Auto-configuration et auto-adaptation de réseaux de capteurs sans fil dans le contexte de la télémédecine

Julien Beaudaux

Numéro d'ordre : 2036bis

UNIVERSITÉ DE STRASBOURG

École Doctorale Mathématiques, Sciences de l'Information et de l'Ingénieur

THÈSE

présentée pour obtenir le grade de

Docteur de l'Université de Strasbourg
Discipline : Informatique

par

Julien Beaudaux

Auto-configuration et auto-adaptation de réseaux de capteurs sans-fil dans le contexte de la télémédecine

Membres du jury :

Directeur : Thomas Noël, Professeur à l'Université de Strasbourg

Co-encadrant : Antoine Gallais, Maître de Conférences à l'Université de Strasbourg

Rapporteur externe : André-Luc Beylot, Professeur à l'Université de Toulouse

Rapporteur externe : Fabrice Valois, Professeur à l'INSA de Lyon

Examinatrice : Nathalie Mitton, Chargée de Recherche à l'INRIA de Lille

Laboratoire ICube

CNRS UMR 7357

"L'amitié n'est pas moins mystérieuse
que l'amour ou l'une quelconque des
facettes de cette chose confuse qu'est la vie."

— Jorge Luis Borges (1899 - 1986)

REMERCIEMENTS

Le présent travail de thèse ne m'aurait très probablement pas été aussi agréable et fructueux sans l'appui plein et entier que m'ont apporté un certain nombre de personnes. Il m'a alors semblé indispensable de les remercier ici pour leur soutien.

En premier lieu, je tiens à remercier André-Luc Beylot, Nathalie Mitton et Fabrice Valois pour l'intérêt qu'ils ont montré dans mes travaux en acceptant d'en évaluer la qualité scientifique. Mes remerciements vont aussi à Antoine Gallais et Thomas Noël, pour l'investissement sans faille dont ils ont fait preuve dans mon encadrement tout au long de ces trois années de thèse. Leurs conseils et l'expérience qu'ils ont su me partager aura joué un grand rôle dans la qualité de ma démarche scientifique.

Je remercie aussi chaleureusement ma femme Maryna Bryyovska, mes parents Brigitte et Gérard, mes sœurs Valérie et Stéphanie, ainsi que Jordan, Thierry, Éric, Damien et Georgios pour leur présence, leur soutien indéfectible et la confiance toujours renouvelée qu'ils ont su m'apporter. Mes remerciements les plus sincère vont aussi à l'ensemble des membres de l'équipe Réseaux pour l'excellente ambiance qu'ils ont su créer et la considération qu'il m'ont apportée.

De même, mes pensées vont à Jean-Claude, dont la Nyckelharpa m'aura donné du fil à retordre, mais m'aura permis de vider mon esprit aux moments les plus critiques, ainsi qu'à Jean-Marc qui m'aura fait découvrir avec une joie sans cesse renouvelée le Tango.

Enfin, ces travaux n'auraient jamais vu le jour sans les conseils avisés et les encouragements répétés de Jean Lorchat. Ma passion pour la recherche, ainsi que pour les mathématiques et l'informatique ne se serait probablement jamais révélée sans son intervention. Jamais je ne saurai lui exprimer suffisamment ma gratitude.

Es la historia de un amor,
Como no hay otro igual.
Que me hizo comprender,
Todo el bien todo el mal,
Que le dio luz a mi vida,
Apagandola después.
Ay, qué vida tan oscura, corazón,
Sin tu amor no viviré !

Voici l'histoire d'un amour,
Comme il n'en existe de semblables.
Qui m'a fait comprendre,
Tout le bon, tout le mal,
Qui illuminait ma vie,
Pour l'éteindre ensuite.
Oh, quelle vie sombre, mon cœur,
Sans ton amour je ne puis vivre !

— "Historia de un amor", Carlos Eleta Almaran

Par un Tango la magie a commencé, et depuis jamais ne s'est éteinte.
Par l'amour que tu m'as donné, ma vie a pris tout son sens.
Par ta présence à mes côtés, je vois l'avenir sans crainte.
Maryna, à toi, toujours...

TABLE DES MATIÈRES

1

INTRODUCTION GÉNÉRALE

"Mesure ce qui est mesurable et
rends mesurable ce qui ne peut l'être."

— Galilée (1564 - 1642)

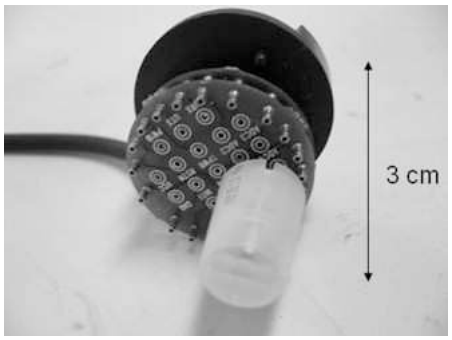
Depuis de nombreuses années, l'espérance de vie moyenne croît de manière constante dans les pays industrialisés (i.e. passant de 69,8 ans en 1960 à 81,6 ans en 2011 en France, selon la banque mondiale). Ainsi, la part de la population âgée de plus de 75 ans devrait être multipliée par trois d'ici à 2050. En parallèle, l'espérance de vie en bonne santé (i.e. sans incapacités) est en recul depuis 2006 (passant de 62,7 ans à 61,9 ans en France, entre 2008 et 2010 selon un étude de l'Ined). La part de la population en situation de dépendance tend donc à s'amplifier (+24,5% entre 2005 et 2020 selon les prévisions de l'Insee), alors que le nombre de places en maisons médicalisées n'évolue que faiblement (+3% entre 1996 et 2003 selon la Drees).

1.1 DES CAPTEURS POUR UN SUIVI MÉDICAL À DISTANCE

Partant de ce constat, de nouvelles solutions sont envisagées pour permettre un suivi médical à distance. Des équipements dénommés *nœuds capteurs* et dotés d'interfaces de capture peuvent alors être déployés directement au domicile de la personne. Ainsi, ils recueillent les informations physiologiques (e.g. poids, rythme cardiaque, oxygénation du sang, glycémie) et contextuelles (e.g. présence, position, température ambiante) nécessaires au suivi médical du patient. Quelques exemples de nœuds capteurs sont proposés en figure 1.

Actuellement, l'un des principaux freins au déploiement de ces capteurs au domicile de la personne vient des travaux d'installation et de câblage. En effet, ceux-ci sont particulièrement coûteux, complexes et ne peuvent être envisagés que pour certaines situations spécifiques (e.g. logements neufs ou en cours de rénovation) [3, 4]. De plus, les capteurs doivent être capables de fonctionner de manière autonome (i.e. sans nécessiter le remplacement des nœuds tombant en panne) dans un contexte dynamique, du fait des évolutions du logement (e.g.

(a) Un nœud TelosB doté d'un capteur de pression sanguine [1].



(b) Un nœud MICA2DOT doté d'un capteur de mouvement [2].

FIGURE 1 – Vue de quelques capteurs sans-fil.

déplacements de meubles, rénovations). Les solutions retenues pour le suivi des patients doivent donc être simples à installer, robustes et peu intrusives. Par conséquent, les capteurs sont généralement de petite taille, alimentés par batterie et dotés d'interfaces de communication sans-fil. Ainsi, ils peuvent être déployés n'importe où, sans nécessiter de travaux d'installation et fonctionner de manière autonome.

Néanmoins, cette situation amène de nouvelles contraintes. En particulier, les nœuds ne disposent que de ressources énergétiques limitées. Or, de tels déploiements peuvent s'étaler sur plusieurs mois selon l'application concernée, sans que les batteries des nœuds soient remplacées ou rechargées. Afin de limiter la consommation des capteurs, ceux-ci sont donc dotés de composants économes en énergie. Cela les amène à ne disposer que de ressources fortement limitées en termes de puissance de calcul ou de mémoire, et de n'être équipés que d'interfaces de communication à faible portée. Un comparatif des spécifications de trois capteurs typiques est proposé dans le tableau 1.

Par conséquent, les capteurs doivent s'organiser entre eux pour former un réseau. Comme leur radio ne leur permet de communiquer qu'avec une portée de quelques dizaines de mètres, ils doivent faire acheminer leurs mesures de proche en proche jusqu'à une entité dénommée *puits de collecte*. Celle-ci est capable de les traiter, les stocker, ou les transmettre via Internet au personnel médical. Cette communication multi-sauts est illustrée dans la figure 2.

Nœud capteur	**MICA2DOT**	**TelosB** [5]	**WSN430** [6]
Taille	$2,5\text{cm} \times 2,5\text{cm}$	$6,3\text{cm} \times 3\text{cm}$	$8\text{cm} \times 6,5\text{cm}$
Microprocesseur	Atmel ATmega 128L	TI MSP430	TI MSP430
Vitesse du processeur	16MHz	8MHz	8MHz
RAM	4Ko	10Ko	10Ko
Espace pour l'exécutable	128Ko	48Ko	48Ko
Interface radio	MICA2	TI CC2420	TI CC1101
Bande de fréquence	315/433/868/915MHz	$2400 - 2483\text{MHz}$	315/433/868/915MHz
Débit	$38,4\text{Ko/s}$	250Ko/s	$76,8\text{Ko/s}$
Portée en extérieur	$< 150\text{m}$	$< 100\text{m}$	$< 200\text{m}$
Batterie	Pile bouton	2 piles AA	Varta PoLiFlex
Prix unitaire	300$	100$	50 €
Déploiements télémédicaux	UbiMon [7]	CodeBlue [8] NETS-NOSS [1]	MOSAR [9] MPIGate [10]

TABLE 1 – Spécifications de quelques capteurs sans-fil.

1.2 APPLICATIONS

À ce jour, plusieurs projets ont mis en pratique l'usage des réseaux de capteurs sans-fil pour la télémédecine et l'assistance à domicile [11]. Les projets UbiMon [7], CodeBlue [8], MEDiSN [12] ou encore MOSAR [1] ont permis de démontrer qu'un tel réseau permettait d'effectuer une étude des pathologies ou un suivi de patients plus efficace, de manière non intrusive, et sans nécessiter de travaux d'installation.

En particulier, un réseau de capteurs complet a été déployé dans une unité de soins intensifs d'un hôpital dans le cadre du projet NeTS-NOSS subventionné par la NSF [1]. 46 patients étaient alors dotés de capteurs mesurant leur rythme cardiaque et leur oxygénation sanguine. Des capteurs étaient aussi déployés dans chaque chambre pour mesurer la température ambiante et apporter la connectivité entre les nœuds des patients et le puits de collecte.

Le travail détaillé dans le présent manuscrit s'inscrit dans le cadre du projet Dahlia [2], en partenariat avec la région Alsace. Ce projet ambitionne de faciliter le maintien des personnes âgées à leur domicile en proposant une solution de suivi médical à distance, facile à déployer et peu intrusive. Pour cela, Dahlia en-

1. Mosar - www.mosar-sic.org
2. Dahlia - www.portailtelesante.org/users_private/DAHLIA/Poster-DAHLIA.pdf

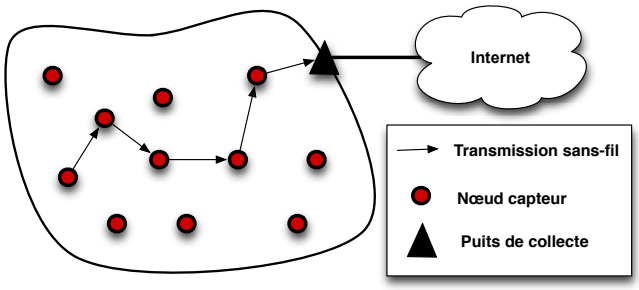


FIGURE 2 – Illustration des communications multi-saut dans les réseaux de capteurs sans-fil.

visage l'utilisation d'un réseau de capteurs sans-fil au domicile de la personne pour permettre la récupération d'informations physiologiques (e.g. poids au jour le jour) et contextuelles (e.g. présence, position). Ces données doivent alors être transmises directement au personnel médical afin d'être analysées et ainsi permettre un suivi à distance.

Plus généralement, le contexte de la télémédecine et de l'aide aux personnes dépendantes est riche en applications potentielles pour les réseaux de capteurs sans-fil. Nous présentons ici quelques-unes d'entre elles, ainsi que les conditions spécifiques qu'elles impliquent.

Suivi de patients Les réseaux de capteurs peuvent être employés par les centres médicaux pour surveiller les paramètres de santé de patients hospitalisés ou, dans un cadre plus large, nécessitant un suivi médical régulier [11, 1, 13]. Cette opération s'ajoute aux visites d'un personnel soignant, pour effectuer un complément de suivi plus régulier et à long terme, en particulier pour des patients évoluant dans un contexte peu ou non médicalisé (e.g. domicile de la personne). Un tel déploiement peut permettre de diagnostiquer plus rapidement certaines pathologies difficiles à détecter autrement (e.g. apnée du sommeil, tachycardie, bradycardie, œdème pulmonaire).
Dans ces situations, les données sont collectées de manière régulière par chaque capteur. Cela engendre un trafic dit orienté-temps. Plus rarement, les informations peuvent être collectées sur demande du personnel soignant. Dans ce cas, des requêtes sont envoyées à certains nœuds (e.g. ceux pouvant surveiller le paramètre requis), qui envoient leurs mesures en retour. Ce modèle de trafic est dénommé orienté-requêtes.

Détection d'incidents Un réseau de capteurs peut aussi servir d'avertisseur lorsqu'un problème majeur survient. Cela peut par exemple permettre de détecter la chute d'une personne âgée ou dépendante [14, 10], ou encore déclencher une alarme lorsqu'un paramètre physiologique dépasse un certain seuil (e.g. rythme cardiaque trop bas, taux de glycémie trop élevé) [12, 15]. Ainsi, le personnel soignant est averti plus rapidement, et peut donc intervenir plus efficacement. Dans ce contexte, les informations collectées sont envoyées lorsqu'un événement particulier survient. Ce modèle de trafic est donc appelé évènementiel.

Bâtiments intelligents Enfin, des réseaux de capteurs peuvent être employés pour faciliter les conditions de vie des personnes dépendantes [16, 17, 18]. Les nœuds récupèrent alors les informations environnementales des patients (e.g. température des chambres, trajet suivi par le patient) et contrôlent les divers équipements du lieu de vie (e.g. ouverture automatique du réfrigérateur, allumage de la lumière, mise en route de la ventilation et de la climatisation). Ce cadre est défini sous le terme de *domotique*, et implique potentiellement les trois modèles de trafic vus auparavant.

1.3 LIMITATIONS ET DÉFIS À RELEVER

Comme nous l'avons vu précédemment, des réseaux de capteurs ont déjà été déployés par le passé dans le cadre de la télémédecine. À partir de ces expériences, certaines contraintes majeures pouvant mettre en péril le succès du déploiement ont pu être identifées. Nous allons à présent les détailler, ainsi que les défis à relever pour les résoudre.

Les conditions dans lesquelles les réseaux de capteurs sans-fil évoluent peuvent varier significativement et de manière imprévisible au cours du déploiement. En effet, les liens radios utilisés sont instables et influencés par de nombreux paramètres extérieurs (e.g. conditions météorologiques [19, 20], activité humaine [21]). Certains capteurs peuvent aussi être mobiles [22] ou tomber en panne (e.g. épuisement des batteries, problème matériel). Enfin, la charge de trafic peut varier en fonction des nœuds et au cours du temps du fait de l'évolution de l'application (e.g. chute du patient).

Or, la gestion des communications dépend fortement des conditions dans lesquelles évoluent les nœuds (e.g. charge de trafic, propriétés de la topologie) et joue un rôle majeur dans la durée de vie du réseau. En effet, l'interface de communication est la plus gourmande en énergie [23]. Ainsi, un capteur WSN430 [6] maintenant sa radio allumée en permanence épuise ses ressources

en 44 heures contre 12 jours en ne l'allumant que 10% du temps. Pourtant, à l'heure actuelle, la plupart des réseaux de capteurs sont configurés de manière homogène (i.e. identique pour tous les nœuds) une fois pour toutes lors de leur déploiement, en fonction des conditions rencontrées initialement et des objectifs fixés par l'application (e.g. durée de vie, criticité des données) [1, 24, 25, 26].

Parallèlement, dans un tel contexte, des pannes peuvent se produire (e.g. disparition d'un lien, panne d'un nœud) et entraîner un partitionnement temporaire du réseau. De nombreux messages sont alors perdus. Ainsi, dans [1], plus de 40% des nœuds ont été concernés par une telle situation. Or, les données récoltées sont le plus souvent critiques. La perte d'une partie des mesures peut en effet entraîner un diagnostic incorrect ou encore empêcher la détection d'incidents majeurs (e.g. arrêt cardiaque, chute du patient).

Ainsi, des mécanismes de préservation énergétique et de tolérance aux pannes sont nécessaires pour rendre les réseaux de capteurs sans-fil plus robustes et adaptés aux applications médicales (e.g. suivi à domicile, suivi sur le long terme). La consommation énergétique des nœuds et la perte des messages étant liés aux conditions dans lesquelles le réseau évolue, nous nous sommes intéressés tout au long de nos travaux aux techniques permettant au réseau de s'adapter en conséquence. En particulier, nous avons développé des solutions permettant de configurer les nœuds de manière hétérogène et dynamique afin de diminuer leur consommation tout en préservant les tâches dévolues au réseau (i.e. la collecte et l'acheminement des mesures). Nous avons aussi mis en place un système permettant au réseau de s'adapter en cas de rupture temporaire.

1.4 ORGANISATION DU DOCUMENT

Ce manuscrit présente nos travaux sur l'auto-configuration et l'auto- adaptation des réseaux de capteurs sans-fil. Nous consacrons le chapitre 2 à la définition du contexte de recherche ainsi qu'à la présentation des travaux existants. Nous y présentons aussi les différents outils utilisés pour l'évaluation de nos contributions. Les chapitres 3, 4, 5 et 6 regroupent les contributions apportées au cours de cette thèse. Dans le chapitre 3, nous montrons jusqu'à quel point les techniques d'ordonnancement d'activité permettent d'accroître la durée de vie du réseau, en s'adaptant aux conditions de déploiement. Nous introduisons aussi un nouvel état d'activité, permettant de réduire davantage la consommation énergétique des nœuds concernés. Une approche pour donner vie à ces états d'activité est détaillée dans le chapitre 4. À chacun d'eux

est alors attribuée une configuration spécifique de la couche MAC. Dans le chapitre 5, un nouveau protocole MAC est proposé. Celui-ci permet d'adapter automatiquement et dynamiquement la configuration MAC des nœuds en fonction de la charge de trafic. Enfin, le chapitre 6 introduit un mécanisme de tolérance aux pannes. Celui-ci utilise les nœuds *Passifs* issus des techniques d'ordonnancement d'activité pour accroître la capacité de stockage du réseau. Ainsi, en cas de panne, les messages peuvent être conservés plus efficacement. Le chapitre 7 sera l'occasion de conclure ce travail et de dégager plusieurs pistes de recherche.

2

CONTEXTE DE RECHERCHE

"Certains livres se goûtent,
d'autres se dévorent, d'autres encore
se mâchent et se digèrent lentement."

— Sir Francis Bacon (1561 - 1626)

Sommaire

2.1 INTRODUCTION

Comme nous l'avons vu plus avant, les réseaux de capteurs sans-fil sont par nature instables et fortement dynamiques. En effet, les propriétés des liens radio (i.e. qualité et symétrie) évoluent avec le temps et les conditions environnementales [19, 20], et les capteurs peuvent tomber en panne ou être amenés à se déplacer. Ainsi, les nœuds doivent être capables de s'adapter automatiquement à leurs conditions de déploiement et aux dynamiques du réseau à tous les niveaux de la pile de communication (e.g. MAC, routage) pour rendre le déploiement plus performant et plus robuste. Ces dernières années, cette

thématique a bénéficié d'un grand intérêt de la part de la communauté scientifique [27, 28, 29].

Dans ce chapitre, nous détaillons le contexte dans lequel s'inscrivent nos contributions. Plus précisément, nous introduisons les concepts d'ordonnancement d'activité, de contrôle d'accès au médium radio et de tolérence aux pannes. De plus, nous donnons les références et présentons l'état de l'art permettant de se situer précisément dans chacun de ces thèmes, dans l'optique de l'auto-adaptation et de l'auto-configuration du réseau. Enfin, nous présentons les différents outils utilisés pour l'évaluation de nos contributions.

2.2 ORDONNANCEMENT D'ACTIVITÉ

2.2.1 *Définition de la problématique*

Comme nous l'avons vu dans le chapitre précédent, les capteurs disposent généralement de ressources énergétiques limitées. Or, les déploiements sont typiquement prévus pour s'étendre sur plusieurs mois, voire des années, sans que les batteries des nœuds ne soient remplacées ni même rechargées (i.e. de manière autonome) [1, 18]. Il est donc essentiel de préserver les ressources énergétiques du réseau, en limitant la consommation des nœuds. Plusieurs approches sont possibles pour y parvenir, la plus directe étant d'exploiter la forte densité de la plupart des réseaux de capteurs [30] pour ordonnancer l'activité des nœuds.

Pour fonctionner, les mécanismes d'ordonnancement d'activité doivent attribuer à chaque nœud du réseau un état d'activité spécifique (i.e. *Actif* ou *Passif*). Ce partage est effectué en garantissant que la topologie résultante, formée par les nœuds *Actifs*, sera à même de remplir les tâches dévolues à l'application. Ces tâches sont représentées par le critère applicatif, et sont choisies par l'utilisateur en fonction des objectifs du déploiement. Des exemples de critères applicatifs et quelques déploiements correspondants sont proposés dans le tableau 2.

Comme de nombreuses recherches antérieures [34, 35, 36], nous avons retenu le critère applicatif de couverture de surface pour la suite de nos travaux. En effet, une majorité des déploiements de réseaux de capteurs nécessitent la surveillance d'une zone d'intérêt [18, 31, 37, 17]. Dans ce cas de figure, les mesures effectuées par chaque nœud (e.g. tempéature, humidité) sont supposées significatives en deçà d'une certaine distance (i.e. le rayon de couverture). Ainsi, chaque capteur est à même de couvrir une partie de la zone d'intérêt (i.e.

