MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITE ABDELHAMID IBN BADIS - MOSTAGANEM

Faculté des Sciences Exactes et d'Informatique

Département de Mathématiques et informatique

Filière : Informatique

MEMOIRE DE FIN D'ETUDES

Pour l'Obtention du Diplôme de Master en Informatique

Option : **Réseaux et Systèmes**

Présenté par :

Mahamat-Saleh Abdoulaye Ali

THEME :

Estimation de la Qualité de Lien dans RPL

Soutenu le : 19/06/2021

Devant le jury composé de :

BESNASSI Miloud	Université de Mostaganem	Président
MOUSSA Mohamed	Université de Mostaganem	Examinateur
ABID Meriem	Université de Mostaganem	Encadrante

Année Universitaire 2020-2021

REMERCIEMENTS

Je tiens dans un premier temps à remercier ALLAH le tout puissant qui m'a donné le courage et la volonté pour mener à bien ce modeste travail.

Je tiens à remercier mon encadrante Dr Abid Merem pour tout le temps qu'elle m'a consacré, pour ses conseils et orientations précieux, pour toute son aide et son appui durant la réalisation de ce travail.

Je tiens à remercier les membres jury pour m'avoir fait l'honneur d'examiner et d'évaluer mon travail. À tous mes enseignants et les membres du département informatique du FACULTE DES SCIENCES EXACTE ET DE L'INFORMATIQUE de Mostaganem.

Une mention spéciale à ma famille pour tous leurs sacrifices, leur amour, leur tendresse, leur motivation, leurs prières tout au long de mes études.

Mes remerciement vont aussi à l'endroit de mes compatriotes et amis dont leur soutien morale et encouragements ont été précieux tout au long de mon cursus.

Que tous ceux et celles qui ont contribué de près ou de loin à l'accomplissement de ce travail trouvent ici l'expression de mes remerciements les plus chaleureux.

RÉSUMÉ

L'emergence de l'internet des objets(IoT) a permis a un grand nombre d'appareils intelligents de se connecter a l'internet. Le protocole de routage RPL(Protocol for Low Power and Lossy Networks) basé sur IPv6, a été conçu pour les réseaux à faible puissance et à perte(LLN), qui sont une catégorie de réseaux dans laquelle les noeuds et leurs interconnexions sont fortement contraints. Les noeuds fonctionnent avec des fortes contraintes sur la puissance de traitement, l'énergie, et la mémoire. Quant aux interconnections, elles sont instables.

RPL a été standarisé par le groupe ROLL(Routing Over Low power and Lossy networks) de l'IETF (Internet Engineering Task Force)[3], qui estime la qualité des liens utilisés pour livrer les paquets de données. Pourtant, il a été démontré que cette solution provoquait des périodes de routage instabilité et taux de livraison de paquets réduits car il estime seulement la qualité des liens utilisés.[2]

Dans ce projet, nous nous intéressons à la question du routage et au fonctionnement de RPL dans un premier temps en suite nous allons faire un état de l'art sur les travaux existantes qui estiment la qualité de lien et puis nous allons parler d'une solution d'estimation et puis son évaluation.

Mot-clés : RPL, LLN, OF, Trickle Timer, estimation, qualité de lien.

TABLE DES FIGURES

LISTE DES TABLEAUX

TABLE DES MATIÈRES

INTRODUCTION GÉNÉRALE

A l'heur actuel, les progrès technologiques nous permettent de se projeter à la connexion des objets du quotidien à l'Internet. L'omniprésence programmée de ces objets implique la fabrication d'objets bon marché (peu d'autonomie énergétique, peu de mémoire de stockage et peu de puissance de calcul). L'ensemble connecté de ces objets est appelé réseau LLN (Low power and Lossy Network). Dans ce contexte, le protocole de routage RPL(IPv6 Routing Protocol for Low Power and Lossy Networks) est conçu à fin de prendre en charge les exigences spéciques de ces réseaux.

Notre objectif principal de ce projet est de faire des recherches sur les travaux qui estime la qualité des lien dans RPL pour une meilleure prise en charge de la mobilité afin d'améliorer les performances des LLNs, notamment la durée de vie du réseau et le nombre de paquets reçus en dépendent puis implémenter une meilleur solution pour l'estimation de qualité des lien. Ce projet comporte quatre(4) chapitres : Dans le premier chapitre, nous présentons les concepts de base des réseaux de capteurs sans fil en mettant l'accent sur les réseaux LLN ou réseaux à faible puissance et à perte. Nous finirons ce chapitre par la présentation des contraints de routage dans les réseaux LLN. Le deuxième chapitre, nous nous focaliserons sur le protocolede routage RPL en présentant les différents mécanismes mis en œuvre dans RPL, ainsi que son mode de fonctionnement. Dans le troisième chapitre, nous présenterons les approches d'estimation de la qualité des liens et la détail de la solution utilisée, et enfin le dernier chapitre, nous allons présenter les simulations de la solutions avec les résultats obtenus. Nous terminons ce mini

projet par une conclusion générale dans laquelle nous précisons les apports de notre démarche et les perspectives à cette dernière.

CHAPITRE

1

GÉNÉRALITÉ SUR LES RÉSEAUX DE CAPTEURS SANS FIL

1 Introduction

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

2. Définition d'un capteur

Un capteur est un dispositif destiné à mesurer une donnée environnementale comme la température, ou même capturer une image par exemple, et de la transformer en information utilisable puis de la transmettre à une unité de traitement de façon analogique ou numérique. Ces capteurs, non autonomes doivent donc être connectés à un appareil capable d'en interpréter la mesure, puis, selon l'usage souhaité permettre l'utilisation. Chaque capteur assure trois fonctions principales : la collecte, le traitement et la communication de l'information vers un ou plusieurs points de collecte appelés station de base.[14]

3. Définition d'un réseau de capteur sans fils (RCSF)

Un Réseau de Capteurs Sans Fil (RCSF) ou Wireless Sensor Network (WSN) est un réseau informatique composé de petits dispositifs autonomes, fixés ou dispersés aléatoirement dans une zone d'intérêt, utilisant des capteurs coopérant pour surveiller des conditions environnementales ou physiques, comme la température, le son, les vibrations, la pression, le mouvement, etc. Puisque les réseaux de capteurs sans fil peuvent être déployés dans des terrains inaccessibles, la position des noeuds capteurs ne peut être prédéterminée. En conséquence, un système de localisation est requis afin de fournir les informations de position aux noeuds. Parmi les domaines d'application on trouve la santé, le domaine militaire, et de la sécurité.[14]

3 Architecture d'un réseau de capteur sans fil

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

4 La notion de QoS dans les réseaux de capteurs sans fils

Le terme QoS (« Qualité de Service, en anglais : “Quality of Service”) désigne la capacité à fournir un service, ici un support de télécommunications, conforme à des exigences de fonctionnement acceptable à assurer par le fournisseur du service envers l'utilisateur.

Appliquée aux réseaux à commutation de paquets (réseaux basés sur l'utilisation de routeurs) la QoS désigne l'aptitude à pouvoir garantir le non-dépassement d'un niveau acceptable de baisse de qualité, défini contractuellement, pour un usage donné (voix sur IP, vidéoconférence, etc.).

En effet, contrairement aux réseaux à commutation de circuits, tels que le réseau téléphonique commuté, où un circuit de communication est dédié pendant toute la durée de la communication, il est impossible sur Internet de prédire le chemin emprunté par les différents paquets.

Cette incertitude est encore plus forte sur les réseaux sans-fil, où le médium radio lui-même est sujet à la diffusion, dont l'état et l'encombrement peuvent varier fortement et rapidement au cours du temps. Ce médium est, comparé aux câbles des réseaux informatiques « classiques », bien moins fiable.

Ainsi, rien ne garantit qu'une communication nécessitant une régularité du débit pourra avoir lieu sans encombre. C'est pourquoi il existe des mécanismes, dits mécanismes de QoS, permettant de différencier les différents flux réseau et réserver une partie de la bande passante pour ceux ayant une importance particulière.

Plus formellement, la recommandation E.800 de septembre 2008 de l'Union Internationale des Télécommunications (UIT) définit la QdS comme étant « l'ensemble des caractéristiques d'un service de télécommunication qui lui permerrent de satisfaire aux besoins explicites et aux besoins implicites de l'utilisateur du service » ; une caractéristique étant définie comme une « propriété (qualitative ou quantitative) qui aide à faire la distinction entre les individus d'une population donnée ».

En termes pratiques, une caractéristique du standard E.800 de l'UIT correspond à un critère de QdS. Le but à atteindre étant d'assurer une valeur minimale en-deça de laquelle ne pas tomber pour respecter un contrat de qualité avec l'utilisateur.

Les principaux critères permettant d'apprécier la qualité de service sont :

1) Perte de paquets

2) Débit

3) Latence, délai, ou temps de réponse

4) Déséquencement

5 Le routage dans les réseaux de capteurs sans fils

Le routage est un mécanisme pour l'acheminement des paquets de données d'un émetteur vers un ou plusieurs destinataires, à travers des routeurs et des protocoles de routage, les routeurs sont des machines qui relient deux ou plusieurs réseaux afin de commuter des paquets d'une interface vers une autre selon la destination des paquets, les protocoles de routage définissent la manière de fonctionnement des routeurs selon la propriété de protocole, les protocoles de routages sont classés en plusieurs grande classe selon des différent critères. Les protocoles de routage servent à deux fonctions, la première est la construction avec la maintenance des routes pour certaines destinations, la deuxième consiste en l'acheminement des données sur ces routes.

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

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

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

5.1. Taxonomie des protocoles de routages

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

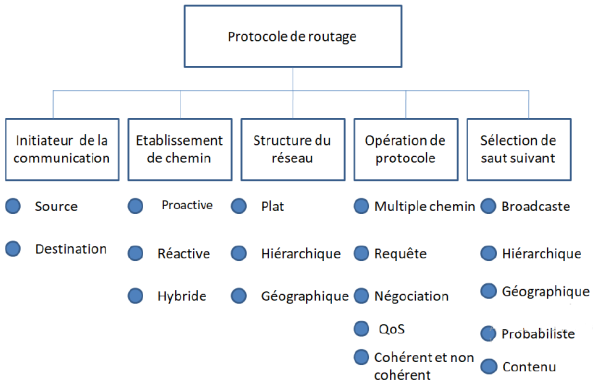

FIGURE 1.1 – Classification des protocoles de routage

5.2. Protocoles de routage : établissement de route

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

1) Protocole réactif : Ces protocoles (dits aussi, les protocoles de routage à la demande) créent et maintiennent des routes selon les besoins. Lorsque le réseau a besoin d'une route, une procédure de découverte de route est lancée. Ce type de protocoles est pratique pour des applications temps réel où les capteurs doivent réagir immédiatement à des changements soudains des valeurs captées. En effet, un prélèvement périodique des données aurait été inadapté pour ce type de scénarios

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

3) Protocole hybride : utilisent un protocole proactif pour apprendre le proche voisinage (par exemple le voisinage à deux ou à trois sauts), ainsi, ils disposent de routes immédiatement dans le voisinage. Au-delà de la zone du voisinage, le protocole hybride fait appel à un protocole réactif pour chercher des routes.

5.3. Protocoles de routage :la structure réseau

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

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

   – Energy Aware Routing (EAR).

   – Directed Diffusion (DD).

   – Sequential Assignment Routing (SAR).

   – Minimum Cost Forwarding Algorithm (MCFA).

   – Sensor Protocols for Information via Negotiation (SPIN).

   – Active Query forwarding In sensor network (ACQUIRE).

