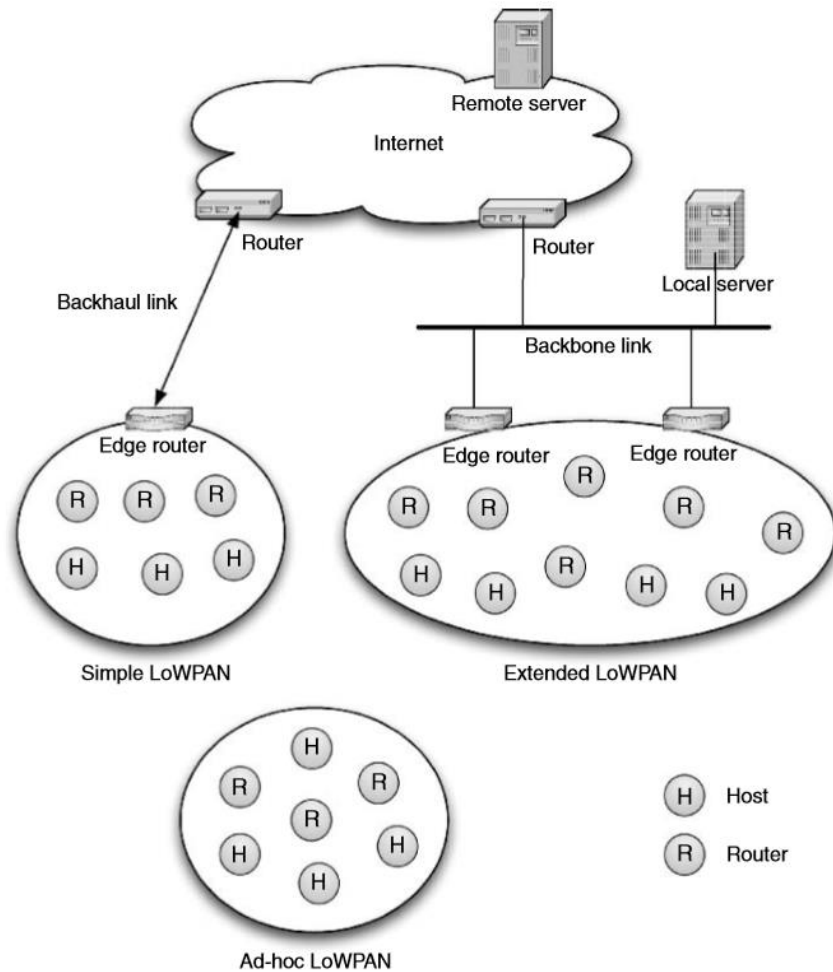


FIGURE 1.8 – Architecture 6LoWPAN [12].

### 1.10.2 Fonctionnement de 6LoWPAN

6LoWAPN c'est une couche d'adaptation situé entre la couche réseau et la couche liaison de données charge de réduire la taille des paquets IPv6 en utilisant la fragmentation et la compression.[12]

## 1.11 Le routage dans les réseaux LLNs

### 1.11.1 Contraintes de routage dans les réseaux LLN

Les réseaux LLN sont fortement contraints par les ressources, les noeuds sont généralement limités en termes de puissance de traitement, de batterie et de mémoire, [13] ces contraintes peuvent nuire le fonctionnement de routage il cause des interconnexions instables avec des taux de perte de paquets élevés et des débits faibles. Les protocoles de routage développés par le passé pour des réseaux ad hoc et les réseaux de capteurs sans fil ne s'adaptent pas à ce dernier, donc pour réussir le routage dans les réseaux LLN il faut choisir ou développer des protocoles de routage spéciales prises en compte l'exigence d'une part et sans négliger la performance d'autre part.

### 1.11.2 Algorithmes de routage dans les réseaux LLN

L'IETF a formé plusieurs groupes de travail (WG) et leur a assigné la tâche de définir des protocoles de routage d'abord pour les réseaux ad hoc avec le groupe de travail MANET et puis pour les réseaux LLN avec les groupes de travail 6LoWPAN et ROLL. [13]

#### **Le groupe de travail 6LoWPAN :**

Créé par L'IETF, dans l'objectif de créer une couche d'adaptation permettant aux paquets IPv6 d'être pris en charge efficacement par des trames de petite taille, afin d'obtenir un protocole plus léger qui maximise l'efficacité de la bande passante dans 6LoWPAN, un protocole de routage LOAD (6LoWPAN Ad Hoc On-Demand Distance Vector Routing) a été proposée par le groupe de travail 6LoWPAN, LOAD c'est un dérivé de l'AODV, mais adapté pour les adresses L2 et le routage sous maillage, et avec certaines simplifications sur AODV, après la création de ROLL le développement de LOAD a été suspendu par le groupe de travail 6LoWPAN, en attendant les résultats de ROLL et les expériences avec RPL. D'autres protocoles ont été développés par le groupe 6LoWPAN tel que DYMO-LOW, HI-LOW.[13]

**Le groupe de travail ROLL (Routing over Low-Power and Lossy Links) :**

C'est un groupe de travail créé par l'IETF dans l'objectif d'élaborer un protocole de routage pour les LLN, basé sur IPv6. A partir de 2011, le groupe de travail ROLL travaille principalement sur RPL (protocole de routage pour les réseaux à faible puissance et à perte). Ce protocole sera largement décrit dans le chapitre qui suit.

## 1.12 Conclusion

Dans ce premier chapitre, nous avons présenté les réseaux de capteurs sans fil, leurs architectures. Cependant, nous avons remarqué que plusieurs facteurs et contraintes compliquent la gestion de ce type de réseaux.

Nous avons aussi présenté le contexte de ce travail et plus particulièrement les réseaux LLN qui connectent des objets bien particuliers qui ont de fortes contraintes en termes de ressources tels que les capteurs sans fil, les sondes etc. L'interconnexions entre ces objets est également caractérisée par de faible débit et des taux de pertes importants [9]. Dans la plupart des cas, les LLN utilisent des médiums de communications à faible consommation tel que la norme IEEE 802.15.4. Ces médiums disposent de trames avec tailles restreintes. Pour permettre aux paquets IPv6 qui a une taille de 1280 octets d'être envoyés ou reçus par ces médiums de communications [6] l'IETF crée la couche d'adaptations 6LoWPAN. Ainsi nous avons décrit le routage et ses contraintes dans les réseaux LLN avec ses différentes classes. Avec la multiplication d'appareils sans fil, les groupes de travail tel que MANET ou ROLL commencent à élaborer de nouveaux protocoles de routage spécialement conçus pour répondre aux contraintes des environnements et de nombreux protocoles de routage tel que OLSR, AODV et RPL.

Dans le chapitre suivant nous décrivons le protocole de routage RPL spécialement conçu pour répondre aux contraintes des réseaux IPv6 à faible consommation et à perte (RPL) .

# Chapitre 2

## Le protocole de routage RPL

### Sommaire

---

2.1	Introduction . . . . .	24
2.2	Historique . . . . .	24
2.3	Définition . . . . .	25
2.4	Fonctionnement du protocole RPL . . . . .	26
2.5	RPL et la notion du temps . . . . .	35
2.6	Fonction Objectife . . . . .	37
2.7	Les principales métriques . . . . .	38
2.8	Sécurité dans RPL . . . . .	40
2.9	Conclusion . . . . .	41

---

## 2.1 Introduction

L'IETF a formé le groupe de travail ROLL dans l'objectif d'élaborer un protocole de routage pour les réseaux LLN, basé sur IPv6, à partir de 2008, le groupe de travail ROLL travaille principalement sur le protocole de routage pour les réseaux à faible puissance et à perte (RPL), Le protocole a été conçu pour être adapté aux contraintes des réseaux LLN . Dans ce chapitre, nous allons aborder le fonctionnement du protocole RPL en citant la procédure de construction de la topologie RPL avec leur concept clés et en citant les deux modes d'opération « Non-Storing mode » et « Storing mode », les paradigmes de la communication, suivies de la classification des travaux antérieurs sur les améliorations de RPL.

## 2.2 Historique

D'un point de vue historique, les travaux sur les protocoles de routage adaptés au LLN sont le résultat d'une évolution des usages et de l'émergence des appareils contraints. En effet, aux abords des années 2000, la multiplication des appareils mobiles sans fil a entraîné la création de nouveaux protocoles de routage spécialement conçus pour prendre en compte la contrainte de mobilité. Au niveau standardisation, l'IETF a donc créé le groupe de travail (Mobile Ad-hoc Networks ) (MANET) en 1998 pour définir des protocoles adaptés aux enjeux du sans fil.[15]

Ces protocoles de routage ont été pensés en ayant à l'esprit des réseaux composés de nœuds mobiles échangeant un grand volume de données et cela sans contraintes énergétiques (voitures, smartphones, etc.). Cependant, le marché a évolué vers un usage d'appareils plus économiques, et donc plus contraints, pour des applications commerciales (capteurs connectés pour surveillance, smart-grid, etc.), entraînant l'apparition des LLN. [15]

La vision initiale du groupe de travail MANET est donc devenue obsolète pour ce type de réseau et, même si en 2008, des adaptations ont été proposées aux protocoles de routage MANET pour les LLN, le groupe de travail ROLL (Routing over Low-Power and Lossy Links ) a été créé à l'IETF afin de standardiser un protocole de routage spécifique pour les LLN : RPL(IPv6 Routing Protocol for Low-Power and Lossy Networks).[15]

**. IETF :** (Mobile Engineering Task Force) L'Internet Engineering Task Force (IETF), élabore et promeut des standards Internet, en particulier les normes qui composent la suite de protocoles Internet (TCP/IP).

**ROLL :** (Routing Over Low Power and Lossy Networks) ROLL assurera une co-

ordination étroite avec les groupes de travail d'autres domaines spécialisés dans les réseaux et / ou les nœuds sous contraintes, tels que 6lo, 6tisch, ipwave, lwig et CoRE. **MANET** : (Internet Ad-hoc Mobile Network) Un "réseau mobile ad hoc" (MANET) est un système autonome de routeurs mobiles (et hôtes associés) connectés par des liaisons sans fil - dont l'union forme un graphe arbitraire. Les routeurs sont libres de se déplacer de manière aléatoire et de s'organiser de manière arbitraire. ainsi, la topologie sans fil du réseau peut changer rapidement et de manière imprévisible. Un tel réseau peut fonctionner de manière autonome ou être connecté à un réseau Internet plus vaste.

## 2.3 Définition

RPL est sans doute l'un des protocoles de routage IPv6 les plus connus pour les réseaux à faible consommation et à perte (LLN) développé par ROLL dans la RFC 6550 pour répondre aux limites des réseaux LLN telles que la faible puissance de traitement, de batterie et de mémoire. [15]

RPL vise principalement les réseaux de collecte, où les noeuds envoient périodiquement des mesures à un point de collecte . Le protocole a été conçu pour être très adapté aux conditions du réseau et pour fournir des itinéraires de rechange, chaque fois que les itinéraires par défaut sont inaccessibles. RPL fournit un mécanisme pour diffuser l'information sur la nouvelle topologie de réseau formée dynamiquement.[15]

RPL est un protocole proactif basé sur un algorithme à vecteur de distance, il est conçu de manière à détecter et à réagir aux boucles de routage. Les protocoles de routage à vecteur de distances permettent de créer des tables de routages où aucun nœud n'a de vision globale du réseau ; la diffusion des routes sur le réseau se faisant de proche en proche. Les protocoles à vecteur de distance permettent de fonctionner avec un minimum de ressources. En effet, les informations de routages à stocker correspondent seulement au voisinage, contrairement aux protocoles à état de liens qui nécessitent de connaître les caractéristiques des liens de l'ensemble des nœuds du réseau. [15]

Cette figure en dessous représente le travail de spécification effectué par le groupe de travail ROLL. Comme on peut le constater, l'environnement RPL est relativement complexe avec de nombreux RFC interagissant entre eux (les RFC pour Request For Comments représentent les standards produits par l'IETF). [16]

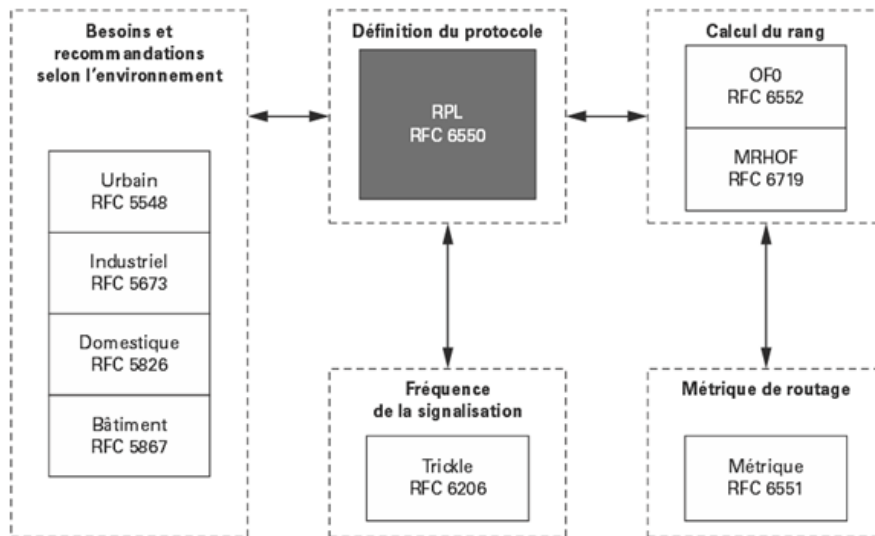


FIGURE 2.1 – Organisation de l'environnement RPL[15].

