



Configuration dynamique et routage pour l'internet des objets

Patrick Olivier Kamgueu

► To cite this version:

Patrick Olivier Kamgueu. Configuration dynamique et routage pour l'internet des objets. Réseaux et télécommunications [cs.NI]. Université de Lorraine, 2017. Français. NNT : 2017LORR0241 . tel-01687704

HAL Id: tel-01687704

<https://theses.hal.science/tel-01687704>

Submitted on 18 Jan 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



AVERTISSEMENT

Ce document est le fruit d'un long travail approuvé par le jury de soutenance et mis à disposition de l'ensemble de la communauté universitaire élargie.

Il est soumis à la propriété intellectuelle de l'auteur. Ceci implique une obligation de citation et de référencement lors de l'utilisation de ce document.

D'autre part, toute contrefaçon, plagiat, reproduction illicite encourt une poursuite pénale.

Contact : ddoc-theses-contact@univ-lorraine.fr

LIENS

Code de la Propriété Intellectuelle. articles L 122. 4

Code de la Propriété Intellectuelle. articles L 335.2- L 335.10

http://www.cfcopies.com/V2/leg/leg_droi.php

<http://www.culture.gouv.fr/culture/infos-pratiques/droits/protection.htm>

Configuration Dynamique et Routage pour l'Internet des Objets

THÈSE EN COTUTELLE

présentée et soutenue publiquement le 18 Décembre 2017

pour l'obtention du

Doctorat de l'Université de Lorraine (France)

et de l'Université de Yaounde 1 (Cameroun)

(mention informatique)

par

Patrick Olivier KAMGUEU

Composition du jury

<i>Président :</i>	Charles AWONO ONANA	Professeur – Université de Yaoundé 1
<i>Rapporteurs :</i>	Nathalie MITTON Yacine GHAMRI-DOUDANE	Directrice de Recherche – INRIA Lille Professeur – Université de la Rochelle
<i>Examinatrice :</i>	Noura LIMAM	Chargée de Recherche – Univ. de Waterloo (Canada)
<i>Encadrants de thèse :</i>	Olivier FESTOR Maurice TCHUENTE Emmanuel NATAF Thomas DJOTIO	Professeur – Université de Lorraine Professeur – Université de Yaoundé 1 MCF – Université de Lorraine MCF – Université de Yaoundé 1

Résumé

L'intérêt croissant de la communauté scientifique et industrielle ces dernières années pour les réseaux de capteurs sans fil (RCSF), a conduit à la définition de nouveaux protocoles normalisés prenant en compte les spécificités matérielles des nœuds utilisés. Dans la couche réseau, le protocole RPL (de l'acronyme anglais *IPv6 Routing Protocol for Low-power and Lossy Network*) a été proposé en 2012 par l'IETF, comme standard de routage pour les réseaux dont les nœuds sont de type "LLN" (Low-power and Lossy Network), *i.e.* caractérisés par une faible autonomie énergétique et transmettant sur des liens radios dotés d'un taux de perte de données élevé. Dans cette thèse, nous nous intéressons à l'optimisation du routage dans ces réseaux (notamment ceux utilisant la pile protocolaire TCP/IP), ainsi qu'à leur interconnexion efficace à Internet à des coûts soutenables. Tout d'abord, nous proposons deux fonctions d'objectif organisant le routage avec RPL. La première se sert de l'unique critère énergétique, avec comme objectif principal la maximisation de la durée de vie du réseau. Pour ce faire, nous avons implémenté un modèle d'estimation d'énergie, intégré par la suite aux nœuds pour leur permettre d'estimer en temps réel leur énergie résiduelle. La deuxième fonction d'objectif proposée, vise à combiner plusieurs critères pour la prise en compte de la qualité de service durant le routage. Nous développons un modèle à base de la logique floue pour mettre en œuvre la combinaison. En effet, elle nous permet d'obtenir un *bon* compromis entre les différentes entrées et requiert une empreinte mémoire faible. Dans la dernière partie de cette thèse, nous concevons et mettons en œuvre une architecture d'activation de passerelles permettant d'assurer une connexion Internet efficace de divers RCSFs utilisant RPL, pour la réalisation de la vision de l'Internet des Objets.

Abstract

In recent years, the growing interest of scientific and industrial community has led to the standardization of new protocols that take into account the unique requirements of Wireless Sensor Networks (WSN) nodes. At network layer, RPL (*IPv6 Routing Protocol for Low-power and Lossy Network*) has been proposed by IETF as the routing standard for network that uses LLN nodes, namely, those where both nodes and their interconnects are constrained. They operate on low-power embedded batteries and use lossy links, making communications unreliable and lead to a significant data loss rates. In this thesis, we aim at optimizing the routing in WSNs (especially those using TCP/IP protocol stack), as well as their efficient and cost effective connection to the Internet. First of all, we proposed two new RPL objective functions. The first uses as unique routing criterion, the node remaining energy with the goal of maximizing the network lifetime. An energy model that allows the nodes to dynamically estimate their remaining energy at runtime has been implemented and integrate to the protocol. The second objective function uses fuzzy logic reasoning to combine several criteria in order to take Quality of Service into account. Indeed, this scheme provides a *good* tradeoff on several inputs and requires a low memory footprint. In the last part of this thesis, we designed and implemented an architecture that enable an efficient integration of several RPL based WSNs to the Internet in order to achieve the Internet of Things vision.

Remerciements

Le travail réalisé tout au long de cette thèse n'a été possible qu'avec l'aide et le soutien de nombreuses personnes. Je profite de l'occasion qui m'est donnée ici pour leur exprimer mes sincères remerciements.

Je voudrais tout d'abord témoigner ma profonde gratitude à mon encadrant de thèse Dr. Emmanuel Nataf pour avoir cru en mon potentiel. Sans ses encouragements et son aide permanente, ce travail n'aurait certainement jamais été mené jusqu'au bout. « Merci pour avoir partagé avec moi votre goût de la recherche et votre passion des réseaux. » Je remercie également mes co-directeurs de thèse, les professeurs Thomas Djojio, Olivier Festor et Maurice Tchuenté pour la disponibilité et le grand professionnalisme dont ils ont fait preuve durant toutes ces années. Je les remercie pour leurs précieux conseils et pour avoir accepté d'encadrer cette thèse. J'ai eu la chance de rencontrer des personnes passionnées par leur métier et c'est grâce à elles que j'ai pu obtenir le cadre institutionnel dans lequel ce travail a été mené (cotutelle entre les universités de Lorraine – France et de Yaoundé 1 – Cameroun).

Je remercie le Professeur Yacine Ghamri-Doudane et Dr/HDR Nathalie Mitton pour avoir expertisé ce travail. Je remercie également le Professeur Charles Awono Onana et Dr Noura Limam d'avoir accepté de faire partie de mon jury de soutenance et pour avoir examiné cette thèse.

Si dans une telle aventure, les interactions scientifiques sont fondamentales, les rapports humains n'en demeurent pas moins importants. Je voudrais ici remercier mes camarades des équipes MADYNES et MASECNESS (devenu IoT4D), ceux qui sont encore là et ceux qui sont déjà partis, pour l'ambiance qui a régné, surtout pendant les pauses café et déjeuné :-). Je remercie également mes collègues du département informatique à l'université de Yaoundé 1, pour avoir accepté de me suppléer dans mes tâches académiques durant mes nombreuses missions de recherche en France. Je ne citerai personne de peur d'en oublier, toutefois j'ai une pensée particulière pour mon étudiant de Master 2, Njine Chuangué pour sa précieuse aide dans ce sens.

Je tiens également à remercier les assistances de l'équipe MADYNES, Céline Simon et Isabelle Herlich pour leur soutien administratif lors de mes voyages entre Yaoundé et Nancy. J'ai une gratitude pour les institutions suivantes : Ministère Français des Affaires Étrangères (Bourse SCAC), LIRIMA (Laboratoire International pour la Recherche en Informatique et Mathématiques Appliquées) et INRIA (Institut National de Recherche en Informatique et Automatique). Leur soutien financier et logistique a rendu possible la réalisation de cette thèse.

Je ne saurais terminer mon propos sans remercier ma famille pour son soutien, son amour inconditionnel et permanent. Un merci tout particulier à ma chérie Ornella qui m'a soutenu durant ces années de thèse, surtout la dernière, longue et interminable ligne droite. Elle a su me remonter le moral pendant les moments de découragement et a pu supporter mes sautes d'humeur. Le meilleur est à venir ...

Publications

Les contributions réalisées dans cette thèse ont fait l'objet de plusieurs publications dans des revues et conférences scientifiques internationales.

Revues internationales

- ✓ P. O Kamgueu, E. Nataf et T. Djotio, « *Architecture for an efficient integration of Wireless Sensor Networks to the Internet through IoT gateways* », International Journal of Distributed Sensor Networks – Special collection on Cloud Computing and Communication Protocols for IoT Applications, Vol. 13(11) 2017.
<https://doi.org/10.1177/1550147717744735>
- ✓ P. O Kamgueu, E. Nataf et T. Djotio, « *Survey on RPL enhancements : a focus on topology, security and mobility* », à paraître dans une prochaine série (2018) d'Elsevier – Computer Communications.

Conférences internationales

- ✓ P. O Kamgueu, E. Nataf et T. Djotio, « *On Design and Deployment of Fuzzy-Based Metric for Routing in Low-Power and Lossy Networks* », actes du 40^{ème} IEEE Conference on Local Computer Networks (LCN 2015) pages 789 – 795, Clearwater Beach, FL, USA.
<https://doi.org/10.1109/LCNW.2015.7365929>
- ✓ P. O Kamgueu, E. Nataf, T. Djotio et O. Festor, « *Fuzzy-based routing metrics combination for RPL* », Doctoral Consortium of 3th international Conference on Sensor Networks (DCSENSORNETS 2014) pages 11 – 17, Lisbon, Portugal.
<https://hal.inria.fr/hal-01093965/document>
- ✓ P. O Kamgueu, E. Nataf, T. Djotio et O. Festor, « *Energy-based metric for the Routing Protocol in Low-Power and Lossy Network* », actes du 2nd international Conference on Sensor Networks (SENSORNETS 2013) pages 145 – 148, Barcelona, Spain.
<https://doi.org/10.5220/0004313401450148>

Table des matières

Résumé	iii
Remerciements	v
Publications	vii
1 Introduction Générale	1
I Contexte et État de l'Art	5
2 Routage dans les Réseaux de Capteurs sans Fil	7
2.1 Problématique de routage dans les RCSF	7
2.1.1 Facteurs influençant la conception du routage dans les RCSFs . .	9
2.1.1.1 Type de déploiement	9
2.1.1.2 Consommation énergétique	9
2.1.1.3 Modèle de transmission de données	9
2.1.1.4 Hétérogénéité et capacité des nœuds	10
2.1.1.5 Agrégation de données	10
2.1.1.6 Tolérance aux pannes	10
2.1.1.7 Passage à l'échelle	11
2.1.1.8 Dynamique réseau	11
2.1.1.9 Qualité de Service	11
2.1.2 Classification des protocoles de routage dans les RCSF	11
2.1.2.1 Routage à « plat »	12
2.1.2.2 Routage hiérarchique	12
2.1.2.3 Routage basé sur la position des nœuds	12
2.1.2.4 Routage orienté-données	13
2.1.2.5 Routage multi-chemins	13
2.2 Métriques de routage	13
2.2.1 Types de métriques	13
2.2.1.1 Métrique de nœud ou de lien	14
2.2.1.2 Dynamique ou Statique	14
2.2.1.3 Métrique unidimensionnelle ou multidimensionnelle . .	14
2.2.2 Considérations liées à l'implémentation	15
2.2.3 Métriques usuelles pour les RCSF	15
2.2.3.1 Le nombre de sauts	16
2.2.3.2 L'ETX et métriques apparentées	16
2.2.3.3 Le délai	17
2.2.3.4 La taille de file d'attente des paquets	18

2.2.3.5	L'énergie	18
2.2.3.6	Le RSSI	19
2.3	Standardisation	19
2.3.1	Standards de l'IETF relatifs aux RCSF	20
2.3.2	Le protocole de routage RPL	21
2.3.2.1	Formation de la topologie	22
2.3.2.2	Messages de contrôle RPL	23
2.3.2.3	Maintenance de la topologie	24
2.3.2.4	Rang, détection et gestion des boucles de routage RPL	25
2.3.2.5	Fonction d'objectif RPL	25
3	Du Réseau de Capteurs à l'Internet des Objets	27
3.1	Technologies de l'Internet des Objets	27
3.1.1	La Radio Identification (RFID)	29
3.1.2	L'IEEE 802.15.4	29
3.1.3	BLE	30
3.1.4	IPv6	31
3.1.5	Web des Objets et CoAP	32
3.2	Problématique de l'interconnexion	33
3.2.1	Routeur de bordure	34
3.2.2	Plate-forme matérielle pour la passerelle	36
3.3	Stratégies de sélection de passerelles dans les réseaux sans fil multi-sauts	36
II	Contribution au Routage	41
4	Routage par l'Énergie	43
4.1	Modèle d'estimation d'énergie pour les capteurs	43
4.1.1	Modèle linéaire	43
4.1.2	Modèle non linéaire	44
4.1.3	Estimation logicielle <i>en ligne</i>	46
4.1.4	Résultats de simulation du modèle non linéaire	47
4.1.4.1	Profil énergétique de démarrage	47
4.1.4.2	Profil énergétique selon la dynamique réseau	47
4.1.4.3	Erreur d'approximation	49
4.1.4.4	Durée de vie des nœuds	49
4.2	Fonction d'Objectif RPL basée sur l'énergie	50
4.2.1	Capacité énergétique de la route	51
4.2.2	Calcul de rang	52
4.2.3	Optimalité et absence de boucle	53
4.3	Expérimentations et évaluation	55
4.3.1	Environnement	55
4.3.2	Répartition énergétique	56
4.3.3	Fiabilité de transmission	57
4.4	Conclusion	58

5	Combinaison de métriques par la Logique Floue	61
5.1	Combinaison de métriques de routage RPL	62
5.1.1	Composition additive	62
5.1.2	Composition lexicographique	63
5.2	Système d'inférence par la logique floue	65
5.2.1	Variables linguistiques	65
5.2.2	Fuzzification	67
5.2.3	Inférence et agrégation	68
5.2.4	Defuzzification	70
5.3	Fonction d'objectif RPL basée sur la logique floue	70
5.3.1	Architecture du moteur d'inférence	70
5.3.2	Intégration à RPL	72
5.3.2.1	Logique floue	72
5.3.2.2	Structure de données pour la topologie de routage	72
5.3.2.3	Annonces de routage	73
5.3.2.4	Calcul de rang	73
5.4	Expérimentations et évaluation	74
5.4.1	Conditions expérimentales et outils	75
5.4.2	Taux de livraison	75
5.4.3	Stabilité de routage	76
5.4.4	Efficacité énergétique	77
5.4.5	Délai de bout-en-bout	78
5.5	Conclusion	78
III	Configuration et Interconnexion pour l'Internet des Objets	80
6	Réseau fédérateur de passerelles pour l'Internet des Objets	82
6.1	Introduction	82
6.2	Considérations liées à la conception	83
6.3	Modélisation	85
6.3.1	Modèle de réseau	85
6.3.2	Formulation du problème	86
6.4	Procédure de résolution	87
6.4.1	Complexité algorithmique	87
6.4.2	Méthode de résolution	88
6.4.3	Illustration	89
6.4.3.1	Scénario N°1	90
6.4.3.2	Scénario N°2	90
6.5	Architecture du système	91
6.6	Mise en œuvre et déploiement	93
6.6.1	Architecture matérielle	93
6.6.2	Fonctionnement réseau	94
6.6.2.1	Côté réseau de capteurs sans fils	94
6.6.2.2	Côté passerelle	94
6.6.3	Déploiement et cas d'utilisation	95
6.6.3.1	Simulation grande échelle	95
6.6.3.2	Expérience en environnement réel	96

6.7 Conclusion	98
7 Conclusion et Perspectives	100
7.1 Résumé des contributions	100
7.2 Perspectives	101

Table des figures

2.1	Composantes d'un réseau de capteurs sans fil	8
2.2	Standards relatifs au protocole RPL	20
2.3	Pile protocolaire standardisée pour les RCSF	21
2.4	Construction du DODAG	23
3.1	Évolution du nombre d'objets connectés de l'IoT [Eva11].	28
3.2	Convergence numérique [VF14].	28
3.3	Structure de la Super-Trame IEEE 802.15.4 en mode Synchrone	31
3.4	BAN : Réseau personnel sans fil pour l'individu [Ban].	32
3.5	Modes de fonctionnement Routeur du 6LBR [Der+13].	35
3.6	Modes de fonctionnement Pont Transparent du 6LBR [Der+13].	36
3.7	Passerelle par Association : Raspberry – TelosB	37
4.1	Premiers instants d'activité	48
4.2	Dynamique du profil énergétique	48
4.3	Gap entre les implémentations flottante et entière	49
4.4	Capacité énergétique d'une route	52
4.5	Calcul de <i>rang</i> de l'OF-énergie	52
4.6	Aperçu de la topologie réseau	56
4.7	Distribution de l'énergie résiduelle sur les nœuds	57
4.8	Évolution énergétique des nœuds les plus sollicités	57
4.9	Nombre de paquets reçus au point de collecte par nœud	58
5.1	Processus de sélection du prochain saut	66
5.2	Fonctions d'appartenances et ensembles flous	68
5.3	Composition et Agrégation pour la QoS	69
5.4	Defuzzification de la QoS	70
5.5	Moteur d'inférence floue	71
5.6	Architecture des structures de données pour la composition	73
5.7	DIO transportant l'ensemble des métriques	74
5.8	Déploiement des nœuds dans le bâtiment	75
5.9	Taux de perte de paquets au point de collecte	76
5.10	Nombre de changement de prochain saut dans le temps	77
5.11	Distribution de l'énergie résiduelle des nœuds	78
5.12	Fonction de répartition du délai de bout-en-bout	79
6.1	Réseau fédérateur de passerelles interconnectant les RCSF	84
6.2	Arborescence de routage pour diverses données en entrée	91
6.3	Architecture du système	92
6.4	Simulation Cooja d'un réseau grande échelle	96

6.5	Capteurs découverts de la passerelle [instance #20]	97
6.6	Expérience pour la domotique : 1 ^{ère} Passerelle – Phase 1	98
6.7	Expérience pour la domotique : 2 ^{nde} Passerelle – Phase 2	98

Liste des tableaux

4.1	Constantes du modèle	46
4.2	Mesures des consommations de courant : TelosB (sky)	47
4.3	Durée d'utilisation de la Batterie	50
4.4	Calcul détaillé du rang	53
4.5	Fiabilité de livraison des paquets	58
5.1	Variable linguistique de sortie QdS	69
5.2	Variable linguistique de sortie : QUALITÉ	71
5.3	Pile protocolaire	76
6.1	Paramètres de simulation	95

Liste des Abréviations

4G	4th Generation broadband cellular network (encore connu sous le nom LTE)
6LBR	IPv6 Low-power Border Router
6LowPAN	IPv6 over Low-power Wireless Personal Area Network
ARM	Advanced RISC Machine
BAN	Body Area Network
BLE	Bluetooth Low Energy
CCA	Clear Channel Assessment
CFLP	Capacitated Facility Location Problem
CoAP	Constrained Application Protocol
CoRE	Constrained RESTful Environments
CPU	Central Processing Unit
CSMA/CA	Carrier Sense Multiple Access with Collision Avoidance
DAO	Destination Advertisement Object
DHCP	Dynamic Host Configuration Protocol
DIS	DODAG Information Solicitation
DIO	DODAG Information Object
DODAG	Destination Oriented Direct Acyclic Graph
ETT	Expected Transmission Time
ETX	Expected Transmission Count
FFD	Full Function Device
HTTP	HyperText Transfer Protocol
IEEE	Institute of Electrical and Electronics Engineers
ICMP	Internet Control Message Protocol
IETF	Internet Engineering Task Force
IoT	Internet of Things (Internet des Objets <i>en français</i>)
IPv6	Internet Protocol version 6
LED	Light Emitting Diode
LLN	Low-power and Lossy Network
LPM	Low Power Mode
LQI	Link Quality Indication
MAC	Medium Access Control
NAT	Network Address Translation
ND	Neighbor Discovery Protocol
OF	Objective Function (<i>fonction d'objectif en français</i>)
OSI	Open Systems Interconnection
PAN	Personal Area Network
QoS	Qualité de Service
RCSF	Réseaux de Capteurs Sans Fil
RDC	Radio Duty Cycle
RFC	Request For Comments

RFD	Reduced Function Device
RFID	Radio Frequency IDentification
RPL	IPv6 Routing Protocol for Low-power and Lossy Network
RoLL	Routing over Low-power and Lossy Network
RTT	Round-Trip delay Time
Rx	Reception
SLAAC	StateLess Address AutoConfiguration
TCP	Transmission Control Protocol
Tx	Transmission
UDP	User Datagram Protocol
USB	Universal Serial Bus
WCETT	Weighted Cumulative Expected Transmission Time
WNM	Wireless Mesh Network
WPAN	Wireless Personal Area Network
WoT	Web of Things

Chapitre 1

Introduction Générale

L'évolution rapide des réseaux depuis quelques années, couplée aux nombreuses avancées technologiques dans différents secteurs, ont conduit à de nouveaux usages de l'Internet. Ceux-ci incluent de façon non exhaustive, la miniaturisation des architectures matérielles, l'informatique embarquée et les technologies de communication sans fil. Nous assistons à l'émergence d'un nouveau paradigme : l'*Internet des Objets* (IOT) où les entités du monde réel (personnes, bâtiments, véhicules, ...) couplées aux composants informatiques (ordinateurs, tablettes, smartphones) et objets embarqués dotés d'interfaces de mesures (camera, microphone, capteur d'humidité, de température, etc.) et d'actionneurs (moteurs, LED) sont connectés à l'Internet. Ceci donne la possibilité aux différents objets déployés de mettre à disposition leur données sur le web, mais aussi de pouvoir recevoir leur ordre à partir d'Internet. L'exploitation des services et données offerts par le Web permet ainsi l'émergence de nouvelles applications et améliorent ainsi la qualité de vie des individus. Parmi les plus significatives on peut citer celles liées aux domaines de l'e-santé, à la ville intelligente (transport, pollution, réseaux électriques, etc.), à la domotique, aux applications industrielles et la surveillance environnementale.

D'une part, de nombreux objets « intelligents » (car dotés de capacités d'acquisition d'information, de traitement et d'interaction avec leur environnement) sont déployés dans un environnement d'intérêt. Ils communiquent entre eux par des liens radio multisautes pour former un réseau de capteurs sans fil (RCSF). D'autre part, le RCSF lui-même peut être connecté à l'Internet à travers des dispositifs robustes constitués de passerelles qui assurent la communication entre les deux infrastructures (Internet et le réseaux sans fil). La volonté de disposer de nœuds (aussi appelés *motes*) en très grand nombre implique la nécessité d'en réduire les coûts de fabrication. On obtient de ce fait, des équipements peu robustes au niveau matériel : autonomie énergétique limitée, faible débit de données, mémoire de capacité réduite et puissance de calcul faible. Par ailleurs, la qualité du lien radio dans ces réseaux est mauvaise. Ceci est dû aux interférences avec les communications d'autres réseaux et dispositifs sans fil environnants ou aux obstacles, de même que la nature « bon marché » des composants matériels internes utilisés. Pour toutes ces raisons, la mise en réseau de ces objets est désignée sous le sigle LLN¹ (Low-Power and Lossy Network).

L'un des principaux défis que soulève les LLN est celui de l'énergie. Recharger la batterie des nœuds est très souvent difficile (emplacement des nœuds) ou économiquement non viable (déploiement à grande échelle). La prise en compte de la contrainte énergétique lors de la conception de tout protocole dédié à ce type d'environnement

1. Réseaux à basse consommation dotés de liens non fiables – i.e. avec pertes

est essentielle, car de la gestion efficace de l'énergie dépend la durée de vie du réseau. Un second défi à considérer lors de la mise en œuvre d'un RCSF est celui de la communication. Les nœuds sont en général déployés sans infrastructure dédiée et doivent s'auto-organiser pour faire remonter les informations aux différents points de collecte dans l'objectif de leur exploitation. La nature versatile du médium de communication combinée à la nécessité d'utiliser efficacement l'énergie embarquée dans la batterie des nœuds ainsi que d'autres contraintes matérielles, ont conduit à des efforts de standardisation d'un protocole de routage adapté à cet environnement. Cette tâche effectuée par le groupe de travail ROLL (Routing Over Low-Power and Lossy Network) de l'IETF a donné naissance au protocole RPL (IPv6 Routing Protocol for Low-Power and Lossy Network). Un des points clés lors de la construction de la topologie de routage est celui de savoir : « comment s'effectue le choix du prochain saut ? ». Le standard laisse ouvert la façon dont est réalisé ce choix, mais offre tout de même un cadre : celui de la fonction d'objectif (OF) pour le définir. Il indique également un certain nombre de métriques et contraintes pouvant orienter cette décision. Les nœuds doivent alors adapter leur fonctionnement selon les conditions imposées par l'environnement et l'application cible. Ces conditions sont multiples et peuvent se décliner en termes d'efficacité énergétique ou de qualité de la transmission (traduite sous forme de délai, de débit ou de taux de perte de données acceptable). Le problème de combiner les métriques en fonction des conditions propres à ces types de réseaux et au niveau de qualité de service requis par l'application cible reste ouvert.

Dans cette thèse, nous nous intéressons à l'optimisation du routage dans les réseaux de capteurs sans fil sous contrainte d'énergie. Nous avons dans un premier temps, conçu et mis en œuvre une fonction d'objectif RPL qui prend en compte l'unique critère énergétique pour organiser le routage. Le modèle d'énergie implémenté considère certains effets non linéaires qui se produisent à l'intérieur de la batterie du nœud pour estimer en temps réel (pendant l'exécution) la capacité énergétique résiduelle de celui-ci. La seconde contribution de cette thèse est la mise en œuvre d'une méthode de combinaison de métriques de routage qui soit réalisable dans l'environnement contraint en ressources dont dispose le nœud-capteur. Nous proposons la logique floue comme méthode de combinaison. En effet, elle nous permet de ne pas nous limiter aux seules métriques additives, et donne la possibilité de rechercher un bon compromis entre les différents critères à optimiser, selon la Qualité de Service (QoS) souhaitée par l'application cible. Nous avons alors conçu une seconde fonction d'objectif qui combine par la logique floue plusieurs métriques, que nous avons intégrée au standard de routage RPL. Des simulations et déploiements en environnement réel ont été effectués pour évaluer ces deux propositions. La dernière contribution consiste en la conception et l'implémentation d'une architecture permettant l'intégration à l'Internet traditionnel, de plusieurs RCSF pour la réalisation de la vision de l'Internet des Objets. Nous concevons cette architecture de façon à réaliser l'intégration visée à la fois de façon économique (au niveau du matériel requis) et robuste (adaptation dynamique à l'évolution du réseau – ajout et retrait de RCSFs ou de nœuds).

Ce manuscrit est organisé en sept chapitres subdivisés en trois parties. Dans la première partie, après le présent chapitre consacré à l'introduction, nous faisons au chapitre 2, un état de l'art de la problématique de routage dans les RCSF. La notion de

métrique de routage est présentée ainsi que l'étude des métriques les plus répandues. Les efforts de standardisation sont présentés en fin de chapitre. Nous poursuivons au chapitre 3 par la présentation du paradigme de l'Internet des Objets et des principales technologies associées. Nous abordons le problème de l'interconnexion des RCSF à l'Internet des Objets et les principaux travaux effectués dans la littérature pour sa mise en œuvre.

Dans la deuxième partie de cette thèse, nous détaillons les principales contributions réalisées pour l'optimisation du routage RPL. Au chapitre 4, nous décrivons le modèle d'énergie utilisé et son implémentation. Nous poursuivons par la conception de la fonction d'objectif proposée. Cette dernière utilise la capacité énergétique résiduelle des nœuds pour construire la topologie de routage. Au chapitre 5, nous présentons le modèle de combinaison de métrique de routage par la logique floue et sa mise en œuvre sous Contiki [DGV04], système d'exploitation largement utilisé pour les capteurs sans fil.

La dernière partie de cette thèse se consacre à la conception d'une architecture d'interconnexion des RCSF à l'Internet ainsi que sa réalisation grâce à des capteurs et passerelles à moindre coût. Nous terminons par une conclusion et quelques perspectives à notre travail.

Première partie

Contexte et État de l'Art

Chapitre 2

Routage dans les Réseaux de Capteurs sans Fil

2.1 Problématique de routage dans les RCSF

Un réseau de capteurs sans fil (RCSF) est constitué de plusieurs centaines voire milliers de nœuds. Ces derniers, déployés à grande échelle dans un environnement d'intérêt, mesurent les paramètres physiques de l'environnement et les transforment en signaux électriques. Le traitement de ces informations (mesures, événements, corrélation entre données collectées, ...) présente un grand intérêt pour de nombreuses applications. Typiquement, chaque nœud est doté d'une unité de traitement (le microcontrôleur), unité de captage (une ou plusieurs interfaces embarquées), unité de communication (puce radio) et une unité d'alimentation (batterie embarquée ou module de récupération d'énergie de l'environnement). La figure 2.1 illustre une architecture générique de réseau de capteurs. Du fait de leurs caractéristiques matérielles limitées, les capteurs acheminent les informations collectées vers un centre de traitement (plus puissant) à travers une ou plusieurs stations de base (puits ou *sink* en anglais). Les nœuds sont généralement dotés d'une interface radio ayant une portée limitée. Ils doivent par conséquent communiquer entre eux ou avec la station de base grâce à des liens radio multi-sauts.

Le routage consiste en un ensemble de mécanismes mis en œuvre par les nœuds pour sélectionner un chemin « *intéressant* » afin d'acheminer les données collectées aux différents points de conservation et d'exploitation. Dans les réseaux de capteurs, la tâche de routage soulève de nombreux défis inhérents à leurs caractéristiques particulières. Les problèmes posés diffèrent grandement de ceux rencontrés dans les autres types de réseaux sans fil tels que les réseaux mobiles Ad-hoc ou cellulaires [AY05]. Premièrement, les nœuds sont peu robustes et contraints en énergie, en puissance de calcul et capacité de stockage. Par conséquent, les ressources doivent être utilisées avec minutie. Deuxièmement, les capteurs sont disséminés de façon ad-hoc et doivent s'auto-organiser pour former le réseau et acheminer les informations. Troisièmement, dans un RCSF, le modèle de trafic dominant est de type multipoint à point (encore désigné par l'anglicisme *convergecast*). Le flux de données capturées est envoyé des différents points d'acquisition vers la station de base. Quatrièmement, les données collectées renferment un certain niveau de redondance. Plusieurs capteurs situés dans le même voisinage récolteront des mesures identiques concernant le phénomène étudié. L'exploitation de cette redondance par le routage contribuerait à améliorer la vie du réseau et

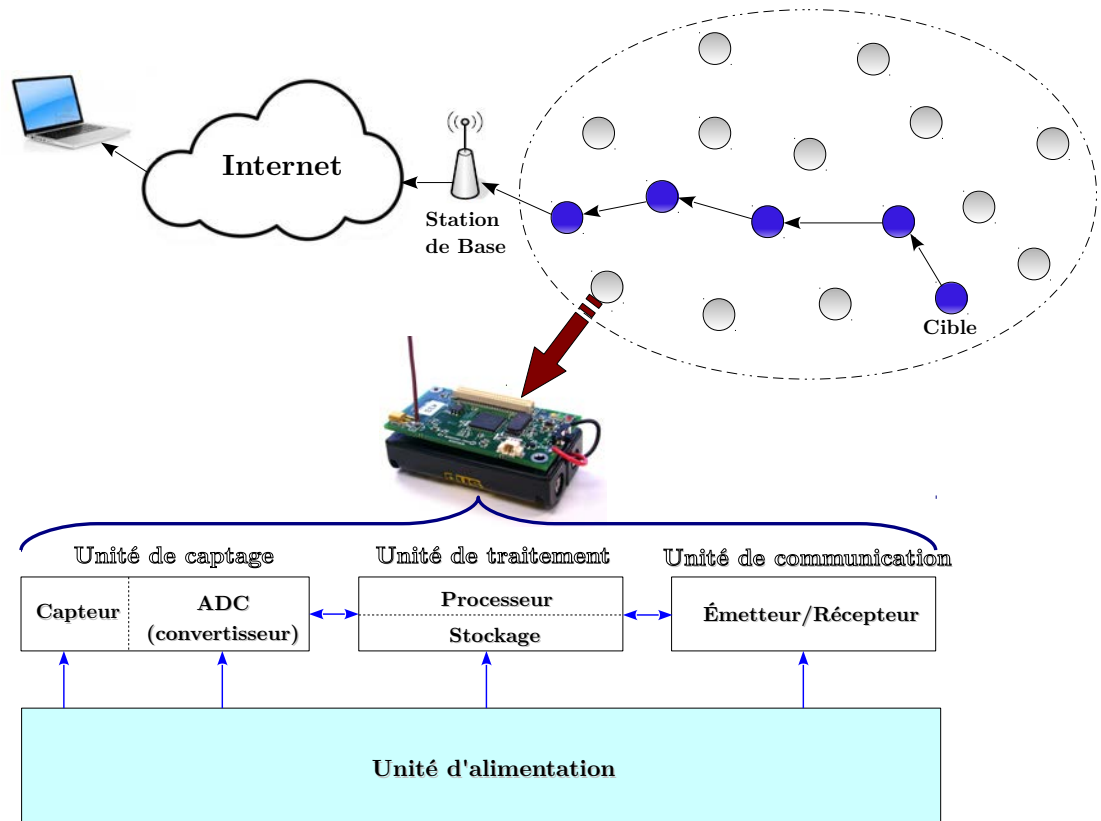


FIGURE 2.1 – Composantes d'un réseau de capteurs sans fil

l'utilisation de la bande passante. Cinquièmement, la conception d'un protocole pour réseau de capteurs peut être étroitement liée à l'application cible. Par exemple, le besoin d'acheminer les informations de surveillance d'un site industriel ou une centrale nucléaire nécessite une latence bornée, qui est différente de celle tolérée par une application de relevé de température classique (surveillance d'un bâtiment).

En raison des différences évoquées précédemment, de nombreux algorithmes ont été proposés dans la littérature pour le routage dans les RCSFs [AY05; PNV13]. Ils prennent en compte à la fois, les caractéristiques matérielles des nœuds et les exigences liées à l'application cible ou à l'architecture du réseau. Rechercher et maintenir une route dans les RCSFs n'est donc pas une tâche aisée, car les contraintes énergétique et de mémoire, couplées à certains événements (défaillance d'un nœud par exemple) entraîne des changements fréquents et imprévisibles dans la topologie. Pour minimiser la consommation énergétique, les protocoles de routage pour RCSF proposés utilisent des mécanismes de routage dits « classiques », auxquels il faut ajouter des techniques spécifiques à ce type d'environnement. Les techniques les plus significatives sont : l'agrégation de données; le *clustering*; l'assignation de rôles spécifiques à certains nœuds; le paradigme de routage centré sur la donnée (*data-centric routing* en anglais). Suivant la façon dont le réseau est organisé, les solutions de routage sont classées comme étant : « à plat », hiérarchiques ou basées sur la position des nœuds. Par ailleurs, selon la méthode d'acheminement des informations ou les traitements réalisés sur celles-ci, on parlera de routage multi-chemins, orienté-QoS (Qualité de Service)

ou à base de requêtes.

2.1.1 Facteurs influençant la conception du routage dans les RCSFs

2.1.1.1 Type de déploiement

La façon dont les nœuds sont déployés dans un réseau est spécifique à l'application. Celle-ci influe sur les performances du routage. Les nœuds peuvent être déployés selon un modèle prédéfini ou de façon aléatoire. Dans un déploiement prédéfini, les nœuds seront manuellement disposés (installation d'alarmes ou dépôt de sondes de relevé de température dans un bâtiment). Les données peuvent alors être routées sur des chemins préétablis. Par contre, dans un déploiement aléatoire, les nœuds sont disséminés au hasard dans l'environnement surveillé (largage à partir d'un hélicoptère par exemple). Ceci résulte très souvent en une distribution non uniforme. Une auto-organisation est alors nécessaire pour la découverte et la construction de la topologie de routage. Du fait des limitations liées à l'énergie et à la bande passante, les chemins inter-nœuds se font alors le plus souvent à travers des liens radio multi-sauts.

2.1.1.2 Consommation énergétique

La principale source de consommation énergétique dans un nœud est la radio. La puissance de transmission étant proportionnelle au carré de la distance séparant la source à la destination, les transmissions *un-saut* ne sont pas toujours possibles : présence d'obstacle, éloignement des nœuds (déploiement aléatoire), ... Pour assurer sa double fonction de *source* et *relayer* d'informations, le nœud doit dépenser son énergie avec soin. De celle-ci dépend la durée de vie du réseau. En cas de défaillance de certains nœuds (principalement due à l'épuisement de la batterie), le routage devrait s'adapter, en effectuant une réorganisation de la topologie suivie du re-routage des paquets par d'autres chemins.

2.1.1.3 Modèle de transmission de données

Le captage et l'envoi des données dans un RCSF sont non seulement dépendants de l'application cible, mais aussi fonction du niveau de criticité de l'information. On distingue alors les modèles de données continus, événementiel, à base de requête ou une combinaison de ceux-ci [TAGH02]. Dans le modèle continu, les nœuds envoient les informations au point de collecte de façon périodique. Ce modèle est approprié pour les applications de surveillance continue d'une grandeur physique, comme un relevé de température ou de luminosité. Dans les modèles à base de requête ou événementiel, la transmission des informations est déclenchée par le *sink* ou lorsqu'un événement se produit : changement soudain ou seuil de valeur dépassé dans les données captées. Ces modèles conviennent pour les applications temps-critique. Une combinaison de tous ces modèles est également possible dans un déploiement.

2.1.1.4 Hétérogénéité et capacité des nœuds

Dans de nombreux travaux et pour des raisons de simplicité, les nœuds du RCSF sont supposés être homogènes : initialement, ils possèdent les mêmes capacités d'énergie et de mémoire, de puissance de calcul et de portée de transmission. Toutefois, certaines applications peuvent nécessiter d'associer des fonctions spécifiques (par exemple capter, relayer, agréger les informations) à certains nœuds. De nombreux protocoles de routage de la littérature sélectionnent une « tête de grappe » (*cluster-head* en anglais) [HCB00]. Ce dernier est alors chargé d'agréger les données de la grappe (*cluster*) avant leur envoi vers la destination finale. Le routage est alors dit hiérarchique, car les transmissions se faisant sur au moins deux niveaux. Cependant, implémenter plus de fonctions sur un nœud comparé à ses pairs, contribue à plus rapidement épuiser son énergie. Des dispositions doivent alors être prévues pour prendre en considération ce fait. Ceci passe notamment par l'introduction de nœuds dotés d'une plus grande capacité énergétique, ayant une plus grande portée de transmission ou offrant une bande passante plus importante.

Par ailleurs, le phénomène observé peut nécessiter de déployer des nœuds embarquant des interfaces de captage différents. On pourrait par exemple déployer une variété de capteurs de température, d'humidité, en même temps que des capteurs de pression ou de détection de présence. Il est alors envisageable que les informations remontées par les nœuds le soient à des fréquences différentes, soumises à diverses qualités de service et selon des modèles de transmission de données différents. Tout ceci impose des défis à surmonter par le routage.