2) Le routage hiérarchique : Les méthodes de routage hiérarchique ont des avantages spéciaux liés au passage à l'échelle et à l'efficacité dans la communication. Par exemple, elles sont utilisées pour exécuter un routage avec économie d'énergie.[6] Le réseau est

divisé en clusters dirigés par un chef de clusters. La création des clusters et l'assignation des tâches spéciales aux têtes de clusters peuvent considérablement renforcer le passage à l'échelle, l'augmentation de la durée de vie et l'efficacité énergétique du système global. Le routage hiérarchique est une manière efficace de réduire la consommation énergétique dans un cluster en exécutant l'agrégation et la fusion de données afin de diminuer le nombre de messages transmis à la station de base.

Exemples de protocoles de routage hiérarchique :

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

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

   On peut citer quelques protocoles appartenant à cette catégorie :

   ❖ Geographic Adaptive Fidelity (GAF).
   ❖ Geographic and Energy-Aware Routing (GEAR).
   ❖ Trajectory-Based Forwarding (TBF).

- ❖ Sequential assignment routing (SAR).
- ❖ Ad-hoc positioning system (APS).
- ❖ Greedy other adaptive face routing (GOAFR).
- ❖ Geographic distance routing (GEDIR).

6. Facteurs et contraintes des RCSF

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

6.1. Durée de vie du réseau

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

6.2. Ressources limitées

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

6.3. Bande passante limitée

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

6.4. Topologie du réseau

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

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

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

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

6.5. L'environnement de déploiement

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

7. L'internet des objets

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

8. Low-Power and Lossy Networks

Les réseaux de faible puissance et à pertes (LLN) sont ceux dans lesquels les noeuds et leurs interconnexions sont fortement contraints par les ressources. Les noeuds sont généralement limités en termes de puissance de traitement, de batterie et de mémoire, et leurs interconnexions

FIGURE 1.2 – l'Internet des objets

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

8.1. IPV6

Au début des années 1990, il est devenu clair que le développement d'internet allait aboutir à l'épuisement des adresses disponibles fournie par IPV4, ce qui a conduit à l'émergence de IPv6 après des travaux menés au sein de l'IETF au cours des années 1990.

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

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

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

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

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

8.2. La norme IEEE 802.15.4

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

9. 6LOWPAN

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

afin de réduire la taille des datagrammes IPv6 à savoir la fragmentation et la compression des entêtes afin de permettre aux paquets IPv6 d'être envoyés ou reçus via les réseaux LLN.

10. Le routage dans les réseau LLNs

10.1. Contraintes de routage dans les réseaux LLN

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

10.2. Algorithmes de routage dans les réseaux LLN

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

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

- Le groupe de travail ROLL (Routing over Low-Power and Lossy Links) : C'est un groupe

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

11. Conclusion

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

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

CHAPITRE

2

LE PROTOCOLE DE ROUTAGE RPL

1 Introduction

Les réseaux à faible puissance et avec perte (LLN) sont des réseaux constitués de noeuds (avec une puissance de traitement, une mémoire et parfois de l'énergie limitées lorsqu'ils sont alimentés par batterie. A partir de 2008, après plusieurs tentatives pour spécifier un protocole de routage efficace pour les LLN, l'IETF a formé le groupe de travail ROLL dans l'objectif principale d'élaborer un protocole de routage efficace pour les réseaux LLN, basé sur IPv6 dont le résultat est RPL(Routing Protocole for Low Power and Lassy Network). La popularité croissante de RPL est due à plusieurs facteurs, dont sa flexibilité pour s'adapter à différentes topologies, la prise en charge de la qualité de service. Dans ce chapitre, nous allons présenter le fonctionnement du protocole de routage RPL.

2 Historique

D'un point de vue historique, les travaux sur les protocoles de routage adaptés au LLN sont le résultat d'une évolution des usages et de l'émergence des appareils contraints. En effet, aux

abords des années 2000, la multiplication des appareils mobiles sans fil a entraîné la création de nouveaux protocoles de routage spécialement conçus pour prendre en compte la contrainte de mobilité. Au niveau standardisation, l'IETF a donc créé le groupe de travail (Mobile Ad-hoc Networks) (MANET) en 1998 pour définir des protocoles adaptés aux enjeux du sans fil.

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

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

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

2) ROLL : (Routing Over Low Power and Lossy Networks) ROLL assurera une coordination étroite avec les groupes de travail d'autres domaines spécialisés dans les réseaux et / ou les noeuds sous contraintes, tels que 6lo, 6tisch, ipwave, lwig et CoRE.

3) MANET(Internet Ad-hoc Mobile Network) Un "réseau mobile ad hoc" (MANET) : est un système autonome de routeurs mobiles (et hôtes associés) connectés par des liaisons sans fil - dont l'union forme un graphe arbitraire. Les routeurs sont libres de se déplacer de manière aléatoire et de s'organiser de manière arbitraire. ainsi, la topologie sans fil du réseau peut changer rapidement et de manière imprévisible. Un tel réseau peut fonctionner de manière autonome ou être connecté à un réseau Internet plus vaste.

3 Pourquoi RPL ?

RPL est un protocole de routage proactif à vecteur de distance. Il est sans doute l'un des protocoles de routage IPv6 les plus connus pour les réseaux à faible consommation et à perte (LLN) développé par ROLL[3] pour répondre aux limites des réseaux LLN telles que la faible puissance de traitement, de batterie et de mémoire. Il cible le plus souvent les réseaux basés sur le collecte des données, où les noeuds envoient momentanément des données à un point de collecte. Une de qualité clé de RPL est qu'il représente une solution de routage spécifique pour les réseaux à faible puissance et à perte, dont la gestion critique du perte des paquets. Compte tenue des exigences des noeuds à ressource limités, le protocole de routage RPL a comme caractéristiques[1] :

1) le débit de données prospectif est généralement faible (moins de 250 kbps)

2) la communication est sujette à des taux d'erreur élevés, ce qui entraîne un faible débit de données

Mise à part le taux d'erreur binaire qui caractérise une liaison avec perte, le long du temps d'inaccessibilité qui aura un impact sur la conception du protocole d'où la conception de RPL pour être hautement adaptatif aux condition du réseau et pour fournir des routes alternatives, chaque fois que les routes par défaut sont inaccessibles.

RPL est basé sur le concept topologique des graphes acycliques dirigés (DAGs). Le DAG définit une structure arborescente qui spécifie les routes par défaut entre les noeuds du LLN. Cependant, une structure DAG est plus qu'un arbre typique dans le sens où un noeud peut s'associer à plusieurs noeuds parents dans le DAG, contrairement aux arbres classiques où un seul parent est autorisé. Plus spécifiquement, RPL organise les noeuds en tant que DAG orientés destination (DODAG), où les noeuds de destination les plus populaires (c'est-à-dire les puits) ou ceux fournissant une route par défaut vers Internet (c'est-à-dire les passerelles) agissent comme les racines des DAG. Une topologie de réseau basée sur RPL est intrinsèquement hiérarchique car elle oblige les noeuds sous-jacents à s'auto-organiser en un ou plusieurs DODAG, en fonction de la relation parent-enfant et prend en charge la topologie maillée car elle permet le routage via des frères et soeurs à la place des parents et des enfants, au besoin. Cette combinaison maillage et hiérarchique offre une grande flexibilité en termes de routage et de gestion de la topologie qui sont entre autres :

1) Configuration automatique

2) L'auto-réparation

3) Évitement et détection de boucle

4) Indépendance et transparence

5) Plusieurs routeurs de périphérie

3 Les caractéristiques de RPL

Dans cette section, nous présenterons les principaux mécanismes et fonctionnalités fournis dans RPL qui sont les méssages de contrôle, les mécanismes de communications, le routage ainsi que la construction des DODAG etc.

3.1 Les graphes DAG et DODAG

Graphe acyclique dirigé un graphe orienté ayant la propriété que toutes les arêtes sont orientées de telle manière qu'aucun cycle n'existe. Toutes les arêtes sont contenues dans des chemins orientés vers et se terminant à un ou plusieurs nœuds racines comme le montre le figure ci-dessous.

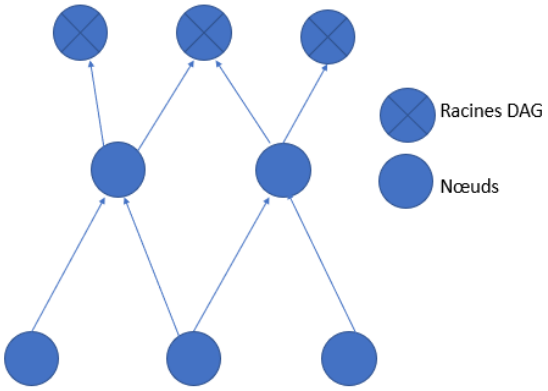

FIGURE 2.1 – Le graphe DAG

RPL s'appuie sur la notion de DODAG (graphe acyclique orienté vers la destination), DODAG est un DAG enraciné dans une seule destination, c'est-à-dire à une seule racine DAG.

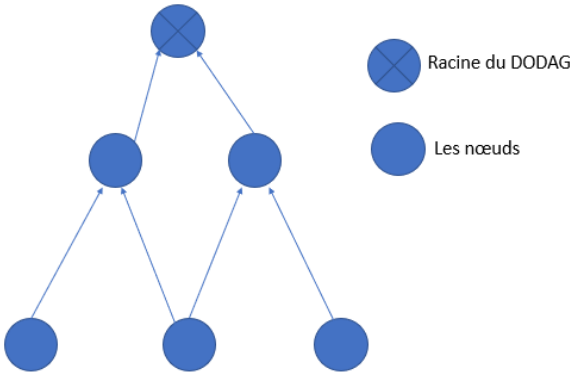

FIGURE 2.2 – Le graphe DODAG

3.2 Les messages de contrôle

Les messages RPL sont spécifiés comme un nouveau type de messages de contrôle ICMPv6. D'un protocole à un autre, le message de contrôle RPL est composé :

– D'un en-tête ICMPv6, qui se compose de trois champs : Type, Code et Checksum

– D'un corps de message comprenant une base de message et un certain nombre d'options.

Le champ Type spécifie le type du message de contrôle ICMPv6 défini de manière prospective dans le cas de RPL (confirmé par l'Internet Assigned Number Authority (IANA)). Le champ Code identifie le type de message de contrôle RPL.

Quatre codes sont actuellement définis :

1) DIS (Sollicitation Information DODAG) : peut être utilisé pour solliciter un objet d'information DODAG à partir d'un nœud RPL. Son utilisation est analogue à celle d'une sollicitation de routeur comme spécifié dans IPv6 pour la découverte du voisinage ; un nœud peut utiliser DIS pour sonder son voisinage DODAG à proximité. Le format de message DIS actuel contient des indicateurs et des champs non spécifiés pour une utilisation future.

2) DIO (Objet d'Information DODAG) : L'objet d'information DODAG transporte des informations qui permettent à un noeud pour découvrir un parent préféré, connaître ses paramètres de configuration,sélectionner un ensemble de parents DODAGs et gérer le

DODAG. Il est émis par la racine DODAG pour construire un nouveau DAG, puis envoyé en multidiffusion via la structure DODAG. Le message DIO transporte des informations réseau pertinentes qui permettent à un noeud de découvrir une instance RPL, d'apprendre ses paramètres de configuration, de sélectionner un ensemble parent DODAG et de gérer le DODAG.

