

Faculté des Sciences Exactes et d'Informatique
Département de Mathématiques et informatique
Filière : Informatique

RAPPORT DE FIN D'ETUDES

Pour l'Obtention du Diplôme de Master en Informatique

Option : Réseaux et Systèmes

Présenté par :

BENIDRIS Rachida Chahrazed

TOUMI Dounia Ikram

THEME :

Support de la mobilité dans le protocole RPL

Soutenu le : 14/06/2023

Devant le jury composé de :

HABIB ZAHMANI Mohamed MCA Université de Mostaganem Président

BAHNES Nacera MAA Université de Mostaganem Examineur

ABID Meriem MCB Université de Mostaganem Encadreur

Année Universitaire 2022-2023

Remerciements

Tout d'abord, nous remercions Allah Tout-Puissant de nous avoir donné la force , la santé et la patience de mener à bon terme ce travail.

Nous remercions nos parents pour leur soutien, leurs prières, leurs encouragements et pour leurs incommensurables sacrifices.

Nous adressons nos respectueux remerciements à notre encadrante, Mm ABID Meriem, pour ses compétences scientifiques, sa patience, sa disponibilité et surtout ses judicieux conseils qui nous ont orienté dans la bonne direction.

Nous remercions également les membres du jury pour l'honneur qu'ils nous ont fait en acceptant de juger ce travail.

Enfin, nos sincères remerciements à tous ceux et celles, qui de près ou de loin, ont permis par leurs conseils et leurs compétences la réalisation de ce mémoire.

Merci pour tous vos encouragements !

Résumé

Au cours des dernières années, l'Internet des objets (IoT) est devenu un paradigme intéressant qui vise à unifier tout dans le monde sous une infrastructure commune, nous donnant non seulement le contrôle des choses qui nous entourent, mais aussi nous tenant informés de l'état des objets. Ces objets peuvent être physiques ou numériques, connectés dans des environnements avec des ressources limitées appelés des réseaux à faible puissance et avec perte (LLN). L'Internet Engineering Task Force (IETF) a développé RPL comme protocole de routage pour les réseaux à faible puissance et avec perte et l'a normalisé dans la RFC 6550 en 2012. Ce protocole est conçu pour les applications IoT stationnaires adaptées à un environnement sans mobilité. Dans ce projet, nous nous sommes intéressées à l'amélioration du protocole RPL afin qu'il soit plus adapté à la mobilité des nœuds. La solution que nous avons proposé a été évalué à l'aide du simulateur Cooja. Les résultats obtenus ont montré une amélioration de notre solution par rapport au protocole original.

Mots clés : IoT, IETF, LLN, RPL, routage, Cooja.

Abstract

In the last few years, the Internet of Things (IoT) has become an interesting paradigm that unify everything in the world under a common infrastructure, not only giving us control of the things around us, but also keeping us informed about the state of objects. These objects can be physical or digital, connected in environments with limited resources called Low-Power and Lossy Networks (LLN). The Internet Engineering Task Force (IETF) has developed RPL as a routing protocol for LLNs and standardized it in RFC 6550 in 2012. This protocol is designed for stationary IoT applications adapted to an environment without mobility. In this project, we focused on improving the RPL protocol to make it more adapted for mobility of nodes. The solution we proposed was evaluated using the Cooja simulator. The obtained results demonstrated an improvement of our solution compared to the original protocol.

Keywords: IoT, IETF, LLN, RPL, routing, Cooja.

Table des matières

Introduction générale	1
1 Le protocole de routage RPL	3
1.1 Introduction	3
1.2 L'internet des objets	3
1.3 Les réseaux de capteurs sans fil	4
1.4 La norme IEEE 802.15.4	4
1.5 Low-Power and Lossy Networks	5
1.6 6LoWPAN	5
1.7 Fonctionnement de 6LoWPAN	6
1.7.1 Compression d'en-tête	6
1.7.2 Fragmentation et reconstitution des trames	7
1.8 Le protocole de routage RPL	7
1.9 Fonctionnement du protocole RPL	9
1.9.1 Messages de contrôle dans RPL	9
1.9.2 Métriques du protocole RPL	9
1.9.3 Construction de la topologie	9
1.9.4 Mécanisme de sélection de chemin par fonction objectif	10
1.9.5 Flux de trafic pris en charge par RPL	12
1.9.6 Maintenance de la topologie	12
1.9.7 L'algorithme Trickle Timer :	13
1.9.8 Déroulement de l'algorithme :	13
1.9.9 La sécurité dans le protocole RPL	14
1.10 Conclusion	15

2	La mobilité dans le protocole de routage RPL	16
2.1	Introduction	16
2.2	Les scénarios de la mobilité	16
2.3	RPL et la mobilité	17
2.4	Les racines DODAG statiques et mobiles dans RPL	18
2.5	État de l’art	18
2.6	Les modèles de mobilité	20
2.7	Conclusion	21
3	Solution proposée	22
3.1	Introduction	22
3.2	RSSI	22
3.3	La solution proposée	23
3.3.1	Le modèle de mobilité choisi	25
3.4	Conclusion	26
4	Implémentation de la solution	27
4.1	Introduction	27
4.2	Environnement de travail	27
4.2.1	Système d’exploitation d’IoT Contiki	27
4.2.2	Le simulateur COOJA	28
4.3	Évaluation des performances de notre solution	29
4.3.1	Collect View	29
4.3.2	Configuration de la simulation	29
4.3.3	Implémentation du modèle de mobilité	30
4.4	Résultats de la simulation	31
4.4.1	Taux de perte de paquets	31
4.4.2	Consommation d’énergie	32
4.5	Conclusion	33
	Conclusion générale	34

Table des figures

1.1	Réseau de capteurs sans fil.	4
1.2	L'architecture 6LoWPAN.	6
1.3	La différence entre Dag et Dodag.	8
1.4	Les instances RPL.	8
1.5	L'algorithme trickle timer.	14
3.1	Algorithme de la solution proposée.	23
3.2	Organigramme de la fonction calculate_path_metric.	24
3.3	Organigramme de la fonction best_parent.	25
4.1	Le simulateur Cooja.	29
4.2	Taux de perte de paquets.	32
4.3	Consommation d'énergie.	33

Liste des tableaux

4.1 Paramètres de la simulation.	30
--	----

Liste des Abréviations

DAG Directed Acyclic Graph.

DAO Destination Advertisement Object.

DAO-ACK Destination Advertisement Object ACKnowledgement.

DIO DODAG Information Object.

DIS DODAG Information Solicitation.

DODAG Destination Oriented DAG.

ETX Expected Transmission Count.

FANET Flying Ad-Hoc Networks.

GPS Global Policy and Strategy.

IEEE Institute of Electrical and Electronics Engineers.

IETF Internet Engineering Task Force.

IOT Internet Of Things.

LLN Low power and Lossy Networks.

MMM Manhattan Mobility Model.

MRHOF Minimum Rank with Hysteresis Objective Function.

OF0 Objective Function Zero.

RDM Random Direction Model.

ROLL Routing Over Low power and Lossy networks.

RPL Routing Protocol for Low power and lossy networks.

RSSI Received Signal Strength Indicator.

RWP Random Waypoint Model.

STRAW Street Random Waypoint.

UAV Unmanned Aerial Vehicle.

UDP User Datagram Protocol.

6LoWPAN IPv6 Low Power Wireless Personal Area Networks.

Introduction générale

L'informatique et l'Internet sont une nécessité pour la vie actuelle, des objets connectés à l'Internet peuvent communiquer entre eux, ils donnent la possibilité d'intégrer le monde physique dans les systèmes informatiques.

L'Internet des objets ou Internet of Things (IoT) marque le début d'une nouvelle ère en termes de mobilité et de connectivité, grâce à elle, les objets courants deviennent intelligents, ces objets sont généralement limités en termes de puissance et de batterie à travers des réseaux LLN (Low power and Lossy Networks), les réseaux formés par ces objets sont caractérisés par des liaisons instables avec un taux de perte élevé et des débits faibles [23]. Pour répondre à ces contraintes l'IETF (Internet Engineering Task Force) a créé des groupes de travail afin de proposer des nouveaux protocoles. Le premier groupe 6LoWPAN qui a standardisé une couche adaptative nommée 6LoWPAN qui permet à un paquet IPv6 d'être pris en charge par des trames de petites tailles et le deuxième groupe est le groupe de travail ROLL qui a défini le protocole de routage RPL (Routing Protocol for Low Power and Lossy Networks). Ce protocole est conçu pour des nœuds statiques, mais certains scénarios impliquent la mobilité des nœuds ou il fallait bien s'adapter la difficulté de ce cas et proposer des solutions qui facilitent le traitement.

Dans le premier chapitre intitulé « Le protocole de routage RPL », nous allons définir brièvement l'Internet des objets, les réseaux LLN, la norme IEEE 802.15.4, le standard 6LoWPAN avec son fonctionnement et finalement nous allons parler du RPL, ses messages échangés, son fonctionnement, sa construction et maintenance de la topologie DODAG, ce qui va ouvrir la porte à la mobilité des nœuds RPL.

Le chapitre 2, intitulé « La mobilité dans le protocole de routage RPL », définit les cas de l'IoT où les nœuds sont mobiles, les scénarios de la mobilité dans l'IoT et l'état de l'art.

Le chapitre trois aborde notre solution visant à améliorer les performances du protocole RPL dans des scénarios où les nœuds sont mobiles, Enfin, le chapitre quatre est consacré à l'implémentation de notre solution ainsi qu'à la présentation des résultats obtenus grâce à

l'utilisation du simulateur Cooja sur Contiki 3.0.

Chapitre 1

Le protocole de routage RPL

1.1 Introduction

L'Internet des objets est une technologie qui permet de connecter n'importe quel ensemble d'objets entre eux et qui hérite aux attributs des réseaux LLN. Ces attributs amènent un problème majeur pour l'adaptation de protocole de routage. Pour cela, le groupe de travail ROLL a déterminé le protocole de routage RPL (Routing protocol for Low Power and Lossy Networks) principalement formé pour répondre aux besoins des réseaux à faible consommation d'énergie. Dans ce chapitre, nous allons introduire ce travail à savoir l'internet des objets et surtout les réseaux à faible puissance, le protocole RPL (en se focalisant sur son mode de fonctionnement), et à définir le protocole 6LoWPAN.

1.2 L'internet des objets

L'Internet des objets est un paradigme qui permet de dérouler une connexion Internet ou un réseau local comme les capteurs sans fil pour interconnecter différents types d'objets comme les ordinateurs, les téléphones mobiles, pour avoir un mode de vie beaucoup plus simple et une bonne qualité de service dans différents domaines d'application. Le CERP-IdO « Cluster des projets européens de recherche sur l'Internet des objets » a défini l'Internet des objets comme une infrastructure dynamique d'un réseau global où il y a des protocoles de communication interopérables, les objets connectés dans cette infrastructure ont des identités, attributs physiques et virtuelles et ils sont intégrés au réseau d'une façon transparente [21].

1.3 Les réseaux de capteurs sans fil

Les réseaux de capteurs sans fil sont une technologie très importante dans l'Internet des objets, Ils présentent l'état d'objets dans lesquels ils sont implantés, il est composé d'un ensemble des nœuds (capteurs) communicants dans une zone de captage, déployés afin de surveiller et transmettre les informations à la station de base comme nous le montre la figure 1.1, son objectif est de capter et expédier des données du monde physique vers un ou plusieurs points [4].

