

République Algérienne Démocratique et Populaire

Université Abou Bakr Belkaid– Tlemcen

Faculté des Sciences

Département d'Informatique

Mémoire de fin d'études

Pour l'obtention du diplôme de Master en Informatique

Option: Réseaux et Systèmes Distribués (R.S.D)

Thème

**Routage basé sur la QoS dans l'Internet des
Objets (IoT)**

Réalisé par :

- Melle CHERIFI Ibtissam Rania

Présenté le 22 Juin 2020 devant le jury composé de :

Président : - Mr BENAMAR Abdelkrim

Encadrant : - Mr LEHSAINI Mohamed

Examineur : - Mr BENMAMMAR Badr.

Co-encadrante : - Melle BOUZEBIBA Hadjer

Année Universitaire : 2019-2020

Remerciements

وَمَا تَوْفِيقِي إِلَّا بِاللَّهِ

Tout d'abord je remercie DIEU le tout puissant, qui m'a donné la force, la santé, le courage et la patience d'accomplir ce modeste travail.

Je remercie profondément mon encadreur Mr LEHSAINI, d'avoir accepté de m'encadrer, pour la confiance qui m'a donné, et pour tous ces précieux conseils.

Je remercie ma chère Melle BOUZEBIBA Hadjer, que je considère comme une sœur, pour son appui, son encouragement et ça présence à mes côtés à tout moment.

Je remercie les membres du jury d'avoir accepté de juger mon modeste travail.

J'adresse aussi mes remerciements à tous les professeurs qui m'ont enseignés durant le cursus universitaire.

Et le grand merci à mes parents, la source de mon bonheur qui mon suivie et encourager durant tous mon parcours.

Merci



Dédicace

Je dédie ce modeste travail

A celle qui m'a donné la vie, ma maman chérie, le symbole de tendresse, pour tout son soutien, ces sacrifices et ces prière pour moi.

A mon cher papa, mon ombre durant toutes ces années d'études, pour tous son amour et ces conseils précieux.

A mes chères sœurs Manal, Yasmine et sawssen que j'aime plus que tous.

A Ssam qui a cru en moi, que je trouvais à tous moment.

A mes chers petits neveux et nièces d'amours.

A mes chères copines Amina ,Nedjma et Rania que j'aime si fort.

A toutes ma famille.

A toute ma promo en particulier Amel, Feriel et Yasmine que j'ai passé avec mes meilleurs moments.

Table des matières

Introduction générale.....	1
Chapitre I L'IoT et les réseaux LLN	3
I.1 Introduction	3
I.2 L'Internet des objets (IoT)	3
I.2.1 Définition.....	3
I.2.2 Les composants d'un système IoT.....	4
I.2.3 Domaines d'application de l'IoT	5
I.3 Les réseaux de capteurs sans fil (RCSF)	7
I.3.1 Présentation d'un nœud capteur.....	7
I.3.2 Le capteur intelligent	8
I.3.3 Topologies des réseaux de capteurs sans fil	8
I.4 Les réseaux LLN (Low power and Lossy Networks)	10
I.5 6LoWPAN.....	11
I.6 Conclusion.....	11
Chapitre II Le routage dans les réseaux LLN.....	12
II.1 Introduction.....	12
II.2 Routage dans les réseaux LLN	12
II.2.1 Définition du routage	12
II.2.2 Les types de communications	13
II.2.3 Contraintes de routage dans les réseaux LLN	13
II.3 Classification des protocoles de routage.....	13
II.3.1 Protocoles de routage basés sur établissement de routes	14
II.3.2 Protocoles de routage basé sur la structure réseau	16
II.4 Les critères de performance des protocoles de routage	17
II.5 Routage basé sur la qualité de service	17
II.6 Conclusion	18
Chapitre III Le protocole de routage RPL.....	19
III.1 Introduction	19
III.2 RPL (Routing Protocol for Low power and lossy networks-LLN)	19
III.3 Fonctionnement du protocole RPL.....	20
III.3.1 Graphes DAG & DODAG	20

III.3.2	Modèle réseau de RPL	20
III.3.3	Messages de contrôle du protocole RPL.....	20
III.3.4	Fonction Objectif (OF).....	21
III.3.5	Procédure de construction du DODAG.....	22
III.3.6	Modes d'opération du protocole RPL	24
III.3.7	Principales métriques	25
III.3.8	Trafic supporté par un DODAG	26
III.4	Conclusion	26
Chapitre IV	Améliorations du protocole RPL.....	27
IV.1	Introduction	27
IV.2	Outils de simulation.....	27
IV.2.1	Contiki	28
IV.2.2	Simulateur Cooja.....	29
IV.3	Travail réalisé	31
IV.3.1	Problématique.....	31
IV.3.2	Contribution	32
IV.4	Résultats et discussion.....	35
IV.4.1	Evaluation de l'overhead	36
IV.4.2	Evaluation de l'énergie consommée	39
IV.4.3	Evaluation du délai	40
IV.5	Conclusion	41
	Conclusion générale	42
	Annexe	44

Liste des Figures

Figure I-1: Infrastructure d'Internet des objets	4
Figure I-2: Exemple de Smart Home	5
Figure I-3: Système de santé électronique	6
Figure I-4: Exemple de technologie portable	6
Figure I-5: Exemple d'un réseau de capteurs sans fil	7
Figure I-6: Architecture d'un nœud capteur	8
Figure I-7: Une architecture typique de réseau en toile [6].....	9
Figure I-8: Une architecture typique de réseau en étoile [6].....	9
Figure I-9: Types de trafic dans les réseaux LLN	11
Figure II-1: Types de cardinalité de communication	13
Figure II-2: Classification des protocoles de routage.....	14
Figure II-3: Fonctionnement d'un protocole proactif [8].....	15
Figure II-4: Fonctionnement d'un protocole réactif [8]	15
Figure III-1: Exemple de graphes (DAG et DODAG).....	20
Figure III-2: Diffusion des DIO par la racine [11].....	23
Figure III-3: Choix du root (racine) comme nœud parent [11]	24
Figure III-4: Poursuite de la construction du DODAG [11]	24
Figure III-5: Modes de fonctionnement du protocole RPL.....	25
Figure IV-1: Architecture de Contiki	29
Figure IV-2: Interface du simulateur Cooja	30
Figure IV-3: Nouveau nœud qui veut joindre le DODAG.....	31
Figure IV-4: Stopper la réception des DIS et l'envoi des DIO	32
Figure IV-5: Organigramme de la solution proposée.....	33
Figure IV-6: Fonction de mise à jour des tables	34
Figure IV-7: Fonction dis_input.....	35
Figure IV-8: Exemple de simulation	36
Figure IV-9: Overhead en fonction du temps (20 nœuds).....	37
Figure IV-10: Overhead en fonction de la taille du réseau	38
Figure IV-11: Overhead en fonction de l'intervalle DIS	38
Figure IV-12: Energie consommée en fonction du temps.....	39
Figure IV-13: Energie consommé en fonction de la taille du réseau	40
Figure IV-14: Variation du délai en fonction de l'intervalle DIS.....	41

Liste des Tableaux

Tableau IV-1 : Caractéristiques de la machine utilisée pour les simulations.....	28
Tableau IV-2: Paramètres de simulation	36

Liste des abréviations

6LoWPAN	ipv6 Low power Wireless Personal Area Networks
AODV	Ad hoc On Demand Distance Vector
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 Directed Acyclic Graph
ETX	Expected Transmission count
HC	Hop Count
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
IHM	Interfaces Homme Machine
IoT	Internet Of Things
IPv4	Internet Protocol Version 4
IPv6	Internet Protocol Version 6
LBR	Low power and lossyBorder Router
LLN	Low power and Lossy Networks
MP2P	Multipoint-to- <i>Point</i>
MRHOF	Minimum Rank with Hysteresis Objective Function
OCP	Objective Code Point
OF	Objective Function
OF0	Objective Function Zero

OLSR	Optimized Link State Routing Protocol
P2MP	<i>Point-to-Multipoint</i>
P2P	<i>Point-to-Point</i>
QoS	Quality of Service
RCSF	Réseau de Capteurs Sans Fil
RPL	Routing Protocol for Low-power and lossy networks
SICS	Swedish Institute of Computer Science
UIT	Union Internationale des Télécommunications

Résumé

Ces dernières années, l'Internet des objets (IoT) a connu une grande croissance vu que notre besoin quotidien des objets intelligents est devenu une nécessité pour rendre nos tâches plus faciles. Ces objets sont souvent très contraints en énergie et peuvent former des réseaux à faible puissance et à pertes (LLN).

IETF a proposé au niveau de la couche réseau le protocole RPL comme standard de routage pour les réseaux dont les nœuds sont à ressources limitées tel que les réseaux LLN. Néanmoins, ce protocole présente toujours quelques anomalies qui nécessitent d'être améliorées. Ainsi, dans le cadre de notre PFE, nous avons visé à apporter une amélioration du protocole RPL en particulier lors de l'évolutivité du réseau en cas de la saturation de la mémoire d'un nœud. Dans ce contexte, nous avons proposé une solution qui consiste à éviter les messages inutiles en cas de saturation de la mémoire d'un nœud. Cette solution nous a permis de réduire le surcoût (messages de contrôles).

Nous avons validé notre contribution à travers plusieurs simulations et les résultats obtenus étaient prometteurs en termes de consommation d'énergie, d'overhead et de délai de bout en bout.

Mots-clés : IoT, QoS, RPL, Routage, Simulations, Réseaux LLN, Contiki.

Abstract

In recent years, the Internet of Things (IoT) has grown significantly, as our daily need for smart objects has become a necessity to make our tasks easier. These objects are often very constrained in energy and can form networks with low power and losses (LLN).

IETF proposed at the network layer the RPL protocol as a routing standard for networks whose nodes are limited resources such as LLN networks. However, this protocol still has some anomalies that need to be improved. Thus, within the framework of our PFE, we aimed to bring an improvement of the RPL protocol in particular during the scalability of the network in the event of the saturation of the memory of a node. In this context, we proposed a solution, which consists in avoiding the useless messages in the event of saturation of the memory of a node. This solution allowed us to reduce the additional cost (control messages).

We validated our contribution through several simulations and the results obtained were promising in terms of energy consumption, overhead and end-to-end delay.

Keywords : IoT, QoS, RPL, Routing, Simulations, LLN Networks, Contiki.

ملخص

في السنوات الأخيرة ، نمت إنترنت الأشياء (IoT) بشكل كبير حيث أصبحت حاجتنا اليومية للأشياء الذكية ضرورة لتسهيل مهامنا. غالبًا ما تكون هذه الأشياء مقيدة للغاية في الطاقة ويمكن أن تشكل شبكات ذات طاقة منخفضة وخسائر (LLN). اقترح IETF على مستوى طبقة الشبكة بروتوكول RPL كمعيار توجيه للشبكات التي تكون عناصرها ذات إمكانيات محدودة كما هو الشأن في شبكات LLN. ومع ذلك ، لا يزال هذا البروتوكول يحتوي على بعض النقائص التي تحتاج إلى تحسين. وبالتالي ، في إطار هذه المذكرة، سعينا إلى تحسين بروتوكول RPL على وجه الخصوص أثناء قابلية التوسع في الشبكة في حالة تشبع ذاكرة أحد العناصر. في هذا السياق ، اقترحنا حلًا يتمثل في تجنب الرسائل عديمة الفائدة في حالة تشبع ذاكرة العنصر المعني. سمح لنا هذا الحل بتخفيض التكلفة الإضافية (رسائل التحكم)،

لقد تحققنا من مساهمتنا من خلال العديد من المحاكاة وكانت النتائج التي تم الحصول عليها واعدة من حيث استهلاك الطاقة و التكلفة الإضافية والتأخير.

الكلمات المفتاحية: IoT, QoS, RPL, Routing, Simulations, LLN Networks, Contiki.

Introduction Générale

Introduction générale

Les progrès technologiques dans les domaines de la communication sans fil et de la microélectronique ont permis d'établir la connexion des objets du quotidien à l'Internet. Ce nouveau mode de connexion a donné naissance à un nouveau paradigme appelé l'Internet des objets "Internet of Things (IoT)". Ces objets sont généralement caractérisés par une faible puissance et des communications instables. De ce fait, des solutions interopérables doivent cependant être utilisées pour assurer une communication optimum entre ces objets. Dans ces solutions, le protocole de routage est une pièce maitresse dans ce contexte, car il permet pour chaque objet de décider comment joindre un autre objet. Les contraintes s'appliquant aux objets (faible puissance de traitement, de batterie et de mémoire) doivent être prises en compte pour le développement de protocoles de routage adaptés à cet environnement.

Dans le cadre de ce projet de fin d'études, nous nous focaliserons sur le protocole de routage RPL¹ en présentant les différents mécanismes mis en œuvre dans RPL et leurs limites tout en visant à améliorer ses performances. Ainsi, nous proposons à satisfaire une qualité de service (QoS) respectable quand un nouveau nœud veut rejoindre le réseau et en particulier un DODAG (graphe acyclique orienté vers la destination) dont les nœuds ayant une file d'attente saturée.

Ce travail est structuré en quatre chapitres, organisés selon le plan méthodologique suivant:

- Dans le premier chapitre, on définit le concept de l'Internet des objets, tous en présentant ses composants et en citant ses domaines d'application. On présente également les réseaux à faibles puissance et de pertes (LLN : Low power and Lossy Networks) ainsi que le trafic des données dans ce type de réseaux.
- Le deuxième chapitre est consacré à la présentation du routage et précisément au routage dans les réseaux à faible puissance et de pertes (LLN).
- Dans le troisième chapitre une présentation détaillée du protocole de routage RPL qui fait l'objet de ce projet de fin d'études.
- Le quatrième chapitre décrit le scénario qu'on a proposé ainsi que les améliorations qu'on a apportées avec leurs résultats accompagnés de graphes de comparaison explicatifs.

¹ RPL : Routing Protocol for Low power and lossy networks

Ce travail est terminé par une conclusion générale dans laquelle nous synthétisons l'objet de ce projet de fin d'études et nous présentons les atouts de la solution proposée.

Chapitre I

L'IoT et les réseaux LLN

Chapitre I

L'IoT et les réseaux LLN

I.1 Introduction

L'Internet des objets est une technologie qui permet de connecter n'importe quel ensemble d'objets entre eux [1]. Pour cela, ce nouveau paradigme est devenu un des systèmes les plus puissants pour créer, modifier et partager des informations entre des objets ayant des caractéristiques différentes. Ces objets sont souvent très contraints en énergie et forment des réseaux de faible puissance et à pertes (LLN). Ces réseaux se caractérisent par une faible consommation d'énergie. En outre, ces réseaux possèdent les mêmes caractéristiques que les réseaux de capteurs sans fil (WSNs).

Dans ce chapitre, on présente le contexte de ce travail à savoir l'Internet des objets et en mettant l'accent sur les réseaux de faible puissance.

I.2 L'Internet des objets (IoT)

I.2.1 Définition

L'IoT est en relation avec tout ce qui tourne autour des réseaux de capteurs qui transmettent des informations à des nœuds de décisions qui peuvent à leur tour redescendre à d'autres capteurs d'ordre (actions). Ce nouveau paradigme de communication permet aux objets de communiquer de manière directe ou indirecte avec des équipements électroniques qui sont eux même connectés à Internet. Selon le groupe de travail (IoT-GSI) dirigé par l'Union Internationale des Télécommunications (UIT), l'IoT est une "infrastructure mondiale pour la société de l'information, qui permet de disposer de services évolués en interconnectant des objets (physiques ou virtuels) grâce aux technologies de l'information et de la communication (TIC) interopérables existantes ou en évolution" [2]. De son côté, l'organisme IEEE a défini l'IoT comme un "réseau d'éléments où chaque élément est muni de capteurs et il est connecté à Internet". La Figure I-1 illustre un exemple d'équipements qui sont connectés à Internet et qui représentent un réseau sous-jacent à l'IoT.



Figure I-1: Infrastructure d'Internet des objets

I.2.2 Les composants d'un système IoT

Une solution sous-jacente à l'IoT s'articule essentiellement autour des composants suivants:

- Les objets : Un objet connecté est un objet physique équipé de capteurs ou d'une puce qui lui permet de transcender son usage initial afin de proposer de nouveaux services.
- Les capteurs : Ce sont tous les équipements actifs ou passifs pouvant générer des données et qui sont installés sur les objets connectés.
- Le réseau est le maillon principal d'une solution IoT. Il doit répondre à un critère d'usage notamment [3] :
 - La couverture de la zone d'usage des objets,
 - L'architecture de la solution,
 - La conception de l'objet,
 - Le cycle de vie de la solution
- Les données : Ce sont les éléments bruts collectés par les objets ou les outils mis en place pour l'IoT.
- Le traitement de données : enregistre et analyse les données collectées pour lancer éventuellement des actions ou pour aider à la prise de décisions.
- Les applications d'exploitation : Ce sont les interfaces homme-machine (IHM) dans lesquelles nous pouvons visualiser d'une manière ergonomique les données sous forme de tableau de bord.

I.2.3 Domaines d'application de l'IoT

De nombreux domaines ont été visés par l'IoT tels que le domaine personnel, le domaine du transport, l'environnement et les services publics, ainsi que plusieurs secteurs ont bénéficié des atouts de l'IoT tels que : la santé, l'agriculture, et l'industrie, etc.

Dans ce qui suit, nous présentons quelques domaines d'applications afin d'illustrer l'intérêt de cette nouvelle technologie :

- a) **Smart Home** : L'avènement de la maison intelligente a pu être possible grâce éléments suivants : Internet, objets connectés et domotique. Ainsi, à travers des équipements pilotables à distance, il est devenu possible de modifier de la température d'un climatiseur ou d'une cuisinière, de contrôler d'éclairage et même la surveiller nos maisons. Smart Home a pour but d'apporter le confort et la sécurité aux occupants de l'habitat et de promouvoir l'autonomie des personnes dépendantes (âgées et handicapées). La Figure I-2 illustre un exemple d'une maison intelligente où tous les équipements de cet habitat sont connectés à Internet.