2.1.1.5 Agrégation de données

Lorsque des nœuds génèrent des données redondantes, les paquets issus de ceux-ci peuvent être agrégés de sorte à réduire le nombre de transmissions nécessaires. L'agrégation est la fusion de données provenant de sources différentes en utilisant un opérateur d'agrégation. Il peut s'agir de la suppression des doublons, de la recherche du minimum, du maximum ou de la moyenne des valeurs collectées. Certaines de ces opérations peuvent être réalisées partiellement sur chaque nœud. Bien que celles-ci ne soient pas aussi consommatrices d'énergie que les communications, un gain significatif d'énergie peut être réalisé à travers l'agrégation. Dans certaines architectures, l'opération d'agrégation est dévolue à un nœud plus puissant en processeur, mémoire et portée radio.

2.1.1.6 Tolérance aux pannes

Un certain nombre de nœuds peuvent subir des défaillances matérielles, avoir des problèmes de communication dus aux interférences, ou être hors d'usage du fait de l'épuisement de leur batterie. Cette défaillance de quelques nœuds ne devrait pas compromettre toutes les opérations réseaux. La couche MAC et le routage doivent s'adapter pour trouver de nouveaux liens et chemins d'acheminement de données à la station de base. Ceci peut nécessiter par exemple, une recherche active d'alternative par une signalisation plus importante ou une adaptation de la puissance de transmission. Une redirection des paquets sur des routes alternatives dont les nœuds disposent encore

de suffisamment d'énergie se fera. Un certain niveau de redondance est requis pour qu'un RCSF soit ainsi tolérant aux pannes.

2.1.1.7 Passage à l'échelle

Le nombre de nœuds déployés dans un RCSF peut être de l'ordre de plusieurs centaines ou milliers, voire d'avantage. Tout schéma de routage devrait s'accommoder avec cette grande quantité de nœuds. Si la densité du réseau est élevée, certains nœuds pourront être maintenus à l'état de sommeil, jusqu'à ce qu'un événement particulier déclenche leur réveil. Les nœuds restants assureront alors la fonction d'acquisition et de transport des informations.

2.1.1.8 Dynamique réseau

Si l'on considère les entités constitutives d'une application de RCSF à savoir, les *senders* (nœud d'acquisition et de relais d'information), le *sink* et le phénomène observé, chacune d'elles peut être soit mobile soit statique. Une grande majorité d'architectures réseaux suppose que les *senders* et le *sink* sont tous statiques. Toutefois de plus en plus de déploiements considèrent une mobilité de l'un ou l'autre. Router les informations en provenance ou à destination de nœuds en mouvement est un véritable défi. La stabilité de la route devient alors un paramètre prépondérant, en plus des autres facteurs (énergie, bande passante, mémoire). Le phénomène observé peut lui aussi être dynamique, comme par exemple, une application de détection et de suivi d'une cible. Pour un phénomène statique, le réseau peut être réactif, remontant uniquement le trafic lorsqu'un événement se produit. Un phénomène dynamique quant à lui, nécessite très souvent un envoi régulier d'informations et par conséquent d'avantage de données à acheminer.

2.1.1.9 Qualité de Service

Pour de nombreuses applications, les données doivent être délivrées à la destination dans un délai borné après leur acquisition par le capteur, autrement elles deviendraient inexploitable et sans intérêt. Toutefois la conservation d'énergie, ayant un impact direct sur la durée de vie du réseau, est généralement considérée comme plus importante que la qualité des données à transférer. Dès que l'énergie se fait rare, pour des applications n'ayant pas de contraintes très strictes par rapport aux délais, les nœuds peuvent être amenés à réduire la qualité des informations livrées de sorte à maximiser l'utilisation du réseau.

2.1.2 Classification des protocoles de routage dans les RCSF

Les différents protocoles de routage proposé pour RCSF peuvent se classer selon la structure du réseau sous-jacent ou suivant les opérations dévolues au routage [AY05]. Dans la première catégorie on distingue : le routage à « plat », le routage hiérarchique et le routage basé sur la position des nœuds. La classification à base d'opérations quant à elle, s'attachera à distinguer les protocoles de routage suivant la façon dont les informations sont manipulées ou accédées. On aura alors dans cette dernière : le routage orienté-données, le routage multi-chemins ou le routage basé sur la QoS.

2.1.2.1 Routage à « plat »

Ici, tous les nœuds ont les mêmes rôles et fonctionnalités. L'ensemble collabore à la tâche d'acquisition d'information (captage) et à leur (re)transmissions. Le routage peut être proactif ou réactif. Dans le premier cas, les routes sont construites au préalable et sont disponibles *en permanence* pour la remontée des informations. Dans une approche réactive, les routes ne sont établies qu'à la demande. Aussitôt que les informations sollicitées sont acheminées à la destination, les routes utilisées sont supprimées.

2.1.2.2 Routage hiérarchique

En raison de la densité élevée des capteurs, un routage à plat (un niveau) peu entraîner la surcharge de la station de base ou celui des nœuds proches (phénomène de *hot-spot*). Cette surcharge engendrerait augmentation de latence à cause de la contention au niveau MAC et une consommation excessive d'énergie. Pour y remédier et faire face à la charge de trafic importante sans dégrader la QoS, plusieurs passerelles peuvent être déployées ou le réseau subdivisé en *clusters* ou grappes [HCB00]. Contrairement au routage à « plat », les nœuds ont des fonctions et rôles différents. L'idée principale étant celle de réserver la part de consommation énergétique dévolue aux communications multi-sauts à l'intérieur du *cluster*. On réduit ainsi grandement le nombre de communications en direction du *sink*. Dans chaque grappe, un nœud particulier : le *cluster-head* se chargera d'agréger les informations de la grappe avant de les acheminer vers la station de base. Le *cluster-head* peut alors, en fonction de ses performances, de l'étendue du réseau ou du protocole de routage, communiquer avec la station de base en un saut ou par multi-sauts impliquant les autres *cluster-heads*. La constitution des *clusters* et la sélection des *cluster-heads* contribuent à améliorer le passage à l'échelle, la consommation énergétique des nœuds et la durée de vie globale du réseau.

2.1.2.3 Routage basé sur la position des nœuds

Dans cette approche encore appelée routage géographique, les informations sur la position des nœuds sont exploitées pour router les informations dans le réseau [KK00; YEG01]. L'échange des informations de position entre voisins permet de calculer des coordonnées virtuelles ou réelles et ainsi d'estimer la distance qui les sépare. À cet effet, les nœuds peuvent être équipés d'un dispositif GPS¹, mais celui-ci est très consommateur d'énergie. La plupart des solutions pour RCSF se fondent plutôt sur des moyens plus économes. L'estimation de la puissance du signal reçu (RSSI : Received Signal Strength Indication) en constitue un exemple. Toutefois, un inconvénient majeur de cette technique est qu'elle n'est pas très précise pour estimer la distance entre nœuds [HV12a]. Des solutions hybrides utilisant quelques ancres (nœuds robustes avec GPS) disséminées dans le réseau sont proposées. Les autres nœuds calculent leur coordonnées virtuelles par rapport à ces ancres via des solutions plus efficaces en énergie et acheminent le trafic sur la base des coordonnées calculées.

1. Global Positioning System : système de géolocalisation fonctionnant au niveau mondial et reposant sur l'exploitation de signaux radio émis par des satellites dédiés.

2.1.2.4 Routage orienté-données

Dans ce paradigme, l'adressage diffère de celui des réseaux traditionnels où chaque nœud possède un identifiant unique : l'adresse IP et les routes sont créées entre des pairs, puis gérées au niveau de la couche réseau de la pile OSI. La station de base diffuse plutôt dans le réseau des *intérêts* correspondant à des requêtes, puis attend d'éventuelles réponses [Int+03]. Chaque nœud qui reçoit un *intérêt* le compare aux données dont il dispose et répond en conséquence. Une approche naïve serait pour un nœud, d'inonder son voisinage de la requête lorsqu'il n'est pas en mesure de la satisfaire. Toutefois, cette inondation conduirait à générer des copies multiples de l'information demandée et engendrerait une consommation excessive d'énergie. Des mesures doivent être prises pour surmonter cela. Puisque l'interrogation est faite sur les données plutôt qu'à un nœud spécifique, la définition d'un système de nommage et d'attributs est également nécessaire pour caractériser les informations interrogées par la requête.

2.1.2.5 Routage multi-chemins

Dans l'objectif d'améliorer les performances et la tolérance aux pannes, plutôt que d'utiliser un seul chemin, plusieurs routes sont installées et exploitées en même temps [YBO09]. La robustesse du protocole est alors mesurée comme son aptitude à trouver une route alternative en cas de défaillance de la route principale [Gan+01]. Maintenir en permanence les routes secondaires nécessite l'envoi périodique de messages de contrôle et par conséquent une consommation énergétique supplémentaire pour les nœuds. La fiabilité du routage peut ainsi être améliorée au détriment d'une surcharge liée à la maintenance de plusieurs routes [Dul+03].

2.2 Métriques de routage

Une métrique de routage est une donnée quantitative ou qualitative utilisée pour évaluer le coût d'un chemin. Le « *meilleur chemin* » du point de vue du routage est celui qui satisfait toutes les contraintes (lorsqu'il en existe) et possède le coût le plus faible au regard de la métrique utilisée. Il faut noter que métrique de routage et contrainte ne sont pas exclusives, l'une par rapport à l'autre. Par exemple, on pourrait construire des routes offrant le nombre de sauts minimum, tout en éliminant les liens dont le délai excède une valeur donnée. La valeur de la métrique est influencée par différents facteurs. Elles peuvent être liées à l'environnement ou émaner du réseau lui-même. Les facteurs environnementaux sont ceux qui ne sont pas sujets aux conditions du trafic. On peut citer par exemple, la position du nœud, sa mobilité et l'interférence due aux sources externes au réseau. Les facteurs émanant du réseau sont ceux qui dépendent directement ou indirectement du trafic généré par le réseau. Parmi ceux-ci on retrouve de façon non exhaustive, la congestion, l'interférence interne (au réseau) et l'énergie consommée.

2.2.1 Types de métriques

On peut classer les métriques en fonction des caractéristiques qu'elles exhibent ou de certaines de leurs propriétés mathématiques.

2.2.1.1 Métrique de nœud ou de lien

Cette catégorie permet de différencier les métriques selon l'objet sur lequel est réalisé la mesure. Il peut s'agir du canal de communication entre les nœuds, on parle de métrique de lien ; dans le cas contraire on parle de métrique de nœud. Le nombre de sauts et l'énergie résiduelle sont des exemples de métriques de nœud. Une métrique de lien peut être symétrique si elle a la même valeur dans les deux sens de la liaison ; elle est asymétrique sinon. Formellement, soit $m_{i,j}$ la mesure de la métrique de lien dans le sens allant du nœud i vers le nœud j . Elle est dite symétrique si et seulement si $m_{i,j} = m_{j,i}$. La latence (délai) et le débit (bande passante instantanée) n'ayant pas forcément la même valeur pour les deux sens de la communication, les métriques correspondantes sont asymétriques. L'ETX qui mesure la fiabilité du canal de communication dans les deux sens (que nous présenterons plus en détail dans une section ultérieure) est un bon exemple de métrique symétrique.

2.2.1.2 Dynamique ou Statique

Une métrique est dite dynamique si sa valeur change dans le temps et elle est dite statique dans le cas contraire. Beaucoup d'attention doit être accordée lors de l'utilisation d'une métrique dynamique. En effet, une utilisation naïve peut conduire à des instabilités de routage. En particulier dans les RCSF, en raison de la variabilité de certains paramètres du réseau (qualité du lien radio, intensité du trafic), la valeur d'une métrique dynamique peut changer à une fréquence élevée dans le temps. Pour atténuer l'impact de ces changements sur les décisions de routage, on doit utiliser un algorithme à base de seuils (ou *hystérésis*²) qui détermine à quel moment il est nécessaire d'envoyer les mises à jour aux voisins. Une grande majorité de métriques utilisées dans les réseaux IP sont de ce type. Toutefois il en existe qui sont invariants dans le temps et constituent des exemples de métriques statiques, comme le nombre d'interfaces d'un nœud donné.

2.2.1.3 Métrique unidimensionnelle ou multidimensionnelle

Une métrique multidimensionnelle pour le routage est représentée comme un vecteur de valeurs, chacune de ces valeurs étant décrite par une métrique unidimensionnelle. On la désigne encore sous le terme « *métrique multiple* » car elle peut se décomposer en ses composantes unidimensionnelles. L'utilisation de ce type de métrique permet d'optimiser plus d'un aspect de performance réseau. Il serait par exemple possible de combiner le nombre de sauts et le débit pour construire une métrique multidimensionnelle qui exhibe les propriétés de ses deux métriques de base. Il est nécessaire de définir un opérateur de combinaison qui indique comment les métriques de base sont utilisées pour réaliser la métrique composite. Il peut s'agir d'une évaluation des métriques dans l'ordre lexicographique ou tout autre opérateur d'agrégation dont la mise en œuvre peut s'avérer plus complexe. À titre d'illustration, une combinaison lexicographique ⟨saut-débit⟩ correspond au chemin offrant le meilleur débit parmi les chemins les plus courts (en nombre de sauts), tandis qu'une combinaison ⟨débit-saut⟩ représente une métrique différente. Cette dernière représente le chemin le plus court

2. propriété d'un système qui tend à demeurer dans un certain état quand la cause extérieure qui a produit le changement d'état a cessé.

parmi les chemins les plus larges (en terme de débit). Dans ces deux exemples, c'est toujours la première composante qui est considérée pour évaluer les chemins, la seconde n'étant utilisée qu'en cas d'égalité de valeur dans la composante précédente. Il a été démontré que la recherche du meilleur chemin dans un réseau utilisant une métrique multidimensionnelle est un problème NP-complet [WC96]. De ce fait, l'utilisation d'une métrique multiple s'avère plus complexe.

2.2.2 Considérations liées à l'implémentation

Il existe diverses possibilités d'acquérir les informations nécessaires à la mise en œuvre d'une métrique :

- **Surveillance passive** (*passive monitoring*) du réseau : Les informations requises sont collectées par le nœud, par analyse du trafic entrant ou sortant. En combinant ces informations avec d'autres données ou en appliquant une formule mathématique la métrique peut être dérivée. C'est la méthode utilisée par exemple pour estimer le débit des informations transitant par un nœud.
- **Informations liées au nœud** : Dans cette situation, aucun effort n'est réalisée par le nœud pour obtenir la mesure souhaitée. Une simple lecture de valeurs dans les variables d'environnement ou systèmes est réalisée. C'est le cas par exemple pour le calcul de la taille de la file d'attente des paquets, de l'énergie résiduelle et du nombre d'interfaces. Il faut noter que pour les mesures dynamiques, un processus interne au nœud peut être utilisée pour alimenter la variable lue.
- **Piggy-backing** : La métrique est conçue en insérant des informations additionnelles (obtenues par sondage par exemple) au trafic régulier ou aux paquets de contrôle du protocole de routage. Il convient de noter qu'aucun paquet de contrôle supplémentaire n'est requis pour la mise en œuvre de la métrique. Cette méthode est communément utilisée pour implémenter la mesure du délai de bout-en-bout.
- **Sondage Actif** (*Active Probing*) : Des paquets spécifiques sont envoyés dans le réseau pour mesurer les paramètres de la métrique.

Les informations requises pour l'implémentation d'une métrique devraient être disponibles à la couche réseau du modèle OSI, car c'est à ce niveau qu'a lieu le routage. Toutefois, une grande majorité de travaux de recherche dans les RCSF s'accordent sur la nécessité d'utiliser les approches *cross-layer*³ pour récupérer certains paramètres de la métrique dans les autres couches.

2.2.3 Métriques usuelles pour les RCSF

Pour répondre aux besoins spécifiques des applications pour RCSF, de nombreuses métriques de routage ont été proposées dans la littérature. L'économie de l'énergie

3. possibilité de s'affranchir de l'utilisation stricte des interfaces standardisées de communication entre couches voisines du modèle OSI. Réalisé dans le but d'obtenir un gain (performance, fonctionnalité, optimisation ...) en fournissant une interaction inter-couches riche au delà de l'interface standard.

étant un facteur majeur pour le réseau, la plupart de ces métriques utilisent une approche passive pour obtenir les informations pour leur mise en œuvre.

2.2.3.1 Le nombre de sauts

Cette métrique est très largement utilisée dans les réseaux sans fil en général, en raison de sa simplicité. Chaque lien traversé pour atteindre la destination participe à coût égal d'une unité, sans tenir compte d'autres caractéristiques de la liaison (qualité du lien, taux de transmission, contention, ...). Par conséquent, le protocole de routage recherche le chemin le plus court en nombre de nœuds traversés. Les auteurs de [DCAB02] ont montré expérimentalement que l'utilisation du nombre de sauts comme métrique de routage dans un RCSF peut résulter en de mauvaises performances. En effet, cette métrique a tendance à privilégier des liens entre nœuds éloignés (qui sont en général de piètre qualité en terme de fiabilité) aux liens courts (beaucoup plus robustes). L'utilisation de liens de grande portée peut résulter en une consommation d'énergie plus importante que la concaténation de plusieurs liens courts.

2.2.3.2 L'ETX et métriques apparentées

Partant du constat précédent sur les performances médiocres du nombre de sauts, [DC+05] propose une métrique qui prend en compte la fiabilité du canal de transmission et qui capture les taux de perte dans les deux sens de communication. L'ETX⁴ est le nombre moyen de transmissions nécessaire (incluant les retransmissions), pour qu'un paquet soit envoyé sur la liaison et soit correctement reçu par le voisin [DC+05]. L'intérêt majeur de cette métrique réside dans le fait que, non seulement elle améliore le taux de livraison global des paquets (en sélectionnant les chemins les plus fiables), elle participe également à réduire l'énergie totale consommée (car évitant les liens nécessitant d'avantage de retransmissions). Si $P_{s \rightarrow d}$ représente le taux de transmission avec succès des paquets de s vers d alors l'ETX sur un saut est défini mathématiquement par la relation 2.1.

$$\text{ETX} = \frac{1}{P_{s \rightarrow d} \times P_{d \rightarrow s}} \quad (2.1)$$

Bien que les auteurs n'aient pas indiqué d'opérateur d'agrégation de cette métrique sur le chemin, l'addition est employée dans la plupart des travaux qui en font usage.

Pour l'implémentation initiale présenté dans [DC+05], chaque nœud diffuse toutes les secondes dans le voisinage des annonces de couche de liaison de données. Le voisin incrémente chaque fois le nombre d'annonces reçues et calcule le taux de perte au bout de 10 secondes, qu'il retourne à l'émetteur. Cette méthode peut engendrer une surcharge importante. Une alternative consiste à effectuer une mesure passive par *piggybacking* des numéros de segments dans les paquets de données. Toutefois, cette approche n'estime l'ETX que sur les liens déjà promus par le routage. Celui des liens vers des voisins potentiels (non encore sélectionnés comme saut suivant) n'est pas évalué.

Pour s'accommoder aux technologies radio et à divers modèles de réseaux sans fil (Réseau Ad-Hoc, Réseau *Mesh*, RCSF, etc.) de nombreuses variantes de l'ETX sont proposées. En l'occurrence, l'ETT (Expected Transmission Time) est une extension de l'ETX

4. littéralement de l'anglais : Expected Tx [i.e. Transmission] Count

prenant en considération le débit de transmission et la taille des données [DPZ04]. Elle correspond au temps moyen nécessaire pour transmettre correctement un paquet à un voisin donné. Elle est déterminée par la formule : $ETT = ETX \times \frac{T}{D}$, où T est la taille moyenne des paquets et D le débit de transmission.

De nombreuses technologies sans fil offrent plusieurs canaux qui ne se chevauchent pas. Les auteurs de [DPZ04] propose d'exploiter cette possibilité dans une métrique basée sur l'ETX et apparentée à la précédente : le WCETT (Weighted Cumulative Expected Transmission Time). Cette métrique prend en compte non seulement les liens n'utilisant pas les mêmes canaux, mais limite également l'interférence intra-flux⁵. Cependant, cette métrique présente un inconvénient majeur : elle n'est pas isotonique [YWK05] (nous examinerons cette propriété ultérieurement au §4.2.3). Le risque de boucle de routage lors de la recherche de plus court chemin est élevé.

2.2.3.3 Le délai

Bien que cette métrique ne soit pas très répandue dans les RCSFs, elle s'avère essentielle pour les applications temps-réels ou sensibles aux retards. Contrairement aux métriques précédentes, elle prend en considération l'effet de la contention (*backoff*) due à la couche MAC, de même que les retards engendrés par la réémission des trames en cas de collision. Plusieurs éléments participent au délai encouru par un paquet qui transite par un nœud spécifique. Soit D ce délai, on a : $D = D_{trait} + D_{file} + D_{trans} + D_{prop}$, où D_{trait} est le délai mis par le nœud pour traiter le paquet et D_{file} le délai mis par le paquet en file de sortie, en attente de la disponibilité du support et la transmission des paquets qui le précèdent. D_{trans} est le délai de transmission, il dépend uniquement de la bande passante et la taille du paquet. Enfin, D_{prop} est le délai de propagation (il est très faible, en raison de la vitesse élevée de la propagation du signal dans l'air). Parmi tous ces éléments le seul qui soit variable pour un paquet donné est D_{file} . En effet, la durée passée par un paquet dans la file d'attente dépend de l'état du trafic dans le réseau. La métrique ETT évoquée plus haut ne prend en compte que la seule composante D_{trans} . Elle ne change malheureusement pas pour des paquets de même taille et ne reflète donc pas la réalité des conditions du trafic.

Le délai est sujet à beaucoup de variations. Pour cette raison, la plupart des protocoles qui s'en servent n'utilisent pas la valeur mesurée à l'instant, mais plutôt une estimation de délai calculé comme une moyenne pondérée, glissante dans le temps.

Lorsque le délai est calculé de façon unidirectionnelle (de l'émetteur vers le récepteur), il est nécessaire que les horloges internes des nœuds soient synchronisées. Dans l'environnement distribué des RCSF, la mise en place d'horloges synchrones engendre une complexité d'implémentation importante. L'implémentation communément réalisée est alors le délai aller-retour, plus connu sous son anglicisme RTT (*Round-Trip Time*). Les données transportent une estampille horaire lors de leur envoi au prochain saut.

5. L'interférence intra-flux se produit lorsque les radios de plusieurs liens appartenant au même flux ou même chemin de données opèrent sur le même canal et sont en compétition pour l'accès au médium. Il faut distinguer celle-ci de l'interférence inter-flux qui est l'interférence causée par d'autres flux, mais opérant sur le même canal radio.

Une fois reçu, le voisin retourne un acquittement contenant l'estampille initiale. L'initiateur peut ainsi évaluer le délai aller-retour mis par le paquet pour un seul saut (*i.e.* vers un nœud voisin). Les délais *un-saut* sont ensuite concaténés sur le chemin pour obtenir le délai de bout-en-bout.

2.2.3.4 La taille de file d'attente des paquets

Les nœuds peuvent recevoir plus de paquets qu'ils ne sont capables de traiter. De plus, ils peuvent concourir pour l'accès au support lors de la transmission des données. En conséquence, ils possèdent chacun un tampon de réception (resp. de transmission) pour conserver les données entrantes (resp. sortantes) en attendant leur traitement (resp. leur transmission). Ces tampons fonctionnent par défaut selon le modèle de file d'attente : « premier entré, premier servi ». La taille de ces files constitue un bon indicateur de la charge du trafic dans le réseau et permet d'anticiper la congestion en évitant les goulots d'étranglement réseau. Dès que le tampon de réception ou de transmission est plein, le nœud n'est plus capable de conserver de nouvelles données. L'excédent des paquets reçus est alors abandonné, ceci conduit à une perte de données pouvant impacter sévèrement sur les performances de l'application.

2.2.3.5 L'énergie

De nombreuses propositions en vue d'optimiser la consommation énergétique dans les RCSFs ont été faites dans la littérature. Elles passent par : la conception de couches MAC efficaces en énergie [Dun11], des solutions de routage prenant en compte la consommation énergétique des nœuds par estimation de l'intensité du trafic [PNV13] ou par construction d'une métrique de routage basée sur l'énergie [YBO09; Yoo+10; Kam+13], le contrôle de la puissance de transmission [Ana+09] et le *clustering* [HCB00]. D'autres travaux optent plutôt pour la possibilité de recharger les batteries à partir de l'énergie récoltée de l'environnement [Kau+14; SZ16].

L'énergie nécessaire pour transmettre k bits à un nœud voisin à une distance d de l'émetteur est donnée par la relation 2.2 et celle consommée pour la réception de k bits à la même distance, donnée par 2.3 [HCB00].

$$E_{Tx}(k, d) = E_{elec} * k + \epsilon_{amp} * k * d^2 \quad (2.2)$$

$$E_{Rx}(k, d) = E_{elec} * k \quad (2.3)$$

où E_{elec} est l'énergie dissipée par les composants électroniques pour la transmission ou la réception d'un bit, et ϵ_{amp} l'énergie consommée par l'amplificateur pour que le signal transmis soit de qualité acceptable.

La façon dont une métrique fondée sur l'énergie est conçue dépend de l'objectif poursuivi par le routage. Deux approches sont généralement adoptées :

- minimiser la consommation globale du réseau,
- maximiser la date à laquelle l'un des nœuds aura complètement épuisé sa batterie.

Dans la première approche, l'on comptabilise l'énergie consommée par les différents flux de données envoyés dans le réseau [SWR98]. Soit $e_{i,j}$, l'énergie nécessaire pour envoyer un paquet du nœud i et le recevoir au nœud j (obtenu comme indiqué par les relations 2.2 et 2.3). L'énergie totale requise pour envoyer un paquet de la source s vers la destination d est alors :

$$E = \sum_{i \in [s \dots d]} e_{i,i+1} \quad (2.4)$$

$[s \dots d]$ étant l'ensemble des nœuds sur le chemin allant de s à d . Le principal inconvénient de cette approche est qu'elle ne considère pas l'énergie disponible sur le nœud et embarquée dans la batterie. Bien que celle-ci maximise l'énergie totale due au trafic, des goulots d'étranglements énergétiques peuvent apparaître dans le réseau, certains nœuds consommant plus rapidement leur énergie que d'autres.

Pour éviter ces goulots d'étranglements énergétiques, la seconde approche recherche plutôt à équilibrer la consommation énergétique de tous les nœuds [Yoo+10; Kam+13]. Elle se fonde sur l'hypothèse suivante : « si les énergies des nœuds sont distribuées le plus équitablement possible, ils épuiseront leur batterie à peu près dans la même période ». Pour atteindre cet objectif, il est possible de considérer l'énergie résiduelle des nœuds pendant le routage, *i.e* celle réellement contenue dans leur batterie. Nous reviendrons en détail sur cette approche dans un chapitre ultérieur.

2.2.3.6 Le RSSI

A la différence de l'ETX qui permet d'estimer les liens les plus robustes au niveau des couches MAC et Réseau, le RSSI (*Received Signal Strength Indicator*) est une métrique exploitant les informations de la couche physique pour sélectionner les « bons » liens (meilleure qualité du signal reçu) [HV12a; LXC14]. Toutefois, cette métrique n'est pas très précise et présente des *zones grises* pour lesquelles le taux de réception de paquets à la destination pour des valeurs similaires de RSSI varie énormément [SL06]. Pour cette raison, la métrique LQI (*Link Quality Indicator*) est plus souvent utilisée, en particulier avec des radios de type CC2420 où elle est beaucoup plus précise que le RSSI et permet d'obtenir un meilleur taux de réception de paquet.

2.3 Standardisation

La large gamme d'applications qu'offrent les RCSF a suscité un fort intérêt de la communauté scientifique et industrielle autour des technologies qui s'y rapportent. Toutefois, les solutions proposées ont tardé à être transférées du milieu scientifique vers le monde industriel. De nombreuses implémentations des protocoles proposés dépendaient de chaque constructeur de matériels, rendant incompatibles les systèmes conçus. La nécessité d'obtenir des systèmes interopérables s'est imposée aux différents acteurs et a conduit à des efforts de standardisation de protocoles pour RCSF.

2.3.1 Standards de l'IETF relatifs aux RCSF

Riche de sa longue expérience en matière de protocoles et technologies standardisées pour l'Internet, l'IETF a mis en place des nombreux groupes de travail relatifs à la standardisation des protocoles pour RCSF, parmi lesquels les plus significations sont listés ci-dessous. Leurs travaux prend en compte la nécessité de rendre compatible les technologies destinées aux RCSF compatibles à IP.

- **ROLL** (Routing Over Low-power and Lossy Network) : ce groupe de travail est chargé de proposer une solution de routage adaptée aux réseaux de type LLN (Low-power and Lossy Network); *i.e* ceux où les communications sont instables et les nœuds équipés d'une interface radio à faible puissance d'émission. Ce groupe de travail a publié en 2012 RPL [Win+12], comme résultat de son cahier des charges initial. De nombreux autres standards constituant des documents d'accompagnement de ce protocole ont été également publiés, ou sont en cours de publication. La figure 2.2 regroupe les principaux standards relatifs au protocole RPL.

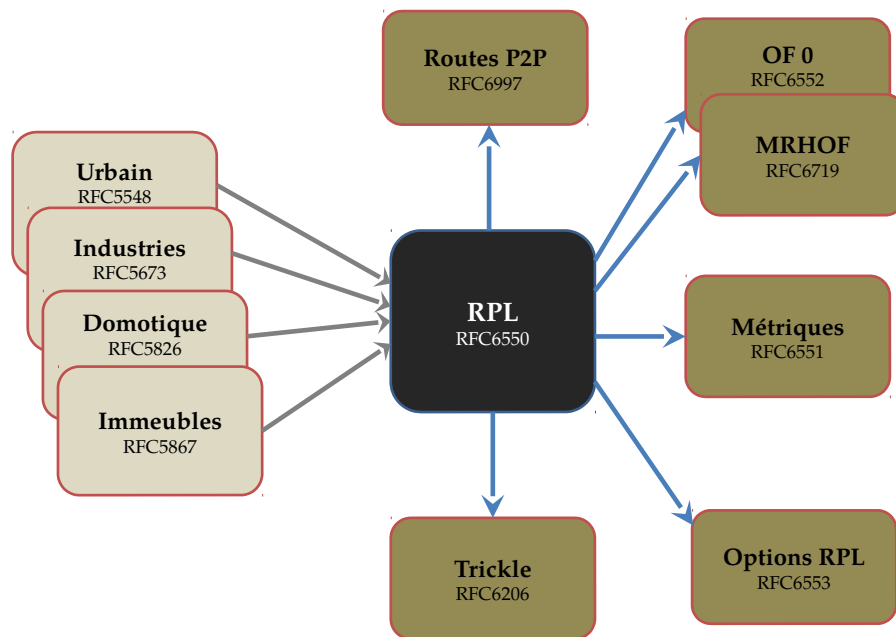


FIGURE 2.2 – Standards relatifs au protocole RPL

- **6LoWPAN** (IPv6 over Low-Power Wireless Personal Area Networks) : Pour permettre aux paquets IPv6 (1280 octets) d'être transportés sur le RCSF, dont la MTU (Maximum Transfer Unit) au niveau physique est très limitée (127 octets pour la couche PHY IEEE802.15.4), des adaptations sont nécessaires. Le groupe de travail 6LoWPAN est chargé de définir les mécanismes de compression à réaliser pour permettre ce transport. Cela rend possible la connexion des réseaux IPv6 standards aux réseaux contraints en ressources que constituent les RCSF. Ils fondent leurs travaux sur le document de travail RFC4919

[KMS07], définissant les hypothèses, problèmes et objectifs poursuivis. Le principal standard publié par ce groupe est désigné sous le nom « couche d'adaptation 6LoWPAN »[Mon+07; HT11].

- **CoRE** (Constrained RESTful Environments) : Il a pour objectif de fournir un cadre pour les applications destinées à être déployées dans les réseaux IP contraints en ressources matérielles. Les travaux de ce groupe ont conduit à la publication du protocole CoAP [SHB14]. Le cadre applicatif est défini selon l'architecture REST⁶. CoAP rend possible l'accès aux informations entre clients et serveur dans un réseau IP contraint.

Ces différents groupes collaborent entre eux. La contribution des universités, instituts de recherche et industriels dans ceux-ci est significative. Il est important de noter que les standards proposés sont tous orientés IPv6. En effet, l'espace d'adressage IPv4 étant très limité et compte tenu du nombre potentiellement élevé de nœuds dans le réseau, la précédente version d'IP n'est plus envisageable. La figure 2.3 présente la pile des protocoles standardisés pour les RCSF.

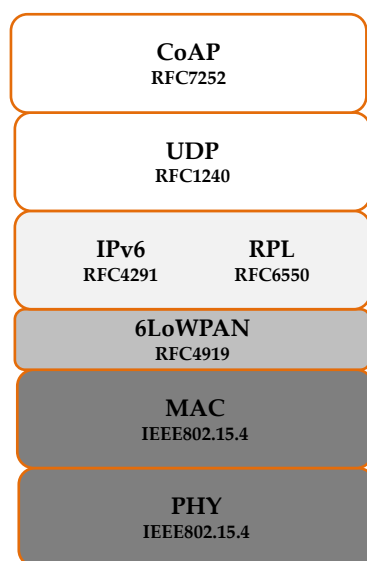


FIGURE 2.3 – Pile protocolaire standardisée pour les RCSF

2.3.2 Le protocole de routage RPL

RPL (littéralement IPv6 Routing Protocol for LLN) [Win+12] est un protocole à vecteur de distance proactif, conçu pour les besoins de RCSF fonctionnant sous IPv6. Il a

6. Representational State Transfer (ou RESTful) : architecture pour les systèmes hypermédia (texte, images, sons, vidéos, etc.) distribués assurant l'interopérabilité entre des machines sur Internet par l'utilisation des services Web. Elle est caractérisée par le modèle de communication client-serveur et des transactions sans état sur des ressources individuelles entre le client et le serveur.

été spécifiquement pensé pour optimiser le trafic de type multipoint à point (*convergecast*). Toutefois, il fournit des mécanismes pour prendre en compte le trafic point à multipoint ainsi que le trafic point à point. Le groupe de travail `ROLL` a commencé par publier l'ensemble des exigences souhaitées pour le protocole relatives aux classes de réseaux et d'applications supportés. Il s'agit d'applications pour la domotique [BBP10], tel que le contrôle de l'éclairage, des stores et volets, la surveillance des données de température ou d'alarmes. Les besoins pour la gestion de grands bâtiments comme les administrations (universités, hôpitaux, bureaux, etc.) sont spécifiés dans le document RFC5867 [Mar+10]. Ceux relatifs à l'automatisation des processus industriels à travers des solutions connectées sans fils sont également spécifiés [Pis+09]. Il faut noter ici que, les contraintes de délai sont beaucoup plus prononcées dans ce type d'environnement. Le dernier champ d'utilisation est celui des applications urbaines participant à l'émergence du concept de ville intelligente (système de distribution intelligent d'électricité, feux de signalisation et gestion du transport, surveillance environnementale, etc.) [Doh+09]. Le routage a été conçu pour permettre à plusieurs applications de cohabiter simultanément dans le même réseau physique grâce à la notion d'**instance RPL**. Ainsi chaque instance optimise une (ou plusieurs) métrique(s) ou satisfait à des contraintes spécifiques liées à l'application.

2.3.2.1 Formation de la topologie

Étant donné un ensemble de nœuds déployés (de façon planifiée ou aléatoire), le routage organise le réseau comme un (ou plusieurs) arbre(s) orienté(s) (**DODAG** : Destination-Oriented Directed Acyclic Graph), chacun étant enraciné en un point appelé racine (DODAG root). Un déploiement RPL dans le réseau peut comporter plusieurs instances, chacune étant constituée d'un ou plusieurs DODAG. Bien qu'il soit possible pour un nœud d'appartenir à plusieurs instances à la fois, il ne peut toutefois participer qu'à un seul DODAG par instance. Pour des raisons de simplicité, nous considérerons par la suite que nous n'avons qu'un seul DODAG, soit une seule racine RPL dans la topologie. Cette dernière peut alors servir de passerelle vers l'Internet classique. La figure 2.4 présente la topologie dont nous nous servirons pour les illustrations. Les tirets représentent la possibilité de communication sans fil entre les nœuds.

Initialement (figure 2.4A), seule la racine fait partie intégrante de la topologie active RPL. Elle envoie périodiquement dans son voisinage des informations de configuration embarquées dans des messages de contrôle RPL spécifiques appelés DIO (DODAG Information Objet), comme illustré sur la figure 2.4B. Les nœuds qui reçoivent ces informations peuvent rejoindre le réseau en choisissant leur prochain saut (parent RPL) vers la racine. Dès qu'il fait partie intégrante de la topologie RPL, le nœud commence par envoyer ses propres DIO (cf. figure 2.4C). Lorsqu'un nœud reçoit plusieurs DIO consistants émanant de voisins différents, un choix doit être opéré pour sélectionner parmi ceux-ci, celui offrant le meilleur coût au regard de l'objectif recherché et des contraintes imposées par l'application. C'est par exemple le cas pour le nœud central dans la topologie finale (figure 2.4D) qui opte pour le parent de droite au détriment de celui de gauche. La fonction d'objectif (OF) est responsable de l'indication sur les mécanismes de ce choix. Il est important de noter que tous les DIO reçus par un nœud,

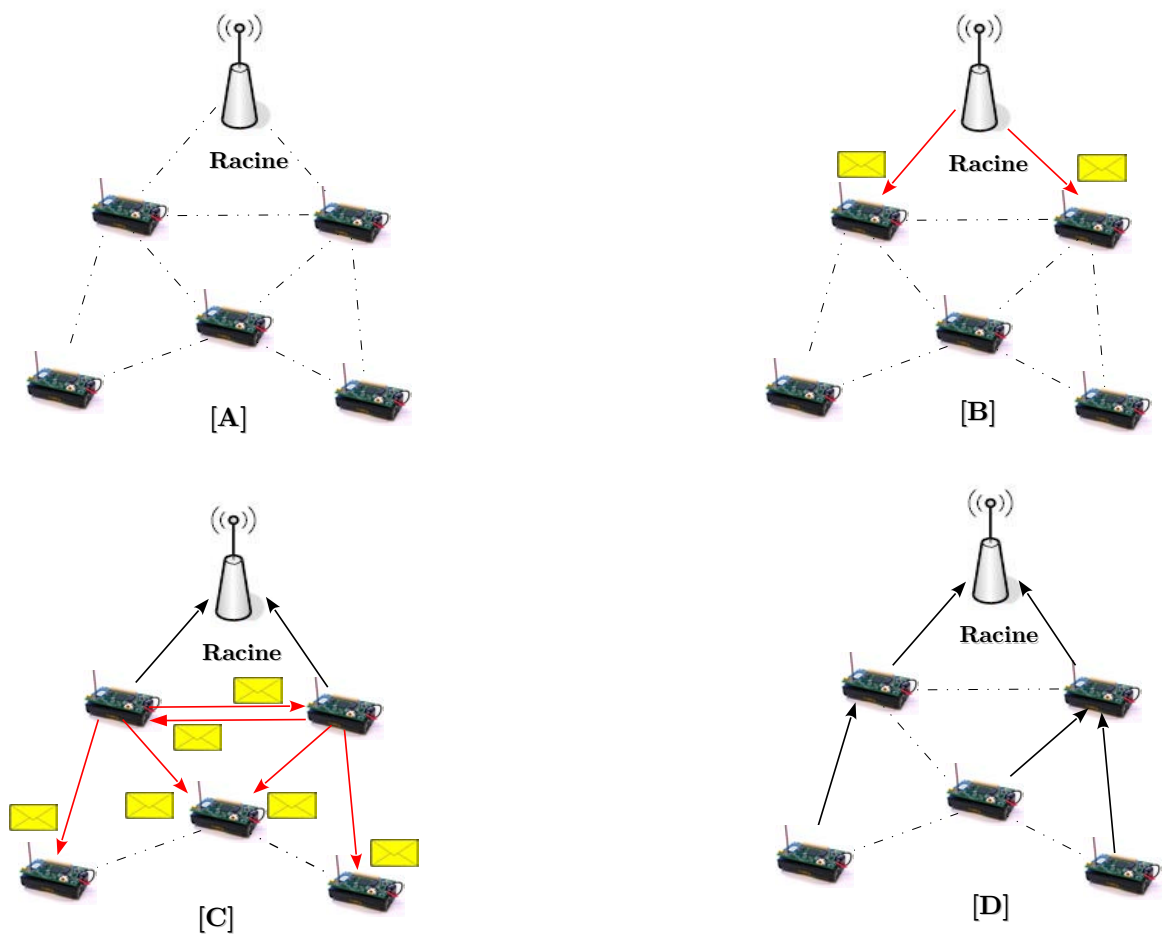


FIGURE 2.4 – Construction du DODAG

ne sont pas toujours valides. En particulier, si celui-ci provient d'un voisin situé dans un niveau inférieur de l'arbre topologique, choisir ce dernier conduirait à une boucle de routage. La notion de rang est alors utilisée pour déterminer quels sont les **DIO** à considérer dans la recherche d'un prochain saut.