Critère applicatif	Définition	Exemples d'usages
Couverture de surface	Chaque point de la zone d'intérêt doit être couvert par au moins un nœud	- Détecter la chute des patients [14, 10] - Suivre l'activité des patients [18, 30] - Détecter une intrusion [31]
Contrôle de densité	L'utilisateur détermine la densité minimale du réseau (nombre moyen de voisins)	- Contrôler les retransmissions pour le routage opportuniste [32] - Contrôler la précision des mesures [33]

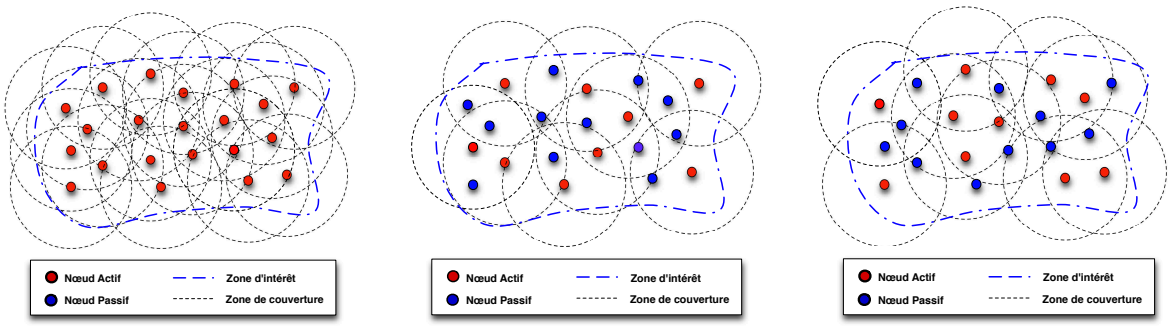
TABLE 2 – Exemples de critères applicatifs et de déploiements correspondants.

sa zone de couverture). Dans ce contexte, l'activité des nœuds est ordonnancée de sorte que la couverture totale de la zone d'intérêt (la zone nécessitant d'être surveillée par les capteurs) est effectuée par le sous-ensemble des nœuds *Actifs*, tandis que les *Passifs* économisent leur énergie.

Les états d'activité peuvent être déterminés une fois pour toute jusqu'à épuisement des nœuds *Actifs* ou re-calculés à intervalle régulier. Cette dernière approche permet d'adapter automatiquement la topologie en fonction du critère applicatif tout en augmentant la durée de vie du réseau. En effet, chaque nœud alterne alors entre un état *Actif* et *Passif* au cours du déploiement. Cela permet de répartir équitablement la charge des nœuds en les utilisant ou les préservant tour à tour, suivant leur état d'activité. Cette opération se fait sans affecter la collecte et la remontée des informations au puits (les critères applicatifs et la connectivité étant préservés). La figure 3 nous montre que l'ordonnancement d'activité permet de garantir une surveillance exhaustive de la zone d'intérêt par un sous-ensemble seulement des nœuds déployés. Elle illustre aussi le re-calcul des états d'activité permettant d'accroître la durée de vie du réseau.

2.2.2 *Travaux connexes*

Le problème de l'odonnancement d'activité dans les réseaux de capteurs sans-fil a donné lieu à de nombreuses études. Des solutions centralisées, distribuées et localisées ont été proposées. Nous détaillons à présent les principales solutions proposées pour traiter ce problème. Nous nous concentrons ici sur les solutions garantissant la connectivité du réseau, et permettent donc de maintenir à tout moment la collecte et l'acheminement des mesures vers le puits.



(a) Déploiement d'un réseau de capteurs sur une zone d'intérêt à surveiller.

(b) Un sous-ensemble des nœuds permet une surveillance de la zone.

(c) Les états d'activité sont recalculés à intervalle régulier, afin de répartir équitablement la charge des nœuds.

FIGURE 3 – Ordonnancement d'activité avec le critère de la couverture de surface.

Yener *et al* [38] proposent une approche fondée sur un modèle de Markov pour ordonnancer l'activité des nœuds. Dans leur algorithme, chaque nœud prend sa décision par l'intermédiaire d'un problème d'optimisation non-linéaire prenant compte des critères de couverture de surface et de connectivité. Cette solution nécessite des calculs importants pour déterminer la probabilité qu'un nœud a de devenir *Passif*, et repose sur des informations relatives à la totalité du réseau. Les auteurs proposent d'y parvenir en plusieurs tours de décision, chaque nœud ajoutant à ses messages de contrôle les informations de ses voisins, puis des voisins de ses voisins, etc. Cela entraîne des imprécisions dans le calcul des sous-sensembles de nœuds dans les premiers tours (et donc des pertes de connectivité), et empêche de traiter efficacement les changements topologiques (e.g. mobilité, défaillances).

Gupta *et al* proposent dans [35] un mécanisme distribué nommé DGA (pour Distributed Greedy Algorithm), pour déterminer le sous-ensemble de nœuds *Actifs*. Cette approche transpose le comportement des algorithmes d'ordonnancement d'activité centralisés d'une manière distribuée. Chaque tour, un nouveau nœud récupère ainsi les informations de tous les autres capteurs, décide de l'ordonnancement pour ce tour, et diffuse sa décision. Cette solution requiert un grand nombre de messages (i.e. 8 messages par nœud en moyenne), mais obtient de bonnes performances dans les cas considérés lors de son évaluation.

Deux mécanismes d'ordonnancement d'activité, dénommés Positive-Only (PO) et Positive-Retreat (PR), sont proposés dans [34]. Avec PO, chaque nœud évalue sa couverture et la connectivité de ses voisins *Actifs* après un délai aléa-

toire. S'il devient *Actif* à l'issue de ce processus, il envoie un message de notification à ses voisins. Ces derniers peuvent alors mettre à jour leurs informations pour leur propre décision d'activité. PR ajoute à cela une seconde décision, lui permettant de se retirer (i.e. de devenir *Passif*). Celle-ci est effectuée à partir des informations complètes du voisinage, après un second délai aléatoire. Ces deux mécanismes fonctionnent donc de manière entièrement localisée et nécessitent au plus un message par nœud *Actif* pour PO et deux pour PR. Cependant, la perte de messages de contrôle peut perturber la décision d'activité, et faire entrer en mode *Passif* un nœud nécessaire à la préservation du critère applicatif et de connectivité.

Certaines recherches considèrent aussi ce problème sous un angle différent. Dans [36], les nœuds adaptent leur portée de communication et leur rayon de couverture, de manière à préserver les critères applicatifs et de connectivité. Pour cela, chaque nœud calcule localement le diagramme de Voronoï (e.g. comme proposé dans [39]) et le graphe RNG [40] (pour Relative Neighborhood Graph) a partir de la topologie composée de ses voisins et de lui même. La portée choisie est alors celle de la plus longue arête adjacente du RNG ou la distance maximale au bord d'une cellule de Voronoï. Ainsi la couverture du réseau reste complète et la connectivité est préservée. Néanmoins, le calcul local du diagramme de Voronoï est une opération complexe. De plus, cette solution a été évaluée dans un environnement idéal, sans collisions ni pertes. Il est donc difficile d'en mesurer l'intérêt pour de véritables déploiements.

Ces techniques d'ordonnancement d'activité sont destinées à réduire la consommation énergétique du réseau en faisant entrer un sous-ensemble de nœuds dans un mode *Passif*, économe en énergie. Néanmoins, peu d'études se sont intéressées à la manière dont cet état pouvait être traduit dans la réalité.

Dans [41], une approche réaliste est proposée pour donner vie au mode *Passif*. Précisément, elle suggère d'adapter la couche applicative en fonction des états d'activité. Ainsi, les nœuds *Passifs* diminuent de moitié la fréquence d'envoi de leurs mesures, en direction du puits. Cette méthode a pour avantage de rendre les capteurs *Passifs* joignables, et ne nécessite aucun mécanisme de réveil synchronisé. De plus, en réduisant la charge de trafic, cette approche améliore les performances du réseau (e.g. latence, occupation du canal). Cependant elle n'a qu'un effet limité sur la consommation énergétique des nœuds concernés, cette dernière résultant principalement de l'écoute passive de la radio (c.f. section 2.3). Ainsi, elle ne remplit pas pleinement les besoins des mécanismes d'ordonnancement d'activité (i.e. réduire la consommation des nœuds *Passifs*) et n'est donc pas adaptée aux situations considérées dans nos travaux.

D'autres solutions [34, 35] suggèrent d'éteindre complètement les nœuds *Passifs* jusqu'au prochain tour de décision d'activité. Cette approche est optimale en terme de consommation énergétique, tous les composants électroniques auxiliaires (e.g. capteurs, radio) étant éteints. Cependant cette approche nécessite un mécanisme de réveil synchronisé, de sorte que tous les nœuds redeviennent *Actifs* simultanément pour le prochain tour de décision d'activité. Sinon, cette dernière serait faite sur la base d'informations incomplètes, ce qui pourrait provoquer d'importantes contre-performances dans le processus d'ordonnancement d'activité. Une telle synchronisation peut s'avérer complexe à mettre en place et à maintenir. De plus, sa complexité croît avec la taille du réseau, cette opération devant être effectuée de manière distribuée. Cette solution ne convient donc que pour des déploiements de taille limitée.

Yang *et al* étudient dans [42] plusieurs procédés permettant d'adapter les couches MAC et routage simutanément pour les nœuds *Passifs*, avec ou sans coordination entre ces deux couches. Cette approche affiche des performances prometteuses en simulation, en améliorant le délai moyen des messages et en diminuant le taux de perte. Cependant, l'impact de ces solutions sur la consommation énergétique des nœuds concernés n'a pas été évalué. Ce paramètre étant de première importance dans le cadre de l'ordonnancement d'activité (ceux-ci visant le plus souvent à accroître la durée de vie du réseau), l'intérêt de cette approche ne peut pas être clairement défini en l'état.

2.3 PROTOCOLES DE CONTRÔLE D'ACCÈS AU MÉDIUM

En plus des techniques d'ordonnancement de l'activité des nœuds, il est possible de réaliser des gains énergétiques à partir des autres couches de la pile de communication. En particulier, la gestion des communications est responsable de la majeure partie de la consommation d'un capteur sans-fil [43], le composant radio en étant l'élément le plus gourmand en énergie [23]. Or, cette gestion est assurée par la couche de contrôle d'accès au médium (Medium Access Control ou MAC). De nombreuses approches ont été proposées pour limiter la consommation énergétique des communications, tout en préservant des performances suffisantes pour permettre un acheminement efficace des données (e.g. délais, taux de perte). Une étude fournissait ainsi en 2010 une liste non exhaustive de plus de 70 protocoles MAC pour les réseaux de capteurs [44].

Ainsi, ces protocoles MAC cherchent à maintenir la radio en veille aussi souvent que possible. En dehors de toute transmission, les nœuds alternent donc entre des phases d'écoute et de veille. Ce cycle, dénommé *cycle de travail* (ou duty-cycle), doit être effectué tout en fournissant un accès fiable et effi-

cace au médium, tant en termes de latence que de débit. Le canal radio étant une ressource partagée, à diffusion (toute donnée transmise sera reçue par l'ensemble des voisins de l'émetteur, ce phénomène est dénommé sur-écoute lorsqu'il concerne un nœud à qui le message n'est pas destiné) et half-duplex (un nœud ne peut pas transmettre et recevoir simultanément), ces conditions sont particulièrement difficiles à réunir.

Afin de coordonner les nœuds désirant communiquer entre eux tout en plaçant la radio en veille aussi souvent que possible, deux principaux procédés de contrôle d'accès au médium ont émergé, par l'intermédiaire des protocoles synchronisés ou à échantillonnage de canal.

2.3.1 *Protocoles MAC synchronisés*

Les protocoles MAC synchronisés reposent sur une synchronisation temporelle des nœuds (i.e. en allumant leur interface radio simultanément) pour leur permettre de communiquer. Ces protocoles peuvent être divisés en deux catégories suivant leur mode de fonctionnement.

D'une part, certains protocoles MAC synchronisés discrétisent la durée de déploiement en plusieurs intervalles, dénommés *créneaux*. Les nœuds attribuent alors à chaque créneau un rôle (i.e. émission, réception ou veille) en accord avec leurs voisins. L'organisation de ces créneaux se répète alors en suivant ce cycle fixe. Ce principe de fonctionnement est appelé *accès multiple à répartition dans le temps* ou TDMA [45]. Les protocoles MAC TRAMA [46] et TSMP [47] entrent dans cette catégorie.

La figure 4 propose un exemple de division du temps en créneaux pour trois nœuds utilisant une couche MAC TDMA. Chaque nœud dispose d'un créneau pour l'émission de ses messages, et un autre pour leur réception. Ainsi, pendant chaque créneau, un seul nœud est autorisé à émettre (le nœud 1 pour le premier créneau, le 2 pour le second créneau), et un seul autre à recevoir (le nœud 2 pour le premier créneau, le 3 pour le second créneau).

Cette approche permet d'éviter toute collision dans le réseau. En effet, dans un même voisinage, un seul nœud sera autorisé à émettre par créneau. Néanmoins, les protocoles TDMA nécessitent le maintien d'une synchronisation globale précise de l'ensemble du réseau, ainsi qu'une négociation entre les nœuds d'un même voisinage pour l'attribution des créneaux. Ces conditions, pour être remplies, nécessitent l'émission d'un nombre important de messages de contrôle, ce qui peut affecter les performances et la consommation énergétique de ces protocoles. De plus, les protocoles TDMA rencontrent des difficultés

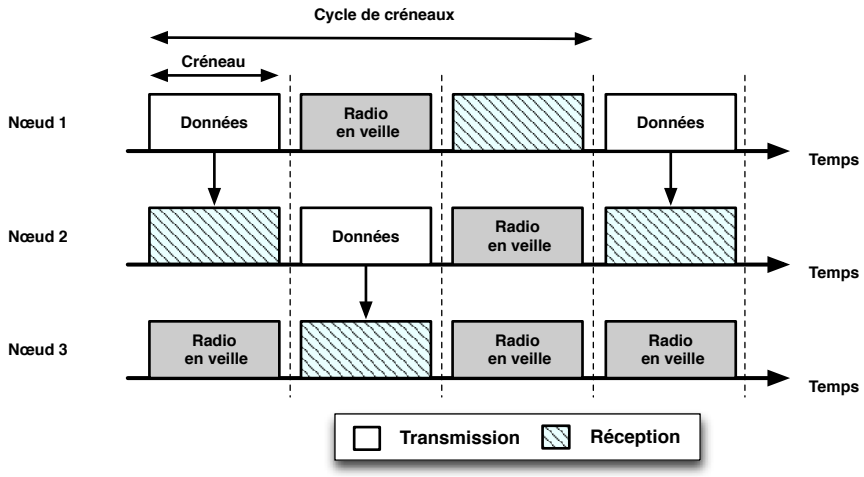


FIGURE 4 – Exemple de division temporelle dans les protocoles TDMA.

pour des déploiement à grande échelle (i.e. la synchronisation et l'attribution des créneaux étant d'autant plus difficile que le réseau est grand) et sont très sensibles aux changements topologiques (i.e. un changement des liens ou du voisinnage entraîne un re-calcul complet des créneaux, ainsi que la propagation de la décision).

D'autre protocoles reposent sur une synchronisation locale des nœuds uniquement. Les capteurs d'un même voisinage se réveilleront donc simultanément pour transmettre ou recevoir des données, mais de manière décorrélée du reste du réseau. Contrairement aux protocoles TDMA, les nœuds sont alors en compétition pour l'accès au médium radio lors des périodes d'activité, sur le modèle de la méthode CSMA/CA. Le fonctionnement de ces protocoles est illustré par la figure 5. Tous les nœuds d'un même voisinage (ici les nœuds 1 et 2) partagent des périodes d'activité identiques. Lorsque l'un d'eux veut transmettre un message, il utilisera alors le mécanisme CSMA/CA pour s'assurer que le canal est disponible. Les protocoles TDMA les plus connus reposant sur ces principes sont S-MAC [48], T-MAC [49] et le standard IETF 802.15.4.

Le remplacement de la synchronisation globale par une synchronisation locale permet de réduire significativement le nombre de messages de contrôle nécessaires. De plus, elle permet de s'adapter plus rapidement aux changements topologiques (e.g. mobilité des nœuds, disparition de certains liens), sans engendrer de diffusion de l'information à l'ensemble du réseau. Néanmoins, cette dernière reste complexe, les nœuds devant être en phase avec tous leurs voisins. De plus, contrairement aux protocoles TDMA, des collisions peuvent survenir, du fait de la compétition pour l'accès au médium radio.

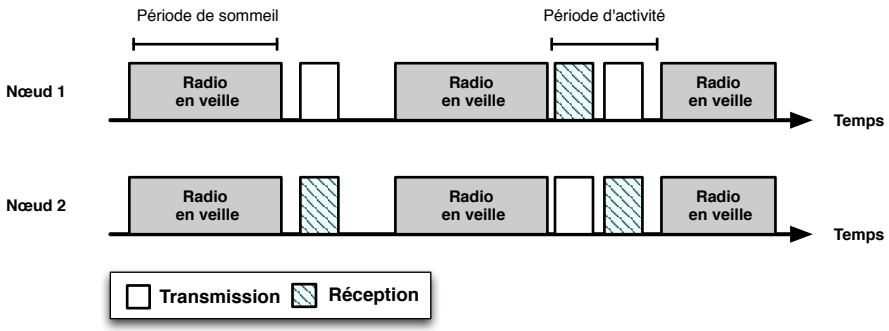


FIGURE 5 – Exemple de transmission avec des nœuds partageant des phases d'activité et de sommeil communes.

2.3.2 *Protocoles MAC à échantillonnage du canal*

D'autres protocoles MAC ont été conçus afin d'éviter toute synchronisation entre les nœuds du réseau, et apporter plus de robustesse aux communications (en diminuant l'impact causé par les changements topologiques) [50]. Dans ces protocoles, les nœuds du réseau s'assurent de l'émission d'un paquet leur étant destiné en échantillonnant cycliquement le canal radio de manière asynchrone et indépendante. Entre deux échantillonnages, ils peuvent alors placer leur interface radio en veille. Cette approche est dénommée LPL (pour Low-Power-Listening).

L'une des premières couches MAC à suivre ces principes fut B-MAC [51]. Dans le mécanisme LPL, les nœuds souhaitant transmettre une information doivent émettre une série de symboles, aussi appelée préambule. La longueur de cette dernière doit nécessairement être supérieure à la période d'échantillonnage, afin de s'assurer que le nœud destinataire est éveillé et prêt à recevoir. Dans B-MAC, lorsqu'un nœud entend un préambule lors de son échantillonnage de canal, il allume sa radio jusqu'à la réception de la donnée. L'identifiant du destinataire n'est contenu que dans le paquet de données. Ainsi, un nœud doit attendre sa réception complète pour savoir si le message lui est destiné. Cela force tous les nœuds d'un même voisinage à réceptionner la donnée, et implique donc une sur-écoute importante.

Par la suite, d'autres protocoles MAC à échantillonnage du canal ont vu le jour pour limiter cette sur-écoute. En particulier, X-MAC [52] propose de diviser le préambule en plusieurs micro-trames, chacune contenant l'identifiant du destinataire. Ainsi, les nœuds recevant l'une d'elles peuvent immédiatement déterminer si le message leur est destiné et éventuellement reprendre leur cycle de travail initial. De plus, deux micro-trames consécutives sont sé-

parées par une courte période d'écoute, permettant au destinataire d'accuser réception directement du préambule. Le préambule prend alors fin pour laisser place à l'émission de la donnée, et ainsi limiter l'occupation du canal radio.

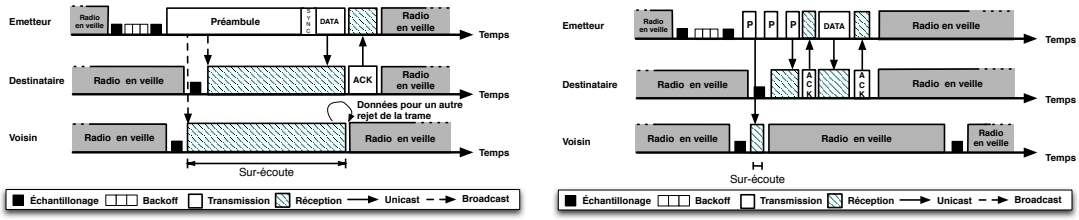
Récemment, une étude [53] a été menée par simulation sur un réseau de 5 à 40 nœuds afin d'estimer et comparer les performances des protocoles MAC synchronisés ou à échantillonnage du canal. Pour cela, les protocoles S-MAC [48] et X-MAC [52] ont été choisis pour représenter chaque approche. Il a ainsi pu être démontré que, avec une configuration équivalente (i.e. cycle de veille similaire), X-MAC et S-MAC affichaient des performances proches en termes de débit (500 octets par seconde avec un cycle de veille de $100ms$), de délai ($10ms$ en moyenne) et de consommation énergétique ($0,01W$ dans les deux cas), sans nécessiter de synchronisation temporelle ni de négociation entre les nœuds.

Enfin, plusieurs protocoles MAC ont été développés pour optimiser le fonctionnement d'X-MAC. Par exemple, BoX-MAC [54] et Contiki-MAC [55] proposent notamment de remplacer les micro-trames par la donnée elle-même. Cette méthode permet de réduire davantage l'occupation du canal et les délais, mais n'est envisageable que pour des données de taille limitée (e.g. 16 octets au plus avec un cycle d'échantillonnage de $100ms$). Avec une taille de données plus importante, la taille des micro-trames dépasse la moitié du préambule. Il devient alors impossible d'émettre plusieurs micro-trames, et il est donc plus avantageux d'utiliser X-MAC.

Le fonctionnement des couches MAC à échantillonnage du canal est illustré dans la figure 6, par l'intermédiaire des protocoles B-MAC et X-MAC.

Dans ces protocoles, la fréquence déchantillonnage a un impact très important sur les performances des communications. Alors qu'une période d'échantillonnage du canal plus élevée permet aux nœuds de diminuer leur consommation énergétique, elle réduit pareillement la quantité de trafic qu'ils sont capables de traiter. Réciproquement, une période d'échantillonnage plus courte permet aux nœuds en bénéficiant d'être capables de traiter plus de trafic, au prix d'une consommation énergétique plus élevée.

Les protocoles MAC à échantillonnage du canal fonctionnent de manière asynchrone et sans négociation entre les nœuds. Ainsi, ils passent facilement à l'échelle, et peuvent donc être utilisés pour des réseaux de toute taille. De plus, les performances de protocoles tels qu'X-MAC sont similaires à celles des protocoles synchronisés tels que S-MAC [53]. Pour toutes ces raisons, nous avons donc utilisé ce dernier comme base pour nos contributions.



(a) B-MAC (b) X-MAC

FIGURE 6 – Illustration du mécanisme de LPL dans B-MAC et X-MAC.

2.3.3 *Méthodes d'auto-adaptation de la couche MAC*

Comme deux nœuds disposant de configurations MAC différentes peuvent s'avérer incapables de communiquer entre eux, la plupart des déploiements de réseaux de capteurs sans-fil considèrent une configuration homogène (i.e. identique pour tout nœud du réseau) et statique (i.e. invariante au cours du déploiement) de cette couche [1, 24, 25, 26]. Bien qu'évitant toute partition du réseau, cette solution est loin d'être idéale. En effet, les performances des communications (e.g. délais, taux de perte, débit) et la consommation énergétique d'un nœud dépendent grandement de sa configuration MAC. Il pourrait donc être avantageux d'affecter dynamiquement à chaque nœud une configuration, selon la charge de trafic qu'il doit traiter.