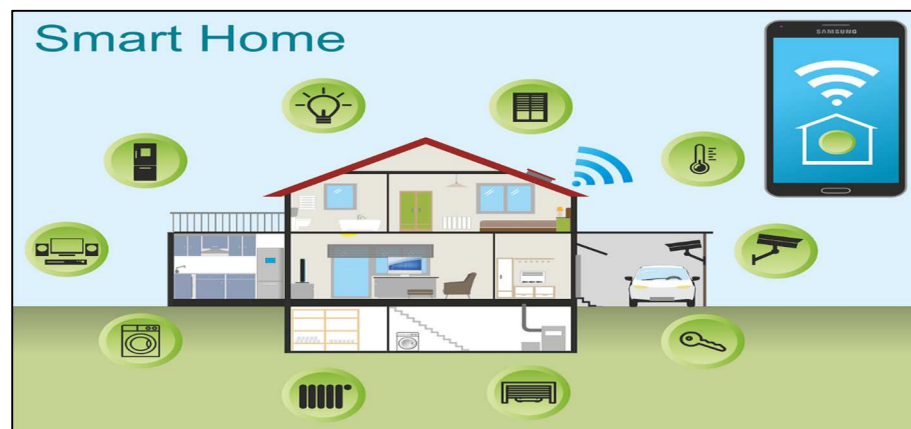


Figure I-2: Exemple de Smart Home

- b) **Le Système de Santé Electronique** : L'IoT a rapidement transformé la prestation de soins de la manière traditionnelle à la manière intelligente. Les équipements et les capteurs sont devenus de plus en plus "intelligents" et génèrent toujours plus de données nécessaires aux équipements médicaux. Dans ce cas de figure, l'IoT consiste en la mise en réseau d'objets physiques tels que les capteurs embarqués, les actionneurs et d'autres équipements qui collectent et transmettent des informations sur l'activité de l'organe surveillé en temps réel à travers le réseau. Par ailleurs, l'utilisation de l'IoT dans la santé a pour but de rendre les réseaux des hôpitaux plus intelligents, en surveillant de manière proactive l'infrastructure critique et en automatisant le déploiement et la gestion de l'infrastructure informatique [4].

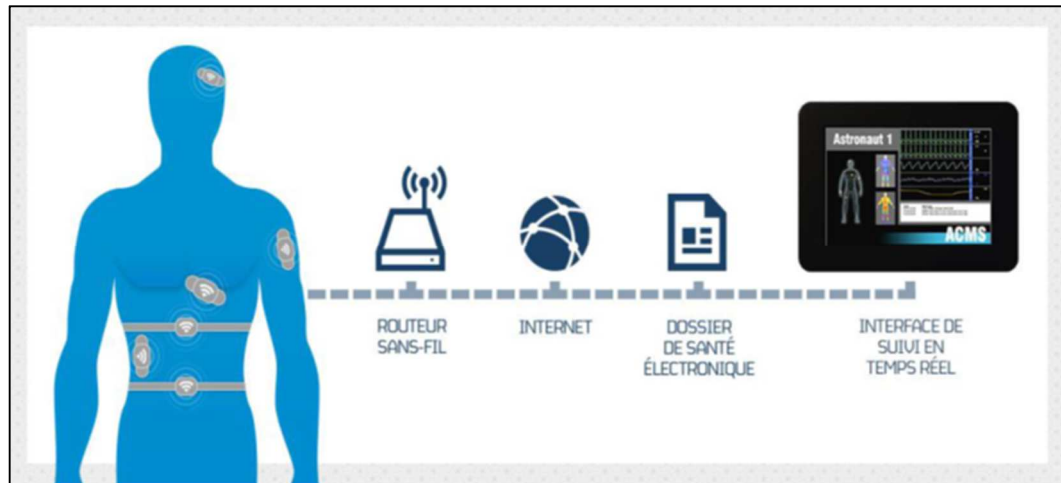


Figure I-3: Système de santé électronique [4]

- c) **Technologie portable (Wearables)** : représente un vêtement ou un accessoire comportant des éléments informatiques et électroniques avancés. La technologie portable devient essentielle pour un mode de vie sain, dans les domaines de la santé et de la forme physique. Elle fait référence aux technologies électroniques ou aux puces qui sont incorporés dans des vêtements et des accessoires qui peuvent être portés confortablement sur le corps. Dans ce contexte, on donne comme exemple le T-Shirt intelligent développé par Bourouis au niveau du laboratoire STIC. Ce T-shirt comporte une puce pour faciliter la localisation d'un enfant égaré ou une personne âgée ayant l'Alzheimer. Un autre exemple consiste à implanter puce dans le corps des personnes cardiaques et les contrôler à distance à travers Internet.



Figure I-4: Exemple de technologie portable

I.3 Les réseaux de capteurs sans fil (RCSF)

Les réseaux de capteurs sans fil (RCSF) sont composés généralement d'un grand nombre de petits dispositifs (des centaines voire des milliers), qui communiquent entre eux via des liens radio (sans fil) de faible portée pour le partage d'information et le traitement coopératif [5]. Ces dispositifs appelés nœuds capteurs, sont déployés manuellement (dans des positions prédéterminées) ou aléatoirement dans un environnement pour surveiller des phénomènes bien particuliers. Après le déploiement initial, les nœuds capteurs s'auto-organisent en une infrastructure réseau appropriée, souvent en mode multi-sauts. Les données collectées par ces nœuds capteurs sont acheminées directement ou via un schéma de routage multi-sauts à un nœud considéré comme "point de collecte", appelé station de base ou "Puits ou Sink". Ce nœud de collecte peut être connecté à une machine puissante via Internet ou par satellite. En outre, les nœuds capteurs peuvent être interrogés par des requêtes à distance ou envoient périodiquement des données au nœud de collecte et dans les applications sensibles les nœuds capteurs envoient l'information lorsqu'un événement pertinent survient. Par ailleurs, l'aspect autonome rend les réseaux de capteurs très intéressants, vu l'intelligence des capteurs qui permet de décentraliser l'intelligence du réseau.

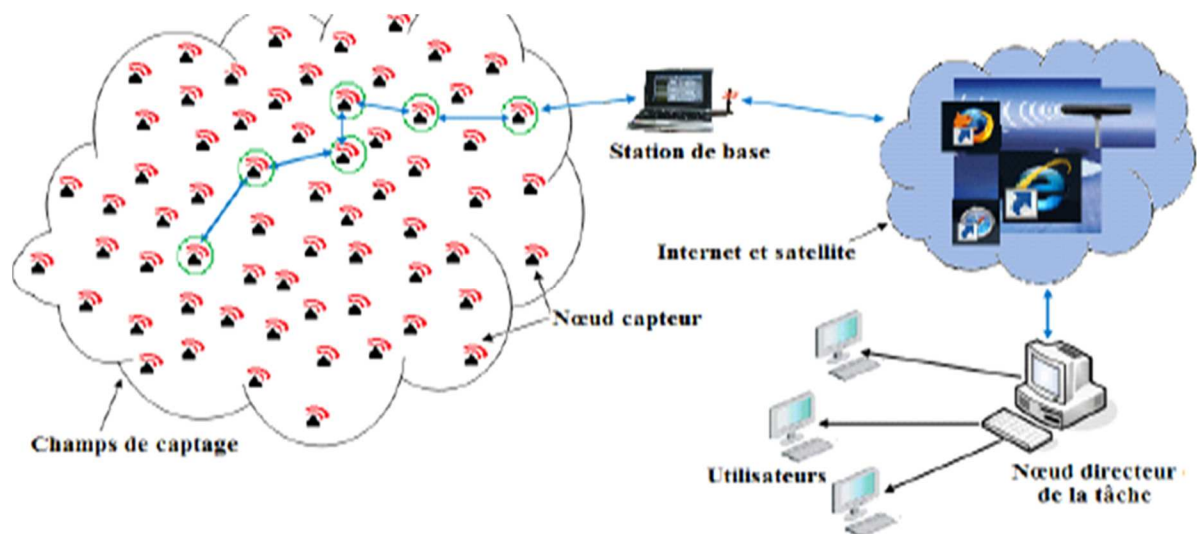


Figure I-5: Exemple d'un réseau de capteurs sans fil [5]

I.3.1 Présentation d'un nœud capteur

Un nœud capteur est un petit dispositif électronique doté d'un ou plusieurs capteurs qui sont capables de mesurer une grandeur physique environnementale (température, luminosité, pression, etc.) ou physiologique (glycémie, tension, etc.), et de la transmettre à un centre de contrôle via une station de base.

Un nœud capteur est capable de transformer une grandeur physique observée (température, humidité, etc.) en une grandeur numérique (intensité électrique, position d'un flotteur). Il dispose d'au moins un transducteur ADC dont le rôle est de convertir une grandeur physique en grandeur numérique calculable.

Grâce aux progrès technologiques dans le domaine de la miniaturisation, les nœuds capteurs sont devenus des éléments de très petite taille et de bon marché. Ils sont équipés également de composants leur permettant : de stocker plus de données, de réaliser des traitements sur ces données et de les communiquer au monde réel via des liens sans fil.

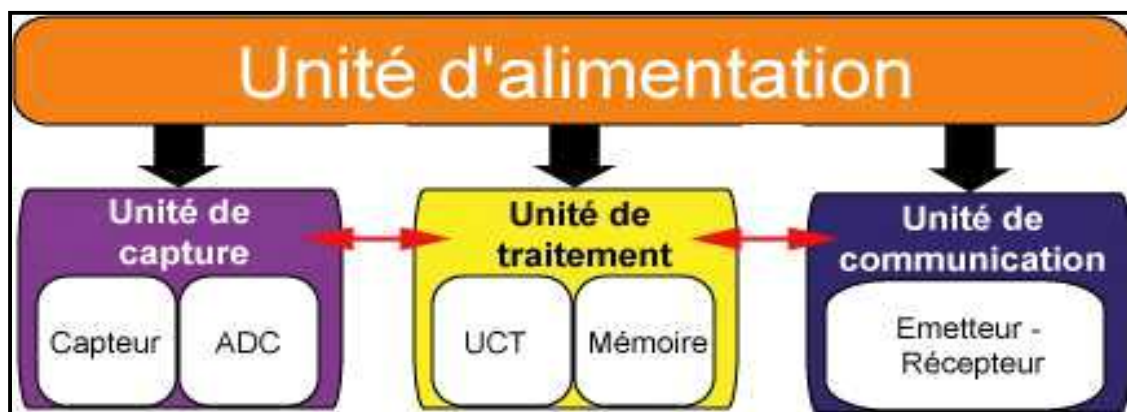


Figure I-6: Architecture d'un nœud capteur [5]

I.3.2 Le capteur intelligent

La notion de capteur intelligent (smart sensor) a été introduite dans l'industrie des capteurs pour distinguer entre les capteurs qui permettent de traiter les données collectées et les analyser de ceux qui transmettent les données brutes sans aucun traitement supplémentaire. Par rapport, à un capteur conventionnel, un capteur intelligent contient d'autres composants électroniques tels que des unités programmables et des logiciels nécessaires au traitement des données. Par conséquent, il se distingue par sa capacité à collecter des mesures, à les traiter et à les communiquer au monde extérieur.

I.3.3 Topologies des réseaux de capteurs sans fil

Dans les réseaux de capteurs sans fil, deux paramètres importants sont à prendre en compte :

- Zone de couverture : Dans cette zone de déploiement tout point doit être couvert par au moins un capteur et tout capteur doit être en contact avec au moins un nœud de décisions.
- L'aspect dynamique de la topologie : c'est l'un des paramètres les plus importants dans un RCSF, ça correspond à l'architecture logique d'un réseau, définissant les liaisons entre les

objets du réseau (capteurs et points de collecte) et une hiérarchie éventuelle entre eux [5].

Parmi ces topologies, on a :

a) Topologie en toile ou grille (Mesh Network)

La topologie en toile est une infrastructure dans laquelle la communication entre les nœuds et la station de base pourrait être directe ou via un schéma de routage multi-sauts.

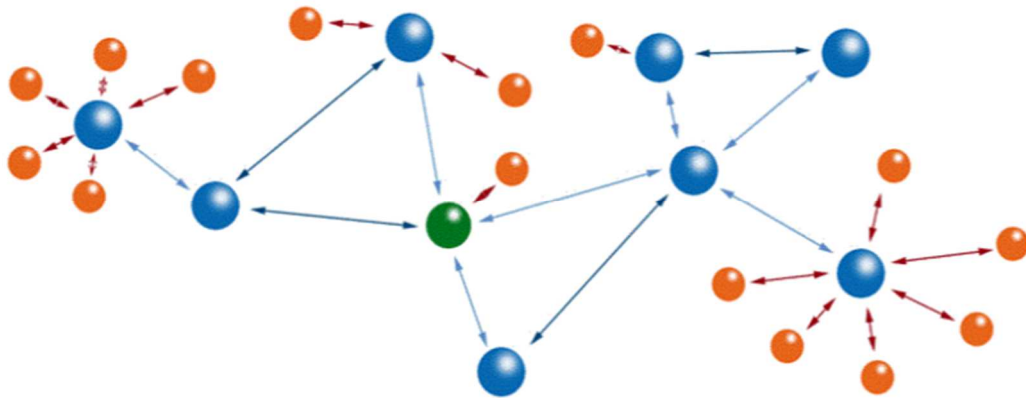


Figure I-7: Une architecture typique de réseau en toile [6]

b) Topologie en étoile

La topologie en étoile est une infrastructure mono-saut. Dans cette infrastructure, les nœuds terminaux communiquent directement avec le nœud coordinateur (station de base ou puits), autrement dit ces nœuds communiquent (envoient ou reçoivent) seulement avec la station de base.

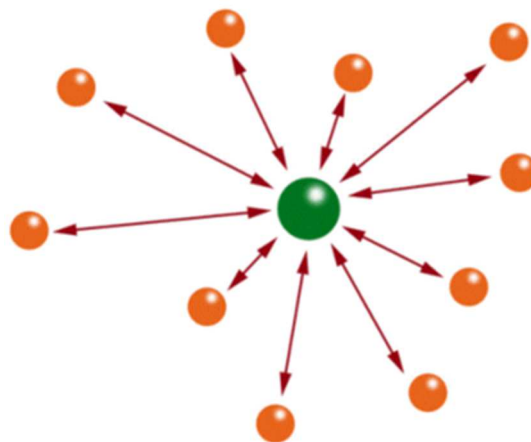


Figure I-8: Une architecture typique de réseau en étoile [6]

c) Topologie hybride

La topologie hybride est une combinaison des deux topologies (étoile et toile). Les stations de base forment une topologie en toile et les nœuds autour d'elles sont en topologie étoile [6].

I.4 Les réseaux LLN (Low power and Lossy Networks)

Les réseaux de faible puissance et à pertes (LLN) sont des réseaux dans lesquels les nœuds et leurs interconnexions sont fortement contraints par les ressources. Ils sont appelés aussi les réseaux à ressources limitées. Ces réseaux sont composés de plusieurs appareils embarqués avec des limitations d'énergie, de mémoire et de capacité de calcul avec des liaisons instables ayant des taux de perte élevés et des débits faibles.

Les réseaux LLN peuvent être composés de milliers de nœuds. Ils adoptent le modèle IPv6 et peuvent utiliser d'autres technologies de communication, tels que IEEE 802.15.4 ou Wi-Fi à faible consommation d'énergie. Dans la plupart des cas, les réseaux LLN sont utilisés au niveau de la couche de liaison avec trames dont les tailles sont réduites pour répondre à la contrainte mémoire dans les nœuds capteurs.

Dans les réseaux LLN, on distingue trois modèles de trafic comme c'est illustré par la Figure I-9:

- a) **Multi-point à point (MP2P)** : Dans ce modèle de trafic, la tâche principale des nœuds capteurs consiste à envoyer des relevés à partir d'appareils qui se trouvent à l'intérieur du réseau LLN vers un point de contrôle central (Racine). On parle alors de trafic remontant ou de MP2P. Ainsi, la passerelle (gateway) a un rôle important pour un protocole de routage puisqu'elle fournit un point de sortie pour chaque nœud qui veut envoyer des messages (paquets de données) vers l'extérieur.
- b) **Point à Multi-point (P2MP)** : Les nœuds capteurs peuvent également recevoir des paquets de données en unicast de la part de la passerelle c'est ce qu'on appelle le trafic descendant ou P2MP. Le trafic se fait à partir d'un point de contrôle central vers un sous-ensemble d'appareils du réseau.
- c) **Point à Point (P2P)** : Les réseaux LLN fournissent une structure de base pour ce trafic où ce type de trafic (P2P) doit être supporté, car dans certains scénarios, il est possible par exemple de faire une découverte de ressources locales ou réaliser une action entre les appareils appartenant au même réseau LLN.

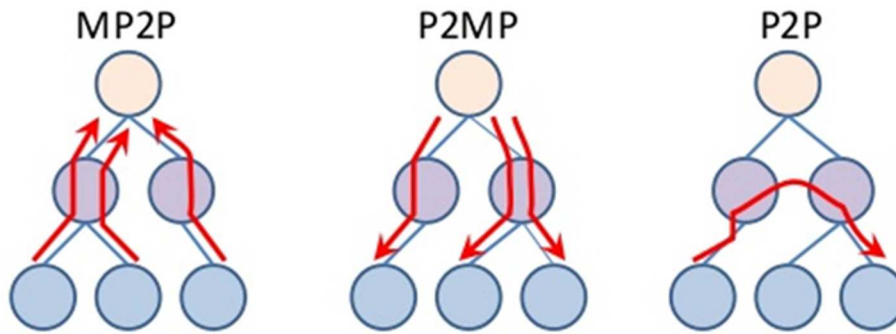


Figure I-9: Types de trafic dans les réseaux LLN

I.5 6LoWPAN

L'adoption de la pile protocolaire IPv6 dans les réseaux LLN a pris de l'unanimité par le groupe de travail créé par IETF à cause des milliards d'objets à connecter à Internet. Néanmoins, un autre problème s'est posé, celui de la taille des trames très petites définies par les protocoles de communications des couches sous-jacentes. Par exemple, si on prend la norme IEEE802.15.4, la taille des trames dans cette norme est seulement 127 octets alors que les paquets IPv6 ont une taille minimale de 1280 octets. 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 la raison pour laquelle le groupe de travail de l'IETF a défini la couche adaptative 6LoWPAN afin de résoudre cette incompatibilité entre les deux technologies de communication. 6LoWPAN est basé principalement sur deux mécanismes pour réduire la taille des paquets IPv6 (datagrammes) à savoir la fragmentation et la compression des entêtes. Ceci permettra aux paquets IPv6 d'être envoyés ou reçus à travers les réseaux LLN.