Un capteur IoT mesure les données et détecte les changements dans l'environnement, c'est une véritable technologie de connectivité et d'intelligence, il est donc l'un des trois éléments principaux d'un écosystème IoT.

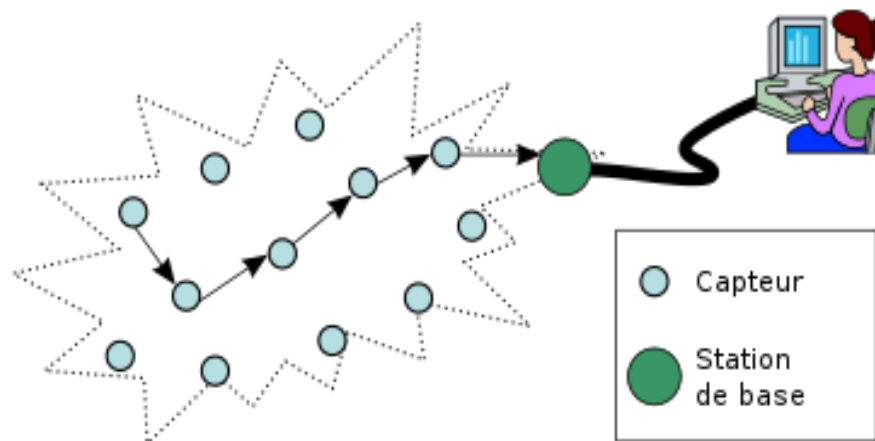


FIGURE 1.1 – Réseau de capteurs sans fil.

1.4 La norme IEEE 802.15.4

Il existe plusieurs normes connues comme 802.11 pour les réseaux LAN sans fil (WLAN) et 802.3 pour les réseaux Ethernet, IEEE a normalisé aussi la norme 802.15.4. La norme IEEE 802.15.4 est une technologie d'accès sans fil à faible coût, à faible débit (250 kb/s, 40 kb/s et 20 kb/s), et à faible portée (100 m), très utilisée dans les protocoles de communication des réseaux de capteurs sans fil. A cause de ces caractéristiques, ils l'ont choisi comme un des armatures de communication utilisés pour les réseaux Low-Power and Lossy Networks (LLN), cette technologie peut supporter un nombre très important de nœuds (jusqu'à 65000) [13].

1.5 Low-Power and Lossy Networks

Les réseaux à faible puissance et avec pertes (LLN) sont des nœuds interconnectés tel que sa batterie, mémoire et puissance de traitement sont limités. Il existe un large éventail de domaines d'application pour les LLN comme la surveillance industrielle, l'automatisation des bâtiments (chauffage, climatisation et ventilation), domaine de la santé, la surveillance environnementale, gestion de l'énergie et les maisons connectées [15]. Les routeurs LLN utilisent dans son fonctionnement des contraintes sur la puissance de traitement, la mémoire et l'énergie. Leurs interconnexions sont généralement caractérisées par des pertes élevées en termes de débit et d'instabilité.

Les réseaux LLN sont composés de quelques dizaines jusqu'à des milliers de routeurs LLN qui prennent en charge le trafic point à point (P2P) entre les appareils à l'intérieur de ces réseaux, le trafic point à multipoint (P2MP) d'un point de contrôle centralisé à des sous-ensemble d'appareils et finalement le trafic multipoint à point (MP2P) qui sont des appareils situés à l'intérieur du réseau vers un point de contrôle centralisé.

1.6 6LoWPAN

6LoWPAN (IPv6 Low power Wireless Personal Area Networks) est une norme créée par l'IETF, C'est une définition d'un protocole qui permet de transporter des paquets IPv6 sur des réseaux sans fil à faible puissance, en particulier IEEE 802.15.4. En Ipv6, la taille minimale des paquets est 1280 octets, et dans la norme IEEE 802.15.4, la taille des trames est uniquement 127 octets. Donc, il était nécessaire de trouver une solution pour cette incompatibilité. 6LoWPAN sert à créer une couche d'adaptation permettant aux paquets IPv6 d'être pris en charge par des trames de petite taille comme celles dans le standard IEEE 802.15.4.

La conception finale sert à créer un ensemble d'en-têtes qui permettent le codage efficace des grandes adresses/en-têtes IPv6 dans un en-tête compressé plus petit, tout en permettant l'utilisation de divers réseaux maillés en prenant charge la fragmentation et le ré-assemblage si nécessaire.

Dans un réseau 6LoWPAN, il existe 3 familles d'architectures :

- Réseau 6LoWPAN simple : Un seul routeur de bordure appelé Edge Router qui permet l'interconnexion avec un autre réseau IP.
- Réseau 6LoWPAN étendu : Plusieurs routeurs de bordures reliés.

- Réseau ad hoc 6LoWPAN : Un réseau non connecté.

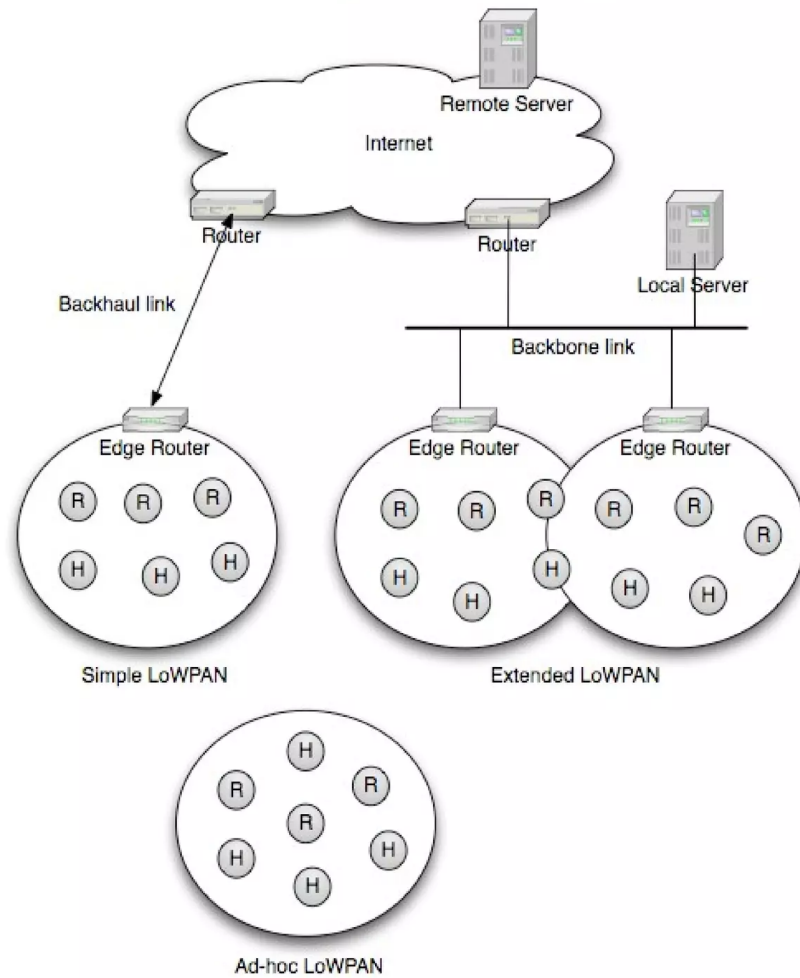


FIGURE 1.2 – L'architecture 6LoWPAN.

1.7 Fonctionnement de 6LoWPAN

6LoWPAN se base principalement sur deux mécanismes :

1.7.1 Compression d'en-tête

Pour les réseaux alimentés par des batteries, les paquets d'en-tête volumineux sont considérés comme un gaspillage d'énergie. L'IPv6 avec des 40 octets d'en-tête est l'un des pires candidats possibles. Sans compression, les données d'un simple paquet UDP ne dépassent pas 53 octets. Avec la compression, les en-têtes 6LoWPAN et UDP ne requièrent que 6 octets.

On distingue plusieurs cas de communication possibles, par exemple dans la communication de deux périphériques internes dans le même réseau, on peut compresser l'en-tête IPv6 à seulement 2 octets, dans la communication externe, l'en-tête IPv6 est compressée à 12 octets [14].

1.7.2 Fragmentation et reconstitution des trames

Sur des réseaux IEEE 802.15.4 où la taille des trames est limitée à 127 Octets, le surcoût du niveau MAC peut ne laisser que 81 Octets pour les niveaux IP et supérieurs. Le standard IPv6 requiert normalement le support d'une taille de paquet minimale de 1280 Octets, donc après la réduction de la longueur des paquets IPv6, la couche 6LoWPAN fragmente les paquets venus de la couche réseau et les envoie sous forme des petites trames à l'équipement distant, ce dernier fait le réassemblage et chaque fragment sera précédé d'un en-tête de fragmentation.

1.8 Le protocole de routage RPL

RPL (Routing Protocol for Low Power and Lossy Networks) est un protocole de routage pour les réseaux à faible puissance et avec perte, il a été développé par le groupe de travail IETF ROLL dans la RFC 6550 pour répondre aux besoins des réseaux à faible consommation d'énergie.

Le protocole RPL est un protocole proactif à vecteur de distance basé sur IPv6, les équipements RPL sont interconnectés selon une topologie nommée DODAG (Destination Oriented Directed Acyclic Graphs), DODAG est basé sur DAG (Directed Acyclic Graph) ou il n'y a pas de boucles, ce qui implique l'impossibilité de passer par le même nœud deux fois. De plus, le DAG est un graphe orienté qui a un ou plusieurs nœuds racines, DODAG est un DAG avec un seul nœud racine comme nous montre la figure 1.3.

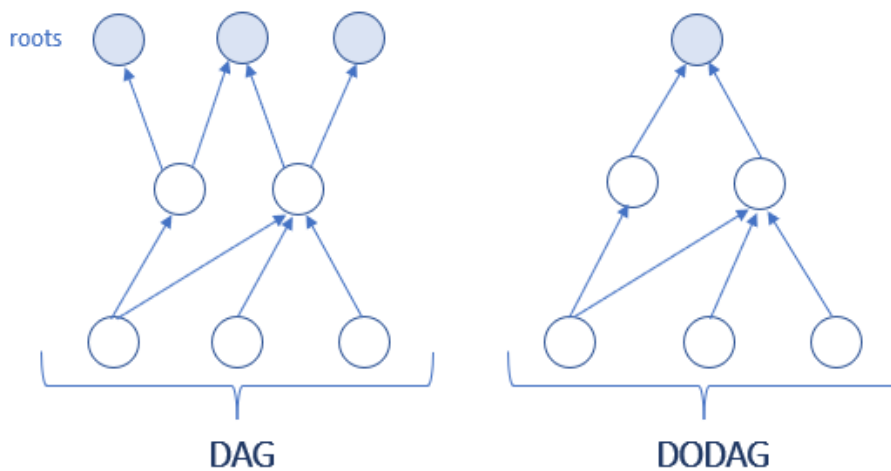


FIGURE 1.3 – La différence entre Dag et Dodag.

Une instance RPL est composée d'un ou plusieurs DODAG, un nœud RPL peut joindre plusieurs instances en même temps, mais il ne peut joindre qu'un seul DODAG par instance (comme les nœuds 13 et 17 dans la figure 1.4) [12].