De nombreuses études se sont penchées sur ce problème, afin de fournir à chaque nœud une configuration MAC qui lui est propre. Deux approches ont ainsi émergé pour y parvenir. D'une part, certains mécanismes se basent sur une adaptation par apprentissage des conditions au fur et à mesure du déploiement. Ainsi, dans WiseMAC [56], les nœuds apprennent le plan d'échantillonnage de leurs voisins par sur-écoute des transmissions. Ils peuvent alors émettre leurs données au moment précis où leur destinataire sera éveillé, et ainsi réduire la taille du préambule au strict minimum. Cette solution s'inspire donc des protocoles synchronisés, mais en évitant toute synchronisation temporelle explicite. Cette approche permet d'améliorer sensiblement les performances des protocoles MAC à préambule. Néanmoins, elle nécessite un trafic important pour obtenir les plans d'échantillonnage, et est très sensible aux changements topologiques (qui engendrent inévitablement une nouvelle phase d'apprentissage).

DutyCon [57] et pTunes [29] reposent sur la théorie du contrôle en utilisant des contraintes de performances définies par l'utilisateur (e.g. délai minimal, taux de pertes maximal, durée de vie attendue) et les flux de données cir-

culant dans le réseau pour définir une configuration MAC spécifique pour chaque saut. Les informations de chaque lien (e.g. charge de trafic, délais) sont ajoutées aux messages transmis au puits. Celui-ci les combine alors avec la liste des contraintes pour obtenir la configuration MAC la plus appropriée aux conditions actuelles du réseau, et diffuse celle-ci à l'ensemble des nœuds. Ces solutions sont dynamiques et ne nécessitent pas de période d'apprentissage longues, mais s'améliorent au contraire au fil du temps en affinant leurs informations. Cependant, elles fonctionnent de manière centralisée. De ce fait, bien que la remontée des informations de chaque lien se fasse par encapsulation dans les paquets de données, la diffusion des configurations MAC à l'ensemble des nœuds repose sur un mécanisme annexe synchronisé. Cela est particulièrement complexe, coûteux et peut en outre perturber le trafic applicatif.

D'autres protocoles ont été développés pour traiter les spécificités de certains types de déploiement et en tirer avantage. C'est notamment le cas pour Wang *et al* [58] qui proposent deux techniques permettant d'apprendre les périodes d'ensoleillement. La période d'échantillonnage des capteurs est alors adaptée en fonction des gains énergétiques obtenus par les panneaux solaires des nœuds en étant équipés (WSN-HEAP).

Parallèlement, des mécanismes et protocoles ont été développés pour attribuer une configuration MAC spécifique à chaque nœud sans nécessiter de phases d'apprentissage. Par exemple, Merlin *et al.* présentent dans [59] AADCC (pour Asymmetric Additive Duty Cycle Control), une méthode inspirée des mécanismes de temporisation de 802.11. Elle se sert du nombre de transmissions réussies consécutives pour adapter la configuration MAC des nœuds. Précisément, lorsqu'un certain nombre de transmissions réussies successives est atteint, la période d'échantillonnage du nœud récepteur est augmentée de 100ms de façon à diminuer sa consommation énergétique. A contrario, chaque échec de transmission d'un message entraîne la diminution de la période d'échantillonnage de 250ms, de manière à accroître les performances locales du réseau (et notamment la charge de trafic que le nœud est capable de traiter). Cette méthode permet de s'adapter efficacement aux variations progressives de charge de trafic dans le réseau. Cependant, elle suppose l'émission ou la perte au préalable de plusieurs paquets afin de déclencher une auto-adaptation de la couche MAC. Elle nécessite donc un trafic en permanence, et requiert un délai élevé pour parvenir à une configuration optimale de la couche MAC.

Dans [28], Kuntz *et al.* présentent BOX-MAC (pour Burst-Oriented X-MAC), un mécanisme d'auto-adaptation de la configuration MAC dédié aux change-

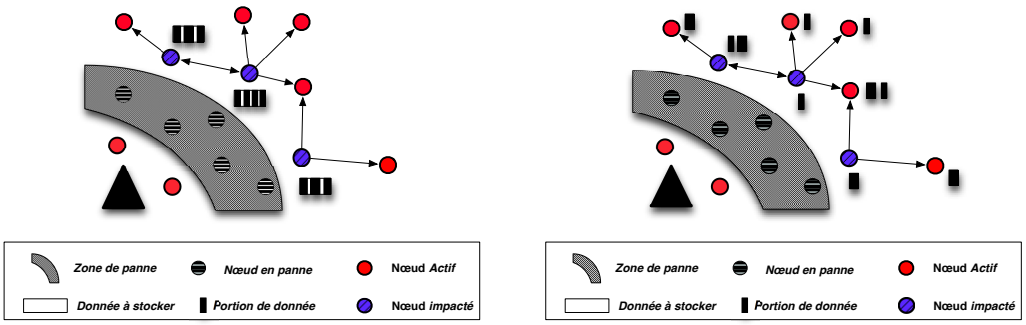
ments rapides de charge de trafic. BOX-MAC permet aux nœuds récepteurs de diminuer automatiquement leur période d'échantillonnage lorsque plusieurs messages sont transmis d'affilée. Pour cela, un nœud voulant émettre plusieurs paquets à la suite le stipulera via un champ dédié dans son premier message. Une fois ce message reçu, le récepteur diminuera sa période d'échantillonnage jusqu'à une valeur minimale pendant 10 secondes, indépendamment du nombre de messages à transmettre en rafale. Ainsi, selon la durée de la hausse de trafic, le nœud récepteur peut passer plus longtemps que nécessaire avec une courte période d'échantillonnage (et ainsi augmenter inutilement sa consommation énergétique), ou bien insuffisamment longtemps pour traiter la rafale dans son intégralité, et alors diminuer les performances pour les paquets restants. BOX-MAC ne permet donc pas une auto-adaptation précise en fonction des variations de trafic.

Nous avons présenté ci-dessus plusieurs techniques permettant au réseau de s'adapter automatiquement et dynamiquement en fonction des conditions dans lesquelles il évolue. Néanmoins, ces mécanismes sont conçus pour optimiser les performances du réseau (e.g. durée de vie, délais, occupation du canal radio) lorsque celui-ci est opérationnel (i.e. la collecte et la remontée des mesures de chaque nœud en direction du puits est assurée). Or, les réseaux de capteurs sans-fil ne sont pas à l'abri de pannes temporaires. les informations collectées par les capteurs pouvant s'avérer critiques (en particulier pour des applications médicales), des mécanismes de sauvegarde sont nécessaire, pour stocker les messages avant la reprise du réseau et ainsi éviter leur perte.

2.4 TOLÉRANCE AUX PANNES

Dans les réseaux de capteurs sans-fil, il n'est pas rare que le puits de collecte devienne temporairement inaccessible pour tout ou partie des nœuds. En effet, ceux-ci peuvent tomber en panne [24], et les liens radio (surtout sur une gamme de fréquences aussi faible) sont fluctuants et instables [19, 20]. De plus, de nombreuses applications médicales nécessitent l'emploi de nœuds mobiles (e.g. capteurs physiologiques sur le patient) [15, 22]. Or, dans les applications médicales en particulier, les mesures sont critiques, le diagnostic pouvant être compromis ou faussé par manque de données. Dans cette section, nous passons en revue différentes méthodes permettant au réseau de détecter ce phénomène et de limiter les pertes de messages en les stockant temporairement.

Plusieurs solutions ont été proposées pour détecter la perte de connectivité entre un nœud et son prochain saut [60, 61], reposant par exemple sur l'envoi régulier de messages de contrôle ou sur la définition d'un seuil de messages



(a) Les nœuds ne pouvant plus joindre leur prochain saut divisent les informations en plusieurs blocs de taille moindre.

(b) Ces blocs sont ensuite répartis entre les voisins pour y être stockés.

FIGURE 7 – Illustration du stockage de données par *erasure coding*.

successifs non acquittés (comme c'est le cas pour le mécanisme de détection des voisins inaccessibles d'IPv6 [62]). Certaines méthodes [63] permettent aussi de détecter plus rapidement les nœuds touchés par une panne lorsque celle-ci touche toute une partie du réseau. Lorsque la perte d'un premier capteur est détectée, d'autres messages de vérification sont envoyés aux nœuds aux alentours. Ainsi, cette solution peut localiser précisément et rapidement l'étendue et la localisation de la panne.

Une fois la panne détectée, les nœuds coupés du réseau doivent stocker temporairement les messages générés ou reçus jusqu'à ce qu'un autre chemin de routage soit trouvé ou que les nœuds en panne soient réparés [64, 65, 66]. Les capteurs ne disposant en général que d'un espace mémoire limité, des solutions doivent être mises en place pour distribuer le stockage entre eux. Pour y parvenir, des méthodes d'*erasure coding* sont employées, via des codes de Reed-Solomon [67], des codes fontaine [68] ou encore des filtres de Bloom [69]. Le fonctionnement général de ces solutions est illustré dans la figure 7. L'information est ainsi divisée en plusieurs blocs de taille réduite (comme pour les nœuds concernés dans la sous-figure 2.7(a)), qui seront ensuite distribués entre les voisins (comme illustré par la figure 2.7(b)). L'information est donc stockée de manière distribuée au sein d'un même voisinage. De plus, si une partie des nœuds tombe à son tour en panne, l'information peut être reconstruite à partir des blocs restants. Néanmoins, ces approches ne permettent de distribuer l'information que parmi les nœuds *Actifs* voisins. Ainsi elles ne peuvent accroître l'espace de stockage que de manière très limitée.

2.5 ENVIRONNEMENTS D'ÉVALUATION DE NOS CONTRIBUTIONS

Pour évaluer efficacement l'ensemble de nos contributions, nous avons mené des campagnes d'évaluations que nous décrivons à présent. Pour cela, nous avons eu recours tant à des simulations qu'à des d'expérimentations. Ces deux environnements d'évaluation présentent des propriétés et avantages distincts. Nous les avons donc utilisés en conjonction, afin de tirer le meilleur parti de chacun, et ainsi produire une évaluation de nos contributions aussi complète que possible.

Notons que chaque résultat est accompagné de son intervalle de confiance à 95%. Ainsi, la précision de nos résultats peut être appréciée directement.

2.5.1 *Environnements de simulation*

Les simulateurs mettent tout en œuvre pour faciliter l'obtention des informations relatives aux performances du réseau. Ils permettent donc de surveiller n'importe quel événement survenant dans le réseau de manière précise, sans affecter le fonctionnement de ce dernier. Nos simulations ont été effectuées par l'intermédiaire de deux simulateurs distincts : WSNet [70] et Cooja [71]. Tous deux sont des simulateurs à événements discrets. Ils proposent des modèles de propagation radio réalistes, nous permettant ainsi de tester nos contributions dans des conditions réalistes, tout en contrôlant finement chaque paramètre de simulation. Néanmoins, chacun d'eux présente des caractéristiques propres, qui le rendent plus approprié à une situation spécifique.

WSNet se distingue notamment par sa faculté à simuler des réseaux à très large échelle sans difficulté. Cela nous a notamment permis de pouvoir évaluer nos solutions d'ordonnancement d'activité et LIFT, dont le fonctionnement nécessite une forte densité. De plus, WSNet laisse un grand nombre d'outils à disposition des nœuds simulés, par exemple pour leur fournir leur consommation en temps réel ou encore leur permettre de modifier précisément la portée maximale de leurs communications (ce qui en réalité n'est pas possible, les radios typiques fonctionnant par paliers de puissance). Cela nous a notamment permis de modéliser le critère de couverture de surface (c.f. section 2.2.1), grâce aux coordonnées géographiques précises à disposition des capteurs.

Au contraire, Cooja ne permet pas de simuler des réseaux denses (i.e. plus de 20 voisins en moyenne). De plus, ce simulateur ne fournit aux nœuds aucun outil supplémentaire par rapport aux nœuds TelosB [5] qu'il cherche à émuler, réduisant d'autant le champ de possibilités pour les simulations. Cooja pré-

Paramètres de simulation	Valeur
Zone de déploiement	$50\text{m} \times 50\text{m}$
Densité du réseau	Jusqu'à 60 voisins en moyenne
Méthode de distribution	Processus ponctuel de Poisson d'intensité $\lambda > 0$
Modèle de couche MAC	X-MAC [52]
Portée radio maximale	10 mètres
Modèle de propagation radio	Affaiblissement du signal en Log-distance

TABLE 3 – Récapitulatif des paramètres de simulations.

sente néanmoins de nombreux avantages par rapport à WSNet. Tout d'abord, il utilise le système d'exploitation pour réseaux de capteurs Contiki OS [72]. Ainsi, le même code est utilisé tant avec Cooja que pour un déploiement sur une plateforme d'expérimentation. Cela facilite grandement les étapes de test et d'implémentation, et garantit la conformité des simulations par rapport aux expérimentations.

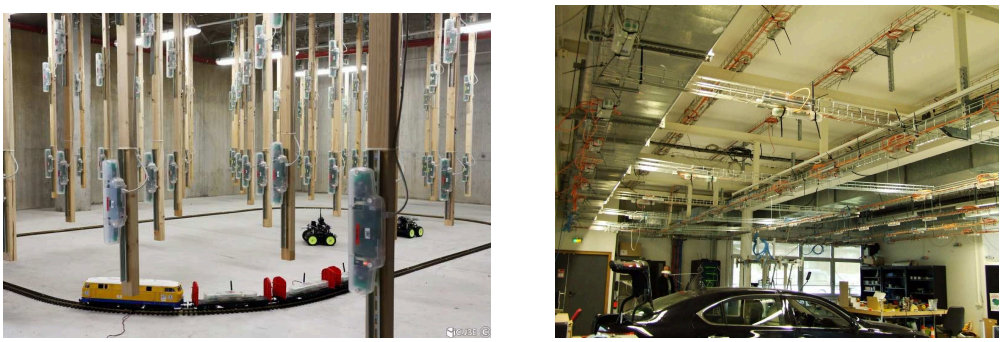
Un récapitulatif des paramètres de simulation utilisés tout au long de nos campagnes d'évaluation est fourni dans le tableau 3.

2.5.2 *Environnements d'expérimentation*

Nous avons aussi mené des campagnes d'expérimentation pour évaluer les performances de nos contributions en environnement réel. Pour cela, nous avons utilisé le système d'exploitation Contiki OS. Contiki a pour avantage de faciliter le passage du simulateur aux expérimentations, en utilisant le même code pour les deux. De plus, ce système d'exploitation a un faible impact sur l'application [73].

Lorsque nous voulions étudier, dans un premier temps, le comportement de nos solutions pour les couches les plus basses (e.g. MAC) et dans des conditions de déploiement optimales (i.e. sans compétition pour le canal radio et en minimisant les perturbations), nous avons eu recours à une plateforme composée de deux capteurs TelosB [5]. L'un agissait alors comme émetteur et l'autre comme récepteur. Nous avons pu ainsi observer précisément le comportement de ces couches, en isolant les facteurs extérieurs.

Ensuite, afin de tester nos contributions en environnement réel à plus large échelle (i.e. un réseau complet de plusieurs dizaines voire centaines de nœuds), nous les avons déployées sur deux des plateformes IoT-LAB [6]. IoT-LAB est



(a) La plateforme IoT-LAB de Strasbourg (b) La plateforme IoT-LAB de Grenoble

FIGURE 8 – Une vue de deux plateformes IoT-LAB.

composé de quatre plateformes de réseaux de capteurs, localisées à Strasbourg, Grenoble, Lille et Rennes. Par la suite, nous n'avons retenu que les plateformes de Strasbourg et Grenoble, leur radio permettant une meilleure configuration de la couche MAC (ce qui est un paramètre essentiel dans nos contributions). La plateforme de Strasbourg est composée de 240 nœuds statiques, déployés sous forme d'une grille 3D régulière de $10m \times 8m \times 3m$. Cette plateforme se situe en intérieur, relativement isolée des perturbations humaines (mais pas atmosphériques [19, 20]). Au contraire, la plateforme de Grenoble est composée de 256 nœuds fixes déployés de manière aléatoire sur une zone de $16m \times 14m \times 3m$. Cette plateforme se situe dans un local mécanique et est donc sujette aux perturbations dues aux activités humaines. Une vue de chaque plateforme est proposée dans la figure 8.

Du fait de leurs spécificités, ces deux plateformes sont complémentaires. En effet, elles permettent d'étudier le comportement de nos contributions dans des environnements différents (perturbations radio plus ou moins fortes, densité homogène ou non), et donc valider leurs propriétés générales, indépendamment du déploiement considéré.

Un récapitulatif des spécifications des plateformes IoT-LAB et des paramètres utilisés expérimentalement sont proposés dans le tableau 4.

2.6 CONCLUSION

Dans ce chapitre, nous avons introduit les principes généraux de l'ordonnancement d'activité, du contrôle d'accès au médium radio et de la tolérance aux pannes, sous la perspective de l'auto-adaptation et de l'auto-configuration. Nous avons aussi présenté l'état de l'art actuel pour chacune de ces théma-

Paramètres expérimentaux	**Valeur**
Nœuds déployés	240 capteurs statiques, en intérieur
Plateforme homogène (Strasbourg, France)	Grille 3D régulière de $10\text{m} \times 8\text{m} \times 3\text{m}$
Plateforme Hétérogène (Grenoble, France)	Nœuds déployés aléatoirement sur une zone de $16\text{m} \times 14\text{m} \times 3\text{m}$
Système d'exploitation	Contiki OS [72]
Modèle de couche MAC	X-MAC [52]
Modèle de couche routage	Gradient [74]
Paramètres matériels	**Valeur**
Processeur	Texas Instrument MSP430
Modèle d'interface radio	Interface Texas Instrument CC1101
Puissance d'émission	-20 dBm
Capacité de la batterie	880 mAh, 3.7 V

TABLE 4 – Spécification des plateformes IoT-LAB et paramètres expérimentaux.

tiques. Dans les chapitres suivants, nous exposerons certaines problématiques restant à traiter, et détaillerons nos solutions pour y parvenir.

3

AUTO-ADAPTATION POUR L'ORDONNANCEMENT D'ACTIVITÉ

"Ce n'est pas l'espèce la plus puissante qui survit, mais celle qui s'adapte le mieux au changement."

— Charles Darwin (1809 - 1882)

Sommaire

3.1 INTRODUCTION

Dans les réseaux de capteurs sans-fil, les déploiements sont réalisés de manière à produire une topologie dense (i.e. en disposant un nombre important de nœuds dans la zone d'intérêt), fortement redondante [30]. Cette approche se justifie par le fait que ces nœuds disposent de ressources énergétiques limitées. En effet, ils sont le plus souvent alimentés par batterie ou par l'intermédiaire de faibles dispositifs de collecte d'énergie [75]. La durée de vie du réseau peut alors être accrue en ordonnançant l'activité des nœuds. Dans ce contexte, seul un sous-ensemble de ces derniers, dénomés *Actifs*, participe activement à son fonctionnement. Les nœuds restants entrent ainsi dans un état dit *Passif*, économe en énergie. La durée de vie du réseau est alors accrue en homogénéisant

le temps de participation active des nœuds, de manière à les épuiser équitablement [76].

Dans le présent chapitre, nous montrons jusqu'à quel point ces techniques permettent d'accroître la durée de vie du réseau, tout en lui laissant la possibilité de réaliser les tâches lui étant dévolues (i.e. critère dicté par l'application comme par exemple la surveillance d'une zone d'intérêt), et sans risquer une partition du réseau (i.e. critère de connectivité). Nous montrons aussi qu'un sous-ensemble des nœuds *Actifs* n'est pas nécessaire aux opérations de routage (i.e. seulement à la garantie de ce critère applicatif). À partir de là, nous proposons l'introduction d'un état d'activité intermédiaire, à mi-chemin entre les états *Actif* et *Passif* déterminés par la plupart des mécanismes d'ordonnancement d'activité issus de la littérature (c.f. section 2.2). Ce nouvel état d'activité, dénommé *Sensing-Only*, permet aux nœuds concernés d'éteindre leur radio en dehors de leurs propres émissions de manière à limiter leur consommation énergétique. Parallèlement, il permet de réduire encore davantage la proportion de nœuds *Actifs*, et donc la consommation énergétique moyenne du réseau.

3.2 INTRODUCTION D'UN ÉTAT SENSING-ONLY

À ce jour, les mécanismes d'ordonnancement d'activité prennent en considération les critères applicatifs et de connectivité dans la sélection des nœuds *Actifs*, sans jamais les dissocier l'un de l'autre. De nombreux nœuds deviennent alors pleinement *Actifs*, sans pour autant participer à la remontée des informations jusqu'au puits [77]. Pourtant, du fait de leur statut, ces nœuds n'auront à transmettre que leurs propres données, sans jamais avoir à en recevoir.

Par conséquent, nous avons introduit un nouvel état d'activité, dénommé *Sensing-Only*. L'ensemble *Sensing-Only* est constitué de nœuds ne faisant que surveiller leur zone de couverture et transmettre leurs propres mesures. Ceux-ci ne participent donc pas au réseau, et ne font que l'utiliser. Ils peuvent donc accroître leur fréquence d'échantillonnage du canal, voire intégralement éteindre leur radio en réception si aucun trafic montant n'est considéré. Cela leur permet ainsi de réaliser d'importantes économies d'énergie, sans pour autant dégrader les performances du réseau (e.g. délais, taux de perte). En effet, dans les réseaux de capteurs, la réception d'un message est souvent aussi coûteuse que sa transmission et se produit bien plus souvent (du fait des densités considérées et des communications multi-sauts).

Le problème consistant à trouver le sous-ensemble maximal de nœuds *Actifs* superflus à la préservation du critère de connectivité peut être réduit au problème de l'arbre couvrant maximisant les feuilles (ou MLST pour Maximum Leaf Spanning Tree). Ce problème étant NP-complet [78], nous proposons ici plusieurs méthodes permettant d'obtenir une approximation locale du sous-ensemble de nœuds *Sensing-Only*.

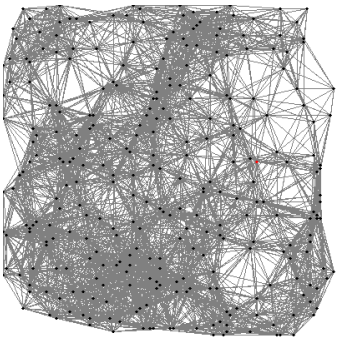
Une méthode pour parvenir à identifier ce sous-ensemble est de s'appuyer sur un graphe acyclique orienté (ou DAG pour Directed Acyclic Graph). De nombreux protocoles de routage dans les réseaux d'objets reposent sur la construction de telles structures [79, 80]. Le standard IETF RPL [81] appartient à cette catégorie, et repose sur un DAG pour décider des routes vers le puits dans le réseau. Dans de telles structures, le prochain saut d'un nœud en direction du puits est dénommé *père*, et un nœud ayant u pour *père* est appelé *fils* de u. Les nœuds n'ayant pas de *fils*, se situant donc à l'extrémité de la topologie virtuelle, sont dénommés *feuilles*. Ces derniers n'auront jamais à retransmettre d'information de leurs voisins, ne sont donc pas nécessaires à la garantie de la connectivité, et peuvent alors s'ajouter au sous-ensemble des nœuds *Sensing-Only*.