Les principaux champs de l'objet de base DIO sont :

RPLInstanceID
Version Number
Rank
Destination Advertisement Trigger Sequence Number
Grounded
Mode of Operation
DDODAGPreference
DODAGID
Option field

TABLE 2.1 – Les champs d'un DIO

3) DAO(Destination Annonce Objet) : Les messages DAO sont envoyés par chaque noeud, autre que la racine DODAG, pour remplir les tables de routage avec les préfixes de leurs enfants et pour publier leurs adresses et préfixes à leurs parents. Après avoir transmis ce message DAO via le chemin d'un noeud particulier à la racine DODAG via les routes DAG par défaut, un chemin complet entre la racine DODAG et le noeud est établi. Les principaux champs de message DAO sont :

   – RPLInstanceID qui est l'ID de l'instance RPL

   – indicateur K

   – DAOSequence est un numéro de séquence incrémenté à chaque message DAO

   – DODAGID qui identifie un DODAG

4) DAO-ACK(Destination Annonce Objet - ACK) : Le message DAO-ACK est envoyé sous forme de paquet unicast par un destinataire DAO (un Parent DAO ou racine DODAG) en réponse à un message DAO monodiffusion. Il contient des informations sur RPLInstanceID, DAOSequence et Status, qui indiquent l'achèvement. Les codes d'état ne sont toujours pas clairement définis, mais les codes supérieurs à 128 signifient un rejet et qu'un noeud doit sélectionner un autre parent.

3.3 La construction de DODAG

La construction DODAG est basée sur le processus de découverte de voisin , qui consiste en deux opérations principales : la diffusion de la transmission des messages de contrôle DIO émis par la racine DODAG pour construire des routes dans le sens descendant de la racine vers les noeuds clients, la mono-diffusion des messages de contrôle DAO émis par les nœuds clients et envoyés à la racine DODAG pour construire des routes dans les directions ascendantes. Afin de construire un nouveau DODAG, la racine DODAG diffuse un message DIO pour annoncer son DODAGID, ses informations de rang pour permettre aux noeuds de déterminer leurs positions dans le DODAG et la fonction d'objectif identifiée par le point de code objectif (OCP) dans le Champs d'options de configuration du DIO. Ce message sera reçu par un noeud client qui peut être soit un noeud prêt à rejoindre, soit un noeud déjà joint. Lorsqu'un noeud souhaitant rejoindre le DODAG reçoit le message DIO :

– il ajoute l'adresse de l'expéditeur DIO à sa liste des parents

– calcule son rang à l'aide de la fonction d'objectif spécifiée dans le fichier OCP, de sorte que le rang du noeud soit supérieur à celui de ses parents

– transmet le message DIO avec les informations de classement mises à jour

Le parent le plus préféré (rang le plus pétit) parmi la liste des parents du noeud client sera choisi comme un noeud par défaut par lequel le trafic entrant est acheminé.

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

– rejeter le message DIO selon certains critères spécifiés par RPL

– traiter le message DIO pour maintenir son emplacement dans un DODAG existant

– améliorer sa localisation en obtenant un rang inférieur dans le DODAG sur la base du calcul du coût de chemin spécifié par la fonction Objective

Chaque fois qu'un noeud change de rang, il doit ignorer tous les noeuds de la liste des parents dont les rangs sont supérieurs au rang du nouveau noeud calculé pour éviter les boucles de routage.

Si l'indicateur de mode de fonctionnement dans l'objet de base DIO est différent de zéro, les routes descendantes de la racine aux noeuds sont prises en charge et doivent être maintenues. Dans ce cas, chaque poste client doit envoyer un message de contrôle DAO mono diffusion pour déterminer les informations de route inverse. Lors du retour à la racine DODAG, les noeuds visités sont enregistrés dans le paquet le long de la route ascendante, et la route complète est alors établie entre la racine DODAG et le noeud client. RPL spécifie deux modes d'opérations pour maintenir les routes descendantes dans une instance RPL :

1) Mode de stockage : en mode de stockage, un message DAO est envoyé en mono diffusion par l'enfant au parent sélectionné, qui est capable de stocker les messages DAO reçus par ses enfants avant d'envoyer le nouveau message DAO avec des informations d'accessibilité agrégées à son parent. Le mode de stockage peut activer ou désactiver le mode de multidiffusion.

2) Mode sans stockage : en mode sans stockage, le message DAO est envoyé en monodiffusion à la racine DODAG, ainsi, les parents intermédiaires ne stockent pas les messages DAO, mais insèrent uniquement leurs propres adresses dans la pile de routes inversées dans le DAO reçu message, puis le transmet à son parent.

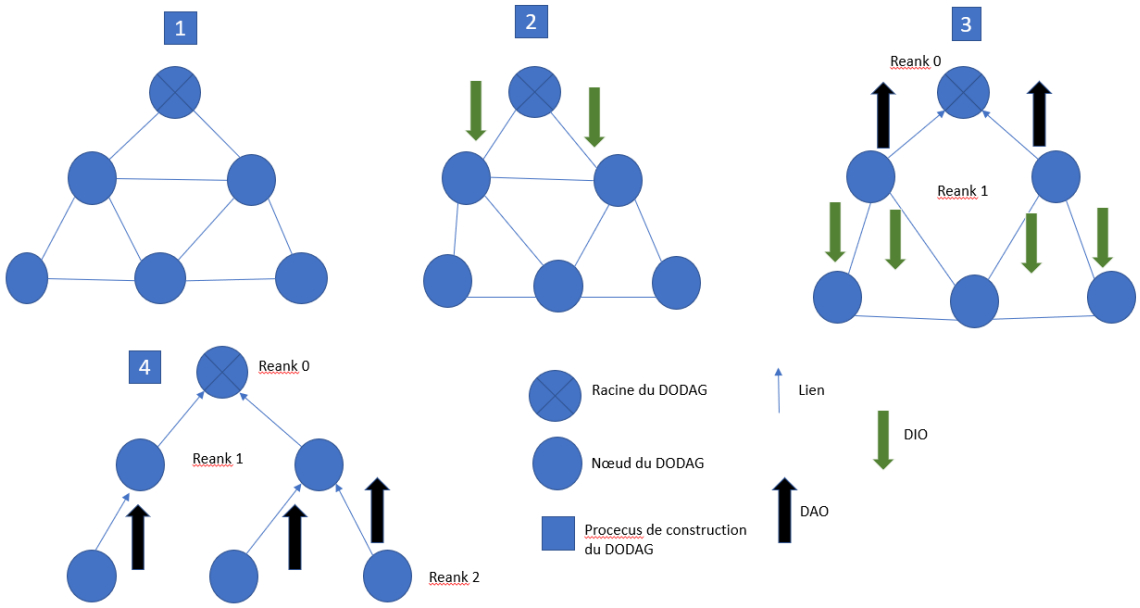

FIGURE 2.3 – Le processus de construction de DODAG

3.4 La maintenance de la topologie

En cas de perte d'un lien, le DODAG peut être maintenue de deux façons :

1) Réparation globale : La racine initie une reconstruction complète du DODAG avec des messages DIO, des numéros de séquences sont utilisés pour différentier les DODAG ancien et nouveau. Cette réparation couteuse en trafic.

2) Réparation locale : Recherche d'un nouveau parent par le noeud affecté. Dans ce cas, le DOGAG n'est plus optimal, seule une réparation globale permettra une nouvelle optimisation

3.5. Paradigmes de communication

RPL prend en charge trois paradigmes de communication : Multi-Point-to-Point (MP2P), Point-à-Multi-Point (P2MP) et le Point à point (P2P)

1) Multi-Point-to-Point Operation RPL prend en charge le trafic multipoint à point qui se rapporte au trafic de collecte de données acheminé dans le sens de la route ascendante à partir de plusieurs noeuds vers la racine DODAG. Le trafic de collecte de données est appelé trafic unicast entrant. Les destinations de MP2P sont principalement des routeurs frontaliers qui jouent un rôle important dans le réseau et fournissent une interface pour

la connectivité avec Internet. Les destinations des flux MP2P sont des noeuds désignés qui ont une certaine importance pour l'application, tels que la fourniture de connectivité à l'Internet ou au réseau IP privé principal.

2) Point-to-Multipoint Operation RPL définit également l'opération point à multipoint, qui représente le trafic transmis dans le sens de la route descendante de la racine vers plusieurs noeuds.

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

4. La fonction objective (OF)

Une fonction objective (OF) permet comment les noeuds sélectionnent et optimisent les chemins dans une instance RPL en association avec des métriques de routage et contraintes. La fonction objectif est indiquée dans le message DIO à l'aide d'un Objective Code Point (OCP), et il indique la méthode qui doit être utilisé pour construire le DODAG. Elle définit aussi comment les noeuds traduisent une ou plusieurs métriques et contraintes, pour les buts de, calculer le rang de chaque noeuds dans le DODAG,et comment les noeuds sélectionnent les parents. Le groupe de travail ROLL a défini deux fonctions objectives dans le RPL originale, cette séparation destinée à permettre à RPL d'être adaptée pour répondre aux critères d'optimisation différents, d'applications et de conceptions de réseaux : OF0(Objective Function Zero) et MRHOF (Minimal Rank with hystérésis Objective Function)

4.1 La fonction OFO

OFO est une fonction objective pour RPL qui indique le résultat du processus utilisé par un nœud RPL pour sélectionner et optimiser les itinéraires dans une instance RPL en fonction de

Constante	Valeur
DEFAULT_STEP_OF_RANK :	**3**
MINIMUM_STEP_OF_RANK :	**1**
MAXIMUM_STEP_OF_RANK :	**9**
DEFAULT_RANK_STRETCH :	**0**
MAXIMUM_RANK_STRETCH :	**5**
DEFAULT_RANK_FACTOR :	**1**
MINIMUM_RANK_FACTOR :	**1**
MAXIMUM_RANK_FACTOR :	**4**

TABLE 2.2 – Les constantes et leurs valeurs

la DIO. Dans cette OF, la métrique d'acheminement adoptée est le nombre de sauts. Elle est utilisée par une instance RPL pour calculer un rang pour le périphérique. La raison principale de sa conception est de permettre l'interopérabilité entre les implémentations diffèrent de RPL. Les caractéristiques principales de OFO sont :

1) Opérations OF0 : les opérations OFO comprennent **le classement de calcul** qui consiste à calculer une variable appeler step_of_rank associé à un parent donné à partir d'un lien pertinent des propriétés et métriques, **la sélection des parents** préférés ou du successeur réalisable de sauvegarde.

2) Les variables : **step_of_rank** qui est l'intermédiaire de calcul basé sur les propriétés du lien avec un certain voisin et le **rank_increase** qui est l'entrée entre le rang du parent préféré et le déclencheur.

3) les constantes sont utilisées lors du configuration ou l'implémentation avec leurs valeurs initiales voir (**Table 2.1**).

4.2 MRHOF(The Minimum Rank with Hysteresis Objective Function)

MRHOF est une fonction objective pour RPL qui utilise l'hystérique tout en sélectionnant le chemin avec la plus petite valeur métrique. Dans cette OF, la métrique utilisée est déterminée dans le conteneur de métrique DIO. Généralement c'est le nombre de retransmission attendue (ETX) qui est utilisé avec une hystérique pour éviter les différences minimum de rang. Cette métrique permet à RPL de trouver les chemins stables à partir des noeuds vers une racine. En

l'absence d'une métrique dans le conteneur métrique DIO, MRHOF utilise par défaut ETX. Les caractéristiques du MRHOF sont :

1) calcul du coût de chemin entre deux nœuds(racine ou non).

2) sélection des parents si le coût du chemin d'un voisin candidat ou parent préféré change.