2.3.2.2 Messages de contrôle RPL

Pour mettre en place et maintenir la topologie de routage, quatre nouveaux messages ICMPv6 [CDG06] ont été définis pour le protocole RPL :

- **DIO** (**D**ODAG **I**nformation **O**bject) : Ce message est utilisé pour la création des routes ascendantes. Il se propage dans le réseau en sens inverse, et transporte de nombreux paramètres nécessaires à la mise en place de la topologie. Ces paramètres sont fournis uniquement par la racine RPL, les autres nœuds ne servant que de relais. La dissémination des **DIO** se fait par défaut en multidiffusion, mais en cas de sollicitation spécifique d'un nœud, la monodiffusion est utilisée.
- **DIS** (**D**ODAG **I**nformation **S**ollicitation) : Elle est utilisée par un nœud qui souhaite rejoindre la topologie (envoi en multidiffusion) ou réclamant des informations de configuration plus récentes (envoi en monodiffusion). Dans tous les

cas, un nœud recevant un `DIS` répond à l'initiateur par une monodiffusion de paquet `DIO`.

- **DAO** (`DODAG Advertisement Object`) : Ce message est utilisé uniquement en cas de nécessité de disposer des routes descendantes (pour le trafic point à point par exemple). Autrement il peut être désactivé pour économiser les ressources (mémoire et bande passante). Il est envoyé à la racine `RPL` quand le mode de fonctionnement est *non conservation* (non-storing mode) ou au(x) parent(s) en mode *conservation* (storing mode). Contrairement aux autres messages de contrôle `RPL`, il est le seul toujours envoyé en monodiffusion et nécessite un acquittement du destinataire.
- **DAO-Ack** (`DAO-Acknowledgment`) : Il est utilisé par la destination en réponse à un message `DAO` reçu pour acquitter ce dernier. En cas de non réception du `DAO-Ack` par la source, celle-ci peut réémettre le `DAO` initial.

Le `DIO` est prévu pour transporter de nombreuses options, toutefois certaines informations sont obligatoires. Parmi celles-ci on a : `RPLInstanceID` qui identifie de façon unique l'instance `RPL` dans le réseau, le `DODAGID` qui caractérise la racine (très souvent l'une de ses adresses IPv6), le numéro de version du `DODAG` permettant de savoir si les informations reçues sont « fraîches » ou obsolètes et le rang du nœud émetteur dans le `DODAG` qui renseigne sur sa position dans l'arbre construit sur le réseau. Le `DAO` quant à lui transporte les préfixes réseaux permettant de rendre accessible par monodiffusion les nœuds situés en profondeur de l'arbre.

2.3.2.3 Maintenance de la topologie

Le routage étant proactif, la cohérence des informations de routage doit être assurée en permanence. Pour atténuer la consommation énergétique inhérente aux messages `DIO` envoyés, le routage utilise l'algorithme du `Trickle` [Lev+11]. L'idée de base de cet algorithme est d'adapter dynamiquement la fréquence d'envoi des `DIO` en fonction de la qualité des informations à transmettre : plus la topologie est stable, moins il est nécessaire d'envoyer les informations de contrôle. Initialement, la fréquence d'envoi des `DIO` est prise dans un intervalle minimum, défini par le paramètre de configuration I_{min} . Cet intervalle double régulièrement à l'expiration du minuteur d'envoi, jusqu'à une valeur maximale : I_{max} à laquelle elle se stabilise. Dès qu'une incohérence est constatée ou qu'une nouvelle information est disponible, le minuteur de l'algorithme est réinitialisé. Ceci entraîne des envois plus fréquents et par conséquent, une propagation plus rapide de l'information souhaitée dans le réseau. Le réseau est dans un état stable (*i.e* consistant) lorsque tous les nœuds ont obtenu les mêmes paramètres de configuration et que ceux-ci sont cohérents (*i.e.* non contradictoires). Toutefois, la nature non fiable des transmissions et les pertes éventuelles de paquets peut conduire à un état d'incohérence (faisant suite à un précédent état stable).

Une situation pratique de résolution d'incohérence se produit, lorsqu'un nœud reçoit d'un voisin des informations obsolètes (numéro de version d'instance antérieur). Il doit alors réinitialiser son minuteur `Trickle` pour diffuser rapidement l'information « fraîche » dont il dispose. Les inconsistances sont ainsi résolues localement, par interaction entre nœuds voisins. Elles se propagent de proche en proche pour couvrir l'ensemble du réseau.

2.3.2.4 Rang, détection et gestion des boucles de routage RPL

Pour éviter la formation de boucles de routage, RPL qui est un routage à vecteur de distance, utilise un gradient. Ce dernier indique pour chaque nœud, sa position relative vis-à-vis de la racine comparée aux nœuds du voisinage. Il s'agit d'une grandeur scalaire encore appelée rang dans le standard. La façon dont le rang est calculé par la fonction d'objectif n'est pas précisée, mais il doit exhiber des propriétés génériques quelle que soit la fonction d'objectif utilisée. Bien que sa valeur soit dérivée de la métrique, il s'en différencie : il doit être monotone (en particulier décroître) lorsqu'on progresse vers la racine, mais ne doit pas nécessairement changer aussi rapidement que la métrique utilisée.

Malgré l'utilisation du rang pour sélectionner comme parent les nœuds plus proches de la racine (*i.e.* ayant un rang inférieur à celui du nœud considéré), des boucles peuvent apparaître. Ceci est dû à la nature non fiable du support (perte d'un ou plusieurs DIO annonçant des métriques de moins bonne qualité) et des inconsistances évoquées précédemment. RPL offre un mécanisme pour détecter l'existence d'une boucle pendant la transmission des données. Une nouvelle option [HV12b] est incorporée dans les paquets IPv6 qui transitent par un réseau utilisant RPL. Elle prévoit un bit qui peut être positionné pour indiquer le sens de progression du paquet (0 $\rightarrow$ vers-le-bas et 1 $\rightarrow$ vers-le-haut). Si un paquet est reçu d'un nœud fils avec un bit marqué 0, le nœud parent en déduit l'existence d'une boucle (inconsistance). Il détruit le paquet et prend les mesures nécessaires pour éliminer la boucle. Ce bit de sens de progression du paquet devrait être utilisé pour toutes les données transitant par un réseau RPL.

Deux mécanismes de réparation sont prévus : la réparation locale et la réparation globale. Dès qu'une boucle est détectée, le nœud concerné déclenche une réparation locale. Celle-ci fonctionne suivant le principe d'empoisonnement de la route. Le nœud se détache de son DODAG et annonce un rang de valeur *infinie*, ayant pour conséquence de l'invalidiser dans la liste des parents potentiels de tous ses fils (*sous-DODAG*). Après s'être détaché du DODAG, le nœud s'y attache à nouveau dès qu'il a trouvé un parent alternatif. Il recommence alors à annoncer un rang de valeur *non-infinie* déterminée à partir de celle de son nouveau parent.

Contrairement à la réparation locale qui n'a d'incidence que sur le voisinage immédiat du nœud l'ayant déclenché (incluant son *sous-DODAG*), la réparation globale impacte l'ensemble du DODAG. Elle est toujours déclenchée par la racine RPL et implique une reconstruction complète du graphe RPL. Une implémentation simple de celle-ci consiste pour la racine, à annoncer dans les DIO un numéro de version supérieur à la valeur courante.

2.3.2.5 Fonction d'objectif RPL

La fonction d'objectif permet de spécifier comment les métriques de routage sont transformées en rang. Elle est également responsable de la définition de la manière dont le nœud sélectionne le meilleur parent de la liste de ses parents potentiels. L'IETF n'a défini que deux fonctions d'objectifs RPL :

- **OFO** (Objective Function 0) : Elle implémente le nombre de sauts comme métrique, tous les liens participent avec le même poids pour la sélection du chemin [Thu12].
- **MRHOF** (Minimum Rank Hysteresis Objective Function) : Elle est implémentée pour fonctionner avec les métriques additives [GL12]. Dans sa définition, l'IETF utilise l'ETX comme métrique à optimiser, mais toute autre métrique additive (nombre de sauts, délai) pourrait être également utilisée. MRHOF se sert d'une hystérésis pour limiter les variations liées à l'emploi d'une métrique dynamique.

Dans les documents d'accompagnement du standard [Vas+12], plusieurs métriques sont prévues pour fonctionner avec le protocole, cependant la liberté est donnée au concepteur de définir comment les utiliser. Par ailleurs, aucune recommandation n'est faite sur la façon de les combiner pour prendre en considération la qualité de service (QoS) ou de les utiliser selon les spécificités liées à l'application.

Chapitre 3

Du Réseau de Capteurs à l'Internet des Objets

3.1 Technologies de l'Internet des Objets

Le terme Internet des Objets (de l'anglais Internet of Things : IOT) a été pour la première fois utilisé par K. Ashton en 1999, co-fondateur de Auto-ID Center au MIT dans l'une de ses présentations [Ash09]. Il y décrit sa vision de la façon dont les objets et personnes du monde physique peuvent être recensés et gérés informatiquement grâce à la technologie RFID (Radio-Frequency IDentification). Depuis lors, ce concept a beaucoup évolué grâce aux énormes progrès effectués dans divers domaines technologiques : allant de celui des microcontrôleurs, de la nanotechnologie, des capteurs et actionneurs à celui des technologies sans fils. Il fait aujourd'hui référence à la possibilité de connecter tout objet du quotidien à l'Internet traditionnel [MF10]. Les objets physiques ne sont plus déconnectés du monde virtuel, mais peuvent être contrôlés à distance et agir comme des points d'accès à différents services Internet. Dans ce contexte, les « objets » sont vus comme des entités dotées de capacités de calcul et de communication, donc possédant un degré d'intelligence et capables d'interagir avec d'autres objets, eux mêmes connectés au réseau mondial de données. En effet depuis quelques décennies, les progrès de l'informatique embarquée ont permis de pouvoir *empaqueter* : unité de traitement (CPU), interface de communication et de capture, unité d'alimentation, ... dans un volume de l'ordre du centimètre cube, voir du millimètre cube, créant ainsi de minuscules dispositifs informatiques intelligents. Ces équipements de par leur taille, peuvent être intégrés dans divers objets usuels (lunettes, porte-clés, barquettes alimentaires et jouets), transportés par des personnes (captage des paramètres physiques et surveillance des personnes vulnérables) ou incorporés dans l'environnement (habitat, forêt, centrale nucléaire et chaussée). Le réseau sans fil ainsi formé [*monde virtuel*] rend possible l'observation, le suivi et le contrôle (actionneur) de l'environnement physique [*monde réel*].

Cisco dans son livre blanc relatif à l' IOT illustre l'évolution du nombre d'objets connectés comme indiqué par la figure 3.1 [Eva11]. Aujourd'hui, celui-ci dépasse de loin le nombre d'habitants sur la planète et il devrait continuer d'augmenter pour atteindre les 50 milliards d'ici 2020. Chacun de ces équipements connectés à l'Internet nécessitera une unique adresse logique. L'adressage IPv4 possède 32 bits et ne dispose que de 4.3 milliards d'adresses potentielles. Il n'est donc plus envisageable comme solution d'adressage, IPv6 s'impose alors comme seule alternative pour l' IOT . Cette

version du protocole est utilisée pour fournir une large gamme de services et d'applications et servir de support à la *convergence numérique* illustré à la figure 3.2.

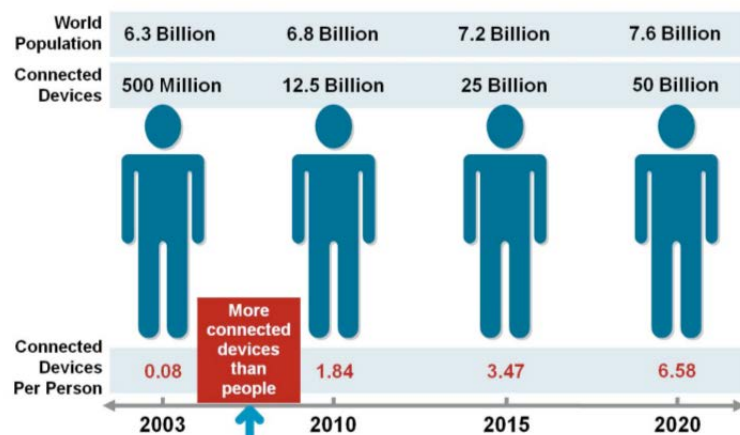


FIGURE 3.1 – Évolution du nombre d'objets connectés de l'IoT [Eva11].



FIGURE 3.2 – Convergence numérique [VF14].

Le RCSF est un élément constitutif essentiel vers la mise en œuvre de cet Internet des Objets. La combinaison de toutes ces innovations technologiques, à la fois dans le domaine du développement logiciel, du matériel embarqué, couplée à l'intégration des technologies de l'Internet et la standardisation des protocoles pour RCSF ont permis une large adoption de ce nouveau paradigme. Toutefois, de nombreux défis sont soulevés :

- Mise en réseau et interconnexion des « objets » à l'Internet, du fait de l'hétérogénéité du matériel et des logiciels impliqués,

- Flexibilité et passage à l'échelle des applications, services et équipements,
- Automatisation des procédures et des configurations,
- Manipulation des grandes masses de données,
- Sécurisation des services et des données personnelles collectées.

L'Internet des Objets n'est pas constitué d'une unique technologie (cf. figure 3.2), mais utilise diverses technologies nouvelles de communications, matérielles et logicielles mises ensemble dans une plate-forme de communication globale basée sur IP.

3.1.1 La Radio Identification (RFID)

Le plus souvent désigné par son acronyme anglais RFID (Radio Frequency Identification), il s'agit d'une technologie permettant de mémoriser et de récupérer à distance des données. Elle revêt une importance capitale pour l'Internet des Objets car elle fut à l'origine de ce paradigme. L'objectif initial étant celui d'identifier et de réaliser un suivi minutieux des biens dans une chaîne d'approvisionnement ou dans le domaine de la logistique. Cette technologie est utilisée aujourd'hui dans beaucoup d'autres domaines. Un système d'identification par radio-fréquence est constitué de trois éléments :

- **Une Radio-étiquette (RFID tag) :** c'est un circuit intégré mémorisant l'information sur l'objet auquel la puce est incorporée. Il est muni d'une antenne pour la réception/transmission des signaux. En général, il s'agit d'un dispositif passif, c'est à dire qui ne nécessite aucune source d'énergie extérieure que celle fournie par le lecteur au moment de l'interrogation. Il faut noter que de nos jours, des versions actives existent. Elles embarquent une petite batterie leur permettant d'émettre des signaux.
- **Un lecteur :** utilisé pour envoyer le signal radio à la puce RFID et capturer la réponse de cette dernière. Le système opère dans la bande de fréquence non licenciée ISM¹ (125 KHz en basse fréquence, 13.56 Mhz en haute fréquence, 868 – 928 Mhz en ultra-haute fréquence et 2.45/5.8 Ghz dans la bande des micro-ondes).
- **Un intergiciel :** il reçoit et traite les informations reçues du lecteur.

Un système RFID assurera deux fonctions basiques pour l'Internet des Objets : l'identification et la communication. Son principe de fonctionnement général est le suivant. Le lecteur initie la communication en diffusant une requête. Les radio-étiquettes du voisinage répondent à ce dernier en fournissant leur identifiant et leurs données stockées. Pour gérer l'accès au médium des techniques de partage de ce dernier utilisant l'évitement de collision basées sur l'ALOHA « slotté » sont utilisées [Kuo95]. Ils permettent aux radio-étiquettes de différer aléatoirement leur réponse.

3.1.2 L'IEEE 802.15.4

C'est le standard de communication dominant dans les RCSFs. Il définit les spécifications pour la couche physique et liaison de données OSI. La norme est maintenue par le groupe 4 du comité IEEE 802.15 qui gère tous les aspects relatifs aux réseaux

1. Industrielle, Scientifique et Médicale: ce sont des bandes de fréquences qui peuvent être utilisées sans accord ou licence préalable dans un espace réduit pour des applications de ces trois domaines.

personnel sans fil (PAN) [Tg4]. De nombreux autres standards tels que Zigbee et WirelessHART l'utilisent comme base mais l'étendent pour prendre en considération les couches supérieures (qui ne sont pas définies dans IEEE 802.15.4). La première version de ce standard a été publiée en 2003 suivie d'une révision en 2006.

La conception de ce dernier a été réalisée pour permettre des communications de faible portées (de l'ordre de quelques dizaines de mètres) en supportant des débits faibles : 20 kbit/s, 40 kbit/s et 250 kbit/s sur les fréquences respectives de 868 MHz, 915 MHz, et 2.4 GHz. La gestion de la consommation énergétique au niveau de la couche physique est optimisée de façon à utiliser des cycles d'activation/désactivation qui maintiennent la radio éteinte jusqu'à 99% du temps. La topologie réseau sous-jacente peut être en étoile, maillée ou une arborescence. Dans la première les nœuds communiquent uniquement avec le coordonnateur du réseau (PAN coordinator) qui lui est toujours en fonction, en attente active des communications. Dans les deux autres cas, la communication n'est pas restreinte au coordonnateur : tout nœud peut communiquer avec n'importe quel autre. Par ailleurs, deux types de fonctionnement sont possibles pour les autres nœuds :

- Implémenter toutes les fonctions possibles (FFD : Full Function Device), *i.e.* que le nœud peut assurer les fonctions d'un routeur, de coordonnateur de réseau et celui de dispositif embarquant un capteur pour l'acquisition des données.
- Supporter un nombre limité de fonctions (RFD : Reduced Fonction Device). Dans ce cas, contrairement au type précédent, le nœud ne peut que acquérir les informations et les transmettre au prochain saut : il ne peut remplir la fonction de routeur dans ce mode.

Au niveau de la couche physique, le dispositif matériel est constitué d'un émetteur/récepteur de radiofréquences et intègre les mécanismes de contrôle bas niveau tels que le contrôle de la qualité du signal (métrique LQI) et la détection de l'activité sur le canal (CCA : Channel Check Assessment). La couche liaison de données gère la transmission des trames de données sur le médium physique et le contrôle d'accès à ce support. Elle peut fonctionner selon deux modes : un mode asynchrone (sans balises) et un mode synchrone (avec envoi de balises). Dans le premier mode, les nœuds utilisent le mécanisme de CSMA/CA pour envoyer leurs trames. Dans le mode avec balise illustré par la figure 3.3, le coordonnateur envoie des balises contenant des informations pour synchroniser les nœuds. La super-trame divise le temps en trois périodes : (a) une période de contention (CAP : Contention Access Period), pendant laquelle les nœuds utilisent le mécanisme CSMA/CA « slotté » pour faire concurrence à l'accès du médium ; (b) une période sans contention contenant un certain nombre d'accès garanti à la tranche de temps (GTS : Guaranteed Time Slots) attribuée par le coordonnateur à des nœuds spécifiques ; (c) une période d'inactivité, où tous les nœuds (incluant le coordonnateur) peuvent passer en mode sommeil pour économiser leur énergie.

3.1.3 BLE

Le BLE (Bluetooth Low Energy) est une technologie basse « consommation énergétique » relativement récente (début 2010) basée sur la spécification Bluetooth [Blu]. Initialement conçu par Ericsson pour s'affranchir de l'utilisation des câbles lors de la

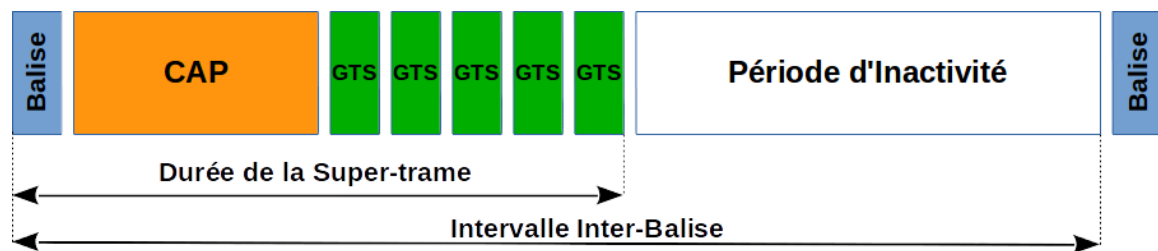


FIGURE 3.3 – Structure de la Super-Trame IEEE 802.15.4 en mode Synchrone

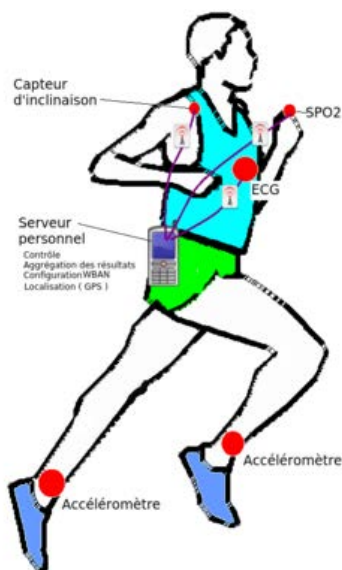
connexion des périphériques d'entrée/sortie, la technologie BluetoothTM permet de relier de petits dispositifs électroniques pour former un réseau personnel sans fil (WPAN : Wireless Personal Area Network) ou un réseau BAN (Body Area Network). Dans ce dernier type de réseau, le sujet est le corps humain (très utilisés dans les applications d'e-Santé) comme illustré par la figure 3.4. Les débits varient de 1 – 3 Mbit/s, pour une portée des communications de l'ordre d'une dizaine de mètres (dans le cas d'usage des smartphones et autres gadgets électroniques) voir une centaine de mètres (scenarii d'applications industrielles). Un nœud maître coordonne les transmissions des autres nœuds (esclaves) pour former un réseau en étoile (encore appelé Pico-réseau). La consommation énergétique de la technologie permet des durées de vie de l'ordre de plusieurs jours voir quelques mois sur des équipements opérant sur des batteries alcalines standards.

Des optimisations pour améliorer la consommation énergétique ont été réalisées sur cette version initiale pour étendre la durée de vie du réseau de 1 – 2 années : ce qui a donné naissance au BLE (aussi désigné sous le nom Bluetooth Smart ou Bluetooth 4.0) [Blu]. Cette nouvelle technologie loin de l'objectif initial de simple remplacement alternative sans fils aux câbles, permet les communications entre équipements basses consommations et offre une large gamme de possibilités et nouveaux services pour les applications de santé, domotique, sport, industriel, etc. Le débit maximal atteint est de 1 Mbit/s sur une portée de quelques dizaines de mètres.

3.1.4 IPv6

Avec ses 128 bits utilisés pour l'adressage des hôtes, ce système offre une espace quasi-inépuisable (3.4×10^{38} adresses) pouvant soutenir l'attribution d'identifiants uniques à chaque nœud de l'Internet des Objets. Un autre avantage d'adopter ce système d'adressage pour les objets de l'Internet est celui de pouvoir tirer partie de nombreux protocoles existants sous IP, et effectuer des communications de bout-en-bout sans nécessiter de traduction (à travers un dispositif comme NAT²).

2. Traduction d'adresse réseau (Network Address Translation) : Technique permettant de faire correspondre une seule adresse [publique] à tout un réseau. Elle est nécessaire lorsque le nombre d'adresses

FIGURE 3.4 – **BAN** : Réseau personnel sans fil pour l'individu [Ban].

IPv6 dispose également du SLAAC (StateLess Address Auto-Configuration) [TNJ07] qui donne la possibilité aux nœuds de se configurer automatiquement un identifiant sans intervention de l'administrateur ou du service DHCP. Les hôtes utilisent alors le protocole *Neighbor Discovery v6* (ND) pour envoyer des messages ICMPv6 de sollicitation d'adresse au routeur. Ce dernier répond en fournissant un préfixe IPv6 sur 64 bits, à partir duquel l'hôte sollicitateur se configurera une adresse IPv6 unique globalement « routable ». La partie identifiant d'hôte de l'adresse construite est basée sur l'adresse physique de ce dernier (EUI-64) [HD98]. Cette nouvelle possibilité permet aux nœuds de pouvoir se connecter au réseau IPv6 et d'être gérés moyennant une surcharge minimale. Toutefois, le protocole IPv6 impose des défis pour des couches MAC et physiques contraintes comme IEEE 802.15.4 ou celles apparentées. C'est dans ce contexte que 6LoWPAN [Mon+07; HT11] a été proposé comme couche d'adaptation entre les couches MAC de l'IEEE 802.15.4 et réseau pour le transport des paquets IPv6.

3.1.5 Web des Objets et CoAP

L'IoT est possible par l'intégration dans l'Internet classique d'un grand nombre d'Objets de tout type, dotés de capacités : d'acquisition d'informations, de calcul, de communication, et de contrôle de l'environnement. Ces nœuds (capteurs, actionneurs, puces RFID, ...) opèrent grâce à une vaste gamme de technologies, parmi lesquelles celles présentées plus haut. L'hétérogénéité des plateformes matérielles, des intergiciels, et constructeurs impliqués, nécessite des développeurs d'applications de prendre en considération tous ces aspects. Par ailleurs, la nature dynamique de l'IoT et les nouvelles possibilités qu'elles offrent imposent également à ces derniers d'être conscients des spécificités de chaque nœuds, de leurs fonctionnalités et des services offerts. Il

IP octroyées par le fournisseur des services Internet est inférieur au nombre de machines à connecter sur Internet.

est évident que les développeurs d'applications pour l'IoT ne souhaiterait pas tenir compte de tous ces détails.

Le Web s'est imposé comme un « standard de fait » pour la création d'applications interopérables et indépendantes de l'architecture matérielle des composants impliqués. Les services Web masquent ainsi l'hétérogénéité des technologies sous-jacentes et permettent aux développeurs de se concentrer sur les applications Webs développées. Le succès du Web a permis la réutilisation et l'extension des outils existants pour les adapter aux spécificités des nœuds de l'Internet des Objets. Ceci a donné naissance au concept du Web des Objets (W_{OT} : Web of Things) [Pfi+11].

En utilisant le formalisme de représentation des données du Web (REST), tout objet (quelque soit la technologie matérielle ou logicielle sous-jacente) peut publier les ressources dont il dispose. On pourrait par exemple vérifier l'occupation d'une salle de réunion à distance ainsi que le nombre de participants en consultant, grâce à un client web à une URL spécifique, le statut des capteurs de pression intégrés aux sièges. C'est dans cette optique que CoAP [SHB14] a été proposé comme solution permettant de faire tourner un serveur web adapté aux objets soumis à des contraintes de ressources matérielles et fonctionnant sur une couche physique IEEE 802.15.4. CoAP rend possible l'accès aux services et ressources possédés par les objets de l'Internet grâce aux URLs. De plus, le contrôle de ces objets est réalisé par une interface simple utilisant les opérations similaires aux GET et PUT du protocole HTTP standard. Mais contrairement à ce dernier, CoAP diffère dans sa conception par l'optimisation qui y est faite de la consommation énergétique et de la surcharge requise. Il s'appuie également sur le protocole UDP (beaucoup plus léger que TCP) pour relayer les messages à la couche transport. Le serveur CoAP hébergé par un nœud ne peut pas directement être accédé par un navigateur Web, il est nécessaire pour cela de disposer d'une passerelle réalisant la traduction CoAP-HTTP. Cette dernière peut être une application ou un plug-in intégré au navigateur, à l'instar de Copper [Kov11].

3.2 Problématique de l'interconnexion

Comme nous l'avons indiqué dans la section précédente, le RCSF constitue un bloc essentiel dans la réalisation de l'Internet des Objets. Avec l'intégration du protocole IP, tout nœud du RCSF devient accessible de l'intérieur comme de l'extérieur (*i.e.* Internet traditionnel) du réseau. Toutefois les exigences propres à chacune de ces deux plate-formes a conduit à la définition de protocoles et standards optimisés en consommation d'énergie et autres ressources matérielles pour la partie RCSF (*cf.* Figure 2.3), tandis que ceux utilisés pour les réseaux traditionnels sont à la fois flexibles et ouverts. Il est alors nécessaire de disposer à la frontière entre les deux réseaux (RCSF et le réseau traditionnel [Internet]) d'un certain nombre de nœuds implémentant les deux piles protocolaires : pile IP optimisée pour le RCSF et la pile TCP/IP standard. Ces passerelles assureront les tâches de conversion de protocoles et de relais de messages de l'un vers l'autre réseau, et vice-versa. La connectivité Internet de l'ensemble du RCSF devient alors tributaire du placement, de la disponibilité et de la gestion efficace des passerelles installées. Pour ce faire, certains facteurs doivent être prises en compte :

- **Problème de Hot-spot** : Le standard de routage pour le réseau de capteurs étant RPL, les nœuds proches de la racine (idéale pour servir également de passerelle)

reçoivent plus de trafic que les nœuds qui en sont éloignés. Ceci contribue à réduire d'avantage leur durée de vie comparée aux autres nœuds du réseau. De nombreuses approches adoptent le principe de changer dynamiquement la position de la racine pour remédier à ce problème [WCD06]. Mais cette solution est difficilement réalisable dans un environnement où la racine doit maintenir une connexion permanente vers Internet.

- **Profondeur (en nombre de sauts) par rapport à la racine du RCSF** : Les transmissions dans le réseau de capteurs sont instables, en raison de la qualité du lien et de la dynamique de l'environnement. Les nœuds éloignés de la station de base peuvent voir la qualité de leur communication se dégrader. Bien que de nombreuses métriques permettent d'améliorer la qualité de la communication (en sélectionnant les liens les plus fiables) une solution simple consisterait à partitionner le réseau de capteurs et d'interconnecter chaque partition à Internet pour assurer une meilleure qualité de la communication.
- **Disponibilité** : La passerelle constitue un point de défaillance unique pour la connectivité Internet. Pour assurer une meilleure disponibilité et tolérance aux pannes, la redondance devra être implémentée et plusieurs passerelles installées. Il est alors nécessaire de développer des mécanismes permettant d'utiliser le plus efficacement les passerelles disponibles en fonction des besoins.
- **Passage à l'échelle** : Les passerelles déployées devront pouvoir prendre en compte l'ajout de nouveaux nœuds et services au RCSF de façon à en limiter leur impact sur les performances du système existant.

Dans la troisième partie de cette thèse, nous nous intéressons à l'interconnexion du RCSF à l'Infrastructure Internet et le moyen de le réaliser de façon efficace à faible coût. [Zac+15] aborde la question de l'interconnexion dans un sens plus général et présente les principales questions de recherches soulevées. Une architecture générique sur la mise en œuvre d'une passerelle sur smartphone en utilisant les nouvelles technologies de communication sans fil : BLE (côté réseau d'objet) et 3G/4G ou WIFI (côté Infrastructure Internet), est également proposée.

3.2.1 Routeur de bordure

La passerelle peut être implémentée comme un routeur de bordure assurant la liaison entre le RCSF et Internet. Il connectera deux technologies de liaisons de données différentes : typiquement, l'IEEE 802.15.4 du côté RCSF et la technologie Ethernet ou l'IEEE 802.11 du côté Internet. Il assure également la séparation des plans de contrôle des deux domaines de routage : RPL dans le RCSF et tout autre protocole de routage (RIP, OSPF, statique, etc.) dans le second réseau. Pour remédier aux problèmes évoqués plus tôt, plusieurs routeurs de bordure devront être déployés. Ce qui permettra d'assurer une haute disponibilité du réseau (en cas de partition) et équilibrer la charge sur ceux-ci (pour le passage à l'échelle). Toutefois, disposer de plusieurs nœuds de ce type soulèvent d'autres problématiques, comme leur coordination, la gestion des pannes, ainsi que leur intégration transparente dans le réseau IPv6 existant.

Deru *et al.* ont récemment développé une passerelle simple mais puissante, permettant de faire transiter les communications du domaine filaire (Ethernet) au RCSF

(6LoWPAN), et vice versa [Der+13]. Nous nous en servons comme base dans la construction de notre plate-forme d'interconnexion dynamique entre le RCSF et Internet. Cette passerelle est désignée sous le terme *6LoWPAN Border Router* (6LBR) et elle peut s'exécuter sur matériel embarqué ou toute plate-forme disposant d'un système d'exploitation Linux. Trois principaux modes de fonctionnement sont possibles pour le 6LBR. Ils sont détaillés ci-dessous.

Mode routeur

Dans le mode routeur, le sous-réseau Ethernet et WPAN sont logiquement séparés, le 6LBR fournissant la route entre les deux. Chaque interface a un préfixe réseau distinct, comme on peut le voir sur la figure 3.5, où celui-ci est `a.a.a.a::/64` sur Ethernet et `b.b.b.b::/64` pour le réseau WPAN. Les deux domaines de diffusion sont parfaitement isolés l'un de l'autre. Pour assurer la communication entre les deux domaines, les nœuds Ethernet devront être configurés pour acheminer le trafic destiné au RCSF à travers le 6LBR. Du côté du RCSF, c'est la racine RPL qui sera chargée de router le trafic destiné au réseau Ethernet via le 6LBR.

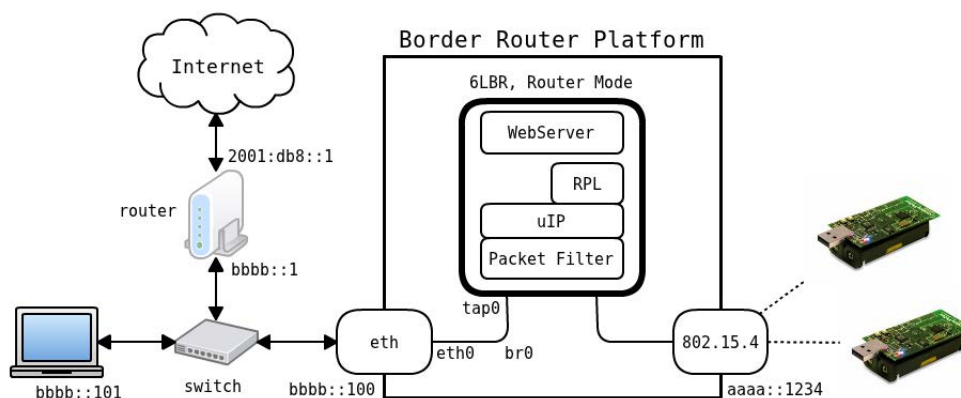


FIGURE 3.5 – Modes de fonctionnement Routeur du 6LBR [Der+13].

Mode pont transparent

Ce mode assure une fonction de commutation de base permettant ainsi à plusieurs RCSF d'être agrégés en DODAG global, la racine RPL pouvant être extérieure au RCSF. Toutes les trames de multidiffusion ou destinées à une interface IEEE 802.15.4 spécifique sont commutées de l'interface Ethernet vers le RCSF. Inversement, celles destinées à un hôte Ethernet spécifique ou celles constituant une adresse de multidiffusion sont commutées du RCSF par le 6LBR vers le segment Ethernet. L'ensemble du réseau constitue un seul domaine de diffusion et tous les nœuds partagent le même préfixe réseau (comme indiqué sur la figure 3.6). Le 6LBR dispose de sa propre adresse physique dans chacun des segments réseaux et traduit les adresses physiques/trames Ethernet en adresses/trames IEEE 802.15.4 et vice versa.

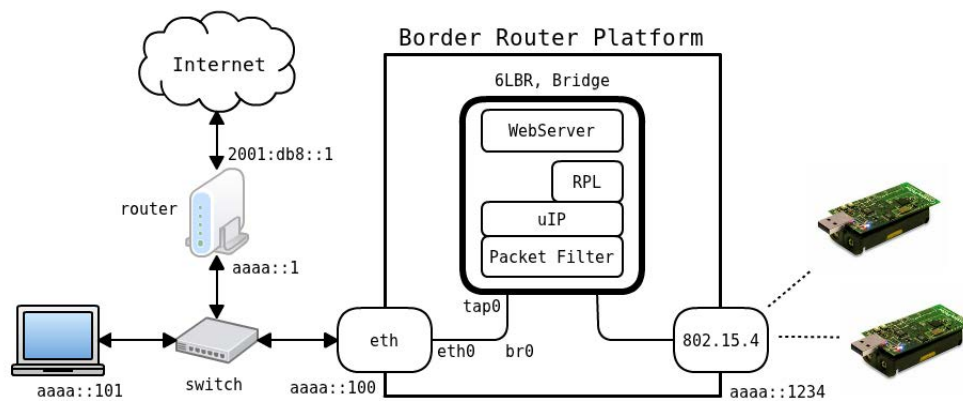


FIGURE 3.6 – Modes de fonctionnement Pont Transparent du 6LBR [Der+13].

Mode pont intelligent

Ici, le 6LBR joue le rôle de serveur de proximité (*proxy*) pour le protocole ND (*Neighbor Discovery*) de l'IPv6 dans le segment Ethernet. Les paramètres ND reçus sont utilisés pour configurer le RCSF. Plusieurs DODAG RPL sont agrégés pour créer un seul sous-réseau virtuel IPv6. Le serveur de proximité sert alors à rendre transparente la mobilité des nœuds d'un DODAG RPL vers un autre, lorsque ceux-ci se recouvrent. Le nœud rejoint le second DODAG s'il y découvre un lien de meilleure qualité.

3.2.2 Plate-forme matérielle pour la passerelle

Le 6LBR pouvant s'exécuter sur toute plate-forme matérielle compatible Linux, le Raspberry Pi se révèle comme une solution robuste et peu coûteuse [Ras]. En effet, la version Raspberry 3 modèle B dispose d'excellentes caractéristiques matérielles qui font d'elle un ordinateur complet (d'un format identique à celle d'une carte bancaire). Elle dispose : d'une mémoire centrale de 1 Giga-octets, d'un processeur ARM 64 bits 4-cœurs cadencé à 1.2Ghz et de 4 interfaces USB 2.0. On y retrouve par ailleurs de nombreuses interfaces de communication (Ethernet, IEEE 802.11n, Bluetooth classique et BLE) qui font d'elle la plate-forme « idéale » pour la mise en place de passerelles pour l'Internet des Objets. Elle peut être connectée via le port USB à un « dongle » IEEE 802.15.4 ou toute *mote* pour servir de pont entre RCSF et Internet. La figure 3.7 illustre un exemple de connexion Raspberry avec une *mote* TelosB.