I.6 Conclusion

Dans ce premier chapitre, on a présenté le contexte de ce travail et plus particulièrement l'environnement IoT, les réseaux de capteurs sans fil, et les réseaux LLN. On a mis l'accent sur les spécificités des réseaux LLN qui permettent de connecter des objets bien particuliers. Ces objets ont de fortes contraintes en termes de ressources tels que les capteurs sans fil. Dans la plupart des cas, les réseaux LLN utilisent des médiums de communication à faible consommation à l'instar de la norme IEEE 802.15.4.

Dans le chapitre suivant, on va aborder le côté routage dans les réseaux LLN, en citant la classification des différents protocoles de routage selon leur fonctionnement, et les contraintes de routage dans les réseaux LLN.

Chapitre 2

Le routage dans les réseaux LLN

Chapitre II

Le routage dans les réseaux LLN

II.1 Introduction

Les caractéristiques et les contraintes présentées dans les réseaux LLN peuvent influencer l'opération de routage. Dans ce type de réseaux, les nœuds sont généralement limités en termes de puissance de traitement, d'énergie et de mémoire. Ainsi, ces contraintes ont un impact négatif sur le processus de routage. Elles peuvent causer des interconnexions instables avec des taux de perte de paquets élevés et des débits faibles. Par ailleurs, les protocoles de routage développés pour les réseaux ad hoc et les réseaux de capteurs sans fil ne peuvent pas s'adapter aux réseaux LLN. De ce fait, pour réussir le routage dans les réseaux LLN il faut concevoir des protocoles de routage qui prennent en compte les exigences des réseaux LLN d'une part et sans omettre les performances de routage d'autre part.

Dans ce chapitre, on revient sur la notion de routage pour mieux la définir, et on va également décrire les principaux protocoles de routage qui existent dans la littérature.

II.2 Routage dans les réseaux LLN

II.2.1 Définition du routage

Le routage est un processus permettant l'acheminement des paquets de données d'un nœud expéditeur vers un ou plusieurs nœuds destinataires, via des routeurs (nœuds relais) et en utilisant des protocoles de routage. Les routeurs (nœuds relais) représentent les nœuds qui relient deux ou plusieurs réseaux afin de commuter des paquets d'une interface vers une autre et les protocoles de routage déterminent la manière de fonctionnement des routeurs selon la propriété du protocole. Les protocoles de routage sont classés en plusieurs grandes classes selon différents critères. Les fonctions principales des protocoles de routage sont : l'établissement de chemins, la maintenance de ces chemins et l'acheminement des données à travers ces chemins.

II.2.2 Les types de communications

Les communications se distinguent par le nombre de destinataires à partir d'un nœud source et de la manière de livraison des messages (voir Figure II-1):

- **Unicast** : Dans ce type de communications, l'acheminement des messages est vers une seule destination bien déterminée.
- **Broadcast** : Le nœud source diffuse les données à tous les nœuds du réseau.
- **Multicast** : Le nœud source délivre le message à un ensemble des nœuds manifestant un intérêt pour un groupe.
- **Anycast** : Le nœud source délivre le message à n'importe quel nœud d'un groupe, mais généralement le plus proche, au sein du réseau.

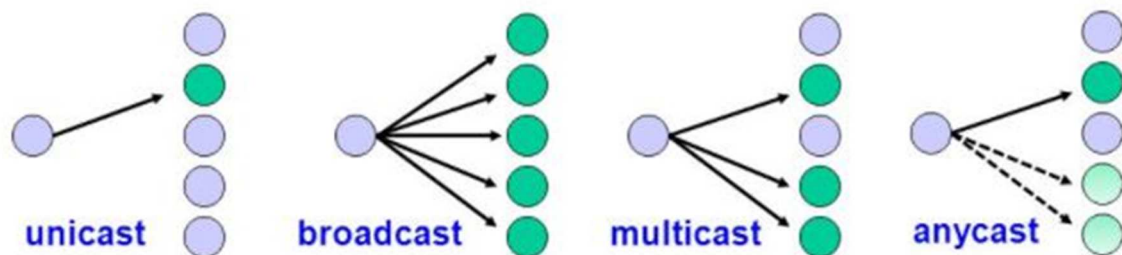


Figure II-1: Types de cardinalité de communication

II.2.3 Contraintes de routage dans les réseaux LLN

Dans les réseaux LLN, les nœuds sont généralement limités en termes de puissance de traitement, d'énergie et de mémoire. Ces contraintes peuvent complexer le fonctionnement de routage. L'objectif d'un protocole de routage dans les réseaux LLN est de minimiser la consommation d'énergie et l'utilisation le moins possible de la bande passante. Pour cela il faut concevoir un protocole de routage qui répond aux exigences des réseaux LLN afin de réussir le routage [7].

II.3 Classification des protocoles de routage

Les protocoles de routage dans les réseaux à capteurs sont classés selon deux types : protocoles de routage basés sur la manière d'établissement de routes et protocoles de routage basés sur la structure réseau comme le montre la Figure II-2.

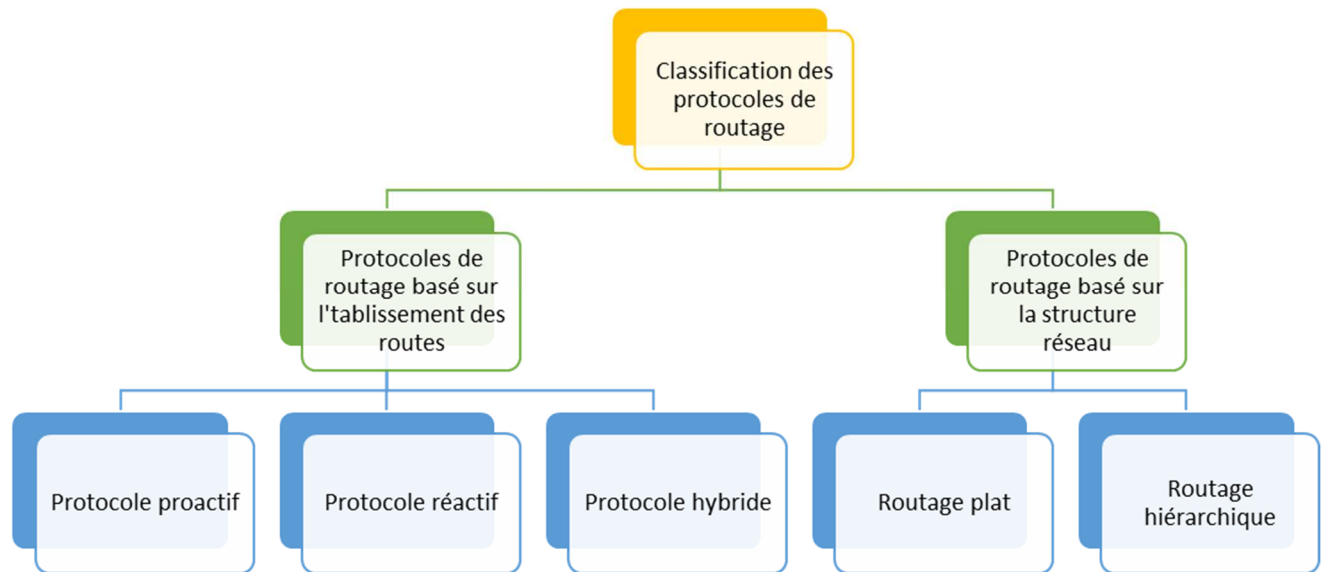


Figure II-2: Classification des protocoles de routage

II.3.1 Protocoles de routage basés sur établissement de routes

Les chemins de routage peuvent être établis selon l'une des trois façons suivantes : proactive, réactive et hybride où chacune d'elles a ses propres caractéristiques et contraintes.

a) Protocole proactif

Ce type de protocoles fait en sorte que les informations de routage soit disponible à tout moment. Il part du principe que chaque nœud doit posséder préalablement d'une route vers n'importe quelle destination dans le réseau. Pour assurer ce principe, il y a des échanges périodiques de paquets de synchronisation (ACK) dans le but de maintenir les tables de routage à jour. Parmi les protocoles proactifs les plus répandus dans la littérature on cite OLSR (Optimized Link State Routing) et DSDV (Dynamic Destination-Sequenced Distance Vector). La figure II-3 illustre le fonctionnement de ce type de protocoles.

Avantage :

- Disponibilité immédiate des routes à partir de n'importe quel nœud source vers n'importe quelle destination ; ce qui permet de réduire le délai.

Inconvénient :

- Overhead important car il y aura des routes établies à l'avance et qui ne seront éventuellement jamais utilisées.

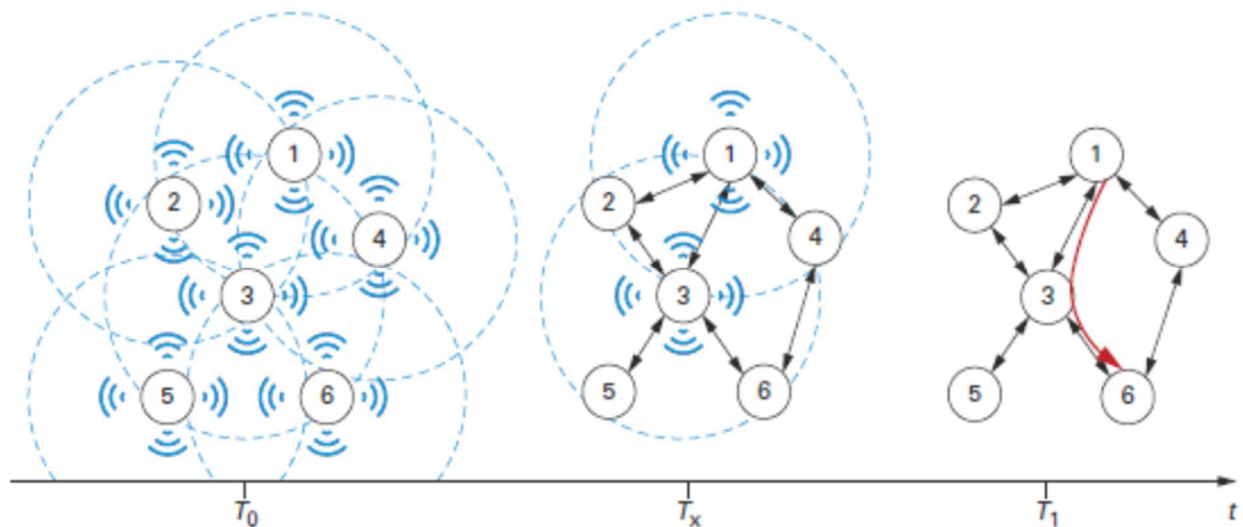


Figure II-3: Fonctionnement d'un protocole proactif [8]

b) Protocole réactif

Ce type de protocoles ne maintient pas les informations de routage au niveau des nœuds. Ces informations de routage ne sont disponibles que suite à la demande d'un nœud. Ainsi, un nœud souhaitant envoyer un paquet doit émettre une requête pour découvrir le chemin vers la destination souhaitée. Autrement dit quand un nœud souhaite communiquer avec un autre nœud, le processus d'établissement de chemins est initié par des échanges de messages de routage (ACK) par les nœuds du réseau (ex : SPIN, AODV) [8]. Ce type de protocoles réduit la surcharge du trafic (overhead) afin de minimiser les ressources consommées. La figure II-4 illustre le fonctionnement d'un protocole réactif.

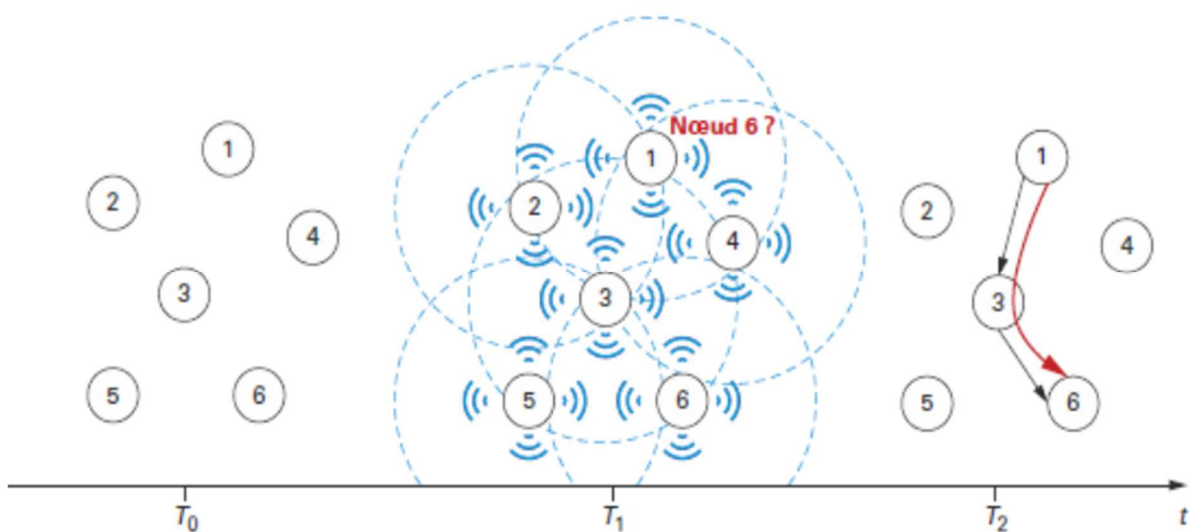


Figure II-4: Fonctionnement d'un protocole réactif [8]

c) Protocole hybride

Ces protocoles combinent les aspects de routage proactifs et réactifs. Ils utilisent un protocole proactif pour atteindre les voisins les plus proches, et pour découvrir des chemins ils utilisent un protocole réactif. Ainsi, avec ce découpage le réseau est partagé en plusieurs zones (ex : ZRP, APTEEN).

II.3.2 Protocoles de routage basé sur la structure réseau

Les protocoles sont répartis en fonction de la structure du réseau qui est très importante pour l'opération requise. On distingue principalement deux types de routage :

a) Le routage plat

Dans un protocole de routage plat, les nœuds sont généralement homogènes et jouent le même rôle c'est-à-dire un nœud peut être initiateur d'une opération de transmission comme il peut jouer le rôle d'un nœud relai. La conception de ce type de protocoles est simple mais elle présente quelques inconvénients tels que le manque de gestion des nœuds qui rend difficile l'optimisation des ressources du réseau (il s'agit généralement d'une gestion distribuée) et la lenteur de la réponse liées aux changements dynamiques du réseau.

b) Le routage hiérarchique

Les protocoles de routage hiérarchique permettent de regrouper les nœuds du réseau en plusieurs clusters (groupes) et chaque cluster est composé de plusieurs nœuds membres avec un seul clusterhead qui coordonne entre eux. Les clusterheads sont sélectionnés parmi les nœuds de chaque cluster en fonction de certains critères tels que l'énergie, le nombre de voisins, etc.... Les membres du cluster recueillent des informations et les transmettent à leur clusterhead correspondant. Les données sont ensuite traitées au niveau des clusterheads ce qui évite des données redondantes. Le clustering permet une utilisation efficace des ressources énergétiques et il est fortement recommandé pour les réseaux de grande dimension.

c) Le routage géographique

Dans ce type de routage, il est supposé que chaque nœud du réseau connaisse sa position et les positions de ses voisins. Le positionnement du nœud peut être obtenu en utilisant un système de géo-positionnement tel que le GPS² ou bien un des algorithmes permettant la détermination de la localisation des nœuds.

² GPS : Global Positioning System

II.4 Les critères de performance des protocoles de routage

La performance des réseaux LLN est fondée sur les facteurs suivants :

- **Evolutivité** : Vu que la zone des réseaux n'est pas toujours statique, elle change selon les besoins des utilisateurs, les nœuds dans les réseaux LLN doivent être en mesure de s'adapter aux changements de la structure du réseau.
- **L'énergie** : Un nœud dans les réseaux LLN doit savoir combien d'énergie sera dissipée pour réaliser une telle tâche à laquelle il est soumis. L'énergie consommée peut varier selon l'activité qu'il a à accomplir, par exemple une opération de transmission consomme plus d'énergie par rapport à une opération de réception.
- **Le temps de traitement** : il se réfère au temps pris par le nœud dans le réseau pour assurer l'ensemble des opérations à savoir la détection, le traitement des données ou le stockage de données, la transmission ou la réception.
- **Le schéma de transmission** : la transmission de données par les nœuds vers une destination ou la station de base se fait par un schéma de routage à un seul saut ou à multi-sauts.
- **Synchronisation** : dans les communications radio (sans fil) entre les nœuds dans un réseau LLN, les capteurs écoutent en permanence le canal de transmission partagé et consomment de l'énergie s'ils ne sont pas synchronisés les uns les autres. Pour cela, un nœud doit avoir la même durée (time-slot) pour se mettre en veille et se réveiller que ses voisins afin qu'il préserve son énergie.
- **Contrôle de paquets (ACK)**: Représente un paquet de synchronisation qui doit être envoyé avant la transmission entre deux nœuds. Il contient le nombre de bits de données envoyés, l'adresse du nœud de destination et certaines informations qui contribuent à éviter les collisions pendant la transmission.

II.5 Routage basé sur la qualité de service

Dans les protocoles de routage basé sur la qualité de service (QoS), le réseau doit équilibrer entre la consommation énergétique et la qualité de la transmission de données. En particulier, le réseau doit satisfaire certaines métriques de QoS, par exemple : délai, réserve d'énergie, la bande passante disponible, etc [9].