3) Le classement de calcul qui consiste lorsqu'un nœud non racine sélectionne son ensemble parent, puis il converti le coût du chemin d'un parent en une valeur de rang.

4) La variable cur_min_path_cost est le coût du chemin depuis un nœud via son parent préféré jusqu'à la racine calculé lors de la dernière sélection de parent.

5) Les paramètres sont : MAX_LINK_METRIC MAX_PATH,_COST, PARENT_SWITCH_THRESHO
PARENT_SET_SIZE
et ALLOW_FLOATING_ROOT dont chacun contenant un entier positif qui répresente soit le valeur max autorisé d'un métrique ou soit un coût d'un chemin,...

5 L'algorithme Trickle Timer

L'algorithme Trickle Timer est utilisé par RPL pour réduire la surcharge des messages de contrôle en ne transmettant que les mises à jour lorsque des incohérences sont détectées dans le réseau. Si un nœud entend des mises à jour DIO de ses voisins qui sont cohérentes avec sa propre compréhension de la topologie de réseau, un compteur de redondance est incrémenté. Si le nombre de mises à jour cohérentes entendues dans un intervalle de temps particulier dépasse le nombre de redondances, le noeud ne transmet aucune mise à jour et la période d'écoute est doublée. Toutefois, si une mise à jour incohérente est entendue, Trickle Timer est réinitialisé et une mise à jour est rapidement propagée.

5.1 Les paramètres et variables

L'algorithme Trickle fonctionne comme une minuterie d'où son fonctionnement est pour un temps défini. Les paramètres de configuration de Trickle sont :

1) **Imin** c'est la taille minimale de l'intervalle de transmission

2) **Imax** c'est la taille maximale de l'intervalle de transmission,

3) **K** entier positif, c'est la constante de redondance,

Les variables de Trickle sont :

1) **I** la taille de l'intervalle actuelle,

2) **t** l'heure de l'intervalle actuelle,

3) **c** le compteur

5.2 Les règles de Trickle

L'algorithme Trickle Timer est basée sur les six règles suivantes :

1) L'algorithme commence le premier intervalle après le démarrage de l'exécution en définissant la valeur de **I** dans le plage **[Imin-Imax]**,

2) Lorsqu'un intervalle commence, l'algorithme réinitialise **c** à 0 et définit **t** à un point aléatoire dans l'intervalle du plage **[I/2,I]**.

3) Chaque fois trickle entend un transmission cohérente, le conteur **c** est incrémenter.

4) Si le conteur **c** est inférieur au constante de redondance **k** au temps **t**, alors Trickle émet.

5) Lorsque l'intervalle **I** expire, Trickle double la longueur de l'intervalle.

6) Si Trickle entend une transmission incohérente et que **I** supérieur à **Imax**, le minuteur de maintien est réinitialisé, sinon **I** est redéfini sur le **Imin** pour le démarrage d'un nouvel intervalle avec la reinitialisation du timer.

6 Sécurité dans RPL

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

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

2) Préinstallé : Dans ce mode, les noeuds joignant une instance RPL ont des clés préinstallées permettant de générer et de traiter les messages sécurisés.

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

6 Conclusion

RPL est un protocole de routage développé par le groupe de travail ROLL pour répondre aux contraintes très spécifiques des réseaux LLN. Le protocole a été conçu pour être très flexible aux différentes variations des ressources du réseau. Sa fonction de construire des routes s'appuie sur la notion de DODAG. Chaque noeud calcule son rang et choisit des parents qui minimisent le coût sur la route vers la racine. Les coûts sont calculés selon une ou plusieurs métriques, le choix de ces métriques dépend de l'application et de ces objectifs/contraintes. Le RPL original tel que défini dans la RFC 6550 ne peut pas répondre à tous les besoins spécifiques des applications.

CHAPITRE

3

ESTIMATION DE LA QUALITÉ DES LIENS DANS RPL

1 Introduction

L'estimation de la qualité des liaisons est cruciale dans les réseaux de capteurs sans fil par contre, elle est une fonctionnalité essentielle pour les réseaux sans fil à expédition multiple. L'estimation précise de la qualité des liaisons est une condition préalable à l'optimisation du routage. En règle générale, la plupart des protocoles de routage multi-sauts pour les réseaux de capteurs visent à sélectionner des chemins du réseau qui minimisent le nombre total des transmissions ou retransmission nécessaires pour livrer un message. Sans une estimation précise de la qualité de lien, le routage ne peut agir qu'à l'aveuglette, sélectionnant ainsi des chemins qui pourraient être non optimal.

Dans RPL, les procédures d'estimation la qualité des liens ne sont pas spécifiées par la norme, ce qui veut dire que différentes implémentations peuvent adopter différents mécanismes. L'approche classique adoptée par la plupart des protocoles de routage pour les réseaux sans fil

multi-sauts est l'utilisation des messages de routage diffusés périodiquement pour la construction et la maintenance de la topologie pour surveiller la qualité des liens et recueillir des mesures de liens. Comme dans RPL la transmission des messages de routage n'est pas périodique mais irrégulière et imprévisible en raison de l'utilisation de minuteries, cette approche n'est pas applicable.

Pour ce fait, dans les sections suivantes nous essayerons d'étaler les travaux littéraires qui estiment la qualité de lien dans RPL en partant de la solution fondamentale qui est l'utilisation des paquets de sonde unicast qui sont envoyés sur un lien spécifique pour mesurer sa qualité en suite l'approche d'utilisation de la mesure du nombre estimatif de transmission(ETX)et en fin l'estimation de la qualité de lien basée sur le Trickle.

2 L'utilisation des paquets de sonde unicast

l'utilisation de paquets de sonde unicast qui sont envoyés sur un lien spécifique pour mesurer sa qualité est l'approche pour le protocole RPL qui l'alternative au celle d'exploitation du trafic de signalisation de routage pour les protocoles de routage pour les réseaux sans fil mulisauts. Hors mis sa capacité à estimer la qualité de lien avec précision, elle entraine un retard supplémentaire au moment de la formation du réseau, des charges générales supplémentaires en tant que trafic de sonde et une compléxité supplémentaire de la mise en œuvre. Compte tenue de ce retard supplémentaire, la surveillance des liaisons passives a été largement utilisée dans les WSNs pour minimiser les retards généraux et la complexité. Les principaux avantages de cette solution sont sa simplicité c'est-à-dire qu'aucune modification du protocole d'origine n'est requise, et les petits retards généraux. C'est pourquoi la plupart des implémentations des prototypes RPL adoptent une telle technique d'estimation des liaisons, et la qualité des liaisons est mesurée par la collecte de statistiques sur les transmissions réussies et les défaillances pour les liaisons utilisées par le trafic de données. Cependant, cette approche n'évalue que les liens qui sont actuellement utilisés ce qui fait que cette approche peut être de fois trop conservatrice.

Cependant, s'il y ait de nombreux voisins mal connectes dans les grandes WSNs, la connaissance partielle de la qualité des liens avec les voisins fournit par cette approche est particulièrement critique.En outre, la détection de meilleurs transitaires alternatifs est difficile car les nouveaux voisins ne sont testés qu'en cas de défaillance de la liaison.

En effet, les procédures d'estimation des liaisons pilotées et de la maintenance de la topo-

logie de RPL entraînent une instabilité des itinéraires et une réduction des taux de livraison des paquets dans les prototypes de RPL existants. Pour résoudre ces problèmes, une autre étude a proposé une méthodologie d'évaluation de la qualité des maillons légers qui exploite les caractéristiques standard de RPL. L'objectif de cette proposition a été de fournir une estimation précise de la qualité des liens avec un minimum de modifications aux implémentations RPL existantes afin de faciliter l'adoption sur les appareils réels. Cette méthodologie est détaillée dans la section 4.

De la qualité des maillons légers L'évaluation de la qualité des maillons légers qui exploite les caractéristiques standard de RPL est caractérisée essentiellement par la sélection dynamique de la surveillance des liens basée sur les données et la surveillance des liens actifs basée sur la diffusion en fonction de l'état du routeur RPL. On observe aussi la régulation faites par les minuteries de transmission basées sur Trickle, l'émission de paquets de sonde de diffusion pendant les phases de sonde actives. Ainsi le temps nécessaire à la formation de la topologie est minimisé par l'exploration et la construction topologique effectuées simultanément.

3 Les travaux connexes

Un aspect fondamental de l'estimation de la qualité des liens est la métrique utilisée pour mesurer la qualité d'un lien et/ou chemin. Cependant, il existe deux types de métriques de lien : les métriques de la couche physique, qui mesurent les attributs directement liés à la radio émetteur-récepteur (par exemple, RSSI ou SNIR), et les métriques de la couche liaison (par exemple, nombre de retransmissions de paquets). Dans ce travail, nous nous concentrons sur l'étude faites sur ce dernier type de métriques pour estimer la qualité de lien. Il existe trois approches pour la surveillance des liens dans les WSN : active, passive et hybride.

3.1 L'approche active

Dans cette approche, les nœuds surveillent la qualité de leurs liens en envoyant des paquets de sonde aux voisins. Les paquets de sonde peuvent être encapsulés dans des trames soit en multi-diffusion[15], soit en mono-diffusion[17]. De ce fait, les paquets de sonde sont généralement transmis avec une périodicité donnée. Ces approches ont été démontré qu'ils produisent des mesures précises de la qualité des liaisons si les sondes sont transmises avec une fréquence suffisamment élevée. Cependant, la surcharge associée à la détection unicast peut être excessif

en particulier dans les WSN. Ainsi, la plupart des protocoles réseau dans les WSN utilisent des sondes de diffusion, qui entraînent une plus petite surcharge par rapport au sondage unicast. Un autre avantage de la diffusion du sondage est la possibilité d'exploiter le trafic de signalisation de routage, qui est déjà nécessaire pour implémenter la plupart des fonctionnalités de routage.

3.2 L'approche passive

Dans cette approche, à la place, exploite le trafic existant sans frais généraux de palpage supplémentaires[15]. À cette fin, un noeud peut utiliser son propre trafic de données, c'est-à dire les paquets qu'il transmet, ou peut entendre les transmissions qui ne lui sont pas adressées pour évaluer qualité des liens avec ses voisins. Cette méthodologie a largement adopté dans les réseaux WSN car il est économe en énergie car il n'implique pas l'émission de paquets supplémentaires. Par contre ont montré que sur entendre consomme d'énergie significative au niveau des récepteurs, ce qui peut annuler l'économie d'énergie grâce à l'absence de sonde. Un autre inconvénient majeur de ces approches passives sont qu'elles dépendent de la présence de données circulation. Cela peut entraîner des inexactitudes de mesure en fonction sur l'intensité du trafic de données et sa répartition spatiale.

3.3 L'approche hybride

Cette approche combinent à la fois les approches actifs et passifs afin d'atteindre un équilibre pratique entre mesures de liaison précises et efficacité énergétique. Dans la phase initiale lorsque le la qualité de la liaison est inconnue, un noeud attribue une estimation optimiste pour la qualité des liens avec les voisins nouvellement découverts afin de faire pivoter la sélection du parent préféré entre tous les noeuds.[17]

4. Solution implémenté : Tricle-L²

Selon la spécification ou implémentation du RPL, chaque noeud non racine à l'heure de l'amorçage du réseau écoute les messages DIO. Lorsqu'au moins un DIO est reçu, le noeud rejoint immédiatement le DODAG : il sélectionne d'abord un noeud comme parent préféré parmi ses voisins et évalue son rang, puis commence à diffuser ses propres DIOs pour annoncer sa présence. Au moment de la réception du premier DIO, la qualité du lien est encore inconnue car sans utiliser le mécanisme de sondage, un noeud peut mesurer la qualité d'un lien uniquement

lorsque le trafic de données est acheminé sur ce lien. Pour cette raison, lorsque le premier DIO est reçu, le noeud ignore la qualité du lien avec le voisin qui avait envoyé ce message DIO et il sélectionne le parent préféré presque aveuglément en initialisant la qualité du lien à un arbitraire valeur qui est égal à 1[2].