3.3 Stratégies de sélection de passerelles dans les réseaux sans fil multi-sauts

Le problème de sélection de passerelles pour offrir un accès Internet à travers une infrastructure filaire a été étudié dans la littérature, aussi bien dans le contexte des RCSF que celui des réseaux maillés sans fil (WMN : *Wireless Mesh Network* en anglais).

Ashraf *et al.* proposent un algorithme de sélection de passerelle offrant la connectivité Internet dans un WMN [AAJ09]. Trois opérations sont mises en évidence pour réaliser l'interconnexion du réseau sans fil multi-sauts et l'infrastructure filaire reliée à

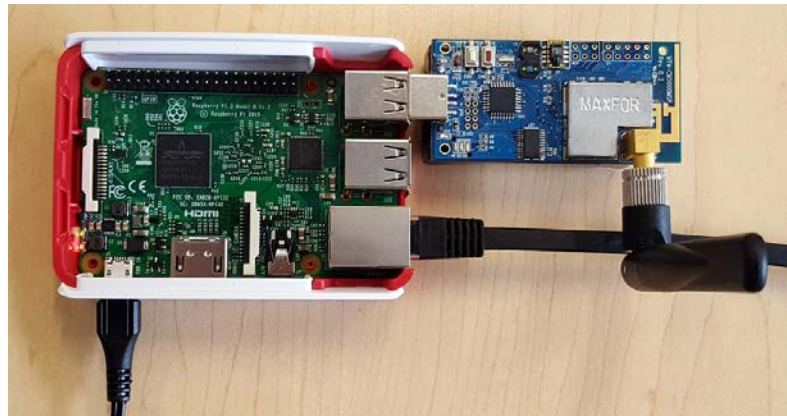


FIGURE 3.7 – Passerelle par Association : Raspberry – TelosB

Internet : une première phase de découverte de passerelle, suivie par une phase de sélection, et enfin la construction de la route vers cette dernière. La phase de découverte de la passerelle peut se faire suivant une approche proactive, réactive ou hybride. Dans l'approche proactive, les passerelles diffusent régulièrement des annonces pour signaler leur présence tandis qu'en mode réactif, ce sont les nœuds qui initient le processus de découverte par des messages de sollicitation de passerelle. La première génère une surcharge liée aux messages de contrôle pour la découverte des passerelles, mais offre l'avantage d'avoir des routes disponibles à tout moment. De plus elle offre une latence faible lors de l'installation des routes. Une fois les informations sur les passerelles obtenues par les nœuds, la sélection de passerelle et la construction de la route peuvent être effectuées. Le mode réactif limite la surcharge en évitant l'inondation du réseau. L'approche hybride combine les avantages de ces deux approches en définissant des zones dans le réseau où l'un ou l'autre des méthodes est utilisée.

[AAJ09] opte d'utiliser une approche proactive pour la découverte des passerelles et construit une métrique composite par combinaison linéaire de trois facteurs : charge, interférence de la route et estimation de la qualité du lien (ETX) pour le choix de la passerelle. La charge est calculée comme la taille moyenne de la file d'attente des paquets au niveau de la passerelle. L'idée étant celui de répartir la charge sur l'ensemble des passerelles plutôt que de surcharger un nombre réduit d'entre eux-ci. Pour estimer le facteur d'interférence, chaque nœud se base sur la mesure du taux d'occupation du médium dans son voisinage. Les résultats des simulations montrent de meilleurs résultats en terme de délai, de débit et de répartition de charge sur les passerelles avec cette approche, comparée à un scénario de sélection de passerelle la proche (en nombre de saut).

Dans le même sens, [Bei+15] propose une architecture pour la sélection de passerelle dans les réseaux capillaires. Un réseau capillaire [AB+12] est défini comme un ensemble de dispositifs électroniques connectés à travers des technologies d'accès à base d'ondes radio à courte portée (BLE, IEEE 802.11ah, IEEE 802.15.4, etc.) à un équipement plus puissant appelé passerelle capillaire. Cette dernière offre une connectivité Internet à l'ensemble du réseau capillaire à travers une technologie d'accès de plus grande capacité (4G ou LTE, Ethernet, etc.) Un tel réseau, à l'instar des RCSF est

déployé sans planification spécifique préalable. Pour assurer une connectivité maximale, plusieurs passerelles pouvant posséder des caractéristiques différentes en terme de connectivité, de capacité, et d'énergie (lorsqu'elle opère sur des batterie) seront installées. Le réseau devra alors pouvoir s'auto-configurer et diriger les communications vers la passerelle appropriée en fonction de critères spécifiques (minimiser la latence, maximiser la disponibilité des passerelle ou répartir la charge sur l'ensemble). Les auteurs [Bei+15] effectuent la sélection sur la base de critères spécifiques : informations d'accessibilité des passerelles (puissance du signal reçu, taux de perte de paquet et qualité de lien, ...), contraintes (nombre d'équipements admis par passerelle ou charge, capacité énergétique, coût de la liaison Internet, ...) et politiques. Les politiques serviront à établir une priorité sur la façon dont les contraintes sont prises en compte pour le choix de la passerelle appropriée. Elles devront être définies de façon statique par l'administrateur, puis disséminées dans le réseaux à destination des équipements. La méthode de sélection proprement dite peut se faire de trois façons :

- **Serveur centralisé** : Toutes les informations de sélection (accessibilité, contraintes et politiques) sont transmises à un serveur centralisé qui exécute l'algorithme de sélection et retourne les décisions aux nœuds concernées. Cette méthode présente l'inconvénient d'avoir un point de défaillance unique. Mais celui-ci peut être atténué en prévoyant un serveur de secours qui prendra le relais en cas de défaillance du serveur principal.
- **De façon distribuée sur chaque passerelle** : L'exécution de l'algorithme de sélection est effectuée par chaque passerelle. Celles-ci se synchronisent en partageant au préalable les informations requises pour le calcul. Le temps nécessaire à la réplication des informations de sélection sur les passerelles peut engendrer des inconsistances conduisant à des décisions incohérentes pendant de courts instants.
- **De façon autonome par chaque nœud du réseau** : Les contraintes et politiques de sélection sont disséminées dans le réseau ou sur la demande des nœuds qui exécutent localement l'algorithme de sélection. Le principal challenge ici est la limitation du nœud en ressources (calcul, mémoire et énergie). Seuls les nœuds les plus robustes seront capables de s'en servir. Ceux qui sont limités en ressources utiliseront l'une des deux premières méthodes.

L'algorithme de sélection calcule pour chaque passerelle une valeur de préférence sur la base des informations citées plus tôt. Ces informations sont mises à jours sur une base régulière. Des expériences en environnement réel montrent qu'une approche centralisée offre des résultats meilleurs, à la fois en terme de messages échangés dans le réseau (surcharge), que de distribution de charge sur l'ensemble des passerelles [Bei+15]. De plus, l'architecture proposée offre un certain degré de flexibilité dans l'utilisation des ressources pour assurer la connectivité du réseau capillaire à Internet.

Feng *et al.* considèrent le problème de sélection de passerelles dans le cadre de l'intégration d'un RCSF avec un réseau de mobiles (cellulaires) [Fen+13]. Ils prennent en compte aussi bien les paramètres de QoS du RCSF que celui de qualité du lien entre la passerelle et le réseau de mobiles. Côté RCSF, le nombre de sauts et la charge de la passerelle sont utilisés, tandis que la distance entre la station de base du RCSF est utilisée comme métrique renseignant sur la qualité de la communication avec le réseau de

mobiles. Une évaluation de performance est réalisée pour mesurer le taux de livraison des messages du RCSF vers le réseau de mobiles, le délai de bout-en-bout moyen, ainsi que le débit de la communication. Les simulations montrent que la considération des trois métriques offrent de meilleures performances, comparé à des scénarios où seules une ou deux métriques sont prises en compte.

[CYDM11] utilise une approche par la logique floue pour combiner : le nombre de nœuds candidats offrant des routes vers chaque passerelle dans le RCSF, avec l'énergie de ces nœuds. Les auteurs indiquent une amélioration de la consommation énergétique globale du réseau ainsi que la réduction de la latence des transmissions.

[WLY09] présente le problème d'une toute autre manière. Le but étant celui de déterminer le nombre de passerelles à installer, leur position en terme de nombre de sauts par rapport aux autres nœuds du réseau, tout en recherchant à répartir le trafic. Cette répartition est faite par minimisation de la variance de la charge supportée par chaque passerelle. Le problème est modélisé comme un programme linéaire multi-objectifs, où l'optimisation de ces trois critères est recherchée. *i.e.* minimiser le nombre de passerelles, tout en satisfaisant la demande des nœuds du réseau de façon équilibrée sur celles-ci. Dans le même ordre d'idée, [WBA09] montre que le placement des différentes passerelles dans le réseau influence grandement les performances. Ils expérimentent quelques heuristiques et comparent les approches utilisant le nombre de sauts avec celles basées sur la distance euclidienne par diagramme de Voronoï.

Deuxième partie

Contribution au Routage

Chapitre 4

Routage par l'Énergie

4.1 Modèle d'estimation d'énergie pour les capteurs

L'énergie est sans doute l'un des paramètres les plus importants à prendre en considération lors de la conception d'un réseau de capteurs sans fil. Les nœuds sont déployés pour des périodes plus ou moins longues, sans source d'alimentation continue. Par conséquent, ils doivent utiliser le plus efficacement possible l'énergie embarquée dans leur batterie. Une bonne connaissance de la consommation énergétique du nœud pendant son activité et la prise en compte de celle-ci à différents niveaux (application, routage, MAC ...) permettent de prolonger la durée de vie du réseau. Bien qu'il soit possible de mesurer le niveau de tension de la batterie, cette dernière ne constitue pas une bonne indication du niveau d'énergie restante dans la batterie ou déjà consommée par le nœud. L'estimation de la consommation énergétique du nœud peut être faite, soit grâce à un dispositif matériel directement intégré au nœud, soit de façon logicielle [Dun+07]. La mise en œuvre à travers un dispositif matériel est difficile à rajouter aux composants électroniques qui constituent le capteur. De plus, elle induit un coût très élevé [Jia+07]. Pour ces raisons, la grande majorité des plate-formes pour capteurs ne fournissent pas de mécanismes intégrés au matériel permettant de connaître l'énergie résiduelle des nœuds pendant leur fonctionnement.

Contrairement à un dispositif matériel, la mise en œuvre logicielle est beaucoup plus facile à intégrer à tout nœud, quel que soit le système d'exploitation cible. L'implantation de celle-ci ne nécessite que quelques lignes de code, sans avoir à modifier les applications et protocoles existants sur le capteur. Elle offre en plus l'avantage de pouvoir prendre en compte, en temps réel, l'information sur la quantité résiduelle d'énergie du nœud. Ceci permet d'influencer la prise de décision des solutions déployées afin d'améliorer la durée de fonctionnement du réseau. Deux approches peuvent être considérées pour estimer de façon logicielle l'énergie résiduelle d'un nœud : l'une basée sur un modèle linéaire, l'autre sur un modèle non linéaire.

4.1.1 Modèle linéaire

Dans le premier modèle, l'énergie totale E consommée par le nœud est constituée de l'ensemble des consommations du nœud à différents états de son fonctionnement [Dun+07].

$$E = \left(I_{cpu}t_{cpu} + I_{lpm}t_{lpm} + I_{tx}t_{tx} + I_{rx}t_{rx} + \sum I_{ci}t_{ci} \right) V \quad (4.1)$$

Celui-ci est décrit par la relation 4.1, où V est la tension d'alimentation du nœud, I_{cpu} l'intensité de courant lorsque le processeur est en activité, t_{cpu} le temps de cette activité.

I_{lpm} et t_{lpm} sont respectivement l'intensité de courant et le temps passé par le nœud en mode basse consommation (en l'occurrence en phase de sommeil). I_{tx} et t_{tx} , l'intensité de courant et le temps lorsque le nœud est en mode de transmission. I_{rx} et t_{rx} , l'intensité de courant et le temps en phase de réception. I_{ci} et t_{ci} , l'intensité de courant et le temps passé par le nœud lorsque divers autres composants tels que les interfaces de capture (humidité, température, ...) et LED sont activés.

Ce modèle a l'avantage d'être simple, donc facile à implémenter. Il suffit de faire appel au module de consommation d'énergie chaque fois que le nœud active ou désactive une de ses composantes matérielles (processeur actif ou en mode basse consommation, radio, capteur, etc.). Le module se chargera alors de conserver une estampille horaire à l'activation du composant concerné et à sa désactivation. La différence de ces deux horodatages fournit la durée passée dans l'état actif. Cette durée pourra alors être cumulée au cours du temps. En réitérant le processus, on obtient la liste des durées passées dans le mode actif pour tous les composants consommateurs d'énergie du nœud. La fiche technique du type de capteur utilisé indique les intensités de courants utilisées par chaque composant. L'énergie totale consommée par le nœud pourra alors être estimée. Partant d'une capacité énergétique initiale connue du capteur, le modèle permet de calculer l'énergie résiduelle en déduisant de la charge initiale les consommations énergétiques successives estimées.

Le principal inconvénient de cette approche est qu'elle ne prend pas en compte certains effets non linéaires liés à la diminution de charge dans les batteries équipant les nœuds. L'erreur associée à une estimation basée sur cette approche est significative car la durée de vie du nœud n'est pas linéairement corrélée à la charge initiale de la batterie.

4.1.2 Modèle non linéaire

Pour estimer de façon plus précise la charge restante dans la batterie du nœud, en plus de sa consommation énergétique à différents états d'activités, une bonne connaissance des caractéristiques inhérentes à sa batterie est nécessaire. Deux phénomènes doivent être considérés : l'*effet de capacité nominale* et l'*effet de récupération* [Pan+01]. Le premier décrit la décharge de la batterie lorsqu'une intensité de courant constante est tirée de celle-ci, résultant en une diminution de sa charge initiale. Pendant les périodes d'inactivité, la diffusion conduit à une réorganisation des électrons dans l'électrolyte résultant à une recharge apparente. Cela contribue à augmenter la durée d'utilisation de la batterie, encore connu sous le nom d'*effet de récupération*.

Les processus électriques et chimiques survenant à l'intérieur de la batterie ont été étudiés dans [RV03] et sont décrits par la relation mathématique 4.2.

$$\alpha = \int_0^L i(\tau) d\tau + 2 \sum_{m=1}^{\infty} \int_0^L i(\tau) e^{-\beta^2 m^2 (L-\tau)} d\tau \quad (4.2)$$

Le paramètre α représente la charge initiale de la batterie. β est le coefficient de diffusion des ions dans l'électrolyte, décrivant le comportement non linéaire de la batterie

pendant les périodes de décharge et de récupération. L , la durée d'utilisation de la batterie et $i(\tau)$ le courant consommé à l'instant τ . Dans ce modèle, la détermination de L , durée de vie du nœud, ne dépend que des seuls paramètres α et β . Bien que beaucoup plus précis pour décrire la décharge de la batterie au cours du temps, la relation 4.2 ne peut pas directement être implémentée dans le nœud capteur. Sa complexité nécessite de rechercher dans de grandes tables et de calculer une série infinie. Sachant que les capteurs sont très limités en ressources matérielles, des adaptations sont nécessaires. Pour ces raisons, nous utilisons l'approximation de [Rah+10] de ce modèle par la relation récursive 4.3.

$$\sigma(L_n) = \sum_{k=1}^n I_k \delta_k + \lambda \left(\sigma(L_{n-1}) - \sum_{k=1}^{n-1} I_k \delta_k \right) + 2I_n A(L_n, L_{n-1} + \delta_k, L_{n-1}) \quad (4.3)$$

$\sigma(L_n)$ est la charge consommée en $mA.min$ (milliAmpère minute) à la date L_n . Elle dépend récursivement de la charge consommée à la date précédente L_{n-1} . L'intervalle de temps $\Delta = L_n - L_{n-1}$ séparant les différentes tranches de temps est constant. Ainsi, on peut déterminer la charge restante dans la batterie à l'instant L_n en effectuant la différence entre $\sigma(L_n)$ et la charge initiale α .

Dans la formule 4.3, le premier terme : $\sum_{k=1}^n I_k \delta_k$ est déterminé suivant le même principe qu'en §4.1.1. L'effet de récupération modélisant le comportement non linéaire de la batterie est calculé par la fonction A , apparaissant dans le troisième terme de l'équation 4.3. Sa valeur est approximée en utilisant la fonction f issue de la relation 4.4. Cette fonction dépend de $\nu = \Delta - \sigma_k$, période d'inactivité à la date L_{n-1} et de σ_k , la période d'activité (*i.e.* celle où le nœud tire du courant de la batterie).

$$A(L_n, L_{n-1} + \delta_k, L_{n-1}) = \sum_{m=1}^{\infty} \frac{e^{-\beta^2 m^2 (L_n - L_{n-1} - \delta_k)} - e^{-\beta^2 m^2 (L_n - L_{n-1})}}{\beta^2 m^2} \simeq \frac{f(\nu)}{\beta^2} - \frac{f(\Delta)}{\beta^2} \quad (4.4)$$

Les termes de la fonction d'approximation f sont donnés par les équations 4.5 et 4.6. Pour plus de détails sur ces approximations, se référer à [Rah+10].

$$\frac{f(\nu)}{\beta^2} = \frac{\nu}{2} - \frac{\sqrt{\pi}}{\beta} \cdot \sqrt{\nu} + \frac{\pi^2}{6\beta^2} \quad (4.5)$$

$$\frac{f(\Delta)}{\beta^2} = \sum_{m=1}^{\infty} \frac{e^{-\beta^2 m^2 \Delta}}{\beta^2 m^2} = c_0 \quad (4.6)$$

Le paramètre β ou coefficient de diffusion physique dans ces relations est une constante dont la valeur ne dépend que de la batterie utilisée. Il a été estimé selon la méthode des moindres carrées et sur des mesures relatives à de nombreux tests de décharge de la batterie (pour différentes valeurs de charge initiale α) [NF13]. Dans la relation 4.6, vu que la série converge rapidement (à partir de $m = 10$), on obtient une constante c_0 calculable hors ligne.

4.1.3 Estimation logicielle en ligne

Le principal défi réside dans la mise en œuvre de ce modèle et son implantation dans un système d'exploitation réel pour capteur sans fil : Contiki dans notre cas [DGV04]. Les principaux verrous étant la taille mémoire limitée et le fait que les nœuds possèdent un microcontrôleur 16 bits ne réalisant que des opérations entières. Ces verrous levés, les nœuds pourront estimer leur charge restante en temps réel. Il sera alors possible d'adapter leur comportement pour influencer sur les choix à faire par les protocoles déployés et ainsi maximiser la durée de fonctionnement du réseau.

Le modèle récursif décrit par l'équation 4.3 lève le premier verrou. Il nous affranchi de la nécessité de disposer d'une mémoire importante pour conserver l'historique de tous les états précédents. Ainsi, la récursion rend possible l'estimation de la capacité énergétique du nœud en se basant sur la valeur précédente qui est mise à jour régulièrement. Par ailleurs, en plus du coefficient de diffusion physique β et la constance c_0 de l'équation 4.6, plusieurs autres paramètres sont calculés hors ligne (cf. tableau 4.1). Le paramètre λ dans le deuxième terme du modèle récursif 4.3 est défini par le rapport $\frac{A(L_{n+1}, \delta_1, 0)}{A(L_n, \delta_1, 0)}$ pour chaque valeur de n . Ce ratio est majoré par la constante $e^{-\beta^2 \Delta}$. Constante qui reste inchangée, $\forall n$ pour $\beta \geq 1.1$ [Rah+10]. En pratique, la période d'inactivité ν varie autour d'une valeur a . Dans l'équation 4.5, on peut réécrire $\sqrt{\nu} \approx \sqrt{a} + (\nu - a) \frac{1}{2\sqrt{a}}$, où $\sqrt{a}$ et $\frac{1}{2\sqrt{a}}$ sont également calculés hors ligne. La détermination préalable de toutes ces valeurs simplifie grandement les calculs, et nous permet d'implanter le modèle sur l'architecture simple du microcontrôleur équipant les nœuds.

	π^2	$\sqrt{\pi}$	β	c_0	λ	a	$\sqrt{a}$	$\frac{1}{2\sqrt{a}}$
Réelle	9.869	1.772	1	1.337	0.967	0.03	0.173	2.886
Entière	9 869	1 772	1 000	1 337	967	30	173	2 886

TABLE 4.1 – Constantes du modèle

Le tableau 4.1 consigne l'ensemble des valeurs des constantes déterminées hors ligne et utilisées dans le modèle. Tous les calculs étant réalisés en arithmétique entière sur un microcontrôleur MSP430, nous multiplions les valeurs réelles par 1000 pour ne se limiter qu'aux opérations entières.

Une implémentation de ce modèle a été réalisée pour des nœuds TelosB et WSN430 sous Contiki [NF13]. Ces deux plates-formes sont dotées d'un microcontrôleur MSP430 et d'une puce radio CC2420 et CC1100, respectivement. Le tableau 4.2 indique les intensités de courant tirées de la batterie pour une plate-forme matérielle de type TelosB à divers états d'activités : CPU, LPM, Tx et Rx. Ils correspondent respectivement aux phases de calcul, basse consommation, transmission et réception. Les données présentées sont extraites de la fiche technique du TelosB et nous permettent de calculer la partie linéaire du modèle récursif. Les nœuds démarrent tous avec une charge initiale fixée à $880mA$, représentant 100% de charge. Cette charge est réévaluée toutes les $2s$ par un processus qui implémente le modèle. La fréquence de réévaluation correspond à Δ , intervalle de temps d'estimation de la charge restante. Les adaptations réalisées nous permettent une implémentation avec une empreinte mémoire de 12Kb représentant 25% de la mémoire flash totale dont dispose le nœud.

Module	Mode opérationnel	Courant (mA)
MSP430	Actif (CPU)	1.8
	Sommeil (LPM)	0.0545
Tranceiver Radio CC2420	Tx	17.4
	Rx	18.8

TABLE 4.2 – Mesures des consommations de courant : TelosB (sky)

Suivant les recommandations relatives à la métrique énergie du standard de routage RPL [Vas+12], l'énergie est estimée par un entier sur 8 bits. Pour obtenir une granularité fine de l'évolution énergétique d'un nœud donné, nous effectuons les calculs avec une précision de 5 décimales (255×10^5 correspond à la capacité énergétique maximale).

4.1.4 Résultats de simulation du modèle non linéaire

Les simulations ont été réalisées sur la plate-forme Cooja [OD06], avec des nœuds de type TelosB équipés du système d'exploitation Contiki [DGV04]. Une application simple de collecte de données, où les nœuds (*Sender*) envoient des informations applicatives à un point (*Sink* ou puits) par routage multi-sauts est développée. Nous utilisons pour cela le protocole RPL [Win+12]. Les nœuds démarrent tous avec une batterie de charge initiale maximale fixée à 880mAh .

4.1.4.1 Profil énergétique de démarrage

Nous débutons la simulation avec uniquement deux nœuds : un *sender* et un *sink*. La figure 4.1 illustre le profil énergétique du *sender*, relatif aux premiers instants d'activités. Les temps passés dans chacun des états (CPU, Tx et Rx) sont également présentés. On peut voir que, pendant la première minute (figure 4.1a), une forte décroissance de la batterie a lieu. Celle-ci est corrélée à la forte activité sur le schéma 4.1b, associée à la même période. Cela se justifie par la période de mise en place du réseau. Il y a une initialisation du système et de la pile protocolaire (utilisation de la CPU), suivi de la découverte du voisinage (activités Tx et Rx). Durant la seconde minute, puis les minutes suivantes, l'effet de récupération est assez marqué. Il correspond à des phases de plus faibles activités de la radio.

4.1.4.2 Profil énergétique selon la dynamique réseau

Le profil énergétique du nœud 2 est examiné sous divers scénarios. La figure 4.2a présente les différentes conditions auxquelles est soumis ce dernier. À la dixième minute de l'expérience, cinq autres nœuds sont rajoutés à la topologie initiale. Ceci nous permet d'évaluer l'évolution énergétique liée à l'accumulation du trafic sur le nœud relayer. À la vingtième minute, les nœuds précédemment ajoutés sont retirés. Puis, 15 minutes plus tard, le nœud 2 perd sa connectivité au *sink* (retiré de la topologie).

La figure 4.2b illustre l'évolution énergétique du nœud 2 sous les conditions précédentes, à la fois suivants les modèles linéaire et non linéaire (avec effet de récupération). La première chute importante de la batterie survient à la dixième minute du fait de la

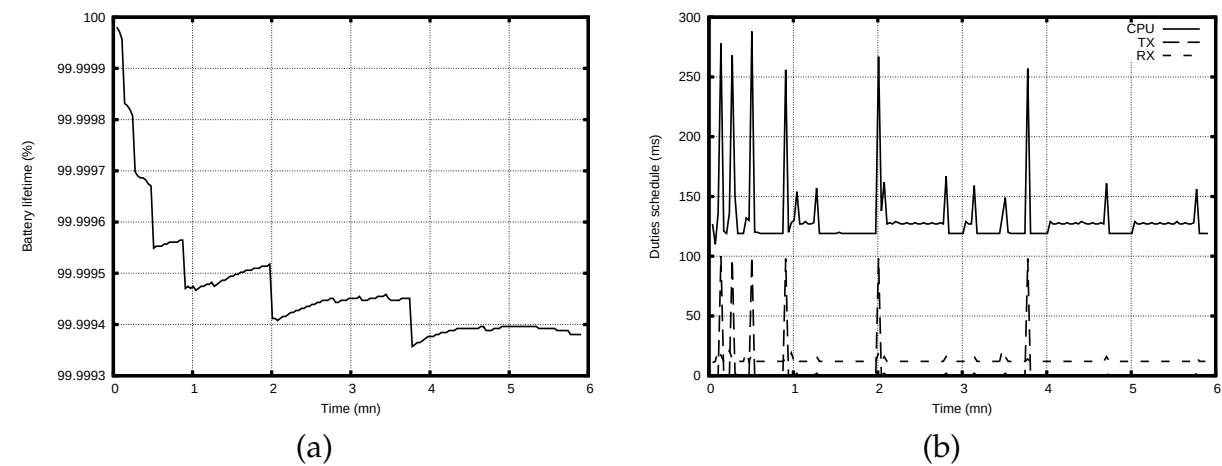


FIGURE 4.1 – Premiers instants d'activité

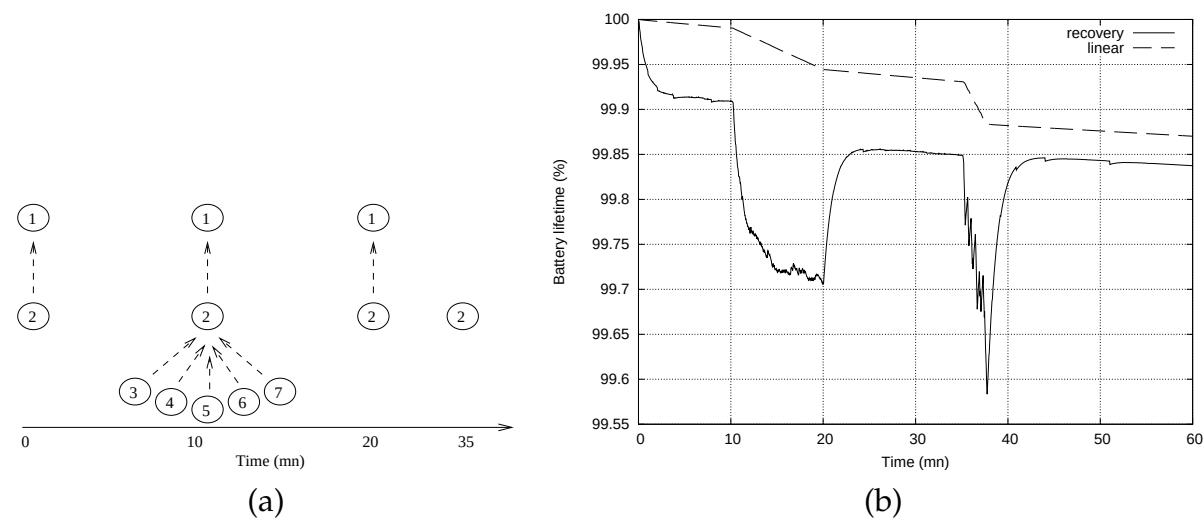


FIGURE 4.2 – Dynamique du profil énergétique

quantité de trafic généré par les nœuds 3 à 7. Dès que les nœuds rajoutés sont retirés à la vingtième minute, une forte récupération a lieu. Celle-ci se stabilise à un certain niveau, après quoi la perte énergétique devient moins abrupte. Il faut noter que pour le modèle linéaire où l'on n'a pas de récupération, seule la pente représentant la déperdition énergétique change entre ces deux dates.

À la trente-cinquième minute lorsque le *sink* est retiré, on observe une forte chute de la batterie du nœud 2. Elle est due à une utilisation importante de la radio pour la transmission des messages de contrôle RPL de type *sollicitation de voisin* (DIS). Le nœud qui a perdu sa connectivité au *sink*, pendant un moment recherche activement un prochain saut. Après environ trois minutes de recherche infructueuse, le nœud diminue le rythme de ses sollicitations, et on observe à nouveau une récupération importante. Encore ici, nous notons que le modèle linéaire est moins réactif. Cette différence nette entre les deux modèles, nous suggère une sous-estimation ou surestimation l'énergie du nœud par cette dernière, relativement à la dynamique du réseau.

4.1.4.3 Erreur d'approximation

Les adaptations réalisées pour implémenter le modèle dans l'environnement contraint du capteur intègre nécessairement des erreurs d'approximation. Celles-ci sont à la fois dues aux opérations réalisées exclusivement en arithmétique entière du microcontrôleur MSP430 et aux erreurs d'arrondies liées aux divisions. Une version du modèle en arithmétique flottante a été implémentée en script Perl. La figure 4.3 permet d'évaluer le gap entre les deux implémentations. Nous observons que l'estimation énergétique calculée par les capteurs est légèrement inférieure à la version flottante, mais la différence entre les deux reste constante.

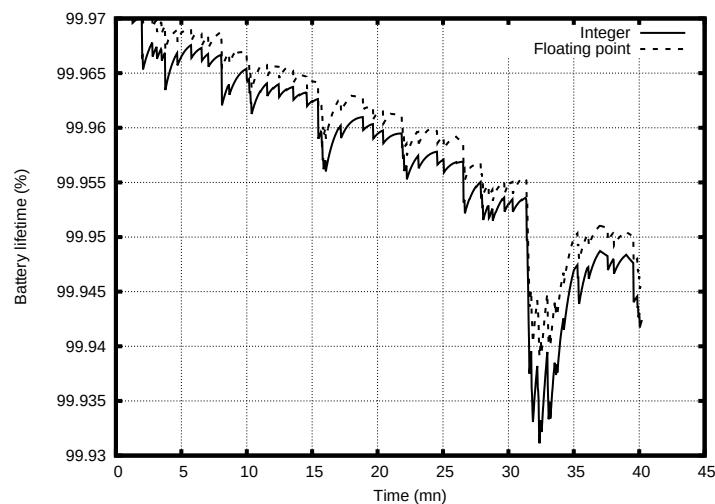


FIGURE 4.3 – Gap entre les implémentations flottante et entière

4.1.4.4 Durée de vie des nœuds

Le principal intérêt de l'estimation de l'énergie des nœuds est de pouvoir planifier la durée d'utilisation du réseau déployé sans remplacement des capteurs ou recharge

de leur batterie. Le système d'exploitation Contiki offre en dessous la couche MAC plusieurs cycles d'utilisation radio (RDC : *Radio Duty Cycle* en anglais). Certaines de ces sous-couches maximisent l'utilisation de l'énergie est désactivant la radio autant que possible (contikiMAC), tandis que d'autres laissent la radio ouverte en permanence (sicslowMAC). Le tableau 4.3 présente le temps nécessaire pour que la batterie soit

RDC	Durée Batterie (jours)	Débit (<i>ppts/mn</i>)	Durée Batterie (jours)
contikiMAC	128	60	77
X-MAC	30	20	105
CX-MAC	26	12	113
sicslowMAC	1.8	2	124
Débit : 1 <i>ppt</i> / <i>mn</i>		RDC : contikiMAC	

TABLE 4.3 – Durée d'utilisation de la Batterie

complètement épuisée (*i.e.* 0%) sous diverses RDC. Dans la partie gauche du tableau, le débit d'émission des paquets est le même pour tous les scénarios, soit 1 paquet toutes les minutes. Comme on aurait pu le prévoir, contikiMAC correspond à la RDC où la batterie a une durée d'utilisation maximale, tandis que sicslowMAC offre une durée minimale. Par ailleurs, l'impact du débit de paquets applicatifs sur la batterie a également évalué (partie gauche du tableau 4.3). Cette évaluation a été réalisée avec contikiMAC, qui est la plus économe en énergie. Nous faisons varier le début de 1 à 60 paquets par minutes. Au regard de ces expériences, nous pouvons conclure que la sous-couche de contrôle du canal radio : RDC, a un fort impact sur la consommation du nœud. Cette influence est encore plus importante que le débit applicatif, et contribue grandement à sa durée de vie.

4.2 Fonction d'Objectif RPL basée sur l'énergie

Le réseau de capteurs sans fil présente des caractéristiques uniques qui rendent difficile l'utilisation des solutions et protocoles des réseaux filaires ou sans fil (tel que les réseaux Ad-hoc) traditionnels. Pour faire face aux contraintes liées à la capacité limitée des nœuds en terme de puissance de calcul et de mémoire, ainsi qu'aux faibles débits de données, à l'instabilité et la non fiabilité du canal radio, le groupe de travail ROLL de l'IETF a publié le standard de routage RPL [Win+12], adapté à ce type d'environnement. Plusieurs métriques de routage sont prévues pour être prises en compte dans la construction de la topologie de réseau. Celles-ci sont décrites dans la RFC6551 [Vas+12], document d'accompagnement du standard. À notre connaissance, aucune implémentation de fonction d'objectif (OF) RPL prenant en considération l'énergie comme *unique* métrique de routage n'existe dans la littérature. Seules deux implémentations de fonctions d'objectif sont largement déployées, la première utilisant le nombre de sauts comme métrique de routage, la seconde se servant de l'ETX - métrique renseignant sur la fiabilité de la transmission. Les recommandations sur la mise en œuvre de ces deux métriques sont publiées dans les RFCs 6552 et 6719 [Thu12; GL12] accompagnant le standard. Nous proposons de nous servir de l'énergie

restante estimée dynamiquement par le nœud, suivant le modèle présenté en §4.1.2 pour organiser la topologie de routage [Kam+13].

4.2.1 Capacité énergétique de la route

Nous définissons la capacité énergétique C_i du chemin allant du nœud i à la racine, comme étant la valeur minimale d'énergie de tous les nœuds sur ce chemin, i compris. Cette information codée sur 8 bits est propagée dans des messages de contrôle RPL de type DIO, transportant d'autres paramètres de configuration. Le nœud puits (ou racine) annonce toujours un niveau de batterie maximal ; i.e. $MAX_{energy} = 255$. Ce dernier est supposé être alimenté par une source d'énergie permanente. Lorsqu'un nœud reçoit plusieurs annonces (émanant de prochains sauts potentiels) de son voisinage V_i , il choisit comme prochain saut, le voisin lui indiquant une capacité énergétique de chemin la plus élevée. Par la suite, le nœud calcule la capacité énergétique du chemin passant par le parent choisi, chemin ayant comme origine le nœud lui-même. Dès lors, ce dernier commence à propager ses propres annonces DIO.

Plus formellement, la capacité énergétique du chemin issu du nœud i est :

$$C_i = \min[\max_{j \in V_i}(C_j), E_i] \quad (4.7)$$

avec E_i énergie résiduelle du nœud i et V_i l'ensemble des voisins de i pouvant le conduire vers la racine. Pour un chemin issu d'un nœud i donné, la relation 4.7 correspond à la valeur minimale d'énergie sur le chemin le plus intéressant et l'énergie du nœud i , lui-même. Cette métrique nous semble justifiée, car la route choisie est contrainte par la capacité énergétique du nœud le plus faible sur le chemin. Par ailleurs, étant donné plusieurs voisins potentiels, le chemin le plus intéressant est celui passant par le voisin annonçant la plus grande métrique de route.

Pour illustrer le principe d'utilisation de cette métrique, considérons la topologie réseau illustrée par la figure 4.4. Le nœud 1 est la racine alimentée en permanence ($E = 255$). Les autres nœuds ont, à l'instant capturé, les niveaux de batterie indiqués. Les liens en pointillés indiquent la possibilité de communication entre nœuds voisins. Le nœud 6 reçoit les annonces DIO de ses voisins (potentiel prochain saut) : 3, 4, 5, et 8. Ces annonces indiquent respectivement des métriques de chemin $C_3 = 220$, $C_4 = 215$, $C_5 = 217$ et $C_8 = 217$ (notez que 8 n'annonce pas sa propre valeur énergétique $E_8 = 240$ comme métrique de chemin, mais plutôt celle contraignant la route vers la racine). Ainsi sur la base la relation 4.7, le nœud 6 choisira 3 comme parent qui est la meilleure route vers le nœud 1, du point de vue capacité énergétique résiduelle.

Le nœud 3 qui est le plus sollicité, épuisera plus rapidement sa batterie. Considérons qu'à une autre date celle-ci ait diminué de 20 unités, contre 5 unités pour les autres nœuds. La figure 4.5 montre comment le réseau réagira en fonction de la nouvelle valeur de métrique énergie propagée par les nœuds :

1. Le nœud 3 n'est plus le meilleur parent pour 6. Ce dernier découvre une route plus intéressante passant par 4 ($C_3 = 200$ tandis que $C_4 = 210$).
2. Pareillement, le nœud 5 passe par 6 qui offre une meilleure route que 3 ($C_3 = 200$ tandis que $C_6 = 205$).

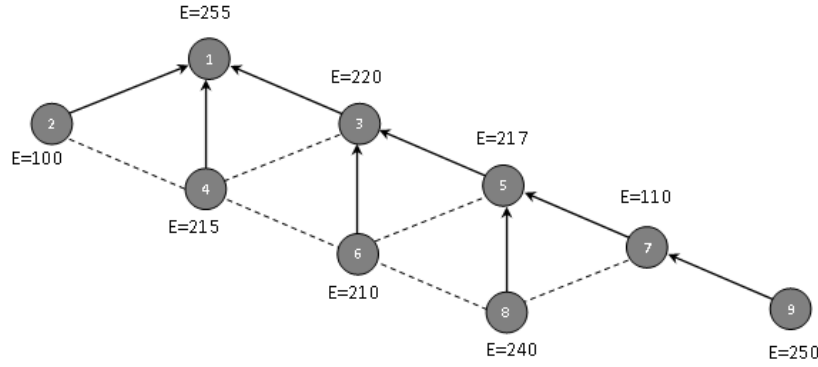
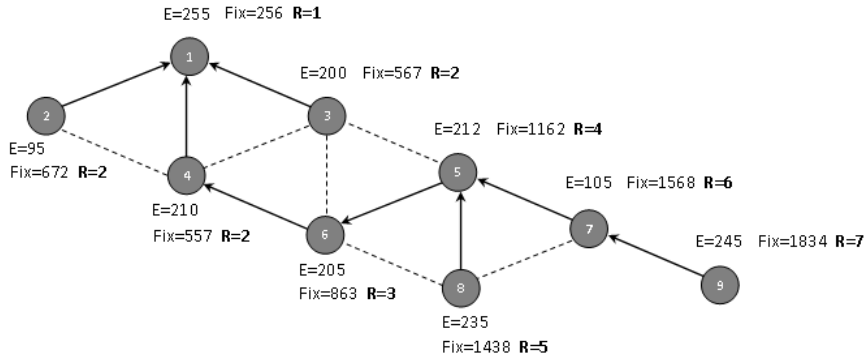


FIGURE 4.4 – Capacité énergétique d'une route

Le réseau s'adaptait dynamiquement à ces changements, et nous faisons l'hypothèse que les niveaux des batteries s'équilibreront au fur et à mesure.