II.6 Conclusion

Dans ce chapitre, on a présenté le routage et ses contraintes dans les réseaux LLN avec ses différentes classes. On a mis l'accent sur la spécificité du routage dans les réseaux LLN et ses exigences.

Dans le chapitre qui suit, on va présenter le protocole de routage RPL spécialement conçu pour répondre aux contraintes des réseaux IPv6 à faible consommation et à perte et qui fait l'objet de notre PFE.

Chapitre III

Le protocole de routage

RPL

Chapitre III

Le protocole de routage RPL

III.1 Introduction

RPL est un protocole de routage élaboré par le groupe de travail ROLL³ qui est formé par l'IETF⁴. Il a été conçu afin de répondre aux exigences des réseaux LLN 6LoWPAN (IPv6 sur IEEE 802.15.4).

Dans ce chapitre, nous présentons le fonctionnement du protocole RPL en décrivant la démarche à suivre pour construire la topologie RPL. Puis, nous exposons ses concepts clés et nous présentons les deux contextes dans lesquels il pourra être exécuté : "mode avec stockage" et "mode sans stockage". Enfin, nous décrivons les paradigmes de la communication ainsi que les modes de fonctionnement du protocole RPL.

III.2 RPL (Routing Protocol for Low power and lossy networks-LLN)

Le protocole de routage RPL est un protocole proactif basé sur l'algorithme à vecteur de distance utilisant la pile protocolaire IPv6, conçu de manière à détecter les boucles de routage et les éviter. Son routage est basé sur des graphes acycliques orientés (DODAG : Destination Oriented Directed Acyclic Graph) vers la destination, afin d'acheminer les données vers la station de base.

RPL est un protocole flexible qui peut permettre en théorie l'interconnexion à grande échelle d'objets communicants d'une hétérogénéité élevée, caractérisés par de fortes contraintes en mémoire, puissance de calcul et ressources d'énergie [7].

Puisque le protocole RPL est un protocole de routage donc il a été intégré au niveau de la couche "Réseau du modèle OSI". Il existe également une couche 6LoWPAN qui prend place entre la couche liaison et la couche réseau. Cette couche intermédiaire détermine des mécanismes d'encapsulation et de compression des entêtes permettant ainsi aux paquets IPv6

³ ROLL : Représente un groupe de travail créé par l'IETF dans l'objectif d'élaborer un protocole de routage pour les réseaux LLN basé sur IPv6.

⁴ IETF (Internet Engineering Task Force) : Représente un organisme de standardisation ouvert, il élabore et promeut des standards Internet

d'être envoyés ou reçus via le protocole de communication IEEE 802.15.4. Cette couche est appelée couche d'adaptation 6LoWPAN. Elle permet de résoudre le problème d'incompatibilité entre les deux technologies de communication IEEE802.15.4 dont les trames ont une taille de 127 octets et IPv6 dont les trames ont une taille de 1280 octets.

III.3 Fonctionnement du protocole RPL

III.3.1 Graphes DAG & DODAG

DAG (Directed Acyclic Graph) est un graphe orienté où les connexions entre les arrêtes ont une direction et une propriété non circulaire i.e RPL ne possède pas de circuit.

RPL se base sur la notion de DODAG où un DODAG est un DAG a une seule destination vers la racine (Root) c'est-à-dire à une seule racine DAG (le puits des données) comme c'est illustré par la Figure III-1.

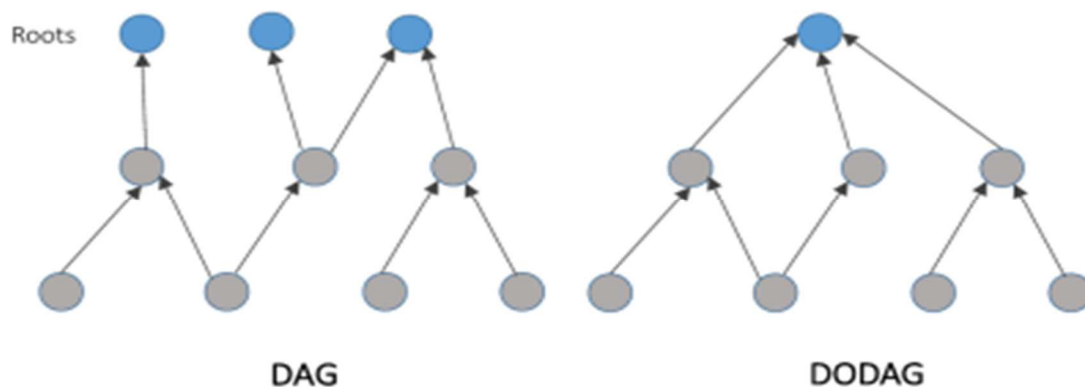


Figure III-1: Exemple de graphes (DAG et DODAG)

III.3.2 Modèle réseau de RPL

RPL définit trois types de nœuds :

- DODAG root ou LBR (Low power and lossy Border Router) : c'est la racine d'un DODAG qui représente un point de collecte dans le réseau.
- Nœud Routeur : c'est un périphérique qui peut transmettre et relayer le trafic des voisins.
- Nœud host : c'est un périphérique terminal capable de produire des données, mais qui n'est pas capable de relayer un trafic [10].

III.3.3 Messages de contrôle du protocole RPL

Les messages de contrôle dans RPL sont considérés comme de nouveaux messages ICMPv6. Ils sont composés d'un en-tête ICMPv6 suivi d'un corps du message identifié dans la

base avec éventuellement un certain nombre d'options. Il existe principalement quatre types de messages de contrôle dans le protocole RPL :

- *DIO (DODAG Information Object)* : Ce message contient des informations qui permettent à un nœud de découvrir une instance RPL, communiquer ses paramètres de configuration, sélectionner un parent DODAG, définir et maintenir le DODAG. Ce message est envoyé périodiquement pour maintenir le DODAG.
- *DIS (DODAG Information Solicitation)* : Un message DIS est utilisé par un nœud RPL pour solliciter des messages DIO de leurs voisins. Ce type de messages ressemble à la sollicitation de routeur dans IPv6 pour découvrir des voisins. D'une manière analogue, un nœud RPL utilise les messages DIS pour sonder ses voisins les plus proches situés dans un DODAG (les voisins directs). Ainsi, lorsqu'un nœud rejoint le réseau ou perd le lien avec son parent, il se met en mode silence en attendant le prochain message DIO, ou demande l'envoi d'un message DIO par l'intermédiaire d'un message DIS. Si après une période de temps bien précise, un nœud ne reçoit pas de message DIO en réponse à son message DIS envoyé, il pourra décider de devenir racine flottante (Root) d'un DODAG et commencera à envoyer des messages DIO.
- *DAO (DODAG Advertisement Object)* : Les messages DAOs sont des messages montants. Ils sont utilisés pour propager les informations sur la destination vers le haut, tout au long du DODAG. Le message DAO est unicast du fils au parent.
- *DAO-ACK* : Le message DAO-ACK est un message d'acquittement du message DAO envoyé par une racine ou un parent vers un fils.

III.3.4 Fonction d'Objectif (OF)

Le protocole RPL assure la qualité de service (QoS) dans la couche réseau à partir d'une fonction objectif qui permet d'optimiser la topologie en fonction d'une métrique ou une combinaison de plusieurs métriques telles que la préservation de l'énergie, le chemin le plus court ou la qualité des liens (ETX).

La fonction objectif définit comment les nœuds RPL sélectionnent d'une manière optimale les routes à l'intérieur d'une instance RPL. Elle est identifiée par le paramètre OCP (Objective Code Point) au sein de l'option de configuration DIO. Elle permet également de définir comment les nœuds doivent traduire une métrique ou une combinaison de plusieurs métriques en une valeur appelée Rank (rang), qui explicite la distance séparant le nœud à la racine DODAG. Elle définit aussi comment sélectionner les nœuds "parents". L'information RPL

incluse dans les paquets de données comprend le rang du nœud émetteur et permet de détecter les anomalies et les boucles de routage de telle sorte qu'un parent ait un rang supérieur à celui de son fils.

Plusieurs variantes et améliorations ont été proposées afin d'adapter RPL aux différentes applications de l'IoT et actuellement deux fonctions objectifs (OF) sont définies : "Objective Function Zero" (OF0 définie dans RFC6552) et "Minimum Rank With Hysteresis Objective Function" (MRHOF définie dans RFC6719). La fonction d'objectif OF0 permet à un nœud de sélectionner le DODAG qui fournit une bonne connectivité à un ensemble spécifique de nœuds (DODAG à joindre) tout en minimisant le nombre de sauts entre le nœud concerné et la racine. MRHOF est la fonction objectif par défaut à utiliser dans la formation de DAG dans RPL Cette fonction permet à un nœud de déterminer le taux de transmission avec succès (ETX) ainsi que l'énergie restante des nœuds.

III.3.5 Procédure de construction du DODAG

La construction d'un DODAG est basée sur des métriques de routage et des contraintes. Avant de présenter les différentes étapes pour construire un DODAG, nous donnons quelques notions sur les éléments qui sont des éléments d'un DODAG :

- Chaque nœud contient les informations suivantes :
 - Un ID du nœud : c'est un identifiant unique qui caractérise chaque nœud dans le réseau,
 - Parent par défaut,
 - Rank (Rang) : Représente le rang d'un nœud et explicite sa position au niveau du DODAG. Le rang est calculé en fonction du rang de son parent à l'aide d'une fonction OCP,
 - Liste des prédécesseurs,
 - Liste des destinataires.
- RPLInstanceID : c'est un identifiant unique sur le réseau. Les DODAGs qui ont le même RPLInstanceID, partagent la même Fonction Objectif.
- InstanceRPL : Une instance RPL est un ensemble d'un ou plusieurs DODAGs qui partagent un RPLInstanceID.

La construction du DODAG est réalisée selon les étapes suivantes :

- 1) Au début, la topologie DODAG est initiée par le nœud racine LBR. Ce nœud diffuse des messages DIO autour de son voisinage à un saut (les voisins directs) où un message DIO inclut le rang du nœud, son ID, Instance RPL et la fonction objectif.
- 2) Les nœuds qui ont reçu le message DIO envoyé par le nœud LBR, analysent ce dernier, puis décident s'ils rejoignent la structure ou non. Cette décision est fondée sur certains paramètres (la fonction objectif, les caractéristiques du DAG et le coût du chemin annoncé). Quand le nœud rejoint la structure, il a un chemin vers la racine de la structure DODAG.
- 3) Les nœuds qui ont choisi le nœud LBR comme parent vont à leur tour diffuser des messages DIO en multicast dans leur voisinage. Les nœuds qui ne font pas encore partie du DODAG et qui vont recevoir ces messages DIO, vont donc choisir des nœuds parents puis répondent par des messages DAO pour montrer leur attachement au DODAG [11].

Ce processus continue jusqu'à ce que tous les nœuds du réseau soient couverts.

Ensuite, s'il s'agit d'un routeur, ce dernier 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 les boucles de routage.

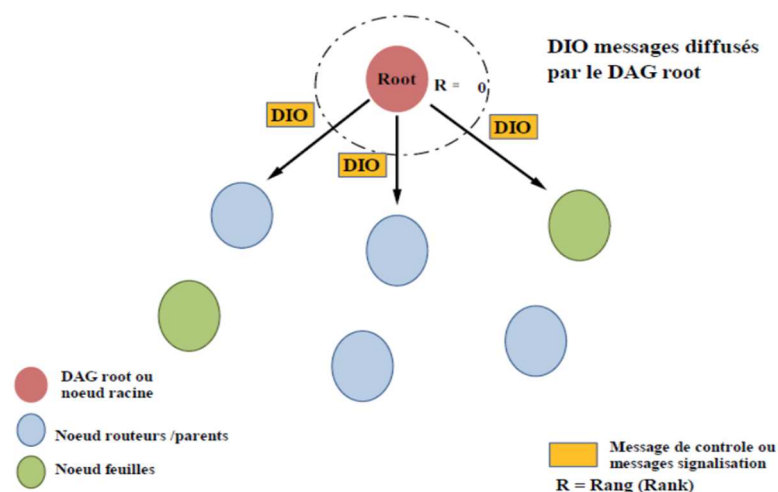


Figure III-2: Diffusion des DIO par la racine [11]

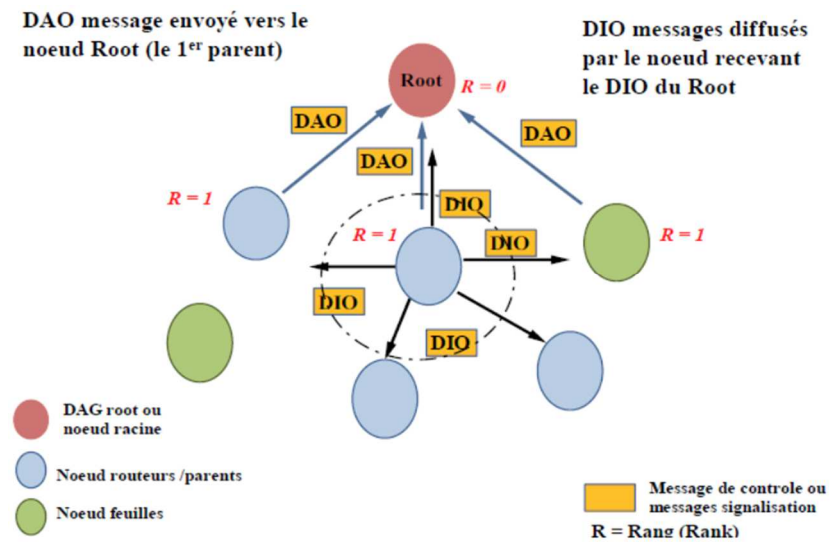


Figure III-3: Choix du root (racine) comme nœud parent [11]

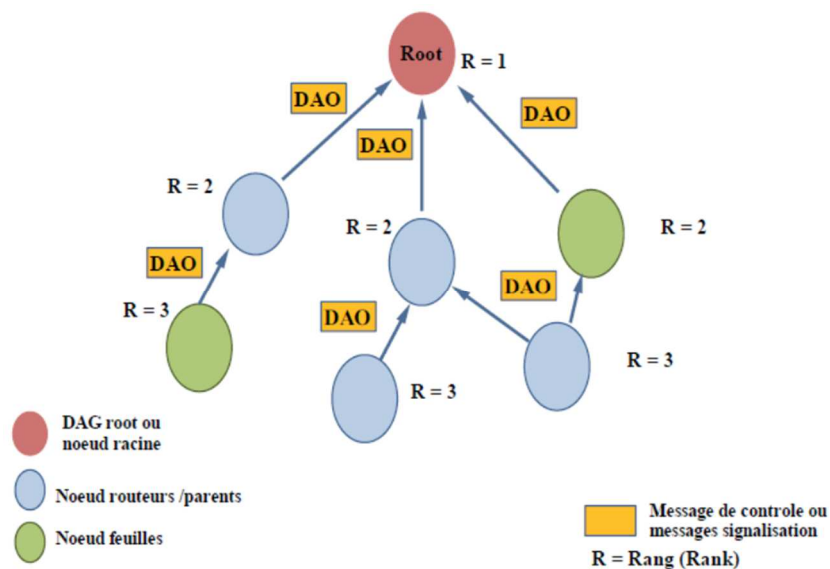


Figure III-4: Poursuite de la construction du DODAG [11]

III.3.6 Modes d'opération du protocole RPL

Il existe deux modes de fonctionnement dans le protocole RPL à savoir "Storing Mode (mode stockage)" et "Non-Storing mode (mode non stockage)". La Figure III-5 illustre les deux modes de fonctionnement dans le protocole RPL.

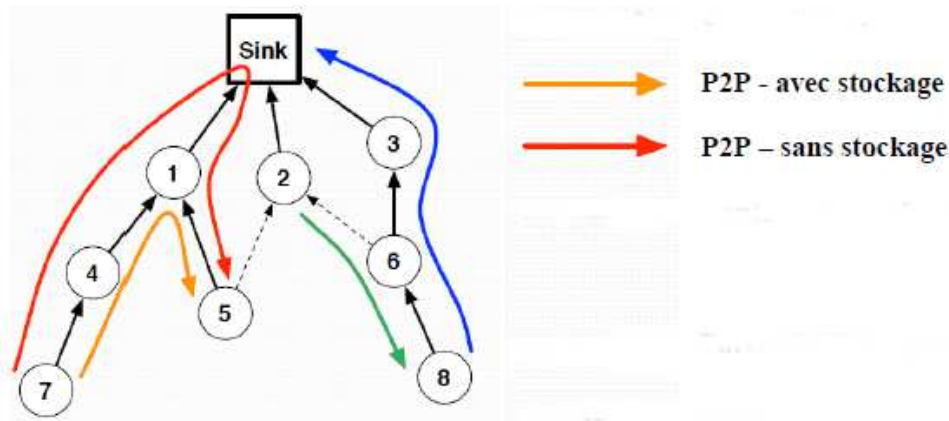


Figure III-5: Modes de fonctionnement du protocole RPL

a) Le fonctionnement en "Storing mode"

Dans ce mode, un nœud appartenant à la structure DODAG envoie un message DAO à l'ensemble de ses parents et à la réception de ce message DAO, le nœud parent ajoute une entrée dans la table de routage. Puis, ce dernier envoie à son tour un message DAO à l'ensemble de ses parents. Ce processus se poursuit jusqu'à ce que l'information atteigne la racine du DODAG. Ce mode de fonctionnement est appelé "mode de fonctionnement avec stockage" où tous les nœuds intermédiaires disposent d'une mémoire pour stocker les tables de routage.

b) Le fonctionnement en "Non-Storing mode"

Dans ce mode, seulement la racine qui a la possibilité de stocker des informations de routage et les autres nœuds du réseau conservent uniquement les adresses de leurs parents directs. Lorsqu'on aura besoin de router des données vers une destination, les nœuds transmettent ces données à la racine en passant par leur parent. Ainsi, la racine va effectuer un routage de la source vers la destination concernée.