Après avoir rejoint le DODAG, un noeud peut effectuer une estimation précise du lien vers son parent préféré par collecte des statistiques pour les paquets de données circulant sur ce lien. Cependant, jusqu'à ce qu'une estimation précise soit obtenue, le réseau reste dans une phase transitoire dans laquelle la topologie est instable car les noeuds changent fréquemment de leurs parents préférés. Ce problème souligne la nécessité d'une mesure de liaison plus précise au moment de la formation de DODAG afin d'améliorer la capacité de construire rapidement une topologie de réseau stable et optimale. La solution dont nous avons implémenté pour l'évaluer, surmonte ce problème en introduisant un sondage phase qui permet aux noeuds de collecter une qualité de liaison précise d'estimation avant de rejoindre le DODAG. Plus précisément, chaque noeud exécute à son tour sa propre phase de sondage en envoyant une diffusion messages de sonde, tandis que les noeuds à portée de transmission du noeud de diffusion exploitent ces messages pour estimer la qualité du lien avec ce noeud. La chronologie des phases réalisées par chaque noeud sont :

1) Amorçage(Bootstrap), le noeud recueille des informations sur ses voisins,

2) sondage, le noeud rejoint le DODAG et émet des messages de sonde pour permettre à ses voisins de mesurer la qualité des liens,

3) normal, le noeud a rejoint le DODAG et exécute régulièrement les opérations décrites dans la norme.

4.1. Bootstrap

Tel que défini dans la norme RPL, un noeud commence à partir d'un état d'amorçage, dans lequel il écoute les DIO. Les critères qui conduire un noeud pour quitter l'état d'amorçage pour rejoindre le DODAG n'est pas spécifié et la spécification RPL ne requiert que pour découvrir au moins un voisin. Dans cette solution, nous présenterons la solution qui consiste à introduire une minuterie pour vérifier périodiquement l'état d'un ensemble de conditions, appelées conditions de jointure, et uniquement lorsqu'au moins une de ces conditions est satisfaite, le noeud peut

quitter la phase de Bootstrap pour passer à la phase de Probing. Plus précisément, au réception du premier DIO une minuterie appelée minuterie de jointure est définie comme suit :

$T\,join = U + ð$

où U est la période sur laquelle un noeud doit vérifier les conditions de jointure, tandis que ǫ est une variable aléatoire uniforme, qui représente un back-off aléatoire introduit pour éviter les noeuds dans le même voisinage pour démarrer l'exécution de la procédure simultanément, d'où le soi-disant problème de tempête de diffusion. Le paramètre phi peut être réglé pour parvenir à un compromis entre le retard pour la construction de la topologie, l'exhaustivité des informations et la consommation d'énergie grâce à l'estimation du lien. Notons que lorsque le minuteur expire, si aucune des conditions de jointure ne sont satisfaits que le noeud prolonge la phase de Bootstrap en réinitialisant la minuterie tjoin. Les conditions de jointure définies dans cette implémentation sont les suivant : qualité de liaison minimale et nombre maximal de minuteries réinitialise. La première condition exige que les liens découverts pendant la phase d'amorçage ont une qualité minimale, c'est-à-dire que la qualité du lien via le parent préféré du candidat, disons Lcpp, doit être supérieur au seuil de qualité de liaison minimum, disons Lth. La deuxième condition introduit un nombre maximum de fois Nmax, le temporisateur de jointure peut être réinitialisé. Plus précisément, lorsque le nombre de réinitialisations Nreset est égal à Nmax, le noeud sort la phase d'amorçage quelle que soit la qualité de liens découverte. Cette condition est nécessaire pour garantir un maximum de temps de séjour dans la phase d'amorçage, qui ne peut pas être plus long que (Nmax +1) * (U + ǫ/2). Les paramètres U et Nmax peut être réglé conjointement pour représenter le temps maximum pendant lequel un noeud est disposé à retarder son statut opérationnel pour obtenir de meilleurs itinéraires.

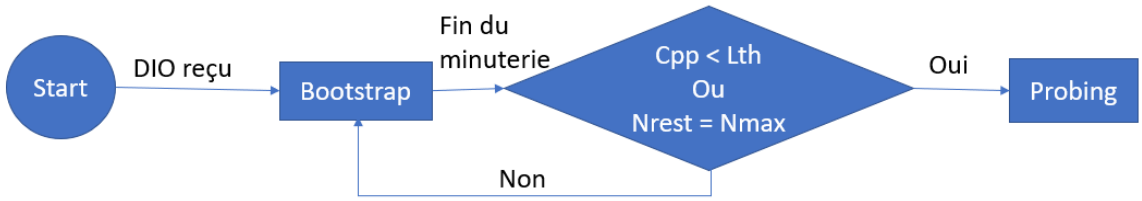

FIGURE 3.1 – Le bootstrap

4.2. Probing

Lorsque le noeud quitte la phase d'amorçage, il rejoint d'abord le DODAG en sélectionnant un parent préféré et en calculant sa valeur de rang selon la fonction objective(OF) adopté. Ensuite, le noeud démarre la phase de détection active au cours de laquelle il diffuse les messages de sonde. Plus précisément, lors de la phase active un noeude diffuse un ensemble de messages de sonde Np, avec l'objectif de permettre aux voisins d'évaluer la qualité de la liaison.

4.3. Normal

Lorsque le train de messages diffusés est terminé, le noeud peut entrer dans l'état normal dans lequel les opérations RPL normales sont effectué.

4.4. DIO de sonde

Afin de garantir que cette technique d'estimation de lien peut être facilement adopté dans la norme RPL, une version légèrement modifiée des messages DIO (appelés sondes DIO) est adopté comme diffusion sonde. Plus précisément, les DIO de sonde ont la même structure que DIO standard à l'exception des modifications suivantes :

1) Indicateur de sonde : ajouté au champ indicateur du message, s'il est défini indique que la DIO appartient à un ensemble de DIO de sonde.

2) Numéro de séquence de la sonde : le numéro de séquence du message dans l'ensemble des DIO de sonde diffusé par un noeud. Ce champ est ajouté dans le champ "Réservé" de le message. Le numéro de séquence est utilisé pour calculer le nombre de DIO de sonde perdues.

Enfin, il convient de souligner que les sondes DIO sont entièrement rétrocompatible avec le standard RPL, c'est-à-dire un routeur RPL ne peux pas implémenter les fonctionnalités de sondage mais peut traiter le message DIO, collecte des informations de routage et élimine la sonde informations. L'émission des sonde DIO est régulée par le Trickle algorithme, le même algorithme adopté par RPL pour planifier l'émission de messages DIO pendant les opérations normales. L'adoption de minuteries Trickle pour réguler également l'émission des sondes DIO représentent les avantages suivants : la procédure de sonde peut être facilement mis en oeuvre car il ne nécessite pas de supplément complexité, la politique de suppression de Trickle peut

```
struct rpl_dio_sonde {
  uip_ipaddr_t dag_id;
  rpl_ocp_t ocp;
  rpl_rank_t rank;
  uint8_t grounded;
  uint8_t mop;
  uint8_t preference;
  uint8_t version;
  uint8_t instance_id;
  uint8_t dtsn;
  uint8_t dag_intdoubl;
  uint8_t dag_intmin;
  uint8_t dag_redund;
  uint8_t default_lifetime;
  uint8_t indication;
  uint8_t n_sequence;
  uint16_t lifetime_unit;
  rpl_rank_t dag_max_rankinc;
  rpl_rank_t dag_min_hoprankinc;
  rpl_prefix_t destination_prefix;
  rpl_prefix_t prefix_info;
  struct rpl_metric_container mc;
};
typedef struct rpl_dio_sonde rpl_dio_sonde_t;
```

FIGURE 3.2 – La structure du DIO de sonde

être exploitée pour réduire la probabilité de collision des messages de sonde dans chaque voisinage, de sorte que les noeuds exécutant des opérations RPL régulières et les noeuds diffusant des messages de sonde peuvent coexister en même temps.

Quel que soit son état actuel, chaque noeud traite en arrière-plan les messages DIO pour collecter des informations sur la topologie. Si l'indicateur de sonde est défini, le numéro de séquence est également enregistré. Lorsque la dernière sonde DIO d'un noeud est reçue, le node évalue le nombre de messages perdus et par conséquent mesure la qualité de la liaison en termes de taux de réception des paquets (PRR). Afin de gérer le cas où le dernier message de sonde est perdu, le noeud définit une minuterie Tprobe qui est réinitialisée à chaque fois une sonde DIO est reçue. Lorsqu'une estimation de la qualité de la liaison est terminé le coût du voisin est mis à jour. Si nécessaire un le nouveau parent préféré est sélectionné.

6 Conclusion

Dans cet chapitre, nous avons présenté un mécanisme léger d'estimer de la qualité des liens dans les WSN adoptant RPL comme protocole de routage. Cet approche exploite le Trickle algorithme, proposé à l'origine dans RPL pour réguler l'inondation des messages de signalisation de routage, pour contrôler également la transmission de messages de sonde de diffusion au mo-

ment de configuration de la topologie. Afin d'assurer la rétro compatibilité de cette technique d'estimation de lien proposée, tels que les paquets de sonde sont basés sur des messages DIO réguliers. Cela permet également d'évaluer simultanément la qualité du lien tout en diffusant informations de routage.

CHAPITRE

4
SIMULATIONS ET RÉSULTATS

1 Introduction

Le protocole RPL présenté dans le chapitre précédent est l'un des protocoles standards de communication. Ce protocole a été largement utilisé dans les réseaux de capteurs sans fil et l'internet des objets puisqu'il est adapté aux systèmes à ressources limitées et qui présentent certaine hétérogénéité. Dans ce chapitre, nous évaluons le protocole RPL selon certaines métriques de routage : ETX et l'énergie consommée. Puis, nous proposons une métrique de routage pour améliorer les performances de RPL. Cette métrique combine les deux métriques de routage ETX et l'énergie consommée. Pour ce faire, nous avons utilisé des outils logiciels spéciaux pour les systèmes à ressources limitées tels qu'un système d'exploitation léger (Contiki) et le simulateur COOJA.

2. Outils de la simulation

2.1. Contiki

Contiki OS est un système d'exploitation open source léger conçu pour l'Internet des Objets. Il a été développé à l'Institut suédois des sciences de l'informatique par Adam Dunkels et est écrit dans la langue de programmation C. Contiki est un système d'exploitation hautement portable et il a déjà été porté sur plusieurs platesformes fonctionnant sur différents types de

processeurs. La plupart des plates-formes utilisent le processeur Texas Instruments MSP-430 ainsi que la série de microcontrôleurs Atmel ATmega.[13]

Le principal avantage de la Contiki est qu'il fonctionne sur un concept qui se situe entre le multi-threading et la programmation événementielle, cela permet aux processus de partager le même contexte d'exécution et donc d'améliorer l'utilisation de la mémoire et de l'énergie. C'est le concept des Protothreads. Contiki prend en charge les implémentations de pile IPv6 et IPv4, ainsi que les normes sans fil peu avancées comme 6lowpan, RPL, CoAP ou encore la pile Rime. Il s'agit d'une pile de communication légère pour les réseaux de capteurs et possède des couches plus petites que les piles traditionnelles. Ce sont des couches simples qui ont de petits en-têtes (seulement quelques octets). Rime prend également en charge la réutilisation du code et le but principal de ce protocole est de simplifier la mise en oeuvre des réseaux de capteurs. [13]

2.2. Les propriétés de Contiki

Les propriétés suivantes, ont permis l'évolution de Contiki :

- Normes Internet : Contiki fournit des communications Internet de faible puissance. Contiki et il prend en charge IPv6 et IPv4 avec des normes sans fil de faibles puissances : 6LoWPAN, RPL, CoAP.