## 2.4 Fonctionnement du protocole RPL

### 2.4.1 Construction de la topologie

#### 2.4.1.1 Les graphes DAG et DODAG

DAG (graphe orienté acyclique, en Anglais (Directed Acyclic Graph)) est un graphe orienté qui ne possède pas de circuit. Il décrit les liens orientés entre les nœuds, se terminant à un ou plusieurs nœuds racines. RPL s'appuie sur la notion de DODAG ( graphe acyclique orienté vers la destination, en Anglais (Destination Oriented Directed Acyclic Graph), DODAG est un DAG a une seule destination à la racine c'est- à-dire à une seule racine DAG.[15]

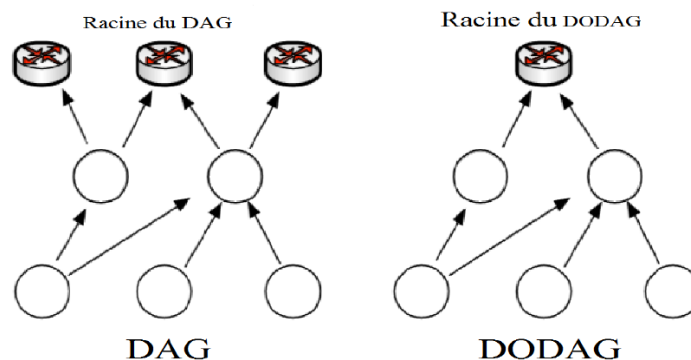


FIGURE 2.2 – les graphes DAG et DODAG[15].

### 2.4.1.2 Le modèle réseau

RPL définit trois types de nœuds :

- **DODAG root ou Low power and lossy Border Router (LBR) :**

Il se réfère à la racine d'un DODAG qui représente un point de collecte dans le réseau et a la capacité de construire un DAG. Le LBR agit également comme une passerelle (routeur edge) entre l'internet et le LLN, il peut avoir des racines multiples configurées dans le réseau. [16]

- **Nœud Routeur (ou stateful node) :**

Il se réfère à un périphérique qui peut transmettre et génère du trafic des voisins. Un tel routeur n'a pas la capacité de créer un nouveau DAG. [16]

- **Nœud host (ou stateless node) :**

Il se réfère à un périphérique terminal capable de générer des données, mais qui n'est pas capable de router le trafic. [16]

### 2.4.1.3 Messages de contrôle du protocole RPL

Le protocole RPL comporte quatre types de messages de contrôle utilisés dans la phase de découverte de routes. Ces messages sont :

**DIO :** DODAG Information Object (DODAGID, IDRoot, Rang du root, infos sur OCF) : envoyé de manière périodique depuis le nœud racine vers tous ses nœuds voisins.

**DAO :** Destination Advertisement Object (ID, Rang, IDs route infos), Le DAO est un message envoyé par les nœuds esclaves (capteurs) au nœud racine afin de répondre au message DIO.

**DIS :** DODAG Information Solicitation, c'est un message de sollicitation envoyé par les nœuds non voisins du nœud racine qui n'ont reçu aucun message DIO vers l'un des voisins du nœud racine. Le nœud voisin se charge du transfert des messages DIO et DAO du et vers le nœud racine.

**DAO-ACK :** C'est un acquittement du message DIO envoyé par les récepteurs. Destination Advertisement Object Acknowledgement DAO.ACK c'est un acquittement du message DIO envoyé par un nœud parent vers son fils.

- A l'exception des messages DAO et DAO.ACK, les deux autres messages ont un objectif local et sont envoyés en multicast vers les voisins en utilisant des adresses IPv6 de lien local
- Les messages DAO et DAO.ACK sont eux envoyés en unicast à l'aide d'adresse globale ou d'une adresse locale unique. [17].



#### 2.4.1.4 Construction du DODAG

Le concept de la construction du graphe DODAG formé par le protocole RPL est une topologie de routage logique construite à partir d'un réseau physique, utilisant un certain nombre de critères fournis par l'administrateur du système. Ci-dessous, nous présentons les principales étapes conceptuelles de construction d'un DODAG, à partir d'un réseau physique donné.[15]

##### **1ère étape :**

La topologie DODAG est initiée par le nœud racine (Root), dont il diffuse les messages DIO dans son voisinage à un saut, le message DIO incluant son instance RPL, son DODAG-ID, son numéro de version DODAG-Ver, son rang (Rank), le coût du chemin, la contrainte ou la métrique et la fonction objectif (FO). La figure 2.3 illustre cette première étape de construction.[15]

##### **Deuxième étape :**

Comme illustre la figure 2.4, chaque nœud ayant reçu le DIO envoyé par le Root rejoint le DODAG, dont il établit une liste à jour des paramètres (calcul de rang (Rank), la liste des nœuds parents mise à jour par l'expéditeur du message DIO, sélection du parent préféré (PP) sur la base de la FO et paramètres dans le DIO). Une fois le parent préféré choisi, le nœud envoie à ce parent un message DAO pour signaler son choix, que le message DAO sera envoyé au nœud racine via l'ensemble des parents pour l'itinéraire inverse. Ces messages DAO contiennent des informations permettant au Root de construire des chemins vers les nœuds feuilles du DODAG [15]. **Troisième étape :**

Les nœuds ayant choisi le Root comme parent vont à leur tour diffuser des messages DIO en multicast dans leur voisinage. Les nœuds ne faisant pas encore partie du DODAG qui vont recevoir ces messages DIO, vont donc faire un choix de nœuds parents puis répondent par des messages DAO indiquant leur attachement au DODAG. La figure 2.5 montre le principe de cette étape.[15]

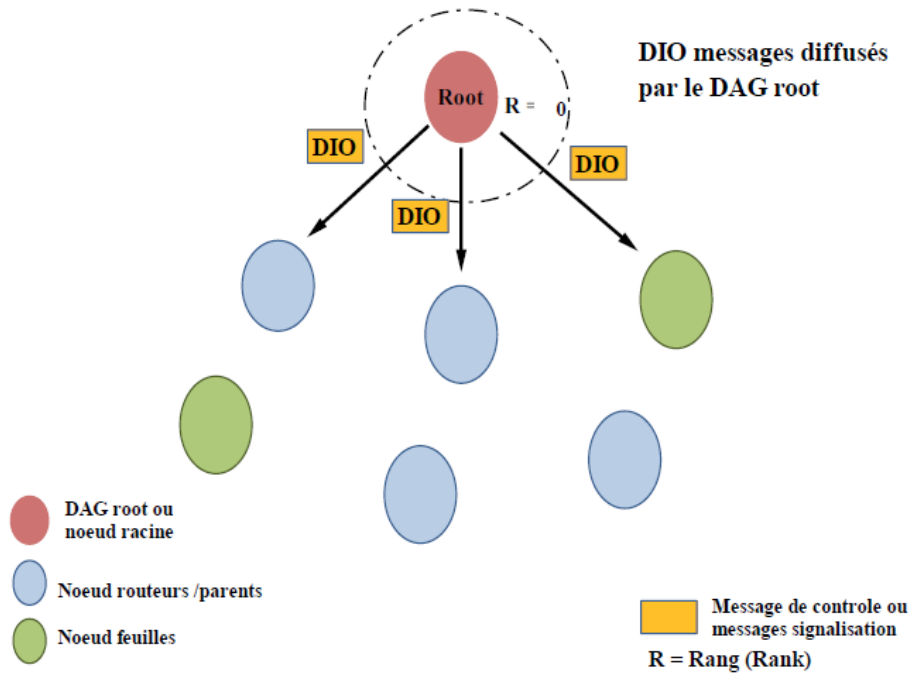


FIGURE 2.3 – Etape 1, diffusion des DIO par la racine (DAG root)[15].

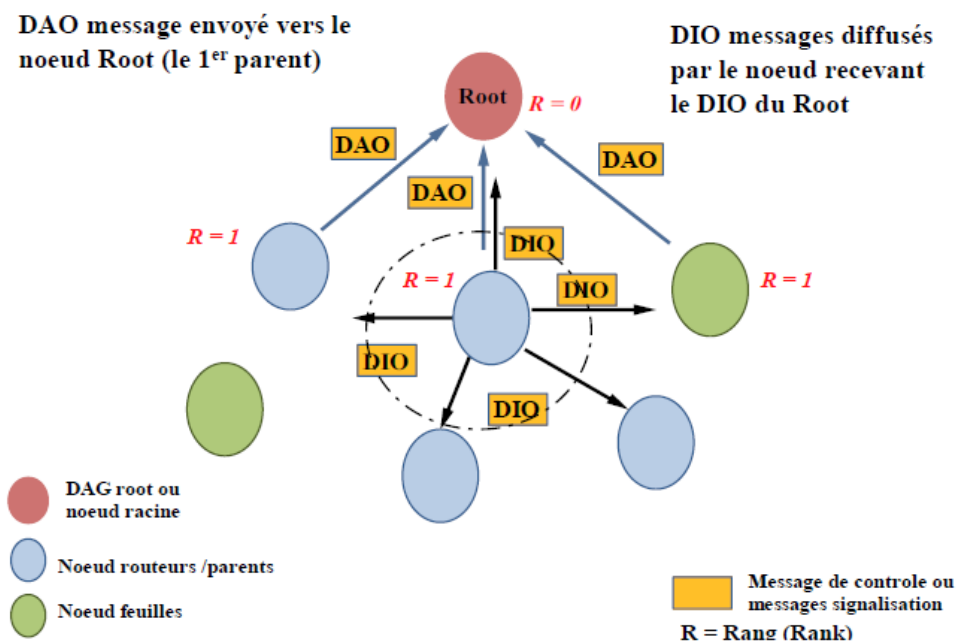


FIGURE 2.4 – Etape 2, choix du root comme noeuds parent[15]

Ensuite, s'il s'agit d'un routeur, il transmet ce message de façon multicast après la mise à jour des valeurs des champs. Un nœud rejette un message DIO, si le rang de l'expéditeur est supérieur à son rang afin d'éviter le routage en boucle (Rooting Loop) , ou s'il provient d'un numéro de version différent. De plus, un nœud doit

diffuser un message DIS pour confirmer la réception d'un message DIO, s'il ne reçoit pas de message DIO et n'appartient à aucun DODAG.[18]

D'un autre côté, un nœud utilise un message DAO indiquant "No-Path" pour supprimer un chemin vers le bas, lorsqu'il perd sa disponibilité pour cette cible. Ce processus est illustré dans l'algorithme 1. Enfin, dans le but de maintenir la topologie DODAG, RPL effectue un envoi périodique des messages DIO à l'aide de l'algorithme trickle timer, qui est ajusté suivant l'état du réseau afin d'optimiser les messages de signalisation de contrôle (overhead).[19]

RPL fonctionne sur la base d'un graphe orienté, aussi les routes qui vont vers la racine (upward) ne sont-elles pas nécessairement les memes que celles qui descendent vers les nœuds (downward) [18].