III.3.7 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. Parmi ces métriques on a :

a) Nombre de sauts (Hop Count)

Hop Count (HC) représente le nombre de sauts entre un nœud et sa racine. Cette métrique permet à la fonction objectif de se baser sur elle-même afin de sélectionner le meilleur parent.

b) Nombre de transmission prévu (ETX)

ETX représente le nombre de retransmissions nécessaires pour qu'un paquet soit reçu avec succès vers la destination. Ce nombre varie de un à l'infini. ETX indique si le support de transmission est parfait ou non (il explicite la qualité du lien).

III.3.8 Trafic supporté par un DODAG

RPL prend en charge trois paradigmes de communication : MP2P, P2MP et P2P.

- Le paradigme de communication MP2P a été fourni pour la construction des routes à partir de chaque nœud vers la racine du DODAG. Lorsqu'un nœud veut transmettre une donnée vers le nœud LBR, cette donnée sera envoyée vers un de ses parents, puis elle remontera la structure jusqu'à ce qu'elle atteigne la destination finale.
- Le paradigme de communication P2MP écoule les informations vers les nœuds feuilles. On parle alors des routes descendantes (du haut en bas). Tout cela nécessite une table de routage qui doit être établie au niveau de chaque nœud et un mécanisme pour remplir ces tables.
- Le paradigme de communication P2P concrétise un trafic à partir de n'importe quel nœud vers n'importe quel autre nœud dans le DODAG. Quand un nœud envoie un message vers un autre nœud dans la structure DODAG, ce message sera envoyé en direction de la racine du DODAG jusqu'à ce qu'il atteigne un nœud ancêtre commun, ayant une connaissance sur la route, au niveau duquel le message sera transmis en direction de la destination finale.

III.4 Conclusion

Dans ce chapitre, on a présenté le protocole RPL qui a été conçu pour être très flexible aux différentes variations des ressources du réseau. La version actuelle du protocole RPL ne peut pas répondre à tous les besoins spécifiques des applications, ainsi elle doit être améliorée afin d'atteindre certains objectifs particuliers. Plusieurs travaux se sont focalisés sur l'optimisation du protocole RPL.

Dans le chapitre qui suit, on va décrire notre problématique ainsi que notre contribution qui consiste à optimiser le protocole RPL.

Chapitre IV

Amélioration du protocole RPL

Chapitre IV

Améliorations du protocole RPL

IV.1 Introduction

Les nœuds dans les réseaux LLN se caractérisent par des tables de routage à capacité limitée. A cet effet, lors de l'ajout d'un nœud dans une topologie réseau déjà existante, cela pourra avoir un impact sur la qualité de service si le nœud dans le réseau qui reçoit le message DIS à une table de routage saturée.

Le nouveau nœud ajouté dans le réseau va envoyer un message DIS pour rejoindre le réseau (un DODAG) et durant cette étape le nœud qui va recevoir ce message DIS, pourra y avoir une table de routage saturée. Donc, le fait d'envoyer un message DIS à un nœud dont la table de routage est saturée, a un impact sur la QoS. Dans ce contexte, on propose une solution qui pourra garantir une QoS respectable lors de l'ajout d'un nouveau dans un réseau de type LLN.

Dans ce chapitre, on va premièrement présenter les plateformes logicielles utilisées ainsi que la configuration matérielle. Ensuite, on présente la solution proposée pour optimiser la QoS dans les réseaux LLN et on évalue cette solution à travers différents scénarios. Enfin, on présente les résultats obtenus et on les interprète tout en explicitant les atouts de la solution proposée dans le cadre de cette problématique.

IV.2 Outils de simulation

Les simulations ont été réalisées dans l'environnement logiciel suivant :

- Système d'exploitation installé dans l'ordinateur : Microsoft Windows 8,
- Lecteur des machines virtuelles : VMPlayer,
- Système d'exploitation installé sous la machine virtuelle : Ubuntu 14.04 LTS,
- Système d'exploitation pour les capteurs : Contiki "Contiki 3.0",
- Simulateur pour les réseaux de capteurs sans fil : Cooja.

Les simulations ont été réalisées sur un ordinateur Acer dont la configuration est la suivante:

Tableau IV-1 : Caractéristiques de la machine utilisée pour les simulations

Champ	Valeur
Processeur	Intel(R) Core(TM) i3 CPU M380 @2.53GHz 2.53GHz
RAM	4.00 Go
Disque dur	200 Go

IV.2.1 Contiki

a) Présentation de Contiki

Contiki [12] est un système d'exploitation open-source qui fonctionne sur les microcontrôleurs de faible puissance. Il permet de développer des applications qui exploitent d'une manière efficace les ressources matérielles tout en fournissant une communication standardisée sans fil de faible puissance pour une gamme de plateformes des microcontrôleurs tels que le TI MSP430 et Atmel AVR, qui sont utilisés dans les familles des capteurs Telos, Tmote et Mica.

Le principal avantage de Contiki est qu'il fonctionne sur un concept qui se situe entre le multi-threading et la programmation événementielle. Cela permet aux processus de partager le même contexte d'exécution et donc d'améliorer l'utilisation de la mémoire et de l'énergie.

Contiki contient deux piles de communications : uIP et Rime13.

- Rime est une pile de communication légère conçue pour des radios basse puissance.
- uIP est une petite pile de TCP/IP qui permet à Contiki de communiquer sur Internet.

b) Architecture de Contiki

Contiki est écrit en langage C et il est composé d'un noyau, de bibliothèques, et d'un ordonnanceur, Contiki permet de gérer les ressources matérielles telles que le processeur, la mémoire et les périphériques (d'entrées/sorties) (voir la Figure IV-1). Dans ce système, le noyau gère des événements asynchrones et synchrones où les premiers sont placés dans une file d'attente après leur appel. Les traitements sous-jacents à ces événements sont déclenchés après une certaine période.

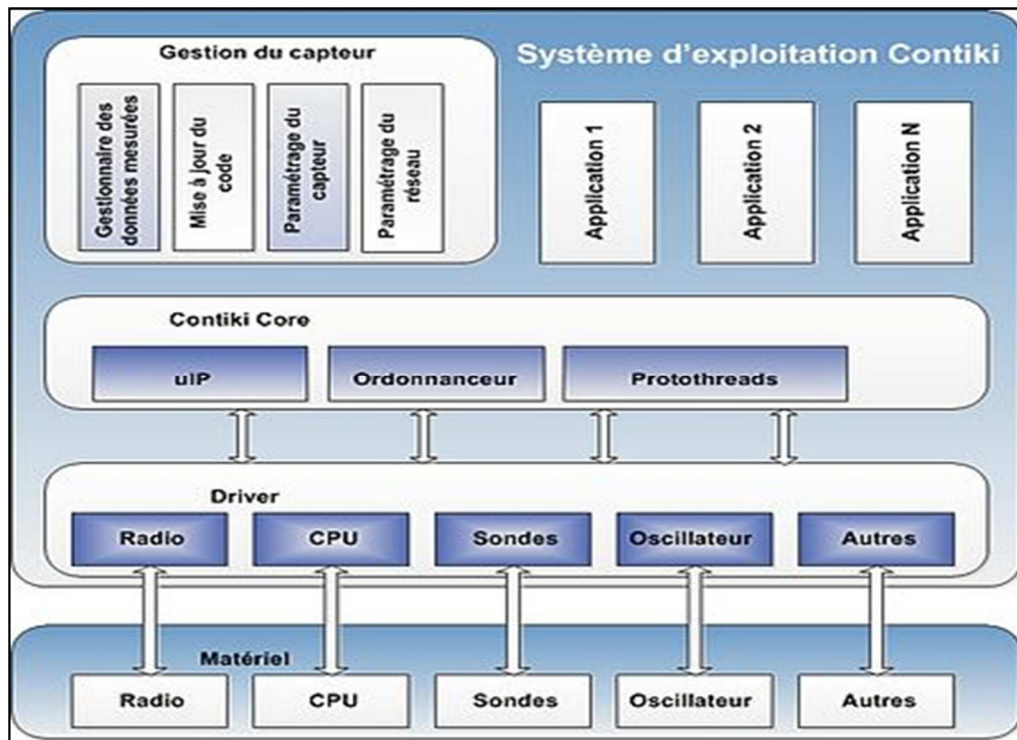


Figure IV-1: Architecture de Contiki

c) Mécanismes de communication

Contiki dispose deux schémas de communication :

- La couche de protocole Rime : Elle permet de dialoguer avec les capteurs voisins ainsi que le routage.
- La couche uIP : C'est une couche orientée Internet. Elle fournit les services essentiels du protocole IP mais exige plus de ressources que la couche Rime [9].

IV.2.2 Simulateur Cooja

a) Présentation du simulateur

COOJA [13] est l'acronyme de Contiki OS Java Simulator. C'est un simulateur réseau pour réaliser des programmes au sein de Contiki. Il permet d'émuler des nœuds capteurs et de charger un programme compilé. Il permet également de simuler les connexions réseaux et d'interagir avec les capteurs. Les données collectées provenant du nœud puits "Sink" via sa sortie standard peuvent être stockées dans des fichiers ou lus par d'autres logiciels qui peuvent par la suite traiter et présenter les données d'une manière ergonomique à l'utilisateur.

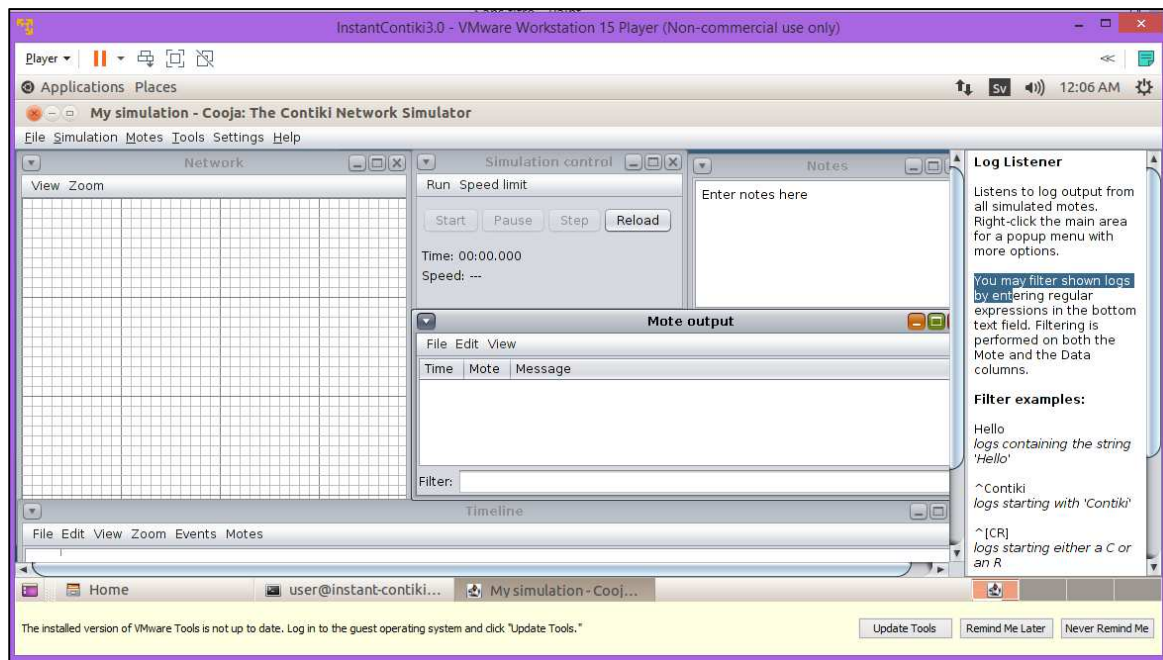


Figure IV-2: Interface du simulateur Cooja

b) Fenêtres de simulation

Dans une simulation on a plusieurs fenêtres qui sont affichées où chacune d'elles a un rôle bien particulier.

- **La fenêtre "Timeline"** : cette fenêtre se situe en bas de l'écran. Elle permet d'afficher tous les événements de communication dans la simulation au cours du temps. Elle est très pratique pour comprendre ce qui se passe dans le réseau.
- **La fenêtre "Network"** : en haut à gauche de l'écran. Elle nous montre tous les nœuds dans le réseau simulé.
- **La fenêtre "Mote Output"** : sur le côté droit de l'écran, nous montre toutes les impressions port série de tous les nœuds.
- **La fenêtre "Notes"** : en haut à droite est l'endroit où nous pouvons mettre des notes pour la simulation en cours d'exécution.
- **La fenêtre "Simulation control"** : c'est une fenêtre qui permet de lancer, mettre en pause et charger une simulation qui existe déjà.

IV.3 Travail réalisé

IV.3.1 Problématique

En mode avec stockage du protocole RPL, les nœuds peuvent épuiser facilement leur mémoire en raison d'une limitation de stockage des voisins, ainsi des tables de routage (saturation). Ce qui représente un sérieux problème surtout lorsque le réseau est dense en plus de l'existence de nœuds évolutifs. En outre, le mode "avec stockage" nécessite que chaque nœud stocke les informations de routage concernant son sous-DODAG. En général, les nœuds situés autour de la racine sont plus susceptibles de manquer de mémoire, en particulier dans les grands réseaux, qui ont besoin de beaucoup de mémoire. Les nœuds autour de la racine peuvent ne pas se permettre de telles ressources de mémoire.

En outre, le protocole RPL ne définit aucune action à entreprendre dans le cas où un parent refuserait d'installer une nouvelle route descendante (en raison de la table de routage surchargée), compromettant la fiabilité du trafic descendant. Ces problèmes peuvent avoir un impact négatif sur l'évolutivité du réseau, par conséquent, tout routeur du réseau ne parvient pas à ajouter une entrée de routage pour une cible spécifique en raison de ses limites de stockage. La figure IV-3 illustre comment un nœud rejoint un DODAG.

Un autre problème qui entravera le processus de RPL est la quantité de surcharge échangée dans le réseau. La surcharge dans RPL est la somme de tous les messages de contrôle : DIS, DIO, DAO et DAO-ACK.

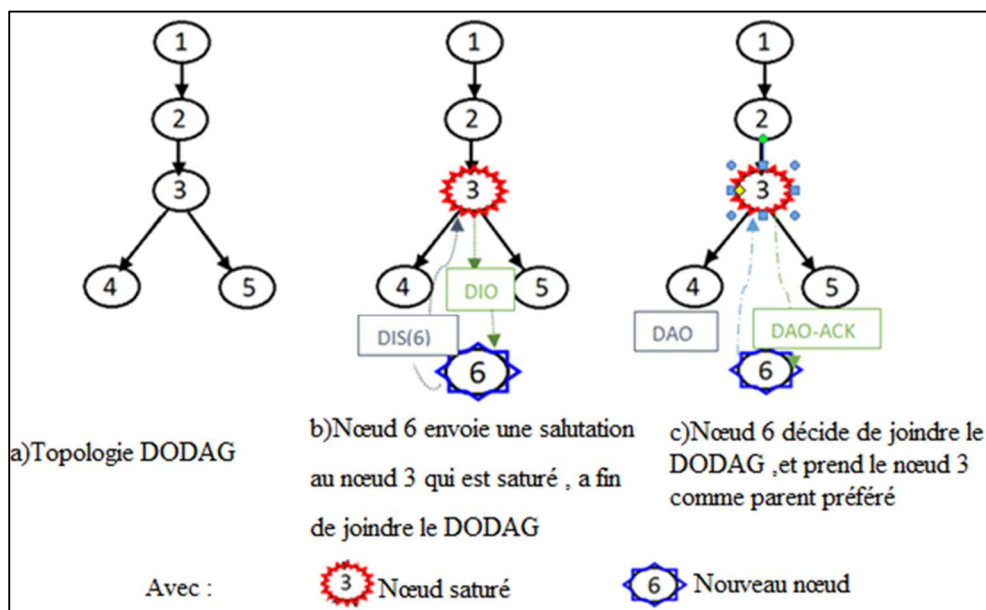


Figure IV-3: Nouveau nœud qui veut rejoindre le DODAG

IV.3.2 Contribution

Afin de traiter la problématique posée précédemment, notre idée est de prendre en compte l'évolutivité du réseau en cas de saturation mémoire d'un nœud tout en minimisant le surcoût. Nous économisons les messages de contrôle sachant que cela réduira la quantité d'énergie, ce qui sera plus bénéfique pour un réseau contraint tel que l'IoT. Notre objectif est d'éviter les messages inutiles en cas de saturation, comme l'illustre la figure IV-4.