Lorsque la couche routage repose sur un DAG, il est donc possible d'identifier le sous-ensemble de nœuds *Sensing-Only* sans introduire de trafic de contrôle. Dans le cas contraire, un DAG virtuel doit être spécialement construit afin d'y parvenir. Nous avons retenu deux solutions permettant d'y parvenir, en raison leur faible surcoût en trafic de contrôle, leurs bonnes performances et leur réalisme.

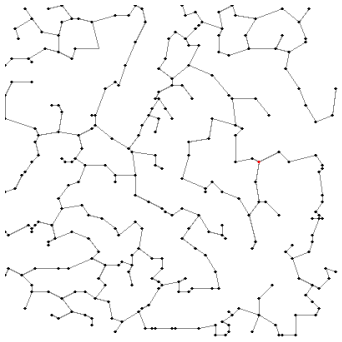
3.2.1 *Identification des nœuds Sensing-Only via un LMST*

Notre première approche pour identifier le sous-ensemble de nœuds *Sensing-Only* utilise une approximation locale de l'arbre couvrant de poids minimal (ou MST pour Minimum Spanning Tree). Le problème visant à trouver localement le MST, c'est à dire sans recourir à une décision centralisée à partir d'un nœud disposant de toutes les informations du réseau, est NP-complet [82]. Par conséquent, nous proposons d'en construire une approximation par l'intermédiaire de l'algorithme proposé dans [40] proposant les meilleures propriétés (e.g. construction locale, faible nombre de messages de contrôle) et dénommé LMST pour Local Minimum Spanning Tree.

Dans cette solution, chaque nœud u calcule le MST de la topologie formée par ses voisins directs et lui-même. Ensuite, u transmet un message à chacun



(a) Topologie physique initiale.



(b) Topologie virtuelle après construction du LMST.

FIGURE 9 – Construction d'un LMST.

de ses voisins v s'il existe un lien entre eux dans le MST local trouvé. Si tous deux s'accordent sur l'existence de cette arête, elle est alors ajoutée à la topologie résultante. Dans cette structure, chaque nœud n'ayant qu'un lien sortant (et donc un seul voisin dans la topologie virtuelle) est localisé à une extrémité du LMST, et n'est donc pas indispensable aux communications multi-sauts. Ces nœuds peuvent donc changer leur état et devenir *Sensing-Only* sans compromettre les communications multi-sauts. La construction du LMST a une complexité de O(n) en termes de messages de contrôle, équivalant ici à trois messages émis par nœud en moyenne. Un exemple de topologie physique et du LMST résultant sont donnés en figure 9, les nœuds aux extrémités de la topologie pouvant devenir *Sensing-Only*.

3.2.2 *Identification des nœuds Sensing-Only via un gradient*

Nous avons aussi considéré l'utilisation d'un gradient [74] afin d'identifier le sous-ensemble de nœuds *Sensing-Only*. La construction de cette structure s'opère en allouant à chaque nœud du réseau un rang et l'identifiant d'un nœud *père*. Le rang représente la distance minimale en nombre de sauts qui sépare le nœud du puits de collecte, alors que le *père* est un voisin direct de rang inférieur. Le puits initie la construction du gradient en diffusant son rang (i.e. 0) et son identifiant à son voisinage direct. Si un nœud u reçoit un de ces messages de construction et ne s'est pas vu attribuer de position dans le gradient, ou si son rang est supérieur au rang contenu dans le message plus un, il prend alors le rang contenu dans le message plus un. Il choisit aussi l'émetteur du message comme son *père* et transmet à son tour son rang nouvellement choisi et son identifiant à ses voisins directs. Dans le cas contraire, u peut ignorer ce

Procédure 1 Processus de construction du gradient

```
début
    rang = ∞ ; /* Distance minimale au puits en nombre de sauts      */
    père = Ø ; /* Prochain saut en direction du puits                */
    tant que Reception d'un paquet P faire
        si P.rang + 1 < rang alors
            rang = P.rang + 1 ;
            père = P.ID ;
            Diffuser_Gradient(ID, rang) ;
        sinon
            Jeter P
        fin
    fin
fin
```

message. Ainsi, la structure du gradient est initiée par le puits et construite récursivement. En outre, par une simple sur-écoute des messages de construction transmis par ses voisins, un nœud peut aisément identifier ses *fils*. Le processus de construction du gradient est détaillé dans le pseudo-code 1.

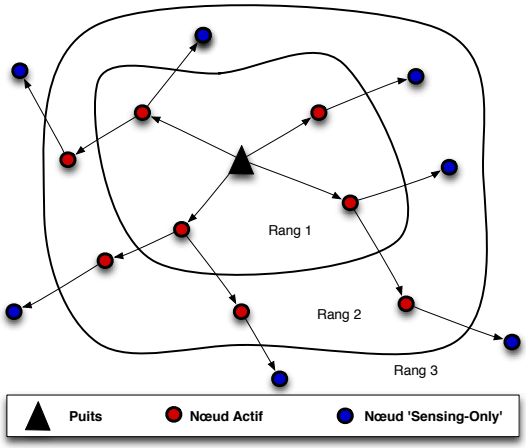


FIGURE 10 – Construction du gradient.

Par l'intermédiaire de cette construction, un sous-ensemble de nœuds *Sensing-Only* peut être identifié. En effet, chaque nœud étant le *père* d'au moins un de ses voisins peut être considéré comme nécessaire à la garantie de la connectivité globale du réseau. Inversement, tout nœud n'ayant aucun *fils* (i.e. chaque nœud *feuille*) est considéré comme inutile à la préservation de cette connectivité, et peut donc devenir *Sensing-Only*. Notons qu'en plus de fonctionner de manière distribuée et sans nécessiter de connaissance (e.g. position exacte),

cette solution engendre un faible coût en messages de contrôle (i.e. un message par nœud). Le processus d'identification des nœuds *Sensing-Only* est illustré dans la figure 10.

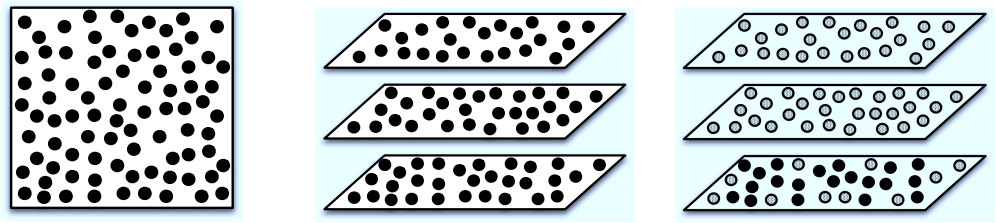
3.3 GÉNÉRALISATION À k ENSEMBLES DISJOINTS

Dans les réseaux de capteurs, les mesures effectuées par un nœud peuvent être imprécises voire faussées, du fait des composants électroniques choisis (e.g. puissance de calcul limitées pour distinguer les erreurs, interfaces de mesure peu consommatrices et miniaturisées, mais peu fiables). Certains déploiements spécifiques, nécessitent donc que la validité de chacune de ces mesures soit garantie, (pour éviter les faux-positifs ou faux-négatifs) [33]. Pour cela, le critère applicatif doit être satisfait à k reprises, par autant de sous-ensembles disjoints de nœuds *Actifs*, afin de pouvoir corréler ensemble plusieurs mesures. Dans le cas de la couverture de surface par exemple, il s'agit de s'assurer que chaque point de la zone d'intérêt se situe dans la zone de couverture d'au moins k capteurs *Actifs*.

Dans de telles situations, une approche par couches successives peut être utilisée pour représenter les différents sous-ensembles de nœuds *Actifs*. Chaque couche est à même de satisfaire, de manière indépendante, le critère applicatif. Partant de cette approche, nous avons étendu nos mécanismes d'identification des nœuds *Sensing-Only* dans le réseau en prenant en compte les spécificités de ces déploiements [83]. Dans un premier temps, nous avons dédié à la première couche (i.e. la plus basse) les opérations de routage. Ainsi, cette dernière garantit à elle seule la connectivité du réseau. Cela dispense alors les autres couches de participer au réseau (i.e. à l'acheminement des mesures vers le puits). Les nœuds les composant peuvent alors devenir *Sensing-Only* sans entraîner de partitionnement de ce dernier. En parallèle, la première couche ayant pour rôle d'effectuer les opérations de routage (i.e. devant satisfaire le critère applicatif et garantir la connectivité), il est possible d'y effectuer les mêmes traitements que décrits en section 3.2 pour identifier les nœuds *Sensing-Only*.

3.4 ÉVALUATION DE NOS PROPOSITIONS PAR SIMULATION

Dans cette section, nous allons présenter les performances des mécanismes détaillés ci-avant et de l'introduction de l'état *Sensing-Only*. Ces résultats ont été obtenus par simulation via le simulateur WSNet [70], comme décrit dans la section 2.5.1. Nous avons retenu le critère applicatif de la couverture de surface,

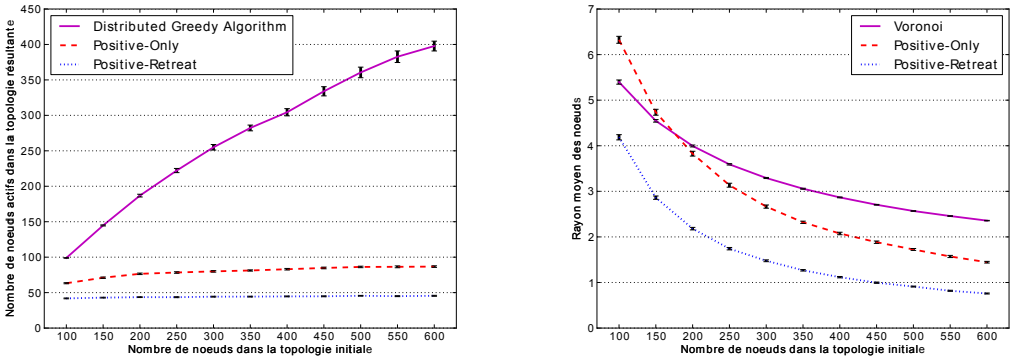


(a) Topologie physique plane.

(b) Division obtenue pour 3 ensembles disjoints.

(c) Identification des nœuds *Sensing-Only*.

FIGURE 11 – Introduction de l'état *Sensing-Only* pour le partitionnement en 3 couches.



(a) Comparaison avec DGA.

(b) Comparaison avec Voronoï.

FIGURE 12 – Évaluation de mécanismes d'ordonnancement d'activité

avec un rayon de identique à la portée de communication (i.e. 10 mètres), pour toutes les raisons citées dans la section 2.2.

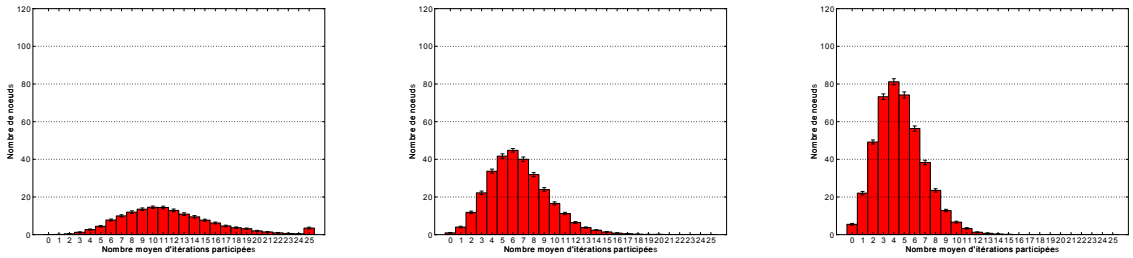
3.4.1 *Étude des méthodes de partitionnement logique*

Dans un premier temps, nous avons évalué plusieurs mécanismes de partitionnement logique présentés dans la Section 2 : Positive-Only (PO) et Positive-Retreat (PR) [34], Distributed Greedy Algorithm (DGA) [35] et un procédé basé sur un diagramme de Voronoï (simplifiée en "Voronoï" par la suite) [36]. Cette étude nous a permis d'évaluer les solutions existantes et de retenir celles qui affichaient les meilleurs propriétés (e.g. messages de contrôle, fonctionnement localisé, performances obtenues) pour servir de fondations à l'introduction de notre nouvel état *Sensing-Only*.

Comme le montre la figure 3.12(a), les mécanismes PO aussi bien que PR offrent de meilleures performances que DGA. En effet, ce dernier nécessite plus de nœuds *Actifs* pour satisfaire les critères de couverture de surface considéré [84]. De plus, avec DGA, la taille du sous-ensemble de nœuds *Actifs* va croissante avec la densité, alors qu'elle reste relativement stable pour PO et PR avec respectivement 90 et 45 nœuds actifs en moyenne et un intervalle de confiance inférieur à 3 nœuds dans les conditions considérées. Cela s'explique notamment par le fait que la perte de quelques messages de contrôle entraîne d'importantes erreurs de calcul de l'ensemble des nœuds *Actifs*. Or, ce phénomène survient invariablement dans un environnement réaliste, en particulier compte tenu du nombre de messages de contrôles générés. En effet, alors que PO induit un message de contrôle par nœud *Actif* et PR deux, quelle que soit la taille de la topologie initiale, DGA fonctionne de manière distribuée, et notifie l'ensemble des nœuds du réseau à chaque étape de décision (i.e. à chaque fois qu'un ensemble de nœuds est sélectionné pour devenir *Passif*). Cela engendre en moyenne trois messages de contrôle par nœud dans la topologie initiale.

Ensuite, nous avons évalué les perfomances de Voronoï. Là encore, nous avons mis ces résultats en parallèle de ceux de PO et PR, et les avons exposés dans la figure 3.12(b). Alors que Voronoï adapte le rayon de couverture et la portée des communications de chaque nœud du réseau, PO et PR déterminent un ensemble de nœuds restant *Actifs*. Ainsi, avec PO et PR, les portées de couverture et de communication sont soit nulles soit maximales. Comme le montrent les résultats, Voronoï induit des ensembles plus grands de nœuds *Actifs* que PO et PR, sauf pour des topologies peu denses. Cela s'explique par le fait que Voronoï adapte toujours les rayons de couverture et de communication. Ainsi, pour des densités importantes, tous les nœuds se verront dotés d'un rayon faible, ce qui s'est avéré moins efficace que d'assurer la couverture et la connectivité par un sous-ensemble réduit de capteurs (les autres devenant *Passifs*). Voronoï induit un trafic de contrôle de deux messages en moyenne par nœud, soit plus que PO ou PR, et repose aussi sur des calculs complexes (i.e. caractérisation des cellules [39]), qui ne sont pas toujours possibles pour des entités aussi contraintes que les capteurs. Enfin, le rayon de couverture n'est pas toujours configurable, la précision des composants de capture étant fixe, et la portée de communication ne peut pas être ajustée avec précision dans les composants radio actuels.

Les campagnes d'évaluations détaillées ci-dessus nous ont permis de sélectionner les mécanismes d'ordonnancement d'activité les plus performants dans nos conditions de déploiement, pour permettre de les associer à nos contributions dans les meilleures conditions. Pour cela, nous avons pris en considération la taille du sous-ensemble de nœuds *Actifs* obtenu et le nombre

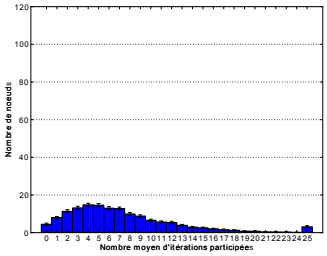


(a) Topologie de 150 nœuds. (b) Topologie de 300 nœuds. (c) Topologie de 450 nœuds.

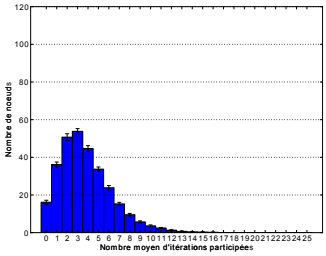
FIGURE 13 – Impact du re-calcul régulier des états d'activité avec PO.

de messages de contrôle induits. À tout point de vue, PO et PR affichent de meilleures performances que les autres solutions considérées, en conservant un nombre réduit de nœuds *Actifs* pour un faible coût en messages. PO et PR s'avèrent aussi complémentaires, PR affichant de meilleures performances (i.e. un sous-ensemble de nœuds *Actifs* plus réduit) que PO, mais nécessitant plus de messages de contrôle. En outre, contrairement à PO, la perte d'un message de contrôle de PR peut entraîner une perte de connectivité ou de garantie du critère applicatif global. Chaque solution est donc adaptée à un type de déploiement spécifique (e.g. réseaux peu sujets aux interférences et collisions, liens instables).

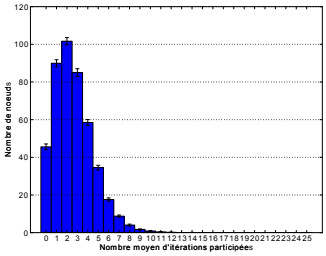
Néanmoins, leur efficacité lors du recalcul à intervalle régulier (permettant d'homogénéiser la charge des nœuds, comme décrit dans la section 2.2.1) n'a jamais été évaluée, et il est donc impossible d'en déduire l'impact sur la durée de vie du réseau. Ainsi, dans les simulations suivantes, nous avons réalisé 25 itérations du processus de décision d'activité. Dans nos conditions de simulation, une valeur plus faible ne permettait pas d'alterner suffisamment les états d'activité (certains capteurs demeurant *Actifs* à chaque tour) et une valeur plus grande donnait des résultats similaires à ceux proposés ici. Chaque fois, les nœuds tiennent l'état du nombre d'itérations pour lesquelles ils ont été *Actifs*, et ajoutent une seconde au délai d'attente aléatoire de leur décision d'activité (compris initialement entre 0 et 32 secondes). Nous avons sélectionné cette solution, car elle permet de privilégier les nœuds déjà sollicités (et donc ayant consommé plus d'énergie) en augmentant artificiellement leur probabilité de devenir *Passifs*, tout en se basant sur une connaissance limitée (e.g. pas de connaissance de l'état des nœuds voisins). D'autres méthodes de pondération du délai ont été proposées dans [34].



(a) Topologie de 150 nœuds.



(b) Topologie de 300 nœuds.



(c) Topologie de 450 nœuds.

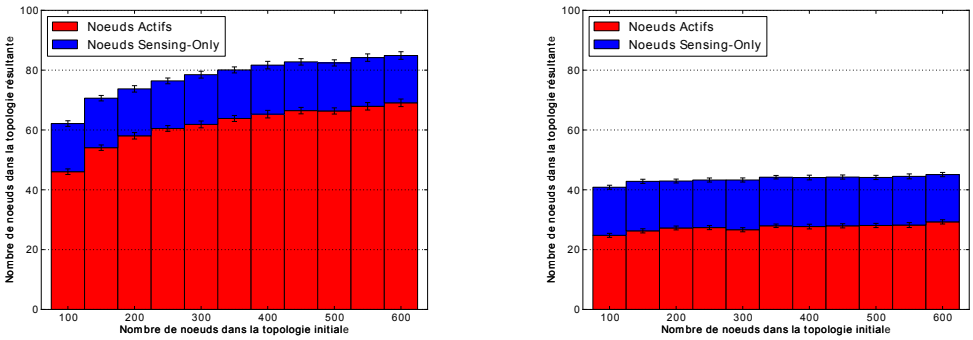
FIGURE 14 – Impact du re-calcul régulier des états d'activité avec PR.

Les résultats pour PO et PR sont respectivement détaillés dans les figures 13 et 14. Avec PO, les nœuds ayant été les plus sollicités auront participé à 18 des 25 itérations pour une topologie initiale de 300 nœuds, et 15 itérations pour une topologie initiale de 450 nœuds. Ainsi, les nœuds sont entrés dans un état *Passif* économe en énergie entre 28% et 40% du temps au moins, permettant par là même d'accroître la durée de vie du réseau en conséquence. Respectivement, avec PR, les nœuds les plus sollicités sont entrés dans un état *Passif* entre 40% et 66% du temps, pour des topologies composées de 300 et 450 nœuds respectivement. Néanmoins, comme l'indiquent les figures 3.13(a) et 3.14(a), à faible densité certains nœuds participent à chaque itération. Ces solutions nécessitent donc une forte redondance dans le réseau, et ne permettent d'accroître sa durée de vie que lorsque la densité est suffisante pour qu'aucun nœud ne soit indispensable à la préservation du critère applicatif et/ou de la connectivité globale du réseau.

Cette première série de simulations nous a permis de faire ressortir deux mécanismes d'ordonnancement d'activité, PO et PR. Ces solutions affichent de bonnes performances, en produisant un sous-ensemble réduit de nœuds *Actifs*, tout en fonctionnant de manière localisée avec un faible nombre de messages de contrôle. De plus, nous avons montré qu'un re-calcul régulier des états d'activité avec PO et PR pouvait permettre d'accroître significativement la durée de vie du réseau. Ainsi, il apparaît intéressant de se baser sur ces mécanismes pour nos contributions ultérieures, et pour servir de support à l'introduction de notre nouvel état *Sensing-Only*.

3.4.2 *Identification des nœuds Sensing-Only*

Nous nous sommes ensuite chargés d'évaluer plus en détail l'impact de l'introduction de notre nouvel état *Sensing-Only*, détaillé en Section 3.2, sur les mécanismes d'ordonnancement d'activité PO et PR. Pour cela, nous avons dans un premier temps évalué la proportion de nœuds *Actifs* et *Sensing-Only* résultants, en fonction des différents mécanismes considérés. Cette étude nous a permis d'identifier la méthode la plus appropriée pour caractériser le sous-ensemble de nœuds *Sensing-Only*, tant pour ses performances que pour ses caractéristiques (i.e. trafic induit, complexité).