4.2.2 Calcul de rang

Le *rang* est implémenté comme une valeur à virgule fixe où la *partie entière* et la *partie décimale* sont déterminées par le paramètre `MinHopRankIncrease` (fourni par la racine et transporté par les annonces DIO). Lorsque le *rang* est utilisé pour déterminer la position relative de deux nœuds ou pour détecter la présence des boucles dans le réseau, seule la partie entière est utilisée. Par contre, l'OF réalise le calcul de la valeur complète en virgule fixe (16 bits).

FIGURE 4.5 – Calcul de *rang* de l'OF-énergie

Soit un nœud N ayant choisi comme prochain saut le parent P . N calcule son *rang* en ajoutant au *rang* du parent, un accroissement (valeur strictement positive). L'accroissement est déterminé comme indiqué par la formule 4.8, avec $pas = MAX_{energy} - E_N$.

$$Rank(N) = Rank(P) + Rank_{incr} \quad (4.8)$$

avec $Rank_{incr} = pas + MinHopRankIncrease$

Cette formule garantit la propriété de monotonie du *rang* qui augmente d'au moins un point (*i.e.* MinHopRankIncrease) entre un nœud donné et son parent. Ceci limite dans le voisinage les nœuds susceptibles d'être parent d'un nœud donné, *i.e.* ceux ayant une valeur de *rang* strictement inférieur à celle de ce dernier. L'augmentation de *rang* est d'autant plus importante que la batterie du nœud fils est faible, puisque le *pas* participant à la partie décimale du *rang* dépend de celui-ci. Par effet de cumul et des pénalités liées aux niveaux des batteries des nœuds sur le chemin, la différence de *rang* entre un nœud et son parent peut être de plus d'un point (voir Tableau 4.4 – accroissement de *rang* entre les nœuds 5 et 7). La figure 4.5 illustre comment nous dérivons la valeur du *rang* de la métrique de chemin, et le Tableau 4.4 met l'accent sur le calcul des *rangs* du chemin 1 – 4 – 6 – 5 – 7 – 9. La valeur du *rang* de la racine est la même que celle du paramètre MinHopRankIncrease (256 dans notre implémentation). Il

ID du Nœud	Valeur en Virgule Fixe (16 bits)	Partie Décimale (8 bits)	Partie Entière (Int)
1	256	0	1
4	557	45	2
6	863	95	3
5	1162	138	4
7	1568	32	6
9	1834	42	7

TABLE 4.4 – Calcul détaillé du rang

convient de remarquer que les nœuds 7 et 8 ont un parent commun (nœud 5), mais leur *rang* respectif nous montre que la position relative de 8 vers la racine est meilleure que celle de 7.

4.2.3 Optimalité et absence de boucle

Les différentes exigences liées à l'utilisation d'un réseau de capteurs et à l'application déployée donne lieu à la conception de diverses métriques de routage. Dans [YW03], les auteurs fournissent une analyse systématique de la relation entre métrique et protocole de routage. Ces travaux repris par [Sob08] définissent un cadre formel des propriétés algébriques de base que la métrique doit posséder pour garantir la consistance, l'optimalité et l'absence de boucle du routage conçu.

- **consistance** : Le routage sera dit consistant, si la décision de transmission d'un paquet le long d'un chemin donné est consistant pour tous les nœuds sur ce chemin. En d'autres termes, si le nœud n_1 transmet le paquet en direction du nœud n_k sur le chemin $p(n_1, n_k) = \langle n_1, n_2, \dots, n_k \rangle$, tous les autres nœuds sur ce chemin en font de même, *i.e.* n_2 transmettra le paquet vers n_k selon le chemin $p(n_2, n_k) = \langle n_2, n_3, \dots, n_k \rangle$, ainsi de suite pour chacun des nœuds $n_3, \dots, n_{k-1}$.
- **optimalité** : Il s'agit d'une propriété générique du protocole du routage. Elle assure la transmission du paquet sur le chemin le plus « léger » (au regard de la métrique implémentée). L'absence de cette propriété conduit à un routage sous-optimal, donc inconsistant.

- **absence de boucle** : C'est l'une des propriétés les plus importantes, car ayant un impact direct sur la qualité de service. Une boucle de routage se produit lorsqu'un paquet transmis passe par un ensemble de routeurs donné, puis transite à nouveau par l'un des routeurs de l'ensemble sans atteindre sa destination finale.

Pour garantir ces trois propriétés fondamentales, les conditions nécessaires et suffisantes que doit satisfaire la métrique sont les propriétés d'*isotonie* et de *monotonie* [Sob08]. Plus formellement, une métrique de routage peut être décrite par une algèbre sur le quadruplet $(\Sigma, \oplus, w, \preceq)$, où Σ est l'ensemble des chemins, $\oplus$ l'opérateur de concaténation des chemins, w une fonction qui transforme un chemin donné à son coût associé et $\preceq$ une relation d'ordre.

Isotonie :

Cette propriété signifie que la relation d'ordre entre deux chemins est préservée, si chacun d'entre-eux est préfixé ou suffixé par un chemin commun. En d'autres termes, la structure algébrique $(\Sigma, \oplus, w, \preceq)$ est dite isotonique si et seulement si : $\forall a, b, c \in \Sigma$, $w(a) \preceq w(b)$ implique à la fois $w(c \oplus a) \preceq w(c \oplus b)$ et $w(a \oplus c) \preceq w(b \oplus c)$.

Monotonie :

La propriété de monotonie garantit que le coût associé à un chemin ne décroît pas (au sens de la relation d'ordre) lorsque celui-ci est préfixé ou suffixé par un autre chemin. Plus formellement, l'algèbre $(\Sigma, \oplus, w, \preceq)$ est monotone si : $w(a) \preceq w(c \oplus a)$ et $w(a) \preceq w(a \oplus c)$ est vérifiée $\forall a, c \in \Sigma$.

Proposition 1 (Isotonie de la métrique énergie).

Soit la structure algébrique $(\Sigma, \oplus, E_{\min}, \geq)$, où $E_{\min}(p)$ est la plus petite énergie résiduelle de nœud sur le chemin p , algèbre définie sur la relation d'ordre $\geq$. $(\Sigma, \oplus, E_{\min}, \geq)$ est isotonique.

Démonstration. L'opérateur $\oplus$ de concaténation des chemins est commutatif pour la fonction de coût $E_{\min}$. Formellement, $E_{\min}(x \oplus y) = E_{\min}(y \oplus x)$, $\forall x, y \in \Sigma$. Nous pouvons donc nous limiter à la démonstration à celle de l'*isotonie-gauche* (Soit, uniquement pour ceux des chemins préfixés).

En effet : Considérons deux chemins a et b tels que $E_{\min}(a) \geq E_{\min}(b)$. Nous voulons montrer que : $\forall c \in \Sigma$, $E_{\min}(c \oplus a) \geq E_{\min}(c \oplus b)$. Pour la première inégalité, nous distinguons trois cas de figures :

1. $E_{\min}(c) > E_{\min}(a)$
On a $E_{\min}(c \oplus a) = E_{\min}(a)$ et $E_{\min}(c \oplus b) = E_{\min}(b) \implies E_{\min}(c \oplus a) \geq E_{\min}(c \oplus b)$.
2. $E_{\min}(a) \geq E_{\min}(c) \geq E_{\min}(b)$
On a $E_{\min}(c \oplus a) = E_{\min}(c)$ et $E_{\min}(c \oplus b) = E_{\min}(b) \implies E_{\min}(c \oplus a) \geq E_{\min}(c \oplus b)$.
3. $E_{\min}(b) > E_{\min}(c)$
On a $E_{\min}(c \oplus a) = E_{\min}(c \oplus b) = E_{\min}(c) \implies E_{\min}(c \oplus a) \geq E_{\min}(c \oplus b)$.

□

Proposition 2 (Monotonie de la métrique énergie).

Soit la structure algébrique $(\Sigma, \oplus, E_{\min}, \geq)$, où $E_{\min}(p)$ est la plus petite énergie résiduelle de nœud sur le chemin p , algèbre définie sur la relation d'ordre $\geq$. $(\Sigma, \oplus, E_{\min}, \geq)$ est monotone.

Démonstration. Pour la même raison évoquée plus haut (commutativité de $\oplus$ pour $E_{\min}$ dans Σ), la démonstration peut se limiter à l'établissement de la *monotonie-gauche*. Cela revient à montrer que : $\forall a, c \in \Sigma, E_{\min}(a) \geq E_{\min}(c \oplus a)$.

Pour tout chemin préfixé c , deux situations sont possibles :

1. $E_{\min}(a) \geq E_{\min}(c)$
On a $E_{\min}(c \oplus a) = E_{\min}(c) \implies E_{\min}(a) \geq E_{\min}(c \oplus a)$.
2. $E_{\min}(c) > E_{\min}(a)$
On a $E_{\min}(c \oplus a) = E_{\min}(a) \implies E_{\min}(a) \geq E_{\min}(c \oplus a)$.

□

Nous avons ainsi montré que : la métrique RPL basée sur l'énergie proposée est à la fois isotonique et monotone. En conformité avec [YW03], nous pouvons conclure qu'elle satisfait les exigences de consistance, d'optimalité et d'absence de boucle.

4.3 Expérimentations et évaluation

4.3.1 Environnement

Pour évaluer notre proposition, nous avons déployé deux scénarios de routage : le premier, basé sur notre proposition de routage par l'énergie, le second, utilisant la version très répandue de RPL : le routage avec l'ETX. Les expériences ont été réalisées dans le simulateur Cooja [OD06]. Les nœuds déployés embarquent une image réelle du système d'exploitant Contiki [DGV04] grâce à l'outil Java MSPSim [Eri+08] qui émulent les instructions machines des plate-formes dotées d'un microcontrôleur MSP430. La topologie réseau est une grille $2D$ $300m \times 300m$ de 20 capteurs, la racine étant située au coin supérieur gauche. Chaque nœud œuvre dans un rayon de transmission de $120m$ et d'interférence de $140m$, envoyant régulièrement des datagrammes UDP avec un ratio de transmission/réception fixé à 80%. Nous utilisons ContikiMAC [Dun11], à la couche 2 pour contrôle d'accès au support. Celle-ci nous permet d'économiser de l'énergie en désactivant la radio le plus souvent (99% du temps). Tous les nœuds sont lancés avec une batterie pleine, fixée à $880mAh$ de charge, et qui représente 100% d'énergie. L'ordinateur utilisé pour les simulations est équipé d'un processeur Intel Dual Core XEON de 3.2 Ghz, d'une mémoire de 8 Go et tourne sous le système d'exploitation Ubuntu 11.10. La figure 4.6 donne un aperçu du réseau. Les flèches représentent les choix de prochain saut effectués par les capteurs, sur la base de la capacité énergétique résiduelle des nœuds se trouvant sur le chemin menant à la racine. Si l'on considère le nœud 7 (mis en évidence), la région verte correspond au domaine de transmission/réception et la zone grise à la zone d'interférence. Les inscriptions sous les nœuds entourés indiquent la qualité du lien radio avec 7.

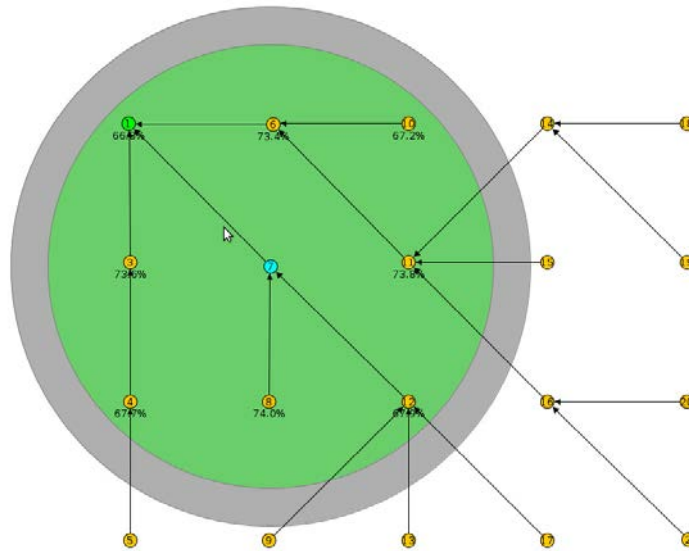


FIGURE 4.6 – Aperçu de la topologie réseau

4.3.2 Répartition énergétique

Pour obtenir un profil énergétique significatif des nœuds, nous avons simulé 13 jours d'activité (réalisé en 30 jours réels sur notre machine). La durée de vie du réseau est la date à laquelle le premier nœud du réseau a complètement épuisé sa batterie [DD09]. Dans les deux scénarios (routage par l'énergie et routage par l'ETX), les nœuds génèrent des données applicatives d'une taille de 87 octets, à diverses fréquences exprimées en nombre de paquets applicatifs par minute (*pkts/min*). Ces données sont envoyées vers la racine. Un routage prenant en compte l'énergie aura tendance à utiliser les nœuds qui en sont le plus dotés, ceux-ci épuiseront plus rapidement leur batterie et deviendront par la suite moins attractifs pour relayer les informations. Le réseau devra alors s'organiser pour sélectionner des nœuds plus intéressants. Ceci devra conduire à une répartition assez équilibrée de l'énergie sur l'ensemble des nœuds.

Les résultats des simulations illustrés par la figure 4.7 indiquent les proportions de nœuds dans le réseau avec leur pourcentage de batterie restante à l'arrêt de l'expérience. Nous observons sur le schéma 4.7a dont les nœuds émettent des données à 1pkt/min , 85% de nœuds ont leur batterie dans l'intervalle $[54\% - 56\%]$ pour le routage par l'énergie, tandis que le routage avec l'ETX répartit inégalement l'énergie sur l'ensemble des nœuds. Ce constat est beaucoup plus prononcé sur le schéma 4.7b où les nœuds émettent des données à une fréquence plus élevée (6pkts/min). L'intensité du trafic étant plus importante, les nœuds épuisent plus rapidement leur batterie, néanmoins le routage par l'énergie maintient un plus grand nombre de nœuds à un niveau de batterie plus important que dans le second scénario. Pour les deux intensités de trafic, le routage avec l'ETX possède toujours un plus grand nombre de nœuds ayant un niveau de batterie plus faible (environ 20%) que celui du routage par l'énergie. Ce dernier retarde d'avantage le(s) premier(s) nœud(s) qui épuisera(ont) leur énergie. Il s'agit d'un point capital, car l'intégrité du réseau est forcément affectée lorsque des

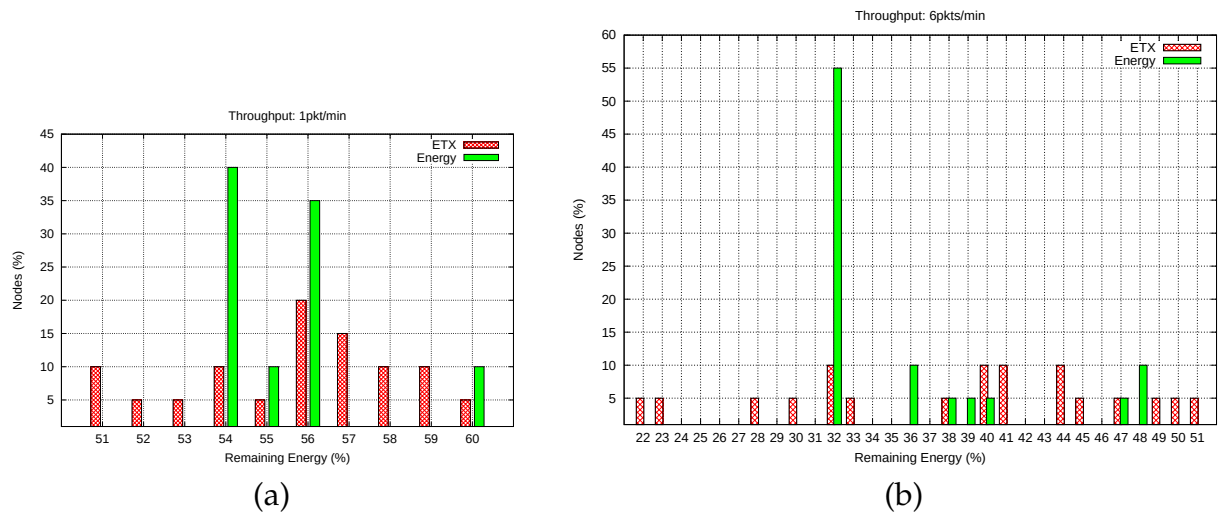


FIGURE 4.7 – Distribution de l'énergie résiduelle sur les nœuds

nœuds commencent à s'arrêter pour défaut d'énergie. Des scissions du réseau apparaîtront, impactant grandement sur la qualité du service rendu.

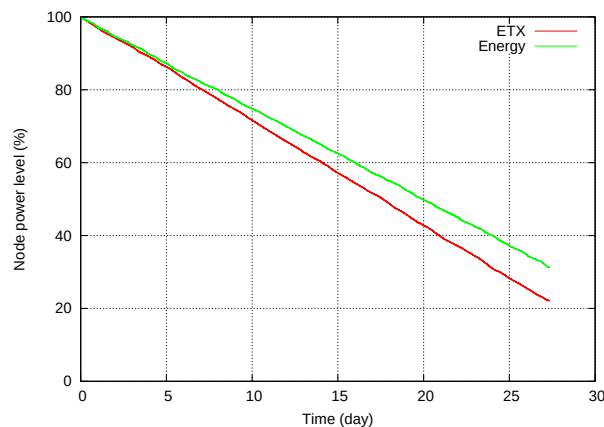


FIGURE 4.8 – Évolution énergétique des nœuds les plus sollicités

La figure 4.8 représente une estimation par régression linéaire de la date à laquelle le nœud le plus faible épuisera complètement son énergie. Elle se fonde sur les données collectées des 13 premiers jours de fonctionnement du réseau, ceci pour une fréquence d'émission de $6pkts/min$ dans les deux routages. Les résultats indiquent une durée de vie de 35 jours pour le scénario ETX et 40 jours pour le routage par l'énergie, impliquant une extension non négligeable de plus de 14% de durée de vie.

4.3.3 Fiabilité de transmission

Nous avons également évalué la fiabilité du réseau à livrer les paquets au point de collecte. Le routage avec l'ETX promeut les routes ayant un taux de livraison plus

important, tandis que le routage par l'énergie n'en tient pas compte. Il ne serait pas surprenant que le nombre de paquets reçus à la racine soit meilleur dans le scénario avec l'ETX que celui avec l'énergie. Ceci est confirmé par la figure 4.9 qui indique le nombre de paquets correctement reçus pour chaque fréquence d'émission. Le tableau 4.5 fait

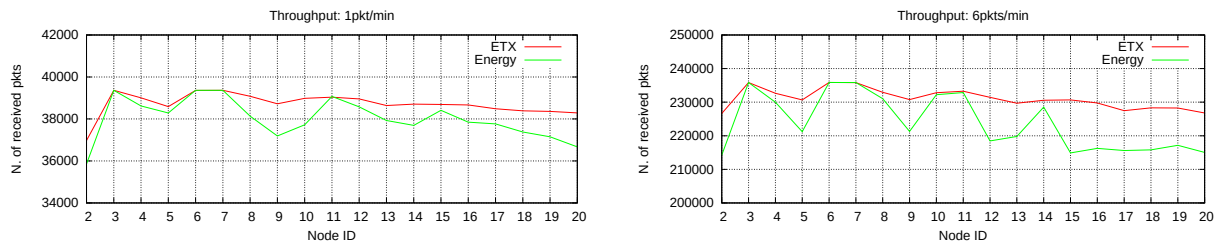


FIGURE 4.9 – Nombre de paquets reçus au point de collecte par nœud

un récapitulatif global de l'ensemble des paquets reçus par la racine, pour chaque fréquence d'émission. Nous constatons que les taux de livraison des paquets à la racine dans les deux scénarios est assez élevé, la différence entre ces taux étant faible. Bien que le routage avec l'ETX ait une meilleure performance, cette différence n'excède pas 3%.

Taux d'émission	Envoyés	Reçus (ETX)	Reçus (Énergie)
6 pkts/min	4 488 635	4 390 282 (97.80%)	4 251 962 (94.72%)
1 pkt/min	748 105	735 680 (98.34%)	722 394 (96.56%)

TABLE 4.5 – Fiabilité de livraison des paquets

4.4 Conclusion

Dans ce chapitre, nous avons conçu un routage pour les réseaux de capteurs sans fil prenant en compte l'énergie résiduelle des nœuds. L'implémentation de ce routage a été réalisée et intégrée à un système d'exploitation largement répandu dans les capteurs sans fil, Contiki. Les nœuds déterminent en temps réel leur capacité énergétique grâce à un modèle d'estimation d'énergie prenant en compte les processus électriques et chimiques à l'intérieur de la batterie. Le modèle intègre les effets non linéaires liés à la *récupération de charge* lors des phases d'inactivités. Nous avons montré que la métrique conçue assure les propriétés de consistance et d'absence de boucle pour le routage. Une évaluation et comparaison du routage RPL par l'énergie a été faite avec la version par défaut du standard utilisant l'ETX comme métrique. Les résultats nous indiquent une extension ($\approx 14\%$) de la durée de vie du réseau avec un faible dégradation (moins de 3%) du taux de livraison des paquets.

Bien que l'énergie soit un facteur primordial dans la conception de protocoles pour réseau de capteurs sans fil, d'autres critères, en fonction de l'application cible et de la qualité de service souhaitée doivent être considérés. Dans le prochain chapitre, nous étudions la possibilité de combiner l'énergie, avec d'autres métriques (ETX, délai, ...)

pour construire la topologie réseau, grâce à un routage qui puisse tenir dans la mémoire des capteurs et être réalisable malgré leur faible puissance de calcul.

Chapitre 5

Combinaison de métriques par la Logique Floue

Un routage prenant en compte la qualité de service (QoS) dans un réseau de capteurs sans fil, vise à considérer outre le facteur énergétique, d'autres paramètres de performance. Ces paramètres peuvent être : le délai de bout-en-bout, la gigue, le taux de perte de paquets, le débit de transmission, ... Plusieurs flux de données ayant diverses exigences de QoS peuvent se partager le même réseau. A titre d'illustration, une application pourrait envoyer périodiquement des relevés de données environnementales (température, humidité et luminosité), tout en réagissant à des événements soudains comme l'atteinte d'un seuil spécifique de pollution (gaz) ou la détection de présence dans la pièce. L'ensemble des services à mettre en place lors de la transmission du flux d'information de la source vers la destination (puits) consistera alors à réduire le délai de bout-en-bout en maintenant une gigue stable ; à minimiser la perte d'information dans le réseau tout en maximisant la durée de vie du réseau. Ces exigences dans le cadre d'un réseau de capteurs sont quelques fois conflictuelles. Par exemple, il serait possible d'améliorer la fiabilité en augmentant pour chaque nœud, sa puissance de transmission ou en exigeant autant de retransmissions que nécessaires par les couches basses de la pile protocolaire. Toutefois, ces deux mesures entraîneraient une dépense énergétique supplémentaire et par conséquent une dégradation de la durée de vie du réseau. Des compromis devront alors être faits entre l'efficacité énergétique et les autres considérations de QoS.

Dans ce chapitre, nous concevons et implémentons un moyen de réaliser ces compromis pendant l'exécution. Le problème général de recherche d'un chemin optimal multi-critères est connu comme NP-complet [RST10]. De nombreuses solutions et approches existent pour les réseaux classiques (filaires et sans fil Ad-hoc), mais celles-ci ne sont pas transposables aux capteurs sans fil qui présentent de fortes contraintes matérielles. Nous proposons la logique floue comme moyen de combinaison de plusieurs métriques. Celle-ci s'avère intéressante car elle nécessite une empreinte mémoire faible pour son implémentation. De plus, elle fournit un bon moyen de trouver un compromis intéressant entre plusieurs critères (même antagonistes). Nous intégrons notre approche dans le standard de routage pour les réseaux de capteurs RPL. La validation de celle-ci est faite à travers de nombreuses simulations [Kam+14] et des expériences sur des capteurs réels déployés en environnement intérieur [KND15].

5.1 Combinaison de métriques de routage RPL

La combinaison de plusieurs métriques de routage dans le but de prendre en considération la QoS dans le contexte du routage RPL a été récemment étudiée [Kar+13]. Celle-ci fait appel à deux types d'opérateurs de composition : la *composition additive* et la *composition lexicographique*.

5.1.1 Composition additive

Considérons m_1 et m_2 deux métriques associées respectivement aux structures algébriques $(\Sigma, \oplus, \omega_1, \preceq_1)$ et $(\Sigma, \oplus, \omega_2, \preceq_2)$. Avec ω_1 et ω_2 les fonctions de coût de chemin relatives à chacune des métriques m_1 et m_2 ; $\oplus$ l'opérateur de concaténation sur Σ , ensemble formé de tous les chemins possibles; $\preceq_1$ et $\preceq_2$ les relations d'ordre liées à m_1 et m_2 .

On appelle composition additive m de m_1 et m_2 , la métrique dont la structure algébrique associée est $(\Sigma, \oplus, \omega, \preceq_{\text{add}})$. La fonction de poids ω de la composition est définie par :

$$\forall \alpha \in \Sigma, \quad \omega(\alpha) = a_1 \cdot \omega_1(\alpha) + a_2 \cdot \omega_2(\alpha) \quad (5.1)$$

a_1 et a_2 sont des coefficients réels strictement positifs, permettant de pondérer la contribution de chacune des métriques m_1 et m_2 lors de la composition.

Théorème 1 (Stricte monotonie de la composition additive). *Si ω_1 et ω_2 sont des fonctions de poids de chemins pour des métriques de base m_1 et m_2 strictement monotones, dans lesquelles la relation d'ordre $\leq$ (inférieur ou égal à) est appliquée aux nombres réels, alors la métrique composite m définie sur la structure algébrique $(\Sigma, \oplus, \omega, \preceq_{\text{add}})$, avec ω défini par la formule 5.1 sur la relation d'ordre $\preceq_{\text{add}} \equiv \leq$ est également strictement monotone.*

i.e. $\forall \alpha, \beta \in \Sigma, \quad \omega(\alpha) < \omega(\alpha \oplus \beta) \text{ et } \omega(\alpha) < \omega(\beta \oplus \alpha)$

Démonstration.

Puisque $a_1 > 0$ et que ω_1 est strictement monotone, on a :

$$\omega_1(\alpha) < \omega_1(\alpha \oplus \beta) \Rightarrow a_1 \cdot \omega_1(\alpha) < a_1 \cdot \omega_1(\alpha \oplus \beta) \quad (5.2)$$

De même, puisque $a_2 > 0$ et que ω_2 est strictement monotone, nous avons :

$$\omega_2(\alpha) < \omega_2(\alpha \oplus \beta) \Rightarrow a_2 \cdot \omega_2(\alpha) < a_2 \cdot \omega_2(\alpha \oplus \beta) \quad (5.3)$$

Par sommation des deux inégalités 5.2 et 5.3, nous obtenons :

$$a_1 \cdot \omega_1(\alpha) + a_2 \cdot \omega_2(\alpha) < a_1 \cdot \omega_1(\alpha \oplus \beta) + a_2 \cdot \omega_2(\alpha \oplus \beta) \Rightarrow \omega(\alpha) < \omega(\alpha \oplus \beta) \quad (5.4)$$

De façon similaire, la monotonie à gauche peut être établie. $\square$

Théorème 2 (Stricte isotonie de la composition additive). *Si les deux structures algébriques $(\Sigma, \oplus, \omega_1, \preceq_1)$ et $(\Sigma, \oplus, \omega_2, \preceq_2)$ sont toutes strictement isotoniques et additives sur les métriques de base m_1 et m_2 (i.e. $\omega_1(\alpha \oplus \beta) = \omega_1(\alpha) + \omega_1(\beta)$ et $\omega_2(\alpha \oplus \beta) = \omega_2(\alpha) + \omega_2(\beta)$), alors leur composition additive $(\Sigma, \oplus, \omega, \preceq_{\text{add}})$ est également strictement isotonique.*

i.e. $\forall \alpha, \beta, \gamma \in \Sigma, \quad \omega(\alpha) < \omega(\beta) \Rightarrow [\omega(\alpha \oplus \gamma) < \omega(\beta \oplus \gamma)] \wedge [\omega(\gamma \oplus \alpha) < \omega(\gamma \oplus \beta)]$

Démonstration.

De la définition de l'additivité de composition des métriques, on peut écrire :

$$\begin{aligned}
 \omega(\alpha) < \omega(\beta) &\Rightarrow a_1 \cdot \omega_1(\alpha) + a_2 \cdot \omega_2(\alpha) < a_1 \cdot \omega_1(\beta) + a_2 \cdot \omega_2(\beta) \\
 &\Rightarrow a_1 \cdot \omega_1(\alpha) + a_2 \cdot \omega_2(\alpha) + a_1 \cdot \omega_1(\gamma) + a_2 \cdot \omega_2(\gamma) < a_1 \cdot \omega_1(\beta) + a_2 \cdot \omega_2(\beta) \\
 &\quad + a_1 \cdot \omega_1(\gamma) + a_2 \cdot \omega_2(\gamma) \\
 &\Rightarrow a_1 \cdot [\omega_1(\alpha) + \omega_1(\gamma)] + a_2 \cdot [\omega_2(\alpha) + \omega_2(\gamma)] < a_1 \cdot [\omega_1(\beta) + \omega_1(\gamma)] \\
 &\quad + a_2 \cdot [\omega_2(\beta) + \omega_2(\gamma)]
 \end{aligned}$$

$$\text{puisque } \begin{cases} \omega_1(\alpha \oplus \gamma) = \omega_1(\alpha) + \omega_1(\gamma) \\ \omega_1(\beta \oplus \gamma) = \omega_1(\beta) + \omega_1(\gamma) \\ \omega_2(\alpha \oplus \gamma) = \omega_2(\alpha) + \omega_2(\gamma) \\ \omega_2(\beta \oplus \gamma) = \omega_2(\beta) + \omega_2(\gamma) \end{cases}$$

$$\Rightarrow \omega(\alpha \oplus \gamma) < \omega(\beta \oplus \gamma)$$

De façon similaire, l'isotonie à gauche peut être établie. $\square$

Le principal inconvénient de cette approche est le fait que les métriques de base (m_1 et m_2) doivent nécessairement être définies sur la même relation d'ordre $\leq$ (i.e. $\preceq_1 \equiv \preceq_2 \equiv \leq$). Ceci limite grandement la classe de métriques pouvant être combinées avec cette approche. A titre d'illustration, les deux métriques, nombre de sauts et bande passante, ne sont pas combinables suivant cette approche. Si la première métrique est additive, la seconde (bande passante) ne l'est pas, mais est plutôt concave. Minimiser cette dernière reviendrait à rechercher la bande passante maximale sur le chemin. A l'opposé, une combinaison additive du nombre de sauts et du délai est possible.

5.1.2 Composition lexicographique

Soient les métriques m_1 et m_2 , décrites par les structures algébriques $(\Sigma, \oplus, \omega_1, \preceq_1)$ et $(\Sigma, \oplus, \omega_2, \preceq_2)$, respectives. Chacune des métriques de base faisant correspondre tout chemin de Σ à un ensemble de valeurs (V_1 et V_2 , respectivement). $\preceq_1$ est la relation d'ordre définie sur les valeurs de V_1 et $\preceq_2$ la relation d'ordre sur les valeurs de V_2 . On appelle composition lexicographique de m_1 et m_2 , la métrique m associée à la structure algébrique $(\Sigma, \oplus, \omega, \preceq_{\text{lex}})$. La relation d'ordre de la composition $\preceq_{\text{lex}}$ étant définie par :

$$\forall \alpha, \beta \in \Sigma, \quad \omega(\alpha) \preceq_{\text{lex}} \omega(\beta) \Leftrightarrow \begin{cases} \omega_1(\alpha) \prec_1 \omega_1(\beta) & \text{ou} \\ [\omega_1(\alpha) = \omega_1(\beta)] \wedge [\omega_2(\alpha) \preceq_2 \omega_2(\beta)] \end{cases} \quad (5.5)$$

NB : $\preceq$ est la relation d'ordre définie au sens large, tandis que $\prec$ est prise sens strict.

Autrement dit, lorsque plusieurs métriques sont combinées suivant une approche lexicographique, cela impose au routage de *prioriser* l'une des métriques de la composition par rapport aux autres. Cela revient à dire que, la route offrant le meilleur coût de chemin au regard de la première métrique est utilisée pour le transfert des paquets, indépendamment du coût associé aux autres métriques. La seconde métrique de la combinaison n'est utilisée que lorsqu'il existe plusieurs chemins de meilleurs coûts

pour la première métrique. Il faut noter que, contrairement à la combinaison additive, des métriques quelconques peuvent être utilisées, peu importe la relation d'ordre associée à chacune d'elle. Ceci du fait de la comparaison entre chemins qui se fait entre les coûts de la même métrique.

Pour que le routage construit sur la composition assure les propriétés de convergence et d'absence de boucle, il suffit que les métriques de base soient monotones et que la première soit strictement isotonique comme défini par les deux théorèmes suivants :

Théorème 3 (Monotonie de la composition lexicographique). *Si les deux structures algébriques $(\Sigma, \oplus, \omega_1, \preceq_1)$ et $(\Sigma, \oplus, \omega_2, \preceq_2)$ associées aux métriques de base respectives m_1 et m_2 , sont toutes deux monotones, alors la structure algébrique $(\Sigma, \oplus, \omega, \preceq_{lex})$ relative à la métrique composite m , telle que définie par la relation 5.5 est également monotone.*

Démonstration.

De la monotonie de la structure algébrique $(\Sigma, \oplus, \omega_1, \preceq_1)$ définie sur m_1 , on a :

$$\forall \alpha, \beta \in \Sigma; \quad \omega_1(\alpha) \preceq_1 \omega_1(\alpha \oplus \beta) \quad (5.6)$$

Deux situations sont possibles.

— **Cas 1 :** $\omega_1(\alpha) \prec_1 \omega_1(\alpha \oplus \beta)$

D'après la définition de la composition lexicographique donnée en 5.5, on en déduit que :

$$\omega(\alpha) \preceq_{lex} \omega(\alpha \oplus \beta) \quad (5.7)$$

— **Cas 2 :** $\omega_1(\alpha) = \omega_1(\alpha \oplus \beta)$

Puisque la seconde métrique est également monotone (i.e. $\omega_2(\alpha) \preceq_2 \omega_2(\alpha \oplus \beta)$), d'après 5.5 on conclut que $\omega(\alpha) \preceq_{lex} \omega(\alpha \oplus \beta)$

La monotonie à droite est ainsi établie. Un raisonnement similaire permet également d'établir la monotonie à gauche. $\square$

Remarque : Il convient de noter que la **stricte** monotonie de l'une ou l'autre des métriques (m_1 ou m_2) suffit pour également établir celle de la composition m .

Théorème 4 (Isotonie de la composition lexicographique). *Si les deux structures algébriques $(\Sigma, \oplus, \omega_1, \preceq_1)$ et $(\Sigma, \oplus, \omega_2, \preceq_2)$ associées aux métriques de base respectives m_1 et m_2 , sont toutes deux isotoniques, et si de plus $(\Sigma, \oplus, \omega_1, \preceq_1)$ est strictement isotonique, alors la structure algébrique $(\Sigma, \oplus, \omega, \preceq_{lex})$ relative à la métrique composite m , telle que définie par la relation 5.5 est également isotonique.*

Démonstration.

Pour toute paire arbitraire de chemin $\{\alpha, \beta\}$ satisfaisant la relation suivante :

$$\omega_1(\alpha) \preceq_1 \omega_1(\beta) \Leftrightarrow \begin{cases} \omega_1(\alpha) \prec_1 \omega_1(\beta) & \text{ou} \\ \omega_1(\alpha) = \omega_1(\beta) \end{cases} \quad (5.8)$$

La relation 5.8 est décomposable en 2 situations que nous détaillons ci-dessous :

— **Cas 1 :** $\omega_1(\alpha) \prec_1 \omega_1(\beta)$

Selon l'hypothèse du théorème 4, la première métrique est strictement isotonique :

$$i.e. \quad \forall \alpha, \beta, \gamma \in \Sigma \quad \omega_1(\alpha) \prec_1 \omega_1(\beta) \Rightarrow \omega_1(\alpha \oplus \gamma) \prec_1 \omega_1(\beta \oplus \gamma)$$

Par définition de la composition lexicographique (cf. relation 5.5), lorsque c'est l'inégalité stricte sur la métrique m_1 qui prévaut, on peut conclure que :

$$\forall \alpha, \beta, \gamma \in \Sigma \quad \omega(\alpha) \preceq_{lex} \omega(\beta) \Rightarrow \omega(\alpha \oplus \gamma) \preceq_{lex} \omega(\beta \oplus \gamma) \quad (5.9)$$

— **Cas 2 :** $\omega_1(\alpha) = \omega_1(\beta)$

D'après 5.5 on peut récrire :

$$\omega(\alpha) \preceq_{lex} \omega(\beta) \Leftrightarrow [\omega_1(\alpha) = \omega_1(\beta)] \wedge [\omega_2(\alpha) \preceq_2 \omega_2(\beta)]$$

Comme la seconde métrique est isotonique,

$$i.e. \quad \forall \alpha, \beta, \gamma \in \Sigma \quad \omega_2(\alpha) \preceq_2 \omega_2(\beta) \Rightarrow \omega_2(\alpha \oplus \gamma) \preceq_2 \omega_2(\beta \oplus \gamma)$$

On peut en déduire que la composition lexicographique telle que définie en 5.5 l'est aussi (cf. relation 5.9).

L'isotonie à gauche est établie par un raisonnement similaire. □

5.2 Système d'inférence par la logique floue

Le système de raisonnement par la logique floue nous permet de transformer plusieurs métriques d'entrée en une unique valeur en sortie. Pour ce faire, nous procédons en trois étapes : une première étape dite de *fuzzification*, suivie par une étape constituée de deux phases : l'*inférence* puis l'*agrégation*. Nous terminons par l'étape de *défuzzification*. De par sa simplicité et son efficacité, nous utilisons le modèle d'inférence floue de Mamdani [Mam77]. Ce dernier utilise des opérations arithmétiques de base (maximum, minimum, produit, ...) comme opérateurs de combinaison et d'agrégat. Cela se révèle judicieux au regard des contraintes matérielles (mémoire centrale, processeur) liées à notre environnement de mise en œuvre.

5.2.1 Variables linguistiques

Les variables linguistiques utilisées en entrée de notre moteur d'inférence sont récupérées des couches basses (physique et MAC) par le nœud ou issues de la surveillance de la qualité du lien de communication. Elles correspondent aux paramètres de performance ou métriques à optimiser lors de la construction de la topologie réseau. Chaque variable linguistique sera décomposée en un certain nombre d'ensembles dit *flous* décrivant celle-ci. La valeur exacte de la variable linguistique pour chacun des *ensembles flous* définis est déterminée par la fonction d'appartenance. Contrairement à la logique dite « classique » où la variable booléenne ne peut prendre qu'une des valeurs de l'ensemble $\{\text{vrai}, \text{faux}\}$, la logique floue admet pour la variable linguistique (plus

précisément, de chacun des *ensembles flous* la décrivant) une infinité de valeurs entre ces deux bornes (*i.e.* une valeur réelle dans $[0, 1]$).