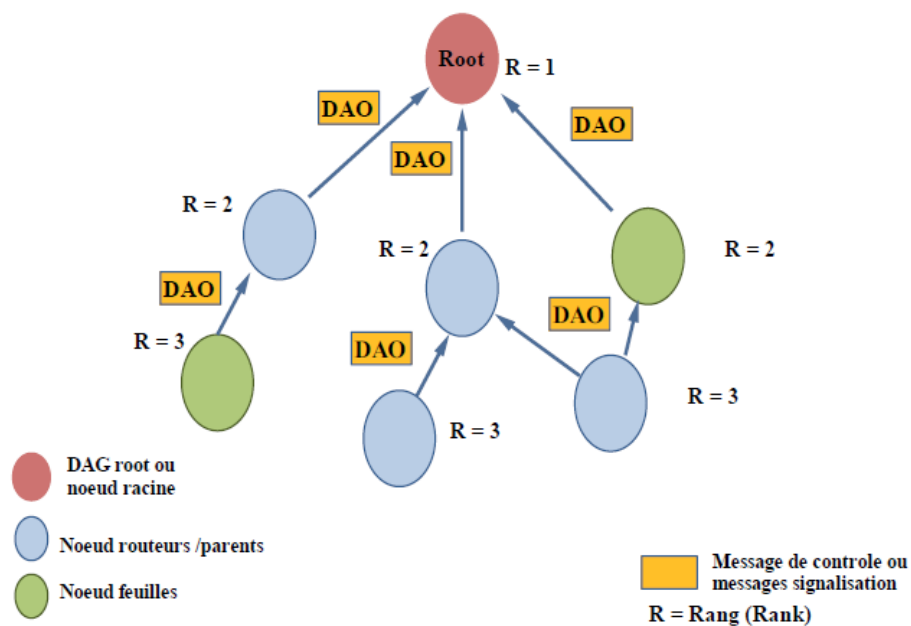


FIGURE 2.5 – Etape 3, pour suite de la construction du DODAG[15].

### 2.4.1.5 Les modes d'opération du protocole RPL

En fonction de la capacité des nœuds en terme de mémoire et de la taille éventuelle du réseau, le protocole RPL offre deux modes de fonctionnement, le fonctionnement en «storing mode» et celui en «Non- Storing mode».

Le fonctionnement en «Non-Storing mode» Dans ce mode, seul la racine est en mesure de stocker des informations de routage. Les autres nœuds du réseau conservent uniquement les adresses de leur parent direct. Toutes les informations sur la structuration du DODAG sont transmises à la racine dans les messages DAO. En cas de besoin de router des données vers une destination quelconque, les nœuds transmettent ces données à la racine en passant par leur parent. La racine effectuera un routage à la source vers la bonne destination. [18]

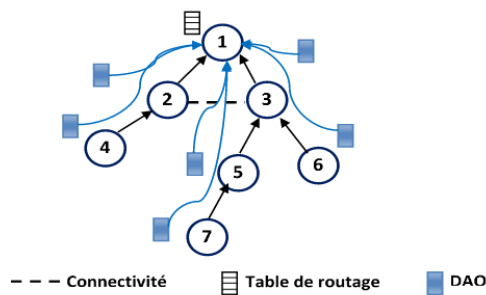


FIGURE 2.6 – Le fonctionnement en «non-storing mode»[18]

Le fonctionnement en «Storing mode» Dans ce mode, les nœuds intermédiaires sont en mesure de garder en mémoire des informations de routage puis de rediriger les données reçues vers la bonne destination en consultant les informations du routage. Contrairement au «Non-Storing mode», dans le fonctionnement en «Storing mode», les messages DAO ne sont pas tous transmis à la racine. Chaque nœud transmet son message à son parent direct (parent à un saut) qui maintient une table de routage à son niveau.

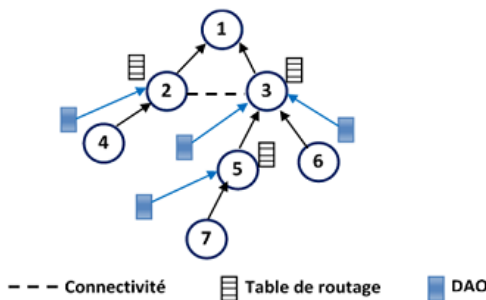


FIGURE 2.7 – Le fonctionnement en «storing mode»[18]

#### 2.4.1.6 Les paradigmes de communication

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

##### **Multipoint à point(MP2P) :**

RPL a été conçu principalement pour optimiser le type de flux de trafic multipoint à point (MP2P) [20] , cette communication MP2P a été fourni par la constructions des routes à partir de chaque nœud vers la racine DODAG à l'aide de DIO du parent préféré d'un nœud il s'agit de «Routes ascendantes» (Upward Routes)[20] ]Les destinations des flux MP2P sont des nœuds désignés qui ont une certaine importance pour l'application, tels que la fourniture de connectivité à l'Internet ou au réseau IP privé principal.[15]

##### **Point à multipoint (P2PM) :**

Il s'agit de Routes descendantes (Downward Routes) , RPL prend en charge le trafic P2MP il utilise un mécanisme de publicité de destination qui prévoit des itinéraires descendants de la racine vers d'autre nœuds (préfixes, adresses ou groupes de multidiffusion) .Par exemple les messages DIO,P2PM est le modèle de trafic requis par plusieurs applications LLN ([RFC5867], [RFC5826], [RFC5673] et [RFC5548]).[25]

##### **Point à point(P2P) :**

Pour le trafic P2P, la construction des routes ça dépend de mode de fonctionnement du protocole RPL. Si le cas de mode Non-Storing mode le paquet dirige vers une racine, ensuite la racine effectuera le routage vers la destination, si le cas de mode Storing mode, le paquet s'écoule vers la racine jusqu'à ce qu'il atteint un ancêtre qui a une route connue vers la destination. Cet ancêtre commun peut être la racine DODAG. Dans d'autres cas, il peut s'agir d'un nœud plus proche de la source ou de la destination.[19]

#### 2.4.1.7 Gestion du réseau RPL

RPL est un protocole proactif, et ce, pour faire face au changement de topologie, qui est causé par des incohérences soudaines telles que la défaillance du nœud ou la boucle de routage.

Ses caractéristiques et son concept de topologie aident à fournir des stratégies pour gérer le changement de topologie et améliorer son impact sur les performances du réseau [21].

## Auto-récupération

RPL peut gérer le changement de topologie grâce à l'auto-récupération, qui consiste à mettre à jour de manière dynamique les décisions de routage et à modifier la topologie. Ce processus peut être établi par l'envoi périodique des messages DIO basés sur le Trickle timer, le mécanisme de découverte de voisins utilisé pour vérifier la disponibilité de certains routeurs, ou par le concept de topologie qui permet de changer dynamiquement les chemins de routage avec un autre parent ou nœud frère de même rang.[19]

### Boucle de routage (Routing Loop)

RPL prend en charge l'évitement de l'occurrence de la boucle de routage(Routing Loop) dans le processus de construction de la topologie, grâce au paramètre rang (Rank). Ce paramètre est utilisé pour organiser la topologie hiérarchique, dont un nœud a un rang supérieur à tous ses rangs parents. Par conséquent, le rang est augmenté dans la direction aval, dans le but de permettre l'acheminement de paquets dans une direction, sans retour au nœud émetteur.

## 2.4.2 Maintenance de la topologie

Afin de réagir aux éventuelles pannes du réseau (lien ou noeuds non fonctionnels), RPL dispose de deux mécanismes : la réparation globale et la réparation locale.

### 2.4.2.1 Réparation globale :

La réparation globale est déclenchée par la RPL racine. Pour cela, elle incrémente son numéro de version DODAGVersionNumber ce qui entraîne la construction d'une nouvelle topologie en autorisant tous les noeuds du réseau à pouvoir choisir une nouvelle position dans le DODAG sans être contraint par leur rang dans l'ancienne version du DODAG.[15]

### 2.4.2.2 Réparation locale :

RPL autorise aussi une réparation plus fine appelée la réparation locale. Ce mécanisme est utilisé par un noeud lorsqu'il n'a plus de parents potentiels de disponibles et permet à un noeud de choisir un parent préféré sans prendre en compte son propre rang. La réparation locale s'accompagne d'un empoisonnement de routes,

afin d'avertir les fils du noeud qu'il n'est plus en mesure d'offrir une connectivité vers la racine. Pour empoisonner, un noeud envoie un DIO contenant un rang infini. Les fils à la réception du DIO avec rang infini ont alors connaissance du fait que leur parent actuel ne peut plus être considéré comme un parent et déclenchent alors la sélection d'un nouveau parent.[15]

La figure 2.8 montre un exemple de réparation locale. (La figure 2.8 a) montre les routes installées par RPL, ainsi que la connectivité de la topologie. Le noeud 1 détecte que sa route montante vers la racine n'est plus fonctionnelle(figure2.8 b).[15]

Il commence alors l'empoisonnement en émettant un DIO avec un rang infini. À la réception de ce DIO, son fils (le noeud 3) voit qu'il ne peut plus continuer à utiliser le noeud 1 comme parent préféré. Il va alors essayer de trouver un autre parent, en l'occurrence le noeud 4 sur la figure 2.8 c. Une fois que le noeud 1 a terminé son empoisonnement, il peut sélectionner potentiellement n'importe quel noeud du réseau avec qui il a une connectivité comme parent (les noeuds 2 et 3 sur notre topologie), puisqu'il n'est plus contraint par son rang. Sur la figure 2.8 c, le noeud 1 a choisi le noeud 3 comme parent préféré.

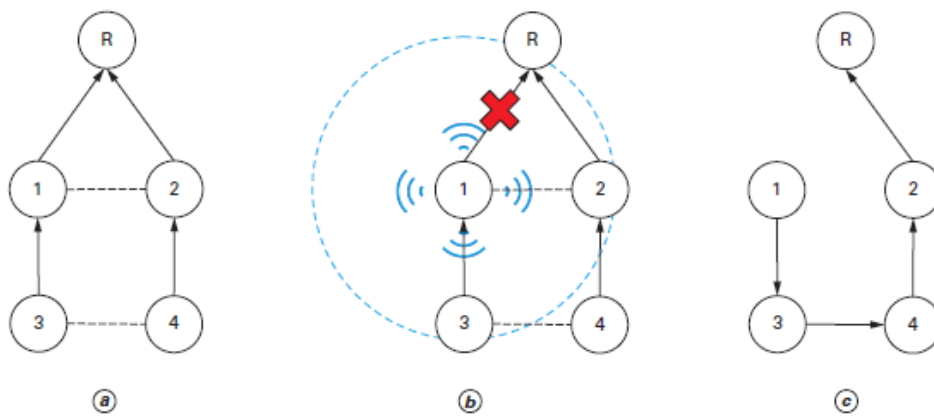


FIGURE 2.8 – Exemple de réparation locale[15][15].

## 2.5 RPL et la notion du temps

### 2.5.1 Le fonctionnement de l'algorithme Trickle Timer

Le protocole RPL utilise la fonction « Trickle Timer » pour réduire la surcharge des messages de contrôle et utiliser les ressources limitées de manière plus optimale.[15]

Il existe trois paramètres configurables dans la minuterie « Trickle Timer » :  $I_{min}$ ,  $I_{max}$  et la constante de redondance  $k$  et expliquées ci-dessous.

#### Constante de redondance ( $k$ )

C'est un nombre naturel supérieur à 0 et est utilisé pour supprimer la transmission DIO si le compteur de redondance «  $C$  » dépasse la valeur de ce paramètre.

#### $I_{max}$

Lorsque la période d'écoute maximal est atteinte, les DIO sont transmis au taux égal à la période d'écoute maximal.