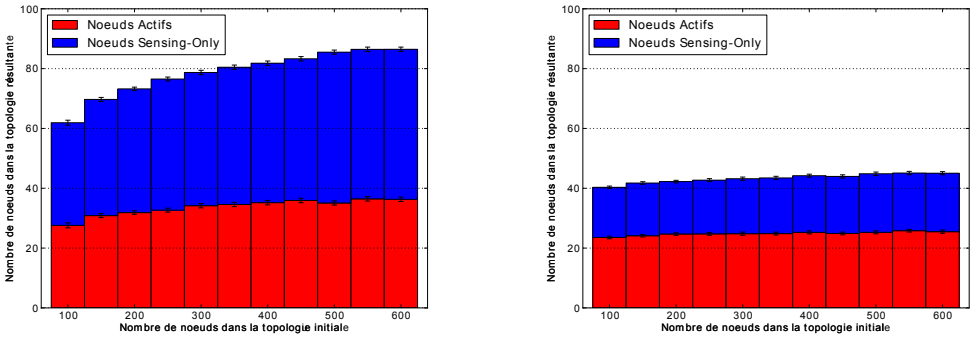


(a) Avec Postitive-Only (b) Avec Postitive-Retreat

FIGURE 15 – Détermination du sous-ensemble Sensing-Only avec un LMST

Les performances induites par l'utilisation d'un LMST, comme décrit dans la section 3.2.1, sont illustrées dans la figure 15. Ici, environ un cinquième des nœuds *Actifs* peuvent devenir *Sensing-Only* en conjonction avec PO, et environ un tiers avec PR, avec un intervalle de confiance toujours inférieur à 5 nœuds. Notons aussi que la proportion de nœuds *Sensing-Only* par rapport aux nœuds *Actifs* reste globalement stable indépendamment de la densité initiale du réseau. En effet, le LMST favorise un degré homogène dans un même voisinage du fait du processus de négociation entre les nœuds (c.f. section 3.2.1). En effet, le LMST construit une structure avec une forte diversité de chemins et un degré assez faible (i.e. 2,06 voisins obtenus en moyenne et 4 au maximum) et homogène dans un même voisinage du fait du processus de négociation entre les nœuds (c.f. section 3.2.1). Ces propriétés sont ici handicapantes. Une large majorité des nœuds devront retransmettre les données de leurs voisins, et ne peuvent donc pas devenir *Sensing-Only*, d'où des performances assez limitées. Enfin, nous avons observé que la construction de la topologie virtuelle effectuée par le LMST nécessitait en moyenne 4,5 messages de contrôle par nœud

Actif, et que la perte de l'un d'entre eux résultait invariablement en la suppression d'un lien dans la topologie résultante, ce qui a pour effet de diminuer d'autant le nombre de nœuds *Sensing-Only*. Ainsi, LMST s'avère être peu résistant aux perturbations dans le réseau et assez gourmand en énergie de par le nombre de messages de contrôle émis.



(a) Avec Postitive-Only (b) Avec Postitive-Retreat

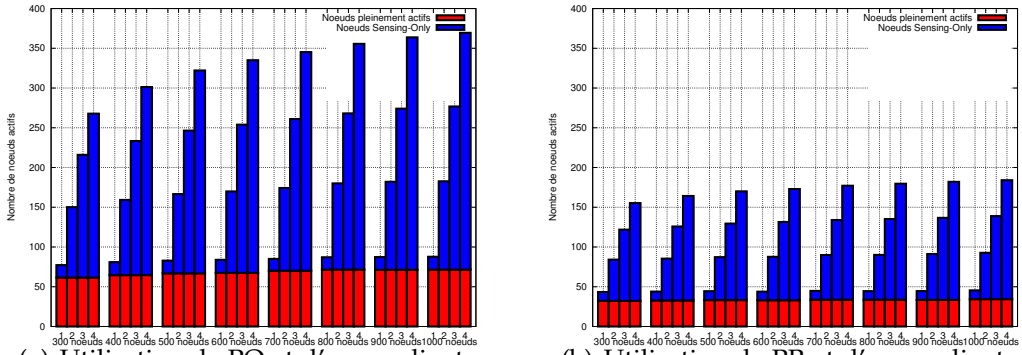
FIGURE 16 – Détermination du sous-ensemble Sensing-Only avec un Gradient

De la même manière, nous avons évalué la taille des sous-ensembles de nœuds *Actifs* et *Sensing-Only* induits par l'utilisation d'un gradient, comme décrit dans la section 3.2.2. Les résultats de cette étude sont présentés en figure 16. Ici plus de la moitié des nœuds *Actifs* deviennent alors *Sensing-Only* en conjonction avec PO, et plus d'un tiers en conjonction avec PR, avec un intervalle de confiance là encore inférieur à 5 nœuds. Lors de la construction du gradient, les nœuds choisissent comme père le voisin dont ils reçoivent le message de contrôle en premier. Cette situation favorise des degrés élevés, les capteurs retransmettant les messages de construction le plus rapidement obtenant un plus grand nombre de fils (i.e. plusieurs dizaines dans ces conditions). Cela explique les grands ensembles de nœuds *Sensing-Only*, comparé aux résultats obtenus avec un LMST. Notons qu'en plus de ses bonnes performances, la construction du gradient ne nécessite qu'un faible nombre de messages de contrôle (i.e. un message par nœud *Actif* en moyenne dans nos simulations) et ne requiert aucune information préalable. En outre, il est peu affecté par les perturbations dans le réseau, la perte d'un message de construction émis par un nœud u est le plus souvent compensée par celui émis par l'un de ses voisins, sauf en cas de rupture temporaire d'un lien (le mécanisme de tolérance aux pannes présenté dans le chapitre 6 permet de compenser ce phénomène). Ainsi, en plus d'afficher de bonnes performances pour l'identification du sous-ensemble de nœuds *Sensing-Only*, la construction d'un gradient

s'avère particulièrement appropriée pour les déploiements de réseaux de capteurs.

3.4.3 *Sensing-Only pour le k-partitionnement*

Comme nous l'avons vu dans la section 3.3, nous avons aussi étendu l'introduction de notre état *Sensing-Only* pour les mécanismes de partitionnement en k couches satisfaisant le critère d'activité. Ici, nous évaluons l'impact de l'introduction de ce nouvel état dans ce cas de figure spécifique.



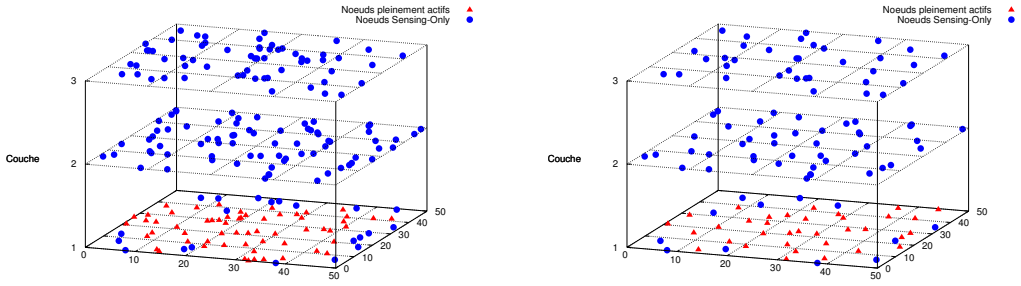
(a) Utilisation de PO et d'un gradient

(b) Utilisation de PR et d'un gradient

FIGURE 17 – Partitionnement en k couches et identification des nœuds *Sensing-Only*

Dans un premier temps, comme pour les mécanismes d'ordonnancement d'activité simple, nous avons évalué la proportion de nœuds présents dans chaque couche par simulation, en utilisant les mécanismes PO et PR étendus au partitionnement en k couches [85]. Comme expliqué en section 3.3, la couche la plus basse (i.e. la couche 1) garantit à la fois le critère applicatif (ici la couverture de surface) et la connectivité du réseau. Ainsi, tous les nœuds appartenant à une couche supérieure peuvent devenir *Sensing-Only*. La construction d'un gradient est aussi initié à la couche 1 pour y identifier là aussi les nœuds *Sensing-Only*. Suivant le nombre de couches désirées et la densité de la topologie initiale, notre solution permet de faire entrer jusqu'à 70% des nœuds *Actifs* en mode *Sensing-Only*, dans les conditions considérées ici, en ne nécessitant qu'un message de contrôle par nœud *Actif* à la couche 1 (du fait de la construction du gradient).

Un exemple de partitionnement d'un réseau initialement composé de 300 nœuds et de la répartition entre états *Actifs* et *Sensing-Only* en découlant est proposé par la figure 18.



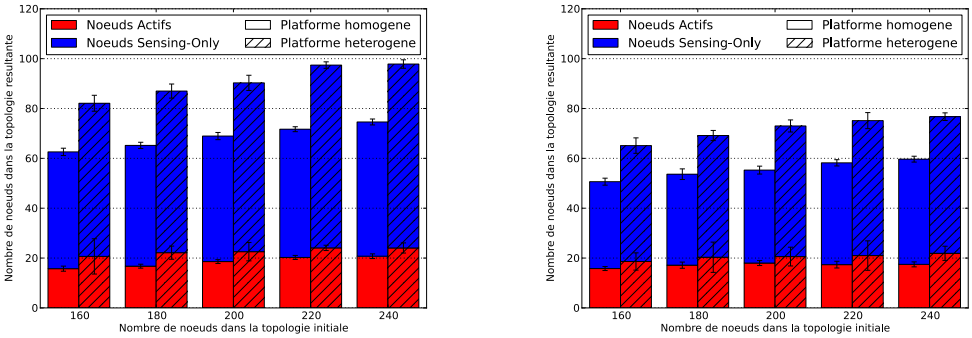
(a) Utilisation de PO et d'un gradient (b) Utilisation de PR et d'un gradient

FIGURE 18 – Exemple de distribution des nœuds suivant leur couche et leur état d'activité

3.5 ÉVALUATION EXPÉRIMENTALE DE NOS PROPOSITIONS

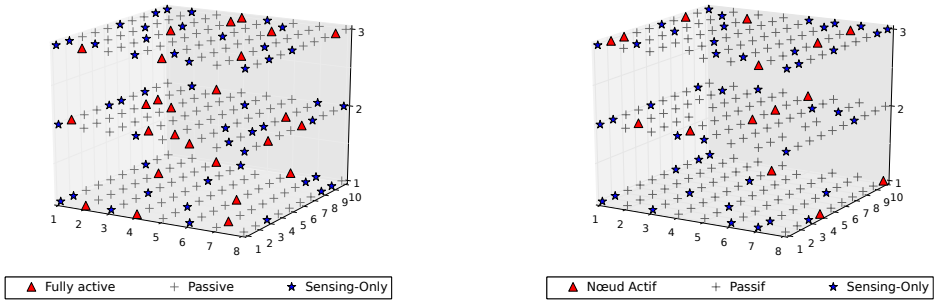
Enfin, nous avons effectué une campagne expérimentale mettant en scène les mécanismes PO et PR, en conjonction avec une identification des nœuds *Sensing-Only* par gradient. Cette campagne a été menée sur les plateformes IoT-LAB de Strasbourg et de Grenoble (c.f. section 2.5.2). Cela nous a permis de mesurer précisément l'impact de nos contributions dans des conditions réelles, et dans des environnements différents, les topologies des deux plateformes étant disjointes. Les nœuds équipant la plateforme IoT-LAB ne disposant pas de mécanisme de localisation, un LMST n'a pu être utilisé par la suite pour identifier les nœuds *Sensing-Only*. De plus, nous avons dû adapter le critère applicatif (i.e. jusqu'alors la couverture de surface) afin d'être conforme à ces conditions. En effet, les capteurs composant la plateforme IoT-LAB ne disposent pas de mécanismes de localisation (e.g. GPS). Nous avons donc considéré à présent un critère de contrôle de densité de paramètre 10 (i.e. le test d'activité est positif *si et seulement si* le nœud a moins de 10 voisins *Actifs*). Notons enfin qu'il s'agit, à notre connaissance, de la première étude expérimentale pour des mécanismes d'ordonnancement d'activité par partitionnement logique effectuée à ce jour.

Nous avons alors évalué les proportions de nœuds *Actifs* et *Sensing-Only* induite par nos mécanismes sur des topologies allant de 160 à 240 nœuds. Le gradient permet ici de faire passer entre deux tiers et trois quarts des nœuds jusqu'alors *Actifs* en *Sensing-Only*, comme le démontrent les résultats fournis dans la figure 19. Les résultats diffèrent selon la plateforme utilisée, celle de Strasbourg étant régulière, de densité homogène et isolée, contrairement à celle de Grenoble qui est irrégulière, de densité hétérogène et plus sujette aux per-



(a) Utilisation de PO et d'un gradient (b) Utilisation de PR et d'un gradient

FIGURE 19 – Identification des états d'activité sur la plateforme IoT-LAB



(a) Utilisation de PO et d'un gradient (b) Utilisation de PR et d'un gradient

FIGURE 20 – Répartition des états d'activité sur la plateforme IoT-LAB de Strasbourg

turbations radio extérieures. Nos mécanismes permettent de faire passer une large proportion de nœuds en mode *Sensing-Only*, et ce même dans des environnements instables. En effet, la perte de messages due aux perturbations radio extérieures conduit à une augmentation de près de 25% de la taille des sous-ensembles de nœuds *Actifs* ou *Sensing-Only* ainsi qu'à une stabilité plus faible (l'intervalle de confiance passant de 3 à 6 nœuds en moyenne lors du changement de plateforme), sans mettre en péril la connectivité de la topologie connectée ni la satisfaction du critère applicatif. Ces résultats démontrent donc le bon comportement ainsi que la validité de nos contributions observées jusqu'alors par simulation. Une vue physique de la plateforme IoT-LAB de Strasbourg présentant la répartition des nœuds suivant leur état d'activité est proposée en figure 20.

Par l'intermédiaire de cette étude expérimentale, nous avons démontré que les méthodes d'ordonnancement d'activité que nous avons choisies se prêtent idéalement aux environnements réels (e.g. instables), en passant de la théorie à la pratique. De plus, nous avons montré que l'introduction d'un nouvel état intermédiaire *Sensing-Only* pouvait concerner plus de la moitié des nœuds jusque là *Actifs*, diminuant d'autant la proportion de ces nœuds participant pleinement au fonctionnement du réseau à un moment donné.

3.6 CONCLUSION

Dans le présent chapitre, en se basant sur les techniques d'ordonnancement permettant de répartir la charge des nœuds en les faisant alterner successivement entre un mode *Actif* et *Passif*, nous avons introduit un état intermédiaire dénommé *Sensing-Only*. Ce dernier concerne une partie des nœuds préalablement considérés comme *Actifs* ne prenant aucun rôle dans la remontée des informations, et ne faisant qu'utiliser le réseau sans y participer. Ils peuvent alors entrer dans un mode plus économe en énergie en limitant l'usage de leur radio en réception (e.g. couper leur radio en réception ou réduire leur fréquence d'échantillonnage du canal) sans porter atteinte au fonctionnement du réseau. L'identification locale de ce sous-ensemble étant un problème NP-complet, nous avons proposé plusieurs techniques d'approximation. Nous avons enfin démontré par simulation et expérimentalement qu'entre un tiers et trois quarts des nœuds jusque là *Actifs* pouvaient en bénéficier.

Dans le prochain chapitre, nous allons montrer comment les états d'activité identifiés par partitionnement logique peuvent permettre de parvenir à une auto-configuration du réseau. Nous proposerons ainsi plusieurs techniques permettant de donner vie au concept de passivité et d'activité définis dans le présent chapitre.

4

AUTO-CONFIGURATION DES NŒUDS PAR ÉTATS D'ACTIVITÉ

"La pierre la plus solide d'un édifice est la plus basse de la fondation."

— Khalil Gibran (1883 - 1931)

Sommaire

4.1 INTRODUCTION

Dans le précédent chapitre, nous avons passé en revue plusieurs méthodes pour attribuer à chaque nœud un état d'activité (i.e. *Actif* ou *Passif*), en fonction de son rôle dans le réseau (e.g. garantie des critères applicatifs, de la connectivité). Nous avons aussi introduit un état intermédiaire *Sensing-Only* pour une partie des nœuds jusqu'alors *Actifs*, réduisant d'autant la taille de ce dernier ensemble. À présent, nous allons proposer des techniques permettant de donner vie à ces états d'activité, en attribuant à chacun d'eux une configuration qui lui est propre. Afin de maximiser les gains énergétiques, nous nous sommes ici concentrés sur la couche ayant la plus grande influence sur la consommation énergétique, en l'occurrence la couche Medium Access Control - Contrôle d'accès au médium (MAC). En effet, la radio est le plus souvent le

composant le plus consommateur d'un capteur [43], et la couche MAC à pour rôle de gérer son utilisation.

Nous basons nos recherches sur les couches MAC à échantillonnage de canal [50], détaillées dans la section 2.3. Ces couches MAC ont pour avantage d'être asynchrones (et donc de ne nécessiter aucun mécanisme de synchronisation coûteux) et peu sensibles aux changements d'échelle. Le paramètre ayant la plus grande importance dans ces couches MAC est la fréquence d'échantillonnage du canal radio ou LPL (c.f. section 2.3.2). En effet, une valeur de LPL élevée permet de pouvoir gérer une charge de trafic plus importante plus rapidement, au prix d'une consommation énergétique accrue. À contrario, une faible valeur de LPL entraîne une consommation énergétique réduite, mais aussi une plus faible capacité à traiter le trafic entrant.

Partant de ce constat, nous proposons aussi deux techniques indépendantes (i.e. ne reposant sur aucun mécanisme d'ordonnancement d'activité) permettant d'adapter la valeur du LPL de chaque nœud aux conditions de déploiement. La première répartit les nœuds en différentes profondeurs de sommeil, en fonction de critères applicatifs. À chaque niveau de sommeil est alors attribuée une valeur de LPL spécifique. Notre seconde solution permet d'attribuer à chaque nœud une configuration MAC correspondant à la charge de trafic qu'il devra traiter lors du déploiement. En partant du constat que cette charge dépend fortement de la position dans l'arbre de routage, notre mécanisme en réalise une estimation, et calcule la configuration MAC idéale correspondante. Par l'intermédiaire de ces solutions, nous visons à réduire la consommation énergétique moyenne du réseau, sans en affecter les performances (e.g. délais, taux de pertes).

4.2 TRANSPOSITION DES ÉTATS D'ACTIVITÉ

Plusieurs options ont été à ce jour envisagées pour donner vie au concept de passivité des nœuds. Comme nous l'avons vu dans le chapitre 2, plusieurs techniques ont été proposées dans ce sens. Cependant, aucune d'entre elles n'est pleinement satisfaisante, soit du fait des conditions qu'elles nécessitent (e.g. synchronisation temporelle des nœuds [35, 34]), soit du fait de leur impact sur l'application (e.g. privation d'une partie des mesures [41]). Nous avons donc cherché à attribuer à chaque état une configuration des nœuds correspondante. Pour cela, nous nous sommes concentrés sur la couche ayant la plus grande influence sur la consommation énergétique, afin de reflfer au mieux le rôle des états d'activité.

Or, dans les réseaux de capteurs, les communications radio sont responsables de la majorité de l'utilisation énergétique [43]. Ainsi, la configuration MAC des nœuds exerce une grande influence tant sur les performances du réseau que sur leur consommation. Plus précisément, dans les couches MAC à échantillonnage de canal, les nœuds vérifient à intervalle régulier si un message leur étant destiné est émis sur le canal radio, et maintiennent leur radio éteinte le reste du temps afin de préserver leurs ressources. Cette fréquence d'échantillonnage du canal radio est aussi appelée LPL pour Low-Power-Listening, en référence au mécanisme dont elle découle [50]. Afin de s'assurer que sa cible sera prête à recevoir, tout nœud désirant transmettre un message devra émettre un préambule au préalable. Pour être certain d'être perçu, ce dernier devra être nécessairement plus long que le LPL de la cible.

Dans ce contexte, nous proposons d'attribuer à chaque nœud une configuration spécifique de la valeur de son LPL, selon son état d'activité (i.e. *Actif*, *Passif* ou *Sensing-Only*) et de manière à refléter au mieux ses besoins. Cette technique présente plusieurs avantages par rapport aux autres solutions envisagées jusqu'alors [34, 35, 41, 42]. En premier lieu, elle permet aux nœuds *Passifs* et *Sensing-Only* de réaliser des économies d'énergie en affectant les principales sources de déperdition énergétique. De plus, elle prévient toute perte de connectivité dans le réseau, tout nœud restant joignable quel que soit son état d'activité. Les nœuds *Actifs* désireux de contacter leurs voisins *Passifs* ou *Sensing-Only* devront simplement ajuster la taille de leur préambule pour y parvenir. De cette manière, aucun mécanisme de réveil synchronisé n'est requis.

4.3 AUTO-CONFIGURATION DE LA COUCHE MAC DU RÉSEAU

Partant de cette approche, nous avons aussi développé deux mécanismes pour permettre d'introduire une configuration hétérogène et dynamique de la couche MAC des nœuds du réseau. Chacun d'entre eux repose sur des critères spécifiques (i.e. applicatifs ou relatifs à la topologie de routage) pour y parvenir.

4.3.1 *Profondeur de sommeil*

Le fonctionnement du réseau est fortement conditionné par l'application considérée. En effet, c'est celle-ci qui détermine les évènements qui seront générés. Il est donc essentiel de traduire le contexte applicatif en paramètres MAC

[86]. Nous nous sommes donc ici inspirés des techniques d'ordonnancement d'activité à k ensembles disjoints présentées dans la section 3.3, pour proposer une méthode d'auto-configuration des nœuds. Ici, nous visons à obtenir plusieurs profondeurs de sommeils, chacune regroupant un sous-ensemble des nœuds du réseau et satisfaisant à elle seule le critère applicatif. Ainsi, nous pouvons affecter à chacune d'entre elles une configuration MAC qui lui est propre et qui correspondra au mieux à ses besoins.

Ainsi, tous les nœuds font initialement partie de la couche 1. Après un délai, chacun d'entre eux décide alors d'y rester ou d'appartenir à une couche supérieure. Ce délai peut être calculé pour favoriser certains nœuds, par exemple les nœuds disposant de ressources limitées en leur affectant un délai plus élevé. Ici nous avons choisi de garder un délai aléatoire dans un premier temps, pour pouvoir mesurer l'impact du mécanisme général. Entre temps, les messages d'activité des voisins ayant choisi un délai aléatoire plus court sont traités. Ceux-ci contiennent, en plus de l'identifiant et de la position de l'émetteur, le niveau de la couche à laquelle ils sont rattachés. Ainsi, chaque nœud tient à jour une liste des voisins répertoriés pour chaque couche d'activité. Une fois le délai aléatoire écoulé, chaque nœud évalue successivement son critère applicatif pour chaque couche, en partant de la plus basse (i.e. la couche 0). Si ce critère est déjà satisfait, la présence de ce nœud n'y est pas nécessaire et ce dernier doit alors effectuer le même test pour la couche suivante, et ainsi de suite. Une fois que ce test est négatif pour une couche ou que le nombre maximal de couches exigées est atteint, le nœud en question la sélectionne comme sienne et le signale à son voisinage par l'envoi d'un message contenant son identifiant et la profondeur de la couche sélectionnée, et toute information nécessaire à l'évaluation du critère applicatif sélectionné.