RPL prend en charge trois types de trafic, trafic multipoint à point (MP2P) des feuilles vers la racine via des voies ascendantes, trafic point à multipoint (P2MP) de la racine à la sortie via des voies descendantes et finalement le trafic point à point (P2P) via des voies ascendantes et descendantes.

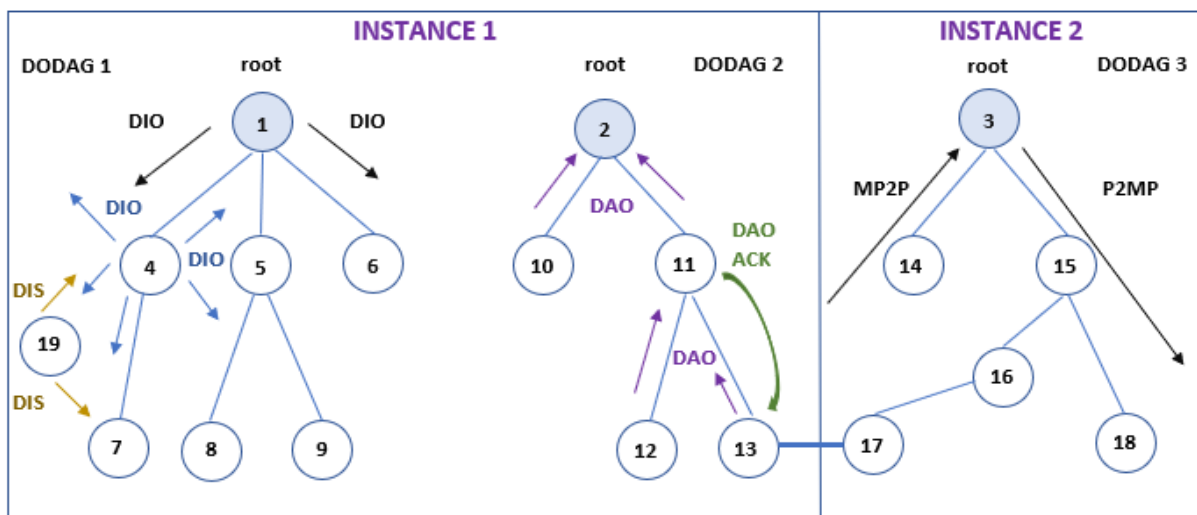


FIGURE 1.4 – Les instances RPL.

1.9 Fonctionnement du protocole RPL

1.9.1 Messages de contrôle dans RPL

Pour gérer DODAG, RPL définit des nouveaux messages de contrôle ICMPv6 :

- DIO (DODAG Information Object) : Ces messages sont composés de 5 champs d'informations, parmi ces informations il y a les identifiants de l'instance RPL et du DODAG, les temporisateurs d'entretien et le rang ce qui permet à un nœud de découvrir une instance RPL, calculer son rang, sélectionner un ensemble de parents et maintenir le DODAG.
- DIS (DODAG Information Solicitation) : Utilisés pour demander des informations de routage afin de rejoindre le DODAG, ces messages sont envoyés par des nœuds déconnectés du DODAG pour recevoir des messages DIO contiennent des informations requises par des nœuds RPL.
- DAO (Destination Advertisement Object) : Utilisés pour construire des routes descendantes dans un DODAG, ces messages sont unicast (monodiffusion) de l'enfant vers le parent.
- DAO-ACK (Destination Advertisement Object ACKnowledgement) : Est envoyé en unicast par un destinataire DAO pour acquitter un message DAO envoyé.

1.9.2 Métriques du protocole RPL

La métrique d'un protocole est la valeur d'une route en comparaison avec d'autres routes. Dans RPL, la métrique peut être classée en deux catégories :

- Métriques de nœuds : En prenant compte les caractéristiques d'un nœud, par exemple le taux d'occupation de CPU, l'énergie résiduelle (restante), le type d'alimentation et le nombre de sauts qui est le nombre de nœuds traversé pour atteindre la racine.
- Métriques de liens : En prenant compte les propriétés de liens, par exemple la fiabilité (qualité du lien) qui est comprise entre 1 et 9, la latence du lien qui est le temps nécessaire à un paquet pour qu'il arrive à la destination, le débit du lien et le ETX (Expected Transmission Count) qui est le nombre de transmissions attendues [16].

1.9.3 Construction de la topologie

Le rang (rank) d'un nœud est sa position individuelle par rapport aux autres nœuds par rapport à la racine DODAG principale, sa valeur augmente vers le bas et diminue vers le haut.

La façon de calculer le rang dépend de la fonction objectif (OF). Pour construire un graphe DODAG, la racine diffuse un message DIO qui est émis périodiquement selon un temporisateur réglé, ce message contient cinq champs principaux correspondant à l'ID d'instance RPL, l'ID DODAG, le numéro de version, le rang et le champ Mode d'opération (MOP). Quand un nœud reçoit ce message il ajoute le nœud expéditeur à sa liste de parents avec sa valeur de rang en tenant compte de la fonction objectif référencée dans le message DIO et il transmet à ses voisins le message DIO mis à jour.

A l'aide de la liste des parents, un nœud sélectionne son parent préféré pour l'envoi des données aux DODAG racine, cette opération permet aux nœuds d'avoir une route par défaut composée de tous les parents préférés et ascendante vers la racine DODAG. Par contre, les routes descendantes sont construites par plusieurs messages DAO selon un mode de fonctionnement qui est bien spécifié par la racine dans les messages DIO. On distingue donc deux modes de fonctionnement :

- Storing mode : Le nœud enfant envoie un message en unicast au parent qui l'enregistre dans sa liste des parents, ce dernier agrège les routes reçues d'autres messages DAO et envoie les informations à son parent par un message DAO [12].
- Non-Storing mode : dans ce mode, les nœuds intermédiaires ne stockent pas les informations de routage, donc les messages DAO sont envoyés en unicast à la racine DODAG avec la présence d'adresses des nœuds intermédiaires pour compléter le chemin inverse.

Finalement, quand un nouveau nœud veut joindre une instance RPL, il diffuse un message DIS afin de recevoir des messages DIO de ses voisins.

1.9.4 Mécanisme de sélection de chemin par fonction objectif

La fonction objective sert à sélectionner des itinéraires dans un DODAG et permet aux nœuds de traduire des métriques en rangs. Le calcul du rang est effectué à l'aide de l'OF en fonction des métriques de routage.

Chaque instance DODAG dans un DAG RPL est associée à un OF particulier, RPL définit deux fonctions objectives : Objective Function Zero (OF0) et Minimum Rank Hysteresis Objective Fonction (MRHOF).

- OF0 est une fonction objective, elle utilise le nombre minimal de sauts pour atteindre la racine. Chaque nœud calcule le rang en prenant compte la valeur du saut vers la racine, le

moins de sauts obtiendra un lien prioritaire sera sélectionné par OF0. Le rang d'un nœud enfant est toujours supérieure au rang du nœud parent.

Un nœud peut calculer son rang à l'aide de l'équation suivante :

$$\mathbf{R(N) = R(P) + Incrémentation}, \text{ tels que :}$$

$$\mathbf{Incrémentation = (Rf * Sp + Sr) * MinHopRankIncrease.}$$

- R(N) : le rang d'un nœud.
- R(P) : le rang du parent.
- Sp : step of rank, une constante qui représente la qualité du lien radio utilisée dans le réseau. Sa valeur est entre 1 et 9.
- Rf : permet d'augmenter la valeur de Sp.
- Sr : stretch of rank, une constante qui permet d'augmenter la valeur de Sp pour la sélection d'un potentiel parent de secours.
- MinHopRankIncrease : La valeur minimale pour incrémenter un rang. Elle est égale à 256 [22].

- MRHOF est une fonction objective, elle utilise la valeur minimale du nombre de transmissions attendu (ETX) sur l'opération de sélection du nœud parent. ETX représente le nombre attendu de transmissions nécessaires pour qu'il soit reçu sans erreur à sa destination.

Chaque nœud calcule l'ETX qu'il a sur le lien radio qu'il partage avec son voisin, comme suit :

$$\mathbf{ETX}_L = \frac{1}{d_f * d_r}, \text{ tels que :}$$

- ETX_L : le nombre moyen de transmissions pour la réception d'un paquet sur un lien L.
- d_f : la probabilité qu'un paquet soit reçu par la destination.
- d_r : la probabilité qu'un acquittement soit envoyé de la destination.

Le calcul du rang pour MRHOF est présenté comme suit :

$$\mathbf{Rang = fonction(pathcost)}, \text{ tels que :}$$

$$\mathbf{Pathcost = parentpath-cost + link-cost.}$$

- link-cost : le coût associé au lien du parent, il dépend à la métrique sélectionnée.
- parentpath-cost : le path-cost du parent.
- La manière de transformer path-cost en rang dépend de la métrique choisie [6].

1.9.5 Flux de trafic pris en charge par RPL

RPL prend en charge trois flux de trafic de base :

Le multipoint à point (MP2P) : C'est la construction des routes ascendantes du nœud vers la racine DODAG, les destinations des flux MP2P sont des nœuds qui ont une signification applicative, par exemple garantir une large connectivité du réseau internet ou du réseau de base IP privé.

Point à multipoint (P2MP) : Ce modèle de trafic est requis par plusieurs applications LLN ([RFC5867], [RFC5826], [RFC5673] et [RFC5548]), il s'agit des routes descendantes de la racine vers les nœuds.

Point à point (P2P) : Pour qu'un réseau RPL prend en charge le trafic P2P, la racine DODAG doit être capable d'acheminer des paquets vers une destination. Dans le Non-Storing mode le paquet sera dirigé vers la racine, et cette dernière prend en charge son routage vers la destination, dans le cas de Storing mode le paquet s'écoule vers la racine jus qu'il trouve un nœud qui connaît une route vers la destination, ce nœud peut être n'importe quel nœud proche de la source ou de la destination et il peut être aussi le nœud racine [23].

1.9.6 Maintenance de la topologie

En cas d'incohérence ou panne dans une instance RPL, deux principaux mécanismes de maintenance peuvent être utilisés :

La réparation locale : Consiste à rechercher un autre chemin alternatif pour acheminer les paquets lorsque la communication avec le parent préféré est rompue. Dans le cas où un nœud n'a pas d'autres parents, il peut envoyer ces paquets à son frère. Le DODAG ne sera plus optimal, seule la réparation globale permettra une nouvelle optimisation.

La réparation globale : Dans le cas où les réparations locales ne suffisent pas, une réparation globale est recommandée. La racine initie une reconstitution optimale du DODAG avec une incrémentation de numéro de version du DODAG pour différencier le DODAG ancien et nouveau, ce numéro de version garantit que les informations circulant sur le réseau sont à jour et lorsque les nœuds reçoivent des DIO, ils n'acceptent que les DIO dont le numéro de version est supérieur ou égal au nombre actuel [12].

1.9.7 L'algorithme Trickle Timer :

RPL utilise Trickle Timer afin de diminuer la surcharge des messages de contrôle dans le cas où la topologie est stable, les nœuds ne transmettent une mise à jour seulement lorsqu'ils détectent une incohérence dans le réseau, donc dans le cas de la réception des messages DIO de mise à jour qui n'indiquent aucun changement de la topologie (cohérence), le compteur de redondance sera incrémenté pour permettre aux nœuds d'augmenter la durée d'attente, si le nombre de mises à jours dépasse le nombre de redondances, les nœuds ne transmettent rien et la durée d'attente sera doublée, et si les nœuds entendent une incohérence dans la topologie, le compteur sera réinitialisé et les messages DIO seront envoyés pour propager les mises à jour sur le réseau.