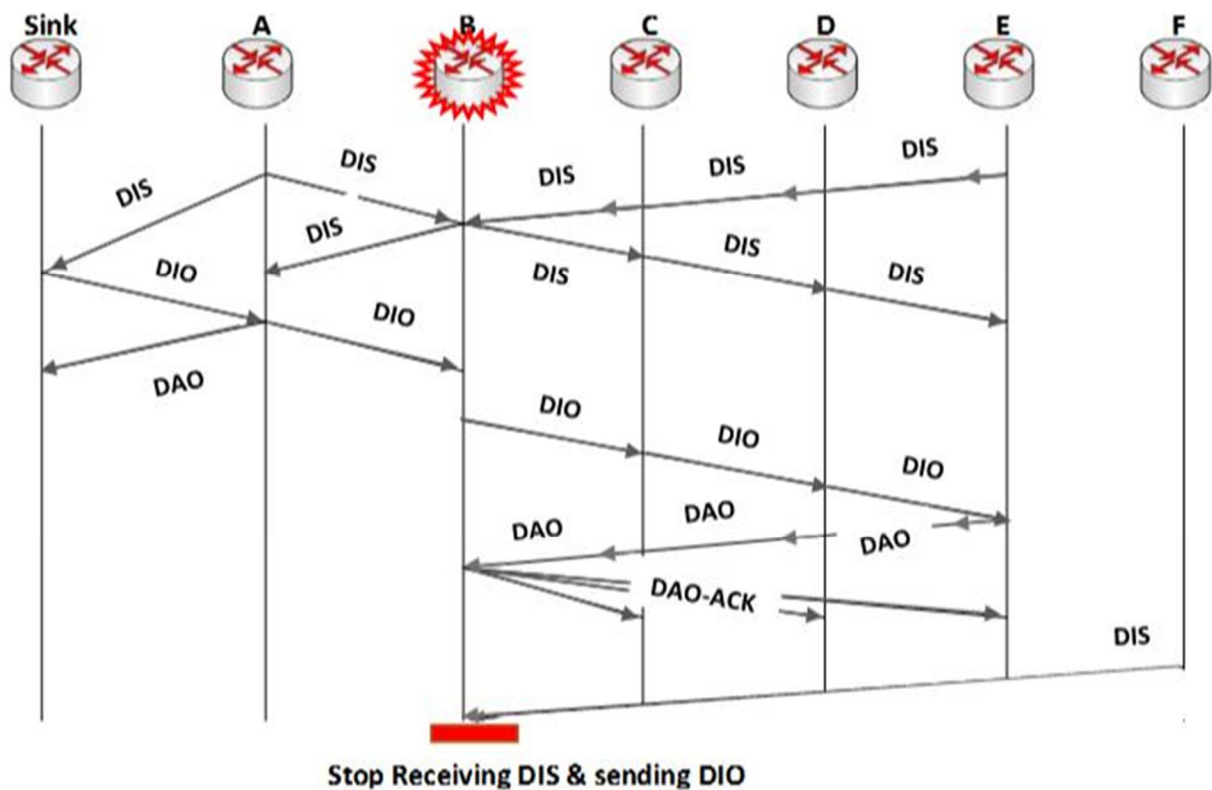


Figure IV-4: Stopper la réception des DIS et l'envoi des DIO

Une fois que le nœud parent reçoit un message de contrôle DIS, il vérifie la mémoire de sa table voisine. S'il y a beaucoup d'espace pour laisser le nœud se joindre, le processus du RPL continuera sinon, en cas de surcharge de la mémoire, le schéma proposé n'échangera aucun message de contrôle qui se répercutera positivement sur les performances du réseau IoT. L'organigramme présenté dans la figure IV-5 résume la démarche prise dans notre contribution.

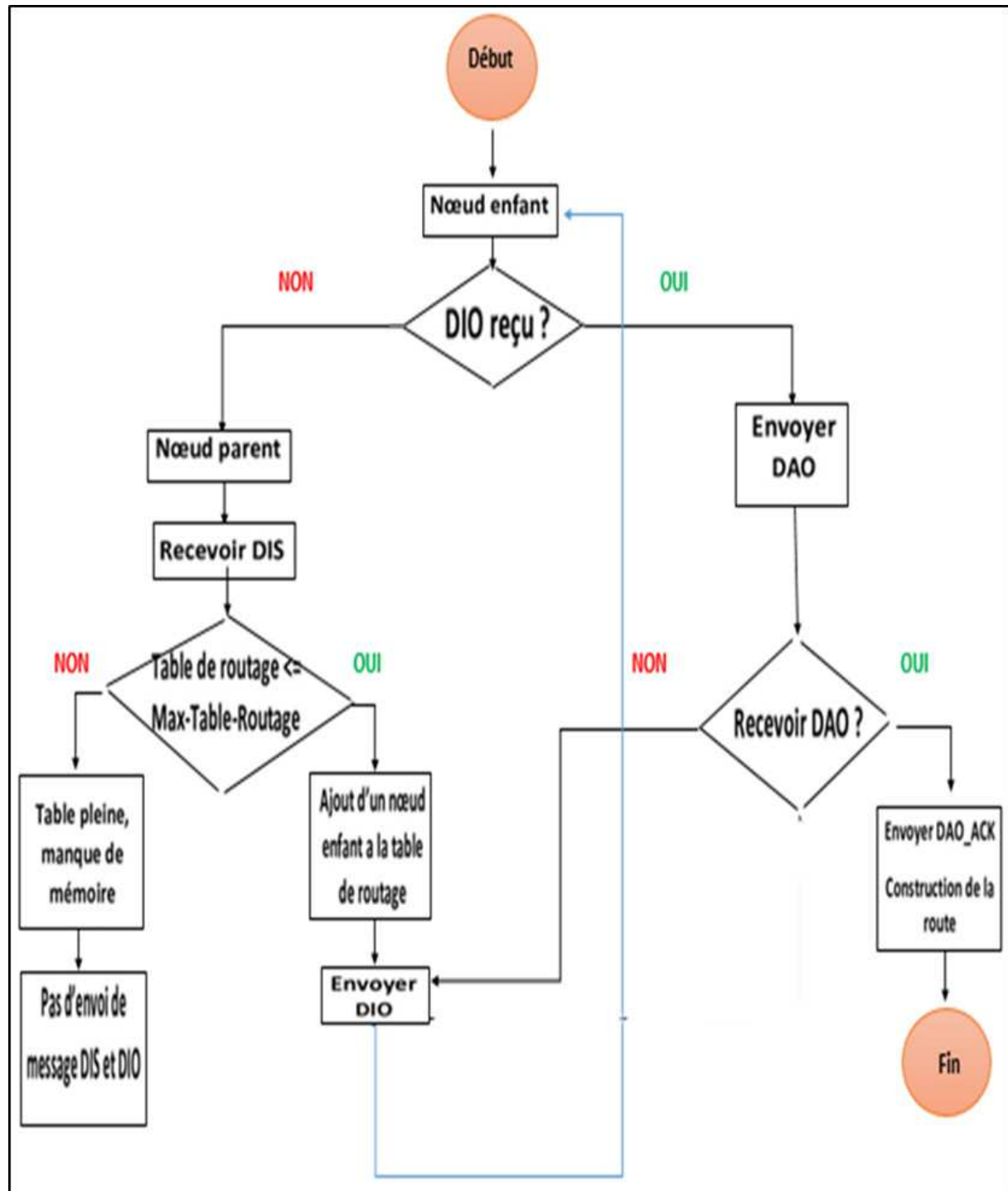


Figure IV-5: Organigramme de la solution proposée

Les modifications faites étaient au niveau du fichier **rpl_icmp6.c** et plus exactement au niveau des deux fonctions : **rpl_icmp6_update_nbr_table** et **dis_input**.

La fonction **rpl_icmp6_update_nbr_table** (Figure IV-6) permet la mise à jour des tables, donc à son niveau on a traité le cas où la table de routage est saturée.

La fonction **dis_input** (Figure IV-7) est chargée de la réception des messages de contrôle DIS, donc à son niveau on a fait appel à la fonction de mise à jour des tables accompagné de quelques traitements.

```

/*-----*/
vip_ds6_nbr_t *
rpl_icmp6_update_nbr_table(vip_ipaddr_t *from, nbr_table_reason_t reason, void *data)
{
    vip_ds6_nbr_t *nbr;

    if((nbr = vip_ds6_nbr_lookup(from)) == NULL) {
        if((nbr = vip_ds6_nbr_add(from, (vip_lladdr_t *)
                                packetbuf_addr(PACKETBUF_ADDR_SENDER),
                                0, NBR_REACHABLE, reason, data)) <= NBR_TABLE_CONF_MAX_NEIGHBORS /*!= NULL*/) {
// table de routage <= MAX_table deroutage"
            printf("RPL: Neighbor added to neighbor cache ");
            vip_debug_ipaddr_print(from);
            printf(", ");
            net_debug_lladdr_print((vip_lladdr_t *)packetbuf_addr(PACKETBUF_ADDR_SENDER));
            printf("\n");
            /***
            return nbr;
        }
        /***
        else {
            return NULL;
        }
    }
}

```

Figure IV-6: Fonction de mise à jour des tables

```

static void
dis_input(void)
{
    rpl_instance_t *instance;
    rpl_instance_t *end;

    /* DAG Information Solicitation */
    printf("DIS from ");
    uip_debug_ipaddr_print(&UIP_IP_BUF->srcipaddr);
    //net_debug_lladdr_print((uip_lladdr_t *)packetbuf_addr(PACKETBUF_ADDR_SENDER))
    printf("\n");

    for(instance = &instance_table[0], end = instance + RPL_MAX_INSTANCES;
        instance < end; ++instance) {
        printf("instance->used == %d \n", instance->used);
        if(instance->used == 1) {
            if(!uip_is_addr_mcast(&UIP_IP_BUF->destipaddr)) { //
#ifdef RPL_LEAF_ONLY
                PRINTF("RPL: LEAF ONLY Multicast DIS will NOT reset DIO timer\n");
#else /* !RPL_LEAF_ONLY */
                printf("RPL: Multicast DIS => reset DIO timer\n");
                rpl_reset_dio_timer(instance);
#endif /* !RPL_LEAF_ONLY */
            } else {
                /* Check if this neighbor should be added according to the policy. */
                if(rpl_icmp6_update_nbr_table(&UIP_IP_BUF->srcipaddr,
                                                NBR_TABLE_REASON_RPL_DIS, NULL) == NULL) {
                    printf("RPL: Out of Memory, not sending unicast DIO, DIS from ");
                    uip_debug_ipaddr_print(&UIP_IP_BUF->srcipaddr);
                    printf(", ");
                    PRINTLLADDR((uip_lladdr_t *)packetbuf_addr(PACKETBUF_ADDR_SENDER));
                    printf("\n");
                    // return UIP_ICMP6_INPUT_ERROR; //
                } else {
                    int test = rpl_icmp6_update_nbr_table(&UIP_IP_BUF->srcipaddr,
                                                            NBR_TABLE_REASON_RPL_DIS, NULL);

                    printf("test=%d \n", test);
                    printf("RPL: Unicast DIS, reply to sender\n");
                    dio_output(instance, &UIP_IP_BUF->srcipaddr);
                }
            }
        }
    }
}
/* } */

```

Figure IV-7: Fonction dis_input

IV.4 Résultats et discussion

Durant la phase de simulation, on a réalisé plusieurs topologies réseaux de capteurs de différentes tailles dans lesquelles les capteurs sont distribués aléatoirement dans la zone réseau. En outre, dans les simulations, on a utilisé l'exemple de rpl-udp (contiki/examples/ipv6/rpl-udp) et un environnement de simulation comme montre le tableau ci-dessous :

Tableau IV-2: Paramètres de simulation

Paramètres	Valeurs
Simulateur	Cooja
Nombre de nœuds	5, 10, 15, 20
Nombre de nœuds racines	1
Durée de la simulation (s)	50, 100, 150, 200, 250, 300, 350, 400, 450, 500
Identité du nœud racine	1
Surface (mètres)	300 X 300
Fonction Objectif (OF)	MRHOF, OF0
Mote types	Z1 mote
Topologie	Grille ordonnée, aléatoire

Ci-dessous (Figure IV-8) un exemple d'une des simulations faite, afin de voir l'échange de messages entre les nœuds.

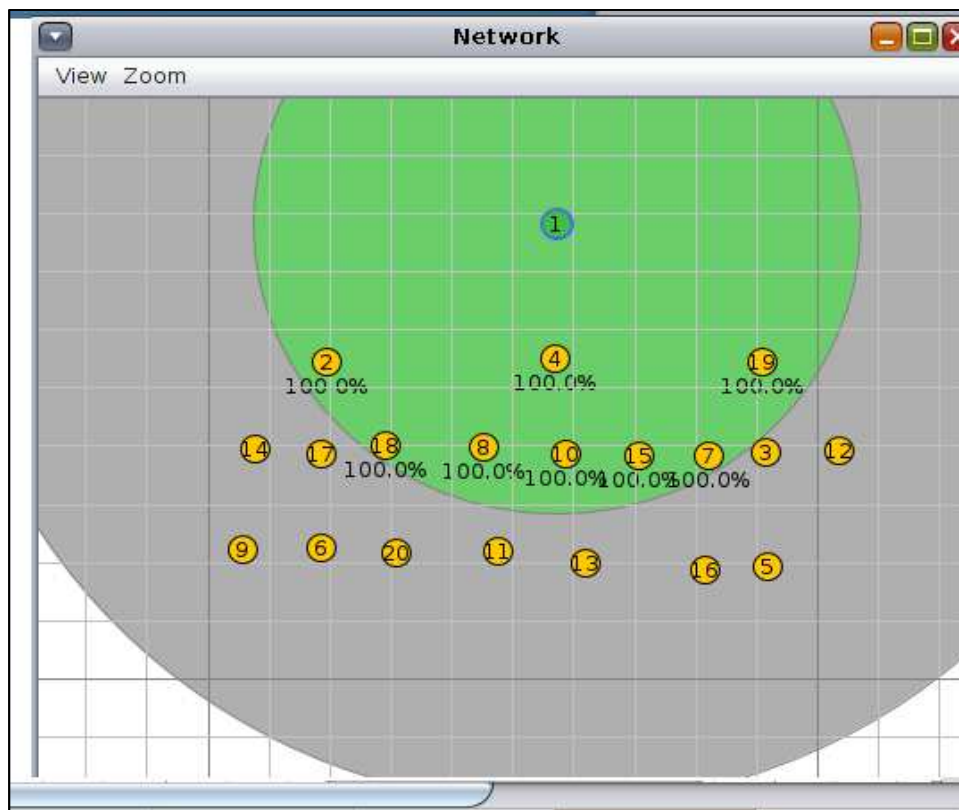


Figure IV-8: Exemple de simulation

IV.4.1 Evaluation de l'overhead

L'overhead représente le nombre total de messages de contrôle transmis par les nœuds afin de construire un DODAG. Ces messages contiennent les messages DAO, DIS et DIO.

$$\text{Control Traffic Overhead} = \sum DAO + \sum DIS + \sum DIO \quad (1)$$

Résultat 1.1

Dans le but d'illustrer l'amélioration Opt_RPL proposé par rapport aux fonctions objectifs OF0 et MRHOF, nous avons évalué un ensemble de métriques notamment l'overhead. Afin de d'évaluer l'overhead en fonction du temps, on a calculé le trafic représenté par la somme des messages de contrôle dans une topologie (voir équation), en faisant varier la durée dans chaque test, et en utilisant les deux fonctions objectifs OF0 et MRHOF ainsi que Opt_RPL..

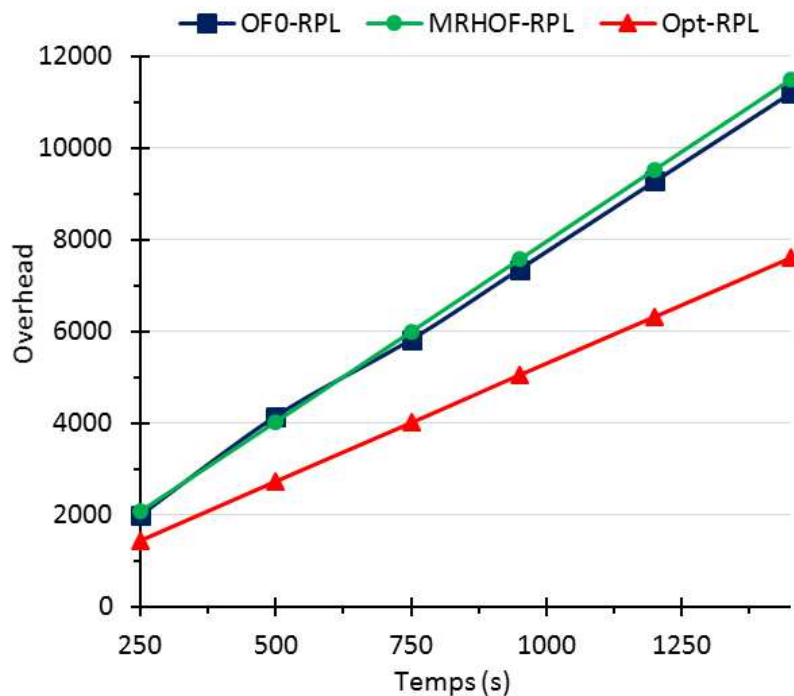


Figure IV-9: Overhead en fonction du temps (20 nœuds)

Les résultats illustrés dans la figure IV-9 montrent clairement que grâce à l'utilisation de RPL optimisé le trafic est moins important dans un réseau comparé celui obtenu avec la fonction OF0 et MRHOF. Cela implique que l'utilisation de Opt-RPL a eu un impact sur l'évolutivité du réseau.

Résultat 1.2

Afin d'évaluer l'overhead en fonction de la taille du réseau, on a calculé le trafic dans plusieurs topologies avec différents nombre de nœuds, en utilisant les deux fonctions objectifs OF0 et MRHOF ainsi que la notre Opt_RPL pour illustrer le gain apporté par notre contribution.

Les résultats illustrés dans la figure IV-10 montre clairement que le trafic augmente quand le nombre de nœuds augmente. On remarque aussi que le trafic obtenu avec les fonctions

objectifs OF0 et MRHOF est plus important comparé à celui obtenu avec Opt_RP, cela implique que Opt_RPL a eu un impact sur l'évolutivité, et ça a influencé sur la QoS.