Ensuite, à chaque profondeur de sommeil est associée une configuration MAC spécifique. Typiquement, la couche 1 étant chargée des opérations de routage, cette dernière se voit attribuer la plus faible valeur de LPL. Chaque couche suivante prend alors une valeur de LPL plus élevée. Ainsi, on obtient plusieurs sous-ensembles de nœuds, chacun satisfaisant le critère applicatif et disposant d'une configuration qui lui est propre. Cette approche est particulièrement adaptée à certains déploiements spécifiques, nécessitant une forme de redondance dans le réseau. C'est notamment le cas des réseaux opportunistes, où la duplication de paquets peut être régulée par un mécanisme de contrôle de densité [32], ou encore les réseaux utilisant des solutions de tolérance aux pannes par redondance [87]. Son fonctionnement est détaillé dans le pseudo-code 2 dans le cas du critère applicatif de la couverture de surface (c.f. section 2.2.1).

Procédure 2 Algorithme de profondeur de sommeil

```
début
    Prof_sommeil = Ø; /* Profondeur de sommeil                    */
    voisins = []; /* # de voisins recensés pour chaque couche     */
    Attendre(Délai_aléatoire);
    i = 1;
    tant que entièrement couvert à la couche i faire
        si SontConnectés(voisins) alors
            i += 1;
        fin
    fin
    Prof_sommeil = i;
    Diffuser_kPositive(ID, position, couche);
fin
```

4.3.2 *Estimation du trafic*

Une autre approche pour partitionner les nœuds en sous-ensembles disjoints est de se baser sur des informations propres à la topologie de routage. Ces informations peuvent soit être directement communiquées par la couche routage ou obtenues par l'intermédiaire d'une analyse de la topologie virtuelle sous-jacente. Comme nous l'avons rappelé précédemment, la majorité des protocoles de routage pour les réseaux d'objets reposent sur une structure de graphe acyclique orienté. Ainsi, la remontée des informations au puits s'effectue récursivement, de *père* en *fils* jusqu'à parvenir au puits. Nous nous sommes dans un premier temps concencentrés sur le cas spécifique d'un modèle de trafic constant (i.e. orienté temps). En supposant le trafic constant, il est alors possible d'en obtenir une estimation pour chaque nœud, en fonction du nombre de nœuds en amont dans la structure de routage (i.e. ses *fils*, les *fils* de ses *fils*, etc.) dont il devra retransmettre l'information.

Afin d'effectuer cette estimation du trafic, il est nécessaire d'identifier pour chaque nœud u, combien d'autres sont en amont dans la topologie. Pour cela, chacun d'entre eux se situant à une extrémité de la topologie (i.e. une feuille) doit envoyer un message à son père, contenant son propre identifiant et le nombre de nœuds en amont (soit initialement 0). Chacun tient alors à jour une valeur, représentant le nombre de nœuds en amont à lui même, via la réception des messages transmis par ses *fils*. À chaque réception, à cette valeur est ajoutée celle contenue dans le message plus un (représentant l'émetteur du message). Une fois les notifications de tous ses *fils* reçues, u peut à son tour

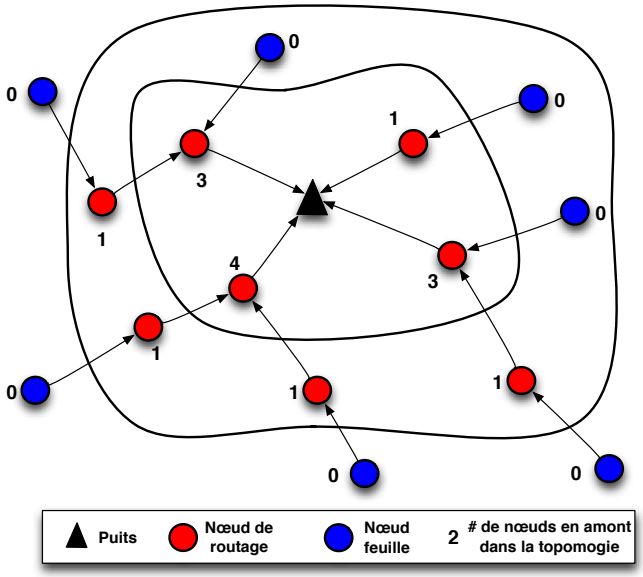


FIGURE 21 – Estimation du trafic dans un DAG

transmettre cette valeur à son père. Ce procédé s'effectue donc récursivement des extrémités de la topologie jusqu'au puits, et nécessite exactement un message de contrôle par nœud du réseau.

Ce processus d'estimation du trafic est détaillé dans le pseudo-code 3 et la figure 21.

Ensuite, chaque nœud peut déduire sa configuration MAC idéale en fonction du trafic moyen prévu par nœud et de cette valeur. Ainsi, plus le nombre de nœuds en amont dans la topologie de routage sera élevé, plus le LPL sera réduit, de façon à pouvoir traiter cette charge de trafic plus importante. À contrario, les nœuds pour lesquels cette valeur est faible voire nulle pourront choisir un LPL élevé, leur permettant ainsi de réaliser des économies d'énergie. Notons que la fréquence des relevés induite du modèle de trafic peut aussi être déduite par apprentissage au cours du déploiement (i.e. en calculant le délai moyen séparant deux messages reçus consécutivement). À partir de là, la configuration MAC de chaque nœud peut être ajustée dynamiquement et de manière indépendante selon les variations de charge de trafic dans le réseau. Il est ainsi possible de déterminer exactement le LPL approprié pour la charge de trafic estimée, en suivant la formule suivante :

$$\mathrm{LPL} = \frac{\mathrm{Fréquence}_{\mathrm{Relevés}}(\mathrm{Hz})}{\mathrm{Nb}_{\mathrm{Amont}}} \times 1000$$

Procédure 3 Algorithme d'estimation du trafic

début
  nbAmont = 0 ; /* Nombre de nœuds en amont dans le DAG */
  nbFait = 0 ; /* Nombre de *fils* déjà notifiés */
  nbFils = Nombre de *fils* dans la structure de routage ;
  si *nbFils* == *0* **alors**
    $\text{Transmettre}_{\text{IDPère}}(\text{ID}, \text{nbAmont})$;
  fin
  tant que *Réception d'un paquet P* **faire**
    nbAmont += P.nbFils ;
    nbFait += 1 ;
    si *nbFait* == *nbFils* **alors**
      $\text{Transmettre}_{\text{IDPère}}(\text{ID}, \text{nbAmont})$;
    fin
  fin
fin

Le fonctionnement de chacune de nos solutions d'auto-configuration est détaillé dans le tableau 5. Notons que celles-ci reposent sur des critères spécifiques (i.e. applicatifs ou relatifs à la topologie de routage) et permettent une configuration des nœuds aussi fine que nécessaire à l'application. En effet, plutôt que de proposer un nombre restreint de configurations MAC disponibles pour configurer les nœuds du réseau, nos contributions laissent ce choix à l'utilisateur, qui prendra sa décision en fonction de l'application ou des limites imposées (e.g. par la couche MAC ou le composant radio).

Contribution	**Fonctionnement**
Profondeur de sommeil	- Séparation des nœuds en plusieurs couches selon le critère applicatif. - La première couche est chargée des opérations de routage. - Le LPL le plus bas revient à la première couche. - Attribution incrémentale d'un LPL pour chaque couche supérieure
Estimation du trafic	- Chaque nœud feuille envoie à son père son nombre de fils (i.e. 0) - Les nœuds recevant cette valeur y ajoutent la leur et retransmettent - Le LPL est calculé à partir de cette valeur et de la féquence des relevés

TABLE 5 – Fonctionnement général de nos solutions d'auto-configuration.

4.4 ÉVALUATION DE NOS PROPOSITIONS PAR EXPÉRIMENTATION

Nous nous sommes ici intéressés à l'intérêt que présente l'introduction de configurations MAC hétérogènes au sein du réseau, et les performances présentées par nos contributions dans ce contexte. Pour cela, nous avons mené plusieurs campagnes expérimentales sur la plateforme IoT-LAB de Strasbourg (c.f. section 2). Nous nous sommes ici concentés sur une évaluation expérimentale, du fait de la difficulté d'obtenir par simulation une mesure précise de la consommation engendrée par les communications.

Nous avons dans un premier temps évalué l'impact de la configuration MAC des nœuds sur leur consommation énergétique. Pour cela, nous avons initialement doté chaque nœud de la plateforme d'un LPL de 125ms. Puis, toutes les 100s, nous avons doublé cette valeur. Les résultats de cette étude sont illustrés par la figure 22. Notons que les valeurs fournies ici valent pour le matériel de la plateforme IoT-LAB, mais cette tendance peut être généralisée à tout autre type de capteurs (la plupart des composants radio opérant sur le même mode de fonctionnement [23]).

Comme nous pouvons l'observer, l'interface radio est bien la principale source de déperdition énergétique, avec une consommation allant de 5mW à 20mW. De plus, nous pouvons remarquer que le niveau de consommation de l'interface radio est fortement lié à la valeur de LPL choisie. Ainsi, passer d'un LPL de 125ms à un LPL de 250ms permet de réduire de 40% l'énergie consommée par le composant radio, et de 65% en passant d'un LPL de 125ms à un LPL de 500ms. Il apparaît donc judicieux d'introduire des configurations de LPL hétérogènes dans le réseau, cette valeur étant de première importance pour contrôler sa consommation énergétique. Notons qu'ici et par la suite, nous avons choisi trois valeurs de LPL (i.e. 125ms, 250ms et 500ms) du fait de leur utilisation répandue dans les déploiements actuels. Cependant, n'importe quel ensemble de valeurs peut être utilisé, ces valeurs dépendant des conditions spécifiques du déploiement (e.g. durée de vie nécessaire, performances requises).

4.4.1 *Des états d'activités sous forme de configurations MAC*

Nous avons alors évalué PO couplé avec une identification des nœuds Sensing-Only via un gradient (c.f. section 3.2.2), et en attribuant à chaque état d'activité une configuration MAC spécifique comme décrit dans la section 4.2. Nous avons attribué aux nœuds *Actifs* un LPL de 125ms, de façon à ce qu'ils soient

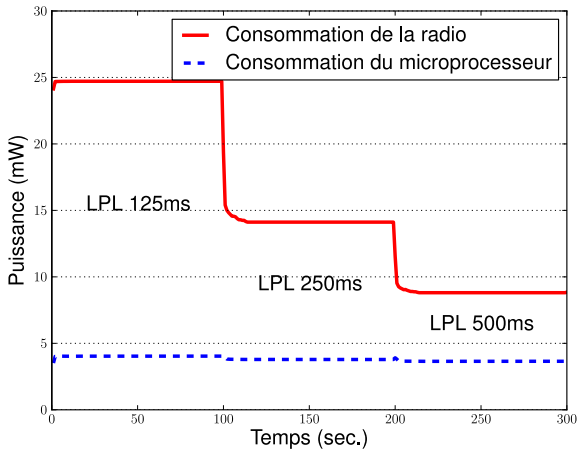
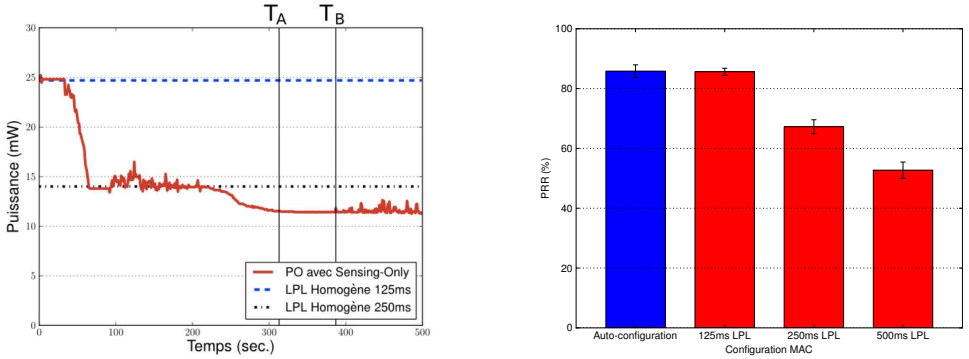


FIGURE 22 – Consommation énergétique en fonction du LPL



(a) Consommation énergétique moyenne pour 3 couches disjointes

(b) Pourcentage de paquets reçus au puits

FIGURE 23 – Impact de kPO sur le réseau.

à même de gérer le trafic au mieux, alors que les nœuds *Passifs* se voyaient attribuer une configuration de 500ms, de façon à leur permettre de réguler au mieux leur consommation énergétique. Enfin, les nœuds *Sensing-Only* se situant à mi-chemin entre ces deux états, nous leur avons affecté un LPL de 250ms. La consommation moyenne du réseau est donnée en figure 4.23(a).

Les 30 premières secondes de l'expérience sont dédiées à l'initialisation des nœuds (s'assurer qu'ils démarrent correctement). Le phase de partitionnement logique, réalisée par PO, est alors réalisée entre 30s et 90s, suivie par la construction du gradient et l'identification des nœuds *Sensing-Only* à partir de 100s et jusqu'au repère T_A. Le réseau reste libre de tout trafic jusque 400s (i.e. entre T_A et T_B), moment à partir duquel le modèle de trafic est initié.

Solution déployée	Délais de bout en bout
PO avec gradient	351,2 ms (±18.8)
LPL homogène à 125ms	346,6 ms (±17.3)
LPL homogène à 250ms	4232,3 ms (±1124.2)
LPL homogène à 500ms	37118,5 ms (±6500.3)

TABLE 6 – Délais de bout en bout

Comme nous pouvons le voir, la translation de nos états d'activité en configurations MAC permet de réduire la consommation énergétique moyenne du réseau de 48% en comparaison d'un réseau bénéficiant d'une configuration homogène du LPL à 125ms, et de 15% comparé à un réseau bénéficiant d'une configuration homogène du LPL à 250ms.

Enfin, nous avons cherché à mesurer l'impact de nos mécanismes sur les performances du réseau. Pour ce faire, nous avons introduit un modèle de trafic de données, pour lequel chaque nœud *Actif* ou *Sensing-Only* transmet un paquet au puits toutes les 30 secondes. Nous avons alors estimé le pourcentage de paquets reçus au puits (ou PRR pour Packet Reception Rate) et les délais de bout en bout, respectivement dans la figure 4.23(b) et le tableau 6. Comme l'illustrent ces résultats, nos mécanismes permettent de préserver des performances similaires à celles obtenues avec une configuration homogène du LPL de 125ms, tant en termes de PRR que de délais. A contrario, ces mêmes performances se dégradent rapidement avec l'accroissement du LPL dans les solutions homogènes. En particulier, les délais augmentent exponentiellement, les nœuds n'étant plus à même de traiter directement les messages reçus (i.e. les paquets restent longtemps en queue avant que le canal radio ne devienne libre). Rappelons aussi que nos mécanismes d'auto-configuration maintiennent une topologie du réseau toujours connetée. En effet, bien que des configurations MAC hétérogènes cohabitent, la communication de bout-en-bout reste possible car nos solutions garantissent que tout nœud aura nécessairement un LPL supérieur ou égal à celui de son père.

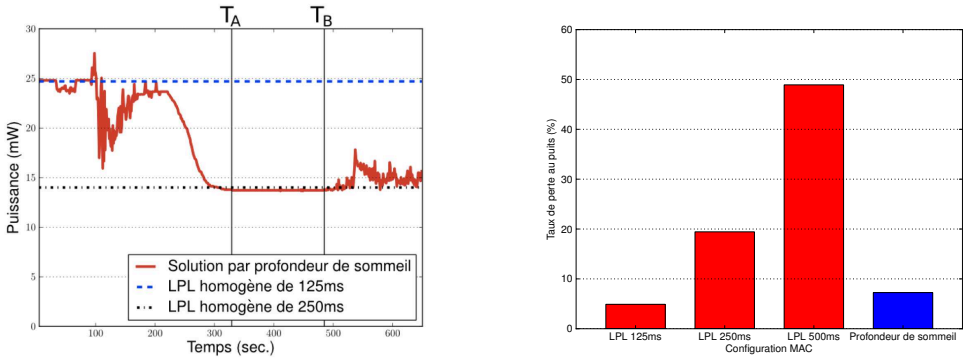
Nous avons jusqu'à présent démontré l'intérêt d'adapter dynamiquement la configuration MAC des nœuds en fonction de critères applicatifs. Nos mécanismes permettent de réduire de près de moitié la consommation énergétique du réseau, 30% de ces gains étant réalisés grâce à la seule introduction d'un nouvel état *Sensing-Only*. De plus, nous avons montré que nos solutions ne dégradent les performances du réseau dans aucune des situations considérées

ici. Notre approche auto-adaptative s'avère donc plus efficace que les solutions impliquant une configuration homogène du LPL.

4.4.2 *Auto-configuration par profondeur de sommeil*

Ensuite, nous avons mené une campagne d'expérimentation en construisant diverses profondeurs de sommeil à partir du critère applicatif de densité du réseau (i.e. la densité de chaque couche doit dépasser un seuil de 10). Chacune de ces couches est alors dotée d'une configuration MAC spécifique, comme décrit en section 4.3.1. Ainsi, la couche la plus basse, chargée de relayer les informations, se voit affecter un LPL de 125ms. Cette valeur est alors doublée pour chaque couche supérieure, Contiki OS ne permettant de choisir une fréquence d'échanitillonnage que par puissance de 2 (e.g. 2Hz, 4Hz, 8Hz). Ici, nous considérons 3 couches disjointes, tout nœud n'appartenant à aucune des trois étant affecté d'office à la dernière.

La consommation moyenne du réseau est détaillée dans la figure 4.24(a). Comme précédemment, l'étape de partitionnement logique prend place jusqu'à la première bande verticale. Puis, le réseau est libre de tout trafic jusqu'à la seconde, moment à partir duquel notre modèle de trafic est introduit. Comme nous pouvons le constater, ce système d'auto-configuration permet d'atteindre le même niveau de consommation énergétique qu'avec une configuration homogène du LPL à 250ms.



(a) Consommation énergétique moyenne pour 3 couches disjointes

(b) Taux de perte au puits des solutions envisagées

FIGURE 24 – Impact de kPO sur le réseau.

Nous avons alors évalué les performances du réseau après application d'un modèle de trafic orienté-temps et initié par tous les nœuds du réseau. Tous

les nœuds générant des messages, et non plus les *Actifs* seulement comme auparavant, nous avons diminué la fréquence d'envoi à 60s pour éviter une congestion complète du réseau. La figure 4.24(b) illustre le taux de perte moyen au puits de notre solution et de réseaux dotés d'une configuration homogène du LPL, alors que le tableau 7 présente le délai de bout en bout dans les mêmes conditions. Avec un taux de perte au puits inférieur à 10% et un délai moyen de bout en bout de 3,7s (et un intervalle de confiance inférieur à 10%), notre solution offre un compromis entre une configuration homogène du LPL à 125ms et à 250ms. Ainsi, elle permet de garantir une consommation similaire à celle d'un réseau configuré de manière homogène avec un LPL de 250ms, tout en ayant des performances plus proches d'un réseau configuré avec un LPL de 125ms.

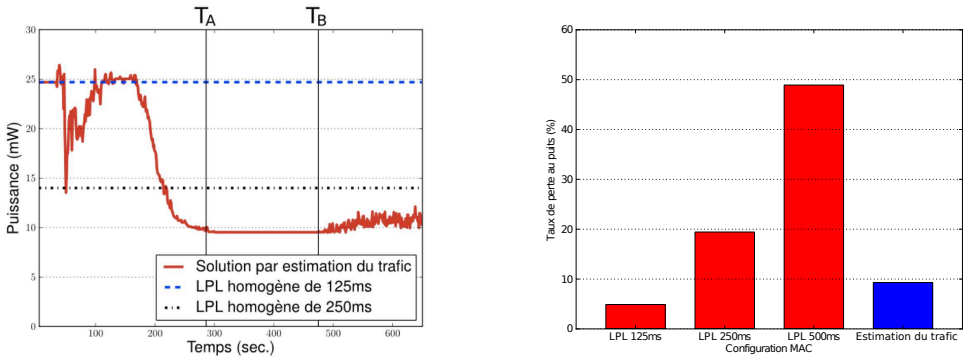
Solution deployée	**Délai de bout en bout**
Profondeur de sommeil	3727,1 ms (±270.8)
LPL homogène à 125ms	1234,6 ms (±142.4)
LPL homogène à 250ms	10243,6 ms (±692.1)
LPL homogène à 500ms	60869,1 ms (±4750.1)

TABLE 7 – Délai de bout en bout de notre solution de profondeur de sommeil

Comme nous l'avons montré, notre contribution permet de partitionner finement le réseau en k ensembles disjoints, chacun à même de remplir le critère applicatif fourni lors du déploiement. Cette solution permet d'amener la consommation énergétique moyenne du réseau au niveau de celle d'un réseau configuré de manière homogène avec un LPL de 250ms, tout en multipliant par trois les performances de ce réseau en termes de taux de réception au puits et de délais. De plus, ici encore notre solution préserve la connectivité du réseau, les nœuds participant au routage (i.e. la couche 1) étant dotés d'une configuration identique du LPL, plus faible que celle des autres nœuds du réseau. Ainsi, cette contribution apporte un compromis plus avantageux que n'importe quelle configuration homogène du LPL considérée dans notre évaluation, et permet donc de s'adapter plus efficacement à tout type de trafic.

4.4.3 *Auto-configuration par estimation du trafic*

Nous avons aussi présenté en Section 4.3.2 une méthode permettant d'attribuer à chaque nœud du réseau une configuration MAC spécifique, non plus selon des critères applicatifs, mais via une estimation du trafic entrant. Pour



(a) Consommation énergétique moyenne

(b) Taux de perte au puits

FIGURE 25 – Impact de l'estimation du trafic sur le réseau.

ce faire, est déterminé récursivement, des feuilles jusqu'au puits, le nombre de nœuds étant en amont dans la topologie de routage. Suivant cette valeur et la fréquence de génération des messages, il est possible d'estimer le trafic moyen devant être traité par chaque nœud, et ainsi proposer une configuration MAC correspondante.

Dans notre campagne expérimentale, les nœuds ayant moins de 3 voisins en amont prennent pour valeur de LPL de 500ms, cette configuration leur permettant de traiter au moins 2 paquets par seconde. De la même manière, les nœuds ayant plus de 8 voisins en amont sont configurés avec un LPL de 125ms. Enfin, les nœuds se situant entre ces deux cas de figure sont dotés d'un LPL de 250ms. En premier lieu, nous avons évalué la consommation énergétique moyenne du réseau. Les résultats de cette évaluation sont illustrés par la figure 4.25(a). Avec une consommation moyenne de $9,8mW$ sans trafic, cette solution a des performances très proches de celles d'un réseau doté d'une configuration du LPL homogène de 500ms. En effet, comme nous l'avons déja indiqué dans la section 3.4.2, le gradient construit des topologies au degré élevé, les capteurs retransmettant les messages de construction le plus rapidement obtenant un plus grand nombre de fils (i.e. plusieurs dizaines dans ces conditions). Ainsi, un grand nombre de nœuds se situent à l'extrémité de la topologie et sont donc configurés avec un LPL élevé. En ce sens, cette solution est donc particulièrement économe en énergie.