A titre d'illustration, supposons que la capacité énergétique d'un nœud soit indiquée par un entier allant de 0 (batterie vide) à 255 (batterie pleine). En logique floue, il sera possible de décrire la variable linguistique « énergie du nœud » par des *ensembles flous*, *plein* et *vide*. Un nœud dont la capacité énergétique de la batterie vaut 200 pourra alors, selon la sémantique donnée à chacun des *ensembles flous* *plein* et *vide* grâce à leur fonction d'appartenance respective, être évalué à 78.50% *plein* et à 21.50% *vide*.

Dans le modèle proposé, les métriques choisies (ou variable linguistiques au sens de la logique floue) sont l'énergie résiduelle du nœud, le délai et l'ETX. Pour illustrer le principe du moteur d'inférence conçu, considérons le schéma de la figure 5.1.

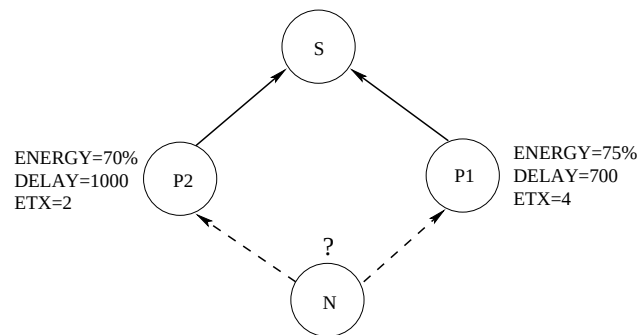


FIGURE 5.1 – Processus de sélection du prochain saut

Le nœud N souhaite envoyer des paquets au point de collecte S , sur la base des métriques énergie, délai et ETX. Nous souhaitons déterminer le prochain saut le plus adapté sur la base de ces paramètres. Il est à noter que, dans une approche de composition lexicographique de ces trois paramètres, la topologie réseau serait construite en priorité sur la base de la première métrique choisie pour la combinaison. Pour ce qui est de la combinaison additive de ces métriques, elle est impossible en l'état. En effet, si les métriques *délai* et *ETX* sont par nature additives, l'énergie ne l'est pas. Nous visons à combiner les métriques d'entrée et calculer par la logique floue un score en sortie, qui nous renseigne sur le parent le plus adapté pour le relai des paquets.

Pour récupérer l'ETX et le délai des couches MAC et réseau, nous utilisons un mécanisme de *cross-layer*. L'ETX mesure la qualité de la transmission entre un nœud et son voisin. Elle est mise en œuvre en comptant le nombre de transmissions nécessaires pour qu'un paquet soit correctement reçu et acquitté par le voisin. Une valeur faible implique des liens fiables et offre un taux de livraison de paquets plus important vers la destination. La couche réseau place le paquet dans le tampon de sortie et fait appel à la couche MAC pour la livraison. Cette dernière transmet le paquet dès que le support est disponible, vérifie par la suite le statut de la livraison. En cas d'échec de transmission, le processus est repris selon la valeur du paramètre `MAX_MAC_RETRANSMISSION` (fixé à 5 dans l'implémentation réalisée). Pour atténuer l'impact des perturbations ponctuelles ou fluctuations transitoires sur la métrique, celle-ci

est calculée comme une moyenne glissante exponentielle au cours du temps.

$$ETX_{new} = \alpha * ETX_{old} + (1 - \alpha) * ETX_{cur} \quad (5.10)$$

où $\alpha \in [0, 1[$ est la pondération caractérisant l'importance accordée à ETX_{old} qui décrit les différentes valeurs d'ETX déjà calculées dans le passé et ETX_{cur} la valeur de l'ETX issue de la transmission du dernier paquet correctement transmis au prochain saut.

Si la métrique précédente mesure la fiabilité du canal de communication, le délai quant à lui, nous permet de capturer l'impact de la couche MAC et de la contention sur les transmissions. Nous éviterons ainsi les goulots d'étranglement du réseau. Il est également calculé comme une moyenne glissante exponentielle. L'énergie résiduelle du nœud est estimée suivant le modèle défini en §4.1.2.

5.2.2 Fuzzification

La première phase du système de contrôle par la logique floue consiste en une fuzzification. L'idée est de déterminer le niveau d'appartenance (sur une échelle allant de 0 à 1) de l'entrée scalaire aux différents *ensembles flous* de la variable linguistique considérée. Elle implique les fonctions suivantes :

- Récupérer en entrée les valeurs scalaires mesurées de la variable linguistique,
- Définir les fonctions de correspondance ou fonctions d'appartenance transformant les données scalaires d'entrée en éléments de l'univers du discours (encore appelés *ensembles flous*),
- Réaliser la fonction de fuzzification qui calcule, pour chaque valeur de la variable linguistique, son degré d'appartenance aux *ensembles flous*.

Pour illustrer le modèle, la variable linguistique utilisée pour le délai est représentée par trois *ensembles flous* : *short*, *average* et *long*. Ils décrivent respectivement le degré d'appartenance de la donnée scalaire d'entrée *délai*, à une valeur faible, moyenne ou grande. Sur la figure 5.2, avec un délai égal à 700ms, les projections réalisées sur les ensembles flous nous indiquent une valeur faible à 83.33% et moyenne à 16.66%. Il faut noter que la projection sur l'ensemble flou *long* n'est pas définie, ceci indique un délai grand à 0%.

Les formules 5.11 et 5.12 correspondent respectivement aux fonctions d'appartenances de délai-faible et délai-moyen. Elles renvoient la valeur numérique du degré d'appartenance à chacun des *ensembles flous*, *short* et *average*. Dans le même ordre d'idée, une formule semblable établie la valeur de la fonction d'appartenance de délai-long sur les différents intervalles définis.

$$\mu_{short}(délai) = \begin{cases} 1 & \text{si } délai \in [0, 600ms] \\ \frac{1200 - délai}{600} & \text{si } délai \in]600ms, 1200ms[\\ 0 & \text{si } délai \in [1200ms, +\infty[\end{cases} \quad (5.11)$$

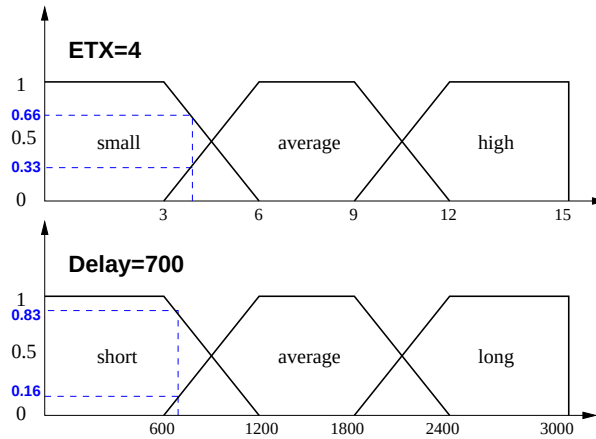


FIGURE 5.2 – Fonctions d'appartenance et ensembles flous

$$\mu_{average}(delai) = \begin{cases} 0 & \text{si } delai \in [0, 600ms] \\ \frac{delai-600}{1200-600} & \text{si } delai \in]600ms, 1200ms[\\ 1 & \text{si } delai \in [1200ms, 1800ms] \\ \frac{delai-1800}{1800-2400} & \text{si } delai \in]1800ms, 2400ms[\\ 0 & \text{si } delai \in [2400ms, +\infty[\end{cases} \quad (5.12)$$

La variable linguistique ETX est représentée par trois *ensembles flous* : small, average et high, correspondant respectivement à une grandeur petite, moyenne ou élevée. Avec ETX = 4, un raisonnement similaire à celui du délai nous conduit à trouver pour les *ensembles flous* small, average et high, les degrés d'appartenance respectifs 66.66%, 33.33% et 0%.

5.2.3 Inférence et agrégation

Les moteurs d'inférence sont au cœur des systèmes à base de raisonnement flou. Ils simulent la prise de décision comme le ferait un humain, en s'appuyant sur une *base de règles* établies suivant les connaissances d'un expert. Un moteur d'inférence passe par :

- La définition de toutes les règles indiquant comment calculer la variable linguistique de sortie à partir des entrées,
- L'exécution de toutes les règles applicables,
- L'agrégation des résultats lorsque les entrées activent plus d'une règle dans la base.

Considérons le système d'inférence par la logique floue mis en œuvre sur la base des variables linguistiques choisies : ETX, énergie et délai. Nous définissons la variable linguistique intermédiaire QdS, qui renseigne sur la qualité de service. Cette sortie est déduite des deux variables linguistiques ETX et délai. Le tableau 5.1 résume la relation entre ces deux variables d'entrée et la sortie QdS. Elle est décrite par les *ensembles flous*

représentant une qualité de service allant de *very_fast* (pour une QoS de très bonne qualité) à *very_slow* (pour une très mauvaise QoS). Le tableau est basé sur l'idée suivante : « plus faible est le délai et petite est l'ETX, meilleure est la QoS ». Nous considérons également qu'un délai moyen et une valeur d'ETX petite produit la même QoS qu'un faible délai et une valeur moyenne d'ETX (en l'occurrence « *fast* QoS »). Pour

ETX / Delay	short	average	long
small	very_fast	fast	moderate
average	fast	moderate	slow
high	moderate	slow	very_slow

TABLE 5.1 – Variable linguistique de sortie QoS

produire la sortie, nous nous servons du modèle d'inférence de Mamdani [Mam77], et utilisons l'opérateur *minimum* comme fonction de composition et le *maximum* comme opérateur d'agrégation. A cet effet, la formule 5.13 indique comment obtenir le degré d'appartenance à l'ensemble flou *moderate* à partir des variables linguistiques d'entrées. Comme le montre le tableau 5.1, cet ensemble est représenté par trois règles, d'où leur agrégation par l'opérateur choisi.

$$\mu_{moderate}(QoS) = \max \left(\begin{array}{l} \min(\mu_{high}(ETX), \mu_{short}(delai)) \\ \min(\mu_{avg}(ETX), \mu_{avg}(delai)) \\ \min(\mu_{small}(ETX), \mu_{long}(delai)) \end{array} \right) \quad (5.13)$$

Reprenant l'exemple de la figure 5.1, sur la base des métriques reçues de son parent P_1 , le nœud N calculera trois fonctions d'appartenance non nulles de QoS : $\mu_{very_fast}(QoS) = 66\%$; $\mu_{fast}(QoS) = 33\%$ et $\mu_{moderate}(QoS) = 16\%$. Partant du schéma 5.2, les entrées produisent quatre combinaisons possibles. La figure 5.3 illustre comment les trois valeurs de sortie pour la QoS en sont déduites, par application des opérateurs de composition et d'agrégation. Il faut noter que, l'ensemble *fast* QoS étant produit plus d'une fois, c'est la plus grande valeur qui est retenue. La phase de defuzzification présentée au prochain paragraphe, nous permet d'obtenir une unique valeur scalaire décrivant la QoS. Dans le cas d'espèce, 78% est obtenue.

ETX \ Delay	μ_{short} 0.83	μ_{avg} 0.16	μ_{long}
μ_{small} 0.66	very_fast 0.66	fast 0.16	moderate
μ_{avg} 0.33	fast 0.33	moderate 0.16	slow
μ_{high}	moderate	slow	very_slow

FIGURE 5.3 – Composition et Agrégation pour la QoS

5.2.4 Defuzzification

C'est la dernière phase d'un système de contrôle à base de la logique floue. Elle permet de produire une valeur scalaire en sortie à partir du degré d'appartenance des entrées aux *ensembles flous* définis pour les variables linguistiques. Toutes les valeurs floues obtenues après la phase d'inférence et d'agrégation sont unifiées en une seule. Plusieurs opérateurs de defuzzification existent. Nous optons dans notre modèle pour le centre de gravité qui a l'avantage de prendre en compte les contributions de tous les ensembles flous calculés à l'étape précédente, chacun étant pondéré par le degré d'appartenance des entrées à ceux-ci. Formellement, la valeur numérique en sortie est obtenue par la relation 5.14.

$$\alpha_0 = \frac{\sum_{i=1}^k (\alpha_i \times \mu_C(\alpha_i))}{\sum_{i=1}^k \mu_C(\alpha_i)} \quad (5.14)$$

α_0 est l'abscisse du centre de gravité du polygone délimité par les valeurs de la variable linguistique de sortie (cf. figure 5.4). k représente le nombre de règles activées dans le moteur d'inférence, α_i le domaine de valeur relatif à la i^{ime} règle et $\mu_C(\alpha_i)$ le degré d'appartenance des entrées à l'ensemble flou C .

La région mise en évidence dans la figure 5.4 définit l'aire dont l'abscisse du centre de gravité est la valeur « défuzzifiée » de QdS pour les trois valeurs floues 66%, 33% et 16%.

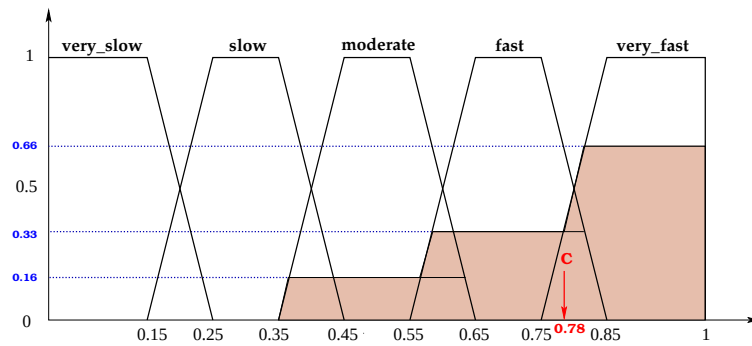


FIGURE 5.4 – Defuzzification de la QdS

5.3 Fonction d'objectif RPL basée sur la logique floue

5.3.1 Architecture du moteur d'inférence

Combiner directement les trois variables linguistiques d'entrée en une seule sortie, entraînerait la gestion de $3^3 = 27$ possibilités de combinaisons des différentes fonctions d'appartenance du moteur d'inférence. Pour éviter la complexité liée à cette gestion, nous organisons le système d'inférence en deux blocs fonctionnels comme illustré par la figure 5.5. Dans un premier temps, l'ETX et le délai sont combinés pour produire la variable linguistique QdS. Par la suite, cette dernière est combinée en deuxième étape

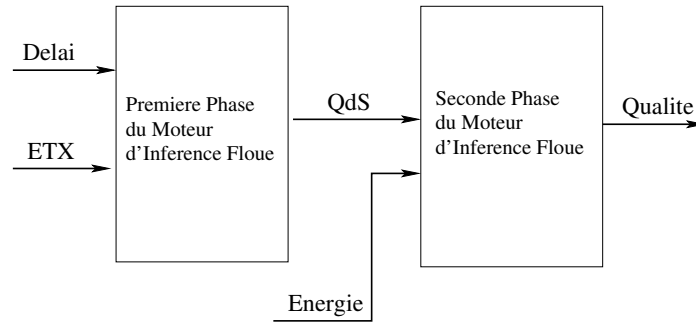


FIGURE 5.5 – Moteur d’inférence floue

avec l’énergie pour produire une unique sortie QUALITÉ. Cette dernière variable linguistique renseigne sur l’aptitude d’un nœud à servir de relai. La variable énergie peut appartenir suivant la valeur de l’entrée scalaire aux *ensembles flous* : *low*, *medium* et *full* (pour une batterie faible, moyenne ou pleine respectivement). La variable finale de sortie QUALITÉ quant à elle varie de *awfull* à *excellent* : pour un voisin de piètre qualité à voisin très recommandé, respectivement. La valeur scalaire *défuzzifiée* en sortie du moteur d’inférence varie elle de 0 à 100. Le tableau 5.2 montre comment nous calculons la sortie à partir des variables linguistiques d’entrée QdS et énergie.

QdS / Énergie	low	medium	full
very_slow	awful	bad	average
slow	bad	degraded	average
moderate	degraded	average	acceptable
fast	average	acceptable	good
very_fast	average	good	excellent

TABLE 5.2 – Variable linguistique de sortie : QUALITÉ

Pour l’exemple illustratif de la figure 5.1, nous obtiendrons alors en entrée du deuxième bloc de notre architecture les valeurs scalaires : $QdS = 78\%$ et $\text{Énergie} = 75\%$. Après la phase de *fuzzification*, elles activent dans le moteur d’inférence les règles correspondant aux trois ensembles flous QUALITÉ suivants : $\mu_{\text{acceptable}}(\text{QUALITÉ}) = 25\%$, $\mu_{\text{good}}(\text{QUALITÉ}) = 70\%$ et $\mu_{\text{excellent}}(\text{QUALITÉ}) = 30\%$. La phase de *defuzzification* les unifie en $\text{QUALITÉ} = 77\%$. Cette valeur correspond à l’aptitude *calculée* du parent P_1 à servir de prochain saut pour le nœud N au vu des métriques $\text{Énergie} = 75\%$, $\text{Délai} = 700ms$ et $\text{ETX} = 4$.

Un raisonnement similaire nous produit la valeur $\text{QUALITÉ} = 70\%$ pour les données issues de P_2 . Nous en concluons que le nœud « *recommandé* » pour servir de prochain saut à N , au regard de la combinaison à base de la logique floue proposée, est P_1 . Il faut noter qu’une utilisation de la combinaison lexicographique de ETX , délai et énergie , aurait conduit à sélectionner P_2 comme prochain saut. En effet, il exhibe le meilleur coût sur la métrique ETX . Par ailleurs, la combinaison additive de ces trois critères est impossible du fait de la métrique énergie utilisée.

5.3.2 Intégration à RPL

5.3.2.1 Logique floue

Nous avons pris soin d'implémenter le modèle à base de la logique proposé dans Contiki, en définissant une nouvelle fonction d'objectif intégrée au protocole de routage RPL. Tenant compte des contraintes d'espace mémoire et dans le but d'optimiser son utilisation, les fonctions d'appartenance (toutes des fonctions affines) ont été implémentées comme des macros. Les bornes de ces fonctions sont quant à elles spécifiées par des constantes (*cf.* script). Toutes les définitions sont faites dans "fuzzify.h", réutilisées par la suite dans la bibliothèque "fuzzify.c" codant toutes les fonctions relatives à la logique floue. Le script ci-dessous correspond aux définitions relatives à la variable linguistique énergie :

```
1 #define ENERGY_MAX 255
2 #define ENG_B1 ENERGY_MAX / 5
3 #define ENG_B2 ENG_B1 * 2
4 #define ENG_B3 ENG_B1 * 3
5 #define ENG_B4 ENG_B1 * 4
6 #define energy_low(eng) FIRST_T_NORM(ENG_B1, ENG_B2, (eng))
7 #define energy_medium(eng) T_NORM(ENG_B1, ENG_B2, ENG_B3, ENG_B4, (eng))
8 #define energy_full(eng) LAST_T_NORM(ENG_B3, ENG_B4, (eng))
```

La ligne n°1 représente l'intervalle des valeurs de la variable linguistique énergie (abrégée *nrg*). Il s'agit d'un entier compris entre 0 et 255. La ligne n°2 partitionne cet intervalle en quatre parties égales qui nous serviront par la suite de bornes (lignes 3 à 5) pour les T-Normes (Trapezoidal Norm) représentant les fonctions d'appartenance. Ces fonctions sont elles mêmes définies par les macros des lignes 6 à 8, et nous renvoient la valeur du degré d'appartenance de l'énergie scalaire brute à chacun des *ensembles flous* décrivant l'énergie. Des définitions similaires sont faites pour tous les ensembles flous.

5.3.2.2 Structure de données pour la topologie de routage

Sous Contiki, la topologie de routage est gérée par un ensemble de structures de données complexes, liées les unes aux autres selon la logique représentée sur la figure 5.6. *rpl_instance* regroupe les informations sur l'instance RPL, parmi lesquelles un pointeur sur l'OF mise en œuvre : *rpl_of_fuzzy*, un pointeur sur *rpl_metric_container*, structure contenant les valeurs mesurées de chacune des métriques utilisées (énergie, délai, ETX, nombre de sauts) et un autre pointeur sur la structure du DAG : *rpl_dag*. Cette dernière structure pointe elle même sur la liste des parents potentiels du nœud, dont l'un en particulier correspond au prochain saut (*best_parent*). La fonction d'objectif est quant à elle représentée par une structure dont tous les champs sont des pointeurs sur des fonctions qui lorsqu'elles sont invoquées permettront : de mettre à jour la valeur des métriques (*update_metric_container*), de déterminer le prochain saut (*best_parent*), le meilleur DAG (*best_dag*) ou de calculer le rang du nœud (*calculate_rank*) à partir de celle de son parent et de la valeur des métriques. Les routines de l'OF sont invoquées selon que certains événements se produisent dans le réseau : expiration d'un minuteur

(évaluation périodique de l'énergie), changement de la qualité du lien (délai, ETX), etc. Si les conditions sont remplies, les paramètres de qualité du prochain saut sont alors comparées à ceux de l'ensemble des parents potentiels du nœud en vue de sélectionner un nouveau candidat comme « *parent préféré* ».

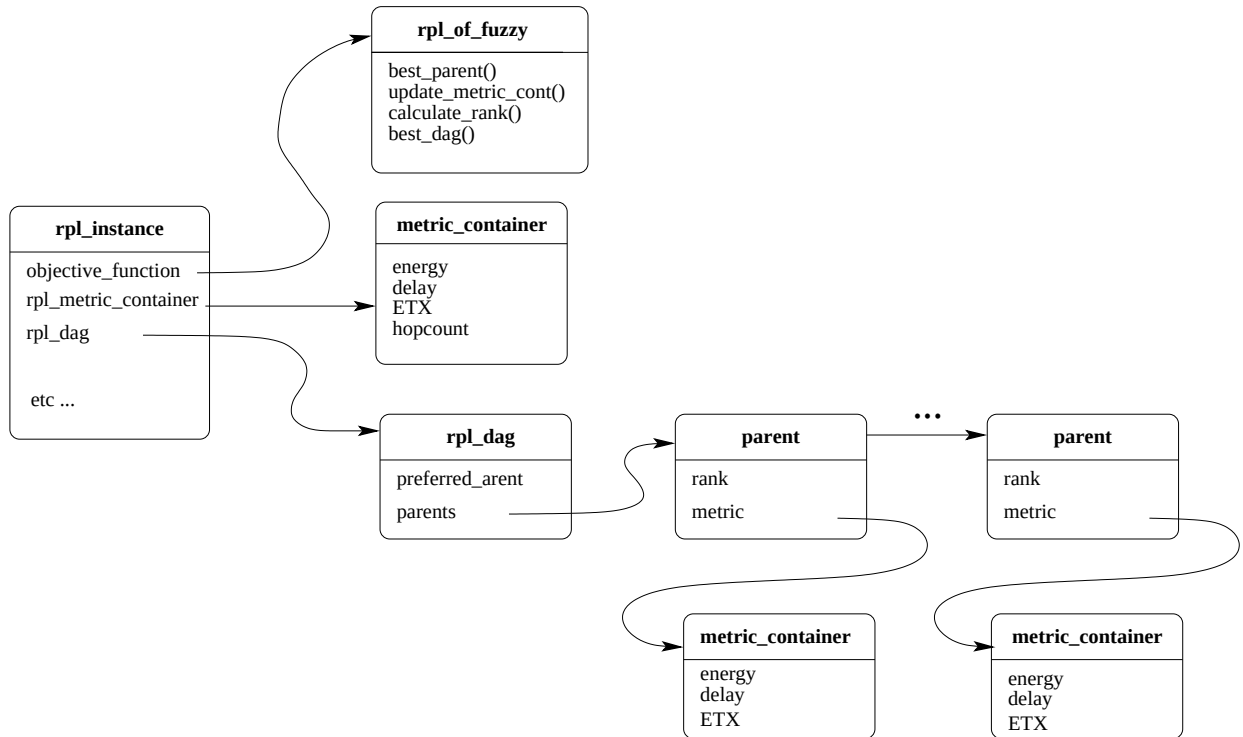


FIGURE 5.6 – Architecture des structures de données pour la composition

5.3.2.3 Annonces de routage

Les annonces contenant les informations de configuration pour le routage sont transportées par les DIO. Nous nous conformons aux recommandations de la norme [Win+12]. La figure 5.7 illustre la structure de l'annonce utilisée lorsque celle-ci transporte des informations d'options de type 2 encore appelée « *conteneur de métrique* ». Les différentes métriques associées à un chemin donné sont transportées dans le *conteneur* les unes derrière les autres, lorsque la valeur du champ type de métrique (Routing-MC-Type) est définie sur RPL_DAG_MC_FUZZY (spécifique à notre implémentation). Ceci nous permet de maintenir la taille du DIO aussi petite que possible, afin d'éviter la fragmentation des trames (la couche physique ne supportant que 127 octet maximum). Comparé à un DIO transportant la seule métrique ETX [GL12], 6 octets sont rajoutés. Ceci représente une surcharge de 20% pour un DIO ne transportant que des métriques (option type 2).

5.3.2.4 Calcul de rang

Nous construisons le rang relatif à la composition des métriques par la logique floue, de sorte que celle-ci exhibe les propriétés de monotonie et d'isotonie. Ceci nous permet d'éviter les boucles de routage et d'assurer la convergence du routage. Le nœud

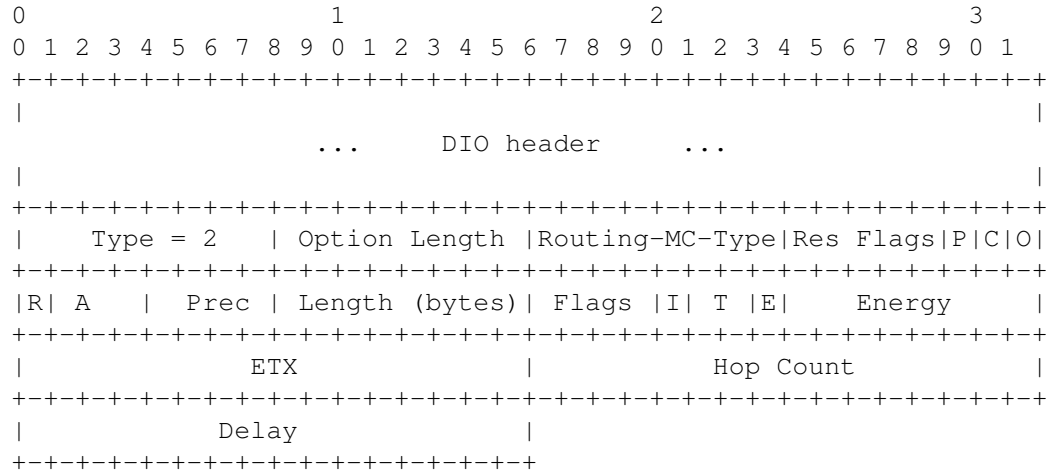


FIGURE 5.7 – DIO transportant l’ensemble des métriques

n’installe dans sa liste de potentiels parents, que les voisins ayant un rang strictement inférieur au sien. Par la suite, il recherche parmi les potentiels parents, celui qui présente le meilleur coût de chemin au regard de la combinaison. Ce dernier est alors promu « *parent préféré* ». Le nœud met à jour son rang en ajoutant à celui de son *parent préféré* un incrément strictement positif. Cette quantité est fonction de la « métrique floue » calculée par la combinaison comme indiqué par la formule 5.15, où N est le nœud considéré et P son parent. Il convient de noter que le nœud devient ainsi inéligible pour tous ses parents potentiels possédant un rang inférieur ou égal au sien.

$$Rank(N) = Rank(P) + Rank_{incr}$$

avec $Rank_{incr} = MinHopRankIncrease + \alpha \cdot [QUALITE_{max} - fuzzy_{metric}(P)]$ (5.15)

Dans la formule, $MinHopRankIncrease$ correspond à un point d’augmentation du rang. $fuzzy_{metric}(P)$ est la valeur de sortie calculée par le moteur d’inférence floue, $QUALITE_{max}$ étant sa valeur maximale. α nous permet de pondérer l’accroissement de rang par rapport à la perte de qualité. Nous l’avons fixé à $\frac{1}{10}MinHopRankIncrease$ à l’issue de plusieurs tests.

5.4 Expérimentations et évaluation

Nous avons dans un premier temps réalisé des simulations du modèle proposé sous Cooja avec des nœuds équipés du système d’exploitation Contiki v2.7. Par la suite, nous avons déployé un réel réseau de capteurs sans fil en environnement intérieur (bâtiment). Nous avons pris soin de vérifier la nécessité du routage par des communications multi-sauts, car les transmissions des nœuds ayant des difficultés à traverser plus de trois cloisons. Deux scénarios ont été développés. Le premier organisant le routage RPL selon les trois métriques : ETX, délai et énergie combiné en utilisant la logique floue comme décrit plus haut ; le second est l’implémentation par défaut qui ne prend en compte que l’ETX pour unique métrique [GL12]. L’application développée est la même pour tous les scénarios : les nœuds captent les informations sur l’état de

l'environnement (température, humidité, luminosité) ou des paramètres de fonctionnement du réseau (nombre de voisins, qualité du lien radio, prochain saut, etc.) et les envoient périodiquement au point de collecte.

5.4.1 Conditions expérimentales et outils

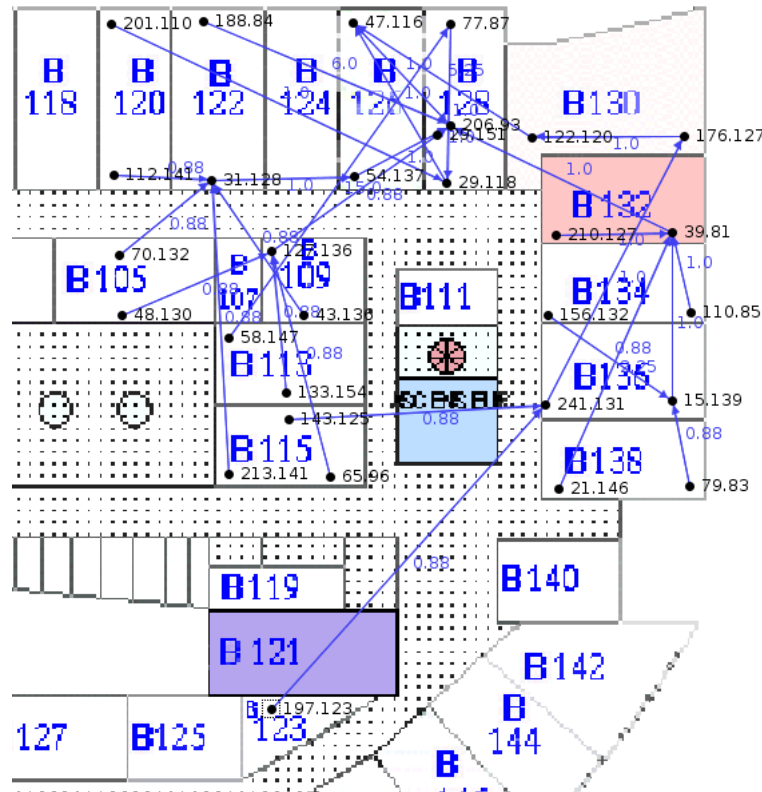


FIGURE 5.8 – Déploiement des nœuds dans le bâtiment

Le plan illustré par la figure 5.8 représente le déploiement réalisé. 28 nœuds de type TelosB MTM-CM5000-MSP ont été répartis dans 14 bureaux (2 par bureau). Les étiquettes correspondent aux adresses physiques des nœuds et nous servent d'identifiant. Les liens (en bleu) indiquent le choix de prochain saut effectué par les nœuds à l'instant capturé. Chaque nœud est équipé d'un microcontrôleur Texas Instrument MPS430 16 bit, d'une puce radio CC2420, de divers capteurs environnementaux, d'un interface USB et embarque deux piles au format AA. Les nœuds sont maintenus fixes tout au long de l'expérience exécutée pendant 2 semaines. Les données applicatives sont envoyées par les nœuds toutes les 5 minutes. Le point de collecte est un nœud connecté via son interface USB à un ordinateur servant de passerelle. Ce nœud est considéré comme ayant une niveau d'énergie maximal, car alimenté en permanence. Le tableau 5.3 résume les protocoles utilisés.

5.4.2 Taux de livraison

Nous avons dans un premier temps évalué la fiabilité de l'application à livrer les informations au point de collecte. Comme le montre la figure 5.9, le scénario combinant

Couche	Protocole
Application	collecte périodique de données
Transport	UDP
Réseau	μ IPv6 + 6LoWPAN + RPL (routage)
sous-couche MAC	CSMA non-persistant
Cycle d'utilisation Radio (RDC)	contikiMAC
PHY + puce Radio	IEEE 802.15.4 & CC2420

TABLE 5.3 – Pile protocolaire

les trois métriques par la logique floue offre de meilleurs résultats que la version RPL avec l'OF-ETX. Alors que le premier stabilise le taux de perte de paquets à 5%, dans le second scénario cette valeur est trois fois supérieure. Lorsque nous examinons les premiers moments de l'expérience (correspondant à la phase démarrage du réseau), nous notons que la combinaison de métriques se comporte mieux. En effet, tandis qu'elle maintient le taux de perte de paquets à un niveau faible, le routage avec ETX présente pendant les trois premiers jours un fort taux de perte de paquets (autour de 25%). Ce comportement du routage par l'ETX s'explique par sa lenteur à accéder à un état stable lorsque des routes non fiables ont été initialement sélectionnées.

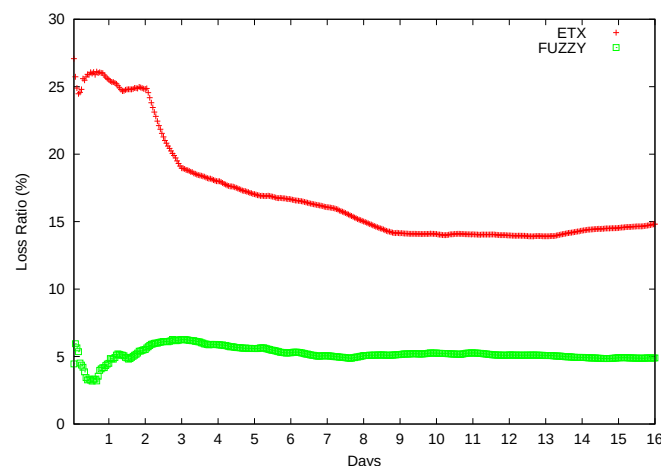


FIGURE 5.9 – Taux de perte de paquets au point de collecte

5.4.3 Stabilité de routage

Nous avons examiné de plus près la stabilité du routage. Cette mesure de performances nous permet d'évaluer le nombre de changements de prochain saut dans le temps. Un grand nombre de changements est révélateur d'une topologie instable, chose indésirable car entraînant le phénomène de *battement de route* et susceptible d'influencer le taux de livraison des paquets. La figure 5.10 présente pour chacun des scénarios, le nombre de changements de parents par heure. Nous mettons également en évidence la tendance principale de ces variations par régression linéaire des données de changement de parents collectées sur une quinzaine de jours.

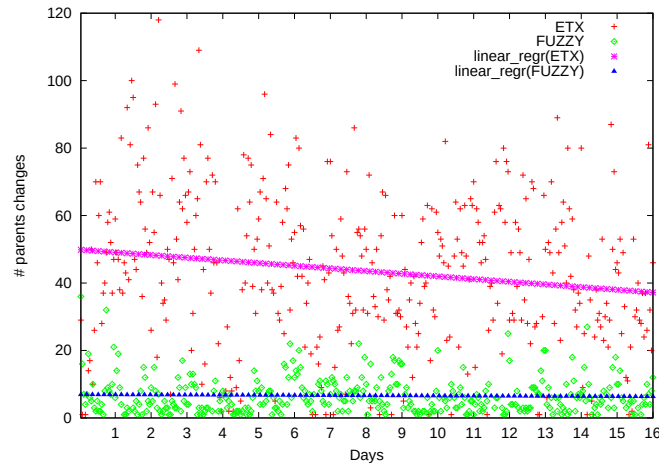


FIGURE 5.10 – Nombre de changement de prochain saut dans le temps

Nous observons que le nombre de changements de prochain saut reste faible et assez stable dans le scénario de routage de métriques combinées par la logique floue. Nous avons une moyenne de 43.52 changements par heure avec le routage par ETX, cette moyenne est de seulement 6.63 pour l'autre scénario. Ces résultats montrent que le routage par l'ETX implique un grand nombre de changements de parents aux cours du temps. Ceci se justifie par le fait que cette métrique soit très dynamique. Par ailleurs, comme nous l'observons, sa combinaison par la logique floue avec d'autres paramètres, limite l'impact de sa forte variabilité, améliorant non seulement la stabilité des routes mais aussi le taux de livraison des informations à la destination.

5.4.4 Efficacité énergétique

Pour évaluer la répartition énergétique des nœuds au bout de 16 jours d'expérience, nous les avons tous déployés tous avec la même capacité de batterie (supposée représenter 100% de charge). La figure 5.11 illustre cette distribution, indiquant la proportion de nœuds par capacité énergétique résiduelle. Nous observons nettement que la pyramide induite par l'histogramme de distribution énergétique des nœuds dans le scénario de métrique combinées par la logique floue est plus à droite que celle du scénario à base de l'ETX. Ceci montre que le premier scénario est plus conservateur d'énergie que le routage par l'ETX. En effet, dans la combinaison par la logique floue, 67.85% de nœuds ont leur énergie supérieure 83% de la charge initiale, tandis que dans la version native de RPL, cette même proportion est seulement de 21.42%. Par ailleurs nous notons que le scénario de combinaison par la logique floue maintient une proportion important de nœuds (39.27%) au même niveau d'énergie résiduelle. Ce qui démontre sa tendance à équilibrer la consommation énergétique des nœuds. A contrario, on observe à la date considérée que : les nœuds de plus faible capacité énergétique (énergie $\leq 81\%$ de l'énergie initiale), sont beaucoup plus nombreux avec le routage par l'ETX (50% de nœuds) que dans la combinaison (17.85%). Ces résultats montrent la considération faite de l'énergie, en plus des autres paramètres dans la combinaison des métriques par la logique floue.

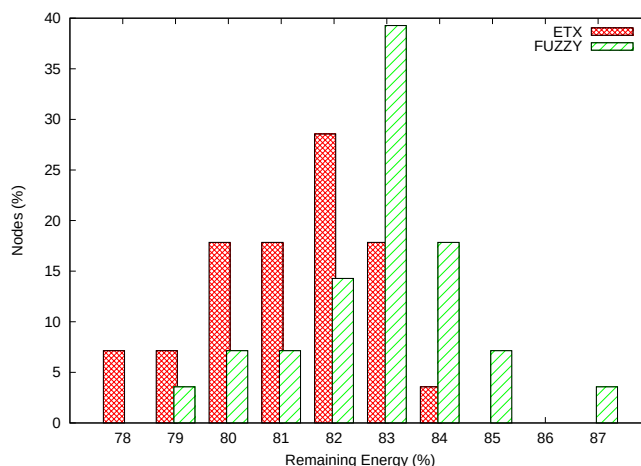


FIGURE 5.11 – Distribution de l'énergie résiduelle des nœuds

5.4.5 Délai de bout-en-bout

Le délai est la troisième métrique que nous considérons lors de la sélection du prochain saut par un nœud. L'évaluation du délai de bout-en-bout constitue un véritable défi dans un réseau de capteurs déployés en conditions réelles. Celle-ci nécessite que les nœuds soient synchronisés et partagent une horloge de référence commune. La mise en œuvre de ces mécanismes seraient très coûteuse en mémoire et processeur, éléments déjà assez limités sur les capteurs déployés. Par ailleurs, cela induit des messages de contrôle supplémentaires et une surcharge réseau. Pour ces raisons et par souci de simplicité, nous n'avons évalué le délai de bout-en-bout que dans la version émulée de nos capteurs sous Cooja, sous des conditions similaires à celui du réseau déployé. Les nœuds partagent le même code image que ceux déployés en conditions réelles, seule la couche radio est simulée. Le fait que tous les nœuds soient émulés dans la même machine et partagent une horloge commune avec le simulateur Cooja, nous permet d'évaluer avec plus de précision le délai de bout-en-bout. La figure 5.12 présente la fonction de répartition (CDF : Cumulative Distribution Function en *anglais*) du délai de bout en bout. Cette courbe n'est considérée que pour les paquets effectivement arrivés à la destination. Bien que l'écart entre les deux scénarios de routage ne soit pas important, la combinaison par la logique floue se comporte mieux. En effet, nous observons que 75% des paquets sont livrés au point de collecte en moins de 1 seconde, tandis que cette proportion est de 68% dans le routage par l'ETX. Il faut noter que, comme indiqué en §5.4.2, le taux de livraison de paquets étant meilleur dans la combinaison de métriques, le délai de bout-bout en bout considéré est beaucoup plus précis que celui avec l'ETX car d'avantage de paquets sont pris en compte.

