

UNIVERSITÉ DU QUÉBEC À MONTRÉAL

SENSIBILITÉ ÉCOÉNERGÉTIQUE DANS LES RÉSEAUX DE CAPTEURS SANS
FIL MOBILES

MÉMOIRE PRÉSENTÉ
COMME EXIGENCE PARTIELLE
DE LA MAITRISE EN GÉNIE ÉLECTRIQUE

PAR

AMINE MOHAMED BOUBEKRI

MARS 2016

UNIVERSITÉ DU QUÉBEC À MONTRÉAL
Service des bibliothèques

Avertissement

La diffusion de ce mémoire se fait dans le respect des droits de son auteur, qui a signé le formulaire *Autorisation de reproduire et de diffuser un travail de recherche de cycles supérieurs* (SDU-522 – Rév.07-2011). Cette autorisation stipule que «conformément à l'article 11 du Règlement no 8 des études de cycles supérieurs, [l'auteur] concède à l'Université du Québec à Montréal une licence non exclusive d'utilisation et de publication de la totalité ou d'une partie importante de [son] travail de recherche pour des fins pédagogiques et non commerciales. Plus précisément, [l'auteur] autorise l'Université du Québec à Montréal à reproduire, diffuser, prêter, distribuer ou vendre des copies de [son] travail de recherche à des fins non commerciales sur quelque support que ce soit, y compris l'Internet. Cette licence et cette autorisation n'entraînent pas une renonciation de [la] part [de l'auteur] à [ses] droits moraux ni à [ses] droits de propriété intellectuelle. Sauf entente contraire, [l'auteur] conserve la liberté de diffuser et de commercialiser ou non ce travail dont [il] possède un exemplaire.»

RÉSUMÉ

Le manque de ressources énergétiques dans les réseaux de capteurs sans fil est un problème majeur pour leur autonomie, en particulier ceux dotés de nœuds mobiles. Les messages de contrôle engendrés par la mobilité aident à garder la topologie du réseau à jour, mais demandent une dépense d'énergie importante, créant des disparités énergétiques considérables entre les nœuds, et provoquant des déconnexions et des pertes de paquets.

Pour pallier ce problème, nous proposons un mécanisme à sensibilité écoénergétique pour la mobilité des nœuds. Ce mécanisme prend, premièrement, en charge la mobilité des nœuds en limitant la perte de paquets. Deuxièmement, il fait intervenir le facteur d'énergie résiduelle dans son processus de décision de manière à limiter les disparités énergétiques entre les nœuds. Ensuite, nous proposons un algorithme adaptatif pour limiter le nombre de basculements dans le réseau pour éviter les coupures fréquentes et les pertes de données qui peuvent en résulter.

Le mécanisme proposé a été simulé, puis comparé à des implémentations disponibles dans la littérature. Les résultats montrent que ce mécanisme parvient à équilibrer la consommation énergétique des nœuds durant la durée de vie du réseau, tout en maintenant la connectivité avec un nœud mobile.

Mots clés : Réseaux de capteurs sans fil, mobilité, basculement, énergie résiduelle, conscience énergétique.

TABLES DES MATIÈRES

TABLES DES MATIÈRES	I
LISTE DES FIGURES	IV
LISTE DES TABLEAUX	V
LISTE DES ACRONYMES	VI
PRÉFACE	1
CHAPITRE I	4
INTRODUCTION	4
1. Internet des objets:	5
1.1. Réseaux de capteurs sans fil	5
1.2. Passerelle	22
1.3. Pourquoi étudier la mobilité dans IoT ?	25
2. Problématique du mémoire	27
3. Objectifs	28
4. Hypothèses de recherche	29
5. Limitations de la recherche	29
CHAPITRE II	34
REVUE DE LA LITTÉRATURE	34
1. Mobilité dans les réseaux WSN	34
1.1. La solution dite : « RPL under mobility »	34
1.2. Co-RPL	36
1.3. MoMoRo	37
1.4. RPL avec mobilité améliorée	39
1.5. Réseau des proxy (Network of Proxies)	40
1.6. mRPL	41
1.7. LoWMob	42
2. Les réseaux conscients énergétiquement :	43
2.1. Energy Aware routing :	43
2.2. Protocoles basés sur les négociations :	44

2.3. EAP :	45
3. Analyse critique :	46
CHAPITRE III	50
SOLUTION PROPOSÉE	50
1. Introduction	50
2. Solution proposée	50
2.1. Phase opérationnelle:	51
2.2. Phase de basculement :	51
3. Fonction objectif :	52
3.1. Qualité du lien « RSSI » :	53
3.2. Facteur énergie résiduelle :	53
3.3. Le poids (weight) :	55
4. Optimisation de la solution proposée	56
CHAPITRE IV	60
SIMULATIONS	60
1. Choix du protocole RPL :	60
2. Paramètres de simulation :	63
2.1. Politique d'économie d'énergie	63
2.2. Messages de controles:	65
2.3. Choix du SEE et du simulateur COOJA	69
3. Mise en oeuvres des simulations :	70
CHAPITRE V	75
RÉSULTATS	75
1. Introduction	75
2. Comparaison avec RPL standard	75
3. Effet de la mobilité	78
3.1. Taux d'acheminement des paquets (TA)	79
3.2. Consommation énergétique	79
4. Optimisation du nombre de basculements	82
5. Comparaison de la solution proposée avec mRPL	84

6. Discussion	87
CHAPITRE VI	88
CONCLUSION	88
BIBLIOGRAPHIE.....	91
ANNEXE I	95

LISTE DES FIGURES

- Figure 1 Illustration des routages route-over et mesh-under dans le routage 6LoWPAN
- Figure 2 Schéma bloc détaillant les composantes de ContikiOS
- Figure 3 Schéma bloc détaillant les composantes de TinyOS
- Figure 4 Schéma bloc détaillant les composantes de LiteOS
- Figure 5 Illustrations de différentes architectures de passerelles utilisées dans le domaine de l'Internet des objets
- Figure 6 Diagramme temporel de la solution proposée
- Figure 7 Handoff Algorithm
- Figure 8 Diagramme d'état de la solution proposée
- Figure 9 Illustration des étapes de la construction de la topologie dans le protocole RPL
- Figure 10 Illustration du fonctionnement de ContikiMAC
- Figure 11 Illustration des différents minuteurs utilisés dans ContikiMAC
- Figure 12 Illustration des modifications apportées aux messages DIS
- Figure 13 Illustration des modifications apportées aux messages DIO
- Figure 14 Illustration de la topologie de réseau simulée
- Figure 15.a Illustration du taux d'acheminement des paquets en fonction des intervalles DIO
- Figure 15.b Illustration de la charge du réseau en fonction des intervalles DIO
- Figure 16 Illustration des résultats obtenus avec le profil 1
- Figure 17 Illustration des résultats obtenus avec le profil 2
- Figure 18.a Bilan énergétique du réseau après 4 minutes de simulation
- Figure 19 Comparaison du nombre de basculements par rapport au temps de deux solutions (HA et No HA)
- Figure 20 Taux d'acheminement (TA) vs cycle de service pour différentes fréquences d'envoi de données
- Figure 21 Illustration de l'impact de l'intervalle du cycle de service sur le délai de découverte d'un nouveau parent
- Figure 22 Illustration de la charge du réseau et de l'énergie consommée en fonction de la fréquence d'éveil

LISTE DES TABLEAUX

Tableau 1-a Résumé de quelques études sur les réseaux de capteurs sans fils mobiles

Tableau 1-b résumé de quelques études sur les réseaux écoenergetiques

Tableau 2 Comparatif entre les différents protocoles de routage WSN

Tableau 3 Illustration des scénarios de réactivité du protocole RPL standard

LISTE DES ACRONYMES

ADC	Analog to Digital Converters (Convertisseur Analogique numérique)
AODV	Ad-hoc On-demand Distance Vector (Protocole vecteur de distance à la demande Ad-hoc)
BSM	Broadcast Solicitation Message (Message de sollicitation diffusé)
CCA	Clear Channel Assessment (Évaluation de canal libre)
COAP	Constrained Application Protocol (Protocole d'application sous contrainte)
DIO	Directed Acyclic Graph Information Object (Objet d'information du graphe orienté acyclique)
DIS	Directed Acyclic Graph Information Solicitation (Sollicitation d'information du graphe orienté acyclique)
DP	Data packets (Paquet de données)
DTN	Delay Tolerant Network (Réseau tolérant aux délais)
ETX	Estimated transmission count (Nombre de transmissions estimé)
HA	Handoff Algorithm (Algorithme de basculement)
HTTP	Hypertext Transfer Protocol (Protocole de transfert Hypertexte)
ICMP	Internet Control Message Protocol (Protocole de messages de contrôle Internet)
IEEE	Institute of Electrical and Electronics Engineers (Institut des ingénieurs électriciens et électroniciens)
IETF	Internet Engineering Task Force (Détachement d'ingénierie d'Internet)
IoT	Internet of Things (Internet des objets)
IP	Internet Protocol (Protocole Internet)

LoWPAN	Low power Wireless Personal Area Networks (Réseaux personnels sans fil à faible capacité énergétique)
LPM	Low Power Modes (Modes faible consommation)
MAC	Media access control (Contrôle d'accès au support)
MRHOF	Minimum Rank with Hysteresis Objective Function
M2M	Machine To Machine (Machine à machine)
NM	Nœud Mobile
NoP	Network of Proxies (Réseau de serveurs mandataires)
NPP	Nœuds Parents Potentiels
OSI	Open Systems Interconnection
OBU	Mobile Onboard Units (Unité mobile)
PAN	Personal Area Network (Réseau personnel)
PP	Preferred Parent (Parent Préféré)
RDC	Radio Duty Cycle (Cycle de service de la radio)
REST	Representational State Transfer (Transfert des représentations des ressources)
RFC	Request for Comments (Appel aux commentaires)
RFID	Radio Frequency IDentification (Identification radio-fréquences)
RPL	Routing Protocol for low-power and lossy networks (Protocole de routage pour les réseaux à pertes et à faible capacité énergétique)
RREP	Route-Reply Message (Message de réponse)
RREQ	Route Request Message (Message de requête de route)
RSSI	Received Signal Strength Indication (Indicateur de puissance du signal reçu)
RSU	Road Side Units (Unités en bordure de route)
SEE	Système d'exploitation embarqué
SER	Symbol Error Rate (Taux d'erreur de symbole)

SICS	Swedish Institute of Computer Science (Institut suédois d'informatique)
SNMP	Simple Network Management Protocol (Protocole simple de gestion de réseau)
TA	Taux d'acheminement
TCP	Transmission Control Protocol (Protocole de controle de transmissions)
UDP	User Datagram Protocol (Protocole de datagramme utilisateur)
UIM	Unicast Information Message (Message d'information monodiffusion)
URI	Uniform Resource Identifier (Identifiant uniforme de ressource)
USM	Unicast Sollicitation Message (Message de sollicitation monodiffusion)
VANET	Vehicular Ad hoc Network (Réseau Ad-hoc de véhicules)
W	Weight (Poids)
WSN	Wireless Sensor Networks (Réseaux de capteurs sans fil)

PRÉFACE

Motivations

Depuis l'apparition des premiers ordinateurs, les scientifiques se sont intéressés à différents aspects relatifs aux données informatiques dont l'acheminement de l'information. Cet intérêt s'est manifesté dans divers domaines tels que les télécommunications, l'infonuagique et l'Internet des objets. Récemment, la mobilité offerte par les téléphones intelligents, les tablettes et les bracelets électroniques dans le domaine de l'Internet des objets, ainsi que ses limitations ont ouvert de nouvelles opportunités de recherche. La mobilité dans les réseaux de capteurs sans fil est difficile à implémenter et très coûteuse en termes d'énergie. En effet, les nœuds mobiles engendrent une quantité considérable de messages de contrôle et sollicitent constamment les nœuds voisins. Ces sollicitations entraînent une consommation excessive d'énergie et un taux de perte de paquets important dû à l'isolement de certains nœuds dont les sollicitations excessives épuisent les batteries.

Ce défi interpelle, et soulève la question suivante : comment améliorer le taux d'acheminement des données envoyées par un nœud de capteur mobile en prenant en considération l'état énergétique des capteurs afin de les maintenir opérationnels le plus longtemps possible ? Autrement dit, de quelle manière un nœud en mobilité peut-il communiquer avec les autres nœuds du réseau afin que ceux-ci consomment leur énergie de manière adéquate pendant toute la durée de vie du réseau? Il existe une approche permettant de maintenir un maximum de connectivité tout en contrôlant la consommation énergétique globale du réseau, grâce à la mise en œuvre de protocoles de routage écoénergétiques.

Contexte et contributions du mémoire

Les travaux de recherche qui traitent la mobilité dans les réseaux de capteurs sans fil portent attention essentiellement aux tenants et les aboutissants de la mobilité. En effet, ils considèrent principalement la manière avec laquelle les données envoyées par un capteur mobile sont acheminées jusqu'à destination, la manière de minimiser le temps de basculement d'un nœud mobile vers un autre et l'impact que cette mobilité a sur la consommation énergétique des nœuds. Cependant, dans le contexte d'un nœud de capteur mobile attaché à des nœuds connectés à Internet, aucune étude ne considère les exigences énergétiques de ces nœuds dans le but de l'intégrer au schéma de mobilité. Or, puisque l'énergie est une ressource précieuse dans ce type de réseau, il est indispensable de la considérer dans notre étude. Pour cela, dans ce travail, nous proposons un mécanisme qui facilite la procédure de basculement ou changement de nœud d'attache (handoff) nécessaire pour tout capteur en mobilité en utilisant l'énergie comme facteur d'aide à la décision. Le mécanisme proposé contient trois éléments essentiels. Premièrement, nous proposons un mécanisme de gestion de la mobilité dédié aux réseaux de capteurs sans fil afin de réduire les pertes de paquets de données. La fonction principale de ce mécanisme est de maintenir le nœud mobile connecté au réseau quel que soit son emplacement ou sa vitesse. Deuxièmement, nous intégrons à ce mécanisme un algorithme qui permet de considérer l'énergie du réseau dans la prise de décision au moment du basculement. Enfin, nous développons un algorithme adaptatif qui permet au nœud mobile de minimiser le nombre de basculements pendant ses déplacements. Ceci permet d'éviter les coupures fréquentes pouvant mener à des pertes de données, et ainsi économiser l'énergie de retransmission qui s'ensuit. Les résultats obtenus montrent que le mécanisme proposé permet d'équilibrer la consommation énergétique des nœuds du réseau pendant toute la durée de vie de celui-ci. Il permet également de maintenir la connectivité du réseau et d'éviter l'isolement des nœuds.

Structure du mémoire

Afin d'expliquer au mieux le travail accompli et sa contribution, nous avons structuré ce document comme suit: le chapitre I présente un aperçu des différents composants de notre projet. Ce chapitre commence par l'explication de la technologie de l'Internet des objets et les éléments de base de notre étude. Puis nous présentons notre problématique, nos objectifs ainsi que le périmètre de notre travail de recherche. Le chapitre II présente un survol de la littérature pertinente afin de résumer l'état de l'art. Ce chapitre aide à analyser certains travaux associés à la mobilité et d'autres qui traitent l'aspect énergétique dans les réseaux de capteurs sans fil. Nous concluons ce chapitre avec les points forts et les points faibles de chacune des études. Le chapitre III décrit les différentes phases du mécanisme proposé en détail. Le chapitre IV présente l'environnement de simulation et spécifie le périmètre d'étude. Ce chapitre commence par une justification du choix du protocole de routage ainsi que du système d'exploitation, suivi d'une explication des différentes variables nécessaires à la simulation. Le chapitre V présente les résultats obtenus et les comparaisons faites avec d'autres travaux de même nature. Finalement, le chapitre VI conclut le mémoire avec un résumé des travaux effectués et des résultats obtenus. Nous y identifions également des travaux futurs possibles.

CHAPITRE I

INTRODUCTION

La notion d'« Internet des objets » est apparue au début du 21e siècle suite à une demande naturelle de l'industrie pour la gestion et la supervision des machines en usine. Le but était de connecter les réseaux informatiques des entreprises aux machines automatisées pour superviser les cycles de productions à distance, en utilisant Internet. Appelée tout d'abord « Internet des machines », cette technologie permettait non seulement de connaître l'état des automates en temps réel, mais aussi de collecter les informations les concernant et celles des produits qui sont acheminés dans les chaînes de productions. La flexibilité offerte par cette nouvelle technologie a mené à la connexion de nouveaux types d'équipements au réseau informatique. Ces équipements, de plus en plus petits, ont fait l'objet de modifications afin de les connecter à des serveurs centralisés capables de les superviser. Nous citons par exemple la RFID (Identification radio-fréquence) qui a contribué énormément à vulgariser la notion « d'Objets Connectés ». En effet, les étiquettes RFID (en anglais RFID tag) , dont chacun peut servir à identifier un objet différent, sont dotés de radios permettant de transporter de l'information entre un objet et un récepteur distant.

L'information locale peut être utilisée sur place à des fins d'inventaire, de prévention des risques, etc. ou acheminée en temps réel vers des centres de traitement lointains grâce à Internet. Aujourd'hui, cette nouvelle technologie a dépassé le cadre initial de gestion des machines et touche à toute source d'information pouvant communiquer sans fil, avec un accès à Internet, d'où l'appellation « Internet des objets », ou en anglais « Internet of Things » ou « IoT ».

Au fil des ans, l'IoT est devenu de plus en plus populaire. Il offre la possibilité aux individus de gérer des objets à travers Internet. Cette tendance a donné lieu à une

nouvelle ère informatique appelée « l'informatique ubiquitaire », dont l'omniprésence des technologies permet d'accéder à diverses applications indépendamment de l'emplacement géographique. C'est sur cette idée que la mobilité dans l'Internet des objets a pris naissance. En effet, l'échange de données par ondes radio devient indispensable compte tenu du nombre important de réseaux de capteurs mobiles. Cependant, cette mobilité subit plusieurs contraintes importantes comme l'efficacité énergétique, la fiabilité des liens ainsi que la latence. Plusieurs travaux de recherches dont nous parlerons plus tard traitent de ces problèmes.

1. Internet des objets:

Les applications de l'Internet des objets se trouvent partout dans notre quotidien, dans la rue, les voitures, les immeubles, les hôpitaux, etc. Face à ce flux, divers organismes de standardisation tels que l'IETF (Internet Engineering Task Force) veillent à réglementer et normaliser les différents acteurs logiciels et matériels de l'IoT. Cette technologie de connexion des objets réels au monde virtuel (informatique) doit être standardisée afin de permettre une interopérabilité entre les différentes solutions des fabricants. L'architecture globale est composée de trois parties : les réseaux de capteurs sans fil, la passerelle et Internet.

1.1. Réseaux de capteurs sans fil

Les réseaux de capteurs sans fil sont des réseaux de communication pour capteurs ou actionneurs qui communiquent entre eux selon des standards de transmission radio. Ils peuvent être construits suivant plusieurs topologies, notamment : topologie en étoile, topologie en grille ou bien un mélange hybride des deux. Cette dernière sera utilisée dans notre projet comme topologie de référence.

1.1.1. Capteurs

Selon George Asch (2010), un capteur est un instrument permettant de mesurer l'état d'une grandeur physique m (mesurande) en y associant un signal électrique s . Ce dernier est considéré comme l'image de l'état de la grandeur physique exprimé suivant une relation $s=f(m)$.

Il existe des capteurs dits intelligents qui sont très répandus dans le domaine de l'IoT. Considérés comme des « systèmes embarqués », ces capteurs sont des systèmes électroniques indépendants et dédiés qui souffrent souvent d'une contrainte de ressources computationnelles et d'énergie. Ces systèmes comportent généralement les quatre composants fonctionnels suivants :

- unité d'acquisition : cette unité est composée d'un capteur qui collecte les mesures des états de grandeurs physiques et qui les transmet à l'unité de traitement sous forme de données binaires après passage par un convertisseur analogique-numérique (en anglais Analog Digital Converter ou ADC),
- unité de traitement : cette unité est composée généralement d'un microprocesseur et d'une mémoire vive et d'une mémoire de programme. Une fois le signal reçu de l'ADC, le microprocesseur accomplit les fonctions préprogrammées par l'utilisateur et envoie les résultats par le biais de l'unité de communication. Ces microprocesseurs fonctionnent souvent sous le contrôle de systèmes d'exploitation dédiés dont nous parlerons plus tard,
- unité de communication : cette unité sert d'interface pour collecter les données résultantes d'un traitement ou interagir avec l'unité de traitement. Elle peut être sous forme filaire ou sans fil dépendamment du système embarqué,
- unité d'alimentation : cette unité fournit l'énergie nécessaire à l'ensemble des unités pour accomplir leurs tâches. Elle peut être à base de batteries (alimentation limitée dans le temps), de source d'alimentation infinie ou d'énergie renouvelable. Cependant, les applications industrielles requièrent souvent des systèmes

embarqués qui opèrent sous des contraintes énergétiques importantes. Or, compte tenu de la taille des capteurs, les batteries sont souvent de taille réduite. Cette contrainte rend l'énergie la ressource la plus précieuse puisqu'elle influe directement sur la durée de vie des capteurs.

Il existe une multitude de systèmes embarqués sur le marché. Ils peuvent être sous forme de plateformes de développement qui comportent plusieurs types de capteurs (température, humidité, accélération, etc.) ou dédiés à des applications particulières. Pour notre projet, nous choisirons une plateforme qui pourra répondre au besoin de mobilité tout en garantissant les prérequis de notre étude. Nous couvrirons le choix de notre plateforme plus en détail dans les chapitres à venir.

1.1.2. Standards et protocoles

L'étendue de la technologie de l'Internet des objets couvre un grand nombre de plateformes et de logiciels. Dans le cadre de notre projet, nous nous intéressons à quelques standards qui définissent des normes de consommation énergétique particulières ainsi que certains facteurs liés à la mobilité dans les réseaux de capteurs sans fil. Nous présentons ces standards selon le modèle OSI (en anglais Open Systems Interconnection) à des fins de classement.

a) Couche d'application

Les protocoles de la couche d'application utilisés dans le domaine de l'Internet des objets ont une importance capitale, car ils représentent le point d'interaction entre les utilisateurs et les capteurs. Ces protocoles sont soumis à de multiples contraintes d'adaptation aux appareils mobiles tout en fournissant la même qualité de service que les applications WEB standards, particulièrement depuis l'avènement des téléphones intelligents. On peut citer parmi eux :

REST (en anglais Representational state transfer): C'est un style d'architecture décrit dans les travaux de recherche de Roy Fielding (Fielding, 2000). REST est un ensemble de directives qui aident à développer des applications destinées à la communication entre machines. L'architecture REST fait l'objet d'un intérêt particulier dans notre projet de recherche, car elle permet d'utiliser des protocoles dits « RESTful ». Ces protocoles sont destinés aux applications orientées ressources qui opèrent sur des réseaux IP limité en ressources comme les réseaux de capteurs sans fil. Elle permet également d'utiliser les fonctionnalités existantes du WEB en tirant parti de l'enveloppe et de l'entête des messages HTTP (Protocole de transfert Hypertexte) afin de véhiculer les informations.

COAP (en anglais Constrained Application Protocol): Littéralement protocole d'application sous contrainte est un protocole utilisé dans les réseaux à faible capacité énergétique qui permet aux capteurs de communiquer à travers Internet. Il est décrit dans l'appel aux commentaires RFC7252 (Z.Shelby, Request for Comments, juin 2014). COAP fonctionne selon un modèle de requête/réponse qui facilite l'interaction avec le protocole HTTP (Boswarthick *et al.*, 2012). L'utilisation de ce protocole est intéressante dans notre projet parce qu'il supporte un service de découverte de ressources et inclut les concepts clés du WEB (T. Berners-Lee, Request for Comments, janvier 2005). Ceci facilite la supervision à distance des nœuds du réseau.

b) Couche Transport :

Le choix d'un protocole de cette couche dépend des exigences de l'application implémentée. Ces exigences peuvent être le séquençement des données, la fiabilité d'acheminement des informations ou la vitesse de transfert des messages. Dans notre projet, nous simulons un scénario de collecte de données dans un réseau doté de nœuds mobiles. Le critère principal à considérer dans ce cas est le taux d'acheminement (TA) des données envoyées par le nœud mobile. L'utilisation du

protocole UDP (protocole de datagramme utilisateur) est adéquate, car celui-ci permet d'acheminer les flux de données plus rapidement que le protocole TCP (protocole de contrôle de transmissions). Ce dernier fonctionne selon un mode connecté qui priorise la fiabilité de l'acheminement sur la vitesse. L'expéditeur doit constamment attendre que le destinataire accuse réception du message envoyé et, pendant ce temps, aucune nouvelle session de communication ne peut être établie. Certes, cela permet d'assurer que le message envoyé soit bien reçu, mais au détriment du temps de réponse. De plus, le séquençement de données qu'offre le protocole TCP n'est pas d'un grand intérêt dans un scénario de capteurs mobiles, car un des objectifs principaux dans notre étude est l'acheminement des messages avant que le nœud mobile ne perde son attache avec le nœud parent. Le protocole UDP délègue la fonction de séquençement à la couche d'application qui dispose généralement de plus d'énergie que les capteurs dans le cas de l'IoT.

c) Couche Réseau :

L'acheminement des paquets de données dans la couche réseau est réalisé en suivant des protocoles de routages standardisés. Comme l'architecture IoT étudiée dans ce projet peut engendrer une saturation d'adresses due au nombre croissant de capteurs, il faut faire un choix judicieux du plan d'adressage avant l'implémentation. Le protocole IP (Protocole Internet) offre des plans d'adressages adaptés et est largement répandu dans le monde.