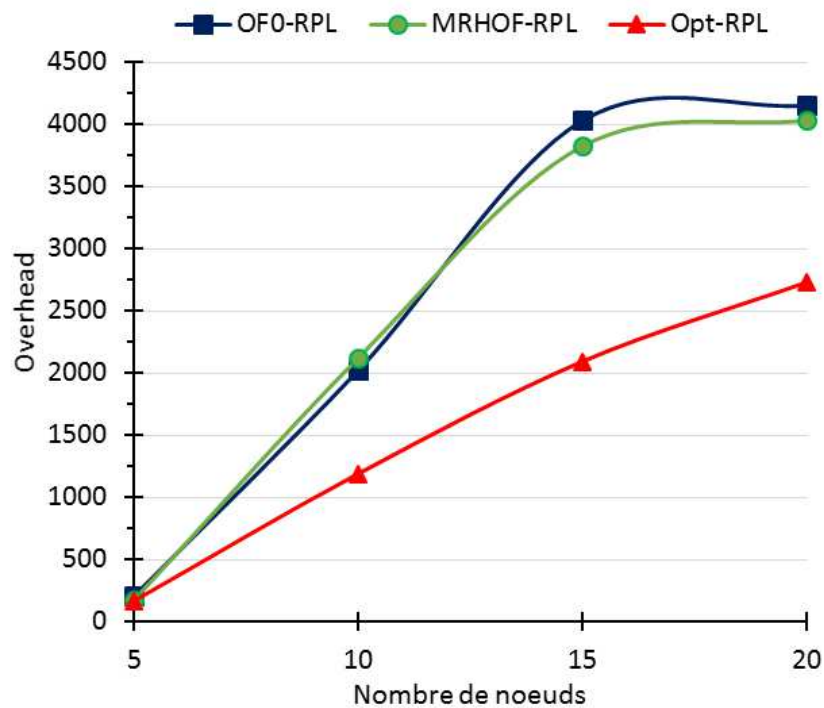


Figure IV-10: Overhead en fonction de la taille du réseau

Résultat 1.3

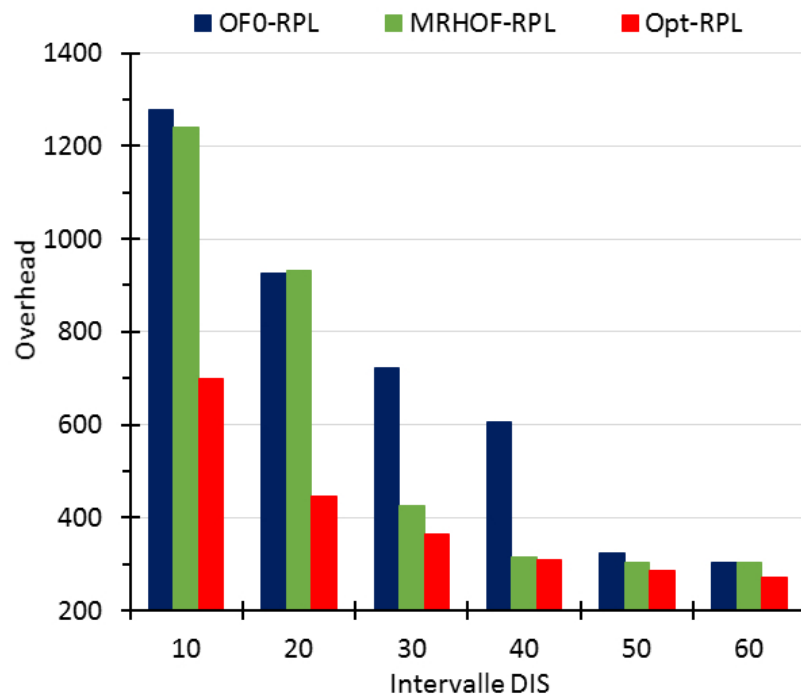


Figure IV-11: Overhead en fonction de l'intervalle DIS

Afin d'illustrer la variation de l'overhead en fonction de l'intervalle DIS, on a calculé le trafic, en changeant l'intervalle DIS après chaque test, en utilisant les deux fonctions objectives OF0 et MRHOF ainsi que Opt_RPL dans le but est de montrer l'apport de notre contribution.

Les résultats illustrés dans la figure IV-11 nous montre le trafic obtenu en faisant varier l'intervalle DIS. On remarque qu'avec une augmentation de l'intervalle de transmission des DIS, le nombre de messages de contrôle diminue particulièrement avec Opt-RPL en le comparant avec les fonctions objectives OF0 et MRHOF. Cela est dû à la réduction des différentes sollicitations des nouveaux nœuds rejoignant le réseau ; qui représentent un échange des messages de contrôle inutile.

IV.4.2 Evaluation de l'énergie consommée

Ce critère de performance représente l'énergie consommée par l'ensemble des nœuds du réseau.

Résultat 2.1

Dans cette simulation, on a évalué la consommation de l'énergie en fonction du temps pour les fonctions : OF0, MRHOF, et Opt_RPL.

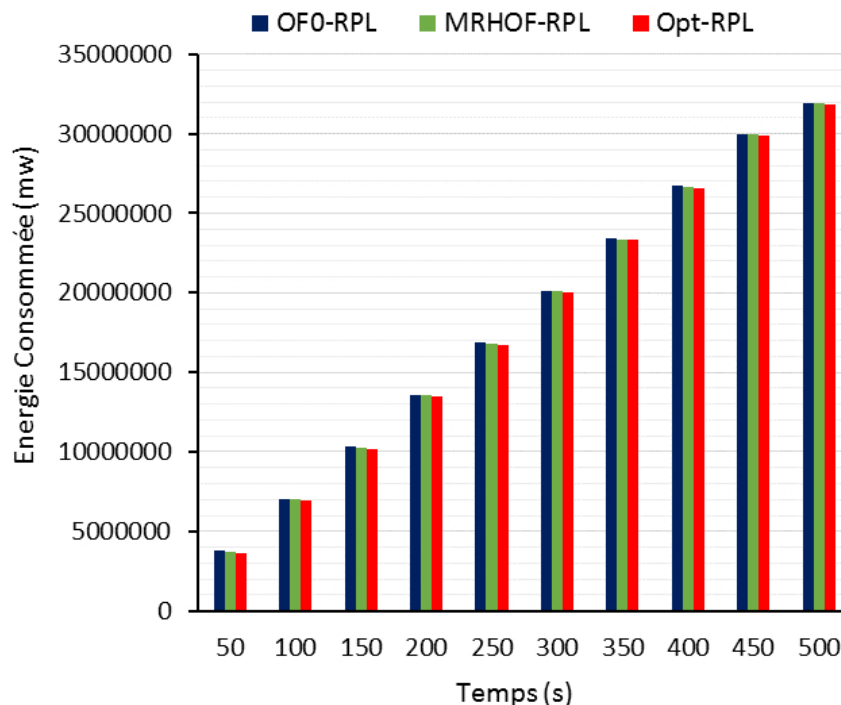


Figure IV-12: Energie consommée en fonction du temps

La figure IV-12 représente la consommation d'énergie en fonction du temps. A travers ces histogrammes, on remarque que l'énergie dans les trois contextes augmente de la même quantité quand le temps augmente.

Résultat 2.2

Dans ce contexte, on a évalué l'énergie consommée en fonction de la taille du réseau (nombre de nœuds) dans les trois scénarios : OF0, MRHOF et Opt-RPL

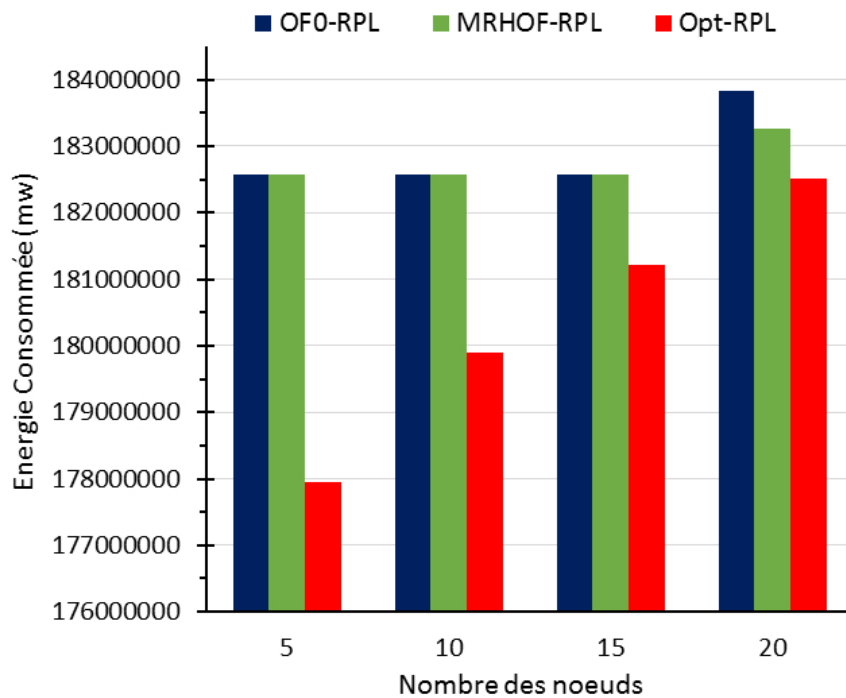


Figure IV-13: Energie consommé en fonction de la taille du réseau

La figure IV-13 représente la consommation d'énergie en fonction de la taille du réseau. On remarque que lorsqu'on a un réseau avec un petit nombre de nœud notre Opt-RPL a donné un bon résultat. Le fait d'économiser les messages de contrôle ça a réduit la quantité d'énergie, ce qui est plus bénéfique pour un réseau contraint tel que l'IoT.

IV.4.3 Evaluation du délai

Résultat 3

Dans ce scénario, nous avons évalué la variation du délai de bout en bout en fonction de l'intervalle DIS.

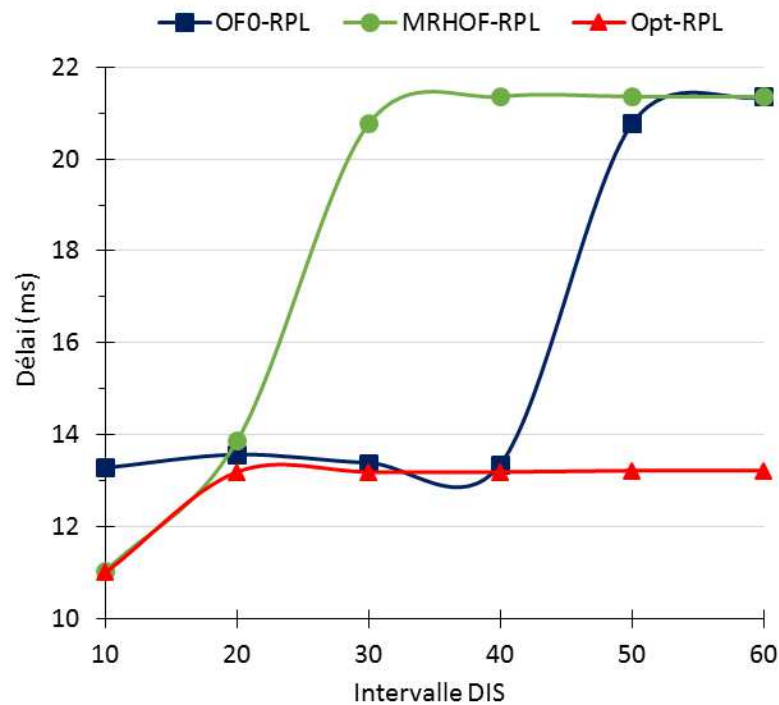


Figure IV-14: Variation du délai en fonction de l'intervalle DIS

La figure IV-14 montre la variation du délai de bout en bout en fonction de l'intervalle DIS. On remarque que le délai n'a quasiment pas changé dans le cas de "Opt-RPL", par contre pour les deux fonctions OF0 et MRHOF, il a augmenté à partir d'un certain intervalle (dans MRHOF à partir de 20 et dans OF0 à partir de 40). Le fait qu'il est resté stable en consommant moins de temps par rapport aux deux autres fonctions ; cela est due principalement au routage optimisé dans OpT-RPL qui a fait gagner du temps lors de la construction du graphe durant la première phase avant le routage des données.

IV.5 Conclusion

Dans ce chapitre nous avons présenté une amélioration du protocole RPL en termes d'overhead, de délai et de consommation d'énergie dans un réseau. Les résultats obtenus à travers les différentes simulations nous ont permis de voir que notre amélioration proposée était bénéfique et elle a eu un impact sur la qualité de service dans l'IoT.

Conclusion Générale

Conclusion générale

Le concept d'Internet des objets "IoT" est en pleine explosion durant ces dernières années. De ce fait, des solutions doivent cependant être utilisées pour assurer une communication optimale entre ces objets. Ces objets sont très contraints en énergie et forme des réseaux à faible puissance et à pertes "LLN". Pour cela plusieurs protocoles sont proposés pour effectuer le routage de données comme le protocole RPL dans ce type de réseaux.

Dans ce projet, nous avons étudié le routage dans les réseaux LLN et plus particulièrement le protocole de routage RPL. Ce protocole a été développé pour répondre aux limites des réseaux LLN telles que la faible puissance de traitement, de batterie et de mémoire.

Le protocole RPL ne définit aucune action à entreprendre dans le cas où un parent refuserait d'installer une nouvelle route descendante (en raison de la table de routage surchargée). Ces problèmes peuvent avoir un impact négatif sur l'évolutivité du réseau, pour cela nous avons proposé une solution qui consiste à éviter les messages inutiles en cas de saturation d'un nœud du réseau, chose qui sera bénéfique pour un réseau contraint tel que l'IoT.

Ce travail nous a permis de se familiariser avec les réseaux de capteurs sans fil. Il nous a aussi fait découvrir une nouvelle plateforme de programmation adéquate qui est Contiki et un nouveau simulateur (Cooja). Ce projet réalisé nous a été très bénéfique où on a pu comprendre le fonctionnement d'un nouveau type de réseaux (IoT) et un nouveau paradigme de routage pour les réseaux à faible puissance et à pertes (LLN) représenté par le protocole RPL.

Bibliographie

Bibliographie

- [1] I.F. Akyildiz et al., Wireless sensor networks: a survey, *Computer Networks* 38 (4) (2002) 393–422.
- [2] Y. Zhang and H. Xiao, “Bluetooth-based sensor networks for remotely monitoring the physiological signals of a patient,” *IEEE Transactions on Information Technology in Biomedicine*, vol. 13, no. 6, pp. 1040–1048, November 2009.
- [3] D.-M. Han and J.-H. Lim, “Smart home energy management system using ieee 802.15.4 and zigbee,” *IEEE Transactions on Consumer Electronics*, vol. 56, no. 3, pp. 1403–1410, August 2010.
- [4] C. Mettu. (2011) Telosb datasheet @ONLINE. [Online]. Available : <https://fr.scribd.com/doc/68138250/Telosb-Datasheet-t>,
- [5] M. Lehsaini, “Diffusion et Couverture basées sur le clustering dans les réseaux de capteurs”, Thèse de Doctorat, Université de Tlemcen et Université de Franche-Comté, 2009.
- [6] A. Dunkels, B. Gronvall, and T. Voigt. ”Contiki a Lightweight and Flexible Operating System for Tiny Networked Sensors”, In *Proceedings of the 9th Annual IEEE International Conference on Local Computer Networks*, Washington, DC, USA, 2004.
- [7] "TinyoS, Sensor Network Programming", https://www.iith.ac.in/~raji/downloads/TinyOS_Tutorial.pdf. [Accès le 16 Juillet 2016].
- [8] B. Malli. (2011, May) Micaz datasheet @ONLINE. [Online]. Available : <https://fr.scribd.com/doc/56641260/Micaz-Datasheet>. [Accès le 12 Avril 2020].
- [9] R. Nath. “A TOSSIM based implementation and analysis of collection tree protocol in wireless sensor networks”. *International Conference on Communications and Signal Processing (ICCSP)*, 2013.
- [10] P. Kugler, P. Nordhus, and B. Eskofier. “Shimmer, Cooja and Contiki : A New Toolset for the Simulation of On-node Signal Processing Algorithms”, *IEEE International Conference on Body Sensor Networks (BSN)*, 2013.
- [11] M. F. T. Mr. Mohamed BABOUCHE, *Optimisation du routage dans les réseaux LLNs*, Memoire de master en informatique, Centre Universitaire Aïn-Témouchent Belhadj Bouchaib 2019.
- [12] Carsten Bormann, JP Vasseur and Zack Shelby, “The Internet of Things”, *IETF Journal*, vol. 6, no. 2, November 2010.
- [13] "Contiki: The Open Source Operating System for the Internet of Things", www.contiki-os.org. [Accès le 12 Mars 2020].

Annexe

Annexe

Installation de Contiki

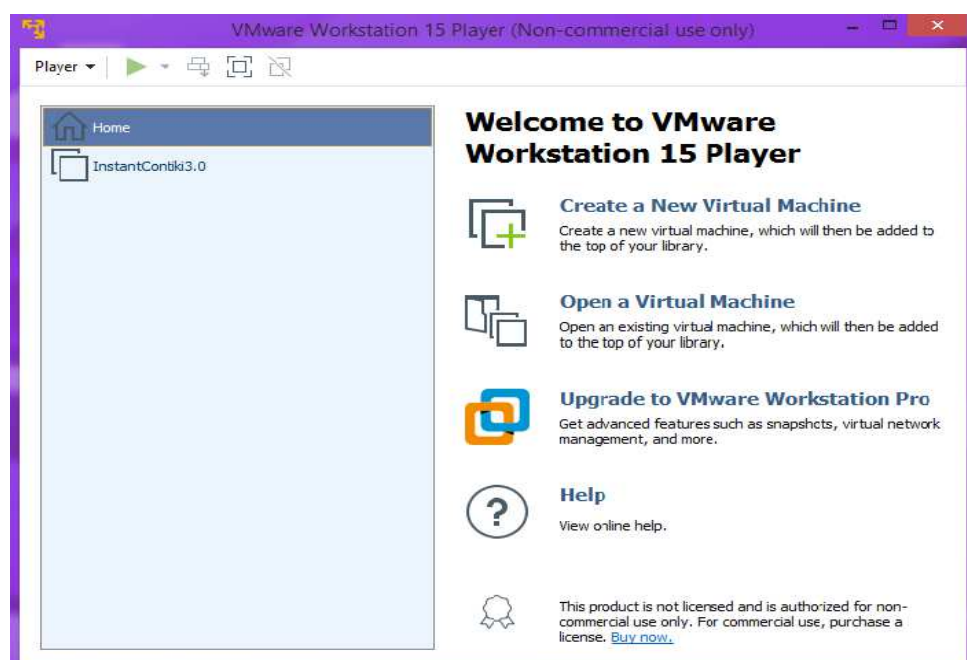
Contiki est un système d'exploitation conçu pour les systèmes IoT. Il supporte l'émulateur Cooja qui permet de réaliser des programmes sans faire recours à du matériel. Dans le cadre de ce projet, nous avons utilisé "Instant Contiki" qui est une machine virtuelle (VM).