#### $I_{min}$

Ce paramètre donne le minimum de temps entre deux DIO. La valeur de démarrage de « Trickle Timer » commence à partir de cette paramètre, Si un noeud entend des mises à jour DIO de ses voisins qui sont cohérentes avec sa propre compréhension de la topologie de réseau, le compteur de redondance «  $c$  » incrémente. Si le nombre de mises à jour cohérentes entendues dans une période d'écoute particulier dépasse le nombre de redondances «  $K$  », le noeud ne transmet aucune mise à jour et la période d'écoute est doublée jusqu'à ce qu'il atteigne une période d'écoute maximal ( $I_{max}$ ), Toutefois, si une mise à jour incohérente est entendue, « Trickle Timer » est réinitialisé ces paramètres et une mise à jour est rapidement propagée.

#### Description de l'algorithme « Trickle Timer » [15][15]

Le fonctionnement de l'algorithme Trickle timer peut être décrit comme suit :  
**Etape 1 :** « Trickle Timer » commence son premier intervalle en ajustant  $I$  à une valeur de la plage  $[I_{min}, I_{max}]$ , généralement il définit le premier intervalle à une valeur de  $I_{min}$ .

**Etape 2 :** Lorsqu'un intervalle commence, « Trickle Timer » réinitialise le compteur  $c$  à 0, Si la taille du nouvel intervalle dépasse la longueur d'intervalle maximum,  $I_{max}$ . « Trickle Timer » définit l'intervalle  $I$  à  $I_{max}$ , et attribue une valeur sélectionnée au hasard dans l'intervalle à une variable nommé  $t$ , choisie dans la plage  $[I/2, I)$ .

**Etape 3 :** Lors de la réception d'un message cohérent, « Trickle Timer » augmente



son compteur d'une valeur de 1.

**Etape 4 :** Au moment choisis au hasard  $t$ , si le compteur  $c$  est supérieur ou égal à la constante de redondance  $k$ , « Trickle Timer » supprime son message planifié. Sinon, le message est transmis.

**Etape 5 :** Lorsque l'intervalle  $I$  expire, « Trickle Timer » double la taille de l'intervalle et revient à l'étape 2.

**Etape 6 :** Si « Trickle Timer » détecte un message incohérent « Trickle Timer » définit  $I$  sur  $I_{\min}$ , et commence à l'étape 1. [15]

**Code source du programme**

```

1. Initialisation
   I <-- I min
2. Démarrer un nouvel intervalle
   c <-- 0
   if I max <= I then
     I <-- I max
   end if
   t <-- random [I/2, I]
3. Transmission cohérent reçue
   c <-- c +1
4. La variable aléatoire t expire
   if c < k then
     transmission de DIO
   else
     suppression de DIO
   end if
5. L'intervalle I expire
   I <-- I 2
6. Transmission incohérente reçue
   I <-- I min
Recommencer depuis l'étape 2 .

```

De façon générale, « Trickle Timer » permet au protocole RPL de réduire le trafic de contrôle lorsque le réseau est consistant et stable puis d'envoyer plus de messages en cas d'inconsistance. [15]

## 2.6 Fonction Objectife

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 du noeud du candidat voisin. Un noeud 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 noeud 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 :[22]

### 2.6.1 Objective Function Zéro

Le but de OFO est pour un nœud de sélectionner le DODAG qui offre une assez bonne connectivité à un ensemble spécifique de nœuds (DODAG à joindre) bien qu'il n'y ait aucune garantie que le chemin sera optimisé en fonction d'une métrique spécifique. OFO sélectionne son parent préféré et un successeur possible si un est disponible. Tout le trafic ascendant est normalement acheminé via le parent préféré sans tentative d'équilibrage de la charge. Lorsque les conditions de liaison ne permettent pas la transmission ascendante à travers le parent préféré, le paquet est passé au successeur possible sélectionné un peu plus tôt.

### 2.6.2 Minimum Rank with Hysteresis Objective Function

Le MRHOF est la fonction objective par défaut à utiliser dans la formation de DAG en RPL. Cette fonction permet à un nœud de déterminer le taux de transmission avec succès ainsi que l'énergie disponible dans les nœuds.[21]

MRHOF, est conçu pour trouver les chemins ayant le plus petit coût tout en empêchant un taux de désabonnement excessif dans le réseau. Il le fait en utilisant deux mécanismes. D'abord, il trouve le chemin de coût minimum.

Deuxièmement, il ne bascule vers ce chemin de rang minimum que s'il est plus court que le chemin actuel d'au moins un seuil donné. Ce deuxième mécanisme est appelé "hystérésis". Dans MRHOF la sélection des parents est améliorée et cette sélection se produit dans le cas suivant :

- Pendant la formation initiale du réseau.
- Lorsque le coût du chemin vers les noeuds voisins change.
- Un nouveau noeud apparait dans le voisinage du noeud.

## 2.7 Les principales métriques

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. Dans ce mémoire, nous nous concentrons sur cinq indicateurs pour étudier les performances des fonctions Objectifs de RPL. Les résultats donneront une idée de quelle fonction objectif est la plus adaptée à quel type de scénario. Dans cette section, nous décrivons tous les métriques que nous avons trouvées :

### 2.7.1 Control Traffic Overhead

C'est le nombre total de messages de contrôle transmis par les noeuds afin de créer un DODAG. Les messages de contrôle contiennent les messages DAO, DIS et DIO. Il peut être calculé comme suit :[20]

$$\text{Control Traffic Overhead} = \sum_1^n DIO + \sum_1^n DIS + \sum_1^n DAO$$

La stabilité du réseau peut être déduite par cet indicateur, au plus sa valeur est faible au plus le réseau est stable. Le contrôle du trafic aérien a un impact direct ou indirect sur la consommation de ressources du réseau. Par exemple, lorsque le réseau devient plus dense, il permet d'avoir une congestion et une collision entre les paquets, ce qui rend le réseau plus lent et donc moins performant. Ce qui est problématique car cela pousse les noeuds à envoyer plus de messages afin de vérifier la disponibilité du réseau ce qui inévitablement les force à dépenser plus de puissance et consommer plus de ressources.

### 2.7.2 Lost Packet

C'est le nombre de paquets perdus pendant la transmission des messages entre les noeuds. Il peut être calculé comme suit :

$$\text{Packet Lost} = \sum_1^n \text{Sent Packets} - \sum_1^n \text{Received Packets}$$

### 2.7.3 Node Energy

C'est l'énergie mesurée sur les nœuds du réseau par rapport à la durée de vie du réseau. La formule utilisée pour calculer l'énergie des nœuds est :[20]

$$\begin{aligned} \text{Energy (mJ)} = & (\text{Transmit} * 19.5 \text{ mA} + \text{Listen} * 21.5 \text{ mA} + \text{CPU.time} * 1.8 \text{ mA} \\ & + \text{LPM} * 0.0545 \text{ mA}) * 3 \text{ V} / (32768) \end{aligned}$$

La consommation d'énergie est l'un des enjeux majeurs des réseaux de l'IoT, la réduire est alors une des priorités de la fonction objective. L'OF peut alors sélectionner l'itinéraire vers la racine en fonction de la consommation énergétique de ses parents. Cette métrique est donc la principale à prendre en compte dans le contexte d'une application efficace au niveau énergétique. [20]

### 2.7.4 Hop Count (HC)

La fonction Objectif peut être basée sur cette métrique pour sélectionner le meilleur parent car HC représente le nombre de sauts entre un noeud et sa racine. En effet au plus le nombre de sauts est faible au plus le noeud est proche de la racine. Une application qui doit être effectuée en temps réel devrait donc considérer le nombre de saut comme métrique de base.[20]

### 2.7.5 Expected Transmission Count (ETX)

Il se réfère au nombre de retransmission Nécessaire pour qu'un paquet soit reçu avec succès à destination. La valeur d'ETX peut donc donner des informations sur l'état du réseau et sa stabilité. Au lieu du nombre de sauts, les fonctions objectives peuvent se baser sur cette métrique pour sélectionner les chemins vers la racine. En effet au plus ETX est faible au plus les liaisons du réseau sont stables car une

valeur élevée est synonyme de beaucoup de retransmissions et donc une consommation énergétique élevée. La valeur d'ETX peut être calculée en utilisant la formule suivante : [20] Le « Df » représente un taux de livraison, c'est la probabilité mesurée

$$ETX = \frac{1}{Df * Dr}$$

qu'un paquet soit reçu par un voisin. Le « Dr » est un rapport de livraison dit inverse, c'est en fait la probabilité mesurée qu'un paquet d'accusé de réception soit reçu avec succès.

## 2.8 Sécurité dans RPL

Le protocole RPL permet de garantir la confidentialité et l'intégrité des messages si nécessaire, il possède trois modes de sécurité : [15]

### **Non sécurisé :**

Les messages de signalisation RPL sont envoyés sans mécanisme de sécurité. Ce mode ne signifie pas forcément que le réseau RPL formé est non sécurisé. En effet, un autre type de sécurité peut être utilisé (sécurité au niveau liaison de données par exemple) . [15]

### **Préinstallé :**

Dans ce mode, les noeuds joignant une instance RPL ont des clés préinstallées permettant de générer et de traiter les messages sécurisés . [15][15]

### **Authentifié :**

Dans ce mode, les noeuds ont des clés préinstallées comme dans le mode « préinstallé », mais les clés préinstallées sont seulement utilisées pour que le noeud joigne la RPL Instance comme feuille (une feuille peut seulement envoyer du trafic sur le LLN, elle ne peut pas router du trafic). Pour joindre une instance RPL opérant en mode « authentifié », il faut obtenir une clé d'une autorité d'authentification. RPL ne définit pas les processus nécessaires pour obtenir cette clé. [15]

## 2.9 Conclusion

Dans ce chapitre nous avons présenté le protocole RPL en cours de standardisation au sein du groupe de travail de L'IETF. Nous avons représenté une description détaillée de ce protocole : son fonctionnement, la construction et le trafic supporté par le DODAG.

Dans le chapitre suivant, nous proposons une combinaison de deux métriques afin d'améliorer les performances du réseau, notamment la durée de vie du réseau et le nombre de paquets reçus en utilisant un sink mobile.

# Chapitre 3

## Améliorations des performances de RPL

### Sommaire

---

3.1	Introduction . . . . .	43
3.2	Outils de la simulation . . . . .	43
3.3	Environnement de développement . . . . .	46
3.4	Evaluation de RPL . . . . .	47
3.5	Métriques de la simulation . . . . .	47
3.6	Amélioration proposée . . . . .	48
3.7	Résultat de la simulation . . . . .	49
3.8	Simulation d'une défaillance partielle de réseau . . . . .	57
3.9	Conclusion . . . . .	59

---

## 3.1 Introduction

Le protocole RPL présenté dans le chapitre précédent est l'un des protocoles standards de communication. Ce protocole a été largement utilisé dans les réseaux de capteurs sans fil et l'internet des objets puisqu'il est adapté aux systèmes à ressources limitées et qui présentent certaine hétérogénéité.

Dans ce chapitre, nous évaluons le protocole RPL selon certaines métriques de routage : ETX et l'énergie consommée. Puis, nous proposons une métrique de routage pour améliorer les performances de RPL. Cette métrique combine les deux métriques de routage ETX et l'énergie consommée. Pour ce faire, nous avons utilisé des outils logiciels spéciaux pour les systèmes à ressources limitées tels qu'un système d'exploitation léger (Contiki) et le simulateur COOJA.