Cependant, le nombre croissant d'équipements connectés a vite dépassé la capacité d'adressage de la version IPv4 qui utilise des identifiants de 32 bits. Pour résoudre ce problème, l'IETF a défini IPv6 au cours des années 1990. Cette version identifie chaque capteur à l'aide d'un champ d'adressage étendu de 128 bits, d'où son adoption dans notre projet.

Le protocole IPv6 tel que défini dans la RFC 2460 (C.Perkins, Request for Comments, juillet 2003) comporte les fonctionnalités suivantes :

- capacité d'adressage étendue en augmentant la taille de l'adresse IP de 32 bits à 128 bits. L'évolutivité du routage multidiffusion est améliorée en ajoutant un champ d'adresses dédié. De plus, un nouveau type d'adresses appelées en anglais « Anycast Address » a été rajouté pour envoyer des paquets à des groupes de nœuds,
- simplification du format d'entête afin de réduire le coût de traitement des paquets et celui de la bande passante,
- support amélioré pour les extensions et les options afin de faciliter le transfert de paquets et d'améliorer la flexibilité d'introduction de nouvelles options,
- étiquetage du flux de paquets afin d'activer l'étiquetage des paquets appartenant à un flux particulier. De cette manière, l'expéditeur a la possibilité de demander un traitement spécial d'un trafic particulier,
- authentification et politique de sécurité améliorées afin de supporter l'authentification, l'intégrité des données et la confidentialité des données.

d) Couche Liaison

Les protocoles tels que le 802.11b ou Wi-Max ont été conçus pour supporter des communications serveur/serveur ou client/serveur dans de larges réseaux métropolitains. Cependant, plusieurs scénarios requièrent une portée beaucoup moins importante. Nous citons à cet effet les périphériques sans fil des ordinateurs ou les composants des cinémas maison. Les réseaux personnels (en anglais Personal Area Networks ou PAN), sont des exemples parfaits de ces scénarios. Ils ont une portée typique entre 1 et 100 mètres et supportent différents taux de transfert. Dans le cadre de notre étude, deux protocoles ont été présélectionnés afin d'assurer la

communication au niveau de la couche de liaison. Leur aspect écoénergétique permet de simuler un réseau de capteurs sans fil mobile tout en respectant les contraintes énergétiques.

Bluetooth : Les versions de Bluetooth antérieures à la version 4, communément appelées « Bluetooth classique », offraient des taux de transfert de données et des débits très intéressants, mais ces performances se faisaient au détriment de l'énergie. En effet, les équipements Bluetooth étaient considérés énergivores étant donnée leur autonomie limitée lorsqu'ils sont alimentés par batteries. Au courant de l'année 2011, « Bluetooth Low Energy » ou « BLE » a été développé. Cette version est d'un grand intérêt pour notre projet, car elle est caractérisée par une consommation énergétique très faible. Un capteur BLE consomme moins d'énergie qu'un capteur Bluetooth classique grâce à son cycle de service qui lui permet de rester en dormance la plupart du temps. En effet, un capteur BLE se réveille uniquement en cas d'initiation d'une connexion. Cette technique permet de maintenir un niveau de consommation énergétique très bas. BLE est d'ores et déjà adopté par les constructeurs de téléphonie mobile où il est déployé dans des millions d'équipements. Cependant, la portée limitée de sa radio nous poussent à envisager d'autres technologies.

IEEE 802.15.4 : La première version du protocole IEEE 802.15.4 a été introduite en 2003, suivie par deux autres révisions en 2006 et en 2011 où des spécifications de consommations énergétiques et de taux de transferts ont été mises à jour.

Dans le cadre de notre projet, la version IEEE 802.15.4-2011 est celle qui nous intéresse le plus, car elle comporte une politique d'économie d'énergie intéressante basée sur des cycles d'inactivités des nœuds.

Ce protocole supporte des réseaux PAN comportant des nœuds coordinateurs qui envoient périodiquement des trames de balises (en anglais beacons). Ces trames comportent des tranches temporelles d'activités qui permettent de synchroniser les nœuds du réseau et d'économiser de l'énergie en éteignant les radios non requises.

Cette fonctionnalité est d'autant plus intéressante dans notre projet, car les nœuds doivent garder leur radio allumée afin de recevoir les messages envoyés par le nœud mobile dans le cas où celui-ci est à proximité. Cette synchronisation requiert une gestion minutieuse des cycles d'activités des nœuds afin de trouver le bon compromis entre le taux d'acheminement des paquets et la consommation énergétique.

Quelques études se sont penchées sur les différences entre les protocoles BLE et IEEE 802.15.4 d'un point de vue énergétique telles que (Siekkinen *et al.*, 2013) et (J. Sens. Actuator Netw. 2013). Elles ont conclu que la consommation énergétique de BLE est nettement moins importante que celle de IEEE 802.15.4. Cependant, la portée de BLE atteint des distances associées généralement aux réseaux PAN. Cette portée ne correspond malheureusement pas au scénario de notre projet de recherche. En effet, nous envisageons des réseaux locaux (en anglais Local Area Networks) qui nécessitent des portées allant jusqu'à une centaine de mètres. Le standard IEEE 802.15.4 est alors mieux adapté à notre scénario.

e) Couche d'adaptation :

Le scénario étudié dans ce projet repose principalement sur l'interconnexion des réseaux de capteurs à Internet. Les mécanismes utilisés doivent permettre une interopérabilité totale entre les objets connectés et les utilisateurs distants. Il existe à cet effet deux méthodes différentes. La première méthode nécessite l'utilisation d'une passerelle qui sert de convertisseur entre les protocoles des réseaux de capteurs et Internet. La deuxième méthode consiste en l'utilisation de nœuds de capteurs qui supportent les protocoles IP afin de mettre en place une communication de bout en bout. C'est précisément cette méthode que nous utilisons dans ce projet. Cette dernière nécessite l'utilisation d'un protocole d'adaptation appelé 6LoWPAN (IPv6 Low power Wireless Personal Area Networks) pour permettre aux capteurs de communiquer avec le monde extérieur (Internet). La Zigbee Alliance a annoncé au

courant de l'année 2013 une technologie appelée Zigbee IP (<http://www.zigbee.org/the-latest/>) qui utilise également le protocole 6LoWPAN pour connecter les capteurs à Internet. Cependant cette version ne sera pas utilisée dans ce travail afin de ne pas se contraindre à travailler uniquement dans un environnement Zigbee uniquement.

La couche d'adaptation est utilisée afin que les trames 802.15.4 soient compatibles avec les paquets IP de la couche réseau, en respectant les contraintes de taille du premier où les données ne disposent que de 81 octets d'espace disponible pour les données. De plus, les contraintes énergétiques et la faible capacité de calcul des capteurs nécessitent l'emploi d'un protocole d'adaptation comme 6LoWPAN pour que les données soient acheminées.

f) Protocole 6LoWPAN

Dans notre projet, les performances de 6LoWPAN nous ont poussés à utiliser ce protocole plutôt qu'une passerelle. En effet, 6LoWPAN comporte des fonctionnalités qui permettent une excellente interopérabilité entre les protocoles IEEE 802.15.4 et IPv6 comme le montrent ses spécifications (N. Kushalnagar, Request for Comments, août 2007).

En présence de capteurs mobiles, les données envoyées doivent être acheminées rapidement sous peine d'un taux de pertes de paquets élevé. Les fonctionnalités de fragmentation et de réassemblage qu'offre le protocole 6LoWPAN permettent une communication nettement plus rapide entre le réseau de capteurs et Internet en comparant avec l'utilisation d'une passerelle où la conversion entre protocoles s'avère très lente. Comme mentionné précédemment, les trames 802.15.4 ne disposent que de 81 octets disponibles pour transporter les données ce qui est largement moindre que les 1280 octets dont dispose la couche IPv6. Un mécanisme de fragmentation et de réassemblage est donc nécessaire afin de fragmenter les paquets IPv6 reçus et de les réassembler à la réception. Chaque fragment de donnée

est précédé d'un entête pour faciliter la reconstitution des paquets originaux. Ces entêtes contiennent des informations de séquençement et d'appartenance. Le protocole 6LoWPAN spécifie une méthode d'autoconfiguration d'adresses de type sans état (en anglais stateless) qui s'avère très utile dans des projets comme les nôtres. Cette méthode permet de réduire les sollicitations sur les équipements et évite la configuration manuelle des adresses. De plus, les protocoles de gestion du réseau utilisés dans les réseaux IP peuvent être utilisés pour des réseaux 6LoWPAN, ce qui offre une compatibilité sans précédent dans ce domaine. Il est cependant nécessaire d'utiliser des versions adaptées de ces protocoles telles que le protocole simple de gestion de réseau (en anglais Simple Network Management Protocol ou SNMP) afin de garantir leur bon fonctionnement dans des réseaux de capteurs à faible capacité en mémoire.

g) Le routage dans 6LoWPAN

Après le choix naturel du protocole 6LoWPAN dans notre projet, il faut lui associer un type de routage. Comme décrit dans (E.Kim, Request for Comments, mai 2012), le standard 6LoWPAN supporte deux types de routages : un routage classique opérant dans la couche IP (en anglais Route-Over) et un routage sur la couche basse (en anglais Mesh-Under). Ce dernier ne reconstitue pas les paquets pour les décisions de routage. Par conséquent, leur acheminement y est plus rapide que dans le routage route-over qui reconstitue systématiquement les paquets au niveau de chaque équipement intermédiaire afin de prendre des décisions de routage. Dans le cas d'un réseau à fort taux de pertes, le routage route-over présente de meilleurs taux d'acheminements de paquets. Dans notre cas, un compromis reste à faire entre vitesse et fiabilité d'acheminement de données.

La figure 1 illustre les couches où les deux types de routage opèrent.

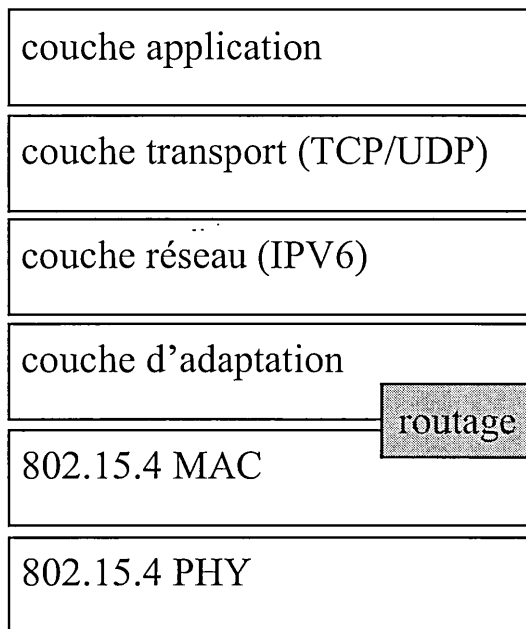


Figure 1.a Illustration du routage mesh-under dans le routage 6LoWPAN

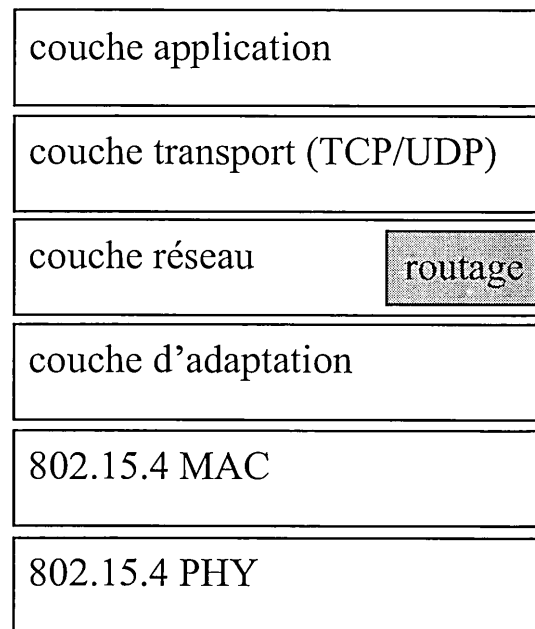


Figure 1.b Illustration du routage route-over dans le routage 6LoWPAN

Il existe plusieurs protocoles de routage développés par la communauté 6LoWPAN, dont le protocole de routage pour les réseaux à pertes et à faible capacité énergétique (en anglais Routing Protocol for low-power and lossy networks, ou RPL) (Winter *et Al*, Request for Comments, mars 2012). C'est un protocole route-over standardisé que nous couvrirons en détail dans les prochains chapitres. Par ce choix, nous avons donc adopté un routage de type Route-Over ce qui nous aide à maximiser le taux d'acheminement de paquets.

3. Systèmes d'exploitations embarquées « SEE »

À l'instar des processeurs, les microprocesseurs requièrent des systèmes d'exploitation afin de gérer l'allocation des ressources matérielles des systèmes embarqués, des utilisateurs, et des applications. Les microprocesseurs sont par définition contraints en énergie et en mémoire. Le noyau d'un système d'exploitation

classique ne pourrait donc être supporté par une mémoire de quelques kilo-octets. Dans ce cadre, plusieurs SEE ont été développés ces dernières années. Ils sont différents des systèmes d'exploitation classiques par le fait qu'ils ont une empreinte mémoire minimale et sont développés à l'aide d'un code optimisé. Il existe plusieurs architectures de SEE qui opèrent selon différents modèles d'exécutions. On cite à cet effet trois modèles. Le premier est le modèle multithread qui permet à un système d'exploitation de gérer son utilisation par plusieurs utilisateurs en même temps. Le deuxième est le modèle événementiel qui accorde la plus grande priorité à l'événement. Un processus en cours d'exécution peut être interrompu par l'arrivée d'un événement, mais pas l'inverse. Le troisième modèle est hybride où le modèle multithread synchrone opère en même temps avec le modèle événementiel asynchrone.

Parmi les nombreux systèmes d'exploitation embarqués disponibles (Reddy *et al.*, 2007), nous détaillons quelques-uns afin de choisir celui qui convient le mieux à notre projet :

a) ContikiOS

Ce système d'exploitation embarqué en distribution libre, écrit en langage C et dédié aux WSN (réseaux de capteurs sans fil) a été créé par une équipe de l'Institut Suédois de Recherche Scientifique SICS. Il est construit autour d'un modèle d'exécution événementiel et protothread qui est un modèle de programmation concurrentiel présentant une très faible surcharge de la mémoire d'où l'intérêt de l'utiliser dans notre projet. Une configuration typique de ContikiOS consomme en moyenne 2 Kilo-octets de mémoire vive et 40 Kilo-octets de ROM. ContikiOS supporte le multitasking, le multithreading et la pile TCP/IP. L'accès aux capteurs utilisant ContikiOS est possible grâce à une interface graphique, un navigateur web, un serveur web ou simplement en Telnet. Son atout majeur est le fait qu'il supporte le

protocole d'adaptation 6LoWPAN qui s'avère être un choix essentiel dans notre projet.

ContikiOS est développé sur une architecture modulaire au niveau de son noyau et il supporte deux types d'événements : asynchrones et synchrones. Toutes les fonctionnalités sont sous forme de services. On cite par exemple : la gestion des données des capteurs, la communication, les périphériques, etc. Chacun de ces services a sa propre interface qui doit être connue par une application avant de pouvoir l'utiliser.

La figure 2 montre un schéma bloc de l'architecture de ContikiOS :

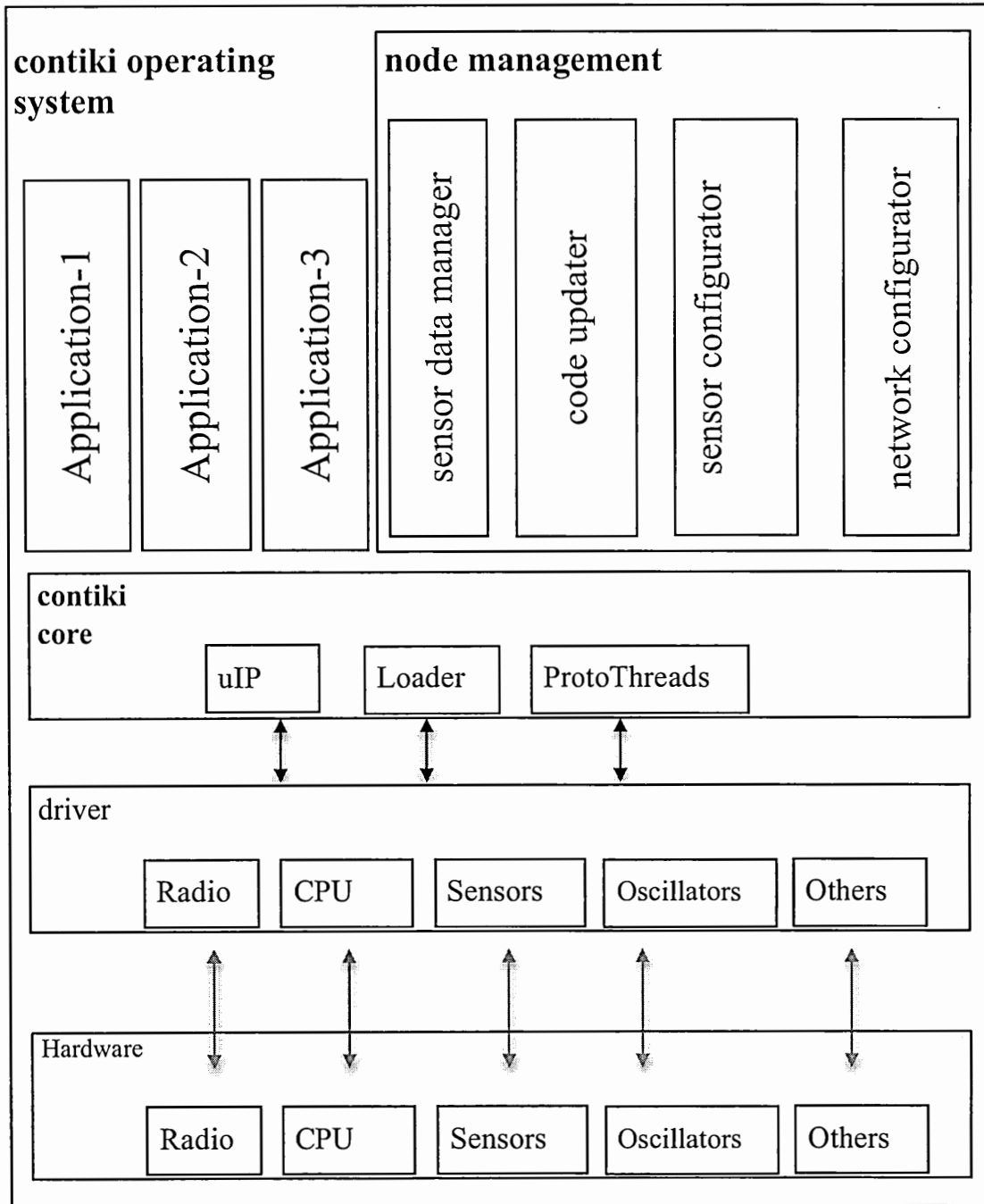


Figure 2 Schéma bloc détaillant les composants de ContikiOS

(Farooq et Kunz, 2011)

ContikiOS supporte un large éventail de protocoles de communications, notamment IPv4 et IPv6. Il comporte également une implémentation appelée MicroIP notée μ IP et une pile TCP/IP pour les microcontrôleurs 8 bits. Étant données les contraintes de mémoire dont souffrent les systèmes embarqués, μ IP utilise l'espace mémoire de manière efficace grâce à un mécanisme de gestion dédié. En effet, μ IP n'utilise pas d'allocation de mémoire explicite, mais plutôt un système de mémoire tampon global qui permet de retenir les paquets de données entrants. À l'instant où un paquet de données arrive, ContikiOS le place dans une mémoire tampon globale et notifie la pile TCP/IP afin d'accéder à l'application cible. Une fois la donnée traitée, Contiki écrase le paquet de données avec les nouveaux paquets entrants. Ce mécanisme permet d'économiser énormément d'espace mémoire.

b) TinyOS

Ce système d'exploitation modulaire, intégré, dédié aux réseaux de capteurs WSN développé à l'université Berkeley de Californie pourrait parfaitement être utilisé dans le cadre de notre projet dû à son empreinte mémoire très faible. Il supporte une programmation événementielle et comporte une librairie de composants incluant des protocoles réseau, des services distribués, des périphériques de capteurs et des outils d'acquisition de données.

L'architecture de TinyOS est monolithique utilisant un modèle de « composants » qui consiste à avoir une entité de calcul indépendante avec plusieurs interfaces d'entrée/sortie. Ces entités peuvent opérer entre elles en utilisant un ordonnanceur afin de composer une image qui tourne sur le microprocesseur. Les versions actuelles de TinyOS appelées « TOS » supportent le multithreading, la gestion des ressources de mémoire protégée ainsi que deux protocoles à multi-sauts : TYMO et Dissemination (Levis *et Al.*, 2005).

L'architecture de TinyOS peut être représentée par le schéma bloc de la figure 3 :

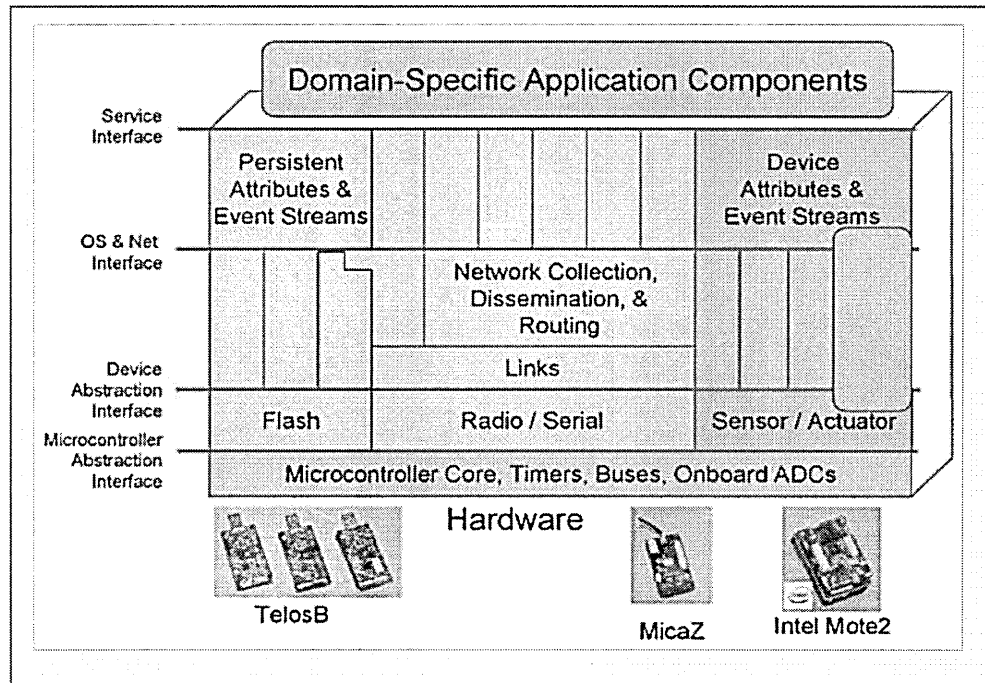


Figure 3 Schéma bloc détaillant les composantes de TinyOS (Farooq et Kunz, 2011)

c) LiteOS

C'est un système d'exploitation basé sur Unix, développé à l'université de l'Illinois à Urbana-Champaign. LiteOS a vu le jour en réponse au besoin de créer un système d'exploitation embarqué basé sur Unix. Le but de LiteOS est de faciliter la programmation aux utilisateurs Unix. Il supporte un système de fichiers et le multithreading et il est orienté objet. Très optimisé, le noyau de LiteOS requiert très peu d'espace mémoire. Il peut d'ailleurs tourner sur le nœud (mote) MicaZ qui ne comporte que 128 Kilo-octets de mémoire flash et 4 Kilo-octets de mémoire vive seulement.