- Développement rapide : Avec Contiki, le développement est facile et rapide , les applications à base de Contiki sont écrites en C standard. En outre, ces applications peuvent être émulées avec le simulateur Cooja avant d'être injectées sur des capteurs réels. Il existe aussi l'environnement Instant Contiki qui fournit un environnement de développement.

- Une sélection du matériel : Contiki fonctionne sur une gamme d'appareils sans fil de faible puissance tels que la famille des Mica et la famille des Telos.

- Communauté active : Contiki est développé par une équipe mondiale de développeurs avec des contributions d'Atmel, Cisco, ETH, Redwire LLC, SAP, Thingsquare, et beaucoup d'autres, menés par Adam Dunkels de Thingsquare.

- Logiciel Open Source : Contiki est un logiciel open source : Contiki peut être utilisé

librement tant dans les systèmes commerciaux et non commerciaux et le code source complet est disponible.

❖ Support commercial : Contiki fournit à la fois le soutien de la communauté des développeurs Contiki et support commercial.

2.3. Les caractéristiques de Contiki

Contiki se caractérise par :

❖ Allocation de mémoire : Contiki est conçu pour les systèmes légers. L'espace mémoire utilisé par le système d'exploitation et par l'application doit être suffisamment faible pour être contenu dans la mémoire du capteur. Une configuration typique de Contiki (le noyau et le chargeur de programmes) consomme 2 Koctets de RAM et 40 Koctets de ROM.

❖ Consommation d'énergie : L'énergie est souvent apportée par une batterie qui est généralement non rechargeable. Les recherches scientifiques explorent les possibilités de réduire la consommation d'énergie dans capteurs. Le module radio qui est chargé de la transmission et la réception, est l'élément qui consomme plus d'énergie. Pour cela, le module radio devrait être mis en mode actif que s'il a des données à transmettre ou à recevoir sinon il devrait passer en mode veille. Cependant, lorsque le module radio est mis en mode veille, le capteur ne reçoit pas les messages qui lui sont destinés. En outre, un réveil périodique risque d'être inutile, et donc de consommer de l'énergie de façon inefficace. Pour gérer cette problématique, Contiki propose par défaut ContikiMAC, un mécanisme conçu pour rester en communication avec le réseau efficacement, tout en permettant la mise hors tension du module radio 99permettent de limiter la consommation telle que le compactage des données transférer, le pré-calcul (afin de ne transmettre que les données réellement utiles), mais aussi une optimisation du routage.

❖ Portabilité : la portabilité consiste à adapter le système d'exploitation aux différents types de capteurs, selon les éléments électroniques les constituant. Contiki est complètement écrit en langage C. Ce langage de programmation est le langage le plus répondu pour la programmation des systèmes. La portabilité de Contiki concerne les plateformes suivantes : Z1, WISMOTE, IRIS, MICAZ, SKY, TELOS.

- ❖ Interopérabilité : l'interopérabilité d'un capteur est le fait de pouvoir communiqué avec les capteurs gérés par un système d'exploitation différent. Adams Dunkels de l'équipe scientifique suédoise présente dès 2003 uIP et IwIP permettant d'implémenter le protocole IP sur les systèmes limités en ressources tels que les capteurs. Jusque là, les capteurs utilisaient des protocoles de communications propriétaires ou alors des adaptations d'IP permettant le fonctionnement des applications mais sans offrir toutes les fonctionnalités du protocole IP. Dès la présentation de Contiki en 2004 uIP et IwIP étaient disponibles. De se fait, les applications exécutées sur Contiki pouvaient dialoguer vers n'importe quel matériel supportant le protocole IP. L'arrivée de IPv6 est uIPv6 sur Contiki apporte une nouvelle interopérabilité vers les matériels supportant ce protocole. Le support de 6LoWPAN permet à Contiki de communiquer avec les matériels via un réseau sans fil suivant la norme 802.15.4. Contiki est réputé pour être un système d'exploitation robuste et mature, fournissant IPv4 et IPv6 pour les réseaux de capteur sans fil. Selon une étude publiée en 2011, comprenant des tests d'interopérabilité entre des capteurs sous Contiki et d'autre sous TinyOS, l'interopérabilité est bien au rendez-vous, mais des efforts sont à déployer pour mesurer et améliorer les performances de la couche réseau.

2.4. Architecture de Contiki

Contiki est développé en langage C et il est constitué d'un noyau, de bibliothèques, d'un ordonnanceur et d'un jeu de processus. Comme tout système d'exploitation, son rôle est de gérer les ressources physiques telles que le processeur, la mémoire et les périphériques informatiques (d'entrées/sorties) (voir figure 3.1) Il fournit ensuite aux applications des interfaces permettant d'utiliser ces ressources. Conçu pour les modules de capteurs sans fil il occupe peu d'espace en mémoire et permet une consommation électrique très faible.[13]

2.5. La connectivité dans Contiki

Contiki offre deux types de connectivité :

1) 1. La couche uIP : Contiki fournit une implémentation de la pile du protocole TCP/IP pour les petits microcontrôleurs de 8 bits. uIP n'exige pas de ses pairs pour avoir une pile complète de protocoles, mais il peut communiquer avec des pairs exécutant une pile légère similaire. La mise en œuvre uIP est écrite en C et elle a l'ensemble minimal de

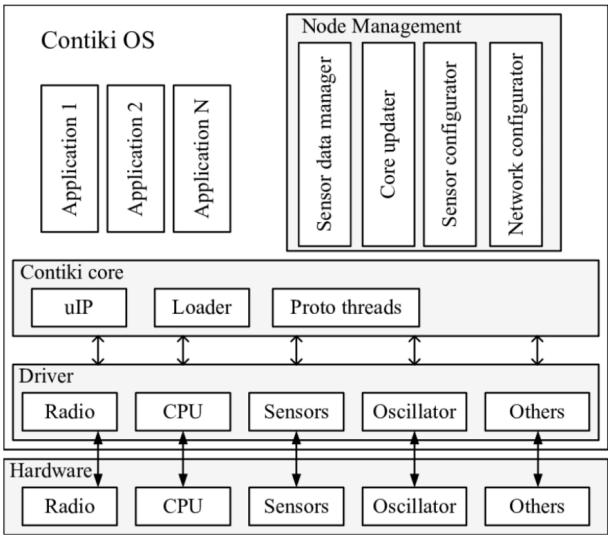

FIGURE 4.1 – L'architecture de ContikiOS

fonctionnalités nécessaires pour une pile complète TCP/IP. uIP ne peut soutenir qu'une interface réseau, et il prend en charge TCP, UDP, ICMP, et des protocoles IP.

L'implémentation traditionnelle de TCP/IP a exigé plus de ressources à la fois en termes de taille du code et utilisation de la mémoire pour être utile dans les petits systèmes 8 ou 16 bits. La taille du code de quelques centaines de Koctets de RAM et les exigences de plusieurs centaines de Koctets ont rendu impossible pour s'adapter à la pile TCP/IP complète dans les systèmes avec quelques dizaines de Koctets de RAM.

La pile uIP peut être exécutée en tant que tâche dans un système multitâche, ou comme un programme principal dans un système multitâche unique. Dans les deux cas, la boucle principale de contrôle fait deux choses à plusieurs reprises :

- ❖ Vérifier si un paquet est arrivé à partir du réseau

- ❖ Vérifier si un délai périodique s'est produit.

2) La couche Rime : D'autre part, Contiki supporte la couche Rime qui est une autre pile de protocoles en couches pour la communication en réseau. Rime régit au principe du meilleur effort et la transmission fiable.

   La pile Rime supporte à la fois les primitives de communication à un saut ou multisauts. Les primitives pour les communications multi-sauts ne précisent pas comment les paquets sont acheminés à travers le réseau. Comme le paquet est envoyé sur le réseau, l'application ou le protocole de la couche supérieure est invoqué au niveau de chaque nœud pour

qu'il choisisse le nœud voisin qui relaye le paquet. Il est ainsi possible de mettre en œuvre des protocoles de routage arbitraires au dessus des primitives à bande multiples.

Les protocoles ou les applications s'exécutant au dessus de la pile Rime peuvent mettre en œuvre des protocoles additionnels qui ne sont pas dans la pile Rime. Si un protocole ou application s'exécute au-dessus de la pile Rime, il aura besoin de primitives de communication qui ne sont pas actuellement dans la pile Rime.

3 Le simulateur Cooja

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

3.1. La fenêtre Timeline

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

3.2. La fenêtre Network

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

3.3. La fenêtre Mote Output

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

3.4. La fenêtre Notes

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

Champ	Valeur
Processeur :	Intel(R) Core(TM) i5-7300U CPU @ 2.60GHz 2.70 GHz
RAM :	8,00 Go (7,88 Go utilisable)
Disque dur :	256 Go
Système d'exploitation :	64 bits, processeur x64

TABLE 4.1 – Les caractéristiques du PC

3.5. La fenêtre Simulation control

C'est ou nous pouvons lancer , mettre en pause et charger de notre simulation.

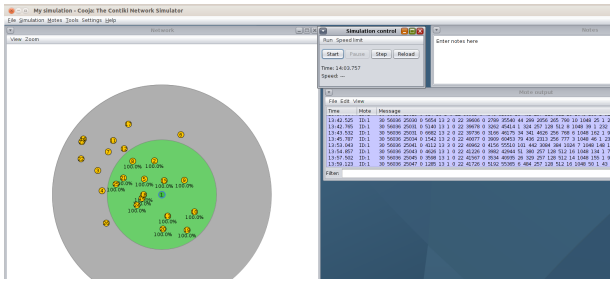

FIGURE 4.2 – Fenêtres de simulations Cooja

4. Environnement de développement

Pour estimer la qualité des liens dans RPL, nous avons utilisé Contiki 3.0, langage « C » pour implémenter la solution. La machine utilisée dans la simulation du réseau est caractérisée par les paramètres suivants :

5. La simulation

5.1. Les paramètres de la simulation

Dans notre simulation, nous avons utilisé l'exemple de rpl-collect (contiki/examples/ipv6/rplcollect) et avec les paramètres de simulation comme montre le tableau :

5.2. Métriques de la simulation

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

Paramètres	Valeur
Simulateur :	Cooja
Nombre de noeuds :	5,10,25,50,100
Nombre de noeuds racines :	1
Identifiant du noeud racine :	1
Surface (mètres) :	300 X 300
La fonction objective :	MRHOF(par défaut)
Topologie :	grille ordonnée, aléatoire
Mote types :	Sky mote
Radio Environment :	fixe et mobile

TABLE 4.2 – Les paramètres de la simulation

suivantes :

1) Expected Transmission Count (ETX)

2) Node Energy

3) Lost Paket

5.3. Expected Transmission Count (ETX)

Il se réfère au nombre de retransmission nécessaire pour qu'un paquet soit reçu avec succès à destination. La valeur d'ETX peut donc donner des informations sur l'état du réseau et sa stabilité. Au lieu du nombre de sauts, les fonctions objectives peuvent se baser sur cette métrique pour sélectionner les chemins vers la racine. En effet au plus ETX est faible au plus les liaisons du réseau sont stables car une valeur élevée est synonyme de beaucoup de retransmissions et donc une consommation énergétique élevée. La valeur d'ETX peut être calculée en utilisant la formule suivante :[10] Le « Df » représente un taux de livraison, c'est la probabilité mesurée qu'un paquet soit reçu par un voisin. Le « Dr » est un rapport de livraison dit inverse, c'est en fait la probabilité mesurée qu'un paquet d'accusé de réception soit reçu avec succès.