Pour cela, Trickle Timer utilise 3 paramètres :

- I_{max} : la valeur de la taille maximale de l'intervalle, c'est le nombre de doublement de la taille minimale de l'intervalle.
- I_{min} : la valeur de la taille minimale de l'intervalle, un protocole peut définir cette valeur en millisecondes.
- Constante de redondance (k) : entier naturel supérieur à 0, il sert à supprimer les transmissions DIO si le compteur de redondance dépasse sa valeur [10].

1.9.8 Déroulement de l'algorithme :

L'algorithme Trickle Timer se déroule comme suit :

- Etape 1 : Trickle timer donne une valeur à I dans l'intervalle $[I_{min}, I_{max}]$.
- Etape 2 : L'algorithme met le compteur c à 0, et donne une valeur aléatoire dans l'intervalle $[I/2, I]$ à une variable nommée t .
- Etape 3 : Lors de la réception d'un message cohérent, la valeur de compteur c sera incrémentée (+1).
- Etape 4 : A un moment t , si la valeur du compteur c est supérieure ou égale à la constante de redondance k , l'algorithme annule l'envoi des messages DIO, dans le cas contraire, si la valeur de c est inférieure à k , les messages DIO sont envoyés.
- Etape 5 : Quand la taille de l'intervalle I expire, Trickle Timer double cette taille et on revient à l'étape 2.
- Etape 6 : Lors de la détection d'un message incohérent, l'algorithme redéfinit la

valeur de I a Imin et on revient à l'étape 1 [10].

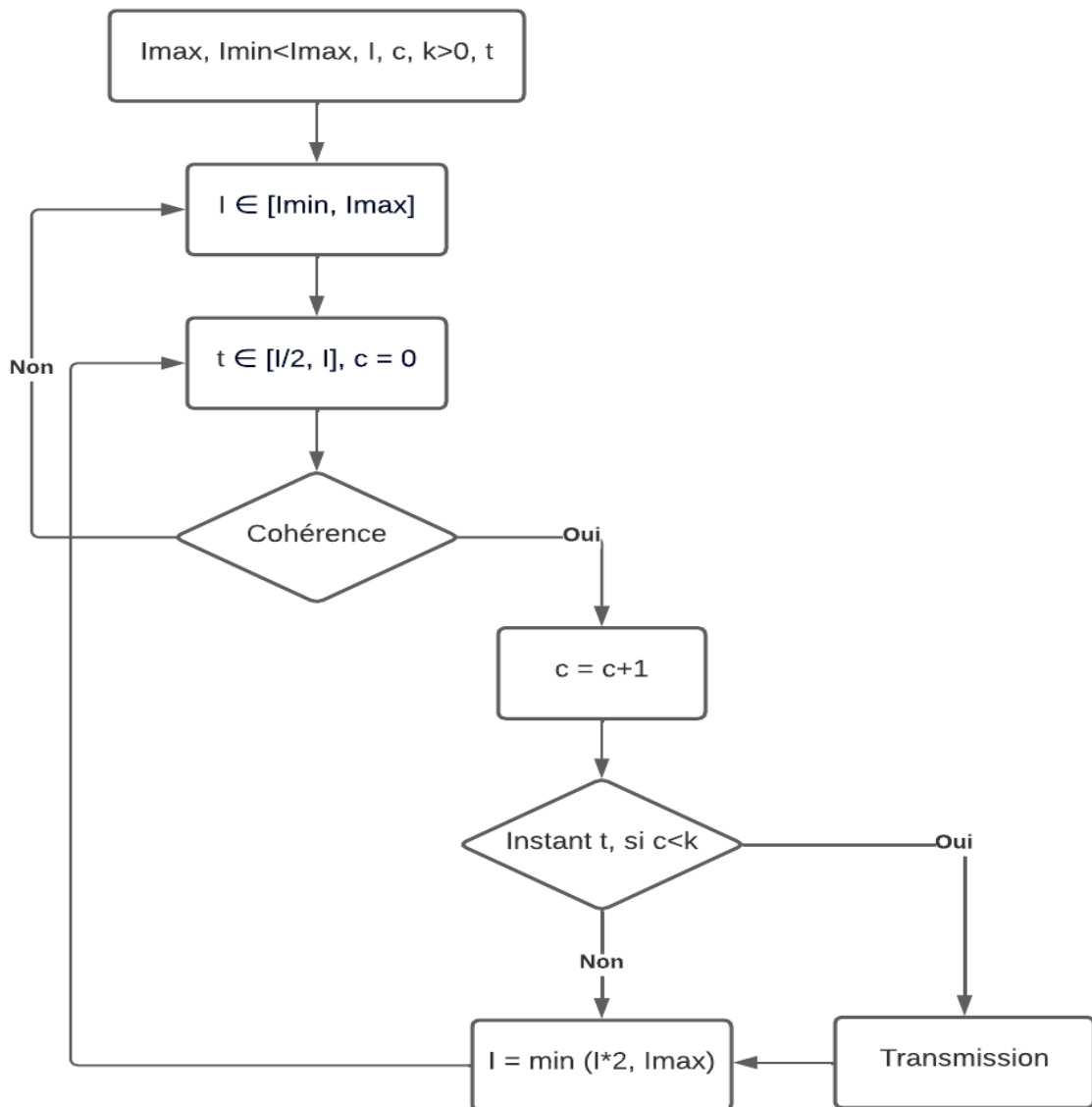


FIGURE 1.5 – L’algorithme trickle timer.

1.9.9 La sécurité dans le protocole RPL

Dans sa version standard, RPL apporte des mesures de sécurité pour assurer une bonne opérations de réseau.

RPL dispose trois modes de sécurité de base :

- Non sécurisé : Les messages de contrôle RPL sont envoyés sans mécanismes de sécurité,

cela n'implique pas que le réseau RPL n'est pas sécurisé, il pourrait utiliser d'autres primitives de sécurité pour garantir la sécurité des applications telles que la sécurité de la couche liaison.

- Préinstallé : Les nœuds qui veulent rejoindre une instance RPL ont des clés préinstallées qui leur permettent de générer des messages sécurisés.
- Authentifié : comme le mode préinstallé, les nœuds ont des clés préinstallées, cette dernière ne peut être utilisée que pour rejoindre une instance RPL en tant que feuille. Pour rejoindre une instance RPL authentifiée en tant que routeur, il faut obtenir une clé d'autorité d'authentification.

1.10 Conclusion

Dans ce chapitre, nous avons défini l'Internet des objets et plus particulièrement les réseaux LLN qui représentent des réseaux à faible puissance et avec perte, ces réseaux utilisent des médiums de communications à faible consommation tels que la norme IEEE 802.15.4. Nous avons défini aussi la norme 6LoWPAN qui permet aux paquets IPv6 d'être pris en charge par des trames de petite taille comme celles dans le standard IEEE 802.15.4. Finalement, nous avons présenté le protocole de routage RPL, ces messages de contrôles qui permettent la construction et le maintien des DODAG, la fonction objectif et les différentes métriques utilisées, l'algorithme trickle timer et son rôle de contrôler l'envoi des messages DIO dans la topologie. Dans le chapitre prochain, nous allons aborder le problème de mobilité dans le protocole RPL et son impact sur le bon fonctionnement du protocole.

Chapitre 2

La mobilité dans le protocole de routage RPL

2.1 Introduction

Aujourd'hui, l'IoT a été largement mis en œuvre dans de nombreuses applications quotidiennes, avec l'utilisation généralisée des applications IoT et le nombre d'utilisateurs connectés aux appareils intelligents, le routage est devenu très difficile. Le protocole de routage IPv6 pour les réseaux à faible puissance et avec perte a une topologie organisée comme un Graphe acyclique dirigé orienté vers la direction (DODAG) conçu pour les applications IoT stationnaires et adapté à un environnement sans mobilité.

Malgré les études sur le réglage du RPL pour les applications IoT mobiles, il y a toujours une forte demande pour plus d'efforts pour trouver une version standard du protocole RPL pour de telles applications. Dans ce chapitre, nous allons définir les domaines d'utilisation des nœuds mobiles, les problèmes de la mobilité des nœuds et les solutions proposées pour la gestion de la mobilité dans le protocole RPL.

2.2 Les scénarios de la mobilité

Les appareils IoT sont exploités sur de nombreux objets mobiles qui ont le potentiel de faciliter la vie quotidienne, transformer la façon dont nous voyageons, transportons des marchandises et utilisons les infrastructures de transport. Les véhicules connectés peuvent être équipés de capteurs, de caméras et de dispositifs de communication. Ces équipements peuvent

envoyer des données en temps réel sur leur position, vitesse, état de fonctionnement, etc.

L'IoT peut également être utilisé pour améliorer la logistique dans les entreprises de transport de marchandises. Les données collectées sur les camions, les conteneurs et les entrepôts peuvent être utilisées pour minimiser les temps d'attente et maximiser l'efficacité de la livraison. Beaucoup d'autres applications IoT mobiles existantes dans le monde réel telles que les vélos électriques connectés, les services de soins de santé en temps réel, par exemple enregistrer les données de santé dans des montres portables et un signal sera envoyé directement à la plate-forme de gestion qui permet de déclarer qu'il y a un changement de l'état de santé. Dans les zones de construction, des robots sont utilisés pour remplacer l'effort humain, des capteurs de surveillance des employés sont possibles pour éviter de s'écarter de la zone de travail à l'aide de la localisation diffusée en temps réel. Dans le domaine militaire, les drones militaires sont bien présents pour se protéger dans une situation de danger ou dans une guerre [18].

La version standard du protocole RPL ne prend pas en charge le GPS et les services basés sur la localisation. Les applications mobiles telles que les Flying Ad-hoc (FANET) et les véhicules aériens sans pilote (UAV) nécessitent des procédures de routage 3D, tandis que RPL est une approche 2D.

2.3 RPL et la mobilité

Le protocole RPL a été développé pour fournir une livraison de données élevée et fiable. Mais éventuellement, il est incapable de gérer la mobilité. Donc il est un peu difficile d'améliorer ce protocole pour prendre en charge la mobilité des nœuds. La topologie du protocole RPL fonctionne avec le principe parent-enfant. Chaque nœud du DODAG a un rang, ce rang est plus élevé lorsque le nœud se situe loin de sa racine. Le nœud avec le rang le plus petit fonctionne comme le nœud parent et celui qui a le rang supérieur devient le nœud enfant. Avec ces principes, RPL détermine la route par défaut des données sur le réseau. Dans la mobilité, lorsque le nœud mobile se déplace, sa connectivité avec ses nœuds parents peut être altérée en raison de l'affaiblissement du signal radio et de l'interférence, donc il doit toujours sélectionner un nouveau parent pour maintenir sa connectivité, ce nouveau parent doit être resélectionné avec des conditions précises (chemin le plus court, fiabilité et faible délai). Pour cela, tous les nœuds nécessitent d'échanger des messages, cette échange pose un problème de surcharge dans le réseau et augmente le délai et la perte de paquets [19]. Donc, la présence de nœuds mobiles

dans une infrastructure IoT sert à avoir des liaisons moins fiables et plus de pertes de paquets dans le réseau, ce qui permet d'avoir des inconvénients tels que l'instabilité de la topologie et la forte consommation d'énergie.