L'architecture modulaire de LiteOS est constituée de trois composantes principales : LiteShell, LiteFS et le noyau (Kernel). L'avantage de LiteOS est que son LiteShell est

implémenté sur la station de base. Cette particularité aide à effectuer des commandes de support beaucoup plus complexes que celles de ContikiOS ou TinyOS, car la station de base n'est pas limitée en ressources. Cette caractéristique nous pousse à envisager LiteOS dans notre projet afin de limiter les sollicitations des nœuds pour économiser en espace mémoire et en énergie. La figure 4 est une représentation sous forme de schéma bloc de l'architecture de LiteOS.

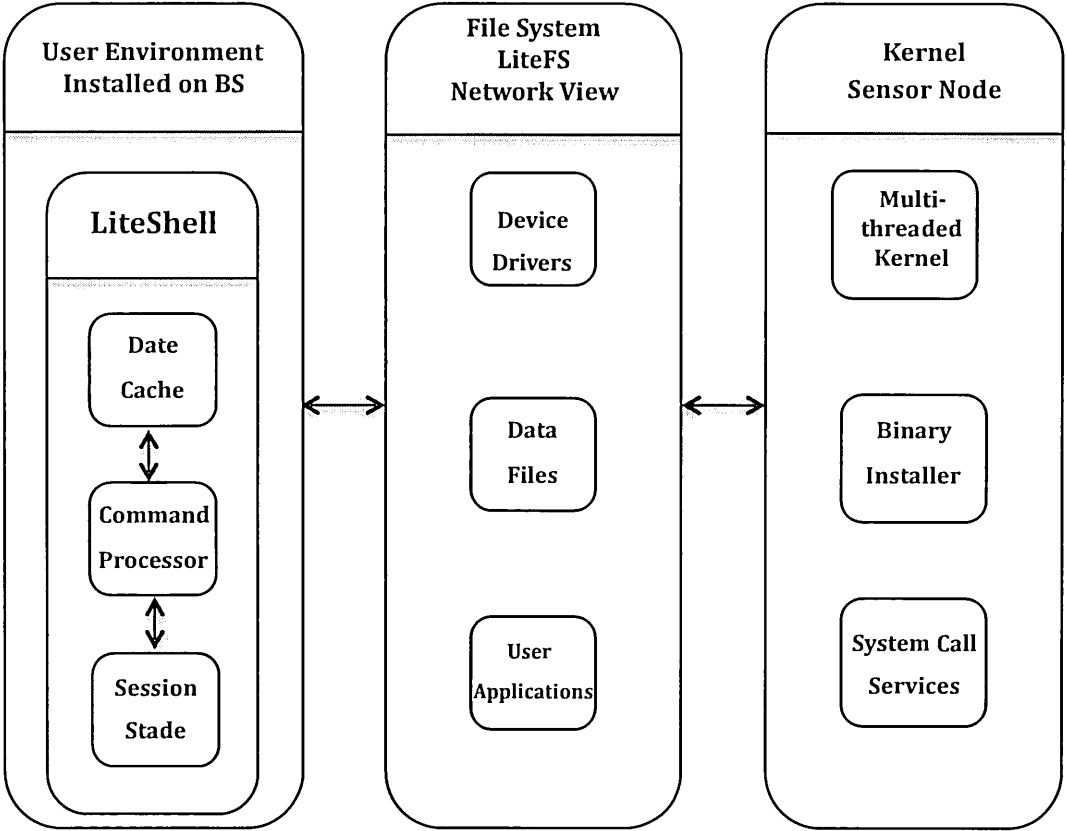


Figure 4 Schéma bloc détaillant les composantes de LiteOS (Farooq et Kunz, 2011)

d) FreeRTOS

FreeRTOS est un système d'exploitation temps réel en distribution libre dédié aux systèmes embarqués. Il supporte plusieurs architectures et reste en phase de développement depuis sa création par Richard Barry en 2002. Le code de FreeRTOS inclut la priorisation et l'agencement des tâches. À la différence des autres systèmes d'exploitation, FreeRTOS fonctionne en mode temps réel et contient un nombre de lignes de code très réduit. Son image binaire typique est d'une taille inférieure à 10 Kilo-octets et son code est divisé en trois parties majeures :

- les tâches sont des fonctions en langage C définies par l'utilisateur. Elles régissent le fonctionnement que les développeurs désirent implémenter. L'utilisation de FreeRTOS dans notre projet nous permettrait d'accorder plus d'importance aux fonctions qui traitent de l'énergie et de la mobilité.
- la communication représente 40 % du code de FreeRTOS et elle définit les règles de gestion des tâches à travers l'envoi de données.
- la partie matérielle : le code de FreeRTOS est principalement indépendant du matériel utilisé, ce qui lui permet de fonctionner sur plusieurs plateformes différentes.

1.2. Passerelle

Le terme passerelle (en anglais Gateway) est un terme générique qui désigne la manière avec laquelle une connexion est établie entre les réseaux de capteurs (environnement local) et le monde extérieur (Internet).

La passerelle est une brique essentielle dans l'architecture de l'IoT. Elle peut être sous forme logicielle ou matérielle, centralisée ou distribuée, déployée au niveau des nœuds ou encore dans un équipement indépendant. Les auteurs de (Kosanovic et Stojcev, 2011) ont procédé à une classification des différents types de passerelles et leurs caractéristiques. La figure 5 reprend cette classification.

1.2.1. Architecture avec serveur mandataire

Dans ce modèle, la communication est effectuée entre les utilisateurs TCP et les capteurs à travers un serveur mandataire ou « proxy ». Le protocole de communication utilisé dans le réseau de capteurs peut être choisi librement.

1.2.2. Architecture basée sur la superposition

Dans cette méthode, il existe deux types d'architectures : « TCP/IP overlay sensor networks » et « Sensor networks overlay TCP/IP ». La première approche consiste à implémenter le protocole TCP/IP sur une plateforme disposant de peu de ressources (Microprocesseur à 8 bits avec 2kB de mémoire vive). Dans la deuxième approche, la pile de protocoles utilisée dans le réseau de capteurs est déployée sur le protocole TCP/IP où les utilisateurs d'Internet sont considérés comme des nœuds virtuels. Un nœud virtuel peut interpréter les paquets du réseau de capteurs du moment que le protocole du WSN est installé dessus en plus du protocole TCP/IP. Il existe cependant plusieurs problèmes associés à cette dernière implémentation. Notamment, une haute consommation énergétique et une charge de réseau causée par les entêtes des paquets.

1.2.3. Architecture de passerelle

Dans cette architecture, la passerelle gère la communication entre les réseaux de capteurs et le réseau TCP/IP. Elle est responsable de plusieurs tâches telles la conversion de protocoles, le délai de message, etc. Toutes les architectures qui utilisent les passerelles comme moyen d'interconnexion peuvent être regroupées dans les deux catégories suivantes :

- passerelle d'application (en anglais application gateway) : Cette catégorie de passerelle est essentiellement opérationnelle au niveau de la couche d'application.
- réseau tolérant au délais (en anglais delay tolerant network ou DTN) : Cette catégorie est similaire à la précédente, sauf qu'elle implémente une nouvelle couche

au niveau de la couche TCP/IP dont la fonction principale est de stocker et transférer les paquets entre les deux couches.

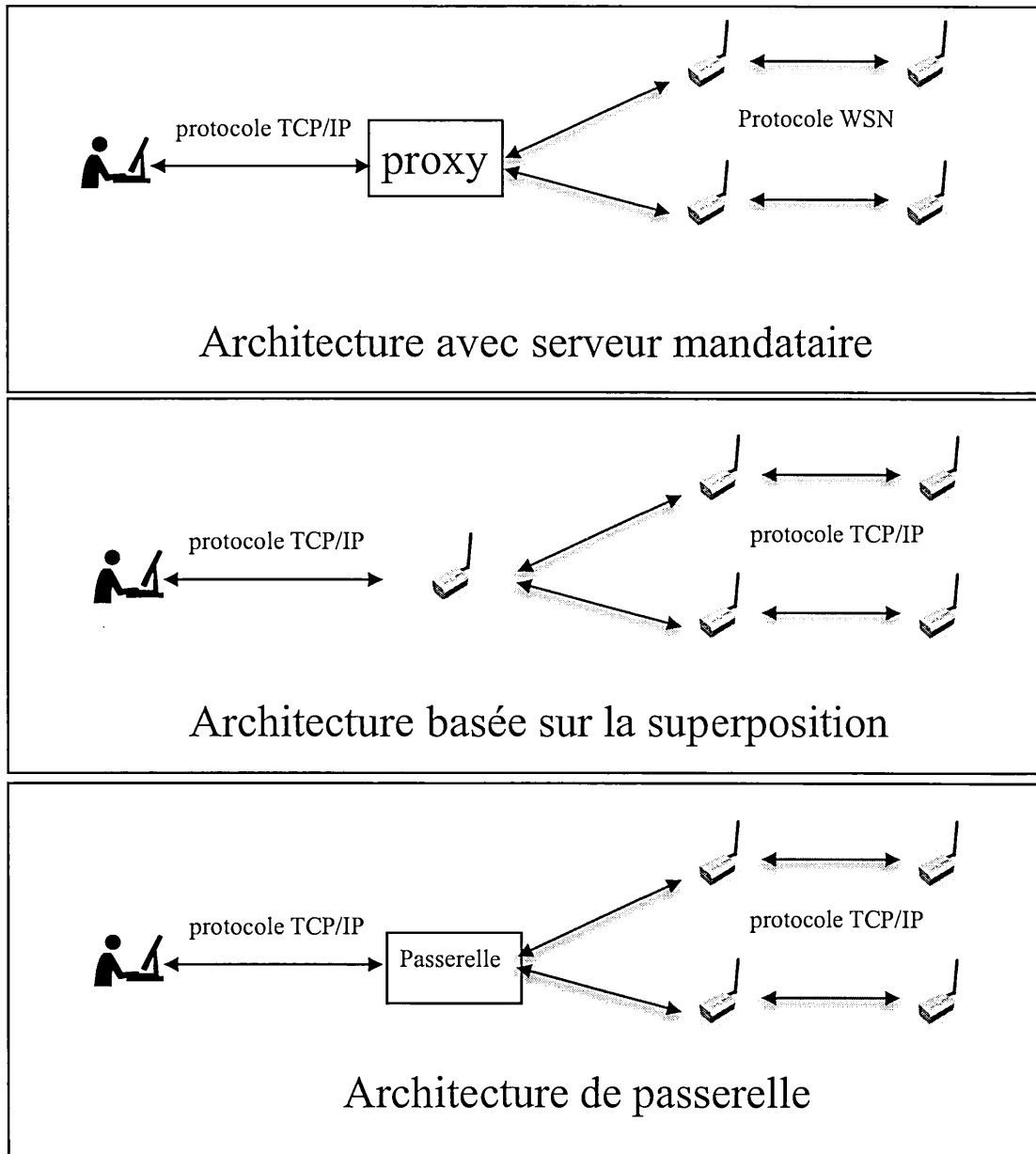


Figure 5 Illustrations de différentes architectures de passerelles utilisées dans le domaine de l'Internet des objets

1.3. Pourquoi étudier la mobilité dans IoT ?

Les exemples qui nécessitent la mobilité dans l'IoT sont multiples et variés. Nous en citons quelques-uns ci-dessous afin de mieux comprendre l'utilité d'une telle technologie. Parmi ces exemples, il y en a qui sont implémentés et opérationnels depuis quelques années et d'autres qui restent des idées fortes intéressantes qui pourraient voir le jour.

1.3.1. Villes intelligentes :

- bicyclettes et voitures collectives (Bixi, Communauto, etc..) : supervision des vélos et véhicules pendant leur utilisation ainsi que la collecte de données pour toutes sortes d'analyses et statistiques,
- festivals et événements : aide à l'orientation et gestion de la foule dans des places désignées telles que les stades ou salles de spectacles, grâce à des bracelets ou tickets d'accès,
- stationnements intelligents : gestion des véhicules, ainsi que les places de stationnement en temps réel.

1.3.2. Gestion intelligente de l'environnement :

- supervision des animaux : collecte d'informations sur les animaux en milieu sauvage ou dans des réserves protégées,
- localisation de bétail dans les pâturages et les étables.
- actifs et Machines d'agriculture : supervision de tracteurs et camions agricoles.

1.3.3. Sécurité et urgences :

- marathons et courses urbaines : supervision de l'état des coureurs pour la relève d'informations et une rapidité d'intervention en cas d'urgences,

- sécurité des enfants et personnes âgées dans les milieux de la petite enfance et les maisons de vieillesse respectivement.

1.3.4. Distribution et logistique :

- contrôle de marchandises : supervision des colis délicats,
- gestion des conteneurs : en émettant des signaux d'alerte dans les centres d'entrepôts portuaires afin d'éviter la proximité des marchandises inflammables et explosives par exemple,
- qualité de fret : supervision de la chaîne de froid des aliments, supervision des vibrations des marchandises délicates à des fins d'expertise d'assurance notamment,
- centres d'achats et grandes surfaces : supervision des caddies et collecte de données sur les tendances de consommation des clients.

1.3.5. Contrôle industriel:

- applications M2M (Machine à machine) : supervision des machines mobiles en milieu industriel,
- localisation intérieure : localisation des actifs au sein d'un espace industriel,
- mines : supervision des mineurs et collecte d'informations relatives à leurs métabolismes.

1.3.6. Immeubles intelligents et domestiques:

- supervision des enfants et personnes âgées en milieu domestique et collecte de données relatives à leur état de santé,
- gestion du personnel : supervision et localisation des différents membres du personnel au sein d'un immeuble (Hotel, administration, etc..).

1.3.7.Santé :

- hôpitaux : surveillance de l'état des patients en temps réel tout en leur permettant une mobilité au sein de l'hôpital,
- diagnostiques et alertes : permettant une intervention rapide en cas d'urgence,
- diagnostic distant : les médecins et infirmiers ont la possibilité d'établir des diagnostics instantanés à distance.

2. Problématique du mémoire

Les domaines d'application précédents montrent l'importance que joue la mobilité dans l'Internet des objets. Le besoin d'accès à distance aux objets en tout temps a propulsé l'IoT au-devant de la scène. De multiples utilisations peuvent être imaginées comme le contrôle d'une maison à distance ou la supervision d'un patient quand celui-ci est en libre mouvement dans un hôpital. Cependant, cette technologie repose principalement sur des réseaux de capteurs caractérisés par des carences énergétiques et des connexions instables. Ces limitations créent beaucoup de problèmes relatifs à la perte de données et la surconsommation énergétique. Par conséquent, si la mobilité dans les réseaux de capteurs sans fil est implémentée de manière non optimale, elle pourrait provoquer des déconnexions et isoler des régions du réseau à cause de l'épuisement énergétique des nœuds, provoquant ainsi un déséquilibre du fonctionnement du réseau et un taux de perte de données élevé. Nous nous posons donc la question afin de savoir quel serait le meilleur moyen d'acheminer le maximum d'information dans un scénario d'IoT doté de mobilité ? Autrement dit, comment éviter les déconnexions et l'isolement des nœuds causés par une surconsommation de l'énergie ? Un nœud mobile devrait constamment trouver un nœud d'attache pour acheminer l'information, et ce, durant toute la durée de vie du réseau. Dans l'exemple cité précédemment, la collecte d'informations de diagnostic du patient est critique et elle doit être faite de manière continue au fur et à mesure que

le patient se déplace dans le campus hospitalier, car une perte de données dans ce cas pourrait être fatale.

Afin d'adresser cette problématique générale, nous la divisons en trois sous problèmes auxquels nous répondrons. Tout d'abord, quelle est la meilleure manière d'intégrer la gestion de la mobilité dans un réseau de capteurs sans fil ? Cette question prend en considération les caractéristiques des réseaux de capteurs afin d'acheminer les paquets de données envoyés par un nœud mobile en tout temps. Puis, nous nous intéresserons à la manière d'améliorer notre solution afin de prendre en considération l'aspect énergétique des nœuds. Comment équilibrer la consommation énergétique des nœuds du réseau ? Finalement, compte tenu du nombre de messages de contrôle engendré par une telle mobilité, nous aimerions savoir comment améliorer notre solution pour minimiser le nombre de basculements et ainsi limiter le nombre de messages de contrôles pour optimiser le taux d'acheminement des paquets ?

Le terme basculement, désigné également en anglais par « Handoff », représente la procédure suivie par le nœud mobile afin de trouver un nouveau nœud d'attache. Le nombre de messages de contrôle est également désigné par « charge de réseau » ou « en anglais overhead ». Il représente tous les messages échangés dans le réseau à l'exception des paquets de données et leurs accusés de réception.

3. Objectifs

L'objectif principal de cette étude est de rendre un nœud mobile accessible en tout temps, et ce, dans les deux sens. Idéalement, la solution proposée doit permettre au nœud mobile d'envoyer des paquets de données de manière effective. Elle doit également permettre de le localiser à tout moment afin qu'il reçoive des paquets à son tour. Pour ce faire, nous procédons par étapes en nous fixant deux sous-objectifs qui répondent aux questions posées dans la section précédente. Le premier est

d'équilibrer la consommation énergétique des différents nœuds au fur et à mesure que le nœud mobile se déplace. Le deuxième sous-objectif est d'améliorer notre solution afin de minimiser le nombre de basculements et les déconnexions que ce dernier provoque.

4. Hypothèses de recherche

Dans le cadre de notre recherche, nous considérons certains acquis afin de mettre en contexte le périmètre de recherche. Les hypothèses émises nous aident à résoudre les sous-problèmes cités précédemment, et pour cela, nous associons aux sous-problèmes des hypothèses comme suit :

Étant donné le changement de topologie fréquent engendré par la mobilité, nous considérons que la force du signal mesuré à la destination est un indice suffisamment fiable pour nous renseigner sur la qualité d'un lien. Cependant, afin de compléter notre hypothèse, nous proposons dans notre solution un mécanisme de validation de la qualité du lien. Les protocoles de mobilité classiques utilisés pour des réseaux IP standards ne sont pas adaptés aux contraintes énergétiques des réseaux de capteurs sans fil. C'est pour cette raison que nous devons choisir un protocole de routage dédié aux réseaux de capteurs que nous améliorons afin de prendre en considération les contraintes énergétiques. La deuxième hypothèse prend comme acquis le fait que le protocole de routage soit en mesure d'acheminer tous les messages transmis par le nœud mobile une fois que celui-ci choisit un nœud d'attache. En effet, il est possible d'intégrer un mécanisme de gestion de la mobilité sur un protocole de routage proactif sans porter atteinte au fonctionnement de celui-ci. Le protocole de routage sera en mesure de répondre aux multiples changements de topologie du réseau de manière rapide et efficace.

5. Limitations de la recherche

Il existe différentes approches pour aborder la problématique générale énoncée. Dans le cadre de notre travail, des scénarios prédéfinis sont utilisés pour mieux baliser le champ de recherche. L'étude proposée a été réalisée sur la base d'une simulation de réseaux de capteurs sans fil dotés de nœuds mobiles. Le terme mobilité représente le déplacement d'un nœud suivant une trajectoire prédéfinie. Cependant, ce déplacement se fait au sein d'un même réseau. Par conséquent, la mobilité interréseaux n'est pas couverte dans ce travail.

La solution proposée repose sur un protocole de routage proactif qui aide à acheminer les données en multi-sauts dans une topologie dynamique. Dans notre étude, notre choix se porte sur le protocole RPL. Ce choix sera justifié dans le chapitre IV. Ce protocole doit construire la topologie du réseau et constamment définir des routes vers le nœud principal afin d'éviter la perte de données. Le scénario simulé consiste à collecter les données envoyées par le nœud mobile puis accuser réception de leur acheminement. Le contrôle à distance du nœud mobile n'est pas étudié dans ce travail.

- Le protocole RPL

RPL est un protocole à vecteur de distance dédié aux réseaux de capteurs sans fil à faible capacité énergétique (en anglais Low-Power and lossy Networks, LLN). Il est défini comme standard par l'IETF dans la RFC 6550 (Winter *et al*, Request for Comments, mars 2012) et décrit comment construire un graphe orienté acyclique (en anglais Destination Oriented Directed acyclic Graph ou DOADG) en utilisant un jeu de fonctions objectives. Ces fonctions opèrent sur la base de métriques et de contraintes pour définir le meilleur chemin possible. Les fonctions objectives sont multiples et dépendent des prérequis fixés. Les graphes construits par RPL représentent des topologies logiques déployées sur des réseaux physiques afin de satisfaire un besoin particulier. Plusieurs topologies logiques peuvent être déployées

en même temps. Les nœuds qui participent à la constitution de ces topologies peuvent véhiculer le flux de données selon des critères différents et marquer ces flux afin de satisfaire des critères de qualité de service QoS différents. Ces topologies sont appelées des « Instances ».

La construction du graphe DODAG commence au niveau du nœud de bordure (en anglais LowPAN Border Router ou LBR) configuré préalablement par le développeur. Il peut y avoir plusieurs LBR dans un même réseau. RPL définit trois types de messages de contrôles ICMPv6 (en anglais Internet Control Message Protocol version 6) afin de véhiculer l'information relative au réseau entre les différents nœuds. Ces messages sont appelés DIS (en anglais DODAG Information Sollicitation), DIO (en anglais DODAG Information Object) et DAO (en anglais DODAG Destination Advertisement Object).

Le nœud LBR commence à envoyer les informations relatives au réseau en utilisant des messages DIO. Les nœuds voisins reçoivent ces messages et les traitent selon les exigences de la fonction objectif. Une fois qu'un nœud rejoint le graphe, il calcule son rang par rapport au rang de son nœud parent. Ce rang sera considéré dans le réseau comme étant les coordonnées du nœud. Cette opération continue jusqu'à atteindre le nœud d'extrémité (en anglais Leaf Node) et le graphe sera construit en totalité.

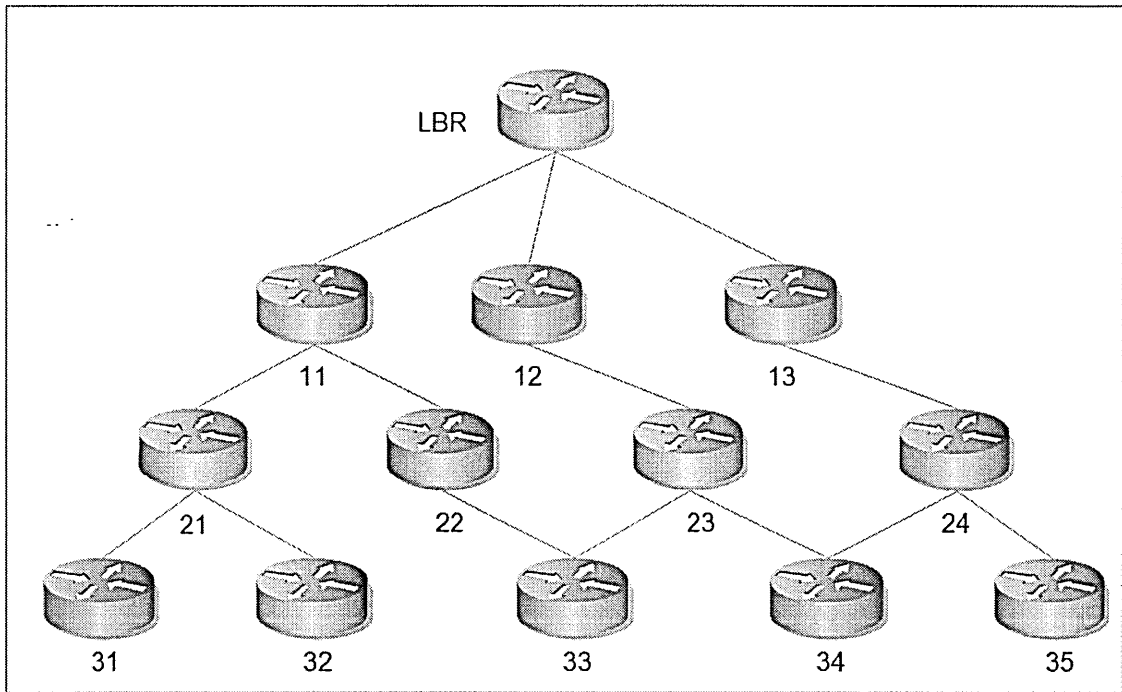


Figure 9 Illustration du graphe DODAG du protocole RPL (Vasseur *et al.*, 2011)

RPL dispose de plusieurs types de trafic qui donnent une certaine flexibilité dans le montage d'une solution. Les types de trafics disponibles sont : un trafic ascendant (en anglais multipoint to point) (Vasseur *et al.*, 2011), un trafic descendant (en anglais point to multipoint) ainsi qu'un trafic point à point. Le trafic descendant est généralement utilisé pour satisfaire une requête provenant de l'extérieur du réseau et qui aurait comme destinataire un nœud d'extrémité particulier. RPL dispose également d'un mode de fonctionnement appelé « mode sauvegrade ». Nous utiliserons ce mode dans notre projet pour allouer suffisamment de mémoire aux nœuds de capteurs afin de stocker les informations dans leur table de routage. RPL emploie un algorithme appelé Trickle (en anglais Trickle Algorithm) (P.Levis, Request for Comments, mars 2011). Cet algorithme aide à envoyer des messages DIO à un intervalle incrémental. Cet intervalle s'incrémente au fur et à mesure que le

réseau se stabilise. Par conséquent, moins de messages DIO seront envoyés dans le réseau. En cas de présence de boucle ou de lien inconsistant, le minuteur du « trickle algorithm » se re-initialise pour permettre un envoi immédiat de messages DIO pour reconstituer le lien. Cette méthode permet une utilisation plus judicieuse de l'énergie et garantit une certaine proactivité en particulier dans les réseaux stables et à faible mobilité.