$ETX = 1 \div (Df \times Dr)$

5.4. Node Energy

C'est l'énergie mesurée sur les noeuds du réseau par rapport à la durée de vie du réseau. La formule utilisée pour calculer l'énergie des noeuds est[10] :

$$\text{Energy (mJ)} = (\text{Transmit} * 19.5\ \text{mA} + \text{Listen} * 21.5\ \text{mA} + \text{CPU\_time} * 1.8\ \text{mA} + \text{LPM} * 0.0545\ \text{mA}) * 3\ \text{V} / (32768)$$

FIGURE 4.3 – La métrique Node Energy

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

6. Résultats de la simulation

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

L'exécution des simulations a généré un ensemble de graphes en courbes qui traitent le changement d'énergie moyenne, et la formation du DODAG par rapport à la taille du réseau (nombre de noeuds).

Dans toutes les simulations, nous évaluons les performances de la solution implémentée avec Np=5, le nombre de sonde DIO envoyés par chaque noeud pendant sa phase de détection et Nmax=4. Aussi avec le Lth et U paramètres nous les réglons à 1,5 et 1 seconde respectivement. Les résultats obtenus avec notre implémentation sont par rapport aux résultats obtenus sans sondage, ce qui est l'approche adoptée par Contiki dans sa mise en oeuvre initiale.

6.1. Simulation 1 : 5 nœuds

Nous avons comparé entre l'énergie consommée par un réseau avec l'implémentation initiale et un réseau avec l'implémentation de la solution. La différence est que avec la solution implémentée, l'énergie consommée par chaque nœud est légèrement supérieur à celui d'un noeud avec pour implémentation initiale, cela est dû par l'echange des méssages de sonde par chaque noeud et la comparaisons des parents préferés avant la formation du DODAG. Illustration dans le figure 4.5.

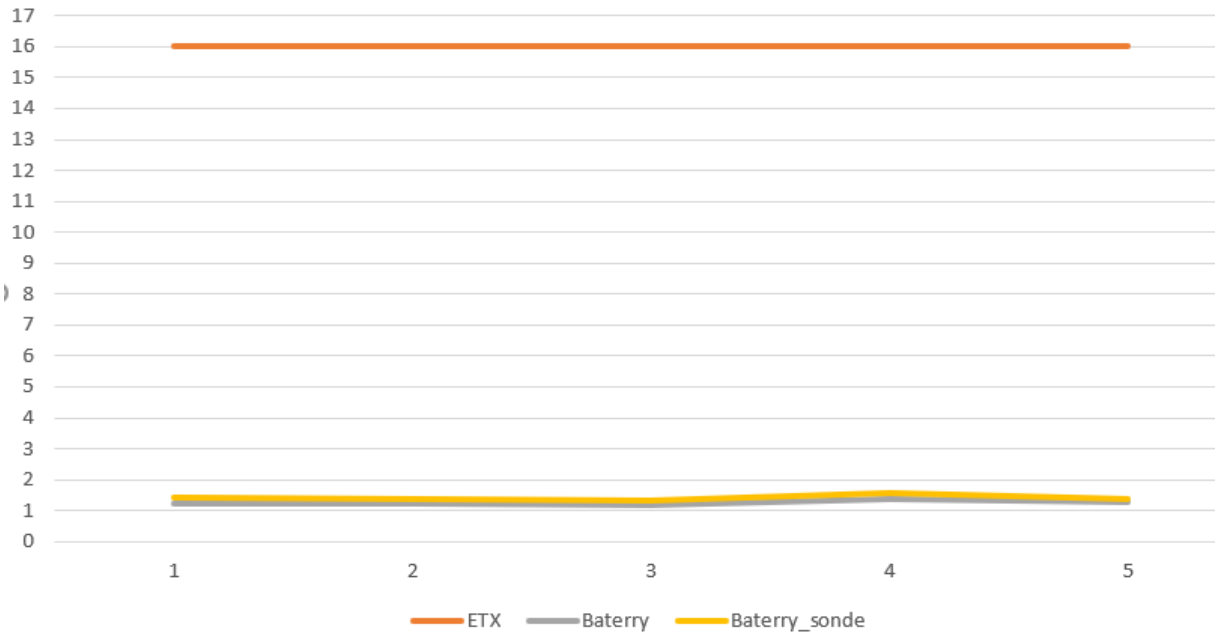

FIGURE 4.4 – La simulation avec 5 nœuds

6.2. Simulation 2 : 10 nœuds

Nous avons comparé entre l'énergie consommée et le temps du choix définitif par les nœuds avec l'implémentation initiale et un réseau avec l'implémentation de la solution. La différence est qu'avec la solution implémentée, l'énergie consommée par chaque nœud est légèrement supérieur à celui d'un noeud avec pour implémentation initiale tandis que les noeuds choisissent le meilleur parent préféré(stabilité du réseaux après la formation du DODAG) avec un temps record avec la solution que nous avons implémenté qu'avec la solution initiale. Bien que la formation du DODAG est un peu rétardé en foncion Nmax et U, le réseau est stable après la formation du DODAG car chaque noeud a le meilleur parent. Illustration dans le figure 4.6.

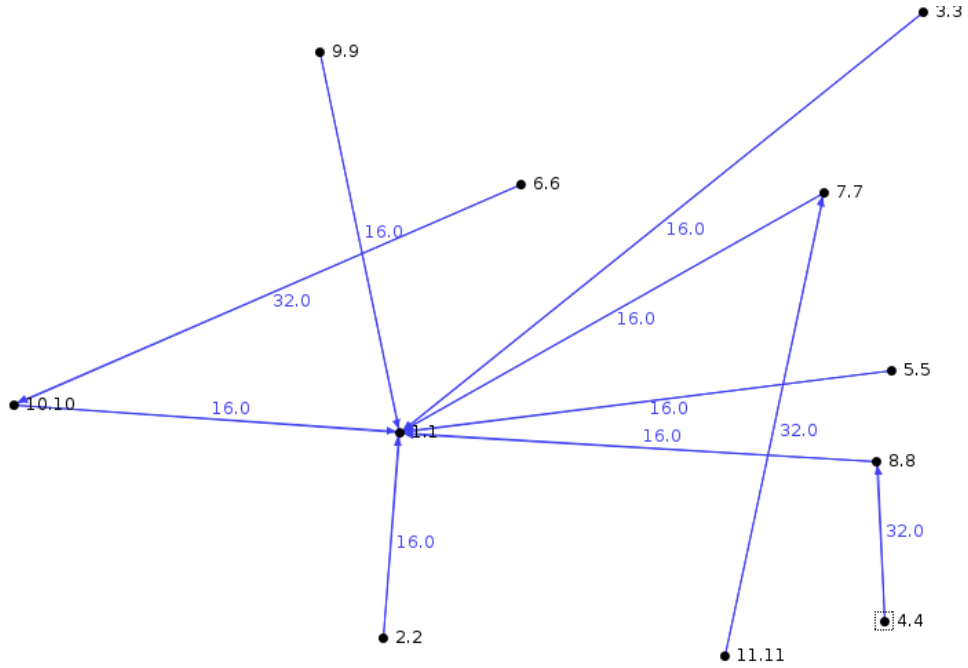

FIGURE 4.5 – La simulation avec 10 nœuds

6.3. Simulation 3 : 25 nœuds

Nous avons comparé entre l'énergie consommée et le temps du choix définitif par les nœuds avec l'implémentation initiale et un réseau avec l'implémentation de la solution. La différence est qu'avec la solution implémentée, l'énergie consommée par chaque nœud est légèrement supérieur à celui d'un noeud avec pour implémentation initiale tandis que les noeuds choisissent le meilleur parent préféré(stabilité du réseaux après la formation du DODAG) avec un temps record avec la solution que nous avons implémenté qu'avec la solution initiale. Bien que la formation du DODAG est un peu rétardé en foncion Nmax et U, le réseau est stable après la formation du DODAG car chaque noeud a le meilleur parent. Illustration dans le figure 4.9.

7. Conclusion

Dans cet chapitre, nous avons implémenté le mécanisme léger d'estimer la qualité des liens dans les WSN adoptant RPL comme protocole de routage et puis comparé les résultats avec l'implémentation initiale de RPL. L'approche implémenté exploite le Trickle algorithme, proposé à l'origine dans RPL pour réguler l'inondation des messages de signalisation de routage, pour contrôler également la transmission de messages de sonde de diffusion au moment de configuration de la topologie. Afin d'assurer la rétro-compatibilité de cette technique d'esti-

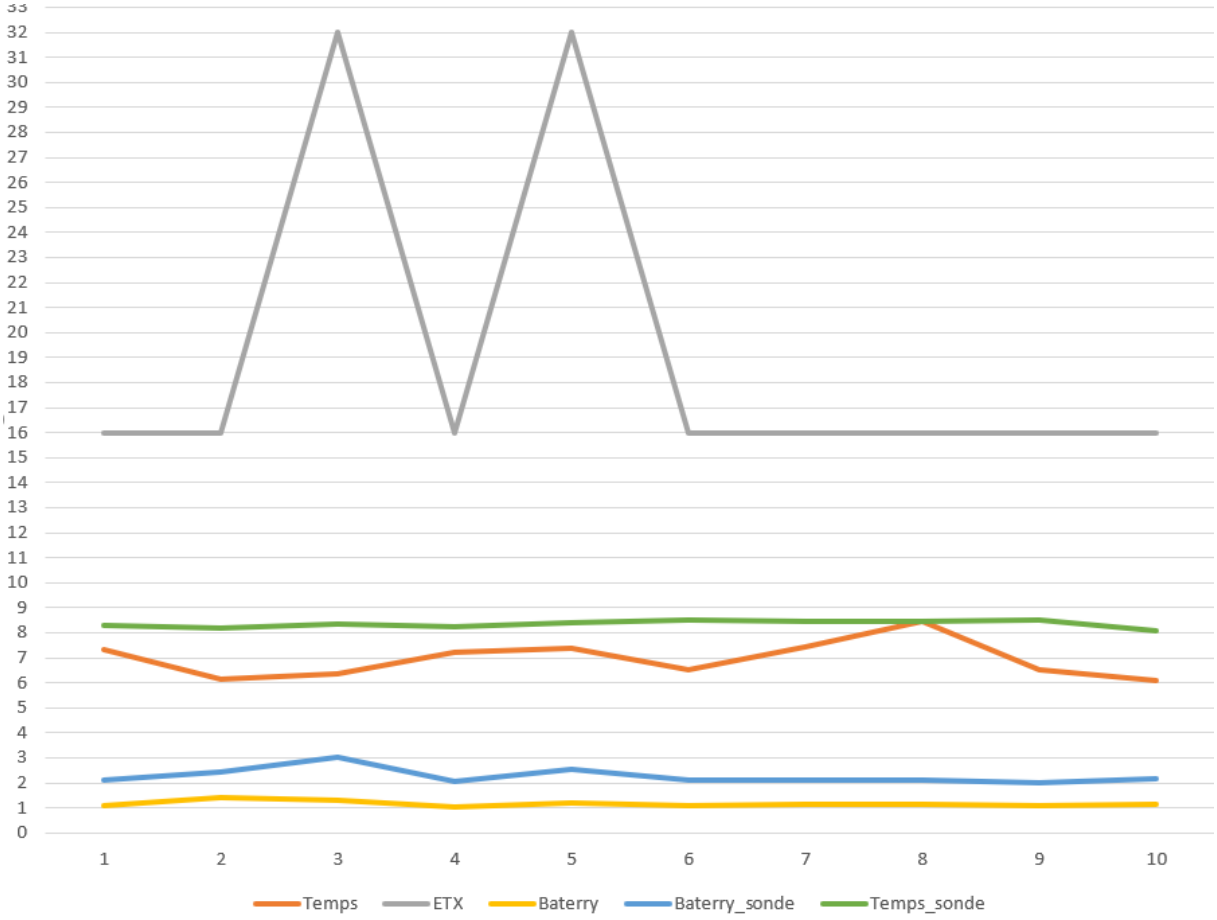

FIGURE 4.6 – La graphe pour 10 nœuds

mation de lien implémenté, tels que les paquets de sonde sont basés sur des messages DIO réguliers. Cela permet également de évaluer simultanément la qualité du lien tout en diffusant les informations de routage.