5.5 Conclusion

Dans ce chapitre, nous avons conçu et mis en œuvre une nouvelle fonction d'objectif RPL combinant plusieurs métriques. Ceci nous permet de prendre en compte la QoS en considérant plus d'un paramètre de performance réseau. Bien que dans la littérature et la documentation relative au standard RPL, plusieurs métriques soient

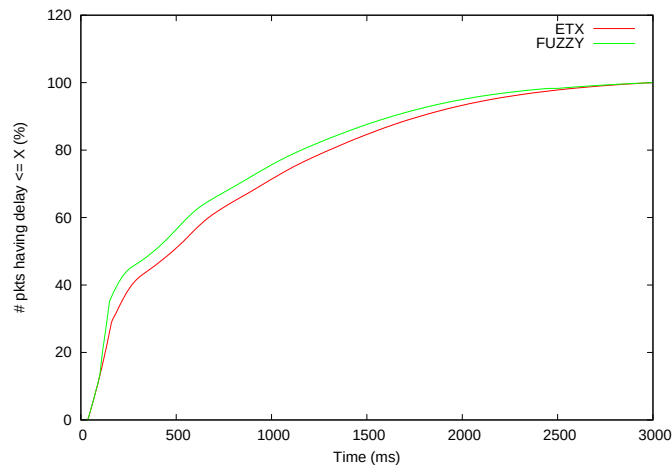


FIGURE 5.12 – Fonction de répartition du délai de bout-en-bout

décrites, la quasi-totalité des implémentations que nous avons rencontrées jusqu'ici ne prennent qu'un seul critère (les plus répandus étant le nombre de sauts et l'ETX). Les auteurs de [Kar+13] ont mis en œuvre puis évalué les méthodes de combinaison additive et lexicographique des métriques dans le contexte de RPL. Mais ces deux formes de combinaison exhibent l'inconvénient de limiter le type de métriques à considérer (additives) ou d'optimiser en priorité le (ou les) premier(s) critère(s) (lexicographique), les métriques suivantes étant utilisées pour faire un choix décisif en cas d'égalité sur celles qui précèdent. Nous avons utilisé la logique floue pour combiner l'ETX, le délai et l'énergie en une valeur de composition guidant le routage. Ce modèle offre de bons résultats dans la recherche d'un compromis entre plusieurs critères et convient à l'environnement contraint en ressources que constitue les réseaux de capteurs sans fil. Les résultats obtenus d'expériences réalisées sur plusieurs capteurs déployés en réseau dans un bâtiment, sont meilleurs que le routage par l'ETX en termes de taux de livraison de paquets, de stabilité de routage, de distribution de la consommation énergétique des nœuds et de délai de bout-en-bout. Un autre avantage qu'offre la logique floue est la possibilité d'adapter en fonction de la QoS souhaitée, le poids (ou contribution) de chaque métrique à la combinaison. Dans ce travail, nous donnons la même importance aux différentes métriques considérées. Comme perspective à celui-ci, il serait intéressant de paramétrer dynamiquement le poids des métriques pendant le fonctionnement du réseau.

Troisième partie

Configuration et Interconnexion pour l'Internet des Objets

Chapitre 6

Réseau fédérateur de passerelles pour l'Internet des Objets

6.1 Introduction

Ces dernières années, les réseaux de capteurs et plus généralement l'Internet des Objets ont profondément révolutionnés la façon dont nous vivons et travaillons, offrant ainsi de nouvelles opportunités et services. Pour soutenir ceux-ci, plusieurs RCSF peuvent être déployés dans le même environnement, chacun répondant à un besoin spécifique. Il est par exemple possible d'avoir un réseau de capteurs dédié à la surveillance de certains paramètres environnementaux comme la température ou l'humidité. Dans le même bâtiment (ou aire géographique), un autre réseau pourrait être chargé de la détection de présence ou le relevé de consommations de compteurs électriques intelligents et leur acheminement vers un centre de traitement d'informations pour ainsi adapter la production électrique à la consommation. Tous ces réseaux devront être pleinement exploitables à travers une connexion Internet. Nous avons pour ambition d'offrir un tel accès de façon fiable et efficace moyennant un coût d'investissement matériel assez faible. Ceci offre la possibilité d'envoyer les données collectées vers des centres de traitement (sur Internet) adaptés aux gros volumes de données générées. Les outils de « cloud computing » et du « big data » sont visées dans ce cas. Par ailleurs, les capteurs et actionneurs déployés recevront leurs ordres ou commandes d'applications métiers déployées sur des machines puissantes depuis Internet ou de smartphones via une interface Web. Les scénarios applicatifs d'une telle infrastructure sont nombreuses et couvrent divers domaines tels que la domotique et les systèmes immotiques (bâtiments intelligents) où ils permettent d'améliorer le confort ou qualité de vie. Dans le domaine de la santé et celui des villes intelligentes il offrent divers services aux citoyens et patients. Lorsqu'ils sont utilisés dans l'industrie, ces réseaux sont sujets à des contraintes de délais et de QoS spécifiques.

Dans le but d'offrir une connectivité Internet aux différentes RCSF déployés, nous proposons d'installer plusieurs passerelles en réseau, sous forme d'une infrastructure fédératrice. Chaque passerelle rendra ainsi possible la communication entre les deux sortes de réseaux à connecter (réseaux sans fil – côté RCSF et réseaux Ethernet ou toute autre technologie d'accès – côté réseau fédérateur). Elles relieront de cette façon (via Internet) non seulement les parties non connexes d'un même RCSF, mais elles assureront également pour chacune, la connexion (Internet) de plusieurs RCSF différents.

En raison de sa capacité finie (en terme de bande passante), la passerelle unique point d'entrée vers Internet, constituera un goulot d'étranglement du trafic issue des nœuds des différents RCSF. Par ailleurs, la faible fiabilité des communications sans fil des capteurs fait qu'il ne serait pas raisonnable de disposer que d'une seule passerelle dans le réseau. D'autant plus que nous visons à déployer plusieurs RCSF et concevoir une architecture passant à l'échelle. Chaque composante connexe de RCSF devra alors avoir la possibilité d'accéder à Internet par une (ou plusieurs) des passerelles, en fonction de la situation du réseau. L'idée générale est la suivante, partant d'un certain nombre de passerelles précédemment déployées nous souhaiterions déterminer celles qui sont les plus appropriées pour assurer la connexion des différents RCSF en fonction de contraintes spécifiques exprimées sous forme de demande du réseau (nombre de nœuds, quantité de trafic et QoS souhaitée) et capacité associée à chaque passerelle. Le réseau devrait pouvoir s'adapter et changer de point d'entrée Internet en fonction des conditions courantes du réseau, de la disponibilité des passerelles ou du trafic déjà supporté par celui-ci.

6.2 Considérations liées à la conception

La figure 6.1 présente un exemple de schéma d'interconnexion du réseau visé. L'infrastructure globale mise en place est conçue suivant une architecture en trois couche : RCSF – réseau fédérateur des passerelles (*backbone*) – Internet. Les passerelles matérialisées comme des routeurs sont des nœuds beaucoup plus robustes que les capteurs déployés dans la partie RCSF. Ils disposent d'une source d'alimentation continue ainsi que d'importantes ressources mémoires et de calcul. Dans le prototype proposé, nous avons opté d'utiliser la plate-forme Raspberry Pi [Ras] en raison de leurs caractéristiques matérielles intéressantes et du faible coût requis pour leur acquisition. Nous pouvons ainsi en déployer un nombre important si l'utilisation du réseau l'exige, avec un budget matériel peu élevé. Les nœuds du RCSF collectent et acheminent les informations par transmission radios multi-sauts au routeur de bordure intégré à la passerelle. Ces nœuds appartiennent à des réseaux distincts et sont matérialisés par des codes couleurs (grise, bleu et rouge dans la figure) représentant chacune une application (ou utilisation) spécifique du RCSF.

Un des problèmes à résoudre est celui de la déterminer le nombre approprié de passerelles à utiliser parmi celles installées. Deux objectifs quelque peu conflictuels sont à relever :

- Chaque RCSF visera à utiliser le plus grand nombre de passerelles disponibles. En effet, plus il de passerelles activées pour un RCSF donné, plus grande est capacité totale offerte (transport des données) et meilleure est la QoS. Par ailleurs, disposer de plusieurs passerelles garantie une tolérance aux pannes.
- Seul un nombre *adéquat* de passerelles doivent être utilisées, *i.e.* celles correspondant aux besoins actuels du réseau. Plus de passerelles pourront être installées mais seule une quantité limitée sera activée, celle pouvant garantir une « bonne communication » des RCSF déjà déployés tout en satisfaisant les contraintes exprimées. La capacité résiduelle (passerelles non utilisées) permettra de répondre

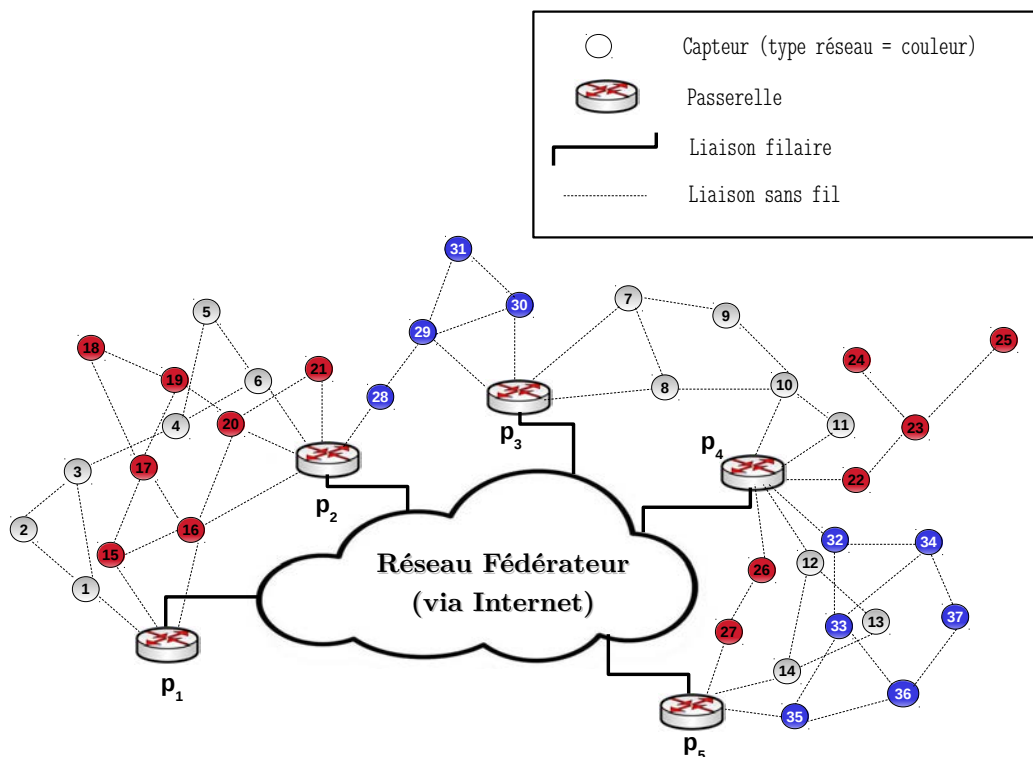


FIGURE 6.1 – Réseau fédérateur de passerelles interconnectant les RCSF

à d'éventuelles défaillances (pannes de passerelles, rupture de lien, ...) ou assurera le passage à l'échelle (ajout de nouveaux nœuds ou déploiement d'un nouveau RCSF).

Un objectif supplémentaire consiste à répartir le trafic (émanant de tous les RCSF déployés) sur l'ensemble des passerelles *mis en service*. Ceci permet de maximiser l'utilisation des ressources actives.

Une fois les passerelles constituant la couche intermédiaire (*backbone*) sélectionnées, les capteurs devront s'organiser en réseau pour leur acheminer de façon efficace le trafic aux différents services. Chaque passerelle pourra alors jouer le rôle de routeur de bordure pour plusieurs RCSF à la fois. Nous nous appuyons sur le protocole RPL et les contributions proposées aux chapitres précédents pour organiser le routage dans la partie RCSF, chaque passerelle assurant également la fonction de racine RPL. Par ailleurs, le protocole est configuré dans sa version multi-instance (*i.e.* possibilité de faire cohabiter plusieurs réseaux simultanément). Une instance offre les services adaptés pour un RCSF distinct et sera, selon les besoins de l'application cible, associée à une fonction d'objectif optimisant un certain nombre de métriques (énergie, ETX, nombre de saut, combinaison, etc.). L'approche de combinaison par la logique floue (cf. chapitre 5) offre la possibilité de *prioriser* certaines métriques en fonction des besoins de QoS. Il sera ainsi possible de mettre d'avantage l'accent sur l'énergie en modifiant les bornes des *fonctions d'appartenance* y relatives.

Les avantages liés à la mise en œuvre de l'architecture proposée sont nombreuses. Tout d'abord, elle offre un accès Internet aux différents RCSF déployés. Deuxièmement, la fiabilité de la connexion à Internet est améliorée en raison de possibilités de points d'entrée redondants grâce aux nombreuses passerelles installées. Troisièmement, la capacité du réseau croît avec l'ajout des nouvelles passerelles. Cette croissance dépend linéairement du nombre de passerelles si les ressources disponibles sur ces dernières sont convenablement allouées et la charge équitablement réparties entre elles. Enfin, les passerelles pourront allouer/désallouer dynamiquement leurs ressources aux différents RCSF, en fonction de la situation (ajout ou retrait de RCSF) et de l'architecture du réseau (nombre de nœuds, graphe d'interconnexion/voisinage).

6.3 Modélisation

Dans cette section, nous présentons une modélisation formelle du problème de sélection de passerelles au niveau de la couche intermédiaire de l'architecture proposée. Le problème est formulé comme un programme linéaire en nombres entiers (ILP – *Integer Linear Program*) visant à atteindre différents objectifs sous certaines contraintes. Les solutions au ILP sert de base à la construction de l'infrastructure fédératrice assurant à faible coût la connexion des différents RCSF à l'Internet suivant de « bonnes performances ».

6.3.1 Modèle de réseau

Le schéma de la figure 6.1 illustre une configuration possible de l'architecture envisagée. Le réseau peut être modélisée par un graphe orienté $G = (S, A)$, où A est l'ensemble d'arcs entre les nœuds et représente les connexions sans fil entre les nœuds. $S = S_p \cup S_n$ est l'ensemble de sommets constitués de $S_p = \{p_1, p_2, \dots, p_m\}$ ensemble de m passerelles et S_n ensemble des capteurs installés. Ces derniers peuvent appartenir à des RCSF distincts déployés dans le même environnement.

Plus formellement, on note $S_n = \bigcup_{j=1}^k S_{n_j}$, k étant le nombre de RCSF déployés. Ils forment une partition S_{n_j} de S_n . Il convient de noter que bien que partageant le même environnement les RCSF sont disjoints

A titre d'illustration, si l'on se réfère au schéma de la figure 6.1, on a $k = 3$, $m = 4$ et $|S_n| = 37$. Soient S_{n_1} , S_{n_2} et S_{n_3} les RCSF respectivement matérialisés par les codes couleurs gris, rouge et bleu. Nous avons alors $|S_{n_1}| = 14$, $|S_{n_2}| = 13$ et $|S_{n_3}| = 10$.

Tous les nœuds (passerelles et capteurs) sont supposés fixes une fois déployés. Les passerelles p_i possède chacune une capacité finie $C(p_i) = \alpha_i$ connue au moment de leur installation. Cette capacité est proportionnelle à la quantité de trafic qu'elle est capable de traiter compte tenu de ses performances et de la bande passante d'accès à Internet qu'elle offre. Pour des raisons de simplicité, on suppose que tous les capteurs génèrent la même quantité de données. La capacité de chaque passerelle représente

alors « le nombre de capteurs pouvant être pris en charge pour un trafic vers Internet. »

Pour minimiser les problèmes d'interférence, chaque RCSF est supposé utiliser sur un canal dédié. Une couche PHY de l'IEEE802.15.4 dispose à cet effet de 16 canaux distincts (numérotés de 11 à 26). Les passerelles seront alors être équipées d'autant d'interfaces que nécessaire chacun transmettant sur un canal donné.

6.3.2 Formulation du problème

Pour la mise en place du réseau, plus il y a des nœuds servant de passerelle, plus importante est capacité totale du système. Toutefois, il est souhaitable de minimiser le coût relatif à l'installation et l'utilisation des passerelles. Par ailleurs, la répartition du trafic sur l'ensemble des passerelles utilisées est recherchée, de même que la garantie de suffisamment de ressources (bande passante) pour permettre un accès Internet de « bonne qualité » aux différents RCSF.

L'objectif général est le suivant, déterminer parmi les passerelles installées celles qui doivent être activées (nombre et position) de façon à assurer une connectivité Internet globale qui soit efficace aux regards des conditions du réseau. Une fois les passerelles déterminées, dans un second temps, il s'agira de construire le réseau en le reconfigurer dynamiquement pour répondre aux modifications topologiques. Le problème peut être formellement défini comme suit :

« Étant donné un réseau représenté, par le graphe $G = (S_p \cup S_n, A)$, constitué de $|S_p| = m$ passerelles, chacune de capacité $C(p_i) = \alpha_i$, $i \in \llbracket 1, m \rrbracket$. Soient k réseaux de capteurs constitués chacun de $|S_{n_j}|$ nœuds, $j \in \llbracket 1, k \rrbracket$, sélectionner le nombre minimum $s \leq m$ de passerelles parmi les m existantes qui satisfait la demande des différents réseaux de capteurs sans fil sous contraintes spécifiques (libellées ci-dessous). »

La recherche de solutions au problème susmentionné est sujette aux contraintes :

- **Couverture de l'ensemble des nœuds** : tous les capteurs doivent obtenir une couverture Internet. Cet accès est possible via n'importe quelle passerelle par communications radios multi-saut. Il doit exister au moins un chemin qui relie tout nœud à au moins une passerelle.
- **Capacité des passerelles** : Le nombre de nœuds desservies par une passerelle donnée ne doit pas dépasser sa capacité. Si celle-ci est atteinte, d'autres passerelles devront être ajoutées au réseau (ou activées si précédemment installées).
- **Utilisation des passerelles** : Seules les passerelles activées par résolution du problème énoncé plus haut peuvent être utilisées comme point d'accès à Internet par les nœuds. De même, une passerelle n'est activé que s'il existe au moins un nœud l'utilisant pour accéder à Internet.
- **Profondeur du réseau** : De part la nature non fiable des communications sans fil, le taux de perte des paquets augmente avec l'éloignement des nœuds de la station de base. Dans le but de contrôler le taux de livraison, une contrainte supplémentaire permettant de limiter cette profondeur lors de la construction du RCSF est introduite. Soit MAX_{Deep} ce nombre maximum de saut. Tout nœud

ne peut utiliser une passerelle que si celle-ci est située à une distance d'au plus MAX_{Deep} .

Sous les contraintes listées ci-dessus, deux objectifs principaux sont recherchés :

1. **Minimiser le nombre de passerelles activées.** Des capteurs peuvent être sous la zone de service de plusieurs passerelles. Les passerelles utilisées sont celles alimentant ces derniers aux meilleurs coûts. L'objectif recherché lors de la construction du réseau est l'optimisation de la métrique de routage considérée (nombre de saut, délai, énergie ou une combinaison de ces critères).
2. **Équilibrage de charge.** Répartir le trafic sur les passerelles activées permet de maximiser leur utilisation.

6.4 Procédure de résolution

6.4.1 Complexité algorithmique

Proposition 1 (NP-difficulté du problème de sélection de passerelles). *Le problème de sélection de passerelles tel qu'énoncé en § 6.3.2 est NP-difficile.*

Preuve. Établir la NP-difficulté de la sélection des passerelles se fait par réduction du problème classique du CFLP (Capacitated Facility Location Problem) [She90] à celui-ci.

En effet, on dispose d'un ensemble U d'usines capables de produire chacune une certaine quantité d'un produit donné. Les usines sont soit ouvertes, soit fermées et doivent satisfaire chacune à une demande d_j émanant de divers clients $j \in D$ situés dans différentes villes. L'ouverture d'une usine $i \in U$ est associée à un coût de fonctionnement f_i et une capacité de production α_i correspondant à la demande maximale qu'elle est capable de satisfaire. Le coût de service d'une demande unitaire d'un client j par l'usine i est c_{ij} . Le problème CFLP revient à déterminer un sous-ensemble $U' \subseteq U$ d'usines à ouvrir de façon à satisfaire la demande de tous les clients et de telle sorte que le coût total d'ouverture de ces usines ainsi que celui associé au service des clients soit minimal.

La réduction du CFLP au problème de sélection des passerelles se fait de façon aisée. Nous pouvons voir le CFLP comme une instance du problème de sélection des passerelles, où les usines correspondent aux passerelles et, ont tous le même coût de fonctionnement $f_i = 1$. Le coût de service c_{ij} d'un client (i.e. nœud) j par l'usine i est associé à au poids du chemin (au sens de la métrique de routage) entre le capteur et la passerelle correspondante. Nous supposons que tous les nœuds expriment la même demande unitaire i.e. $d_j = 1, \forall j \in D$ correspondant au trafic généré. On peut facilement généraliser cette demande à une valeur arbitraire pour tout nœud. Ainsi, la capacité d'une usine est la quantité de trafic que la passerelle associée peut relayer. Elle est fixée à α_i comme étant le nombre maximum de nœuds pris en charge par la i^{ime} passerelle. Par cette réduction, le CFLP est bien vu comme une instance du problème de sélection des passerelles. Puisque la NP-difficulté du CFLP est établie [ZCY04], nous pouvons en déduire celle de la sélection des passerelles. $\square$

6.4.2 Méthode de résolution

Le problème peut être modélisé comme un programme linéaire en nombres entiers. Les notations utilisées dans le modèle sont les mêmes que celles introduites en §6.3.1. Les passerelles éléments de S_p sont indexées par i tandis que les capteurs éléments de S_n sont indexés par j . Deux nouvelles variables multidimensionnelles X et Y à valeurs binaires (resp. X_{ij} et Y_i) sont introduites et définies comme suit :

$$X_{ij} = \begin{cases} 1 & \text{Si la passerelle } i \text{ est utilisée par le nœud } j \\ 0 & \text{Sinon} \end{cases} \quad (6.1)$$

$$Y_i = \begin{cases} 1 & \text{Si la passerelle } i \text{ est activée} \\ 0 & \text{Sinon} \end{cases}$$

Le problème de sélection des passerelles peut être récrit sous la forme du programme linéaire suivant :

$$\text{Minimiser} \left[\sum_{i=1}^{|S_p|} \sum_{j=1}^{|S_n|} c_{ij} X_{ij} + \sum_{i=1}^{|S_p|} Y_i + \sigma \left(\sum_{j=1}^{|S_n|} X_{ij} \right) \right] \quad (6.2)$$

$$\text{Sujet à :} \left\{ \begin{array}{ll} \sum_{i=1}^{|S_p|} X_{ij} = 1, & \forall j \in S_n \end{array} \right. \quad (6.3)$$

$$\sum_{j=1}^{|S_n|} X_{ij} \leq \alpha_i, \quad \forall i \in S_p \quad (6.4)$$

$$X_{ij} \leq Y_i, \quad \forall i \in S_p, \forall j \in S_n \quad (6.5)$$

$$\sum_{i=1}^{|S_p|} c_{ij} X_{ij} \leq \text{MAX}_{Deep}, \quad \forall j \in S_n \quad (6.6)$$

$$0 \leq X_{ij}, Y_i \leq 1, \quad \forall i \in S_p, \forall j \in S_n \quad (6.7)$$

La fonction d'objectif donnée par la relation 6.2 vise non seulement à minimiser le nombre de passerelles à ouvrir ($\sum_{i=1}^{|S_p|} Y_i$), mais également le coût du chemin d'accès (par la métrique de routage) du capteur j à la passerelle i ($\sum_{i=1}^{|S_p|} \sum_{j=1}^{|S_n|} c_{ij} X_{ij}$). Par ailleurs, pour prendre en considération la répartition de la charge sur l'ensemble des passerelles activées, un troisième terme $\sigma \left(\sum_{j=1}^{|S_n|} X_{ij} \right)$ est défini comme étant l'écart type de la charge des passerelles activées. Plus petite est la valeur, meilleure est la distribution de charge des passerelles par rapport à la moyenne.

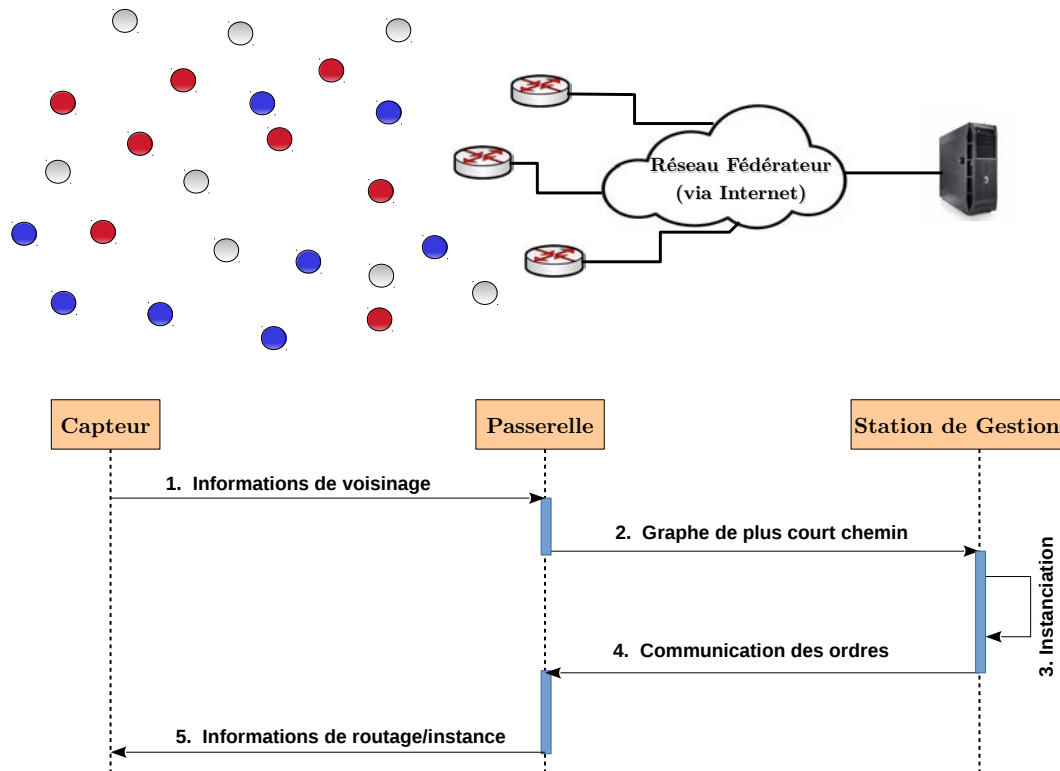


FIGURE 6.3 – Architecture du système

La figure 6.3 présente l'architecture globale du système. Trois entités sont nécessaires à la mise en œuvre de la plate-forme proposée, les nœuds-capteurs, les passerelles et une station de gestion centralisée. Le diagramme de séquence présenté dans la partie inférieure du schéma souligne les interactions entre ces différentes entités.

1. **Collecte des informations topologiques :** Pour chaque instance, les nœuds collectent régulièrement les informations de voisinage qu'ils font remonter par multidiffusion aux différentes stations de base. Dans l'implémentation proposée, station de base et passerelle sont assurées par le même dispositif physique (Raspberry Pi). Pour collecter et gérer les informations topologique un protocole dédié (DRPLCP – présenté un peu plus loin cf. § 6.6.2.1) a été conçu et mis en œuvre.
2. **Graphe des coûts :** Chaque passerelle construit à partir du graphe topologique reçu des nœuds sous-jacents, l'arbre des plus court chemins où elle est elle-même racine et les capteurs sont soit des nœuds internes soit des feuilles. Cet arbre, qui est le résultat de l'arborescence de routage optimale pour la métrique considérée, est envoyé par chacune des passerelles à la station de gestion qui les agrège et constitue la matrice des coûts C . Une ligne de la matrice représente les coûts de plus courts chemins de la passerelle associée vers tous les nœuds du réseau. L'infini dénotant une inaccessibilité du nœud par la passerelle.
3. **Instanciation du modèle :** Une fois les entrées du modèle constituées (matrice

des coûts C , vecteur des capacités des passerelles α et longueur maximale des chemins MAX_{Deep}) le programme linéaire en nombres entiers est exécuté par le solveur CPLEX sur la station de gestion. Les sorties de ce programme sont converties en ordres d'activation ou fermeture de passerelle. Les différents rôles (nombre d'instance RPL à configurer par racine) de passerelle à activer sont également déterminés.

4. **Communication des ordres/rôles** : Les ordres et rôles déterminés à l'étape précédente sont transmis par la station de gestion aux différentes passerelles via le réseau. Celle-ci adaptent leur fonctionnement en conséquence et mettent à jour leur capacité pour tenir compte des ressources déjà allouées. La nouvelle capacité est obtenue par déduction à la précédente de la charge de trafic additionnelle à prendre en compte. Plus formellement, $\alpha_i(t) = \alpha_i(t-1) - \beta_i(t)$, où $\alpha_i(t)$ est la nouvelle capacité de la passerelle i à l'instant t , $\alpha_i(t-1)$ sa capacité précédente et $\beta_i(t)$ la charge de trafic supplémentaire à recevoir en raison de reconfiguration réseau.
5. **Configuration du routage** : Les passerelles nouvellement activées commencent à envoyer leurs paramètres de configuration (DIO) pour les instances RPL prise en charge. Les nœuds rejoignent selon leur fonction d'objectif et l'arborescence de routage est alors construite en optimisant la métrique en vigueur (nombre de saut, énergie, ETX, délai, ou combinaison)

Pour prendre en compte la dynamique du réseau, les capteurs continuent de surveiller leur voisinage au cours de leur activité. Afin de limiter la consommation des ressources, les informations de voisinage ne sont remontées qu'en cas de modifications topologiques (découverte d'un nouveau nœud ou mort/disparition d'un voisin). Les passerelles recevant ces informations les relayent à la station de gestion qui reconfigure le réseau si nécessaire. Une passerelle est dite fermée dans le sens où elle ne construit pas d'arborescence de routage RPL et n'assure donc pas l'interconnexion vers Internet.

6.6 Mise en œuvre et déploiement

6.6.1 Architecture matérielle

Comme indiqué plus tôt, le système est constitué de deux infrastructure réseaux, le RCSF et Internet (via le *backbone*). Dans l'implémentation proposée, les nœuds déployés dans la partie RCSF sont des Telos B de type CM5000 [Tel] équipés d'un micro-contrôleur MSP430 et d'une puce radio CC2420 opérant selon le norme IEEE802.15.4. Diverses interface de captage sont embarquées pour l'acquisition des données environnementales (*i.e.* température, humidité, luminosité etc.). Dans la partie *backbone*, c'est le Raspberry Pi-3 agissant à la fois comme passerelle et racine pour le DODAG RPL, qui a été choisi. Les principales raisons de ce choix sont la taille assez réduite de la plateforme et ses performances intéressantes au regard de son faible coût d'acquisition. Par ailleurs, ses caractéristiques le rendent apte à pouvoir assurer à la fois les services dévolus à la passerelle et ceux liés au routage. Le mode d'utilisation réalisé avec ce

dernier est le même que celui décrit plus tôt en § 3.2.2 et illustré par la figure 3.7 (association Raspberry–Telos B). Le Telos B est utilisé comme interface IEEE802.15.4 pour connecter le RCSF tandis que le port Ethernet assure la connexion l'équipement à Internet.

6.6.2 Fonctionnement réseau

6.6.2.1 Côté réseau de capteurs sans fils

Dans la partie RCSF, les nœuds exécutent chacun une application spécifique (dépendant du service) gravée dans leur mémoire flash avant leur déploiement. Les différentes applications sont répertoriées par un numéro d'instance RPL au niveau de la passerelle. Deux catégories d'instance sont gérées : les instances dites *applicatives* dédiées aux applications et l'instance de *configuration*. Dès qu'un nœud est installé dans le réseau, il commence par s'enregistrer dans un groupe de diffusion grâce à l'instance de *configuration* et attend de recevoir de la racine RPL une instance applicative. Un protocole dédié, DRPLCP (Dynamic RPL Configuration Protocol) a été conçu pour la gestion des enregistrement de nœuds et le dispatching par la racine des instances applicatives. La possibilité d'installer de façon progressive les capteurs de même que la nature dynamique de l'allocation des services (applications) permet de déployer le réseau graduellement.

Aussitôt qu'un nœud obtient son instance applicative, il rejoint l'arborescence RPL correspondante et commence à exécuter l'application associé à l'instance reçue. Pendant les phases d'activité normale le nœud continu de remonter à sa passerelle l'information de voisinage si nécessaire (changement topologique) qui infère un nouveau graphe réseau servant à l'instanciation du modèle.

6.6.2.2 Côté passerelle

La passerelle est mise en œuvre par l'association Raspberry–Telos B. Dans ce couplage le Raspberry prend en charge à la fois la distribution des instances mais aussi le pontage entre RCSF et Internet. Elle assure également une communication bidirectionnelle avec la station de gestion : envoie des paramètres topologiques et réception des ordres de fonctionnement. Pour interfacer le RCSF, le Telos B exécute une application SLIP basique [Rom88] : lecture d'un flux de données brutes sur l'interface radio IEEE802.15.4 qu'il réécrit sur le port USB de l'équipement et inversement.

Une fois qu'il a démarré, la passerelle dissémine dans des DIO les informations de configuration de toutes les instances locales pour lesquelles elle assure la fonction de racine. En conséquence, les nœuds du voisinage appartenant à l'une des instances supportées peuvent rejoindre le réseau, sélectionner leur prochain saut et commencer à y participer. Un moment important qu'il convient de présenter est la gestion des instances nouvellement créées au niveau de la passerelle. Elle annonce celle-ci via l'instance dédié à la *configuration* en utilisant le groupe de multidiffusion. Tout nœud nouvellement déployé, *i.e.* et n'ayant pas encore reçu d'instance *applicative*, peut alors en recevoir une et contribuer aux services fournis par réseau associé (exécution de l'application + collecte des informations de voisinage).

6.6.3 Déploiement et cas d'utilisation

Pour évaluer la plateforme développée pour le modèle proposé, dans un premier temps nous avons évalué la formation du réseau et son passage à l'échelle. Étant donné que le prototype que nous avons développé pour le simulateur utilisé ne supporte actuellement qu'une seule passerelle, un réseau aussi grand que le permet l'ordinateur hôte de la simulation a été monté. La simulation permet également de pouvoir évaluer la prise en charge par le modèle de plusieurs instance ainsi que la communication entre Internet et les nœuds du RCSF.

Dans un second temps, un scénario réel a été mis déployé pour évaluer le fonctionnement du modèle en présence de plusieurs passerelle. Le cas d'utilisation choisi est celui de la domotique, considérant un réseau recueillant les informations environnementales dans un bâtiment.

6.6.3.1 Simulation grande échelle

En utilisant le simulateur Cooja intégré à Contiki, une topologie réseau de 500×500 de plus de 100 nœuds disposé de façon aléatoire a été installé. Plus qu'une simulation les nœuds disposés émulent code image que de réels capteurs Telos B. La fonction de passerelle est assurée par la machine hôte de la simulation qui fait tourner en arrière plan l'exécutable associé. La passerelle communique avec le RCSF monté dans le simulateur grâce à un nœud particulier qui exécute un programme SLIP. Les deux programmes échangent en utilisant une simple Socket TCP. L'ordinateur hôte des simulations fonctionne sous Linux Ubuntu 16.04 et possède un processeur de type Core 2 Extreme QX6850 cadencé à 3Ghz avec 8Go de mémoire. Pour des raison de simplicité, les identifiant d'instance sont les mêmes que le nombre de nœud y opérant, à savoir 50, 30 et 20. Le tableau 6.1 regroupe les principaux paramètres de simulation et ceux de la pile protocolaire.

Paramétrages	Valeurs
Couche Application	Collecte régulière de données captées
Couche Transport	UDP
Couche Réseau	μ IPv6 + 6LoWPAN + RPL (routage)
Couche MAC	CSMA non-persistant
RDC	ContikiMAC
PHY + Puce Radio	IEEE 802.15.4 & CC2420
Modèle Canal sans fil	Graphe de Disque Unitaire (UDG) & Perte avec distance
Porté Communications	80m (Tx/Rx), 100m (Interf.)
Type de nœud	Tmote sky (Telos B)

TABLE 6.1 – Paramètres de simulation

Le schéma de la figure 6.4 illustre le réseau simulé où les nœuds coloriés en jaune, violet et bleu représentent les instances d'identifiant respectif #50, #30 et #20. Le nœud vert situé au centre du réseau est celle assurant la connexion du simulateur au programme de passerelle s'exécutant sur l'ordinateur hôte de la simulation. Les relations entre nœuds matérialisées par des flèches correspondent aux prochains sauts

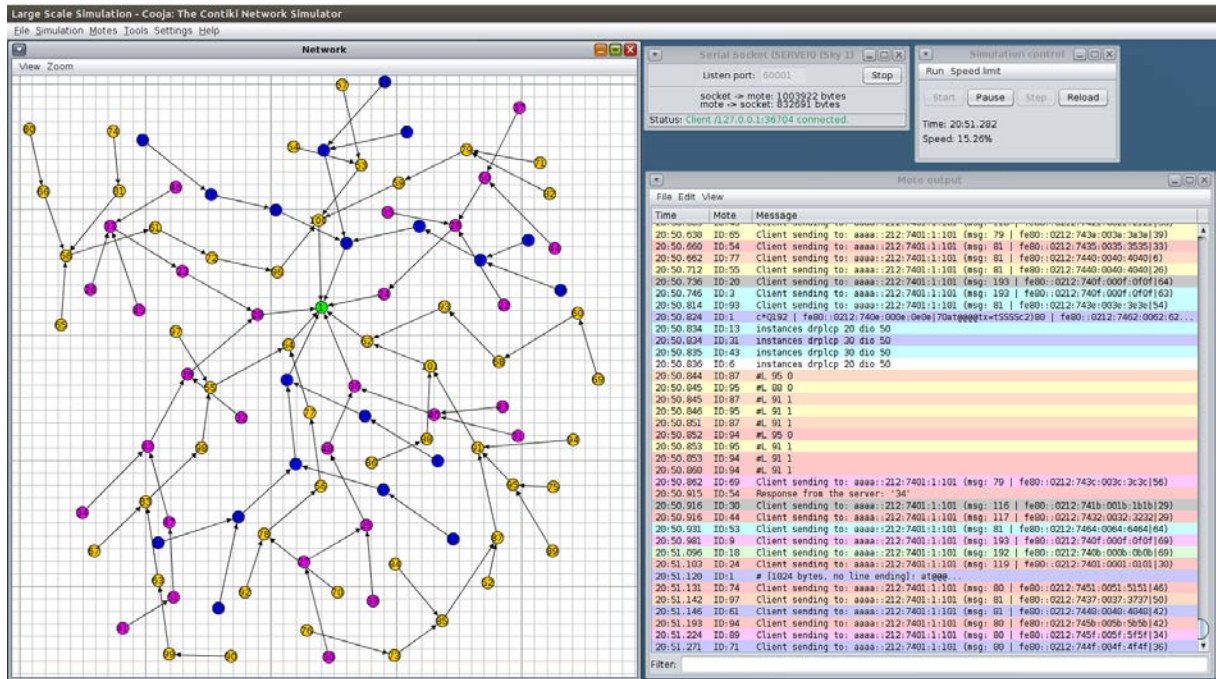


FIGURE 6.4 – Simulation Cooja d'un réseau grande échelle

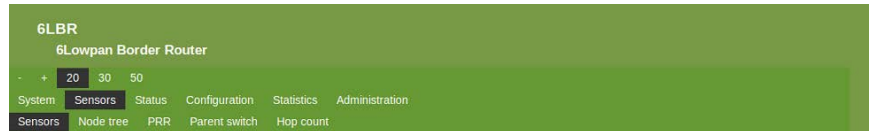
(parents préférés) sélectionnés par la fonction d'objectif RPL. Comme voulu la topologie réseau s'organise autour de trois arborescences de routage disjointes identifiable par leur couleur d'instance.