RPL a été développé afin de supporter les réseaux 6LoWPAN dans une configuration route-over. L'association des deux protocoles ne requiert pas de considérations particulières d'où l'intérêt de l'utiliser dans notre projet.

CHAPITRE II

REVUE DE LA LITTÉRATURE

Dans ce chapitre, nous analysons différents travaux réalisés afin de positionner notre problématique et ses différents aspects. Les travaux sont regroupés en deux groupes distincts. Le premier groupe couvre les travaux traitant de la mobilité dans les réseaux WSN. Nous analyserons ces travaux afin de comprendre comment la mobilité des nœuds a été implémentée et quels impacts cela a eu sur les performances du réseau. Le deuxième groupe traite les travaux sur la conscience énergétique (en anglais *energy awareness*) dans les WSN, la manière avec laquelle ces réseaux détectent l'énergie et l'incidence que cela a sur le bilan énergétique global.

1. Mobilité dans les réseaux WSN

1.1. La solution dite : « RPL under mobility »

Dans (Kevin *et Al.*, 2012), les auteurs étudient un scénario de mobilité dans un contexte de réseaux véhiculaires (en anglais *Vehicular ad hoc Network*, ou VANET), une sous-catégorie des réseaux Ad-hoc (en anglais *Mobiles Ad-hoc Networks*). Il sert de standard à un groupe de véhicules qui communiquent entre eux, ainsi qu'avec des unités en bordure de route (en anglais *Road Side Units* ou RSU).

Des véhicules équipés d'unités mobiles (en anglais *Mobile Onboard Units* ou OBU) comportent de multiples interfaces sans fil pouvant supporter diverses technologies de communication, à savoir : 3G, 4G, Wifi, etc. Ces interfaces permettent une communication « véhicule-véhicule » et « véhicule-RSU ». Le but est de construire une topologie en arbre (en anglais *Tree Based Topology*) qui change au fur et à mesure que les véhicules se déplacent.

Ce travail de recherche suppose également que les véhicules peuvent avoir différents états de mobilité, par exemple : autoroute, congestions en milieu urbain, etc. De ce fait, les auteurs optent pour le protocole de routage dit RPL (en anglais Routing Protocol for low-power and lossy networks) (Winter *et Al*, Request for Comments, mars 2012) qui permet de construire une architecture multisaut arborescente afin de couvrir une large zone géographique à moindre coût.

L'architecture implémentée considère les RSUs comme des noeuds principaux. Les OBUs essayent de se connecter aux noeuds principaux via RPL, soit directement, ou indirectement en passant par d'autres OBUs (multi-sauts). Pendant la phase d'initiation, le noeud principal commence à construire la topologie du réseau en envoyant des messages d'informations appelés DIO (ou Directed Acyclic Graph Information Option), contenant les coordonnées de chaque véhicule. Une fois qu'un noeud (véhicule) reçoit un message d'information d'un noeud voisin, il calcule son rang et le coût du chemin pour atteindre ce voisin. Ce coût s'exprime en nombre de retransmissions estimées (ETX). Il est à noter que si un noeud possède un rang inférieur à celui de son voisin, il sera systématiquement considéré comme parent d'attache préféré par le noeud voisin (en anglais Preferred Parent). Une fois la topologie stable, si un noeud découvre un nouveau voisin avec un rang inférieur, il le considérera systématiquement comme nouveau parent d'attache préféré. Dans cette solution, le changement de parent d'attache préféré est dû au simple fait qu'un autre parent existe dans la zone de couverture. Ce sondage est réalisé grâce à un « minuteur » qui permet d'envoyer des messages à intervalle régulier afin de vérifier l'existence d'un parent potentiel.

L'architecture proposée est évaluée en utilisant le simulateur Qualnet 4.5 et le protocole MAC 802.11a. Le taux de transmission est fixé à 24 Mb/s et la puissance de transmission à -80dBm. La portée de la transmission est de 250 mètres et la vitesse des véhicules est de 65 MPH.

Les auteurs commencent par vérifier le protocole RPL modifié en observant le changement de rangs des voitures par rapport au temps, à la vitesse des véhicules, ainsi qu'au taux d'acheminement des paquets (en anglais Packet Delivery Ratio ou PDR). La charge du réseau et le débit moyen ont également fait l'objet d'évaluations. Comme deuxième phase de test, les auteurs ont désactivé l'algorithme Trickle. Ce dernier est un algorithme qui maintient la topologie. Les simulations ont donné les résultats suivants :

- la vitesse des véhicules a un impact direct sur les rangs. En effet, plus la vitesse augmente, plus vite les rangs des nœuds changent.
- plus l'intervalle des ETX augmente, plus le taux d'acheminement et la charge du réseau diminuent. Ceci est dû au délai important que prend un nœud mobile pour sonder son parent. Par conséquent, le débit diminue pour laisser place à un délai moindre, dû au fait que les paquets reçus n'effectuent pas beaucoup de sauts.
- les variations d'intervalles des messages de contrôles en même temps que la vitesse des véhicules ont un effet immédiat sur le taux d'acheminement, le débit et le délai.

1.2. Co-RPL

Le travail (Gaddour *et Al.*, 2014) propose Co-RPL qui est une extension de RPL. Co-RPL supporte la mobilité en gardant la traçabilité des nœuds mobiles afin d'accroître la performance du réseau. Cette approche repose sur un mécanisme appelé « CORONA » qui permet la localisation des routeurs RPL en mouvement. Ce mécanisme divise le réseau en zones circulaires appelées DAG avec un routeur au centre de chaque zone appelé « DAG router ». Co-RPL opère selon deux algorithmes différents. Un premier algorithme est déployé au niveau du DAG router pour construire le sous-réseau DAG. Celui-ci est découvert grâce à un échange de messages d'informations et de sollicitations.

Le deuxième algorithme est déployé sur les nœuds mobiles. Chaque nœud écoute le réseau de manière continue afin de détecter des messages d'information (DIO). Si aucun DIO n'est reçu, le nœud en question sera isolé du réseau et ne cessera d'envoyer des messages de sollicitations DIS jusqu'à la réception d'un DIO. Une fois ce dernier reçu, le nœud mobile calcule son rang et envoie un DIO en diffusion au reste du réseau afin d'informer les autres nœuds de ces nouvelles coordonnées. Si jamais plusieurs DIO sont reçus, le nœud mobile choisira le DIO comportant le moindre ID et avec la meilleure qualité de lien.

Les tests de simulation ont été conduits sur le simulateur Cooja, avec les nœuds Tmote Sky. La puissance de transmission a été fixée au maximum de 0dBm et les rayons des Corona à 50 m. Les résultats des simulations ont montré que Co-RPL permettait de diminuer le taux de pertes des paquets jusqu'à 45 % et la consommation énergétique de 50 % comparativement à RPL. Il permet également de réduire les délais de transmissions des paquets de 2,5 seconds de bout en bout.

1.3. MoMoRo

Dans (Ko et Chang, 2014), les auteurs proposent une couche de support de la mobilité appelée MoMoRo afin d'assurer un taux d'acheminement de paquets élevé. Cette couche intermédiaire peut être implémentée dans différents protocoles de routage. Selon l'étude, MoMoRo n'introduit pas de charge de réseau considérable quand les nœuds sont stationnaires. Sa fonction principale est de fournir une estimation robuste de la qualité des liens une fois des changements de topologies détectés.

MoMoRO supporte la mobilité des nœuds en réalisant deux opérations majeures. Premièrement, il fait en sorte de réagir rapidement à la perte des paquets en reconstruisant la route par défaut ou en notifiant que celle-ci n'est pas disponible. Deuxièmement, il reconstruit la route par défaut. Il regroupe rapidement les informations et les fait passer par un algorithme d'estimation flou afin de calculer les

bonnes et les mauvaises routes et ainsi aider la couche réseau dans sa construction de la table de routage.

Une fois qu'un nœud transmet un paquet de données plusieurs fois au même parent et que le nombre maximal de transmissions tolérées est atteint, la couche de routage notifie le mécanisme MoMoRo qu'un parent est éventuellement hors de portée. MoMoRo tente alors de renvoyer le paquet de données une dernière fois en mode monodiffusion. Cette dernière tentative permet d'éviter toute charge de réseau additionnelle due à la reconstruction de la route si la déconnexion n'est que temporaire. Si ce n'est pas le cas, MoMoRo diffuse à ses nœuds voisins un message (COUNTBroadcast) et les nœuds qui reçoivent ce message accusent réception avec un message unicast. Partant des réponses reçues, le nœud émetteur choisit le meilleur parent d'attache en utilisant un algorithme d'estimation flou avec les 3 paramètres suivants : RSS (Received Signal Strength), le taux d'erreur de symbole (en anglais Symbol Error Rate Variance, ou SER Variance) et ETX.

L'évaluation de la solution proposée est faite à travers une validation interne grâce à un banc d'essai avec 30 nœuds Tmote Sky distribués dans un étage d'immeuble. Les Tmote Sky comportent des radios Chipcon TI CC2420 IEEE 802.15.4. Le tout fonctionne en utilisant le système d'exploitation TinyOS pour collecter les RSS et SER. La solution est évaluée en utilisant trois paramètres de performances. Le premier est le taux de réception de paquets (PRR) mesuré à la destination finale. Le deuxième est le nombre de retransmissions. Ce critère représente l'efficacité des routes sélectionnées. Finalement, le troisième paramètre est la charge du réseau qui permet de superviser le coût nécessaire pour des routes de qualité.

Les expériences MoMoRo sont conduites puis comparées à RPL avec une fonction objectif MRHOF (Minimum Rank with Hysteresis Objective Function) ainsi qu'avec le protocole de routage dit AODV (Ad-hoc On-demand Distance Vector) ou (protocole vecteur de distance à la demande Ad-hoc). Les résultats montrent que

MoMoRo et AODV présentent de meilleures performances que RPL en termes de taux de réception de paquets et de charge de réseau. Les auteurs concluent que MoMoRo permet d'améliorer le PRR de RPL avec une augmentation minimale de la charge du réseau.

1.4. RPL avec mobilité améliorée

L'étude (El Korbi *et Al.*,2010) vise à améliorer RPL afin qu'il puisse supporter des nœuds mobiles. La solution proposée, nommée ME-RPL, apporte trois changements majeurs à RPL :

- annonce explicite de mobilité : les nœuds mobiles sont identifiés avec des tags et communiquent constamment leur position au fur et à mesure qu'ils se déplacent.
- amélioration du mécanisme de sélection du parent d'attache : l'algorithme de sélection de parents est modifié afin de privilégier les nœuds fixes plutôt que ceux mobiles et cette information est enregistrée pour être réutilisée au prochain passage. Cette méthode s'apparente à un algorithme d'apprentissage.
- adaptation de la vitesse des messages de sollicitation : ceci permet aux nœuds qui perdent leur lien avec un parent d'attache de solliciter une nouvelle liaison rapidement grâce aux nœuds voisins qui peuvent aider à partager leurs propres routes.

L'évaluation de ME-RPL est réalisée en utilisant le simulateur COOJA et une implémentation standard de RPL. Le système d'exploitation utilisé est ContikiOS. Les auteurs de cette recherche ont évalué ME-RPL par rapport à RPL Standard, et ce, en évaluant le taux de pertes de paquets (en anglais Packet Loss Rate) et la stabilité des routes. Les résultats montrent que ME-RPL présente un taux de perte de paquets plus bas comparé à RPL.

Pour appuyer leur solution, les auteurs ont évalué la stabilité des routes en dérivant le pourcentage du temps durant lequel chaque route est valide et peut être effectivement utilisée pour transporter des paquets de données. Les résultats montrent que le pourcentage de routes stables est toujours plus élevé avec ME-RPL par rapport à RPL, quel que soit le nombre de nœuds présents dans le réseau. Ceci peut être expliqué par le fait que ME-RPL favorise les nœuds fixes plutôt que les nœuds mobiles dans son choix de parents d'attache.

1.5. Réseau des proxy (Network of Proxies)

Dans l'étude (Silva *et Al.*, 2012), les auteurs démontrent que l'utilisation de Mobile IPv6 (MIPv6) dans les réseaux WSN provoque des délais considérables pendant la phase de basculement, sans compter la complexité et la consommation énergétique excessive. Ils proposent alors un mécanisme de gestion de la mobilité basée sur des serveurs mandataires qui jouissent d'une énergie infinie. Cette solution appelée réseau de proxys (en anglais Network of Proxies, ou NoP), gère la mobilité des nœuds sans interférer avec les messages de contrôle du réseau WSN. NoP est implémentée sur un équipement indépendant qui est responsable de superviser le niveau de RSSI du nœud mobile dit indicateur de puissance du signal reçu (en anglais Received Signal Strength Indication) et partage cette information avec les autres serveurs mandataires. L'analyse de ces informations aide à sélectionner le prochain nœud parent à utiliser. L'évaluation de cette solution a été réalisée en comparant entre MIPv6 et NoP dans un milieu de simulation et sur un banc d'essai en utilisant différents nœuds mobiles et noeuds passerelles. Les résultats de simulations où le protocole MIPv6 a été utilisé démontrent que le temps de basculement est très élevé et la consommation énergétique importante. Ces inconvénients sont dus à la complexité du protocole. Les simulations utilisant NoP, quant à elles, montrent de meilleurs résultats en termes de temps de basculement et d'énergie.

1.6. mRPL

Dans l'étude (Fotouhi, 2014), les auteurs démontrent que les deux mécanismes qui gèrent les changements dans RPL, à savoir le l'algorithme Trickle et la découverte des voisins (en anglais Neighbor Discovery) sont des mécanismes réactifs. Par conséquent, ils ne sont pas préparés aux exigences dynamiques d'un réseau mobile. Une solution est proposée afin d'intégrer un mécanisme proactif (Smart-Hop) permettant de supporter la mobilité par RPL. Smart-Hop permet de réduire le taux de perte de paquets et le délai de basculement.

L'algorithme Smart-HOP proposé comporte deux phases principales :

- phase de transmission de données : dans cette phase, le nœud mobile est censé avoir une liaison fiable avec son parent d'attache. Il supervise la qualité du lien grâce aux paquets reçus du parent d'attache pendant un intervalle de temps. Une fois ce temps écoulé, il accuse réception au nœud mobile. Si aucune réponse n'est reçue ou si le lien est rompu, le réseau passe à la deuxième phase de l'algorithme.
- phase de handoff : après constat de dégradation du lien, le nœud mobile envoie une rafale (en anglais burst) de messages de sollicitations DIS aux nœuds voisins. Chacun de ces derniers répond avec un message monodiffusion comportant une certaine valeur de RSSI relatant la qualité du lien qui le lie avec le nœud mobile. Sur cette base, et une fois toutes les réponses reçues, le nœud mobile sélectionne le meilleur parent d'attache et revient à la phase de transmission de données.

La solution proposée a été testée en utilisant ContikiOS 2.6.1 sur le simulateur Cooja où trois scénarios ont été évalués. Les résultats montrent un taux d'acheminement de paquets nettement supérieur pour les trois architectures comparées à celui de RPL. De même pour la charge du réseau qui est plus élevée dans le cas de réseaux à haute fréquence d'envoi de données (30 paquets/seconde). Les auteurs ont également

réalisé un banc d'essai en attachant un nœud Tmote Sky sur le corps d'un humain et défini une trajectoire parmi huit autres nœuds stationnaires répartis dans une chambre de 80 m². Les résultats de ce teste expérimental montrent de meilleurs taux d'acheminement de paquets comparé aux simulations. La charge de réseau quant à elle est nettement plus basse, car les zones de chevauchement dans le test expérimental sont plus importantes que dans les simulations et le modèle de propagation est également plus réaliste sur le terrain.

1.7. LoWMob

Dans ce travail de recherche (Bag, 2009), les auteurs présentent LoWMob, un schéma de mobilité destiné aux nœuds 6LoWPAN mobiles. LoWMob assure une communication multisaut entre le nœud mobile et la passerelle à l'aide des nœuds stationnaires. Afin de réduire la charge de réseau de ces derniers, LoWMob propose un format de paquets de support de mobilité au niveau de la couche d'adaptation 6LoWPAN. Les auteurs présentent également une version distribuée de LoWMoB nommée DLoWMoB qui utilise des points de support à la mobilité (en anglais Mobility Support Points ou MSP) afin de distribuer la concentration du trafic au niveau de la passerelle et optimiser le routage entre la source et la destination.

La performance de la solution proposée est évaluée en utilisant le coût de signalisation de la mobilité, le délai de la source à la destination ainsi que le taux d'acheminement de paquets. La détection de la mobilité est basée sur des transmissions fréquentes de messages de balise de la part des nœuds statiques et la mobilité est basée sur les principes conventionnels de Mobile IPv6. Cette étude considère également un cycle de service (en anglais duty cycle) des parents d'attaches où les radios sont éteintes puis allumées consécutivement. En observant une qualité de lien faible avec un parent d'attache, un nouveau parent est sollicité. Ceci est réalisé grâce à un mécanisme de localisation fourni par les MSP. De cette manière, la

direction du nœud mobile à un certain instant est déduite. Les résultats montrent que la solution proposée diminue la charge du réseau entre la passerelle et les nœuds stationnaires sans une implication excessive des nœuds mobiles. Également, l'étude a pu démontrer que l'utilisation de DLoWMob offre une meilleure performance que le LoWMob en termes de taux d'acheminement, de délais et de coût de signalisation de la mobilité.

2. Les réseaux conscients énergétiquement :

2.1. Energy Aware routing :

Dans (Shah et Rabaey, 2002), les auteurs considèrent qu'un des problèmes majeurs des protocoles de routage actuels réside dans le fait qu'ils se basent sur le chemin le moins coûteux énergétiquement. Cette approche peut mener à la surexploitation des batteries des nœuds et provoque des coupures dans le réseau. Afin de pallier au problème, les auteurs proposent un nouveau protocole de routage conscient énergétiquement. L'idée est d'accroître la durée de vie du réseau en choisissant les chemins de manière judicieuse, évitant ainsi l'utilisation excessive de certains chemins et conservant l'énergie des nœuds. Afin d'atteindre cet objectif, le protocole de routage découvre plusieurs routes entre la source et la destination et attribue une probabilité à chacune d'elle dépendamment de l'énergie. À chaque fois qu'un paquet de données doit être envoyé, un des chemins sera choisi aléatoirement basé sur la probabilité. Cette logique veut dire qu'aucun des chemins ne sera utilisé de manière excessive pour prévenir une consommation énergétique disproportionnée. Le protocole proposé comporte trois phases différentes :

- une phase d'inondation de messages (en anglais flooding) afin de trouver toutes les routes possibles.
- une phase de propagation de données où les informations sur l'énergie collectées par la phase d'inondation sont envoyées de la source vers la destination.

- maintenance de la route : cette opération n'est pas fréquemment effectuée. Un flooding local est envoyé afin de maintenir les chemins possibles vers une destination particulière.

La fonction objectif utilisée repose sur le coût de transmission et de réception d'un lien particulier ainsi que sur l'énergie résiduelle. Les simulations montrent que la solution proposée réduit la consommation énergétique moyenne de 21,5 % dû au taux de charge de réseau très bas et prolonge la durée de vie de 44 %.

2.2. Protocoles basés sur les négociations :

Les auteurs de (Vidhyapriya et Vanathi, 2002) présentent un protocole appelé « Energy Aware Routing » destiné à faire de manière fiable le routage des données avec une faible consommation énergétique. Il utilise l'énergie disponible et la qualité des liens afin d'identifier le meilleur chemin possible vers une destination. Les auteurs considèrent un réseau WSN statique déployé sur une région plate. Le protocole proposé a été développé afin d'aider à la prise de décision dans les choix des nœuds. Un nœud est sélectionné pour transférer les paquets de données sur la base du meilleur niveau d'énergie résiduelle et la distance qui le sépare du nœud expéditeur. Les nœuds qui ne sont pas sélectionnés se mettent en mode endormi instantanément afin de conserver leur énergie. Le protocole proposé définit des routes complètes de la source vers la destination et prépare différents chemins alternatifs afin de balancer le niveau d'énergie. L'algorithme proposé est basé sur trois phases : la première est la découverte de nœuds voisins. Cette étape se fait en créant une liste d'informations concernant les nœuds voisins. Le protocole collecte les identifiants des nœuds, l'énergie nécessaire pour les atteindre et le nombre de sauts. La deuxième phase est la réponse (en anglais route reply). Elle consiste en l'envoi d'un message de réponse de la part du nœud destinataire à l'issue de la sollicitation de l'expéditeur. Ce message comporte des informations comme le nombre de sauts nécessaires pour

atteindre la destination. La dernière étape, appelée table de routage (en anglais routing table), permet de regrouper toutes les routes découvertes pour d'éventuelles utilisations futures. Les données à analyser ont été obtenues à travers des simulations comparatives du protocole de routage AODV et la solution proposée. Les résultats montrent que la solution proposée permet d'atteindre de meilleures performances en termes de taux d'acheminement de paquets. Cette performance est maintenue même dans des environnements de haute densité de nœuds.

2.3. EAP :

Les auteurs de cette étude (Liu *et Al.*, 2009) proposent un protocole de routage écoénergétique, nommé « EAP », qui permet de prolonger la durée de vie du réseau en assurant une consommation énergétique minimale. EAP introduit des paramètres de clustering pour élire la tête du groupe (en anglais cluster head, ou CH) parmi les nœuds présents. Ce nœud est responsable de gérer les différentes capacités énergétiques disponibles dans le réseau. Le CH reçoit toutes les données envoyées par les différents nœuds du réseau à des fins de traitements. Il nécessite par conséquent une plus grosse consommation énergétique comparativement aux autres nœuds. Afin de pallier ce problème, les auteurs proposent d'élire un nouveau CH suivant un intervalle défini. Chaque intervalle commence par une phase d'initialisation réalisée une fois que la topologie du réseau est stable, puis une phase opérationnelle durant laquelle les données sont envoyées. Cette logique est réalisée en définissant des profils que les nœuds du réseau utilisent respectivement : nœud tête, nœud candidat, nœud membre, etc. où chaque profil de nœud répond à une fonction particulière.

EAP a été comparé à deux protocoles connus, LEACH (Low Energy Adaptive Clustering Hierarchy) (Heinzelman *et Al.*, 2000) et HEED (Hybrid, Energy-Efficient, Distributed Clustering Approach for Ad-hoc Sensor Networks) (Younsi *et Al.*, 2006), et les résultats des simulations montrent que EAP présente de meilleures

performances que LEACH. Il égale aussi celles de HEED dans des réseaux à faible densité avant de devenir plus performant dans des réseaux peuplés avec une densité minimale de 0.01 nodes/m^2 .

3. Analyse critique :

Nous procédons ici à l'analyse critique de quelques-uns des travaux cités afin de relever leurs faiblesses et incomplétudes. Nous commençons par ceux qui traitent la mobilité des nœuds dans les réseaux sans fil, puis nous nous pencherons sur les travaux qui visent la conscience énergétique. Nous retenons les points forts et les lacunes suivantes :

- RPL Under Mobility : le modèle de détection de la mobilité dans cette solution est obtenu en utilisant un minuteur fixe plutôt que le minuteur de l'algorithme Trickle. Ce modèle engendre un taux de transmission des messages DIO très important. Cette logique permet certes d'augmenter la connectivité, mais augmente aussi la charge du réseau et pousse le réseau à consommer beaucoup d'énergie. De plus, cette proposition désactive le minuteur de l'algorithme Trickle qui s'avère être très efficace en l'absence de mobilité.
- MoMoRo : le modèle de détection de la mobilité de cette étude repose uniquement sur la perte d'un paquet. Cette approche est passive et ralentit le temps de réponse du réseau. De plus, les liens entre les nœuds dans les réseaux WSN sont sujets à des déconnexions temporaires fréquentes. Ces dernières causent souvent des pertes de paquets. Par conséquent, la décision de basculement basée sur la perte de paquets pourrait engendrer des maintenances de routes non nécessaires et augmenter la charge du réseau.
- Mobility Enhanced RPL : l'algorithme d'apprentissage proposé par les auteurs ne permet pas de détecter les mouvements rapides d'un nœud mobile. Celui-ci se base

sur les antécédents géographiques d'un nœud afin d'opérer de manière efficace. Cette solution ne fournit donc pas un protocole réactif qui puisse supporter la mobilité.