2.4 Les racines DODAG statiques et mobiles dans RPL

Les réseaux des capteurs sans fil et mobiles sont composés d'un grand nombre de capteurs qui ont une alimentation limitée. Les capteurs jouent le rôle de source de nœuds, ces capteurs sont situés à proximité des racines et ils sont plus susceptibles de consommer l'énergie beaucoup plus rapidement que les nœuds distants car ils transportent plus de charge de travail donc ils deviennent des points chauds, ce qui empêche les nœuds éloignés de relayer leurs données vers les racines DODAG et diminuer la durée de vie du réseau. Une catégorie différente des recherches se concentre sur des solutions pour prendre en charge les racines mobiles pour étendre la durée de vie du réseau et résoudre le problème de point chaud [17]. Dans d'autres situations, on peut trouver des racines statiques qui ne bougent pas ou il faut optimiser les flux de trafic depuis un nœud mobile vers une racine statique. Pour cela, un nœud mobile sélectionne un nouveau nœud parent optimal pendant son mouvement.

2.5 État de l'art

De nombreuses études ont évalué les performances de RPL dans différents cadres de simulation et de scénarios, plusieurs solutions ont été proposées pour supporter la mobilité dans le protocole RPL.

Les auteurs dans [5] ont introduit un mécanisme corona nommé Co-RPL (RPL Routing for Mobile Low Power Wireless Sensor Networks using Corona Mechanism). Co-RPL est une extension de RPL, elle permet de fournir un support de mobilité en gardant les traces des positions des nœuds mobiles afin d'améliorer les performances du réseau. Cette approche repose sur le mécanisme corona qui permet la localisation des routeurs RPL en mouvement. Co-RPL divise le réseau en zones circulaires autour des racines DAG appelées coronas ou les routeurs RPL calculent ses coordonnées en fonction de sa distance par rapport à la racine DAG. Ce mécanisme repose sur un modèle de mobilité aléatoire des points de cheminement, il intègre trois nouveaux mécanismes. Le mécanisme corona pour le calcul des ID corona pour

chaque nœud appartenant à un DAG, les algorithmes distribués pour le fonctionnement du nœud racine et du nœud mobile du DAG et un mécanisme de récupération de chemin qui permet une récupération rapide en cas de défaillance du nœud. L'objectif de simulation est d'examiner l'impact de plusieurs paramètres sur le comportement du réseau et de comparer les résultats avec le protocole RPL standard, ces résultats montrent que Co-RPL permet d'obtenir un PDR plus élevé, moins de retard de bout en bout et une meilleure énergie que le protocole RPL natif.

Dans [8], les auteurs ont proposé une implémentation du protocole RPL appelée D-RPL « RPL dynamique » conçue pour les réseaux où les nœuds peuvent être attachés à des personnes ou des objets. Cette implémentation propose une nouvelle fonction objectif dynamique appel (D-OF) pour améliorer le taux de livraison de paquets (PDR), le délai de bout en bout et la consommation d'énergie, et une minuterie d'entretien inversée contrôlée basée sur la force du signal reçu (RSSI), cela permet de maintenir une réactivité élevée avec un minimum de surcharge et consulter la fonction objectif lorsqu'un mouvement ou une incohérence est détectée. Les résultats de la simulation montrent que le D-RPL améliore le PDR et l'efficacité énergétique du réseau, il s'adapte mieux aux changements de mobilité. Les résultats de la simulation montrent également l'importance de la fonction objectif et son impact sur la gestion de la mobilité en RPL. La fonction objectif proposée D-OF complète le fonctionnement de D-RPL en offrant des performances fiables et un mécanisme de routage efficace.

Une autre solution proposée dans [7] où ils ont introduit un protocole de routage par trajets multiples basé sur RPL appelé DMR où n'importe quel nœud peut être mobile, dans cette solution les chemins sont redondants ce qui permet aux nœuds mobiles de trouver plusieurs chemins alternatifs sur les défaillances de routes locales et globales. Pour une réparation locale, chaque nœud ajoute ses nœuds frères à sa table de routage. Dans le cas où un nœud ne trouve pas des nœuds frères, il demande une reconstitution d'un nouveau DODAG à son puits (réparation globale).

Les auteurs dans [19] ont proposé un protocole d'extension RPL pour la mobilité des nœuds appelé imRPLv1 qui introduit un mécanisme de détection mobile et un autre pour la sélection de parent. Le mécanisme de détection mobile détecte la mobilité du nœud pendant que le nœud mobile s'éloigne du parent d'origine, il est équipé d'une minuterie pour déterminer le délai de réception du message ACK ou les nœuds reçoivent des ACK du nœud racine, si le RSSI de ce dernier est inférieur à un seuil, il sera traité dans la sélection parente, le message DIO avec la fiabilité la plus élevée est sélectionné. ImRPLv2 améliore la détection mobile et le

processus de sélection parente via une nouvelle métrique de coût basée sur le nombre de sauts, les valeurs RSSI et sur la somme des retards pour sélectionner un nouveau parent. Le résultat montre que l'utilisation du processus de détection mobile proposé et de la nouvelle métrique de coût dans imRPLv2 entraîne un retard de bout en bout de imRPLv2 inférieur à imRPLv1, imRPLv2 fournit un taux de livraison de paquets élevé, un délai de bout en bout faible et une augmentation de taux de livraison de paquets.

2.6 Les modèles de mobilité

Plusieurs efforts ont été déployés par des chercheurs pour améliorer les performances du protocole RPL pour gérer les systèmes mobiles, et pour cela, des simulations et des émulations sont obligatoires.

Les modèles de mobilité dans le protocole RPL décrivent le mouvement des nœuds dans les réseaux basse consommation et leur comportement en fonction de leur emplacement, vitesse et de leur direction sur une période de temps, ils ont été développés principalement pour adapter, présenter le modèle de mouvement des appareils IoT. Ces modèles donnent une représentation formelle du mouvement dans le monde réel d'applications, en d'autres termes, ils définissent la manière dont les nœuds se déplacent dans l'environnement. Dans chaque modèle de mobilité, les nœuds sont obligés d'effectuer leurs mouvements en fonction des règles et des restrictions bien déterminées.

Les modèles de mobilité sont essentiellement classés en :

- Modèles stochastiques : Des mouvements aléatoires ou les nœuds sont libres de se déplacer dans n'importe quelle direction tels que Random Waypoint Mobility Model (RWP) et Random Direction Mobility Model (RDM) [20].
- Modèles détaillés : Sont dédiés pour des scénarios spécifiques par exemple dans des salles de classe tels que Street random waypoint (STRAW).
- Modèles hybrides : C'est un équilibre entre les modèles stochastiques et modèles détaillés comme Manhattan Mobility Model (MMM).
- Modèles de trace réels : Des mouvements d'utilisateurs réalistes basés sur des scénarios spécifiques.

2.7 Conclusion

Il y a eu plusieurs études sur les mécanismes de routage qui tiennent en compte la mobilité des nœuds RPL qui sont principalement basées sur des modifications (de la fonction objectif par exemple), il est nécessaire d'avoir une évaluation complète des performances du protocole RPL basé sur la mobilité à l'aide de différents modèles de mouvement dans divers modèles de mobilité. Pour cela, il faut analyser les fonctionnalités de ce protocole à partir des paramètres les plus fondamentaux des nœuds tels que la consommation d'énergie, la fiabilité, la latence etc. C'est ce que nous allons détailler dans le chapitre qui suit.

Chapitre 3

Solution proposée

3.1 Introduction

Après avoir étudié les solutions déjà proposées par les chercheurs pour améliorer le protocole RPL dans le cas de la mobilité des nœuds, nous avons proposé notre solution nommée RSSI-RPL que nous allons l'expliquer en détails dans ce chapitre.

3.2 RSSI

Le RSSI (Received Signal Strength Indicator) est une mesure du niveau de puissance du signal radio reçu d'un nœud voisin [3], il est généralement utilisé pour déterminer la qualité de la liaison entre les nœuds dans un réseau qui peut être soumis à des interférences en raison de la distance et des obstacles physiques, sa valeur est exprimée en décibels milliwatts (dBm).

Dans un réseau, surveiller la qualité de liaison entre les nœuds à l'aide du RSSI, sert à optimiser les paramètres de transmission en sélectionnant les chemins les plus robustes en termes de qualité de signal.

Le RSSI peut également être utilisé par le protocole RPL pour déterminer les voisins les plus proches ou les plus adaptés pour le routage des paquets, en utilisant des métriques basées sur le RSSI pour évaluer la qualité des liens.

3.3 La solution proposée

Pour améliorer les performances du protocole RPL avec des nœuds mobiles, nous avons surveillé la qualité de liaison entre les nœuds en utilisant les RSSI. Ensuite, nous avons fixé une valeur pour la constante $RSSI_{Threshold}$ après avoir lancé des simulations de tests pour évaluer les performances du réseau avec différentes valeurs de seuil.

$RSSI_{Threshold}$ est une valeur seuil utilisée pour déterminer si un nœud est accessible ou non. Si le RSSI mesuré est inférieur à la valeur seuil définie, le nœud voisin sera considéré comme inaccessible et si un nœud établit une connexion avec lui, il risque de perdre la connectivité et il sera obligé de sélectionner un nouveau parent, ce qui augmente la surcharge dans le réseau, le délai, la perte des paquets, etc.

Ensuite, nous avons utilisé $RSSI_{Threshold}$ dans la fonction `calculate_path_metric` dans la fonction objectif MRHOF. Cette fonction sert à calculer la métrique du chemin (`path_metric`) qui représente le coût du chemin d'un nœud à la racine du DODAG.

L'objectif est de comparer la valeur RSSI du parent pris en paramètres $RSSI_{Parent}$ avec la valeur du $RSSI_{Threshold}$. Si elle est supérieure, aucune modification ne sera effectuée, sinon une valeur ($RSSI_{Threshold} - RSSI_{Parent}$) sera ajoutée à la métrique du chemin comme il est expliqué dans l'algorithme dans la figure 3.1.

```
p_rssi= packetbuf_attr(PACKETBUF_ATTR_RSSI); /* Récupérer le RSSI du nœud */
RSSI_THRESHOLD = -75;
if (rssi < RSSI_THRESHOLD)
    return p_path_metric + (RSSI_THRESHOLD - p_rssi);
else
    return p_path_metric;
```

FIGURE 3.1 – Algorithme de la solution proposée.