Une navigation sur l'adresse IPv6 de la passerelle permet de lister les différents capteurs appartenant à chacune des instances ainsi que différentes statistiques (chemins ascendants et descendants). A titre d'illustration la figure 6.5 montre bien que tous les nœuds associés à l'instance #20 ont été effectivement découverts et sont accessibles de n'importe quel nœud depuis Internet.

6.6.3.2 Expérience en environnement réel

A des fins de preuve de concept, un prototype de modèle a été déployé dans une habitation. Deux applications ont été développées pour matérialiser la notion d'instance. La première, enregistrée sous l'identifiant #10 permet de collecter des données de température tandis que la seconde, d'identifiant #20 réalise le relevé d'informations de luminosité. Les capteurs exécutent le système d'exploitation Contiki 2.7 et la passerelle fonctionnent avec un noyau Raspbian 4.4. Le projet 6LBR (présenté en § 3.2.1) a été modifié pour prendre en charge plusieurs instances de même que les ordres d'ouverture/fermeture de passerelles en fonction des commandes reçues de la station de gestion. Deux passerelles et dix capteurs sont disposés dans divers endroits d'une habitation de cinq pièces. Les passerelles sont connectés par leur port Ethernet (cf. figure 3.7) en réseau local qui est connecté lui même à Internet.

Les paramètres du modèle sont fixées de sorte que les passerelles soient à mesure de capturer la moitié des capteurs chacun, *i.e.* $\alpha = [5, 5]$. Par ailleurs la profondeur maximale est définie sur $MAX_{Deep} = 2$, ainsi tout nœud peut atteindre devra atteindre passerelle en au plus 2 sauts. Il faut noter que l'environnement d'expérience permet (en



Sensors list

Node	Type	Web	Coap	Parent	Up PRR	Down PRR	Last seen	Status
aaaa::212:7412:12:1212	Moteiv Telos	web	coap	fe80::212:740b:b:b0b	63.8%	37.9%	14	OK
aaaa::212:7409:9:909	Moteiv Telos	web	coap	fe80::212:740f:f:f0f	65.5%	36.2%	15	OK
aaaa::212:7414:14:1414	Moteiv Telos	web	coap	fe80::212:740f:f:f0f	61.4%	40.4%	38	OK
aaaa::212:740a:a:a0a	Moteiv Telos	web	coap	fe80::212:740e:e:e0e	75.0%	26.8%	64	OK
aaaa::212:7415:15:1515	Moteiv Telos	web	coap	fe80::212:7409:9:909	84.5%	17.2%	19	OK
aaaa::212:740b:b:b0b	Moteiv Telos	web	coap	fe80::212:7414:14:1414	82.8%	19.0%	5	OK
aaaa::212:740f:f:f0f	Moteiv Telos	web	coap	fe80::212:7401:1:101	77.6%	24.1%	22	OK
aaaa::212:7402:2:202	Moteiv Telos	web	coap	fe80::212:7401:1:101	82.8%	19.0%	0	OK
aaaa::212:7404:4:404	Moteiv Telos	web	coap	fe80::212:7408:8:808	88.1%	13.6%	0	OK
aaaa::212:7410:10:1010	Moteiv Telos	web	coap	fe80::212:7408:8:808	79.3%	22.4%	23	OK
aaaa::212:7405:5:505	Moteiv Telos	web	coap	fe80::212:7409:9:909	88.1%	13.6%	1	OK
aaaa::212:7413:13:1313	Moteiv Telos	web	coap	fe80::212:740e:e:e0e	81.0%	20.7%	18	OK
aaaa::212:7408:8:808	Moteiv Telos	web	coap	fe80::212:7403:3:303	79.7%	22.0%	3	OK
aaaa::212:7407:7:707	Moteiv Telos	web	coap	fe80::212:7402:2:202	91.5%	10.2%	3	OK
aaaa::212:7411:11:1111	Moteiv Telos	web	coap	fe80::212:7402:2:202	89.8%	11.9%	3	OK
aaaa::212:740e:e:e0e	Moteiv Telos	web	coap	fe80::212:7407:7:707	84.7%	16.9%	5	OK
aaaa::212:740d:d:d0d	Moteiv Telos	web	coap	fe80::212:740c:c:c0c	96.6%	5.1%	9	OK
aaaa::212:7406:6:606	Moteiv Telos	web	coap	fe80::212:7411:11:1111	55.2%	46.6%	33	OK
aaaa::212:740c:c:c0c	Moteiv Telos	web	coap	fe80::212:7407:7:707	69.5%	32.2%	10	OK

FIGURE 6.5 – Capteurs découverts de la passerelle [instance #20]

absence de bruits) à tous nœud d'être à portée de communication de tous les autres. L'ETX (fiabilité de transmission bidirectionnelle au niveau MAC) est utilisé comme de routage, ce qui rend possible un chemin passant par un voisin (indirect) plus intéressant que le chemin direct (passerelle).

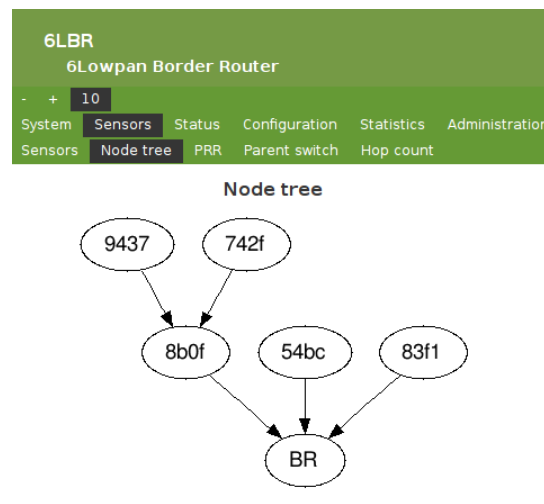
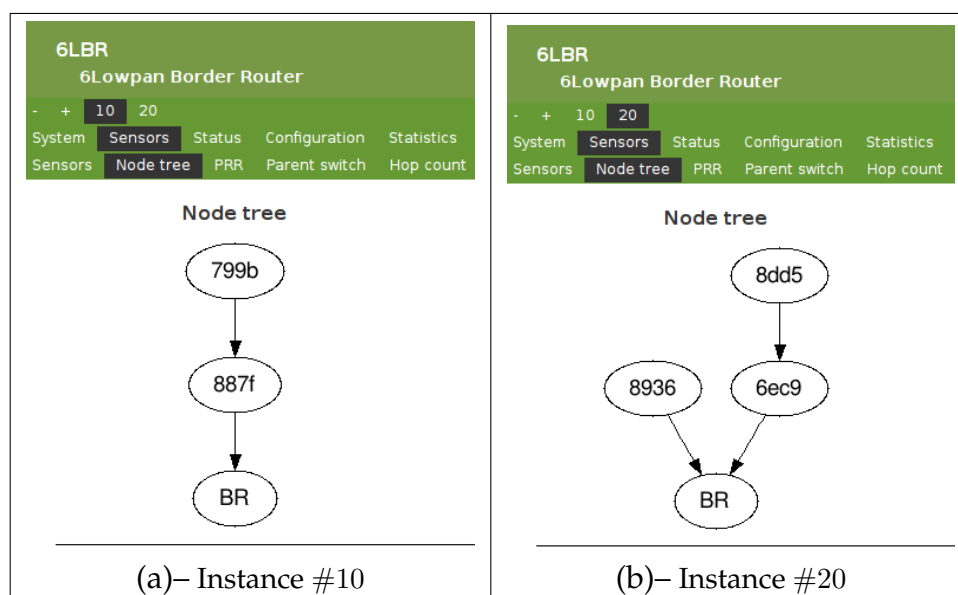
Dans un premier temps, cinq capteurs exécutant l'instance #10 associé à la première application sont déployés. L'instance est préalablement créée au niveau de deux passerelles. Après la phase d'initialisation, sur la base des paramètres susmentionnés, le système recommande l'activation d'une seule passerelle, notée *passerelle n°1*. La figure 6.6 illustre la topologie réseau obtenu à partir de l'interface web utilisé pour interagir avec cette passerelle. Comme indiqué, tous les nœuds (ici représentés par leur identifiants MAC) utilisent la *passerelle n°1* comme unique point de sortie vers Internet.

Les sorties produites par le modèle sont $X^{(1)} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$ et $Y^{(1)} = [1, 0]$.

Dans un second temps, la seconde instance est créée sur les passerelles et les cinq nœuds restant sont ajoutées au réseau de sorte que deux exécutent l'instance relative à la première application et les trois autres la seconde instance. Après ceci, au bout d'un certain temps, *passerelle n°2* (précédemment fermée) commence à opérer comme racine RPL pour les deux instances, tandis que *passerelle n°1* continue de capturer les nœuds précédant (tous dans l'instance #1). Les résultats produits par le modèle sont à présent

$X^{(2)} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$ et $Y^{(2)} = [1, 1]$.

La figure 6.7 présente le résultat de la navigation sur la seconde passerelle. Deux nœuds opèrent dans l'instance #10 (fig. 6.7-a) et trois dans l'instance #20 (fig. 6.7-b).

FIGURE 6.6 – Expérience pour la domotique : 1^{ère} Passerelle – Phase 1FIGURE 6.7 – Expérience pour la domotique : 2^{nde} Passerelle – Phase 2

6.7 Conclusion

Dans ce chapitre, nous examinons la question sélection de passerelle pour l'intégration efficace de RCSF à Internet dans la réalisation de la vision de l'IoT. Une architecture fédératrice permettant de prendre en charge plusieurs RCSF fournissant divers services (grâce à la notion d'instance) a été conçue et mise en œuvre. L'élément central du système proposé est la passerelle, nous l'avons choisi façon à obtenir un système final qui soit à la fois efficace mais aussi économiquement intéressant. Par ailleurs le

système est conçu sous forme d'une architecture en trois couches de façon à être flexible et évolutive. Le problème de sélection de passerelle est formulé comme un programme linéaire en nombres entiers. Une librairie basée sur CPLEX a été développée pour configurer dynamiquement le réseau en fonction des conditions variables du réseau. Celle-ci sélectionne les passerelles à activer (nombre et position) pour assurer une connexion Internet respectant les contraintes de capacité des différentes passerelles et garantissant une bande passante suffisante (pour les RCSF) tout en évitant les goulots d'étranglements.

Un prototype du modèle proposé a été implémenté comme preuve de concept en utilisant la plateforme matérielle Raspberry Pi pour héberger les services de la passerelle. Un déploiement réel pour une application de domotique qui témoigne de la faisabilité et fonctionnement du modèle proposé a été réalisé. Des simulations d'un réseau de grande taille ont également été réalisées sous Cooja pour évaluer le passage à l'échelle de l'architecture. Nous soutenons que la limite de la taille de simulation (100 nœuds) n'est due qu'aux ressources mémoire disponibles sur l'ordinateur hôte des simulations car le code de chaque capteur est entièrement émulé.

Chapitre 7

Conclusion et Perspectives

Le concept de l'Internet des Objets permet aux entités du monde réel (humains, véhicules), emplacements (maison, bâtiment, usines, villes) et toutes sortes de gadgets électroniques (montres, lunettes, jouets, etc.) d'être connectés à l'Internet, de publier leurs états et informations collectées par divers capteurs. De même, il est possible de contrôler ces objets à distance à partir d'Internet (via les services Web) et influencer grâce à des actionneurs sur le monde environnant. Ce nouveau paradigme a changé la façon dont les personnes, objets, données et services interagissent, offrant une large gamme d'applications et de nouveaux usages du réseau de données Internet.

Les réseaux de capteurs sans fil constituent un élément essentiel dans la réalisation de cette vision de l'Internet des Objets. En effet, ils permettent de couvrir de vastes zones grâce à de minuscules objets autonomes dotés d'interface de communication à faible puissance d'émission et communiquant par des liens radios multi-sauts caractérisés par leur instabilité et un fort taux de perte. Les nœuds utilisés présentent ainsi des caractéristiques uniques qui rend impossible la transposition des solutions et protocoles existant dans les réseaux traditionnels vers ce type d'environnement. Des efforts de standardisation ont conduit depuis quelques années à la conception de nombreux protocoles adaptés à l'environnement contraints des *motes* utilisées, mais des améliorations restent possibles.

7.1 Résumé des contributions

C'est dans ce contexte que s'inscrivent nos travaux, portant notamment sur des contributions au niveau du routage dans les RSCF et l'interconnexion de plusieurs réseaux de ce type à l'Internet traditionnel. Nous proposons deux nouvelles fonctions d'objectif RPL pour la construction de la topologie de routage. La première, décrite au chapitre 4 considère la prise en compte de la question énergétique dans le routage [Kam+13]. Le modèle d'énergie sur lequel nous avons fondé notre implémentation estime de façon assez précise la capacité énergétique résiduelle des nœuds comparés aux modèles linéaires classiques [Dun+07]. Pour cela, nous avons pris en compte l'*effet de récupération* qui se produit lors des phases d'inactivité des nœuds, représentant une proportion importante (99%) de la vie d'un capteur, en particulier lorsque certaines couches MAC optimisées comme ContikiMAC [Dun11] sont d'usage. Une comparaison du routage RPL basé sur la fonction d'objectif énergie avec l'implémentation répandue qui utilise l'ETX montre une amélioration considérable de la durée de vie du réseau (jusqu'à 14%).

Les exigences en matière de qualité de service des applications utilisant les RCSF peuvent varier de l'une à l'autre. Il est alors nécessaire de considérer en plus de l'énergie d'autres critères de performance. Nous avons étudié dans la littérature les moyens existant pour combiner plusieurs métriques, notamment dans le contexte du standard de routage RPL. Les seules formes de combinaison proposées se déclinent en composition additive et lexicographique [Kar+13], limitant ainsi la classe des métriques combinables. La logique floue constitue un bon moyen pour trouver un compromis intéressant entre plusieurs critères ne suivant pas nécessairement le même sens de variation. Par ailleurs, son implémentation n'a nécessité qu'une faible empreinte mémoire [Kam+14]. Ce qui s'avère judicieux pour nos capteurs très limités en capacité mémoire. Par ce moyen, nous avons combinés les métriques : énergie, délai et ETX. Les résultats obtenus des simulations et d'expériences sur des capteurs réels dans le bâtiment sont meilleurs que ceux d'un routage ne prenant en compte qu'une seule métrique. Ceci à la fois pour la durée de vie du réseau, la stabilité du routage, le taux de livraison des paquets et le délai de bout en bout [KND15].

Dans la dernière partie de cette thèse, nous avons pour objectif de concevoir une architecture permettant l'intégration de différents réseaux de capteurs sans fil à l'Internet. Cette intégration devrait permettre, non seulement la communication dans les deux sens, mais aussi la possibilité de l'adapter dynamiquement à la croissance du réseau. Nous avons modélisé le problème d'allocation des passerelles comme un programme linéaire en nombre entiers et réalisons sa résolution par un solveur CPLEX sur une station de gestion centralisé. Cela nous permet d'allouer efficacement les ressources nécessaires au réseau fédérateur en fonction des réels besoins. L'implémentation d'un prototype a été réalisée, celle-ci associe chaque réseau de capteurs à une instance RPL, les passerelles pouvant prendre en charge plusieurs instances, configurant dynamiquement le routage selon les besoins.

7.2 Perspectives

Comme perspectives à ce travail, concevoir une version distribuée de la station de gestion proposée pour le réseau fédérateur (chapitre 6) permettra de résoudre le problème de défaillance unique liée à une gestion centralisé. De nouvelles fonctionnalités devront également être rajoutées à la plate-forme développée pour assurer le *monitoring* des différents RCSF à travers le protocole nouvellement standardisé CoAP [SHB14].

Dans la combinaison par la logique floue proposée au chapitre 5, la même importance est accordée à chacune des métriques considérées. La possibilité d'adapter dynamiquement (pendant l'exécution) la contribution de chacune est possible par modification des bornes des différentes fonctions d'appartenance. Par ailleurs, le modèle utilisé pour les opérateurs de la logique floue est celui de Mamdani [Mam77], il serait intéressant d'implémenter d'autres modèles et de comparer les résultats obtenus avec ce dernier. De même, la T-norme utilisée se sert des opérateurs *minimum* et *maximum* comme opérateurs de composition et d'agrégation. Il existe de nombreuses autres familles de T-normes aussi intéressant à explorer.

Une contrainte forte au cours de notre travail a été la limitation de l'espace mémoire disponible pour conserver à la fois l'image système (système d'exploitation Contiki +

pile protocolaire) et l'ensemble des applications sur la plate-forme Telos B (uniquement 48 kilo-octets disponibles). Depuis lors, d'autres plate-formes matérielles pour RCSF disposant d'avantage de mémoire et processeurs plus puissant sont disponibles sur le marché à des prix comparables. Le Re-mote de Zolertia en constitue un bon exemple, avec 512 kilo-octets de flash ainsi que la possibilité d'étendre cette mémoire par ajout de carte SD. Il est également doté d'un processeur ARM Cortex-M3 avec 32 MHz de fréquence horloge. Porter les propositions développées dans cette thèse vers ces nouvelles plateformes, nous permet d'envisager de meilleures optimisations.

Bibliographie

- [AAJ09] U. ASHRAF, S. ABDELLATIF et G. JUANOLE. « Gateway Selection in Backbone Wireless Mesh Networks ». In : *IEEE Wireless Communications and Networking Conference (WCNC)*. Budapest, Hungary, avr. 2009, p. 1–6.
- [AB+12] I. AUGÉ-BLUM et al. « Capillary networks : a novel networking paradigm for urban environments ». In : *Proceedings of the first workshop on Urban networking (UrbaNe)*. Nice, France, déc. 2012, p. 25 –30.
- [Ana+09] G. ANASTASI et al. « Energy conservation in wireless sensor networks : A survey ». In : *Ad Hoc Networks* 7 (2009). Sous la dir. d'ELSEVIER, p. 537 –568.
- [Ash09] K. ASHTON. « That 'internet of things' thing ». In : *RFiD Journal* 22 (2009), p. 97–114. URL : <http://www.rfidjournal.com/articles/view?4986>.
- [AY05] K. AKKAYA et M. YOUNIS. « A survey on routing protocols for wireless sensor networks ». In : *Ad Hoc Networks* 3 (2005), p. 325–349.
- [Ban] *Body Area Network*. (accédé le 3 Sept. 2017). URL : https://fr.wikipedia.org/wiki/Body_Area_Network.
- [BBP10] A. BRANDT, J. BURON et G. PORCU. *Home Automation Routing Requirements in Low-Power and Lossy Networks*. RFC 5826 (Informational). Internet Engineering Task Force, avr. 2010. URL : <http://www.ietf.org/rfc/rfc5826.txt>.
- [Bei+15] N. BEIJAR et al. « Gateway Selection in Capillary Networks ». In : *Proceedings of 5th IEEE International Conference on the Internet of Things (IOT)*. Seoul, South Korea, oct. 2015, p. 90 –97.
- [Blu] *Bluetooth Core Specification*. (accédé le 1er Sept. 2017). URL : <http://www.ieee802.org/15/pub/TG4.html>.
- [CDG06] A. CONTA, S. DEERING et M. GUPTA. *Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification*. RFC 4443 (Draft Standard). Updated by RFC 4884. Internet Engineering Task Force, mar. 2006. URL : <http://www.ietf.org/rfc/rfc4443.txt>.
- [CYDM11] A. T. CABRERA, A. J. YUSTE-DELGADO et D. C. MACÍAS. « A Fuzzy Logic-Based and Distributed Gateway Selection for Wireless Sensor Networks ». In : *Proceedings of Highlights in Practical Applications of Agents and Multiagent Systems (PAAMS)*. Salamanca, Spain, avr. 2011, p. 243 –248.
- [DC+05] D. DE COUTO et al. « A High-Throughput Path Metric for Multi-hop Wireless Routing ». In : *ACM Journal of Wireless Networks* 11 (2005), p. 419–434.

- [DCAB02] D. DE COUTO, D. AGUAYO et A. BENJAMIN. « Performance of multihop wireless networks : Shortest path is not enough ». In : *Proceedings of the 1st ACM SIGCOMM Workshop on Hot Topics in Networks (HotNets-I)*. New Jersey, USA, oct. 2002.
- [DD09] I. DIETRICH et F. DRESSLER. « On the lifetime of WSN ». In : *ACM : Transaction on Sensor Networks* 5 (1 fév. 2009).
- [Der+13] L. DERU et al. « Redundant Border Routers for Mission-Critical 6LoWPAN Networks ». In : *Proceedings of the Fifth Workshop on Real-World Wireless Sensor Networks*. Como, Italy, sept. 2013, p. 195–203.
- [DGV04] A. DUNKELS, B. GRONVALL et T. VOIGT. « Contiki - a lightweight and flexible operating system for tiny networked sensors ». In : *Proceedings of 29th Annual IEEE International Conference on Local Computer Networks (LCN)*. Tampa, FL, USA, nov. 2004, p. 455–462.
- [Doh+09] M. DOHLER et al. *Routing Requirements for Urban Low-Power and Lossy Networks*. RFC 5548 (Informational). Internet Engineering Task Force, mai 2009. URL : <http://www.ietf.org/rfc/rfc5548.txt>.
- [DPZ04] R. DRAVES, J. PADHYE et B. ZILL. « Routing in Multi-radio, Multi-hop Wireless Mesh Networks ». In : *Proceedings of the 10th ACM annual international conference on Mobile computing and networking (MobiCom)*. New York, USA, sept. 2004, p. 114–128.
- [Dul+03] S. DULMAN et al. « Trade-Off between Traffic Overhead and Reliability in Multipath Routing for Wireless Sensor Networks ». In : *Proceedings of Wireless Communications and Networking Conference (WCNC)*. New Orleans, USA, mar. 2003, p. 1918–1922.
- [Dun+07] A. DUNKELS et al. « Software-based On-line Energy Estimation for Sensor Nodes ». In : *Proceedings of the 4th workshop on Embedded networked sensors (EmNets'07)*. New York, USA, 2007, p. 28–32.
- [Dun11] A. DUNKELS. *The ContikiMAC Radio Duty Cycling Protocol*. Rapp. tech. ISBN 1100-3154. SICS, déc. 2011.
- [Eri+08] J. ERIKSSON et al. « Demo abstract : MSPSim - an extensible simulator for MSP430-equipped sensor boards ». In : *Proceedings of the 5th European Conference on Wireless Sensor Networks (EWSN 2008)*. Bologna, Italy, 2008.
- [Eva11] D. EVANS. *The Internet of Things : How the Next Evolution of the Internet Is Changing Everything*. Sous la dir. de Cisco Internet Business Solutions Group (IBSG). Cisco Press, 2011.
- [Fen+13] J. FENG et al. « An Optimum Gateway Discovery and Selection Mechanism in WSN and Mobile Cellular Network Integration ». In : *Proceedings of 8th International Conference on Communications and Networking in China*. Guilin, China, août 2013, p. 483–487.
- [Gan+01] D. GANESAN et al. « Highly-resilient, energy-efficient multipath routing in wireless sensor networks ». In : *ACM SIGMOBILE Mobile Computing and Communications Review* 5 (2001), p. 11–25.

- [GL12] O. GNAWALI et P. LEVIS. *The Minimum Rank with Hysteresis Objective Function*. RFC 6719. Sept. 2012. URL : <http://www.ietf.org/rfc/rfc6719.txt>.
- [HCB00] W. R. HEINZELMAN, A. CHANDRAKASAN et H. BALAKRISHNAN. « Energy-Efficient Communication Protocol for Wireless Microsensor Networks ». In : *Proceedings of the 33rd Hawaii International Conference on System Sciences*. Hawaii, USA, jan. 2000.
- [HD98] R. HINDEN et S. DEERING. *IP Version 6 Addressing Architecture*. RFC 2373 (Proposed Standard). Obsoleted by RFC 3513. Internet Engineering Task Force, juil. 1998. URL : <http://www.ietf.org/rfc/rfc2373.txt>.
- [HT11] J. HUI et P. THUBERT. *Compression Format for IPv6 Datagrams over IEEE 802.15.4-Based Networks*. RFC 6282 (Proposed Standard). Internet Engineering Task Force, sept. 2011. URL : <http://www.ietf.org/rfc/rfc6282.txt>.
- [HV12a] K. HEURTEFEUX et F. VALOIS. « Is RSSI a Good Choice for Localization in Wireless Sensor Network? » In : *Proceedings of 26th IEEE International Conference on Advanced Information Networking and Applications*. Fukuoka, Japan, mar. 2012, p. 732–739.
- [HV12b] J. HUI et JP. VASSEUR. *The Routing Protocol for Low-Power and Lossy Networks (RPL) Option for Carrying RPL Information in Data-Plane Datagrams*. RFC 6553 (Proposed Standard). Internet Engineering Task Force, mar. 2012. URL : <http://www.ietf.org/rfc/rfc6553.txt>.
- [Ibm] IBM ILOG CPLEX Optimizer 12.6.3. (accédé le 3 Sept. 2017). URL : <https://www.ibm.com/us-en/marketplace/ibm-ilog-cplex>.
- [Int+03] C. INTANAGONWIWAT et al. « Directed Diffusion for Wireless Sensor Networking ». In : *IEEE/ACM Transactions On Networking (TON)* 11 (2003), p. 2–16.
- [Jia+07] X. JIANG et al. « Micro power meter for energy monitoring of wireless sensor networks at scale ». In : *Proceedings of 6th International Conference on Information Processing in Sensor Networks (IPNS'07)*. Massachusetts, USA, 2007, p. 186–195.
- [Kam+13] P. KAMGUEU et al. « Energy-based Metric for the Routing Protocol in Low-power and Lossy Network ». In : *Proc. of 2nd Sensornets*. Barcelona, Spain, fév. 2013.
- [Kam+14] P. KAMGUEU et al. « Fuzzy-based routing metrics combination for RPL ». In : *Doctoral Consortium SENSORNETS*. Lisboa, Portugal, jan. 2014.
- [Kar+13] P. KARKAZIS et al. « Evaluating routing metric composition approaches for QoS differentiation in low power and lossy networks ». In : *Trans. of Wireless Networks* 19 (août 2013), p. 1269–1284.
- [Kau+14] A. KAUSAR et al. « Energizing wireless sensor networks by energy harvesting systems : Scopes, challenges and approaches ». In : *Renewable and Sustainable Energy Reviews* 38 (2014). Sous la dir. d'ELSEVIER, p. 973–989.

- [KK00] B. KARP et H. T. KUNG. « GPSR : Greedy perimeter stateless routing for wireless sensor networks ». In : *Proceedings of the 6th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom)*. Boston, USA, août 2000, p. 243–254.
- [KMS07] N. KUSHALNAGAR, G. MONTENEGRO et C. SCHUMACHER. *IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs) : Overview, Assumptions, Problem Statement, and Goals*. RFC 4919 (Informational). Internet Engineering Task Force, août 2007. URL : <http://www.ietf.org/rfc/rfc4919.txt>.
- [KND15] P. KAMGUEU, E. NATAF et T. DJOTIO. « On Design and Deployment of Fuzzy-based Metric for Routing in Low-Power and Lossy Networks ». In : *IEEE SenseApp 2015*. Clearwater Beach, FL, USA, oct. 2015.
- [Kov11] M. KOVATSCH. « Demo Abstract : Human–CoAP Interaction with Copper ». In : *Proceedings of the 7th IEEE International Conference on Distributed Computing in Sensor Systems (DCOSS)*. Barcelona, Spain, juin 2011, p. 1–2.
- [Kuo95] F. KUO. « The ALOHA System ». In : *ACM Computer Communication Review* (1995).
- [Lev+11] P. LEVIS et al. *The Trickle Algorithm*. RFC 6206 (Proposed Standard). Internet Engineering Task Force, mar. 2011. URL : <http://www.ietf.org/rfc/rfc6206.txt>.
- [LXC14] T. H. LEE, X. S. XIE et L. H. CHANG. « RSSI-Based IPv6 Routing Metrics for RPL in Low-power and Lossy Networks ». In : *Proceedings of IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. San Diego, CA, USA, oct. 2014, p. 1714–1719.
- [Mam77] E. H. MAMDANI. « Application of fuzzy logic to approximate reasoning using linguistic synthesis ». In : *IEEE Transaction on Computing* C-26 (12 1977), p. 1182–1191.
- [Mar+10] J. MARTOCCI et al. *Building Automation Routing Requirements in Low-Power and Lossy Networks*. RFC 5867 (Informational). Internet Engineering Task Force, juin 2010. URL : <http://www.ietf.org/rfc/rfc5867.txt>.
- [MF10] F. MATTERN et C. FLOERKEMEIER. « From the Internet of Computers to the Internet of Things ». In : *Lecture Notes in Computer Science (LNCS)* 6462 (2010). Sous la dir. de SPRINGER, p. 242–259.
- [Mon+07] G. MONTENEGRO et al. *Transmission of IPv6 Packets over IEEE 802.15.4 Networks*. RFC 4944 (Proposed Standard). Updated by RFCs 6282, 6775. Internet Engineering Task Force, sept. 2007. URL : <http://www.ietf.org/rfc/rfc4944.txt>.
- [NF13] E. NATAF et O. FESTOR. « Online Estimation of Battery Lifetime for Wireless Sensors Network ». In : *Proc. of 2nd Sensornets*. Barcelona, Spain, fév. 2013.
- [OD06] F. OSTERLIND et A. DUNKELS. « Cross-Level Sensor Network Simulation with COOJA ». In : *Proc. of 31st IEEE Conf. SenseApp*. Tampa, Florida, 2006, p. 641–648.

- [Pan+01] D. PANIGRAHI et al. « Battery life estimation of mobile embedded systems ». In : *Proceedings of the the 14th International Conference on VLSI Design (VLSID'01)*. Washington, USA, 2001, p. 57–63.
- [Pfi+11] D. PFISTERER et al. « SPITFIRE : Towards a Semantic Web of Things ». In : *IEEE Communications Magazine* 49 (nov. 2011), p. 40–48.
- [Pis+09] K. PISTER et al. *Industrial Routing Requirements in Low-Power and Lossy Networks*. RFC 5673 (Informational). Internet Engineering Task Force, oct. 2009. URL : <http://www.ietf.org/rfc/rfc5673.txt>.
- [PNV13] N. A. PANTAZIS, S. A. NIKOLIDAKIS et D. D. VERGADOS. « Energy-Efficient Routing Protocols in Wireless Sensor Networks : A Survey ». In : *IEEE Communications Surveys & Tutorials* 7 (2 2013), p. 551 –591.
- [Rah+10] J. RAHME et al. « A recursive battery model for nodes lifetime estimation in wireless sensor networks ». In : *Proceedings of Wireless Communications and Networking Conference (WCNC)*. Sydney, Australia, avr. 2010, p. 1–6.
- [Ras] Raspberry Pi Foundation. (accédé le 20 Juin 2016). URL : <http://www.raspberrypi.org>.
- [Rom88] J.L. ROMKEY. *Nonstandard for transmission of IP datagrams over serial lines : SLIP*. RFC 1055 (Internet Standard). RFC. Fremont, CA, USA : RFC Editor, juin 1988. URL : <https://www.rfc-editor.org/rfc/rfc1055.txt>.
- [RST10] G. G. ROSARIO, G. STEFANO et L. TAVANTI. « A survey on multi-constrained optimal path computation : Exact and approximate algorithms ». In : *Computer Networks* 54 (17 2010), p. 3081 –3107.
- [RV03] D. RAKHMATOV et S. VRUDHULA. « Energy management for battery-powered embedded systems ». In : *Transaction of Embedded Computer System* 2 (3 août 2003). 2003, p. 277–324.
- [SHB14] Z. SHELBY, K. HARTKE et C. BORMANN. *The Constrained Application Protocol (CoAP)*. RFC 7252 (Proposed Standard). Internet Engineering Task Force, juin 2014. URL : <http://www.ietf.org/rfc/rfc7252.txt>.
- [She90] B. SHETTY. « Approximate solutions to large scale capacitated facility location problems ». In : *Applied Mathematics and Computation* 39 (2 1990), p. 159–175.
- [SL06] K. SRINIVASAN et P. LEVIS. « RSSI is Under Appreciated ». In : *Proceedings of 3rd Workshop on Embedded Networked Sensors (EmNets)*. Cambridge, MA, USA, mai 2006.
- [Sob08] J. SOBRINHO. « Network Routing with Path Vector Protocols : Theory and Applications ». In : *Proceedings of ACM SIGCOMM*. Karlsruhe, Germany, août 2008, p. 49 –60.
- [SWR98] S. SINGH, M. WOO et C. RAGHAVENDRA. « Power-aware routing in mobile ad hoc networks ». In : *Proceedings of the 4th Annual ACM/IEEE International Conference on Mobile Computing and Networking*. Dallas, TX, USA, oct. 1998, p. 181 –190.

- [SZ16] F. K. SHAIKH et S. ZEADALLY. « Energy harvesting in wireless sensor networks : A comprehensive review ». In : *Renewable and Sustainable Energy Reviews* 55 (2016). Sous la dir. d'ELSEVIER, p. 1041–1054.
- [TAGH02] S. TILAK, N. B. ABU-GHAZALEH et W. HEINZELMAN. « A Taxonomy of Wireless Micro-Sensor Network Models ». In : *Mobile Computing and Communications Review*, 6 (2 2002), p. 28–36.
- [Tel] Telos B CM5000. (accédé le 15 Sept. 2017). URL : <http://www.epssilon.cl/files/EPS5000.pdf>.
- [Tg4] IEEE Task Group 4 (TG4) – IEEE 802.15 WPAN Standard. (accédé le 15 Mars 2017). URL : <http://www.ieee802.org/15/pub/TG4.html>.
- [Thu12] P. THUBERT. *Objective Function Zero for the Routing Protocol for Low-Power and Lossy Networks (RPL)*. RFC 6552. Mar. 2012. URL : <http://www.ietf.org/rfc/rfc6552.txt>.
- [TNJ07] S. THOMSON, T. NARTEN et T. JINMEI. *IPv6 Stateless Address Autoconfiguration*. RFC 4862 (Draft Standard). Updated by RFC 7527. Internet Engineering Task Force, sept. 2007. URL : <http://www.ietf.org/rfc/rfc4862.txt>.
- [Vas+12] JP. VASSEUR et al. *Routing Metrics Used for Path Calculation in Low-Power and Lossy Networks*. RFC 6551. Mar. 2012. URL : <http://www.ietf.org/rfc/rfc6551.txt>.
- [VF14] O. VERMESAN et P. FRIESS. *Internet of Things Applications - From Research and Innovation to Market Deployment*. River Publishers, juin 2014.
- [WBA09] S. WAHARTE, R. BOUTABA et P. ANELLI. « Impact of Gateways Placement on Clustering Algorithms in Wireless Mesh Networks ». In : *Proceedings of IEEE International Conference on Communications (ICC)*. Dresden, Germany, juin 2009, p. 1–5.
- [WC96] Z. WANG et J. CROWCROFT. « Quality-of-service routing for supporting multimedia applications ». In : *IEEE Journal on Selected Areas in Communications* 7 (1996), p. 1228–1234.
- [WCD06] X. WU, G. CHEN et S. K. DAS. « On the Energy Hole Problem of Nonuniform Node Distribution in Wireless Sensor Networks ». In : *IEEE International Conference on Mobile Ad Hoc and Sensor Systems (MASS)*. Vancouver, Canada, oct. 2006, p. 180–187.
- [Win+12] T. WINTER et al. *RPL : IPv6 Routing Protocol for Low-Power and Lossy Networks*. RFC 6550. Mar. 2012. URL : <http://www.ietf.org/rfc/rfc6550.txt>.
- [WLY09] W. WU, J. LUO et M. YANG. « Gateway placement optimization for load balancing in wireless mesh networks ». In : *Proceedings of International Conference on Computer Supported Cooperative Work in Design (CSCWD)*. Santiago, Chile, avr. 2009, p. 408–413.
- [YBO09] B. YAHYA et J. BEN-OTHTMAN. « REER :Robust and Energy Efficient Multipath Routing Protocol for Wireless Sensor Networks ». In : *Proceedings of the 28th IEEE conference on Global telecommunications (GLOBECOM)*. Honolulu, USA, déc. 2009, p. 1–7.

- [YEG01] Y. YU, D. ESTRIN et R. GOVINDAN. *Geographical and Energy-Aware Routing : A Recursive Data Dissemination Protocol for Wireless Sensor Networks*. Rapp. tech. UCLA-CSD TR-01-0023. University of California Los Angeles (UCLA) Computer Science Department, mai 2001.
- [Yoo+10] H. YOO et al. « GLOBAL : a Gradient-based routing protocol for LOad-BALancing in large-scale wireless sensor networks with multiple sinks ». In : *Proceedings of IEEE Symposium on Computers and Communications (ISCC)*. Riccione, Italy, juin 2010, p. 556 –562.
- [YW03] Y. YANG et J. WANG. « Design Guidelines for Routing in Wireless Sensor Network ». In : *Proceedings of 27th IEEE INFOCOM*. Phoenix, USA, avr. 2003, p. 1615 –1623.
- [YWK05] Y. YANG, J. WANG et R. KRAVETS. *Interference-aware load balancing for multihop wireless networks*. Rapp. tech. UIUCDCS-R-2005-2526. University of Illinois at Urbana-Champaign, mar. 2005.
- [Zac+15] T. ZACHARIAH et al. « The Internet of Things Has a Gateway Problem ». In : *Proceedings of the 16th International Workshop on Mobile Computing Systems and Applications (HotMobile)*. Santa Fe, USA : ACM, fév. 2015, p. 27 –32.
- [ZCY04] J. ZHANG, B. CHEN et Y. YE. « A Multi-Exchange Local Search Algorithm for the Capacitated Facility Location Problem ». In : *Proceedings of the 10th International Integer Programming and Combinatorial Optimization Conference (IPCO)*. New York, USA : Springer, juin 2004.