- Réseau de proxys (NoP): cette approche nécessite un matériel supplémentaire appelé NoP à rajouter dans le réseau. La fonction principale de NoP est la gestion de la procédure de basculement. Le principal inconvénient de cette solution réside dans le fait que le NoP doit disposer d'une énergie illimitée afin de mener à bien sa mission. Une énergie dont nous ne disposons pas dans les réseaux WSN.
- Smart-Hop over RPL : cette solution ne prend pas en considération l'énergie disponible dans les nœuds. En effet, aucune politique de cycle de service n'est implémentée au niveau des nœuds que les auteurs considèrent comme ayant une énergie infinie.
- LoWMob : cette étude repose sur le standard mobile IPv6 qui s'avère être gourmand en échange de paquets ce qui implique un important taux de charge du réseau. Par conséquent, cette approche n'est pas très efficace dans les réseaux contraints en énergie.

Les travaux avec conscience énergétique : les travaux qui traitent de la conscience énergétique cités précédemment considèrent tous des topologies statiques. L'étude de ce genre de travaux nous permet de retenir certaines techniques utilisées afin de nous en inspirer dans le cas de la mobilité.

Des solutions précédentes, nous retenons certaines techniques qui permettent d'implémenter un scénario de mobilité ou encore de rendre un réseau statique écoénergétique. Nous reprendrons ces techniques dans le chapitre suivant. Cependant, nous constatons un manque de travaux de recherche qui traitent de problématiques similaires à la nôtre. Le fait de considérer la mobilité et l'énergie séparément ne

permet pas de tenir compte des leurs dépendances mutuelles ou de trouver un compromis pour des installations fiables et performantes. D'où l'idée dans ce travail de développer une solution qui répond aux exigences de la mobilité dans les réseaux WSN en prenant en considération l'énergie comme facteur clé. Nous illustrons les différences en quelques travaux étudiés dans le tableau 1-a et 1-b. Le chapitre III donne les détails de cette approche.

Tableau 1-a Résumé de quelques études sur les réseaux de capteurs sans fils mobiles

	Environnement	Détection de la mobilité	Sélection du noeud parent	Conclusions des études	Désavantages
MoMoR^o	Couche supplémentaire (L2,L3)	MoMoRo est lancé après perte de TX	ETX, SER,RSS	+ comparé à RPL (PRR, Overhead, Nb TX)	-detection mobility basé sur perte de paquets - maintenanc e de routes superflus
CoRPL	Corona areas	NP ne répond pas	Min CO-ID	+ comparé à RPL (taux de perte bas et efficacité énergétique)	-Compared to RPL -No energy consideration -CO-ID not good criteria for PP select
RPL Under Mobility	Ensemble de noeuds statiques	Algorithme Trickle désactivé	Couverture (Scénario d'autoroute)	+Comparé à RPL (Connectivité augmente)	- Grande charge de réseau overhead à cause des minuteries utilisées

	Environnement	Détection de la mobilité	Sélection du noeud parent	Conclusions des études	Désavantages
mRPL	Smart-hop (recensement continu des NP)	Niveau RSSI	Meilleur signal RSSI		-Variance RSSI -Pas de considérations énergétiques
ME-RPL	Algorithme d'apprentissages	Changement de NP	Préférence des noeuds statiques	+ comparé à RPL (PDR, PLR, stabilité des routes, etc)	-Pas de considérations énergétiques - Mouvement soudains non détectés

Tableau 1-b résumé de quelques études sur les réseaux écoenergetiques

	Principe	Fonction objectif	Conclusions des études	Désavantages
Energy Aware routing 3	Probabilité dans le choix des routes	coût de tx et rx d'un lien + énergie résiduelle	réduction de la consommation énergétique de 21,5 % + augmentation durée de vie de 44 %	réseau statique
Protocole basée sur négociation	énergie résiduelle + distance qui le sépare du nœud expéditeur	énergie disponible + qualité des liens	Meilleure performance même dans des milieux denses	réseau statique

CHAPITRE III

SOLUTION PROPOSÉE

1. Introduction

Nous répondons à la problématique posée grâce à une méthode en trois phases complémentaires. La première vise l'implémentation d'un mécanisme de gestion de la mobilité dédié aux WSN afin de réduire les pertes de paquets de données. La seconde vise l'amélioration de notre algorithme afin que la solution proposée soit écoénergétique. Cette phase est complémentaire à la première et à pour objectif de répondre aux exigences générales de notre problématique. Enfin, nous proposons une amélioration du mécanisme de gestion de la mobilité afin de minimiser le nombre de basculements.

2. Solution proposée

Afin d'atteindre notre premier sous-objectif, nous proposons un algorithme de basculement de point d'attache à caractère proactif et opération indépendante. À cet effet, nous proposons de créer quatre messages de contrôle qui permettent le détachement des aspects technologiques des protocoles de routage WSN disponibles et aident à implémenter notre solution. Les messages proposés sont :

- message d'information monodiffusion (en anglais Unicast Information Message, ou UIM) : permettant de véhiculer les informations des nœuds à un destinataire,
- messages de sollicitation diffusé (en anglais Broadcast Sollicitation Message, ou BSM): permettant de solliciter les nœuds voisins afin d'effectuer une tâche particulière,

- message de sollicitation monodiffusion (en anglais Unicast Sollicitation Message, ou USM) : permettant de solliciter un nœud en particulier,
- paquets de données (en anglais Data Paquets, ou DP) : servant à véhiculer des données en simple ou en multisaute et peut être de taille variable.

L'algorithme proposé comprend deux phases distinctes. La première, appelée « phase opérationnelle », vise l'envoi des données et la supervision de la qualité des liens. La deuxième, appelée « Phase de basculement », a pour objectif d'élire un nouveau nœud parent (NP)

2.1. Phase opérationnelle:

Dans cette phase, un nœud mobile (NM) envoie des paquets de données au nœud parent (NP) qui doit retourner un accusé de réception pour chacun tout en collectant la valeur du RSSI (Received Signal Strength Indication) pour la stocker en mémoire. Après l'envoi de sept paquets de données (DP) consécutifs par le nœud mobile, le nœud parent envoie un UIM comportant la moyenne des RSSI collectés (en anglais Average RSSI ou ARSSI). Une fois ce message reçu, le nœud mobile procède à une comparaison de la qualité du signal reçu par rapport à un seuil fixé au préalable. Si le ARSSI est supérieur au seuil, cela signifie que la liaison entre le NM et le NP est bonne. La phase opérationnelle est par conséquent maintenue. Sinon, le réseau bascule vers la phase de basculement afin de trouver un nouveau nœud parent.

2.2. Phase de basculement :

Ce mode s'enclenche avec l'envoi d'un BSM de la part du nœud mobile afin de solliciter les nœuds parents potentiels (NPP). Ces derniers répondent en envoyant des messages UIM comportant le niveau du signal RSSI ainsi que la quantité d'énergie résiduelle de chaque NPP. Le NM procède à une première sélection du nœud parent sur la base d'une fonction objectif, puis valide ce choix en envoyant un USM au

nœud parent sélectionné afin de vérifier la stabilité des données reçues. Le nœud parent renvoie alors un nouveau UIM afin que le nœud mobile puisse recalculer la fonction objectif. Les deux valeurs reçues sont comparées et le choix d'un nouveau nœud parent est effectué. Cette phase de validation est indispensable étant donné la variance de l'indicateur RSSI et les fluctuations qu'il subit. Si la valeur calculée en second lieu diffère de la première de plus de 15 %, le nœud mobile optera alors pour un nœud parent différent. Le diagramme temporel des deux phases citées est illustré dans la figure 6.

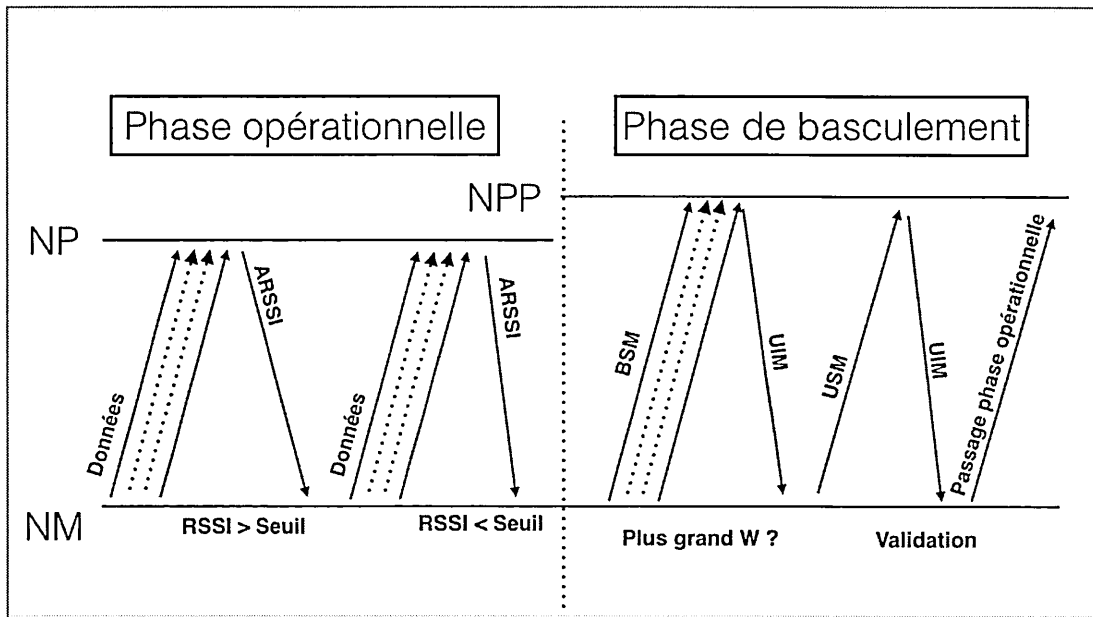


Figure 6 Diagramme temporelle de la solution proposée

3. Fonction objectif :

La fonction objectif proposée utilise deux métriques qui aident non seulement à détecter la mobilité, mais aussi contribuent à une prise de décision sensible à l'énergie. Ces deux métriques sont les suivantes :

3.1. Qualité du lien « RSSI » :

Cette mesure de puissance du signal peut être extraite et lue à partir des messages de la couche de liaison sans calcul additionnel de la part des nœuds. Cet avantage est non négligeable dans un contexte de mobilité où le temps de basculement doit être le plus court possible. Dans le cadre de la solution proposée, cette métrique intervient dans les deux phases (opérationnelle et basculement). Dans la première, la fiabilité du lien repose sur le niveau du RSSI tandis que dans la deuxième phase, la mesure du RSSI contribue au calcul de la fonction objectif en temps réel.

Une fois que le nœud mobile reçoit l'ensemble des messages UIM, il procède à une comparaison des différents RSSI reçus en prenant comme référence le meilleur d'entre eux, noté $maxrssi(t)$. Le nœud mobile calcule alors la fonction $f(t)$ afin d'attribuer un poids à chaque nœud et le facteur RSSI noté $f_n(t)$ sera calculé comme suit :

$$f_n(t) = \left(\frac{rssi_n(t)}{maxrssi(t)} \right) \quad (3.1)$$

où l'indice n représente l'identifiant des nœuds, $rssi_n(t)$ est la valeur de RSSI provenant de chaque NPP et $maxrssi(t)$ représente la meilleure valeur de RSSI reçue à un instant t .

3.2. Facteur énergie résiduelle :

Afin d'implémenter une logique de conscience énergétique dans le réseau, nous utilisons l'énergie résiduelle des nœuds stationnaires comme critère d'aide à la décision pour le basculement. Le but de cette logique est de maintenir le niveau énergétique des nœuds WSN relativement égal durant toute la durée de vie du réseau.

Une fois que le nœud mobile sollicite ses nœuds voisins pour élire un nouveau nœud parent, il envoie un message BSM afin de collecter les valeurs énergétiques des différents nœuds parents potentiels. Ces valeurs sont calculées au niveau de chaque NPP comme suit:

$$\frac{E}{V} = (I_{cpu}.T_{cpu}) + (I_{tx}.T_{tx}) + (I_{rx}.T_{rx}) + (I_{led}.T_{led}) \quad (3.2)$$

où E est l'énergie totale consommée par un nœud, V est la différence de potentiel sous laquelle le nœud opère. I_{cpu} , I_{tx} , I_{rx} et I_{led} représentent les intensités des courants utilisés par l'unité de calcul, la radio de l'émetteur « Tx », la radio du récepteur « Rx » et les diodes émettrices respectivement. Les variables T_{cpu} , T_{tx} , T_{rx} et T_{led} représentent les temps que l'unité de calcul, la radio de l'émetteur « Tx », la radio du récepteur « Rx » et les diodes émettrices respectivement ont passé dans les différents états.

Une fois l'énergie consommée calculée, les NPP transmettent cette valeur dans un message UIM à l'intention du nœud mobile qui calcule l'énergie résiduelle de chaque nœud avec la formule suivante :

$$Er(t) = \left(\frac{Et - Ec(t)}{Et} \right) \quad (3.3)$$

où $Er(t)$ est la valeur de l'énergie résiduelle attribuée à chaque nœud mobile et sera comprise dans l'intervalle $[0,1]$, « Et » représente l'énergie totale disponible dans les nœuds à l'instant $t = 0$. Cette énergie est égale pour tous les nœuds stationnaires.

3.3. Le poids (weight) :

Dans la phase de basculement, une fois les différentes informations en provenance des nœuds parents potentiels collectées, le nœud mobile utilise les deux métriques citées précédemment pour une prise de décision basée sur le poids (en anglais weight) de chaque nœud potentiel. Cette logique permet de prioriser ces nœuds afin de les exploiter intelligemment. Le poids W_n est calculé de la manière suivante :

$$W_n = \alpha \left(\frac{1}{f_n(t)} \right) + \beta (Er_n(t)) \quad (3.4)$$

L'utilisation des deux constantes α et β dans le calcul de la fonction objectif permet d'attribuer plus d'importance à une métrique par rapport à une autre. L'emploi de α et β permet également de créer différents profils de simulations lors du test de notre algorithme.

Afin d'analyser le comportement du réseau, avec et sans la fonction de conscience énergétique, nous avons créé deux profils distincts.

Profil I : c'est un scénario de simulation où la fonction objectif repose sur les deux métriques

$$W_n = \alpha \left(\frac{1}{f_n(t)} \right) + \beta (Er_n(t)) \quad (3.5)$$

Profil II : c'est un scénario de simulation où la fonction objectif repose uniquement sur la qualité du lien

$$W_n = \alpha \left(\frac{1}{f_n(t)} \right) \quad (3.6)$$

4. Optimisation de la solution proposée

Afin d'améliorer notre solution, nous proposons un algorithme intelligent que nous appelons algorithme de basculement (en anglais Handoff Algorithm ou HA). Cet algorithme nous renseigne sur la trajectoire du nœud mobile afin de minimiser le nombre de basculements tout en maintenant un niveau d'acheminement de paquets raisonnable. Le but de l'algorithme HA est de permettre au nœud mobile de s'adresser à un nouvel ensemble de parents potentiels à chaque nouvelle phase de basculement. Cette logique permet d'équilibrer le niveau d'énergie globale, de minimiser la charge du réseau et d'éviter les coupures engendrées par des phases de basculement trop fréquentes. Par conséquent, le réseau observera moins de pertes de paquets.

HA fonctionne sur la base de séquences de basculement. Le basculement à l'instant t est noté $\text{basculement}(t)$ et celui à l'instant $t+1$ est noté $\text{basculement}(t+1)$. Comme le montre la figure 7, le nœud mobile à la séquence de $\text{basculement}(t)$ envoie des messages BSM aux nœuds voisins 1 et 2 afin d'élire l'un d'entre eux comme parent d'attache. Nous supposons, dans cet exemple, que le nœud 1 est choisi et que la phase opérationnelle est enclenchée. Les deux nœuds parents potentiels seront gardés en mémoire au niveau du nœud mobile. Une fois que ce dernier sort de la zone de couverture du nœud parent 1, il enclenche une nouvelle séquence de $\text{basculement}(t+1)$ en envoyant des messages BSM aux NPP 2, 3 et 4. Étant donné que le nœud 2 est sélectionné dans les deux séquences consécutives t et $t+1$, il est alors dans notre intérêt de choisir le nœud 3 ou 4 comme nœud parent, car la probabilité de quitter la zone de couverture du nœud 2 est plus élevée. Par conséquent, le choix du nœud 2 engendrera rapidement un nouveau $\text{basculement}(t+2)$. Cette logique est basée sur le fait que la trajectoire du nœud mobile ne présente pas de scénario de « va-et-vient » qui n'est pas un cas fréquent dans les schémas de mobilité standards.

Les nœuds 3 et 4 seront priorisés par rapport au nœud 2 au moment du calcul de la fonction objectif. En effet, le nœud mobile procède à une comparaison systématique des NPP du basculement($t+1$) avec l'ensemble des NPP de la séquence de basculement(t). Si un nœud est présent dans les deux ensembles, le poids W_n qui lui sera attribué sera calculé avec des coefficients $\frac{\alpha}{2}$ et $\frac{\beta}{2}$. À l'opposé, les nœuds qui n'auront existé que dans l'ensemble de NPP de la séquence de basculement($t+1$) seront calculés avec des coefficients α et β . Il est à noter que les coefficients $\frac{\alpha}{2}$ et $\frac{\beta}{2}$ ont été définies de manière empirique dans notre travail.

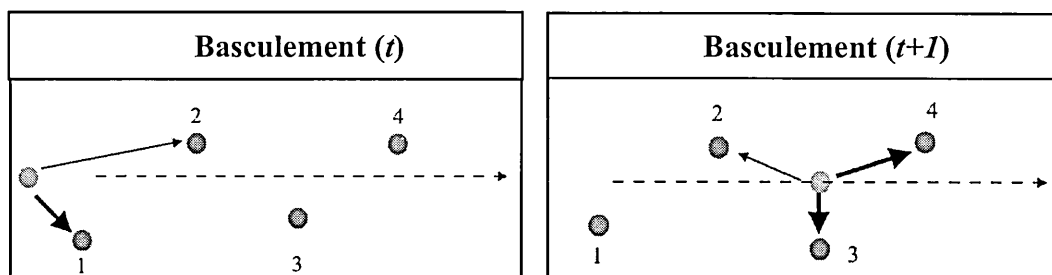


Figure 7 Mécanisme de Handoff Algorithm

À titre d'exemple, les poids W_n attribués aux noeuds présentés dans la figure 7 sont comme suit:

- séquence de basculement (t):

$$W_1 = \alpha \left(\frac{1}{f_1(t)} \right) + \beta (Er_1(t)) \quad (3.7.a)$$

$$W_2 = \alpha \left(\frac{1}{f_2(t)} \right) + \beta (Er_2(t)) \quad (3.7.b)$$

- séquence de basculement ($t+1$):

$$W_2 = \left(\frac{\alpha}{2}\right)\left(\frac{1}{f_2(t)}\right) + \left(\frac{\beta}{2}\right)(Er_2(t)) \quad (3.8.a)$$

$$W_3 = \alpha\left(\frac{1}{f_3(t)}\right) + \beta(Er_3(t)) \quad (3.8.b)$$

$$W_4 = \alpha\left(\frac{1}{f_4(t)}\right) + \beta(Er_4(t)) \quad (3.8.c)$$

Afin d'avoir une vue globale de la solution proposée, nous illustrons les différentes étapes dans le diagramme d'états de la figure 8 qui comporte les deux phases majeures (opérationnelle et basculement) ainsi que la procédure suivie par l'algorithme HA. Les détails de simulation de cette solution sont exposés dans le chapitre IV.

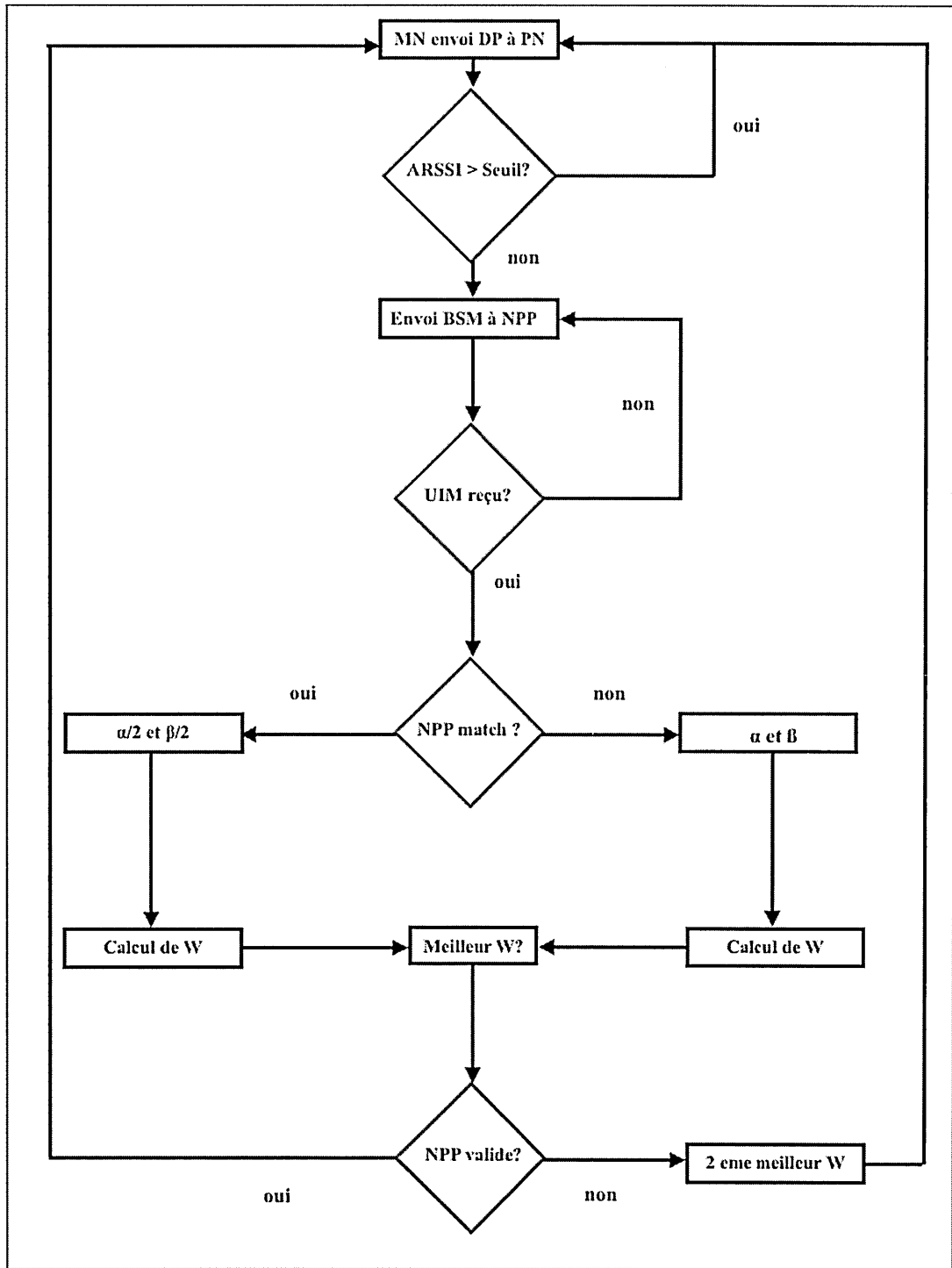


Figure 8 Diagramme d'état de la solution proposée

CHAPITRE IV

SIMULATIONS

Dans ce chapitre, nous présentons le contexte de travail et l'environnement de simulation que nous avons utilisé pour vérifier l'efficacité de notre modèle.

Le choix des composantes matérielles et logiciels est très important pour aboutir aux résultats escomptés. Pour cela, nous avons choisi le protocole de routage RPL qui concorde avec les hypothèses que nous nous sommes fixées. Nous énumérons les avantages qui ont amené à le choisir parmi d'autres protocoles. À ce titre, nous avons conduit une étude comparative des différents protocoles de routage dédiés aux réseaux de capteurs sans fil. Cette étude ainsi que les différents protocoles étudiés sont expliqués dans l'annexe I.