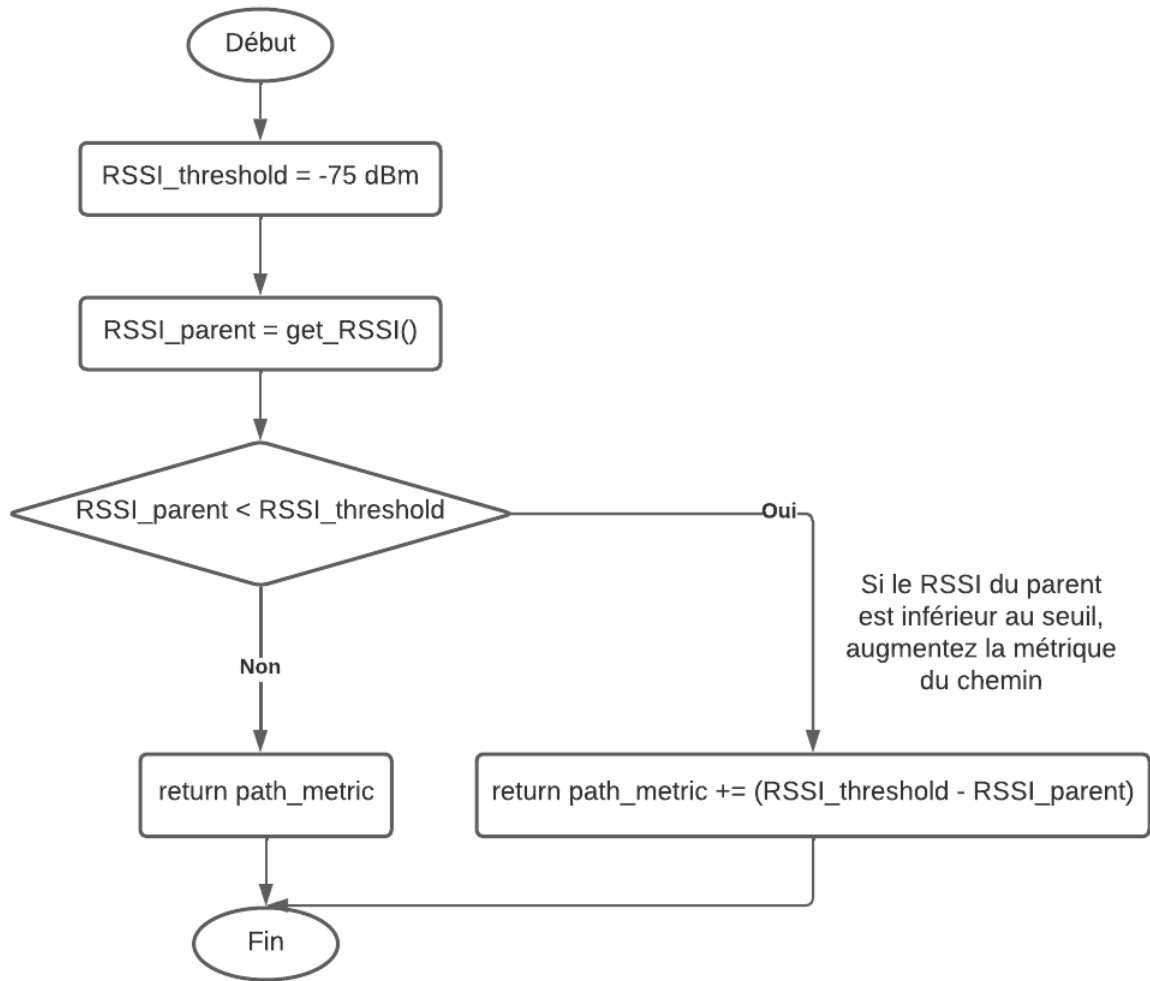


FIGURE 3.2 – Organigramme de la fonction calculate_path_metric.

L'ajout d'une valeur à la métrique du chemin sert à diminuer les chances qu'un nœud ayant un signal faible soit sélectionné comme parent préféré dans la fonction best_parent (voir la figure 3.3). Cette stratégie vise à réduire les changements de la topologie, les pertes de paquets et autres problèmes associés à une connectivité instable. Cette approche renforce la stabilité et la résilience du réseau en offrant une meilleure sélection des parents préférés.

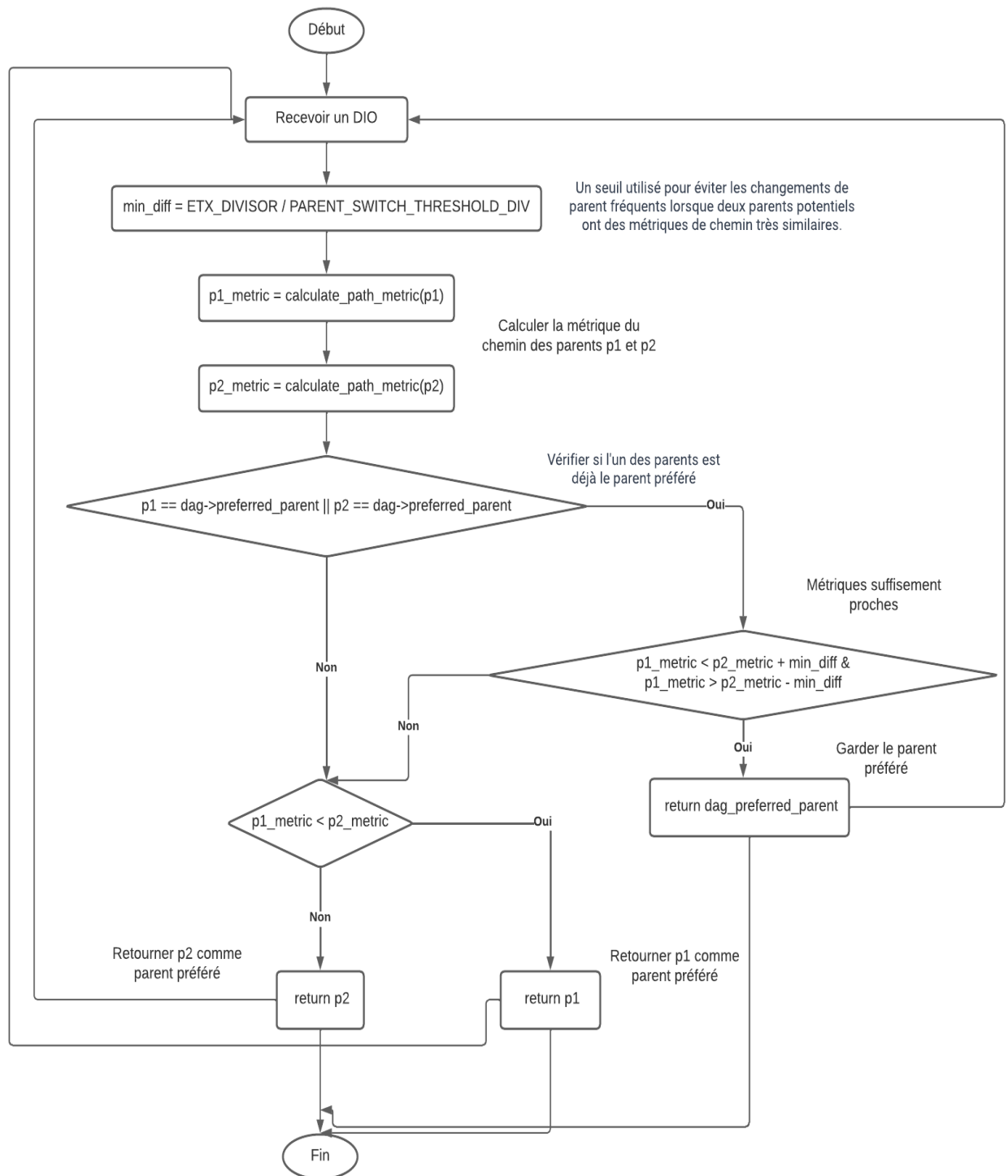


FIGURE 3.3 – Organigramme de la fonction best_parent.

3.3.1 Le modèle de mobilité choisi

Le choix du modèle de mobilité est un aspect crucial dans les simulations des protocoles de routage tels que le protocole RPL. Parmi les nombreux modèles de mobilité disponibles,

nous avons choisi le modèle de mobilité aléatoire RWP (Random waypoint mobility model) pour l'implémentation de notre solution. Cette décision a été prise après avoir examiné les avantages et les inconvénients de plusieurs modèles de mobilité et avoir considéré nos besoins.

Le modèle de mobilité RWP présente plusieurs avantages tels que sa simplicité, sa flexibilité et son réalisme. De plus, étant donné que c'est l'un des modèles de mobilité les plus couramment utilisés, il y a beaucoup de recherches et de documentations disponibles pour faciliter l'utilisation. Ce modèle est connu par des périodes de mouvement aléatoires, chacune suivie d'un temps de pause [18].

Dans un premier temps, chaque nœud choisit une destination aléatoire dans la zone de simulation et se dirige vers elle avec une vitesse choisie uniformément, une fois un nœud arrive à sa destination, il s'arrête pendant un certain temps avant de se diriger vers une autre destination, ce processus porte jusqu'à la fin du temps de simulation.

Le modèle de mobilité aléatoire est un modèle sans mémoire, ce qui implique que les prochaines valeurs de direction et de vitesse sont fixées au hasard, c'est-à-dire les valeurs futures de la direction et la vitesse sont totalement indépendantes des valeurs précédentes.

3.4 Conclusion

Dans ce chapitre, nous avons expliqué notre solution et tous ce que nous avons modifié dans le protocole RPL statique. Dans le chapitre prochain, nous allons évaluer la solution dans un logiciel et analyser les résultats obtenus.

Chapitre 4

Implémentation de la solution

4.1 Introduction

Après avoir proposé une solution qui vise à améliorer les performances du protocole RPL en cas de mobilité des nœuds, nous allons aborder la partie mise en œuvre de notre solution en utilisant l'environnement de développement Cooja qui sera présenté dans ce qui suit, nous allons également montrer les résultats obtenus et quels sont les critères améliorés.

4.2 Environnement de travail

4.2.1 Système d'exploitation d'IoT Contiki

Contiki est un système d'exploitation open-source conçu pour les réseaux de capteurs sans fil, il est utilisé en réseaux systèmes embarqués et objets intelligents. Contiki fournit des bibliothèques pour l'allocation de mémoire, la manipulation de listes chaînées et les abstractions de la communication. C'est le premier système d'exploitation qui a fourni une communication IP, il permet de dialoguer avec n'importe quel matériel supportant le protocole IP, y compris IPv6. Il est développé en C, donc il est hautement portable à différentes architectures.

Contiki est divisé en deux parties : un noyau et des programmes chargés. Le noyau est constitué du noyau Contiki, d'un chargeur du programmes et d'une pile de communication avec des pilotes de périphérique pour le matériel de communication [1]. Le chargeur du programmes charge les programmes dans la mémoire.

Le système d'exploitation Contiki fournit les modules de routage dans un répertoire

séparé "contiki/core/net/rpl", ce répertoire se compose de plusieurs fichiers séparés logiquement en fonction des fonctionnalités qu'ils fournissent tels que rpl-dag.c, rpl-icmp6.c, etc.

4.2.2 Le simulateur COOJA

Cooja est un simulateur basé sur Java avec une interface d'utilisateur graphique conçue pour simuler des réseaux de capteurs où est chargé Contiki. Ce simulateur est implémenté en Java mais permet de charger du code en C dans les capteurs appelés des motes. Cooja permet une simulations à trois niveaux différents : niveau réseau, niveau système d'exploitation et niveau d'instruction du code machine.

Le simulateur Cooja est largement utilisé par les chercheurs dans le domaine d'IoT avec un pourcentage de travaux qui a atteint 63 % en 2017 [1]. Le simulateur Cooja permet aux développeurs d'estimer les consommations électriques des nœuds et de montrer les transmissions et les réceptions radio. Cooja simule le fonctionnement de plusieurs types de capteurs réels tels que Tmote Sky, Z1 et WiSMote qui représentent une interface entre la description matérielle du noeud et le code contiki.