## 3.2 Outils de la simulation

### 3.2.1 Contiki

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 est écrit dans la langue de programmation C. Contiki est un système d'exploitation hautement portable et il a déjà été porté sur plusieurs plateformes 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.[22]

Le principal avantage de la 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. Il s'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. [22]



### 3.2.2 Architecture

Contiki est développé en langage C et il est constitué d'un noyau, de bibliothèques, d'un ordonnanceur et d'un jeu de processus. Comme tout système d'exploitation, son rôle est de gérer les ressources physiques telles que le processeur, la mémoire et les périphériques informatiques (d'entrées/sorties) (voir figure 3.1) Il fournit ensuite aux applications des interfaces permettant d'utiliser ces ressources. Conçu pour les modules de capteurs sans fil il occupe peu d'espace en mémoire et permet une consommation électrique très faible.[22]

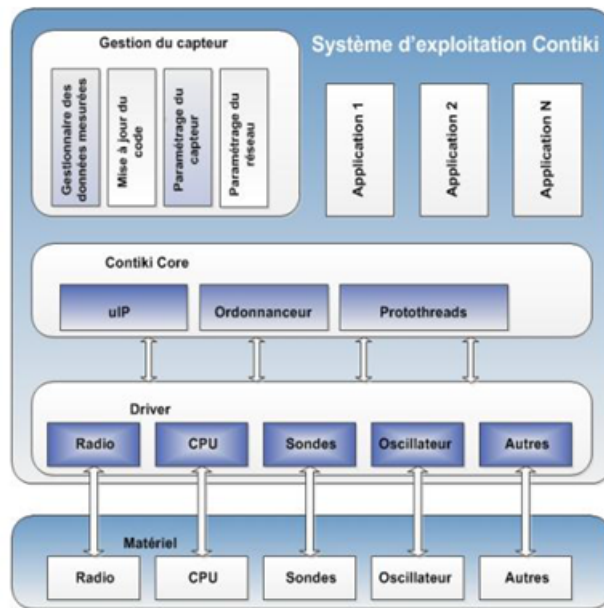


FIGURE 3.1 – Architecture de Contiki[22].

### 3.2.3 Connectivité

Contiki offre deux types de connectivité comme montre la figure 3.2 :

#### 3.2.3.1 La couche Rime

Elle permet un dialogue avec les capteurs voisins ainsi que le routage.[22][22]

#### 3.2.3.2 La couche uIP

Orientée Internet, elle offre les services essentiels du protocole IP mais nécessite plus de ressources que Rime. Contiki gère les standards 6LoWPAN, RPL, CoAP. [22]

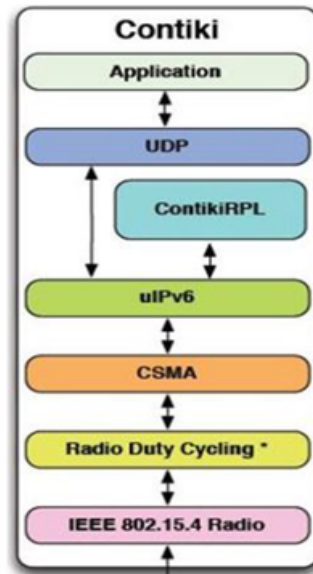


FIGURE 3.2 – Pile protocolaire de Contiki [22].

### 3.2.4 Simulateur Cooja

Contiki propose un simulateur de réseau appelé Cooja. Ce simulateur permet l'émulation de différents capteurs sur lesquels seront chargés un système d'exploitation et des applications. Cooja permet ensuite de simuler les connexions réseaux et d'interagir avec les capteurs. Cet outil permet aux développeurs de tester les applications à moindre coût.[22] Dans une simulation nous avons plusieurs fenêtres selon la figure 3.3 :

#### 3.2.4.1 La fenêtre Timeline

En bas de l'écran, nous affiche tous les événements de communication dans la simulation dans le temps, très pratique pour comprendre ce qui se passe dans le réseau.[22][22]

#### 3.2.4.2 La fenêtre Network

En haut à gauche de l'écran, Cette zone permet de visualiser chaque noeud du réseau et de visualiser leur état (identifiant, adresse, LED, etc.). A l'initialisation de la simulation, cette zone est vide et il faut lui ajouter des noeuds.[22]

#### 3.2.4.3 La fenêtre Mote Output

Sur le côté droit de l'écran, Cette zone permet d'afficher toutes les sorties des différentes interfaces des noeuds. On peut disposer d'une fenêtre "Mode Output"

différente pour chaque noeud.[22][22]

### 3.2.4.4 La fenêtre Notes

En haut à droite est l'endroit où nous pouvons mettre des notes pour notre simulation.[22]

### 3.2.4.5 La fenêtre Simulation control

C'est où nous pouvons lancer , mettre en pause et charger de notre simulation.[22]

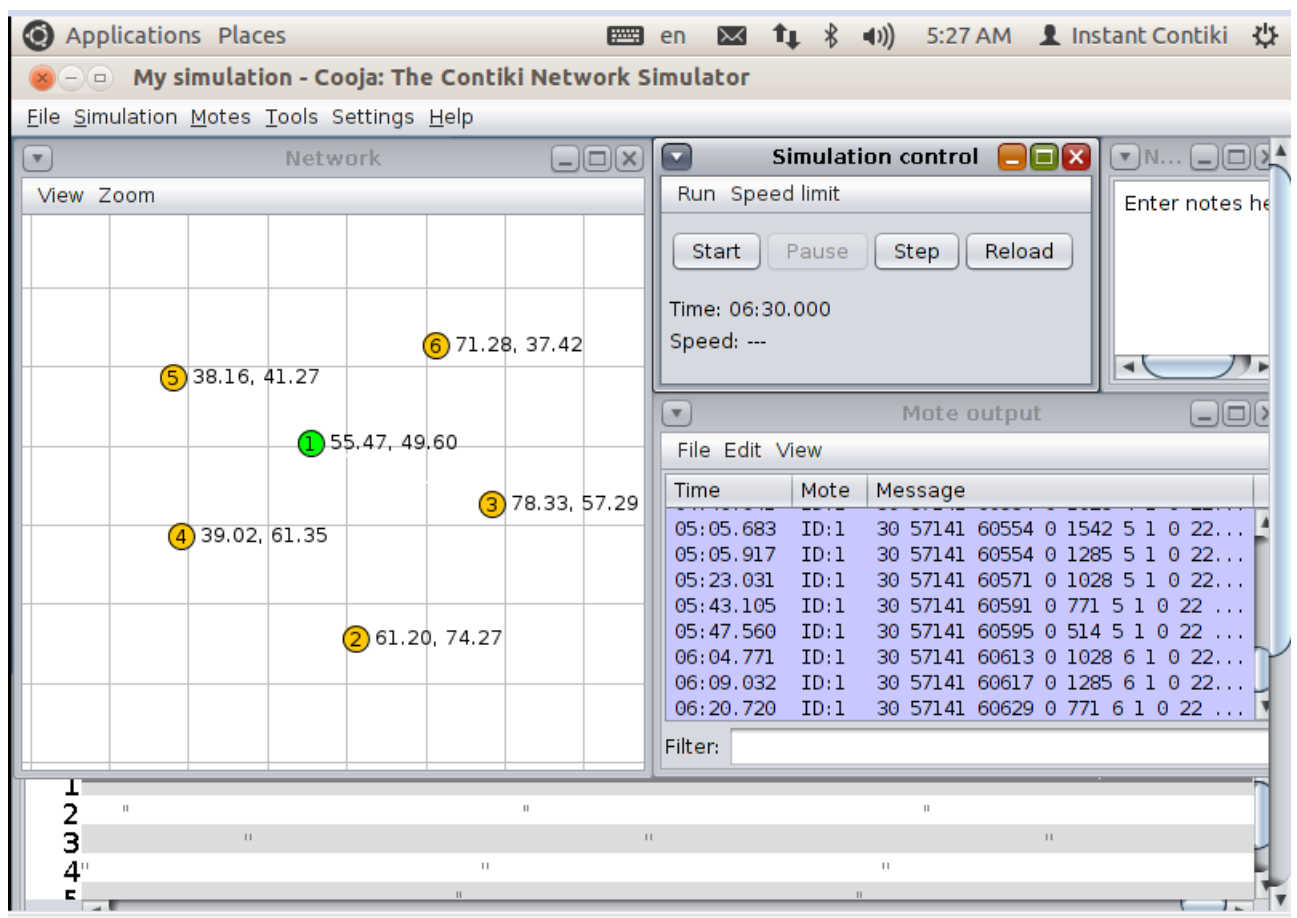


FIGURE 3.3 – Fenêtres de simulations Cooja [22].

## 3.3 Environnement de développement

Pour évaluer les performances de RPL, nous avons utilisé Contiki 2.7, langage « C » pour programmer les capteurs. La machine utilisée dans la simulation du réseau est caractérisée par les paramètres suivants :

Champ	Valeur
Processeur	Intel(R) Core(TM) i7-3632QM CPU @ 2.20GHz
RAM	6.00 Go
Disque dur	1000 Go

TABLE 3.1 – Caractéristiques de la machine utilisée pour les simulations

### 3.4 Evaluation de RPL

Dans notre simulation, on a utilisé l'exemple de rpl-collect (contiki/examples/ipv6/rpl-collect) et un environnement de simulation comme montre le tableau :

Paramètres	Valeurs
Simulateur	Cooja
Nombre de noeuds	9,16,25,36,49,20
Nombre de noeuds racines	1
Durée de la simulation (s)	750,900
Identité du noeud racine	1
Surface (mètres)	300 X 300
La fonction objective	MRHOF
Mote startup delay (ms)	1.000
Random seed	123.456
Topologie	grille ordonnée, aléatoire
Mote types	Sky mote
Radio Environment	UDGM(Distance Loss)
sink	fixe,mobile
modèle de mobilité	way point

TABLE 3.2 – Paramètres de simulation

### 3.5 Métriques de la 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 :

1. **PDR** : représente le rapport entre le nombre de paquets de données livrées à la racine et le nombre de paquets envoyés par les différents noeuds du DODAG.

$$PDR = \frac{\sum \text{Messages reçus par la racine}}{\sum \text{Messages envoyés par les noeuds du DODAG}}$$

2. **Energie** : représente l'énergie consommée par l'ensemble des noeuds du réseau.

## 3.6 Amélioration proposée

Pour améliorer les performances de RPL, nous avons proposé de combiner les deux métriques de routage : Energie et ETX avec la mobilité .

### 3.6.1 Démarche à suivre

Tout d'abord, nous avons ajouté la nouvelle métrique définie RPL-DAG-MC-EN-ETX dans le fichier rpl-con.h (contiki/core/net/rpl/rpl-con.h).

Enfin, nous avons modifié dans les fichiers rpl-dag.c (contiki/core/net/rpl/rpl-dag.c), et la grande modification dans le fichier mrhof.c (contiki/rpl-mrhof.c) la fonction objectif.

**Partie 1** : (dans le fichier mrhof.c)

```
#if RPL_DAG_MC == RPL_DAG_MC_NONE
    return p->rank + (uint16_t)p->link_metric;
#elif RPL_DAG_MC == RPL_DAG_MC_ETX
    return p->mc.obj.etx + (uint16_t)p->link_metric;
#elif RPL_DAG_MC == RPL_DAG_MC_ENERGY
    return p->mc.obj.energy.energy_est + (uint16_t)p->link_metric;

    #elif RPL_DAG_MC == RPL_DAG_MC_EN_ETX
    metric2 = p->mc.obj.energy.energy_est;
    metric1 = p->mc.obj.etx;
    metric = 50*metric1 + 50*metric2;
    p->mc.obj.cout=metric ;
    return metric + (uint16_t)p->link_metric;

#else
#error "Unsupported RPL_DAG_MC configured. See rpl.h."
#endif /* RPL_DAG_MC */
}
```

**Partie 2** : (dans le fichier mrhof.c)

```
    return MAX_PATH_COST * RPL_DAG_MC_ETX_DIVISOR;
}

#if RPL_DAG_MC == RPL_DAG_MC_NONE
    return p->rank + (uint16_t)p->link_metric;
#elif RPL_DAG_MC == RPL_DAG_MC_ETX
    return p->mc.obj.etx + (uint16_t)p->link_metric;
#elif RPL_DAG_MC == RPL_DAG_MC_ENERGY
    return p->mc.obj.energy.energy_est + (uint16_t)p->link_metric;

    #elif RPL_DAG_MC == RPL_DAG_MC_EN_ETX
    metric2 = p->mc.obj.energy.energy_est;
    metric1 = p->mc.obj.etx;
    metric = 50*metric1 + 50*metric2;
    p->mc.obj.cout=metric ;
    return metric + (uint16_t)p->link_metric;

#else
#error "Unsupported RPL_DAG_MC configured. See rpl.h."
#endif /* RPL_DAG_MC */
}
```

### 3.7 Résultat de la simulation

Les résultats des simulations sont récupérés à partir des tableaux récapitulatifs de Contiki pour tracer des courbes à l'aide du tableur microsoft excel, les courbes obtenues serviront à comparer le protocole RPL avec la fonction objective MRHOF selon les différentes métriques avec et sans sink mobile.