Nous nous sommes ensuite penchés sur l'impact induit par cette méthode d'auto-configuration sur les performances du réseau. Les résultats concernant le taux de perte moyen au puits sont proposés en figure 4.25(b), alors que les délais moyens sont fournis dans le tableau 8. Comme nous pouvons le voir,

Solution deployée	Délai de bout en bout
Estimation du trafic	1748,6 ms (±133.5)
LPL homogène à 125ms	1234,6 ms (±142.4)
LPL homogène à 250ms	10243,6 ms (±692.1)
LPL homogène à 500ms	60869,1 ms (±4750.1)

TABLE 8 – Délai de bout en bout de notre solution d'estimation du trafic

notre approche induit un taux de perte et des délais situés entre ceux obtenus avec une configuration du LPL de 125ms et de 250ms. Cette situation résulte du fait que, dans quelques cas, la perte d'un message de contôle entraîne une sous-estimation du nombre de fils, et amène donc un nœud à adopter une configuration du LPL qui ne lui convient pas. Cette situation, bien que rare, entraine un accroissement local des délais et du taux de perte. Cette situation peut néanmoins être compensée par l'utilisation d'un mécanisme annexe de tolérance aux pertes locales, afin de rapprocher les performances de notre solution de celles d'un réseau doté d'une configuration homogène du LPL de 125ms.

Les deux approches présentées ci-dessus (i.e. par profondeur de sommeil et par estimation du trafic) s'attaquent au problème de l'auto-configuration suivant deux angles différents (i.e. via le critère applicatif ou par analyse de la topologie de routage) et sont en cela complémentaires. Leur utilisation permet au réseau de s'adapter à son contexte de déploiement en fournissant à chaque nœud une configuration spécifique de son LPL. Ainsi, sans être meilleure en tout point à toute solution impliquant une configuration homogène des nœuds, cette approche permet d'obtenir un meilleur compromis entre préservation énergétique et performances réseau qu'aucune d'elle.

4.5 CONCLUSION

Dans le présent chapitre, nous avons proposé plusieurs méthodes permettant une auto-configuration hétérogène de la couche MAC au sein du réseau, par anticipation des conditions de déploiement. Dans un premier temps, nous avons donné corps aux états d'activité introduits dans le chapitre précédent, en attribuant à chacun d'eux une configuration MAC spécifique correspodante. Nous avons aussi présenté deux approches permettant de configurer les nœuds aussi finement que désiré (le nombre et la valeur des configurations

utilisées peut être paramétré à loisir par l'utilisateur), soit selon des critères applicatifs, soit en fonction des propriétés de la topologie du réseau pour opérer une estimation du trafic.

Les résultats obtenus par expérimentation montrent que nos solutions permettent d'adapter le réseau pour traduire son contexte de déploiement en configurations MAC appropriées. De plus, nos contributions n'engendrent qu'un faible surcoût en terme de messages de contrôle (environ un message par nœud du réseau) et ne nécessitent aucune connaissance préalable de la topologie ou des conditions de déploiement.

Après avoir présenté ci-avant quelques techniques d'auto-adaptation de la configuration MAC reposant sur une étude préalable de ce dernier (e.g. de la topologie), nous allons par la suite introduire une approche différente. Cette méthode permet au réseau de s'auto-configurer dynamiquement, en adaptant localement la configuration MAC des nœuds, lorsque la charge de trafic touchant ces derniers varie.

5

LE PROTOCOLE BAT-MAC

"Se réunir est un début, rester ensemble est un progrès, travailler ensemble est la réussite."

— Henry Ford (1863 - 1947)

Sommaire

5.1 INTRODUCTION

Dans le chapitre précédent, nous avons présenté plusieures méthodes d'adaptation du réseau. Dans ces approches, des informations relatives au déploiement (e.g. topologie réseau, critère applicatif) sont collectées à intervalle régulier afin de partitionner les nœuds en divers sous-ensembles disjoints. À chacun d'entre eux est alors affectée une configuration MAC spécifique afin de préserver les performances et la connectivité du réseau tout en diminuant la consommation énergétique. Cette approche a pour avantage de permettre aux nœuds de se configurer automatiquement selon les propriétés du réseau. Elle permet aussi de réaliser ce choix en fonction de critères variés (e.g. critères applicatifs, propriétés de la topologie de routage), et donc de s'adapter aisément aux besoins et contraintes spécifiques du déploiement. Néanmoins, bien qu'une ré-évaluation à intervalle régulier de cette configuration permette

d'homogénéiser la consommation énergétique au sein des nœuds du réseau et ainsi accroître sa durée de vie, elle ne permet pas de prendre en considération les variations rapides de charges de trafic pouvant survenir lors de la remontée des informations.

Pour cela, une autre approche est nécessaire. Les nœuds du réseau s'auto-configurent alors à la volée, en fonction d'informations reçues au cours du déploiement, et en anticipant les changements à venir à court terme pour déclencher une re-configuration des nœuds concernés. Dans le présent chapitre, nous proposons un nouveau protocole MAC, dénommé BAT-MAC. BAT-MAC peut être implanté en combinaison avec n'importe quelle couche MAC à échantillonnage de canal (e.g. B-MAC, X-MAC), et permet de calculer la durée prévue de la hausse d'intensité du trafic et d'adapter la configuration MAC des nœud impactés de manière à la prendre en charge plus efficacement. Une fois le trafic revenu à la normale, BAT-MAC permet à ces nœuds de reprendre leur configuration initiale.

5.2 DÉFINITION DE LA PROBLÉMATIQUE

Les approches décrites dans le chapitre 4 permettent d'adapter la configuration MAC des nœuds du réseau à partir d'informations récoltées à intervalle régulier. Ainsi, elles extrapolent le comportement futur du réseau à partir de données actuelles, en partant du principe que ces informations seront encore valables pendant un certain intervalle de temps (i.e. la période séparant deux re-calculs des configurations). Ces méthodes permettent d'effectuer une auto-configuration du réseau, et d'en diminuer la consommation énergétique sans perte de performances lorsque le comportement de ce dernier est stable.

Or, il n'est pas rare que la charge de trafic varie au cours du déploiement. Tout particulièrement, les nœuds doivent bien souvent transmettre plusieurs messages d'affilée, en rafale. Cette situation se produit notamment lorsqu'un nœud est amené à stocker une partie de ses relevés, par exemple lorsque le canal radio n'est pas libre pour qu'il les transmette sur le moment, ou parce que le puits est momentanément injoignable [87]. Une fois cette période d'inaccessibilité du réseau passée, les relevés doivent être transmis au plus vite au puits. Cette situation se produit aussi pour les réseaux reposant sur un modèle applicatif orienté-événements, où les relevés sont émis à haute fréquence lorsqu'un événement particulier défini par l'application est perçu (e.g. chute d'une personne [14], prédiction d'une attaque cardiaque [1, 13]).

Ces différents cas impliquent une augmentation rapide de la charge de trafic, et donc de l'occupation du canal radio. Cela a pour conséquence une diminution significative des performances du réseau (e.g. délais, consommation énergétique due à l'émission des préambules) jusqu'à ce que le trafic se stabilise à nouveau. Ce phénomène est illustré par la figure 26 dans le cas des protocoles B-MAC et X-MAC (c.f. section 2.3.2). Face à ces situations, il apparaît donc nécessaire de mettre en œuvre une adaptation à la volée de la configuration MAC des nœuds concernés. C'est à cet effet que nous proposons ici le protocole BAT-MAC.

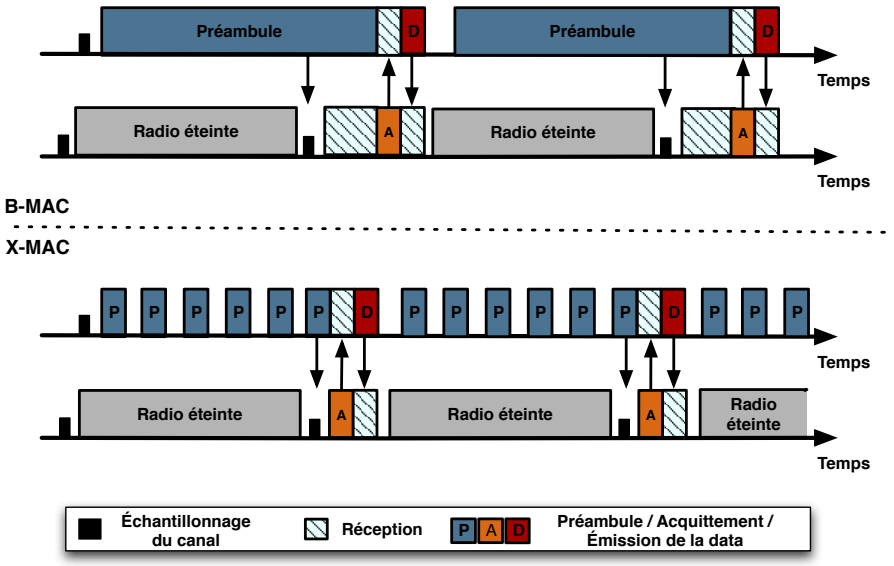


FIGURE 26 – Gestion des messages consécutifs avec B-MAC et X-MAC

5.3 FONCTIONNEMENT GÉNÉRAL DU PROTOCOLE BAT-MAC

Avec BAT-MAC, nous proposons de diminuer sensiblement la valeur du LPL en cas de variations rapides de trafic, afin de pouvoir les gérer plus efficacement et plus rapidement. Ainsi, lorsque la charge de trafic reste stable, les nœuds sont configurés tel que prévu initialement pour le déploiement (ou selon la configuration induite par les méthodes proposées en section 4). De cette manière, le réseau peut remplir les besoins de l'application (e.g. durée de vie souhaitée, trafic moyen envisagé), tels que définis par l'utilisateur. A contrario, lorsque survient une augmentation soudaine de la charge de trafic, les nœud touchés adapteront temporairement leur valeur de LPL, de façon à pouvoir traiter les messages suivants avec une plus grande efficacité. Ainsi,

la surcharge de trafic pourra être traitée plus rapidement, résultant en une amélioration des performances générales du réseau.

Comme nous l'avons expliqué dans la section précédente, ces accroissements rapides de charge de trafic sont souvent dus à l'émission de plusieurs messages dans un intervalle de temps restreint. Le nombre de messages étant bien souvent connu par l'émetteur, au moins pour partie, celui-ci peut indiquer cette valeur dans le premier message de la série (0 lorsque le trafic est stable). Cette valeur, appelée Paq_{MIN}, représente la plupart du temps le nombre de messages stockés dans la file au niveau applicatif, ou dans la file d'attente MAC. Le récepteur de ce message diminuera alors son LPL autant que possible pendant la durée prévue de la hausse de trafic, de manière à pouvoir traiter ces messages plus efficacement. Cette valeur, appelée LPL_{MIN}, correspond à la configuration minimale autorisée par l'interface radio ou par le système d'exploitation. Il reprendra ensuite sa valeur initiale de LPL, dénommée LPL_{INIT}, de manière à ne pas accroître inutilement et de manière prolongée sa consommation énergétique. Le fonctionnement général du protocole BAT-MAC est détaillé dans la Figure 27 et l'algorithme 4. Ici, nous nous contentons de réduire le LPL à une valeur minimale en cas de hausse de la charge de trafic, sans le supprimer vraiment. En effet, il peut s'avérer contre-productif de garder sa radio constamment allumée dans ces situations, car un intervalle de quelques millisecondes est nécessaire à la radio entre l'émission de deux paquets. Ainsi, les messages ne peuvent être reçus l'un à la suite de l'autre, et de meilleures performances sont obtenues en conservant un LPL minimal. Enfin, le passage d'une couche MAC à l'autre est une opération complexe, consommatrice et non recommandée sur des systèmes d'exploitation si limités.

La durée de l'augmentation de la charge de trafic, et donc la durée pendant laquelle le récepteur devra adopter LPL_{MIN} comme LPL, peut être aisément calculée à partir du nombre de messages Paq_{AUG} devant être envoyés lors de celle-ci et du LPL adopté dès lors (i.e. LPL_{MIN}). À partir de ces informations, on peut déduire la durée maximale de cet accroissement de trafic T_{adapt} :

$$T_{adapt} = LPL_{INIT} + (Paq_{AUG} - 2) \times LPL_{MIN} \times (1 + Marge_{err}) \quad (1)$$

Ici, LPL_{INIT} correspondra au temps maximum pris par le premier paquet (le temps que la radio affecte la nouvelle valeur de LPL). Nous avons aussi ajouté une marge d'erreur $Marge_{err}$ pour compenser le temps pris par la radio pour se reconfigurer, ou palier les éventuels problèmes de transmissions (e.g. pertes de paquet, mauvais échantillonnage). Cette marge d'erreur est déterminée en

Procédure 4 Fonctionnement général du protocole BAT-MAC

```
début
    tant que Réception d'un paquet P faire
        si P.Paq_AUG > 0 alors
            T_adapt = Calcul_Tadapt(LPL_INIT, LPL_MIN, P.Paq_AUG);
            Lancer_temporisateur(T_adapt);
            Change_LPL(LPL_MIN);
        fin
        Traiter(P);
    fin
    tant que Temporisateur écoulé faire
        Change_LPL(LPL_INIT)
    fin
fin
```

fonction des conditions de déploiement (e.g. perturbations du réseau, qualité du signal radio, composants du nœud).

Ainsi, le protocole BAT-MAC a pour objectif de réduire l'occupation du canal et les délais de ces trafics accrus en permettant un acquittement plus rapide des préambules des messages concernés. Par la même occasion, il vise à diminuer la consommation énergétique au niveau de l'émetteur sans accroître celle du récepteur, et sans jamais porter atteinte aux performances du réseau ou à la consommation des nœuds en présence d'un trafic stable.

5.4 ÉVALUATION DES PERFORMANCES PAR SIMULATION

Dans cette section, nous allons présenter les performances présentées par BAT-MAC sur le simulateur Cooja, comme décrit dans la section 2.5.1. Pour obtenir ces résultats, nous avons implanté BAT-MAC en combinaison avec X-MAC. Nous avons alors mené une campagne d'évaluation impliquant soit BAT-MAC, soit un réseau configuré de manière homogène et statique avec X-MAC (avec un LPL de 125, 250 ou 500ms). Dans notre cas d'étude, une valeur de LPL inférieure à 32ms entraînait une augmentation de la consommation énergétique sans amélioration des performances, à cause du délai imposé entre deux émissions par le système d'exploitation. C'est pourquoi nous avons choisi d'affecter cette valeur à LPL_{MIN}. Parallèlement un grand nombre de déploiements utilisant X-MAC sont configurés avec une valeur de LPL proche de 500ms [29, 52, 32]. Nous avons donc utilisé cette valeur comme LPL initial, LPL_{INIT}.

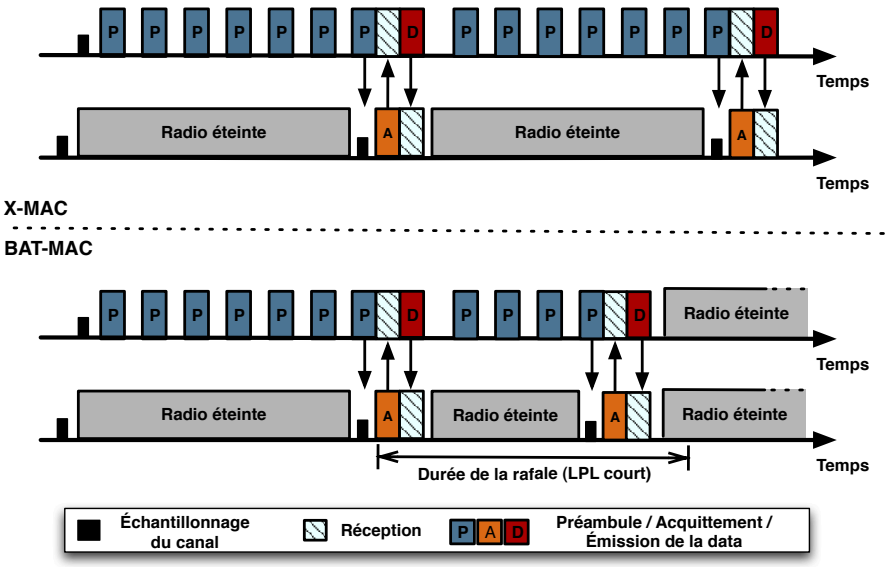


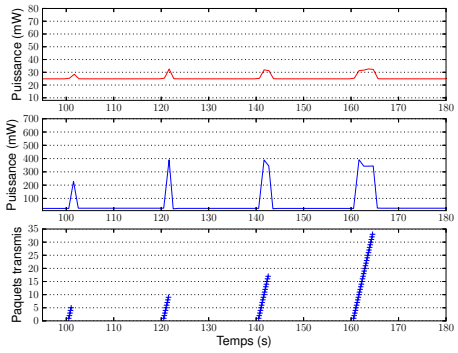
FIGURE 27 – Illustration du fonctionnement général du protocole BAT-MAC

Enfin, en réaction aux observations faites lors d'une étude préalable du taux de perte MAC et des perturbations, nous avons assigné une marge d'erreur de 15%, qui s'est avérée être la marge nécessaire pour qu'aucun message résultant de l'augmentation du trafic ne soit émis en après de T_{adapt}.

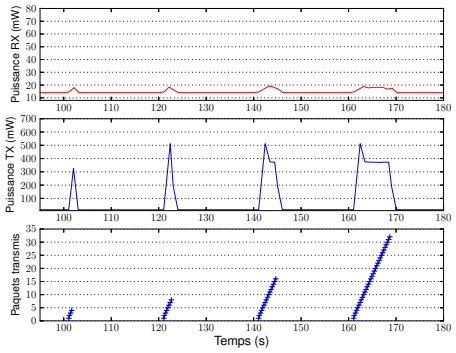
5.4.1 *Évaluation en environnement sans contention*

Dans un premier temps, nous avons évalué les performances de chaque mécanisme par simulation, hors de toute contention ou compétition pour le canal radio. Pour ce faire, nous avons mis en place un déploiement composé de deux nœuds. L'un d'eux joue ici le rôle d'émetteur, envoyant à intervalle régulier une série de paquets l'un à la suite de l'autre. L'autre nœud se chargeait alors simplement de les recevoir. Nous avons fait varier, à chaque itération, la taille de la série de messages émis, de façon à évaluer l'évolution des performances selon la durée de la hausse de trafic.

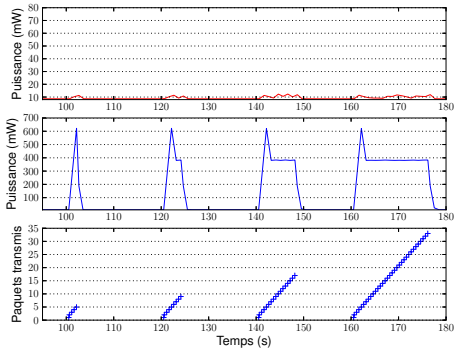
Les résultats de ces simulations sont fournis en Figure 28. Comme nous pouvons le constater, chaque configuration de X-MAC permet de faire face à une situation spécifique, mais échoue à les traiter toutes. En effet, X-MAC permet de traiter efficacement les charges importantes de trafic avec un LPL de 125ms en induisant des délais réduits et une consommation plus faible que toute autre configuration homogène présentée ici. A contrario, en l'absence de trafic, un échantillonnage du canal radio moins fréquent (e.g. un LPL de



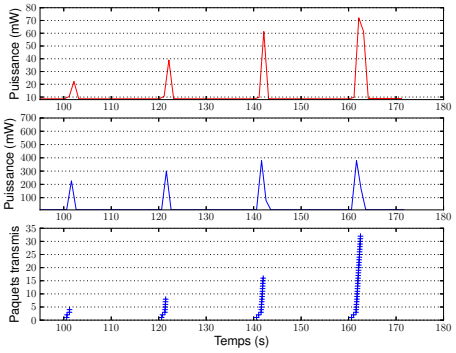
(a) X-MAC doté d'un LPL de 125ms



(b) X-MAC doté d'un LPL de 250ms



(c) X-MAC doté d'un LPL de 500ms



(d) BAT-MAC

FIGURE 28 – Comportement d'X-MAC face à un trafic en rafale par simulation

500ms) permet de réaliser des économies d'énergie. Ainsi, chaque configuration homogène correspond à un volume de trafic propre (e.g. sporadique, à haute fréquence).

BAT-MAC permet alors, dans des situations où la charge de trafic varie rapidement, d'adopter un LPL long en présence d'un trafic sporadique ou nul, et passer à un LPL court dès que l'augmentation de charge de trafic survient. Ainsi, BAT-MAC affiche des délais inférieurs à ceux obtenus avec X-MAC doté d'un LPL de 125 ms (i.e. $0,99$ ms contre $1,11$ ms pour une série de 8 messages, soit une diminution de 11%), pour une consommation énergétique moindre que celle obtenue avec X-MAC doté d'un LPL de 500 ms (i.e. 25.64 mW contre 83.59 mW pour le nœud émetteur d'une série de 8 messages, soit une réduction de 70%), que ce soit en avec ou sans charge de trafic.

Ainsi, dans ce contexte, BAT-MAC permet de réaliser des gains sur les deux tableaux (i.e. performances et consommation énergétique), sans jamais endom-

mager le fonctionnement des nœuds. Il permet une meilleure prise en charge du trafic, entraînant une réduction des délais et de l'occupation du canal radio. Parallèlement, il entraîne une réduction de la consommation énergétique du nœud récepteur, sans accroître celle de l'émetteur. Ces résultats obtenus sans contention ni compétition pour le canal radio permettent donc d'avoir un aperçu des gains potentiels introduits par BAT-MAC dans un cadre favorable.

5.4.2 *Évaluation en environnement avec contention*

Par la suite, nous avons utilisé un réseau complet composé de 50 nœuds, répartis sur une grille de $50\mathrm{m} \times 50\mathrm{m}$, de façon à évaluer notre protocole dans un environnement réaliste (e.g. compétition pour le canal radio, collisions). Nous avons aussi mis en place un modèle de routage en gradient [74], chaque nœud étant alors séparé du puits d'au plus 7 sauts. Nous avons enfin implanté un modèle de trafic, pour lequel une série de 10 messages est transmise toutes les $500\mathrm{ms}$ par un nœud tiré au hasard dans le réseau, reproduisant ainsi de brusques augmentations de la charge de trafic en différents points du réseau.

Comme nous pouvons le constater dans le Tableau 9, BAT-MAC induit une consommation énergétique moyenne réduite (entre 4% et 37% moindre qu'X-MAC, selon le LPL choisi pour ce dernier, avec un intervalle de confiance inférieur à 1%), en permettant un acquittement plus rapide des préambules des messages émis en rafale. Parallèlement, comme décrit en Figure 29, BAT-MAC permet de diminuer sensiblement le délai moyen à un saut de la série de messages transmis, en passant de $118,9\mathrm{ms}$ en moyenne dans le meilleur des cas pour X-MAC à $100,6\mathrm{ms}$ pour BAT-MAC, avec un intervalle de confiance inférieur à $0,1\mathrm{ms}$. L'acquittement plus rapide des messages entraîne donc une diminution du délai moyen d'au moins 16%.