Cooja peut exécuter des programmes Contiki soit compilés nativement sur le CPU ou compilés pour l'émulateur MSP430. Dans ce simulateur, les interactions avec les nœuds sont effectuées via des plugins tels que Timeline et Radio logger, et le stockage de simulation est effectué dans un fichier xml avec l'extension 'csc' (configuration de la simulation Cooja) qui contient des informations sur l'environnement de simulation, les plugins, les nœuds, leurs positions, etc.

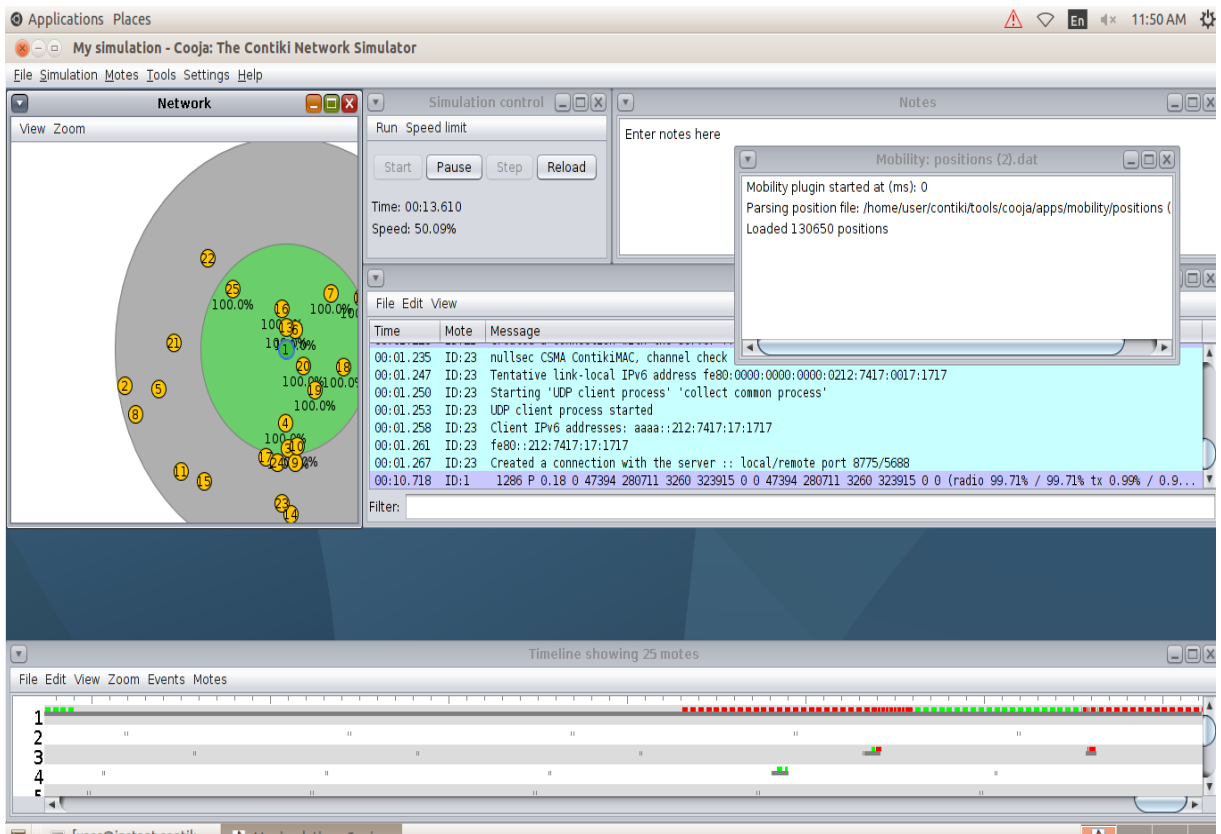


FIGURE 4.1 – Le simulateur Cooja.

4.3 Évaluation des performances de notre solution

4.3.1 Collect View

Dans le cadre de notre évaluation, nous avons utilisé la fonctionnalité Collect View de Cooja pour collecter les données générées par les nœuds lors une simulation. Collect View est un outil de visualisation qui permet d'afficher les données collectées par les nœuds d'un réseau de capteurs sans fil, il permet aux utilisateurs de voir les données de manière graphique ainsi que les métriques de performance du réseau.

Collect View est utilisé pour observer le comportement du réseau dans des conditions différentes, pour détecter les erreurs de communication, améliorer la conception et l'optimisation du réseau de capteurs sans fil.

4.3.2 Configuration de la simulation

L'objectif principal de notre simulation consiste à analyser l'effet de divers paramètres sur les performances d'un réseau de capteurs sans fil et à comparer les résultats du standard

RPL avec le protocole RSSI-RPL.

Pour comprendre les performances du protocole RSSI-RPL, nous avons effectué des simulations détaillées en les comparant avec le protocole RPL standard tout en analysant son effet sur le taux de perte de paquets et la consommation d'énergie. Nous avons fait varier le nombre de nœuds dans un DODAG de 5 jusqu'à 25 nœuds mobiles avec un nœud racine mobile (sink) dans une zone de simulation de 150m x 150m. Ces nœuds se déplacent en fonction du modèle de mobilité aléatoire des points de cheminement à 1-4 m/s avec une pause maximale de 10 secondes, le tableau suivant résume les paramètres de la simulation.

Paramètres	Détails
Simulateur de réseau	Cooja (Contiki OS 3.0).
Zone de simulation	150m x 150m.
Temps de simulation	25 minutes.
Nombre de nœuds	25 nœuds.
Portée de transmission	50 m.
Modèle de mobilité	Modèle de mobilité aléatoire des points de cheminement.

TABLE 4.1 – Paramètres de la simulation.

Dans notre scénario de simulation, nous avons considéré que les nœuds mobiles de faible puissance étaient attachés à des personnes, des objets ou des animaux avec une vitesse maximale de 4 m/s. Cela permet également d'appliquer nos résultats à d'autres applications IoT, telles que les villes intelligentes et la gestion d'usines.

4.3.3 Implémentation du modèle de mobilité

Pour implémenter le modèle de mobilité RWP, nous avons utilisé RWMMSim, qui est un code générateur de positions du modèle RWP adapté pour fonctionner avec le simulateur Cooja et écrit en python [11]. Il permet de générer des données de positions pour les nœuds se déplaçant dans une simulation, selon certains paramètres tels que le temps, l'espace, la vitesse et la durée.

Nous avons mis à jour le code afin de contrôler le nombre de nœuds, puis nous avons copié le dossier de mobilité dans notre dossier d'applications Cooja. Ce dernier est situé au chemin `"/home/user/contiki/tools/cooja/apps/".`

Le code générateur a été adapté pour permettre une génération de mouvement aléatoire, réaliste et diversifié, en prenant compte la distance entre les nœuds.

4.4 Résultats de la simulation

Les résultats de la simulation ont permis de mettre en évidence les performances de la solution proposée. Ces performances ont été évaluées en termes de taux de perte de paquets et de la consommation d'énergie, et ont été comparées avec celles du protocole RPL classique avec mobilité.

4.4.1 Taux de perte de paquets

Les résultats de la simulation présentés dans la figure 4.2 montrent que le taux de perte de paquets diminue légèrement avec l'augmentation de la taille du DODAG. Cela peut s'expliquer par le fait que lorsque le réseau est plus grand, il y a plus de nœuds disponibles pour acheminer les paquets et les nœuds mobiles ont plus de choix pour trouver des chemins alternatifs, ce qui réduit également la perte de paquets.

De plus, les résultats montrent que RSSI-RPL subit une perte de paquets inférieure à celle du protocole RPL. Cela est dû à l'augmentant de métrique du chemin (path metric) du parent dont le RSSI est inférieur au $RSSI_{Threshold}$ qui diminue les chances que ce parent soit choisi comme parent préféré. Cela permet aux nœuds mobiles de choisir un meilleur parent avec une connectivité plus stable, ce qui diminue la probabilité de perdre des paquets en transit.

Enfin, cette amélioration de la performance est essentielle pour les réseaux de capteurs sans fil, où la perte de paquets peut avoir des conséquences néfastes sur la collecte et la transmission des données.

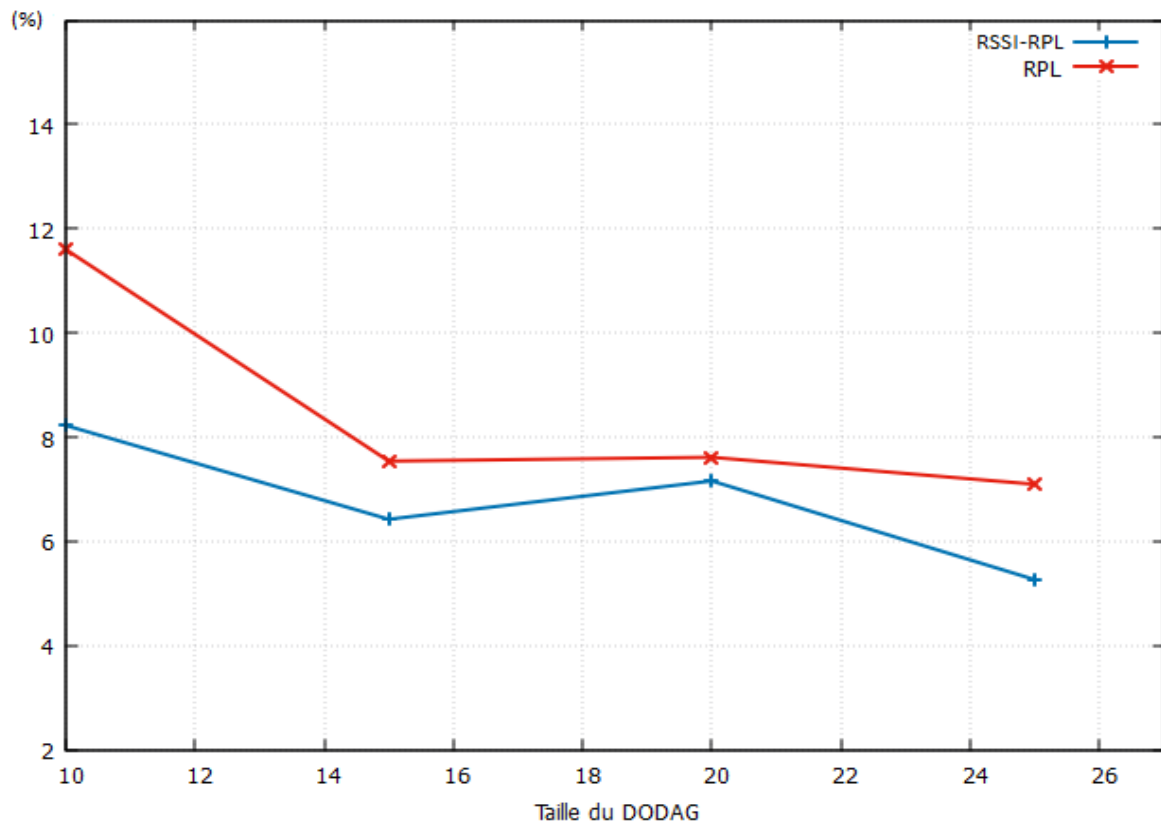


FIGURE 4.2 – Taux de perte de paquets.