L'exécution des simulations a généré un ensemble de graphes en courbes qui traitent le changement d'énergie moyenne, et le PDR par rapport à la taille du réseau (nombre de nœuds variant voir figure 3.4 et figure 3.5 et figure 3.6) :

### 3.7.1 topologie sous forme d'une grille ordonnée

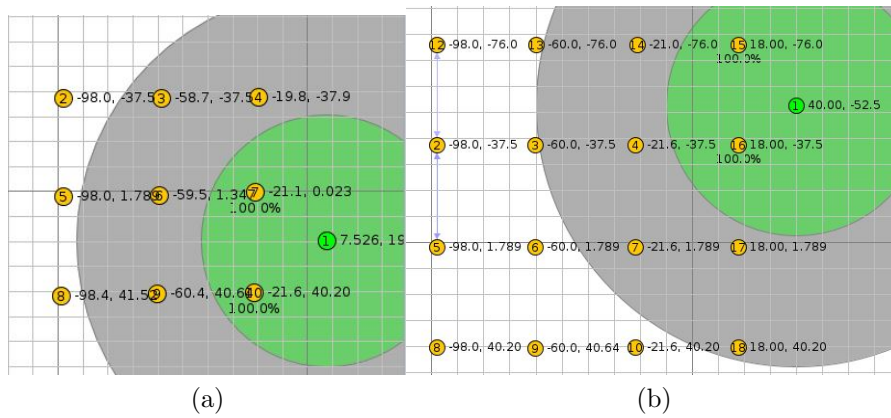


FIGURE 3.4 – Topologie : (a) 9 Nœuds , (b) 16 Nœuds

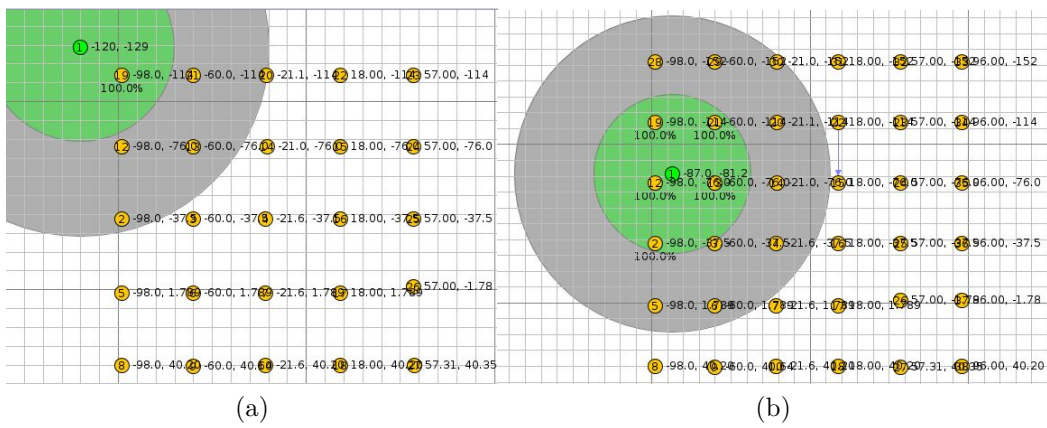


FIGURE 3.5 – Topologie : (a) 25 Nœuds , (b) 36 Nœuds

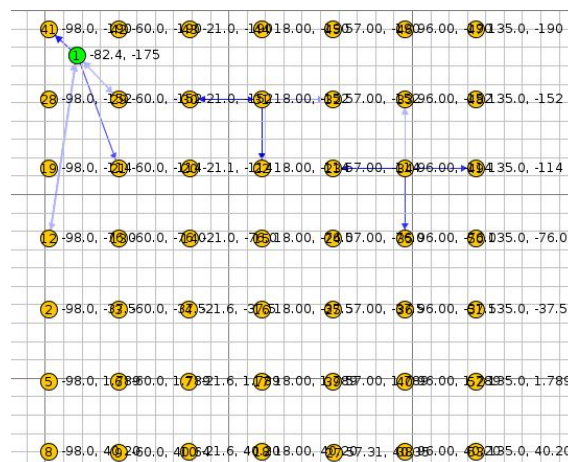


FIGURE 3.6 – Topologie 49 Nœuds

### 3.7.2 Simulation 1

Comparaison entre l'énergie consommée par un réseau avec un sink fixe et un réseau avec un sink mobile en utilisant l'énergie comme métrique de routage.

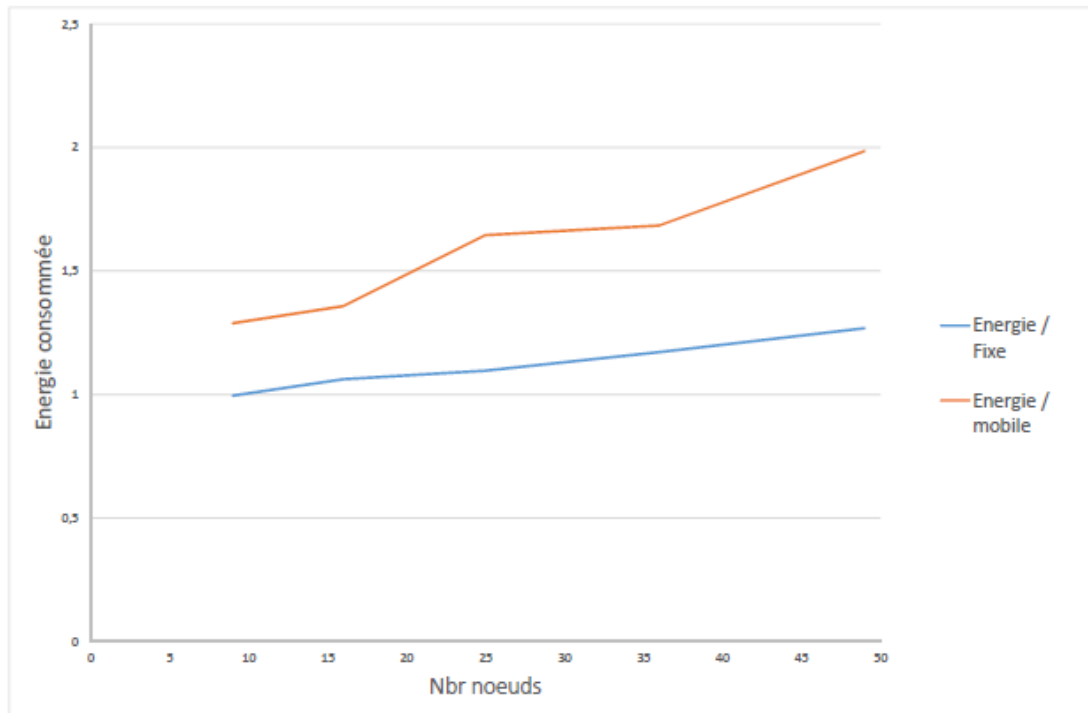


FIGURE 3.7 – Energie consommée par un réseau avec et sans sink mobile (MRHOF–Energie)

La figure 3.7 représente la consommation d'énergie en fonction de la taille du réseau. Nous remarquons que la consommation énergétique augmente dans les réseaux avec sink mobile. Cela revient au fait que les échanges des messages entre les nœuds et le sink en mouvement consomment de l'énergie.



### 3.7.3 Simulation 2

Comparaison entre l'énergie consommée par un réseau avec un sink fixe et un réseau avec un sink mobile en utilisant l'ETX comme métrique de routage.

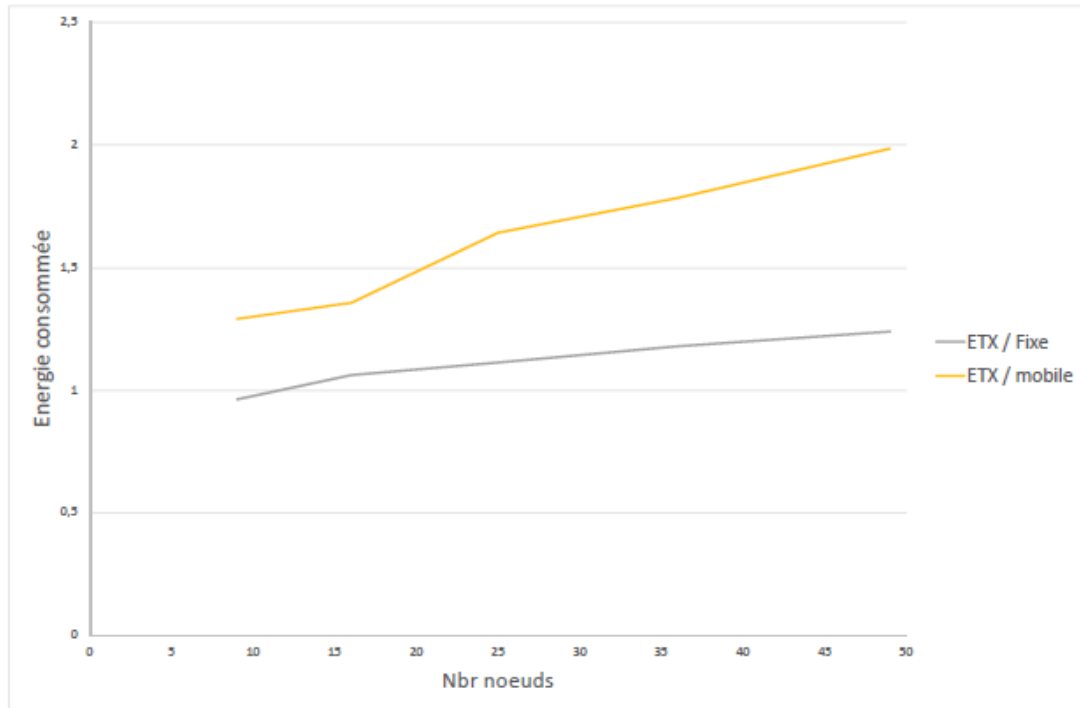


FIGURE 3.8 – Energie consommée par un réseau avec et sans sink mobile (MRHOF-ETX)

La figure 3.8 même en changeant la métrique de routage vers l'ETX (qualité de lien), Nous remarquons que la consommation d'énergie est plus importante dans un réseau avec un sink mobile que dans un réseau avec un sink fixe.

### 3.7.4 Simulation 3

Comparaison entre l'énergie consommée par un réseau avec un sink fixe et un réseau avec un sink mobile en utilisant l'énergie+ETX comme métriques de routage.

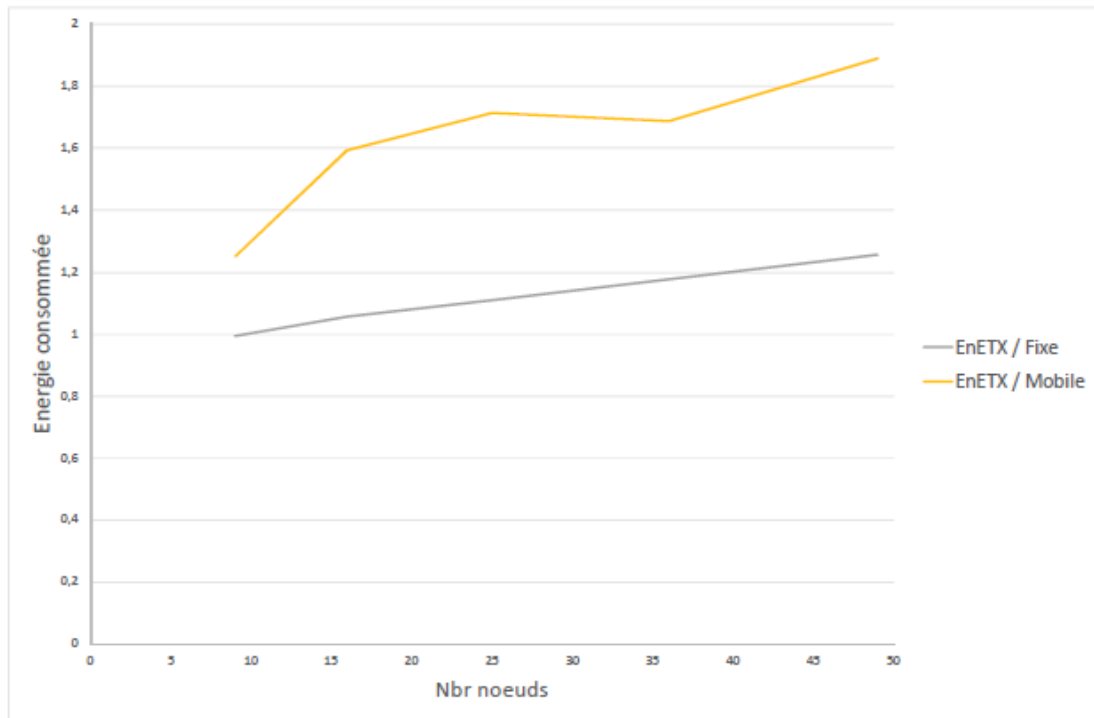


FIGURE 3.9 – Energie consommée par un réseau avec et sans sink mobile (MRHOF-Energie+ETX)

La figure 3.9 Nous avons opté une combinaison entre les deux métriques de routage l'énergie consommée et l'ETX avec une pondération de 50/50, nous avons eu le même constat, toujours le mobile est gourmand en énergie par rapport au fixe. A partir de là on peut confirmer notre première remarque qui consiste sur le fait que les échanges des messages entre les nœuds et le sink en mouvement consomme beaucoup d'énergie.