FIGURE 4.7 – Le réseau formé avec 25 nœuds en grille non-ordonné

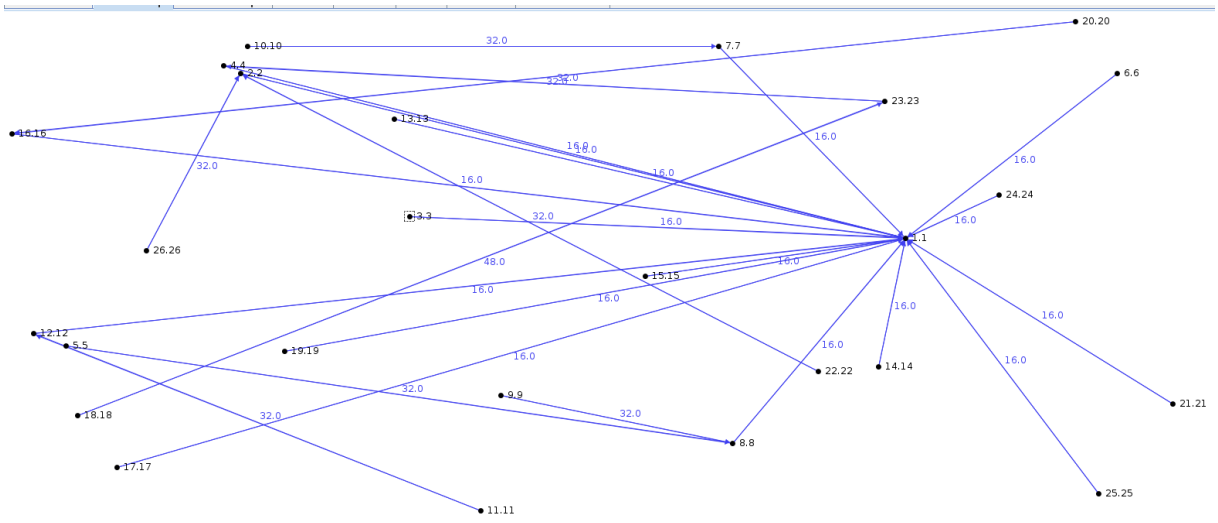

FIGURE 4.8 – Le réseau formé avec 25 nœuds

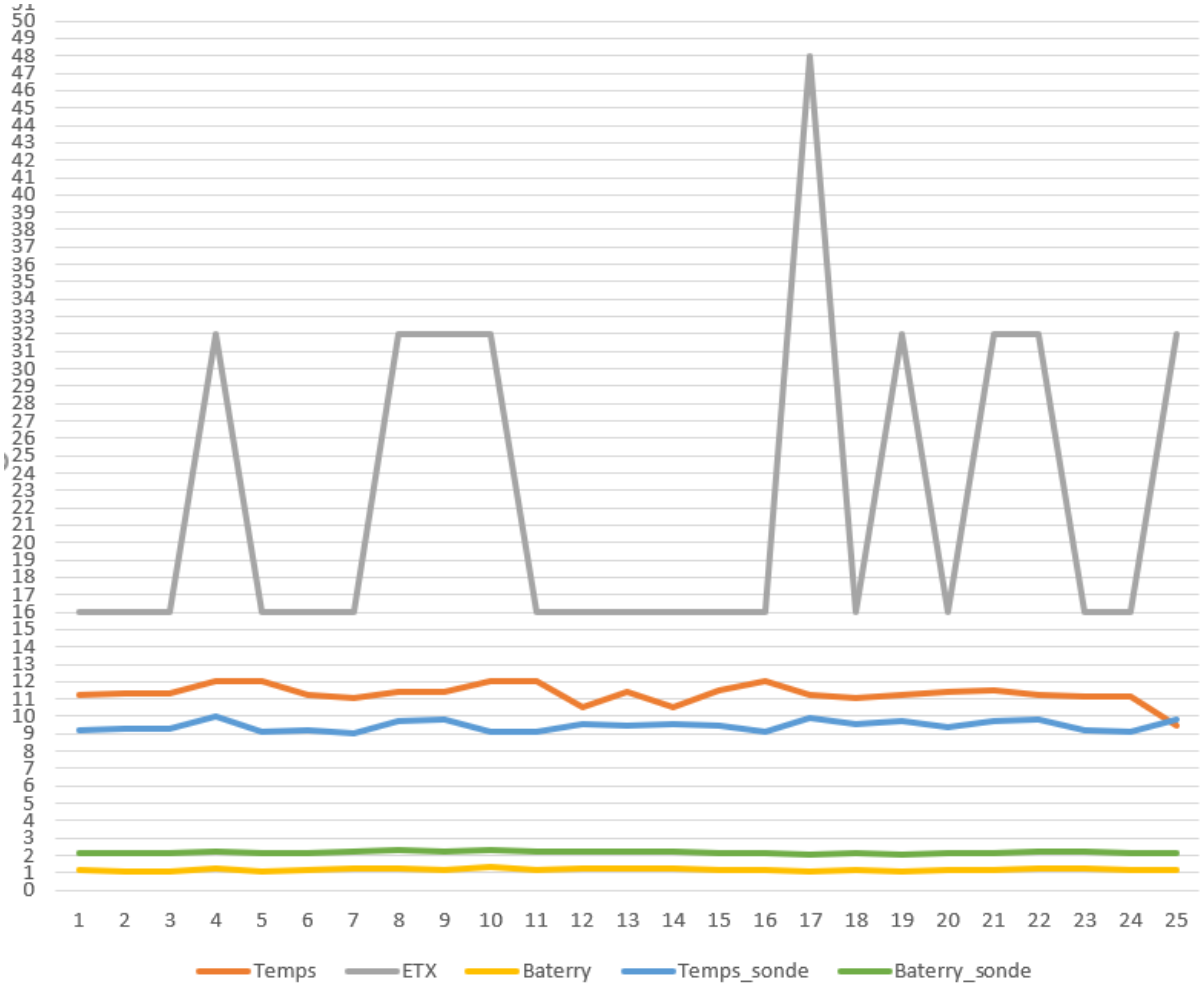

FIGURE 4.9 – La simulation avec 25 nœuds

CONCLUSION GÉNÉRALE

Dans ce projet, nous avons parlé un peu brièvement sur les réseaux de capteurs sans fils, nous avons présenter le fonctionnement du protocole de routage RPL (IPv6 Routing Protocol for Low Power and Lossy Networks) qui est conçu pour prendre en charge les exigences spécifiques des réseaux LLNs à savoir peu d'autonomie énergétique, peu de mémoire de stockage et peu de puissance de calcul. Ce qui rend le protocole RPL peu adapté à la mobilité des nœuds et le changement de la topologie du réseau. En suite, nous avons implémenté le mécanisme d'estimation de la qualité des liens proposé par d'autres chercheurs afin de pouvoir estimer avec précision la qualité de lien dans un réseau LLN. Les résultats obtenus à l'aide de l'émulateur Cooja indiquent que l'impémentation de la technique d'estimation est capable de mesurer les qualités des liens avec une précision décente et de petits frais généraux supplémentaires, et sans introduire de retard significatif dans le temps nécessaire le réseau pour former la topologie. De plus, la solution implémenté permet à RPL de converger rapidement vers une routes, ce qui améliore considérablement la stabilité et la fiabilité du processus de transmission des données. Cette implémentation, nous a permit d'approfondir nos connaissances sur les LLNs mais surtout le protocole de routage RPL et son mécanisme d'estimation de la qualité de lien et son implémentation. Dans les travaux futurs, nous prévoyons d'étudier comment étendre les techniques proposées pour fonctionner dans des scénarios dynamiques (par exemple, des noeuds joignent le réseau, ou à court de batteries, etc.).

BIBLIOGRAPHIE

[1] RPL in a nutshell : A survey, Olfa Gaddour and Anis Koubâa

[2] : Trickle-L2 : Lightweight Link Quality Estimation, Ancillotti, Raffaele Bruno, Marco Conti through Trickle in RPL Networks

[3] RPL : protocole de routage IPv6 pour les réseaux à faible puissance et avec perte, RFC 6550

[4] Routing Metrics Used for Path Calculation in Low-Power and Lossy Networks, RFC 6551

[5] L'algorithme Trickle, RFC 6206

[6] : «Les Réseaux de capteurs (WSN : Wireless Sensor Networks) », Rapport interne, Université de Technologie de Compiègne, France, 2008.

[7] , LoWPAN demystified, Texas Instruments 2014(Article).

[8] ,"RPL The IP routing protocol designed for low power and lossy networks", (IPSO) alliance, april 2011.

[9] A Performance Evaluation of RPL in Contiki, Hazrat Ali.

[10] OPTIMISATION DU ROUTAGE DANS LES RÉSEAUX LLNs, Mr. Mohamed BABOUCHE Mr. Fethallah TOUATI.

[11] Les Réeseaux de Capteurs sans Fil dans l'Internet des Objects, Sarra HAMMOUDI.

[12] Design and Analysis of Routing Protocol for IPv6 Wireless Sensor Networks, Elias Wendm Atalay

BIBLIOGRAPHIE

[13] Olfa Gaddour, Anis Koubâa, RPL in a nutshell, Polytechnic Institute of Porto, 02.14.2013 Mémoire de Stage de fin d'études Master Informatique, institut de la Francophonie Lyon 1

[14] Yacine CHALLAL, Hatem BETTAHAR, Abdelmadjid BOUABDALLAH, «Les Réseaux de capteurs (WSN : Wireless Sensor Networks) », Rapport interne, Université de Technologie de Compiègne, France, 2008.

[15] D. J. De Couto, D. Aguayo, J. Bicket, and R. Morris, "A High-throughput Path Metric for Multi-hop Wireless Routing," in Proc. of ACM MobiCom '03, ser. MobiCom '03. New York, NY, USA : ACM, 2003

[16] Lucien Loiseau, Thèse de Doctorat De l'exploitation des réceptions opportunistes dans les mécanismes de relayage pour les réseaux sansfil, Université de Rennes, 1 Décembre 2013.

[17] K.-H. Kim and K. G. Shin, "On accurate measurement of link quality in multi-hop wireless mesh networks," in Proc. of ACM MobiCom'06, 2006