1. Choix du protocole RPL :

Il reste envisageable d'utiliser d'autres protocoles dans des travaux futurs afin de réaliser les mêmes simulations. Cependant, RPL bénéficie des avantages suivants qui le mettent en évidence par rapport aux autres protocoles :

- le support natif de 6LoWPAN qui permet d'implémenter un scénario d'IoT,
- le support de plusieurs types de flux pour les besoins de notre implémentation : multipoint à point, point à point, et point à multipoint,
- la proactivité de RPL dans le maintien de la topologie, propice à une implémentation de nœuds mobiles,
- l'entretien du réseau à faible coût énergétique notamment grâce à l'algorithme trickle ce qui fait de RPL un protocole très efficace énergétiquement, ainsi qu'une très faible empreinte en mémoire,
- la possibilité d'utiliser plusieurs fonctions objectives de manière native,

- la construction hiérarchique ou en arbre de la topologie d'un réseau,
- il est en libre distribution et peut être programmé en langage C.

Après analyse des différents protocoles, nous proposons le tableau 2 pour mettre en évidence les différentes caractéristiques des protocoles étudiés et justifier le choix du protocole RPL. Ce tableau est réalisé, en autres, en se basant sur les études suivantes (Vucinic *et al.*, 2013), (C. Perkins, Request for comments, juillet 2003), (Ee *et al.*, 2010), (Winter et Al, Request for Comments, mars 2012) et (Herberg et Clausen, 2011). Les protocoles étudiés sont détaillés dans l'annexe 1

Tableau 2 Comparatif entre les différents protocoles de routage WSN

Caractéristiques	AODV	LOAD	DYMO-low	HiLow	CITP	RPL
Nbre de sauts	utilise	optionnel	optionnel	utilise	-	optionnel
reparation locale des routes	utilise	utilise	n'utilise pas	n'utilise pas	utilise	utilise
utilisation énergétique	élevée	basse	basse	basse	-	moyenne
Utilisation de la mémoire	élevée	moyenne	moyenne	basse	basse	basse
support IPv6	oui	oui	oui	oui	oui	oui
type routage	réactif	réactif	réactif	réactif	proactif	proactif
support de la mobilité du noeud	bas	bas	bas	bas	moyen	moyen
temps de réponse	bas	bas	bas	bas	élevé	élevé

Tableau 2 Comparatif entre les différents protocoles de routage WSN

control de trafic	bas	bas	bas	bas	élevé	élevé
types de tables	table de routage	table de routage	table de routage	table de routage	prochain saut	table de routage
types de trafic	P2P, P2MP	P2P, P2MP	P2P, P2MP	P2P, P2MP	MP2P, P2MP	P2P, P2MP, MP2P

Les caractéristiques étudiées ont été choisi de sorte à répondre aux besoins de notre solution. Nous notons par exemple la pro-activité qui demeure une condition importante dans un environnement doté de mobilité.

2. Paramètres de simulation :

2.1. Politique d'économie d'énergie

Dans le cadre de la mobilité dans les réseaux de capteurs sans fil, les différents nœuds doivent réagir rapidement aux sollicitations des nœuds voisins afin de diminuer le temps de reconstruction de la topologie et limiter la perte de paquets. En effet, les nœuds mobiles suivent des trajectoires parfois aléatoires et envoient des paquets de données de manière constante. Ces envois permanents poussent les capteurs à maintenir leurs radios constamment allumées afin d'acheminer ces paquets en tout temps. Cette approche maintient la réactivité du réseau au détriment de la consommation énergétique. Dans notre étude, nous implémentons une politique de cycle de service (en anglais Duty Cycle) afin d'analyser le comportement énergétique du réseau sous des contraintes de mobilité.

ContikiMAC (Dunkels, 2011) est un protocole de cycle de service de la radio appelé RDC (en anglais Radio Duty Cycle) qui utilise des réveils périodiques afin d'écouter les transmissions qui proviennent des nœuds voisins. Si la transmission d'un paquet est détectée pendant la période d'éveil, le nœud destinataire garde sa radio allumée pour recevoir les paquets puis envoie un accusé de réception au niveau de la couche de liaison. L'expéditeur de son côté envoie les paquets voulus de manière répétitive jusqu'à ce qu'il reçoive un accusé de réception de la part du destinataire. Dans le cas d'une diffusion, aucun accusé de réception n'est reçu. Cependant, l'envoi persiste pendant toute la durée d'éveil du nœud. Le principe de ContikiMAC est illustré dans la figure 10.

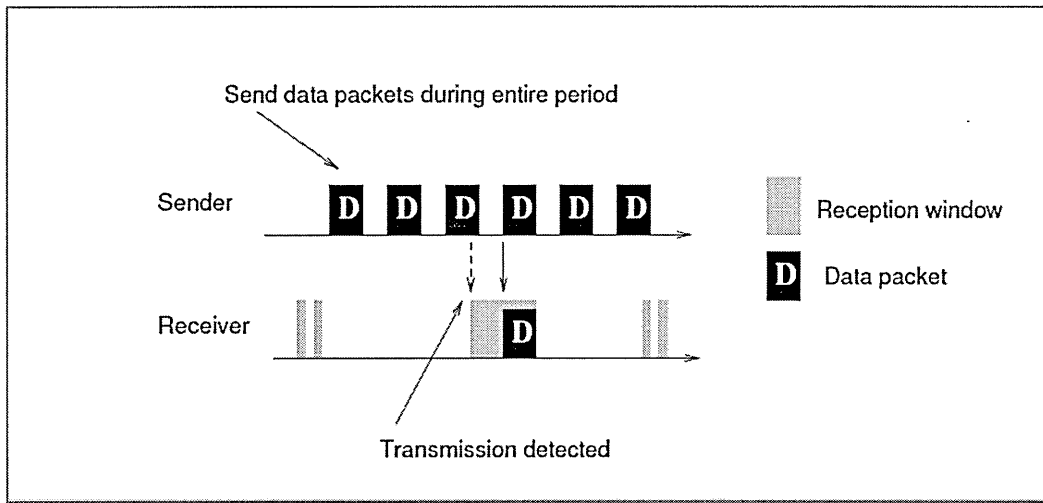


Figure 10 Illustration du fonctionnement de ContikiMAC

ContikiMAC utilise un mécanisme d'éveil qui repose sur une synchronisation précise entre les transmissions. Les cycles d'éveil sont réalisés à l'aide d'un mécanisme d'évaluation de canal libre (en anglais Clear Channel Assessment, ou CCA) qui donne des indications sur l'activité de la radio dans le canal grâce au RSSI mesuré. Si le RSSI est en dessous d'un seuil donné, le CCA retourne une valeur positive voulant dire que le canal est vide, sinon, il retournera une valeur négative afin de montrer que le canal est occupé. La figure 11 montre les différents minuteurs utilisés.

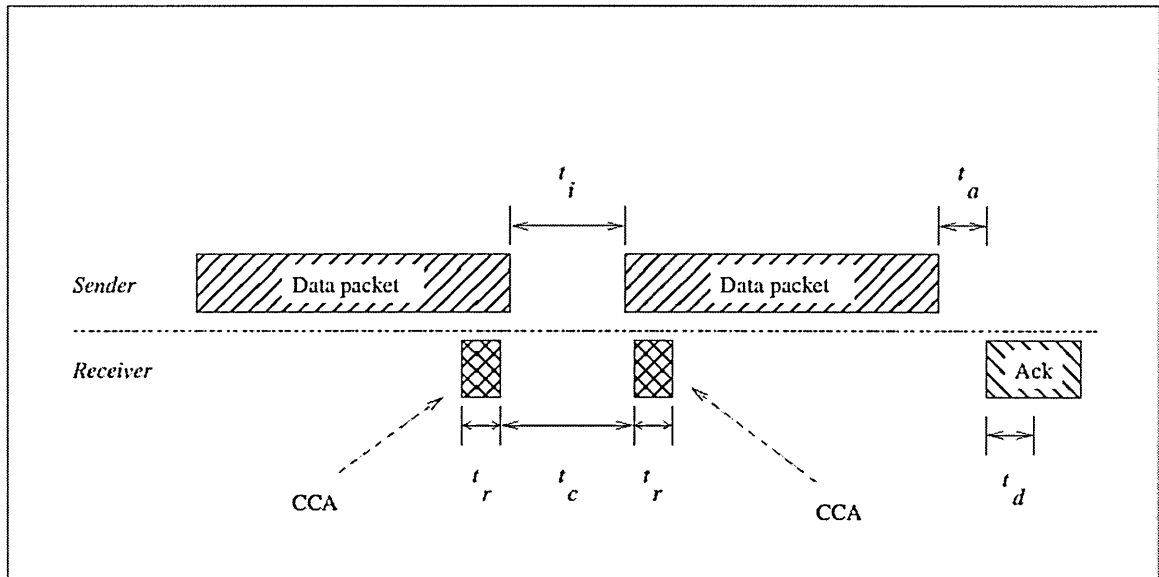


Figure 11 Illustration des différents minuteurs utilisés dans ContikiMAC

t_i : l'intervalle entre chaque transmission de paquet

t_r : le temps nécessaire pour avoir une mesure stable du RSSI, requise pour une indication stable du CCA.

t_c : l'intervalle entre chaque CCA.

t_a : le temps nécessaire entre la réception d'un paquet et l'envoi de son accusé de réception

t_d : le temps nécessaire afin de détecter un accusé de réception provenant du récepteur

2.2. Messages de controles:

Nous avons décidé d'utiliser les messages de contrôles natifs de RPL pour assurer une compatibilité avec le protocole sans empiéter sur les messages de contrôles responsables du maintien de la topologie. Afin d'utiliser les DIO et DIS qu'offre RPL et d'éviter de les confondre avec les messages de contrôles que nous créons, nous leur avons apporté quelques modifications. La solution proposée requiert l'envoi de

messages DIS et DIO en mode unicast. Les messages DIO du mode opérationnel et du mode basculement doivent être clairement distingués. Dans le premier mode, les DIO envoyés par le nœud parent contiennent la valeur du ARSSI seulement, tandis que dans le mode de basculement, les messages contiennent le ARSSI et la valeur énergétique des nœuds. Afin d'appliquer les modifications nécessaires, nous avons utilisé les champs d'options (en anglais Options Fields) de chaque type de message comme montré dans les figures 12 et 13. Nous avons rajouté un bit de distinction dans le message DIS qu'on notera (U) qui signifie monodiffusion (en anglais Unicast). La valeur $U = 0$ sera attribuée aux messages DIS natifs. Si $U=1$, le DIS sera en monodiffusion utilisé dans le mode de basculement.

Pareillement pour le message DIO, nous avons rajouté 2 bits dans le champ d'options. Nous avons rajouté 8 bits pour le ARSSI ainsi que 8 bits pour le niveau d'énergie.

$U= 00$ signifie que le DIO est natif. $U=01$ ou 10 signifie que le DIO est en monodiffusion dans le mode opérationnel comportant un ARSSI. Finalement, $U= 11$, signifie que le DIO est monodiffusion et est destiné à véhiculer le RSSI de chaque NPP ainsi que leur niveau d'énergie (mode de basculement).

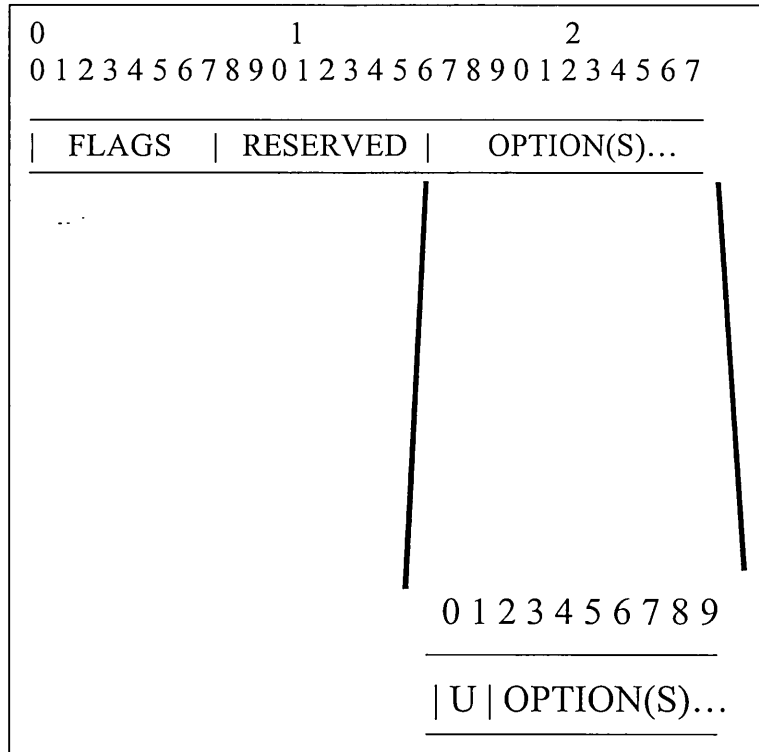


Figure 12 Illustration des modifications apportées aux messages DIS

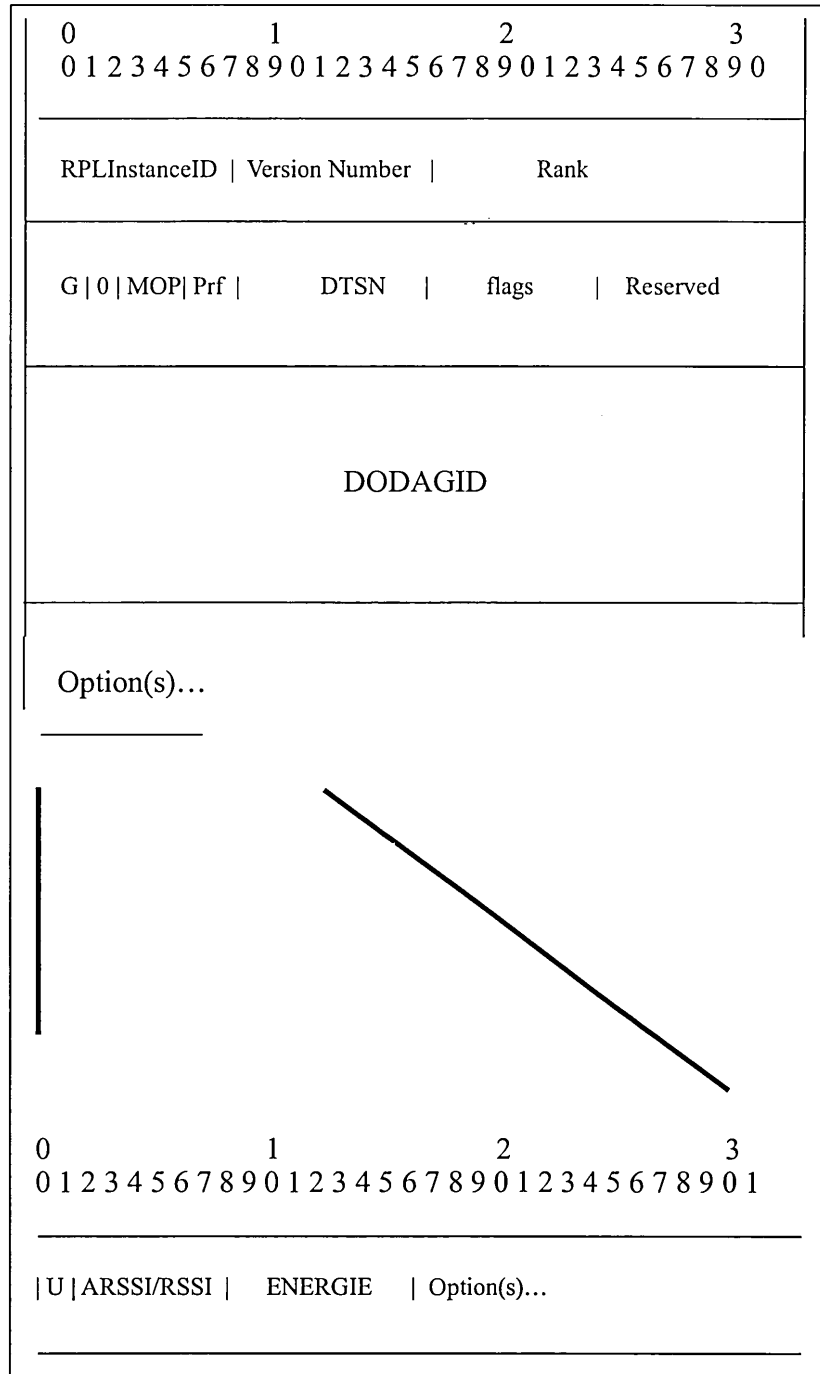


Figure 13 Illustration des modifications apportées aux messages DIO

2.3. Choix du SEE et du simulateur COOJA

Après avoir comparé les différents systèmes d'exploitation énoncés précédemment, nous avons opté pour ContikiOS grâce à son efficacité énergétique et son modèle de programmation mixte (en anglais protothread) qui permet de programmer des événements internes et externes relatifs à notre scénario tout en limitant la charge du réseau sur la mémoire de manière considérable. De plus, ContikiOS comporte une implémentation stable de RPL/6LoWPAN et largement utilisée dans le domaine de la recherche scientifique.

Nous avons également opté pour le simulateur Cooja (Osterlind, 2006) qui est un simulateur flexible basé sur Java. Il est utilisé pour simuler les réseaux de capteurs qui supportent le système d'exploitation ContikiOS. Son extensibilité permet de rajouter plusieurs fonctionnalités, comme les entrées/sorties des nœuds, le type de nœuds et leur rôle, etc.

Cooja offre la possibilité de simuler des nœuds non natifs pour Contiki et exécute un code natif en faisant des appels JNI (en anglais Java Native Interface calls) à partir de l'environnement Java. Cette particularité permet de simuler et de déployer le même code sans pour autant y apporter des modifications, et ainsi, gagner du temps entre les simulations et le déploiement. Cooja offre également un système de sauvegarde qui permet de récupérer des versions antérieures des scénarios de simulations ainsi qu'un debugger. Toutes les interactions entre l'utilisateur et l'environnement de simulation sont faites grâce à des modules d'extensions (en anglais Plug-ins). Ces modules permettent d'activer ou d'arrêter la simulation, de visualiser l'activité radio pendant une simulation, ou encore de simuler la mobilité dans un nœud comme c'est le cas dans notre travail.

Cette flexibilité qu'offre Cooja nous est d'une grande utilité afin de simuler notre solution en respectant les conditions fixées.

3. Mise en oeuvre des simulations :

Pour les besoins de la simulation, nous proposons un environnement constitué d'un nœud mobile, un nœud principal (en anglais Root) et plusieurs nœuds statiques. Cette installation pourrait convenir aux différents scénarios de mobilités cités précédemment en prenant soin de calibrer les différents paramètres pour chaque projet. La topologie utilisée est illustrée dans la figure 14 qui montre une disposition triangulaire des nœuds sur 5 rangs afin de mieux contrôler l'aspect hiérarchique avec lequel notre protocole de routage fonctionne. Cette topologie aide à mieux analyser le comportement du réseau en disposant judicieusement les zones de chevauchement des différents nœuds stationnaires. Ces derniers sont placés à une distance de 50 mètres les uns des autres avec une zone de couverture circulaire de 100 mètres. Le nœud au sommet de la pyramide représente le nœud principal. Les nœuds noirs sont des nœuds stationnaires et en rouge un nœud mobile qui suit la trajectoire mentionnée. Les segments horizontaux de la trajectoire aident à analyser le comportement du NM dans un même rang tandis que les segments verticaux aident à étudier la mobilité au fur et à mesure que le NM change de rang hiérarchique.

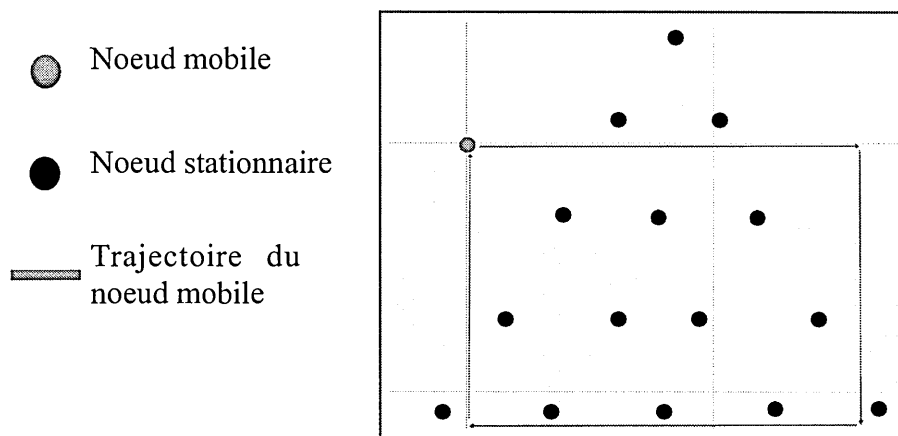


Figure 14 Illustration de la topologie de réseau simulée

Trois profils différents ont été développés au niveau des nœuds pour les besoins de la solution. Le premier est le profil du nœud mobile qui a pour fonction d'envoyer des paquets de données à des fréquences régulières. Le deuxième est un profil qu'on retrouve majoritairement dans le réseau étudié qui est le profil des nœuds stationnaires. Ces nœuds ont pour rôle de transmettre les paquets de données reçus par le nœud mobile à leurs nœuds parents. Le troisième et dernier profil est celui du nœud principal. Ce dernier reçoit les paquets qui lui sont destinés et accuse réception en affichant un message sur l'écran.

Afin d'étudier la mobilité dans un contexte énergétique, nous implémentons un cycle de service de type ContikiMAC aux nœuds stationnaires. Le nœud mobile et le nœud principal sont en mode NullRDC ce qui veut dire que leur radios restent constamment allumées.

Le système d'exploitation utilisé est Contiki version 2.7 (<http://www.contiki-os.org/download.html>) et la plateforme des nœuds simulés est le Tmote Sky (Moteiv Corporation, 2006) équipé d'une radio Chipcon CC2420 (Moteiv Corporation, 2006) qui opère à une fréquence de 2.4 GHz avec un débit de données de 250 kbit/s. En considérant ce débit et la taille des paquets, la radio peut transmettre jusqu'à 246 paquets/s (1 paquet chaque 4 ms). Pour nos simulations, nous optons pour des délais minimums de 33 ms compte tenu des délais apportés par la procédure de fragmentation, de défragmentation de la couche d'adaptation ainsi que le délai relatif au modèle de propagation choisi.

Pour les besoins de l'étude, nous faisons varier les paramètres suivants et nous mesurons leur impact sur les performances du réseau :

- la vitesse du nœud mobile : nous simulons l'intervalle de vitesse d'un être humain en marche pour prélever les performances et l'efficacité de la solution proposée. Cet intervalle de vitesse sera appliqué également sur d'autres implémentations à des fins de comparaisons. L'intervalle en question est : [0.5-5] m/s

- la fréquence d'éveil : c'est la fréquence à laquelle les nœuds sortent de leur état de dormance pour sonder le réseau. Ce paramètre est d'une importance capitale dans cette étude, car il a un impact direct sur l'énergie résiduelle des nœuds. Plus la fréquence d'éveil est basse, moins la consommation énergétique est importante. Les valeurs de ce paramètre de simulation sont entre 8 Hz et 128 Hz ce qui correspond à un intervalle compris entre 7,81 ms et 125 ms. Cet intervalle permet de simuler des fréquences d'éveils plus ou moins importantes par rapport à la valeur minimale de la fréquence d'envoi de données de 33 ms que nous avons choisi.
- la fréquence d'envoi des données : c'est la fréquence à laquelle le nœud mobile envoie des paquets à destination du nœud principal. Ce paramètre est important dans la mesure où il permet d'étudier la capacité du réseau à acheminer un certain nombre de paquets dans un intervalle de temps. Également, il permet d'alléger la charge du réseau durant les simulations afin que les résultats ne soient pas noyés dans des messages de contrôles excessifs. Les valeurs choisies pour ce paramètre résident entre 33 ms et 5 s. Ces valeurs tiennent compte des recommandations du constructeur de la radio Chipcon CC2420 (Moteiv Corporation, 2006) ainsi que les règles d'usages de certains domaines d'application cités précédemment comme le domaine médical ou industriel.
- l'algorithme Trickle : les principaux paramètres qui impactent les performances du protocole standard RPL résident dans l'algorithme trickle. En effet, ce dernier comporte deux variables appelées I_{min} et I_{max} qui définissent les périodes d'envoi des messages d'information DIO. Ces derniers servent à maintenir la topologie du réseau à jour. Plus I_{min} et I_{max} sont petits, plus le réseau sera réactif. Dans notre étude nous comparons la solution proposée à une implémentation standard de RPL en prenant les deux scénarios extrêmes de réactivité. Le tableau 3 (voir chapitre

V-2) comporte les valeurs de ses scénarios ainsi que la manière avec laquelle I_{min} et I_{max} ont été calculées.