### 3.7.5 Simulation 4

Comparaison entre l'énergie consommée par un réseau avec un sink fixe en utilisant l'énergie, l'ETX et l'énergie+ETX comme métriques de routage.

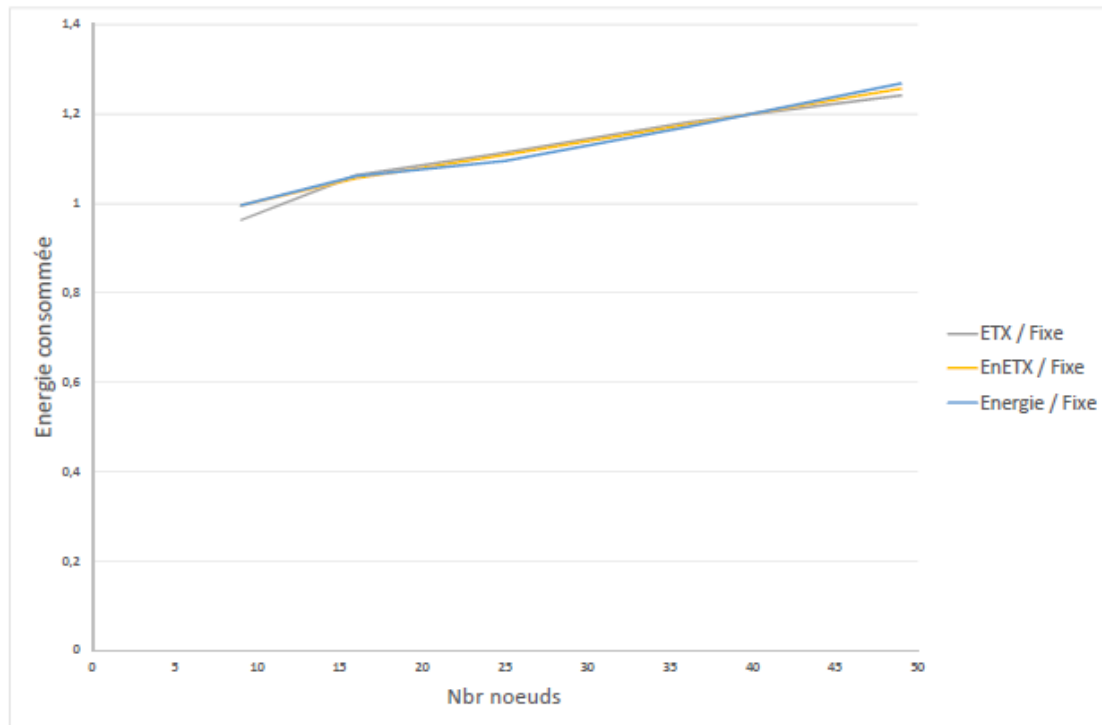


FIGURE 3.10 – Energie consommée par des réseaux avec sinks fixes (MRHOF-Energie-ETX-Energie+ETX)

Les résultats illustrés dans la figure 3.10 montrent clairement que grâce à l'utilisation de la double métriques en routage on trouve facilement un compromis entre énergie et qualité du lien en matière de consommation d'énergie dans un réseau avec sink fixe.

### 3.7.6 Simulation 5

Comparaison entre l'énergie consommée par un réseau avec un sink mobile en utilisant l'énergie, l'ETX et l'énergie+ETX comme métriques de routage.

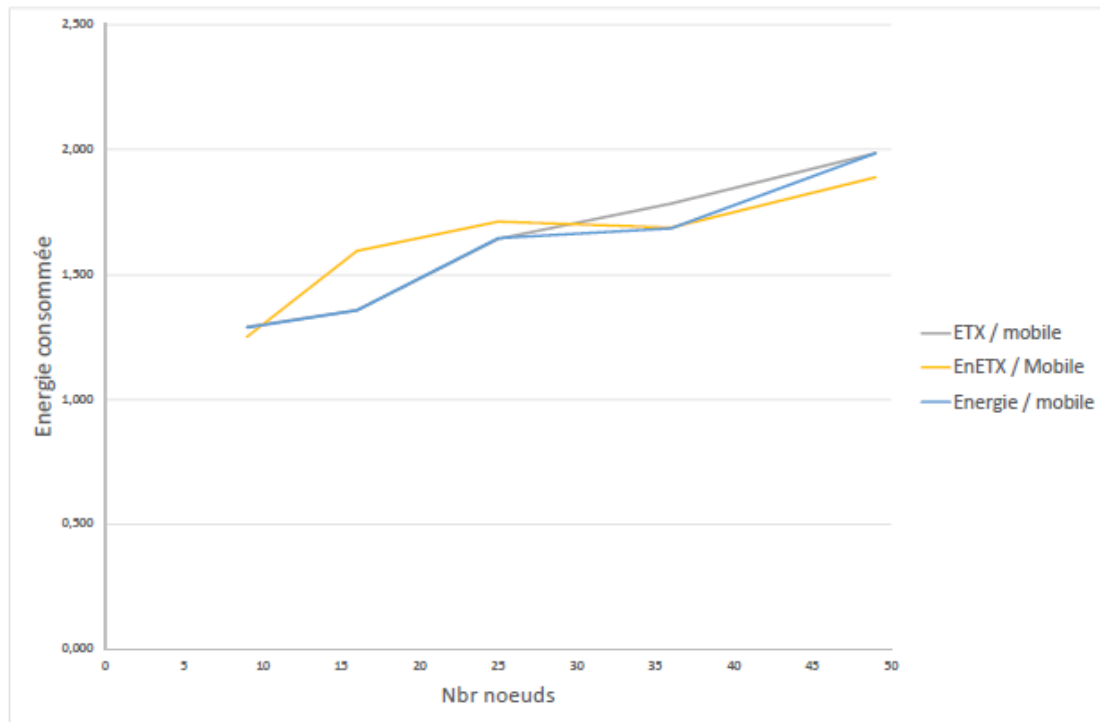


FIGURE 3.11 – Energie consommée par des réseaux avec sinks mobiles (MRHOF-Energie-ETX-Energie+ETX)

Les résultats de la figure 3.11 montrent que la fonction objective basée sur la combinaison de l'énergie et l'ETX est bonne pour les réseaux avec sink mobile de grande taille, mais pas pour un nombre limité de nœud.

De manière générale, il y a deux raisons qui expliquent ce rapport. Premièrement, le nombre d'opération pour calculer le rang dans fonction objective et deuxièmement, il se peut que la topologie utilisée (grille ordonnée) puisse avoir un impact sur l'énergie consommée par le réseau.

### 3.7.7 Simulation 6

Comparaison entre le PDR d'un réseau avec un sink fixe (voir figure 3.9) et mobile (voir figure 3.10) en utilisant l'énergie, l'ETX et l'énergie+ETX comme métriques de routage.

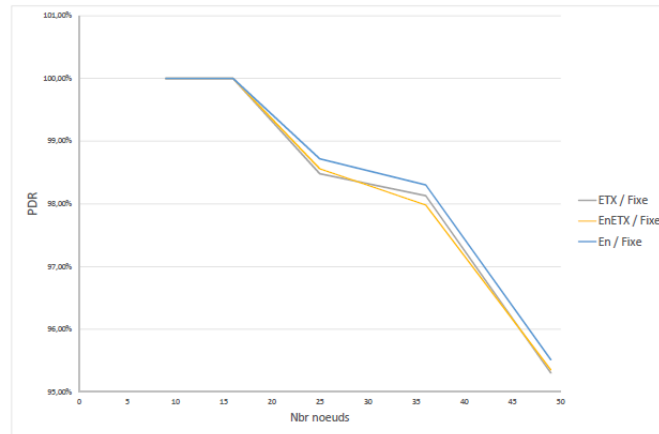


FIGURE 3.12 – Les PDR des réseaux avec sinks fixes (MRHOF-Energie-ETX-Energie+ETX)

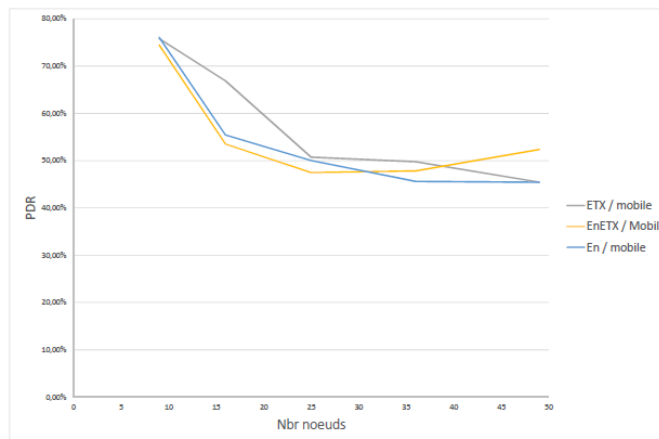


FIGURE 3.13 – Les PDR des réseaux avec sinks mobiles (MRHOF-Energie-ETX-Energie+ETX)

Afin de varier les métriques de performance, nous avons tracer des courbes en fonction du PDR (nombre des paquets reçus / nombre des paquets reçus + nombre des paquets perdus), les résultats obtenus confirment que l'utilisation de la double métrique proposé dans la MRHOF rend le réseau plus fiable dans la transmission des paquets surtout pour les réseaux avec un nombre important de noeuds.