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

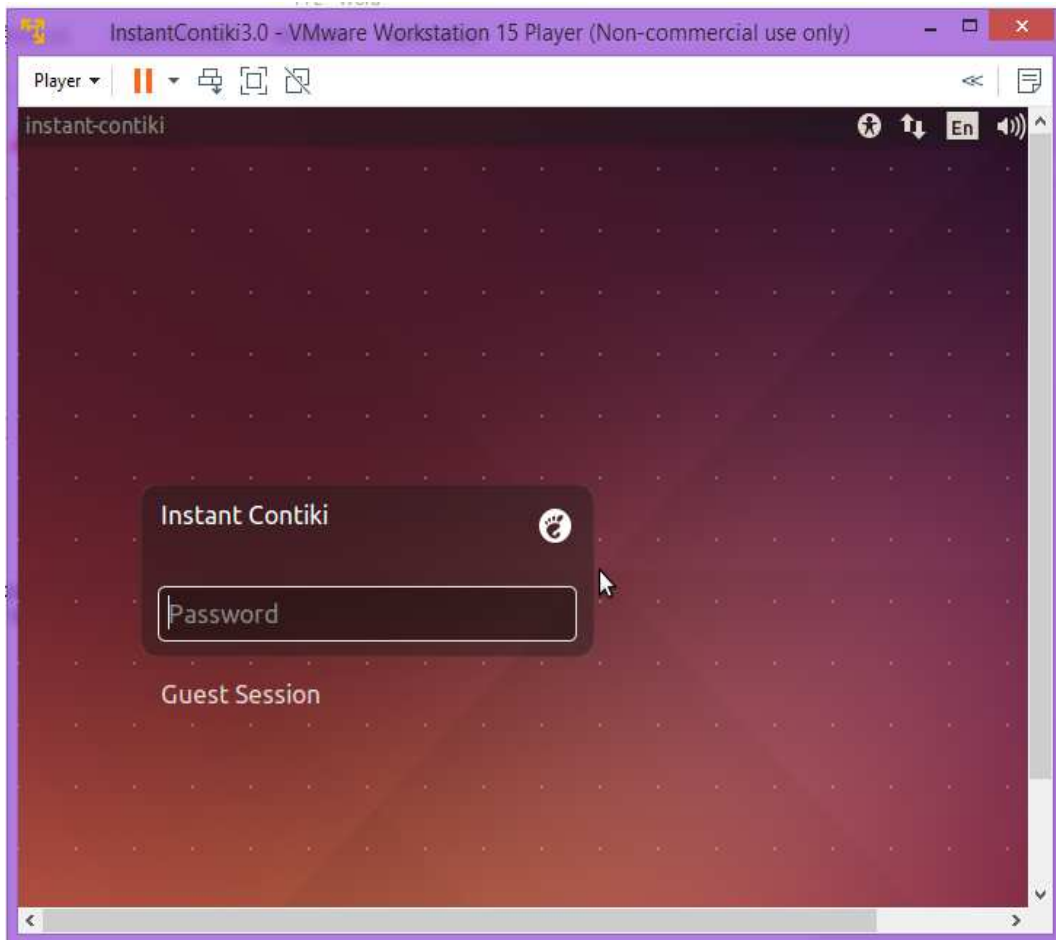
Dans notre PFE, nous avons utilisé VMware Workstation Player comme une application de virtualisation. Cet environnement est recommandé par la communauté scientifique Contiki pour exécuter l'environnement de développement "instant Contiki".

Dans ce qui suit, nous présentons les différentes étapes pour mettre en place la VM :

- Télécharger la machine virtuelle "Instant Contiki 3.0" à partir du site : <https://sourceforge.net/projects/contiki/files/Instant%20Contiki/> . Une fois qu'elle est téléchargée, décompressez le fichier et placez le répertoire décompressé sur le bureau.
- Pour manipuler cette machine virtuelle, il faut télécharger un lecteur des machines virtuelles tels que VirtualBox ou VMPlayer. Le lien de téléchargement est le suivant:
<https://www.clubic.com/telecharger-fiche121950-vmware-workstation.html>
<https://www.01net.com/telecharger/windows/Utilitaire/systeme/fiches/37588.html>



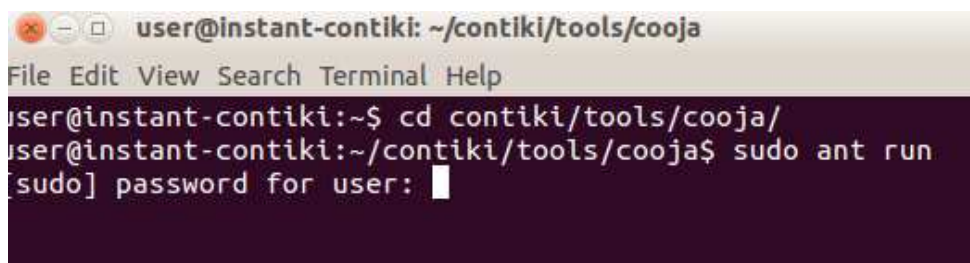
Une fois que le lecteur des machines virtuelles est installé avec la machine "Instant Contiki", utilisez le mot de passe : **user** pour pouvoir accéder.



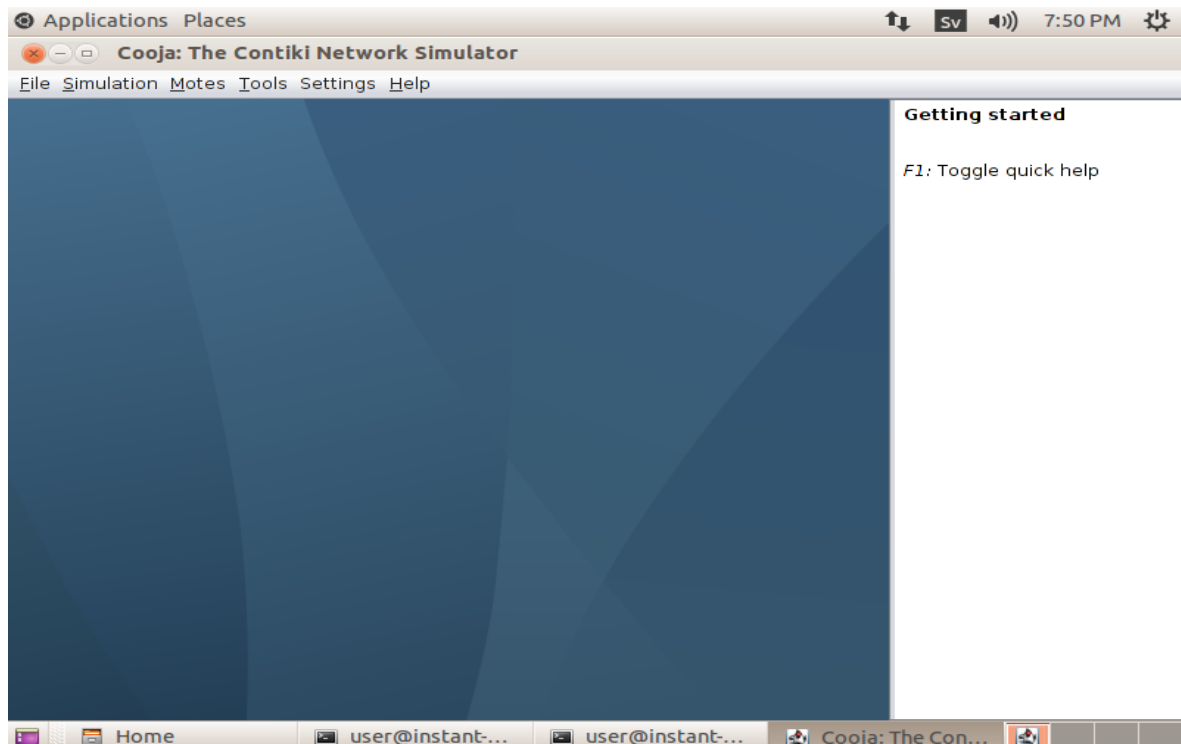
Description des simulations

Pour lancer Cooja, il faut d'abord ouvrir une fenêtre de terminal, puis accédez au répertoire Cooja avec la commande : **cd contiki/tools/cooja/**

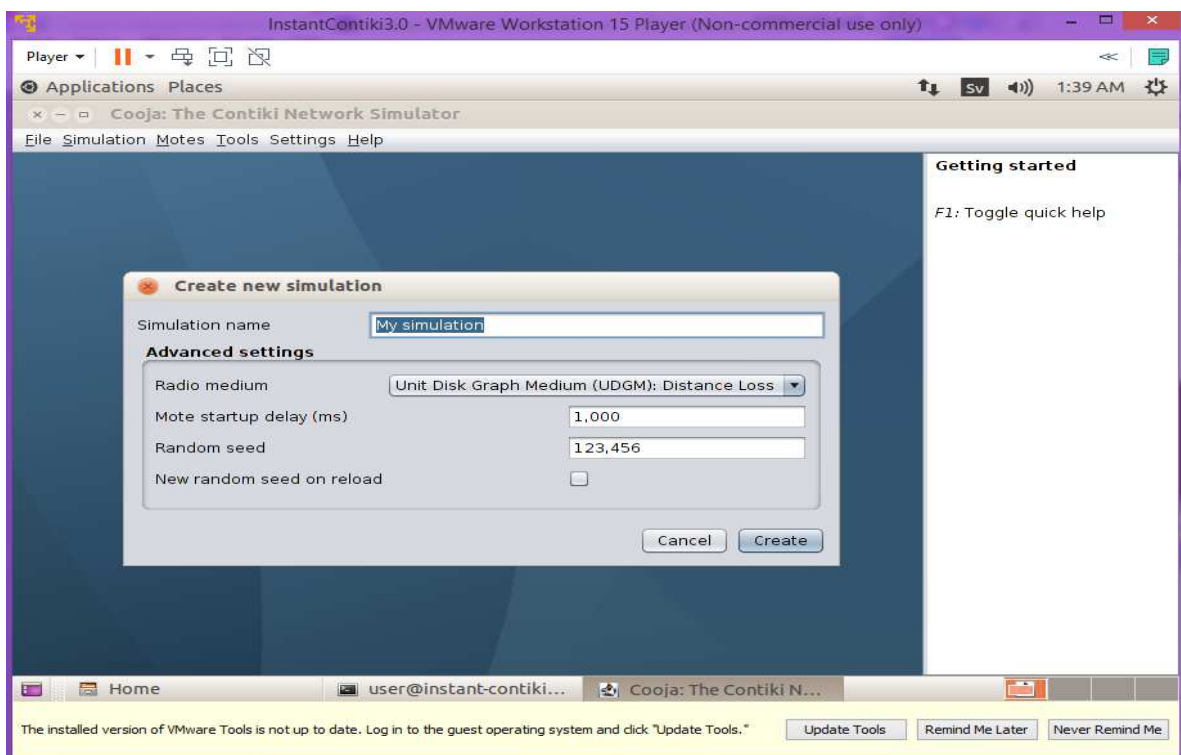
Puis lancer Cooja avec la commande : **ant run** ou **sudo ant run**



Une fois Cooja est lancé, vous auriez cette fenêtre, à ce moment-là vous pouvez commencer à simuler vos scénarios.

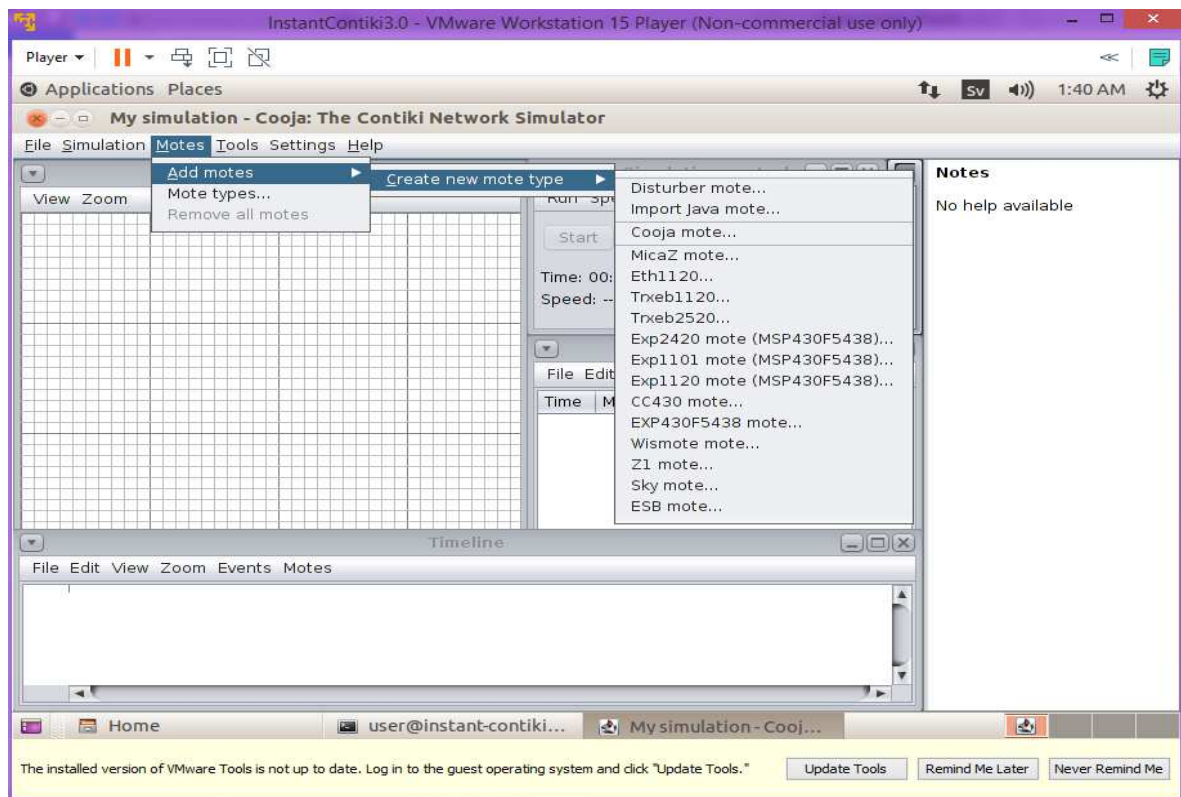


Cliquez sur le menu "File" puis cliquez sur "New simulation" pour générer une nouvelle simulation.



Une fois la configuration terminée, cliquez sur le bouton "Create" et Cooja fait apparaître la nouvelle simulation. Avant de pouvoir simuler un réseau, nous devons ajouter plusieurs

nœuds. Nous faisons cela via le menu "Motes", en cliquant sur "add motes" et nous devons définir le type de mote à ajouter.



Résumé

Ces dernières années, l'Internet des objets (IoT) a connu une grande croissance vu que notre besoin quotidien des objets intelligents est devenu une nécessité pour rendre nos tâches plus faciles. Ces objets sont souvent très contraints en énergie et peuvent former des réseaux à faible puissance et à pertes (LLN).

IETF a proposé au niveau de la couche réseau le protocole RPL comme standard de routage pour les réseaux dont les nœuds sont à ressources limitées tel que les réseaux LLN. Néanmoins, ce protocole présente toujours quelques anomalies qui nécessitent d'être améliorées. Ainsi, dans le cadre de notre PFE, nous avons visé à apporter une amélioration du protocole RPL en particulier lors de l'évolutivité du réseau en cas de la saturation de la mémoire d'un nœud. Dans ce contexte, nous avons proposé une solution qui consiste à éviter les messages inutiles en cas de saturation de la mémoire d'un nœud. Cette solution nous a permis de réduire le surcoût (messages de contrôles).

Nous avons validé notre contribution à travers plusieurs simulations et les résultats obtenus étaient prometteurs en termes de consommation d'énergie, d'overhead et de délai de bout en bout.

Mots-clés : IoT, QoS, RPL, Routage, Simulations, Réseaux LLN, Contiki.

Abstract

In recent years, the Internet of Things (IoT) has grown significantly, as our daily need for smart objects has become a necessity to make our tasks easier. These objects are often very constrained in energy and can form networks with low power and losses (LLN).

IETF proposed at the network layer the RPL protocol as a routing standard for networks whose nodes are limited resources such as LLN networks. However, this protocol still has some anomalies that need to be improved. Thus, within the framework of our PFE, we aimed to bring an improvement of the RPL protocol in particular during the scalability of the network in the event of the saturation of the memory of a node. In this context, we proposed a solution, which consists in avoiding the useless messages in the event of saturation of the memory of a node. This solution allowed us to reduce the additional cost (control messages).

We validated our contribution through several simulations and the results obtained were promising in terms of energy consumption, overhead and end-to-end delay.

Keywords : IoT, QoS, RPL, Routing, Simulations, LLN Networks, Contiki.

ملخص

في السنوات الأخيرة ، نمت إنترنت الأشياء (IoT) بشكل كبير حيث أصبحت حاجتنا اليومية للأشياء الذكية ضرورة لتسهيل مهامنا. غالبًا ما تكون هذه الأشياء مقيدة للغاية في الطاقة ويمكن أن تشكل شبكات ذات طاقة منخفضة وخسائر (LLN). اقترح IETF على مستوى طبقة الشبكة بروتوكول RPL كمعيار توجيه للشبكات التي تكون عناصرها ذات إمكانيات محدودة كما هو الشأن في شبكات LLN. ومع ذلك ، لا يزال هذا البروتوكول يحتوي على بعض النقائص التي تحتاج إلى تحسين. وبالتالي ، في إطار هذه المذكرة، سعينا إلى تحسين بروتوكول RPL على وجه الخصوص أثناء قابلية التوسع في الشبكة في حالة تشبع ذاكرة أحد العناصر. في هذا السياق ، اقترحنا حلًا يتمثل في تجنب الرسائل عديمة الفائدة في حالة تشبع ذاكرة العنصر المعني. سمح لنا هذا الحل بتخفيض التكلفة الإضافية (رسائل التحكم).

لقد تحققنا من مساهمتنا من خلال العديد من المحاكاة وكانت النتائج التي تم الحصول عليها واعدة من حيث استهلاك الطاقة و التكلفة الإضافية والتأخير.

الكلمات المفتاحية: IoT, QoS, RPL, Routing, Simulations, LLN Networks, Contiki.