4.4.2 Consommation d'énergie

La figure 4.3 montre que la consommation d'énergie augmente avec l'augmentation de la taille du DODAG. Cela est dû au fait que plus le DODAG est grand, plus il y a de nœuds dans le réseau, ce qui entraîne une augmentation de la quantité de messages de contrôle échangés entre les nœuds pour maintenir le DODAG.

Mais, les résultats montrent également que RSSI-RPL consomme moins d'énergie par rapport au RPL standard, cela est attendu car le principe d'augmentation de la métrique du chemin (path metric) lorsque le RSSI du parent est inférieur au $RSSI_{Threshold}$ évite les nœuds mobiles de maintenir des connexions avec des parents instables qui ont une mauvaise qualité de transmission, ce qui réduit la consommation d'énergie des nœuds.

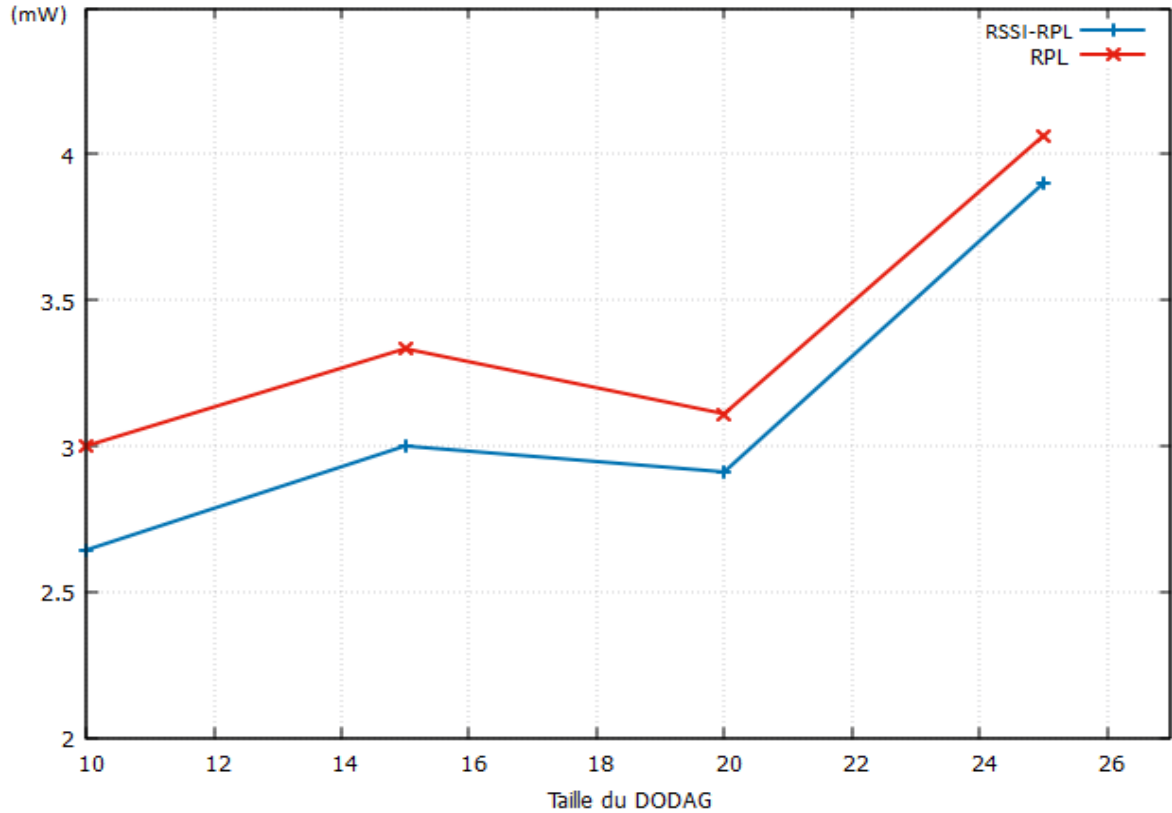


FIGURE 4.3 – Consommation d'énergie.

4.5 Conclusion

Dans ce chapitre, nous avons expliqué par beaucoup de détails de code l'implémentation de notre solution en utilisant Cooja sur Contiki. Les simulations ont produit des résultats satisfaisants en termes de performances, avec une diminution notable du taux de perte de paquets ainsi que de la consommation d'énergie.

Conclusion générale

Le protocole de routage pour les LLN RPL est devenu rapidement le protocole de routage de facto pour l'IoT, étant donné que ce protocole est un élément crucial de l'architecture du réseau. En raison de son importance pour soutenir un grand nombre d'applications IoT potentielles dans des environnements difficiles, y compris les applications en temps réel dans des zones inaccessibles, le protocole RPL joue un rôle clé. Cependant, ce protocole souffre de certains problèmes qui peuvent avoir des conséquences rigoureuses sur la fiabilité des réseaux.

L'objectif principal de ce mémoire est d'améliorer les performances du protocole RPL dans le cas de la mobilité des nœuds. Nous avons tout d'abord consacré notre temps à l'étude du fonctionnement du protocole RPL, nous avons parlé de son fonctionnement, sa méthode de construction de la topologie, ses fonctions objectives, ainsi que les diverses métriques prédéfinies.

Ensuite, nous avons exploré en détail les scénarios de mobilité associés au protocole de routage RPL, ainsi que les problèmes majeurs de la mobilité qui peuvent se poser. Nous avons également étudié l'état de l'art en matière de solutions proposées par les chercheurs pour améliorer la gestion de la mobilité dans ces réseaux et nous avons analysé les métriques de performances fournies par le protocole RPL, qui permettent d'évaluer sa capacité à gérer les mouvements des nœuds dans un réseau de capteurs sans fil.

Enfin, nous avons introduit le contexte de notre solution en expliquant comment nous avons utilisé le RSSI pour améliorer les performances du protocole RPL. Nous avons également fourni des diagrammes de classe pour renforcer notre explication. Le dernier chapitre de notre mémoire décrit l'implémentation de notre solution en utilisant le simulateur Cooja, ainsi que les paramètres et les résultats de la simulation. Afin de faciliter la compréhension et l'analyse des performances de notre solution, nous avons présenté les résultats sous forme de courbes. Ces dernières ont permis de mettre en évidence que les nœuds utilisant RSSI-RPL consomment moins d'énergie et perdent moins de paquets que ceux utilisant le protocole RPL.

Comme perspective, nous allons essayer d'améliorer nos résultats et étendre les tests et

les évaluations sur des scénarios réels.

Bibliographie

- [1] Hazrat Ali. A performance evaluation of rpl in contiki, 2012.
- [2] Prabhugoud I Basarkod and Sunilkumar S Manvi. Mobility and qos aware anycast routing in mobile ad hoc networks. *Computers & Electrical Engineering*, 48 :86–99, 2015.
- [3] Karl Benkic, Marko Malajner, P Planinsic, and Z Cucej. Using rssi value for distance estimation in wireless sensor networks based on zigbee. In *2008 15th international conference on systems, signals and image processing*, pages 303–306. IEEE, 2008.
- [4] Yacine Challal. Réseaux de capteurs sans fils. *Cours, Systèmes Intelligents pour le Transport, Université de Technologie de Compiègne, France*, 17, 2008.
- [5] Olfa Gaddour, Anis Koubäa, Raghuraman Rangarajan, Omar Cheikhrouhou, Eduardo Tovar, and Mohamed Abid. Co-rpl : Rpl routing for mobile low power wireless sensor networks using corona mechanism. In *Proceedings of the 9th IEEE international symposium on industrial embedded systems (SIES 2014)*, pages 200–209. IEEE, 2014.
- [6] Omprakash Gnawali and Philip Levis. The minimum rank with hysteresis objective function. Technical report, 2012.
- [7] Ki-Sup Hong and Lynn Choi. Dag-based multipath routing for mobile sensor networks. In *ICTC 2011*, pages 261–266. IEEE, 2011.
- [8] Harith Kharrufa, Hayder Al-Kashoash, Yaarob Al-Nidawi, Maria Quezada Mosquera, and Andrew H Kemp. Dynamic rpl for multi-hop routing in iot applications. In *2017 13th Annual Conference on wireless on-demand network systems and services (WONS)*, pages 100–103. IEEE, 2017.
- [9] Kevin C Lee, Raghuram Sudhaakar, Lillian Dai, Sateesh Addepalli, and Mario Gerla. Rpl under mobility. In *2012 IEEE consumer communications and networking conference (CCNC)*, pages 300–304. IEEE, 2012.

- [10] Philip Levis, Thomas Clausen, Jonathan Hui, Omprakash Gnawali, and JeongGil Ko. The trickle algorithm. Technical report, 2011.
- [11] M. Linderoth. RWMMSim mobility plugin, 05 2018.
- [12] Anthéa Mayzaud. *Monitoring and Security for the RPL-based Internet of Things*. PhD thesis, Université de Lorraine, 2016.
- [13] Mr MA MESSAOUDI and Mr A BENZERBADJ. Optimisation du routage dans les réseaux IInS.
- [14] Geoff Mulligan. The 6lowpan architecture. In *Proceedings of the 4th workshop on Embedded networked sensors*, pages 78–82, 2007.
- [15] Afonso Oliveira and Teresa Vazão. Low-power and lossy networks under mobility : A survey. *Computer networks*, 107 :339–352, 2016.
- [16] Aishwarya Parasuram, David Culler, and Randy Katz. An analysis of the rpl routing standard for low power and lossy networks. *Electrical Engineering and Computer Sciences University of California at Berkeley*, 2016.
- [17] Leila Ben Saad and Bernard Tourancheau. Sinks mobility strategy in ipv6-based wsns for network lifetime improvement. In *2011 4th IFIP International Conference on New Technologies, Mobility and Security*, pages 1–5. IEEE, 2011.
- [18] Bardia Safaei, Aliasghar Mohammadsalehi, Kimia Talaei Khoosani, Saba Zarbaf, Amir Mahdi Hosseini Monazzah, Farzad Samie, Lars Bauer, Jörg Henkel, and Alireza Ejlali. Impacts of mobility models on rpl-based mobile iot infrastructures : An evaluative comparison and survey. *IEEE access*, 8 :167779–167829, 2020.
- [19] Pak Satanasaowapak and Chatchai Khunboa. The improvement of node mobility in rpl to increase transmission efficiency. *International Journal of Electrical and Computer Engineering*, 5(9) :4238–4249, 2019.
- [20] Mihail L Sichitiu. Mobility models for ad hoc networks. *Guide to Wireless Ad Hoc Networks*, pages 237–254, 2009.
- [21] Harald Sundmaeker, Patrick Guillemin, Peter Friess, Sylvie Woelfflé, et al. Vision and challenges for realising the internet of things. *Cluster of European research projects on the internet of things, European Commision*, 3(3) :34–36, 2010.
- [22] Pascal Thubert. Objective function zero for the routing protocol for low-power and lossy networks (rpl). Technical report, 2012.

- [23] Tim Winter, Pascal Thubert, Anders Brandt, Jonathan Hui, Richard Kelsey, P. Levis, Kris Pister, Rene Struik, J. P. Vasseur, and Roger Alexander. RPL : IPv6 Routing Protocol for Low-Power and Lossy Networks. Request for Comments RFC 6550, Internet Engineering Task Force, March 2012.