Les simulations ont été conduites sur des intervalles de temps allant de 90 secondes à 10 minutes afin de vérifier l'exactitude des valeurs prélevées sous différentes périodes. Nous n'avons cependant pas pu aller au-delà de 10 minutes à cause des limitations du logiciel utilisé. Chaque simulation a été reconduite une dizaine de fois afin d'être sûre de la stabilité des données extraites. À ce propos, les résultats auxquels nous nous sommes intéressés concernent les paramètres suivants :

- le taux d'acheminement : c'est le rapport entre la quantité de données envoyée par le nœud mobile et celle reçue au niveau du destinataire (le nœud principal). Ce paramètre est un bon indicateur de l'efficacité de la solution proposée étant donné le scénario de collecte de données étudié. Le taux d'acheminement de paquets (TA) est extrait grâce à un programme que nous avons développé sous Python.
- la charge du réseau (en anglais network overhead) : nous identifions tous les paquets échangés dans le réseau comme étant des paquets de charge du réseau à l'exception des paquets de données et leurs accusés de réception. Autrement dit, tous les messages de contrôle sont considérés comme étant des messages de charges du réseau. Le contrôle de ce paramètre permet de diminuer la consommation énergétique globale et d'éviter les retransmissions. Ces données sont également extraites grâce à un programme Python que nous avons développé.
- consommation énergétique : ce paramètre indique la quantité d'énergie résiduelle des nœuds. C'est le ratio entre l'énergie de départ et l'énergie restante après les simulations. Les batteries des nœuds du réseau disposent de la même énergie de départ et ne bénéficient pas de logique de récupération d'énergie (en anglais energy harvesting). Ce paramètre est mesuré en utilisant une fonction d'estimation d'énergie appelée « energest » intégrée au simulateur Cooja. Cette fonction permet

de superviser la consommation énergétique de chaque entité du capteur, à savoir: CPU (unité centrale de calcul), LPM (modes faible consommation), radio en mode réception, radio en mode transmission, etc.

Toutes les données extraites des simulations sont présentées et analysées dans le chapitre V.

CHAPITRE V

RÉSULTATS

1. Introduction

Cette section présente d'abord les résultats issus de la comparaison de notre solution avec une implémentation standard de RPL. Ensuite, nous étudions l'impact de la variation de la vitesse du nœud mobile sur le comportement du réseau. Puis, nous procédons à une analyse énergétique au niveau des nœuds afin de faire le bilan énergétique du réseau. Finalement, nous comparons la solution proposée à d'autres travaux similaires.

2. Comparaison avec RPL standard

Afin de positionner la solution proposée dans ce travail, nous procédons à une comparaison avec une implémentation standard de RPL. Selon les spécifications de ce dernier, la seule manière qui permet d'augmenter la réactivité du réseau face aux changements de topologies provoqués par un nœud mobile est de faire varier les paramètres de l'algorithme Trickle. À cet effet, nous considérons deux scénarios extrêmes où nous faisons varier les intervalles I_{min} et I_{max} d'envoi de messages DIO tout en maintenant constante la vitesse du nœud mobile durant la simulation. Ces intervalles sont prédéfinis comme suit :

Scénario le moins réactif (12-8)	Scénario le plus réactif (8-1)
$I_{min} = 2^{12} = 4096\text{ms}$	$I_{min} = 2^8 = 256\text{ms}$
$I_{max} = I_{min} \times 2^8 = 1048,576 \text{ s}$	$I_{max} = I_{min} \times 2^1 = 512\text{ms}$

Tableau 3 Illustration des scénarios de réactivité du protocole RPL standard

Tout au long de cette section, nous noterons le scénario le plus réactif : (8-1) et le scénario le moins réactif : (12-8) à des fins de simplification. Les simulations sont conduites avec le scénario (8-1) puis avec le scénario (12-8) pour la solution proposée et pour RPL. Nous simulons à chaque fois sur une durée de 6 minutes. Ce temps est nécessaire avant que la simulation ne se stabilisent et nous permette d'obtenir des résultats fiables. Cette durée permet également au nœud mobile d'effectuer plusieurs tours afin de forcer le calcul de plusieurs routes. Le nœud mobile se déplace à une vitesse constante de 2 m/s et envoie un paquet chaque 33 ms. La fréquence d'éveil des nœuds est fixée à 64 Hz (15,6 ms) ce qui permet à un nœud de sortir deux fois de son cycle de dormance dans un intervalle d'envoi de (33 ms). Ceci est fait de sorte à limiter les pertes de paquets et pour se concentrer sur l'impact des scénarios étudiés. Nous obtenons alors les résultats de la figure 15 :

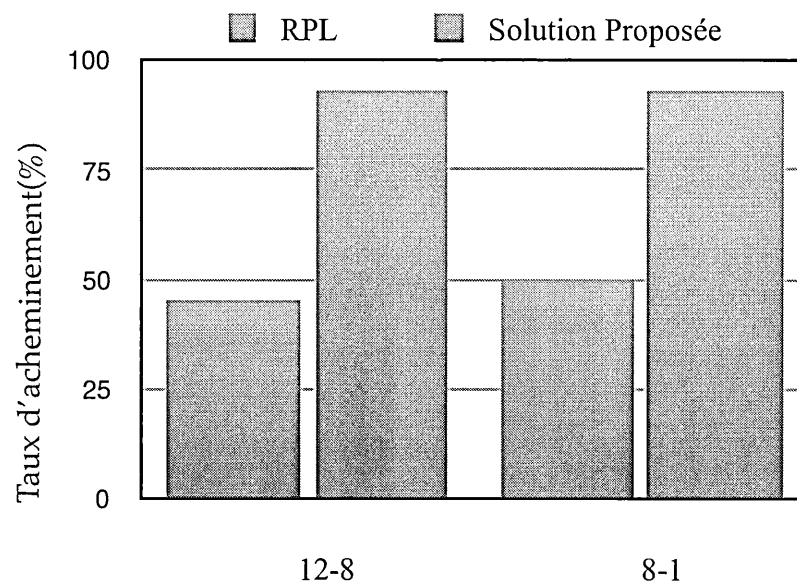


Figure 15.a Illustration du taux d'acheminement des paquets en fonction des intervalles DIO

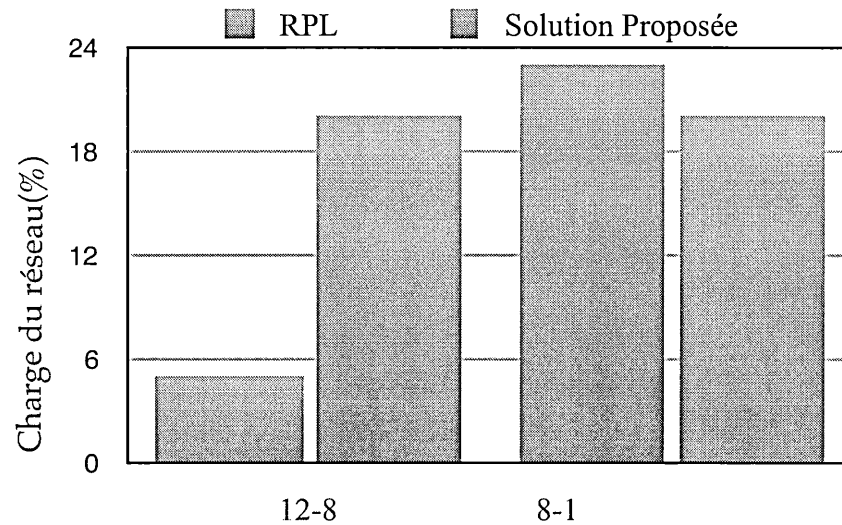


Figure 15.b Illustration de la charge du réseau en fonction des intervalles DIO

La figure 15.a illustre le taux d'acheminement des paquets dans les deux scénarios de simulation qu'on note (12-8) et (8-1) (voir tableau 3). Nous constatons que le taux d'acheminement des paquets de la solution proposée est nettement supérieur à celui de RPL, et ce, même dans le scénario le plus réactif de celui-ci. Il est également à mentionner que le changement des paramètres de l'algorithme trickle n'affecte pas le comportement de la solution proposée, car celle-ci a été développée indépendamment des minuteurs du protocole de routage. Les mécanismes de changements d'état et les seuils qui activent l'envoi des messages de contrôles de la solution proposée sont rattachés à la solution elle-même.

La figure 15.b montre le taux de charge du réseau pour les deux solutions, nous constatons que la charge de réseau de la solution proposée reste inchangée en fonction des paramètres de l'algorithme trickle, ce qui n'est pas le cas de RPL. En effet, la charge du réseau dans RPL augmente considérablement en rapprochant les

intervalles de l'algorithme trickle à cause du nombre de messages de contrôles échangés.

3. Effet de la mobilité

Afin d'étudier l'impact de la vitesse du nœud mobile sur le réseau, nous varions sa vitesse en partant d'un état statique jusqu'à une vitesse de 5 m/s. L'intervalle d'envoi des données est de 33 ms et la fréquence d'éveil est de 64 Hz. À chaque vitesse, nous effectuons dix simulations afin de nous assurer de la stabilité des données. Nous nous intéressons au taux d'acheminement, de la charge du réseau ainsi qu'à l'énergie globale consommée dans le réseau. Cette dernière variable a été mesurée en utilisant la fonction « *energest* » de Cooja. La figure 16 montre les résultats obtenus suite à la simulation du profil I. À titre de rappel, ce profil utilise la fonction objectif suivante :

$$W_n = \alpha \left(\frac{1}{f_n(t)} \right) + \beta (Er_n(t))$$

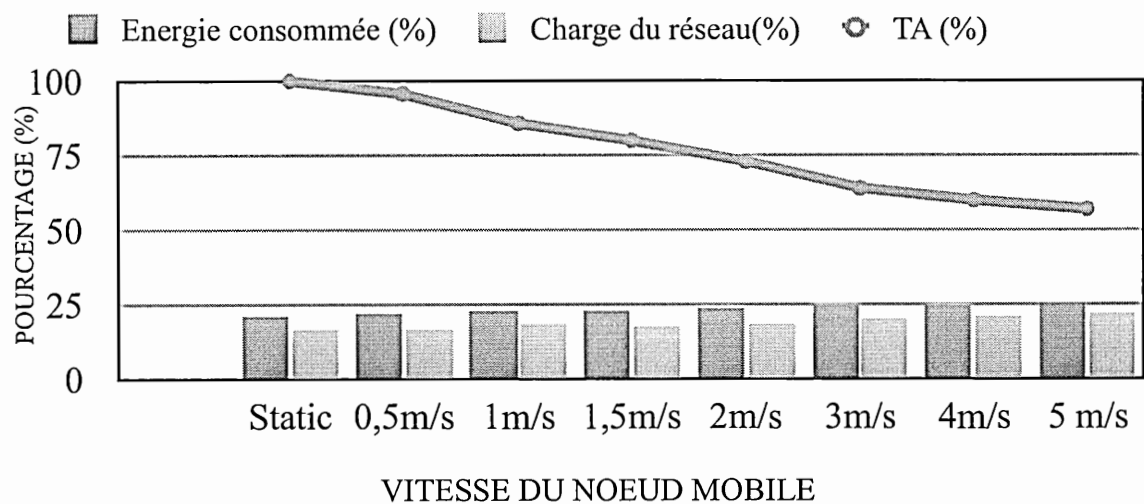


Figure 16 Illustration des résultats obtenus avec le profil I

3.1. Taux d'acheminement des paquets (TA)

Le taux d'acheminement diminue considérablement au fur et à mesure que la vitesse du nœud mobile augmente. Ce phénomène est causé par le nombre important de basculements dans les situations de haute mobilité. De plus, le taux de perte des paquets augmente, car les procédures de basculements ne vont pas jusqu'au bout de leur tâche. En effet, le changement rapide d'emplacement du nœud mobile ne garantit pas la sélection d'un nœud parent à chaque fois.

3.2. Consommation énergétique

Nous mesurons la consommation d'énergie globale du réseau en augmentant la vitesse du nœud mobile. La figure 16 montre en vert que la solution proposée consomme plus d'énergie avec l'augmentation de la vitesse. Cette augmentation est due au taux de pertes de données de plus en plus élevé. En effet, les paquets perdus nécessitent en règle générale plus d'énergie que les paquets livrés avec succès à cause du protocole CSMA. Ce dernier enclenche des retransmissions des paquets perdus et par conséquent l'énergie consommée dans le réseau augmente. Également, l'appel fréquent à l'algorithme de basculement augmente le nombre de messages de contrôles dans les réseaux ce qui pousse à une consommation plus importante. Cependant, cette augmentation reste très faible dans notre scénario (moins de 5 %), nous pouvons donc conclure que la vitesse du nœud mobile n'affecte quasiment pas la consommation énergétique.

Les mêmes conditions de simulation ont été conduites en utilisant le profil II qui

opère avec la fonction objectif suivante :
$$W_n = \alpha \left(\frac{1}{f_n(t)} \right)$$

Les résultats obtenus sont illustrés dans la figure 17. Nous constatons que ce scénario présente les mêmes résultats que le profil I et la modification de la fonction objectif n'apporte pas de changements au fonctionnement global du réseau.

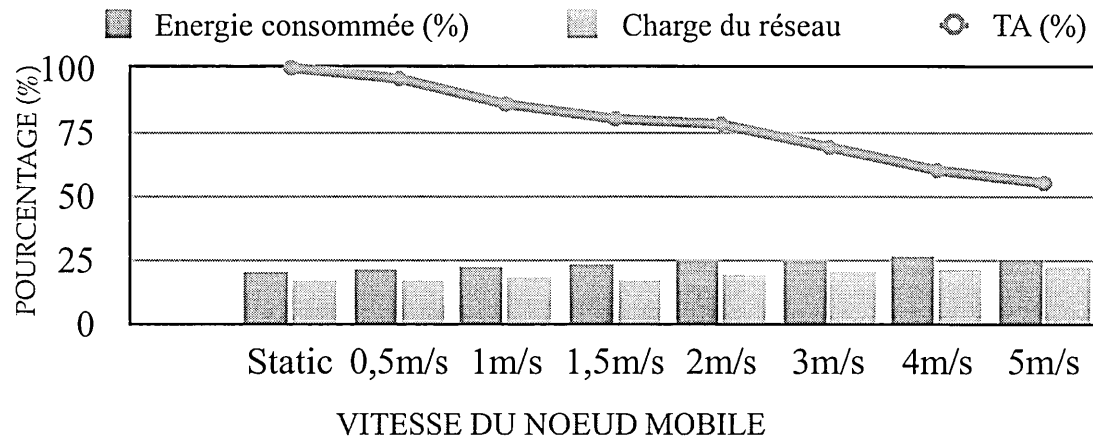


Figure 17 Illustration des résultats obtenus avec le profil II

Afin de mieux analyser l'impact de la solution proposée sur le plan énergétique, nous avons conduit des simulations avec les profils I et II pendant 4 et 8 minutes en maintenant la vitesse du nœud mobile à 2 m/s, l'intervalle d'envoi de données à 33 ms et la fréquence d'éveil à 64 Hz. Nous avons considéré que l'énergie globale disponible dans les nœuds offre une durée de vie de 15 minutes seulement. Cette supposition nous permet d'avoir des résultats significatifs sur de plus courtes durées de simulations. Nous prélevons le niveau de consommation énergétique de chaque nœud après les simulations. Nous avons réalisé 10 itérations dans chaque profil afin de nous assurer de l'exactitude des données extraites. Les résultats sont illustrés dans la figure 18.

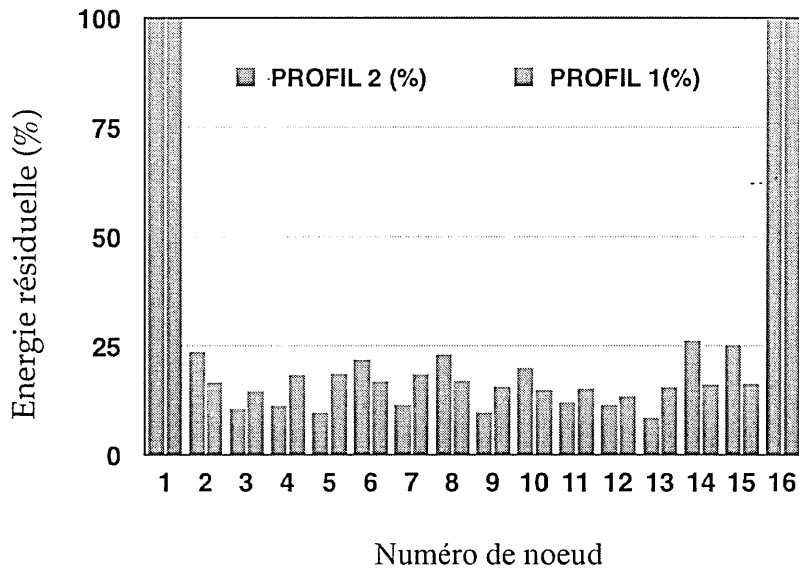


Figure 18.a Energie résiduelle par noeud à 4 minutes de simulation

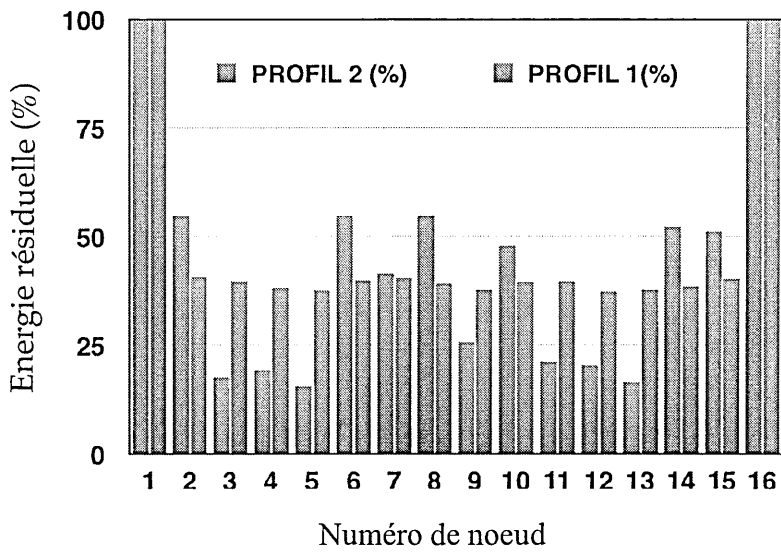


Figure 18.b Energie résiduelle par noeud à 8 minutes de simulation

Comme expliqué précédemment, le nœud mobile et le nœud principal respectivement représentés par les chiffres 1 et 16, bénéficient d'une alimentation infinie à des fins de simulation. Dans la figure 18.a, nous illustrons les résultats à l'issue de 4 minutes de simulation. Nous constatons que la consommation énergétique des nœuds dans le profil I (représentée en vert) reste relativement égale pour tous les nœuds. Par contre, elle présente une grande disparité dans le profil II. Cette même figure montre que les mêmes nœuds du réseau sont sélectionnés à chaque passage du nœud mobile, ce qui pousse à la surexploitation de certains nœuds par rapport à d'autres. Ce phénomène peut causer des disparités énergétiques importantes entre les nœuds. Ces disparités sont accentuées pour des temps de simulations prolongés. La figure 18.b illustre les résultats en bleu à l'issue de 8 minutes de simulation où la sollicitation constante de certains nœuds devient flagrante. À l'opposé, l'utilisation du profil I (en vert) aide à réguler la consommation énergétique de chaque nœud d'où l'importance de la fonction objectif proposée.

4. Optimisation du nombre de basculements

À des fins d'analyse du comportement du réseau et de vérification de l'algorithme proposé, nous avons procédé à des simulations comparatives avec et sans HA. L'efficacité de cet algorithme s'exprime en nombre de séquences de basculements réalisées dans une simulation. À cet effet, nous avons conduit des simulations sur des durées de 1,2,3 et 4 minutes avec un nœud mobile se déplaçant à une vitesse de 2 m/s. Chaque simulation est conduite une dizaine de fois pour valider les résultats. Le nombre de basculements a été calculé grâce à un programme que nous avons développé sous Python. Ce programme calcule le nombre de messages DAO envoyés par le nœud mobile. Les messages DAO sont envoyés à chaque fois qu'une connexion est établie avec un nouveau nœud parent pour informer le réseau des coordonnées du nœud mobile. Dans notre cas, l'envoi d'un message DAO veut dire

qu'une phase de basculement vient de s'achever. Nous illustrons les résultats dans la figure 19 et nous concluons ce qui suit :

L'utilisation de l'algorithme de basculement HA représenté en vert permet de réduire effectivement le nombre de basculements dans le réseau sans dégradation du taux d'acheminement des paquets. Son utilisation est parfaitement justifiée, en particulier pour des simulations étendues dans le temps. Il permet une baisse du nombre de basculements d'approximativement 25 % ce qui n'est pas négligeable considérant la consommation énergétique et le nombre de messages de contrôles qu'une seule séquence de basculement engendre.

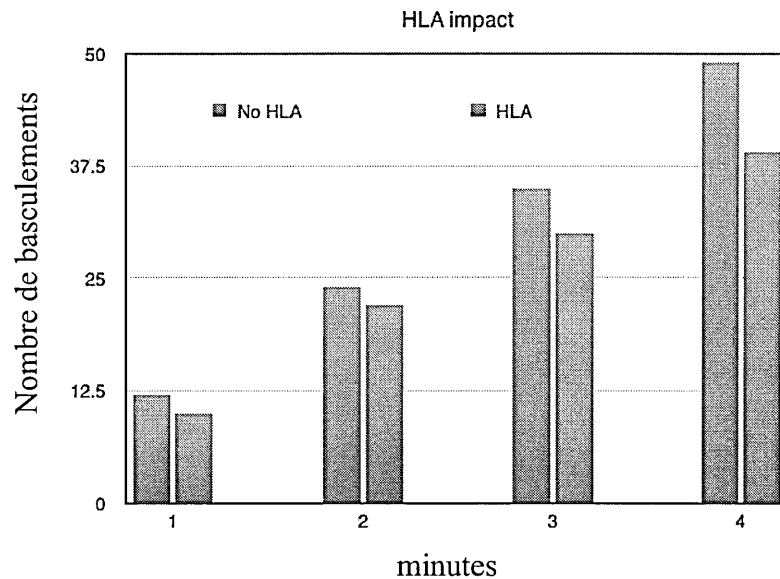


Figure 19 Comparaison du nombre de basculements par rapport au temps dans deux solutions (HA et No HA)

5. Comparaison de la solution proposée avec mRPL

Afin de comparer la solution proposée à un protocole de routage qui supporte la mobilité des nœuds dans les WSN, nous avons choisi mRPL de par les similitudes présentées entre les deux études. Comme expliqué dans la revue de littérature, mRPL supporte la mobilité sans prendre en considération l'énergie des nœuds. À cet effet, aucun nœud simulé ne dispose de mécanisme de gestion de cycles de dormance des radios. Les résultats des tests énoncés dans cette étude montrent que l'utilisation d'une politique de cycle de service au niveau des nœuds dégrade considérablement la performance et perturbe le fonctionnement du réseau. Nous avons comparé les résultats obtenus avec la solution proposée sous les mêmes conditions, et ce, en faisant varier la fréquence d'envoi des données du nœud mobile en fonction de l'intervalle d'éveil des nœuds afin de collecter le taux d'acheminement (TA). Les résultats obtenus sont illustrés dans la figure 20.