### 3.8 Simulation d'une défaillance partielle de réseau

Afin de mieux tester notre démarche, nous avons réalisé une simulation avec un scénario de défaillance, la mise en œuvre de la simulation passe par quatre phases principales :

- La première phase dans laquelle les nœuds sont déployés aléatoirement dans l'espace opérationnel (voir Figure 3.14)

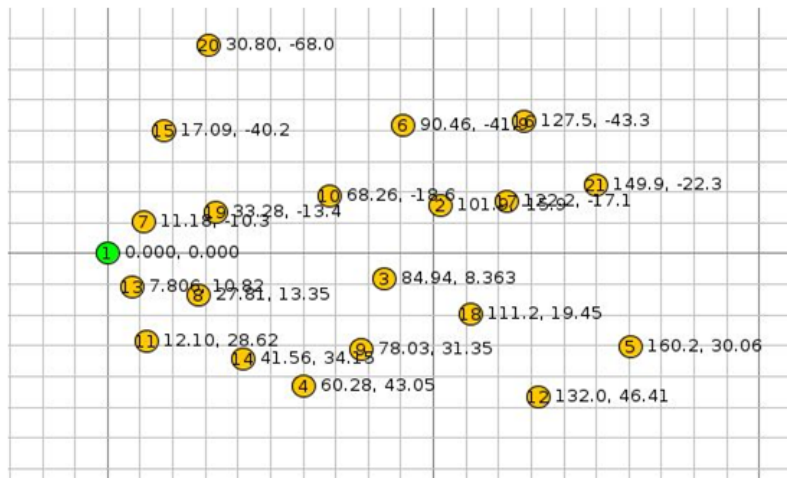


FIGURE 3.14 – Topologie initiale (réseau fonctionnel)

- la phase opérationnelle ou notre réseau fonctionne normalement (construction du DODAG Voir Figure 3.15)

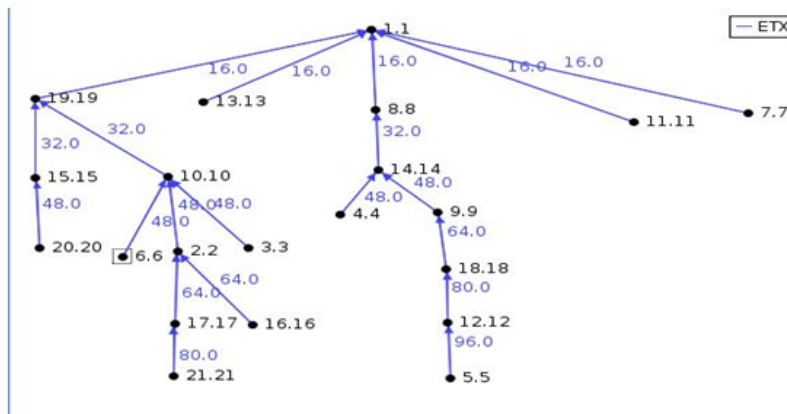


FIGURE 3.15 – DODAG de la topologie initiale

- Phase de défaillance ou un événement indésirable se produit à savoir la perte de trois noeuds critiques 3, 9 et le 10.
- la dernière phase c'est le déplacement du sink mobile vers la zone défaillante (voir la Figure 3.16)



FIGURE 3.16 – Topologie après le déplacement du sink

la Figure 3.17 représente le nouveau DODAG après le déplacement du sink ou on constate un changement considérable par rapport au DODAG initial.

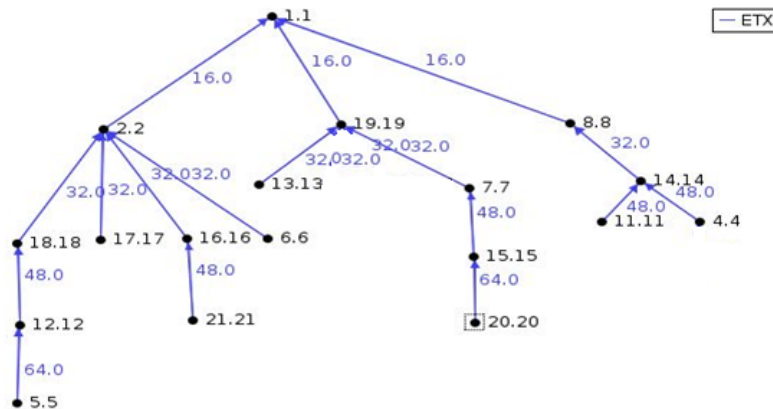


FIGURE 3.17 – DODAG après le déplacement du sink

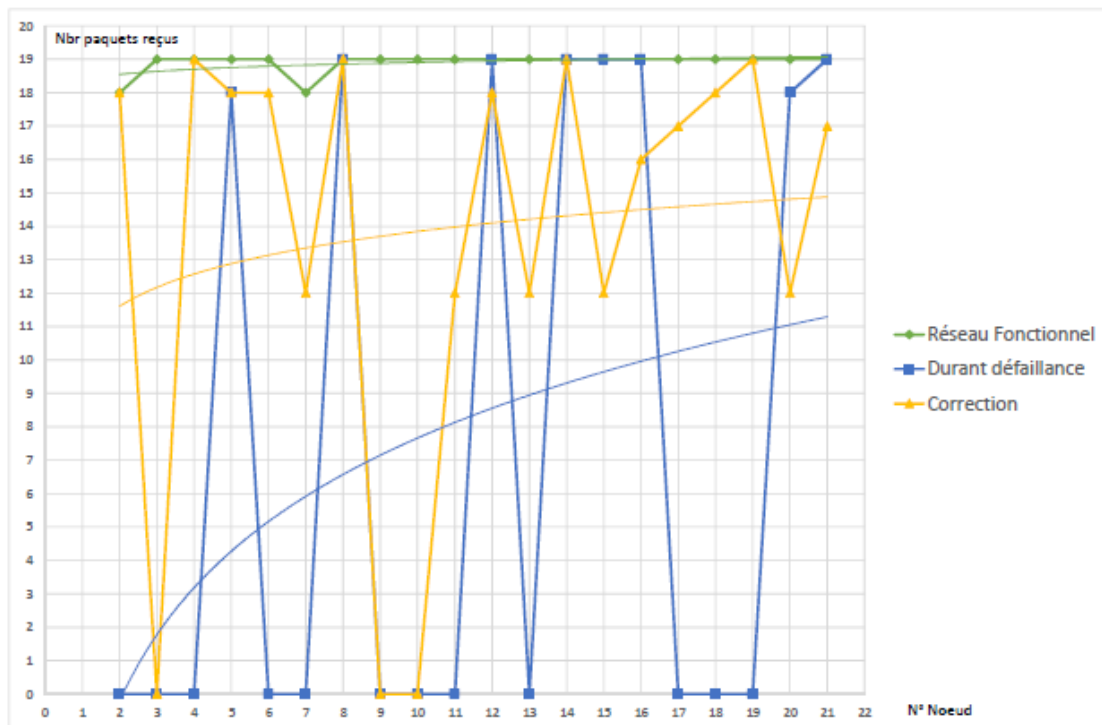


FIGURE 3.18 – Analyse des résultats

D’après la Figure 3.18 on remarque que notre réseau partiellement défaillant a pu récupérer une partie des nœuds non connectés après le déclenchement d’un déplacement du sink vers la zone critique, la courbe de tendance en jaune montre clairement l’amélioration apportée.

Donc on peut considérer que la mobilité à l’événement comme une action corrective pour un réseau LLN défaillant

### 3.9 Conclusion

Dans ce chapitre nous avons réalisé une série de simulations avec intégration d’un sink mobile dans un réseau LLN. L’utilisation le protocole RPL avec variation des métriques de routage nous permet de dire que la combinaison des deux métriques de routage énergie + ETX donne un meilleur résultat en matière de consommation d’énergie et de PDR.



# Conclusion Générale

Dans ce mémoire, nous avons traité la problématique du protocole de routage RPL (IPv6 Routing Protocol for Low Power and Lossy Networks) qui est conçu pour prendre en charge les exigences spécifiques des réseaux LLN's à savoir peu d'autonomie énergétique, peu de mémoire de stockage et peu de puissance de calcul. Ce qui rend le protocole RPL peu adapté à la mobilité des nœuds et le changement de la topologie du réseau.

Notre objectif a été de proposer une amélioration du protocole RPL pour une meilleure prise en charge de la mobilité afin d'améliorer les performances des LLN's, notamment la durée de vie du réseau et le nombre de paquets reçus. Pour cela nous avons proposé une combinaison de deux métriques de routage à savoir l'énergie consommée et la qualité du lien et les intégrées dans la fonction objectif MRHOF toute en ajoutant un sink mobile.

Nous avons validé notre démarche à travers une série de simulations en variant à chaque fois plusieurs paramètres. Les résultats peuvent être récapitulés comme suit :

- Dans un réseau LLN avec sink mobile, la combinaison de deux métriques de routage (consommée et la qualité du lien) donne un meilleur résultat en matière de consommation d'énergie et de transmission des paquets surtout pour les réseaux de grande taille.

- La topologie de réseau, le type de la mobilité et la vitesse de déplacement des nœuds mobiles ont un impact considérable sur les performances des réseaux LLN's.

- La mobilité à l'événement (dans notre cas défaillance partielle de réseau), peut être considéré comme une action corrective pour un système défaillant d'une part, et comme un meilleur type de mobilité d'autre part.

Pour terminer, nous avons identifié deux axes de recherche comme perspectives à notre travail. Premièrement, c'est l'auto-détection des nœuds défaillants au sein d'un réseau qui va nous permettre de prédire les futurs changements de la topologie. Deuxièmement, nos résultats nous laissent croire que l'application du routage multicritères est toute à fait possible en ajoutant par exemple la latence et la vitesse de déplacement.

# Bibliographie

- [1] Yacine CHALLAL, Hatem BETTAHAR, Abdelmadjid BOUABDALLAH, «Les Réseaux de capteurs (WSN : Wireless Sensor Networks) », Rapport interne, Université de Technologie de Compiègne, France, 2008.
- [2] Subha, C. and Malarkkan, S. (2016), “H-CERP : Energy Efficient Scheme for Industrial WSN Applications” Journal of Wireless Personal Communications. Vol. 18 No.4, pp 1-14.
- [3] Sohail, J., Abid, A. , Muhammad, I. , Shehzad, K. and Kashif, S. (2015), “Energy Efficient Strategy for Throughput Improvement in Wireless Sensor Networks”. Journal of Sensors. Vol.15 No.2, pp 2473-2495.
- [4] Lucien Loiseau, Thèse de Doctorat De l’exploitation des réceptions opportunistes dans les mécanismes de relayage pour les réseaux sansfil, Université de Rennes, 1 Décembre 2013.
- [5] Jean Robert HOUNTOMEY, Le Routage Statique AFNOG-NAIROBI-KENYA, 2006.
- [6] Salleddine Kabou, état de l’art sur les réseaux de capteurs sans fil, Université de Bechar Algérie, 2012.
- [7] RFC 6550, T. Winter et al, IPv6 Routing Protocol for Low-Power and Lossy Networks, March 2012.
- [8] Pat Kinney, Phil Jamieson, [ieee802.org, http ://www.ieee802.org/15/pub/TG4.html](http://www.ieee802.org/15/pub/TG4.html), 11 May 2017(Article).
- [9] Jonas Olsson, LoWPAN demystified, Texas Instruments 2014(Article).
- [10] Stéphane Lohier, Internet des objets, [Lohier@univ-mlv.fr](mailto:Lohier@univ-mlv.fr) (Article).
- [11] Pierre-Jean Benghozi, Françoise Massit-Folléa, Sylvain Bureau, L’internet des objets Quels enjeux pour les Européens, Ecole polytechnique et TELECOM Paris Tech. 2008

- [12] Olfa Gaddour, Anis Koubâa, RPL in a nutshell, Polytechnic Institute of Porto, 02.14.2013 Mémoire de Stage de fin d'études Master Informatique, institut de la Francophonie Lyon 1
- [13] Dr A.Benzerbadj cours IOT centre université Belhadj bouchaib Ain Temouchent.
- [14] JP Vasseur, Navneet Agarwal, Jonathan Hui, Zach Shelby, Paul Bertrand, Cedric Chauvenet, "RPL The IP routing protocol designed for low power and lossy networks", (IPSO) alliance, april 2011.
- [15] P Levis, T.Clausen, J.Hui, O.Gnawali,J.Ko, "The Trickle Algorithm", IETF Request for Comments 6206, March 2011.
- [16] Aishwarya Parasuram, David Culler Ed, Randy Katz Ed, An Analysis of the RPL Routing Standard for Low Power and Lossy Networks University of California at Berkeley, May 14, 2016
- [17] JP. Vasseur, M. Kim, K. Pister, N. Dejean, D. Barthel, Routing Metrics Used for Path Calculation in Low-Power and Lossy Networks, IETF Request for Comments 6551, March 2012.
- [18] [www.contiki-os.org](http://www.contiki-os.org).
- [19] Généralité sur les capteurs (source.<https://www.academia.edu>) (visité le 14/05/2019).
- [20] Gartner <http://www.gartner.com/newsroom/id/2636073>, 2013.
- [21] Pat Kinney, Phil Jamieson, [ieee802.org](http://www.ieee802.org), <http://www.ieee802.org/15/pub/TG4.html>, 11 May 2017(Article).
- [22] [datatracker.ietf.org](https://datatracker.ietf.org), <https://datatracker.ietf.org/wg/roll/charter/>, 2017(Article).
- [23] Sur [www.techniques-ingenieur.fr](http://www.techniques-ingenieur.fr) visité le 14/03/2019.