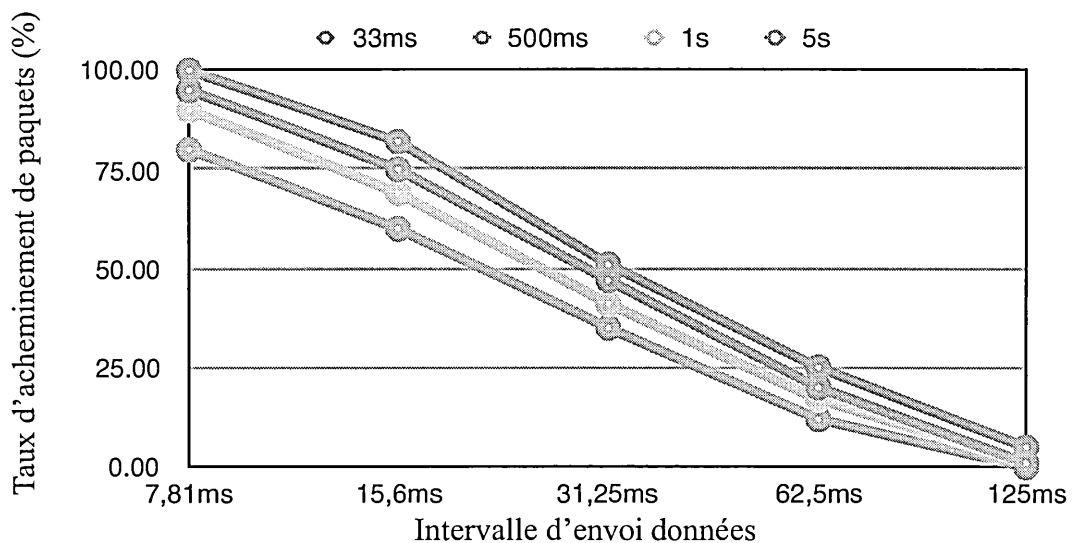


Figure 20 Taux d'acheminement vs cycle de service pour différentes fréquences d'envoi de données

En prenant en considération un intervalle d'envoi de 33 ms, nous constatons que le taux d'acheminement diminue au fur et à mesure que nous baissions la fréquence d'éveil des nœuds. En effet, plus la radio des nœuds est en dormance, plus le temps de basculement sera long. Une fois que le NM envoie des messages de sollicitations pour trouver un nouveau parent, il se passe un certain temps avant qu'un nœud voisin ne sorte de son état de dormance pour accuser réception de ses messages de sollicitations. Entre-temps, le nœud mobile continue à envoyer des paquets de données qui seront alors perdus. Ce comportement est similaire quelle que soit la fréquence d'envoi des données. La figure 20 explique ce fonctionnement.

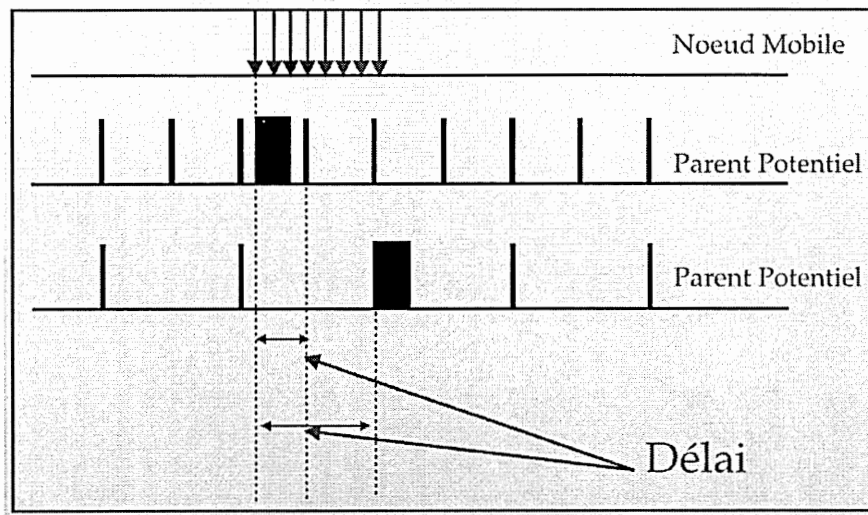


Figure 21 Illustration de l'impact de l'intervalle du cycle de service sur le délai de découverte d'un nouveau parent

Afin de vérifier notre explication, nous avons mesuré la charge du réseau et le niveau d'énergie globale pour les différentes fréquences d'éveils. La figure 21 montre que la charge du réseau augmente au fur et à mesure que la fréquence d'éveil diminue. Ce phénomène est dû à l'envoi de messages de sollicitations en mode diffusion (broadcast) de la part du NM afin de trouver un nouveau parent. L'énergie a tendance à diminuer, car les nœuds stationnaires deviennent de moins en moins actifs.

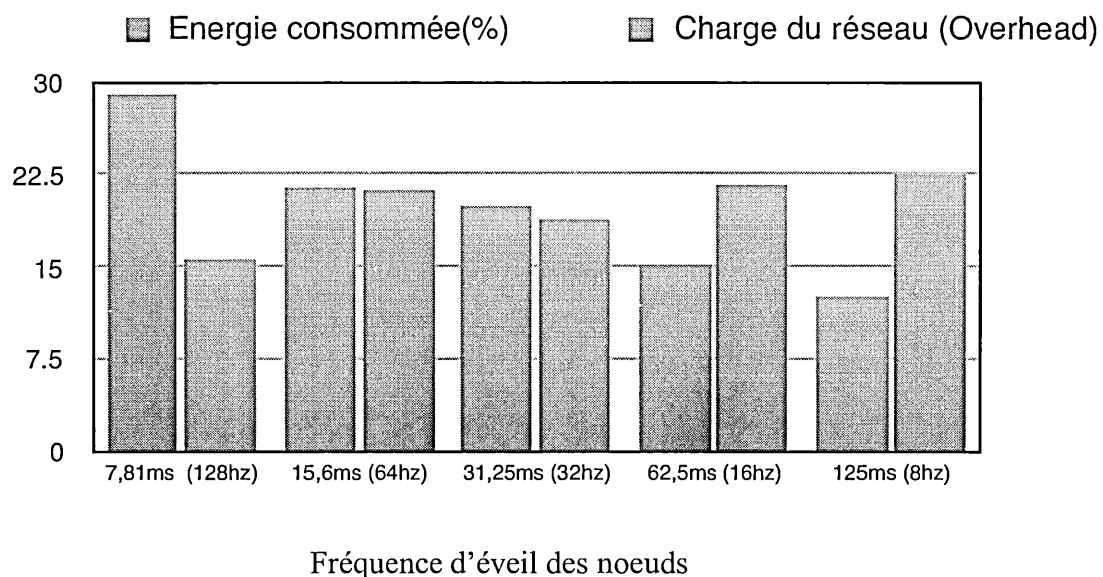


Figure 22 Illustration de la charge du réseau et de l'énergie consommée en fonction de la fréquence d'éveil

6. Discussion

Dans ce chapitre, nous avons présenté les résultats des différentes simulations effectuées afin d'analyser le comportement de la solution proposée. Nous constatons que la fonction objectif utilisée dans notre étude répond à notre objectif, car en comparant les résultats obtenus dans le profil I et II, nous constatons que la consommation énergétique n'est pas distribuée de la même manière. En effet, la solution proposée aide à équilibrer le niveau d'énergie des différents nœuds du réseau au fur et à mesure que le nœud mobile se déplace. Comme déjà énoncé, l'objectif principal de notre étude est de maintenir une connectivité continue entre le réseau et le nœud mobile afin de limiter les pertes de paquets tout en maintenant un équilibre énergétique sur toute la durée de vie du réseau. Nos résultats de simulation montrent que la solution proposée permet de maximiser cette connectivité avec un algorithme de basculement efficace et contribue aussi à limiter les pertes de paquets. Cette solution ne permet pas nécessairement d'économiser l'énergie globale du réseau, mais elle rend la mobilité des nœuds plus consciente des ressources disponibles.

Nous constatons également qu'un compromis doit être fait afin que la solution proposée réponde à des applications particulières. Ce compromis met en jeu des variables telles que la vitesse du nœud mobile, la fréquence du cycle de service et la vitesse d'envoi des données. En effet, les applications qui ont pour critère principal la vitesse doivent diminuer le cycle d'éveil des nœuds afin d'augmenter le taux d'acheminement des paquets, mais subiront dans ce cas une consommation énergétique plus importante. Si l'économie d'énergie est une condition importante dans le cahier des charges, l'implémentation devra prendre en compte des limitations de vitesse et des cycles d'éveils plus longs et par conséquent supporter un taux de perte de données élevé. Cependant, quelle que soit la priorité accordée, nous pouvons garantir que la solution proposée maintiendra la disponibilité des nœuds le plus longtemps possible afin de véhiculer les paquets de données.

CHAPITRE VI

CONCLUSION

Dans ce projet, nous avons abordé des thèmes apparentés à la mobilité et à la gestion de l'énergie dans le domaine des réseaux de capteurs sans fil. La recherche conduite s'est attardée sur la manière de prolonger la durée de vie d'un réseau de capteurs mobiles en évitant une consommation énergétique disproportionnée entre les nœuds. Ce raisonnement permet de maintenir l'ensemble des nœuds du réseau opérationnels afin de répondre au besoin d'acheminement de paquets envoyés par le nœud mobile. Se faisant, le taux de pertes de paquets est diminué et la durée de vie du réseau est prolongée. La solution proposée comporte deux parties. La première a été développée de sorte à prendre en charge la mobilité des nœuds dans le réseau en maximisant le temps de connexion entre eux-ci. Cette prise en charge est basée principalement sur l'énergie résiduelle des nœuds. La deuxième partie opère selon un algorithme qui permet de limiter le nombre de basculements dans le réseau pour éviter les coupures fréquentes et les pertes de données qui peuvent en résulter.

À la lumière de la littérature portant sur notre sujet, cette étude s'est imposée afin de prendre en considération les exigences énergétiques d'un réseau de capteurs mobiles. L'énergie est une ressource précieuse dans les réseaux de capteurs. Pour cette raison, il est indispensable de la considérer dans la conception d'une solution de mobilité.

Les sous-objectifs visés dans notre étude ont été atteints. D'abord, nous avons équilibré la consommation énergétique des différents nœuds au fur et à mesure que le nœud mobile se déplace. Pour y parvenir, nous avons développé un algorithme écoénergétique qui base la décision de basculement d'un nœud mobile sur l'énergie résiduelle des nœuds statiques. Enfin, nous avons proposé un algorithme qui réduit le nombre de basculements pour diminuer les déconnexions dans le réseau.

Afin que cette étude soit positionnée par rapport à l'état de l'art, nous avons recensé les travaux issus de sources scientifiques durant ces deux dernières décennies. À cet effet, Ce document comporte des études comparatives des différents protocoles de routage et systèmes d'exploitation embarqués, illustrées sous forme de tableaux et figures pour faciliter la synthèse des informations présentées. De plus, certains domaines d'applications ont été cités en exemple pour mieux baliser le contexte d'utilisation de la solution proposée.

La méthodologie suivie dans ce travail a permis de collecter un certain nombre d'informations pertinentes. La mise en contexte de celles-ci a abouti à des résultats qui ont confirmé les hypothèses fixées au départ. En effet, nous sommes parvenus à maintenir un niveau de consommation énergétique équilibré tout au long de la durée de vie du réseau, et ce, en utilisant le niveau du signal RSSI et le protocole de routage RPL. Cependant, notre étude comporte des limites. Nous notons, par exemple, sa portabilité sur d'autres protocoles de routage. En effet, cette solution opère en harmonie avec le protocole RPL, mais elle requiert l'adaptation des messages de contrôles dans le cas où un autre protocole de routage est utilisé. Également, la solution proposée n'a pas fait l'objet d'une validation interne sur banc d'essai ni d'analyses statistiques des sources d'erreurs.

À la lumière de ces limites, nous préconisons, pour d'éventuels travaux futurs, de considérer d'autres protocoles de routage afin d'adapter la solution proposée à d'autres environnements. De plus, l'algorithme proposé peut être amélioré pour simuler plusieurs nœuds mobiles à la fois. La gestion des données envoyées par les nœuds mobiles sera certes plus complexe, mais ce scénario sera plus réaliste. La densité du réseau statique peut être plus ou moins étendue afin de simuler une telle logique sur de plus grandes superficies. Également, il pourrait être pertinent d'implémenter ces solutions sur des plateformes matérielles pour confirmer les résultats obtenus.

En définitive, cette recherche s'ajoute aux différents travaux de la littérature scientifique qui traitent des réseaux de capteurs sans fil. Elle nous permet de constater qu'une association entre les avantages de la mobilité et les contraintes énergétiques sont envisageables. Les projets de recherche connexes que nous recommandons permettront à d'autres chercheurs d'implémenter des solutions prêtes à l'emploi.

BIBLIOGRAPHIE

Alexander, L., Simon, A. et Nathaniel L. B. (2013). IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs): Overview, Assumptions, Problem Statement; and Goals Status. *Journal of Chemical Information and Modeling* 1(36).

Bag, G., Muhammad, T.R, Ki-Hyung, K et Seung-Wha, Y. (2009). LoWMob: Intra-PAN Mobility Support Schemes for 6LoWPAN. *Sensors (Basel, Switzerland)* 9(7): 5844–77.

El Korbi, I., Ben Brahim, M. Adjih, C et Azouz Saidane, L. (2010). Mobility Enhanced RPL for Wireless Sensor Networks. *2012 Third International Conference on The Network of the Future*. 978-1-4673-5265-9.

Berners, T et Fielding, F. (2013). Uniform Resource Identifier (URI): Generic Syntax Status. *Journal of Chemical Information and Modeling* 53: 1689–99. doi:10.1017/CBO9781107415324.004.

Deering, S., et R Hinden. (2013). Internet Protocol, Version 6 (IPv6) Specification Status. *Journal of Chemical Information and Modeling* 53:1689–99. doi:10.1017/CBO9781107415324.004.

Perkins, C., Belding-Royer, E et Das, S. (2013). Ad Hoc On-Demand Distance Vector (AODV) Routing. *Journal of Chemical Information and Modeling* 53: 1689–99. doi:10.1017/CBO9781107415324.004.

Dunkels, A. (2011). The ContikiMAC Radio Duty Cycling Protocol. *SICS Technical Report T2011:13*: 1–11. <http://dunkels.com/adam/dunkels11contikimac.pdf>.

Gee Keng, E., Chee Kyun, N. Kamariah, N et Borhanuddin, M.A. (2010). A Review of 6LoWPAN Routing Protocols. *Proceedings of the Asia-Pacific Advanced Network* 30 (12): 71. doi:10.7125/APAN.30.11.

Kim, E., Kaspar, D. Gomez, C. Bormann, C. (2012) Problem Statement and Requirements for IPv6 over Low-Power Wireless Personal Area Network (6LoWPAN) Routing. *Internet Engineering Task Force (IETF) Request for Comments: 6606*. 33: 3–8. doi:10.1073/pnas.0703993104.

Levis, P., Clausen, T. Hui, J. Gnawali, O et Ko, J. (2012). The Trickle Algorithm. *Internet Engineering Task Force (IETF) Request for Comments 6206*. doi:10.1073/pnas.0703993104.

Fielding, R.T. (2000). Architectural Styles and the Design of Network-based Software Architectures. *Doctoral dissertation; University of California, Irvine*.

Gaddour, O., Koubaa, A. Raghuraman, R. Cheikhrouhou, O.Tovar, E et Abid, M. (2014). Co-RPL : RPL Routing for Mobile Low Power Wireless Sensor Networks Using Corona Mechanism. *Proceedings of the 9th IEEE International Symposium on Industrial Embedded Systems (SIES 2014)*. 978-1-4799-4023-3. DOI: 10.1109/SIES.2014.6871205

Herberg, U., et Clausen, T. (2011). A Comparative Performance Study of the Routing Protocols LOAD and RPL with Bi-Directional Traffic in Low-Power and Lossy Networks (LLN). *Proceedings of the 8th ACM Symposium on Performance Evaluation of Wireless Ad Hoc, Sensor, and Ubiquitous Networks*, 73–80. doi: 10.1145/2069063.2069076.

JeongGil, K et Chang, M. (2014). MoMoRo: Providing Mobility Support for Low-Power Wireless Applications. *IEEE Systems Journal*, 1–10. doi:10.1109/JSYST.2014.2299592.

Kosanovi, M.R, et Stoj, M.K.(2011). Connecting Wireless Sensor Networks to Internet. *Mechanical Engineering* 9: 169–82.

Kevin, C., Raghuram, S. Dai, L. Addepalli, S et Gerla, M. (2012). RPL under Mobility. *2012 IEEE Consumer Communications and Networking Conference (CCNC)*, 300–304. doi:10.1109/CCNC.2012.6181106.

Levis, P. Madden, S. Polastre, J. Szewczyk, R. Whitehouse, K. Woo, A. Gay, D et al. (2005). TinyOS: An Operating System for Wireless Sensor Networks. *Ambient Intelligence*, 115–48. doi:10.1007/3-540-27139-2_7.

Ming, L. Cao, J. Chen, C et Xiaomin, W.(2007). An Energy-Aware Routing Protocol in Wireless Sensor Networks. *2007 International Conference on Signal Processing, Communications and Networking (2007)*. 9(1): 445–62. doi:10.3390/s90100445.

Mikhaylov, K., Plevritakis, N. et Tervonen, J. (2013). Performance Analysis and Comparison of Bluetooth Low Energy with IEEE 802.15.4 and SimpliciTI. *Journal of Sensor and Actuator Networks* 2(3): 589–613. doi:10.3390/jsan2030589.

Moreira, D., Fotouhi, H. et Alves, M. (2014). Mobile IoT: Smart-HOP over RPL. CISTER, Research Center in Real-time & Embedded Computing Systems .

Moteiv, Co. (2006). Moteiv: Tmote Sky Low Power Wireless Sensor Module. *Product Data Sheet*, 1(12). doi:6020-0094-01

Winter, et Al. (2014). RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks. *Internet Engineering Task Force (IETF) Request for Comments: 6550* no. 1: 1–5. doi:10.1007/s13398-014-0173-7.2.

Shelby, Z., Hartke, K. et Bormann, C. (2014). The Constrained Application Protocol (CoAP) Abstract. *Internet Engineering Task Force (IETF) Request for Comments: 7252*. 1: 1–5. doi:10.1007/s13398-014-0173-7.2.

Edosoft Factory, S.L (Edosoft). (2012). Contiki and Tinyos. *ISN - Interoperable Sensor Networks - Deliverable 4*: 1–15.

Österlind, F., Dunkels, A., Eriksson, J., Finne, N. et Voigt, T. (2006). Cross-Level Sensor Network Simulation with COOJA. *Proceedings - Conference on Local Computer Networks, LCN*, 641–48. doi:10.1109/LCN.2006.322172.

Mallikarjuna, A., Kumar, P., Janakiram, D. et Kumar, A. (2009). Wireless Sensor Network Operating Systems: A Survey. *International Journal of Sensor Networks* 5 (4): 236. doi:10.1504/IJSNET.2009.027631.

Shah, R.C. et Rabaey, J.M. (2002). Energy Aware Routing for Low Energy Ad Hoc Sensor Networks. *2002 IEEE Wireless Communications and Networking Conference Record WCNC* 1: 350–55. doi:10.1109/WCNC.2002.993520.

Siekinen, M., Hienkari, M., Nurminen, J.K. et Nieminen, J. (2012). How Low Energy Is Bluetooth Low Energy? Comparative Measurements with ZigBee/802.15.4. *2012 IEEE Wireless Communications and Networking Conference Workshops, WCNCW*. 232–37. doi:10.1109/WCNCW.2012.6215496.

Silva, R., Sa Silva, J et Boavida, F. (2012). A Proposal for Proxy-Based Mobility in WSNs. *Computer Communications* 35(10): 1200–1216. doi:10.1016/j.comcom.2012.03.005.

Vasseur, J.P., Agarwal, N. Hui, J. Shelby, Z. Bertrand, P et Cedric Chauvenet. (2011), RPL: The IP Routing Protocol Designed for Low Power and Lossy Networks, *Internet Protocol for Smart Objects (IPSO) Alliance (2011)*.

Vidhyapriya, R., et Vanathi, P.T. (2007). Energy Aware Routing for Wireless Sensor Networks. *2007 International Conference on Signal Processing, Communications and Networking*. doi:10.1109/ICSCN.2007.350661.

Vucinic, M., Tourancheau, B et Duda, A. (2013). Performance Comparison of the RPL and LOADng Routing Protocols in a Home Automation Scenario. *IEEE Wireless Communications and Networking Conference, WCNC, 1974–79*. doi:10.1109/WCNC.2013.6554867.

Wang, H. (2011). M2M COMMUNICATIONS A SYSTEMS APPROACH. *Communication Technology and Application (ICCTA 2011), IET International Conference*. doi:10.1049/cp.2011.0829.

Zigbee Alliance, récupéré en 2013, <http://www.zigbee.org/the-latest/>

Younis, O., et Fahmy, S. (2004). HEED: A Hybrid, Energy-Efficient, Distributed Clustering Approach for Ad Hoc Sensor Networks. *IEEE Transactions on Mobile Computing* 03(4): 366–79. doi:10.1109/TMC.2004.41.

Heinzelman, W. Chandrakasan, A et Balakrishnan, H. (2000). Energy-Efficient Communication Protocol for Wireless Microsensor Networks. *Proceedings of the 33rd Annual Hawaii International Conference on System Sciences* 00(c): 3005–14. doi:10.1109/HICSS.2000.926982.

Farooq, M. Kunz, T. (2011). Operating Systems for Wireless Sensor Networks: A Survey. *SENSORS ISSN 1424-8220*.

ANNEXE I

Il existe une multitude de protocoles de routage qui peuvent être utilisés dans un scénario de mobilité d'Internet des objets, cependant, des modifications majeures doivent leur être apportées pour leur intégrer la solution que nous proposons dans ce projet. Comme cité précédemment, nous présentons dans cette annexe les protocoles de routage que nous avons considéré pour notre projet en plus de RPL, puis nous récapitulons les caractéristiques de chacun dans un tableau afin d'appuyer le choix de RPL.

- LOAD

Le protocole 6LoWPAN Ad-hoc On-demand Distance Vector Routing (LOAD) est un protocole de routage dédié aux WSN basé sur AODV, sa consommation énergétique très faible et les types de trafics qu'il offre le rendent intéressant pour un scénario comme le notre. Si un équipement expéditeur désire envoyer un paquet, il doit initier une requête en envoyant un message de requête de route (en anglais Route Request Message, ou RREQ) distribuée dans le réseau. Les nœuds intermédiaires inscrivent cette route dans leur table de routage afin de l'utiliser ultérieurement comme route par défaut vers l'expéditeur. Une fois la RREQ arrivée au nœud destination, celui-ci répond avec un message de réponse RREP (en anglais Route-Reply) destinée au nœud expéditeur. Si un paquet n'atteint pas la destination, un message au niveau de la couche de liaison notifie la perte du paquet et le réseau en déduit une perte du lien. Le protocole LOAD comporte une procédure de réparation de route (en anglais Route-Repair) qui permet de rétablir les liens, cependant, cette procédure ne sera évidemment pas utilisée dans notre projet si jamais ce protocole de routage est choisi, car nous proposons un mécanisme de basculement qui gère la mobilité des nœuds.

- DYMO-low

Dynamic MANET On-demand for 6LoWPAN Routing est également un protocole de routage basé sur AODV, il utilise les mêmes messages de contrôles et le même mécanisme de peuplement des tables de routage que AODV. Cependant, DYMO supervise la qualité des liens grâce à des « HELLO Message » en utilisant le protocole de transport (UDP). Ce protocole est caractérisé par une consommation en mémoire et en énergie importante et ne peut donc pas supporter le protocole 6LoWPAN. Pour cela, la version DYMO-low a été proposée afin de créer un réseau de topologie mesh au niveau de la couche de liaison et de permettre à la couche IP de percevoir les différents nœuds comme étant dans un lien unique où ils partagent le même préfixe IPV6. Ce protocole consomme très peu d'énergie ce qui le rend attrayant pour une éventuelle utilisation dans notre projet.

- HiLow

HiLow est un protocole de routage développé pour accroître l'extensibilité des réseaux, il utilise des adresses de 16-bits afin d'économiser l'espace mémoire et permettre d'augmenter le nombre de nœuds. Si un nouveau nœud désire rejoindre le réseau HiLow, ce protocole commence par scanner les nœuds 6LoWPAN voisins, si aucun nœud n'est détecté, il est considéré alors comme nœud coordinateur avec une adresse courte de « 0 ». HiLow utilise une organisation hiérarchique où chaque nœud connaît sa « profondeur » dans le réseau. Cette organisation correspond parfaitement au scénario de collecte de données que nous voulons simuler dans notre projet. HiLow ne dispose cependant pas de procédure de réparation de route en cas de coupure de lien.

- CTP

Collection Tree Protocol est un protocole de routage destiné aux réseaux WSN, il fonctionne selon un principe de vecteur de distance (Distance Vector) et il calcule la qualité du lien de chaque nœud du réseau vers le nœud principal (Root) en utilisant les trois mécanismes suivants : le premier est basé sur une estimation des liens issue des informations collectées de la couche physique et la couche de liaison, cette estimation est véhiculée dans les messages de contrôles. Le deuxième mécanisme sert à valider les liens en utilisant les paquets de données ainsi que les accusés de réception afin de s'assurer de la qualité des liens. Le dernier mécanisme repose sur un système de beacons variable qui augmente l'intervalle d'envoi au fur et à mesure que la topologie se stabilise. Ce dernier mécanisme donne à CTP le caractère proactif dont notre projet a besoin, en effet, CTP procède à la découverte de la topologie de manière implicite sans attendre la perte d'un paquet ou même une sollicitation externe. Ceci est indispensable dans notre projet afin de préparer le réseau à recevoir des paquets de données du nœud mobile en tout temps.