TABLE 9 – Consommation énergétique moyenne des nœuds du réseau

Protocole MAC considéré	Moyenne	Int. de conf. à 95%
BAT-MAC	$9,74\mathrm{mW}$	$0,07$
XMAC-125	$26,06\mathrm{mW}$	$0,07$
XMAC-250	$15,18\mathrm{mW}$	$0,10$
XMAC-500	$10,05\mathrm{mW}$	$0,13$

Ainsi, BAT-MAC permet de traiter plus efficacement les variations brusques de trafic, en diminuant sensiblement le délai moyen tout en améliorant la préservation énergétique dans le réseau. Il présente donc une solution particulière-

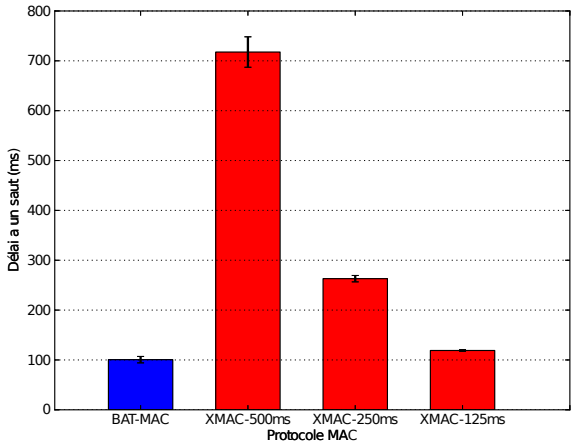


FIGURE 29 – Délai moyen à un saut pour une série de messages

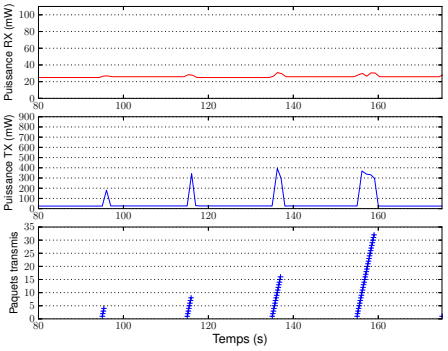
rement efficace pour gérer les réseaux à trafic variable (e.g. sujets aux pertes de connectivité, reposant sur un modèle de trafic orienté-événements), sans nécessiter le moindre message de contrôle.

5.5 ÉVALUATION EXPÉRIMENTALE DES PERFORMANCES

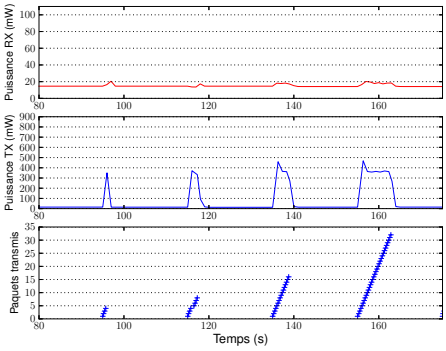
Par la suite, nous avons implémenté et évalué expérimentalement les performances du protocole BAT-MAC, afin de vérifier si les observations faites par simulation tenaient toujours dans des conditions réelles, et ainsi pouvoir les valider intégralement. Nous avons alors transposé les campagnes d'évaluations menées jusque là sur simulateur, dans des conditions réelles.

5.5.1 *Expérimentation sans contention*

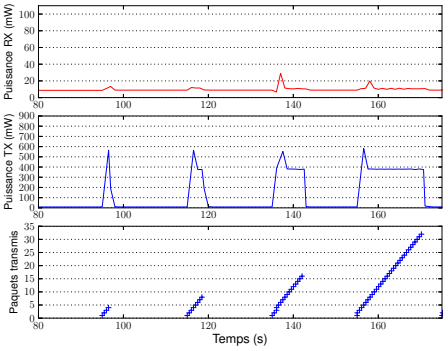
Tout d'abord, nous avons reconduit notre évaluation en dehors de toute contention présentée dans la section 5.4.1 sur deux nœuds TelosB (c.f. chapitre 2). Les résultats correspondants, fournis en figure 30, confirment ceux obtenus par simulation. En effet, là encore, une configuration homogène X-MAC ne permet pas de faire face à toutes les situations. Un LPL de 125ms entraine des délais réduits et une consommation énergétique réduite en cas de charge de trafic importante, alors qu'un LPL de 500ms entraine une consommation énergétique faible en l'absence de trafic mais ne peut supporter des charges de trafic élevées. Dans cette situation réelle, BAT-MAC permet là encore de



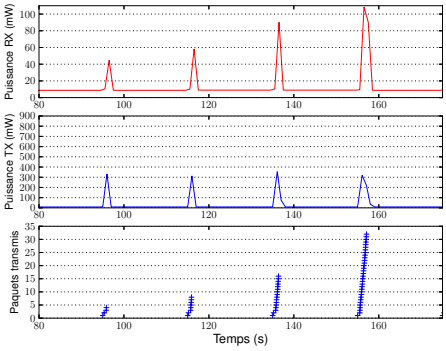
(a) X-MAC doté d'un LPL de 125ms



(b) X-MAC doté d'un LPL de 250ms



(c) X-MAC doté d'un LPL de 500ms



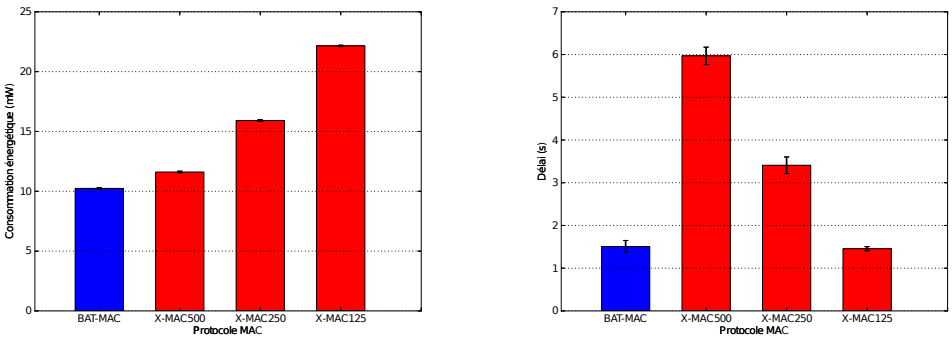
(d) BAT-MAC

FIGURE 30 – Comportement face à un trafic en rafale par expérimentation

diminuer sensiblement le délai moyen (et parallèlement l'occupation du canal) lors des hausses de trafic, qui passe alors de 1297 ms avec X-MAC doté d'un LPL de 125 ms à 940 ms avec BAT-MAC. De la même manière, BAT-MAC permet de diminuer de plus de moitié la consommation énergétique du nœud émetteur, sans augmenter celle du nœud récepteur.

5.5.2 *Expérimentation sur un réseau complet*

Ensuite, nous avons déployé BAT-MAC en conditions réelles sur la plateforme IoT-LAB (c.f. chapitre 2.5.2), avec un modèle de remontée des données au puits à trafic constant [88]. Ici, chaque nœud envoie alors 10 messages d'affilée toutes les 1000 secondes. Nous avons choisi cette valeur élevée pour éviter une sur-congestion du réseau avec une configuration homogène du LPL.



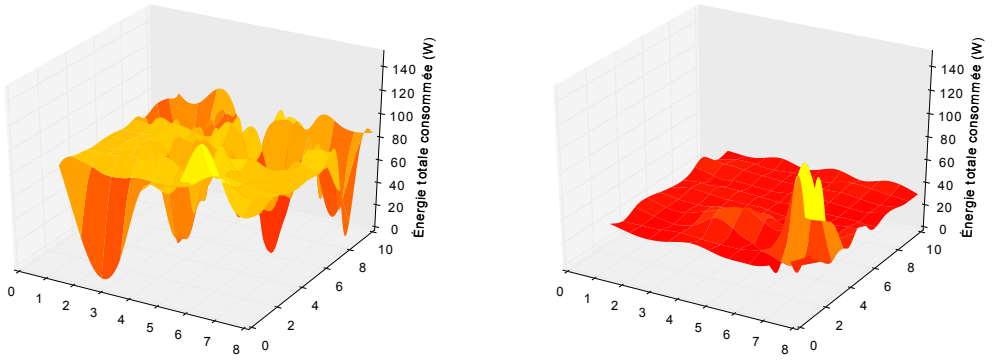
(a) Consommation énergétique moyenne (b) Délai moyen à un saut pour une série de 10 messages

FIGURE 31 – Performances d'X-MAC et de BAT-MAC en situation réelle

Les résultats de cette campagne expérimentale sont fournis par la figure 31. Comme le montrent les résultats portant sur la consommation énergétique, illustrés par la figure 5.31(a), BAT-MAC permet dans ce cas de figure de porter cette dernière à $10mW$ en moyenne, contre au moins $12mW$ pour une configuration homogène du réseau (avec un intervalle de confiance inférieur à $1mW$ pour chaque cas de figure), soit une réduction de près de 16% face à un LPL homogène de $500ms$ et jusqu'à 55% face à un LPL homogène de $125ms$. De manière similaire et comme illustré par la figure 5.31(b), BAT-MAC permet de réduire le délai moyen des séries de messages à $100ms$, soit 12% de moins qu'une configuration homogène de $125ms$ et jusqu'à 7 fois moins qu'une configuration homogène de $500ms$. Ainsi, BAT-MAC permet de s'adapter dynamiquement et efficacement à chaque charge de trafic, et offre de meilleures performances qu'une configuration homogène d'X-MAC quelle que soit la situation considérée.

Nous avons enfin effectué une cartographie de la consommation au sein du réseau, afin d'identifier sa répartition et sa disparité entre les nœuds. Pour cela, nous avons mesuré la consommation énergétique totale de chaque nœud pour la durée totale d'une expérimentation. Comme le montrent les résultats de cette analyse, illustrés par la figure 32, la consommation des nœuds dans le cas d'une configuration homogène de X-MAC varie grandement en fonction de leur rôle dans le réseau. En effet, les nœuds devant relayer des informations consomment en moyenne de $80W$ à $100W$ pour la durée d'expérimentation, contre $25W$ en moyenne pour les feuilles de l'arbre de routage. A contrario, la consommation des nœuds utilisant BAT-MAC reste homogène dans le réseau à $30W$ en moyenne, quel que soit le rôle des nœuds dans ce dernier (à l'exception notable du puits, celui-ci devant prendre en charge l'ensemble des

messages). En effet, en traitant plus efficacement et plus rapidement les variations soudaines de charge de trafic, les nœuds diminuent sensiblement leur consommation énergétique. Ainsi, en évitant les fortes disparités dans la consommation énergétique des nœuds, BAT-MAC permet d'en accroître la durée de vie.



(a) X-MAC doté d'un LPL de 125ms (b) BAT-MAC

FIGURE 32 – Distribution de la consommation énergétique dans le réseau

Par cette étude expérimentale, nous avons démontré le réalisme et les bonnes performances affichées par notre protocole BAT-MAC. Ce dernier permet aux nœuds de s'adapter efficacement et rapidement à chaque charge de trafic, en les dotant d'une configuration spécifique répondant à cette situation. Ainsi, BAT-MAC permet de réduire jusqu'à 7 fois les délais des messages, en abaissant la consommation énergétique jusqu'à 55%, tout en l'homogénéisant au sein du réseau.

5.6 CONCLUSION

Nous avons proposé dans le présent chapitre un nouveau protocole MAC, dénommé BAT-MAC, permettant d'adapter automatiquement et à la volée la configuration MAC d'un nœud en fonction des variations de trafic dans le réseau. Cette approche se distingue de celles présentées dans le chapitre précédent, en permettant une anticipation des changements de charge de trafic, et en introduisant une auto-adaptation immédiate à ces situations. À partir d'une information locale uniquement et sans nécessiter aucun trafic de contrôle, BAT-MAC permet d'adapter temporairement la configuration des seuls nœuds concernés par ces hausses temporaires de trafic, afin de gérer

cette phase plus efficacement. En outre, en présence d'un trafic plus sporadique (comme prévu par l'application), BAT-MAC ne modifie pas le comportement initialement prévu du réseau. Nous avons démontré que BAT-MAC permet ainsi de réduire tant le délai des messages transmis que la consommation énergétique des nœuds, tout en homogénéisant cette dernière au sein du réseau, entraînant par là même une augmentation de la durée de vie du réseau. BAT-MAC se distingue des solutions proposées à l'heure actuelle (c.f. section 2) par un fonctionnement localisé permettant une adaptation à la volée de la configuration MAC des nœuds en fonction de la charge de trafic du réseau.

Cependant les variations de charges de trafic ne sont pas le seul problème auquels les réseaux de capteurs doivent faire face. En effet, leur nature instable et contrainte les rend particulièrement sujets aux pannes et aux pertes de données. Dans un contexte où chacune d'entre elles peut s'avérer vitale, des mécanismes de tolérance aux pannes et aux fautes doivent être mis en œuvre, et le réseau doit s'organiser automatiquement et dynamiquement pour permettre aux données d'être préservées en toutes circonstances. Dans le prochain chapitre, nous présenterons LIFT, un mécanisme permettant un stockage distribué des données entre les nœuds en cas de panne survenant dans le réseau.

6

LIFT : UN MÉCANISME SANS ÉTAT DE TOLÉRANCE AUX PANNES

"La seule voie qui offre quelque espoir d'un avenir meilleur pour toute l'humanité est celle de la coopération et du partenariat."

— Kofi Annan (1938 -)

Sommaire

6.1 INTRODUCTION

Dans les chapitres précédents, nous avons présenté plusieurs contributions permettant d'adapter automatiquement la configuration des nœuds en fonction des conditions de déploiement et de la dynamique du réseau (i.e. critères applicatifs, propriétés de la topologie de routage, variations de charge de trafic). Néanmoins, celles-ci ne permettent pas d'endiguer les pertes causées par des pannes temporaires dans le réseau. Or, il n'est pas rare que le puits de collecte devienne temporairement injoignable pour tout ou partie des nœuds. En effet, ceux-ci sont par nature sujets aux défaillances [24], et les liens radio (surtout sur une gamme de fréquences aussi faible) sont fluctuants et instables [19, 20]. De plus, de nombreuses applications médicales nécessitent l'emploi de nœuds mobiles (e.g. capteurs physiologiques sur le patient) [15, 22].

Pourtant, les informations collectées par les capteurs pouvent s'avérer critiques, en particulier pour des applications médicales. Par exemple, la perte momentanée des relevés de rythme cardiaque ou d'oxygénation peut perturber le suivi du patient en empêchant la détection rapide d'un œdème pulmonaire, d'une apnée du sommeil ou encore d'une Bradycardie [1]. Des mécanismes d'auto-adaptation sont donc nécessaires pour stocker temporairement ces mesures, jusqu'à restauration de la connectivité du réseau.

Néanmoins, les nœuds le composant sont fortement contraints et disposent de ressources limitées, notamment en terme de mémoire. La capacité de stockage des seuls nœuds concernés (i.e. ceux désirant transmettre leurs informations) ne suffit alors plus. Par exemple, dans [1], les capteurs mesurent et transmettent le rythme cardiaque et l'oxygénation sanguine de chaque patient à une fréquence de quatre messages par minute, ce qui conduirait à épuiser la mémoire en moins d'une heure en cas de rupture du réseau (les capteurs disposant d'une mémoire de 1Mo). Nous introduisons donc dans le présent chapitre *LIFT* (pour *Layer Independant Fault Tolerance mechanism* [87], un mécanisme sans-état de tolérance aux pannes. Il tire profit de la forte densité et de la présence de nœuds *Passifs* (dont la présence peut par exemple résulter des mécanismes d'ordonnancement d'activité introduits dans le chapitre 3) pour permettre un stockage distribué des informations suite à la partition du réseau, en vue de les conserver et les transmettre une fois la connectivité rétablie.

6.2 FONCTIONNEMENT DE LIFT

L'objectif principal de LIFT est de permettre un stockage efficace des informations (i.e. des messages) lorsque la connectivité au puits de collecte n'est plus assurée, et donc lorsque ce dernier devient injoignable. Les réseaux de capteurs sans-fil étant par nature fortement dynamiques (e.g. nœuds mobiles, sujets aux pannes, liens radio instables), ce type de situations n'est pas rare, et entraîne invariablement la perte des messages émis dès lors. LIFT vise donc à permettre le stockage par un nœud de ses informations propres, mais aussi à faire participer son voisinage dans ce processus, si nécessaire.

Une fois la connectivité du réseau restaurée, les paquets mis en mémoire sont acheminés au puits de manière à permettre une collecte des données fiable (i.e. en évitant une grande partie des pertes). LIFT fonctionne indépendamment de la pile de communication, et n'influe donc pas sur le fonctionnement général des autres couches du réseau (e.g. Routage, Application). Les messages stockés sont en effet retransmis naturellement une fois la connectivité restaurée, comme s'ils avaient simplement été retardés.

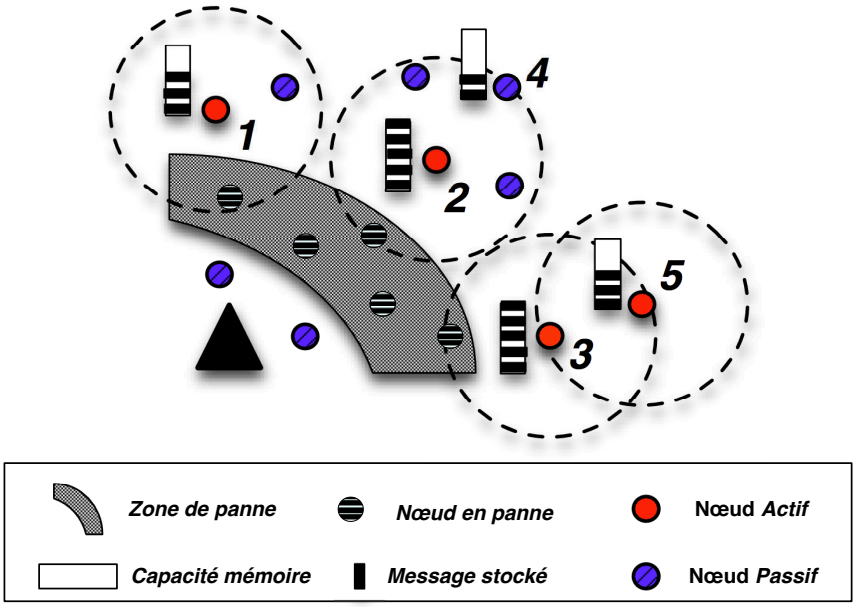


FIGURE 33 – Illustration du processus de stockage distribué de LIFT

6.2.1 *Stockage par les nœuds Actifs*

LIFT commence à opérer dès lors que le prochain saut d'un nœud est injoignable. La détection de ce phénomène a fait l'objet de plusieurs contributions [64, 61], reposant par exemple sur l'envoi régulier de messages de contrôle ou sur la définition d'un seuil de messages successifs non acquittés (comme c'est le cas pour le mécanisme de détection des voisins inaccessibles d'IPv6 [62]). Dès qu'un nœud détecte que son prochain saut n'est plus accessible, il initie le stockage des messages reçus ou générés par lui-même, comme illustré par les capteurs 1, 2 et 3 dans la figure 33.

Ce nœud se comporte de cette manière jusqu'à ce que son prochain saut redevienne joignable (c.f. section 6.2.3), ou jusqu'à ce que sa capacité de stockage dépasse un certain seuil (fixé en fonction de sa capacité de stockage maximale). En effet, du fait de leurs faibles ressources, les nœuds ne peuvent stocker qu'un nombre très limité de messages. De toute évidence, étant donné les architectures considérées (c.f. section 2), l'espace mémoire requis pour la tolérance aux pannes pourrait inciter les concepteurs d'applications à ne pas considérer d'éventuelles pertes de connectivité de longue durée.

Par conséquent, nous supposons ici qu'un nœud *Actif* détectant la perte de connectivité avec son prochain saut ne stockera les messages que pendant un

intervalle de temps limité, selon ses ressources et la fréquence d'arrivée des messages de ses voisins. Nous avons donc défini le seuil S_{Stock}, correspondant au nombre maximum de messages qu'un nœud *Actif* est à même de stocker avant de se considérer lui-même comme plein. Une fois que le nombre de messages stockés par un nœud *Actif* atteint S_{Stock}, il commence à solliciter l'assistance de nœuds *Passifs* dans son voisinage direct. Cet événement est illustré par les nœuds 2 et 3 sur la figure 33. Chacun d'eux ayant atteint sa limite de stockage, ils initient leur recherche d'un relais passif.

Ce seuil S_{Stock} doit être choisi avec précaution. En effet, un seuil trop élevé pourrait entraîner la perte de messages reçus ou générés lors de la recherche d'un relais *Passif*. De la même manière, un seuil faible pourrait déclancher cette élection prématurément, et donc engendrer l'émission de messages de contrôle sans que cela soit nécessaire. S_{Stock} peut alors être fixé au préalable par l'utilisateur (e.g. à partir d'une étude du déploiement), ou par apprentissage à partir de la féquence des mesures à stocker (i.e. reçues et générées). Notons enfin que ce seuil n'est pas nécessairement homogène et peut être attribué indépendamment à chaque nœud, par exemple en fonction de la position dans la structure de routage, comme proposé pour le LPL dans la section 4.3.2.

6.2.2 *Stockage par les nœuds Passifs*

Lorsque le seuil de stockage S_{Stock} est atteint pour un nœud u, ce dernier doit initier un processus d'élection entre les nœuds *Passifs* présents dans son voisinage direct. Une fois que l'un d'eux est sélectionné, u peut retransmettre les paquets qu'il reçoit ou génère à ce nœud *Passif* (alors dénommé relais) en vue de les y stocker. Le stockage devient alors distribué, afin d'éviter la perte des messages pour des pannes plus longues. Cette situation est illustrée par le nœud *Actif* 2 et le relais *Passif* 4 dans la figure 33.

De façon à sélectionner un nœud voisin *Passif*, u diffuse un message de sollicitation de stockage. Tout nœud *Passif* l'acceptant envoie alors en retour une réponse positive, incluant éventuellement des informations complémentaires sur son statut (e.g. état de sa batterie, capacité de stockage restante). Ainsi, à partir des réponses obtenues, u sélectionne son relais *Passif*, à qui il transfèrera tout message pour sauvegarde. La sélection de ce relais peut se faire selon plusieurs critères, soit suivant son statut (e.g. le nœud disposant du plus large espace de stockage, celui dont les ressources énergétiques sont les plus grandes) soit aléatoirement (e.g. le premier à répondre à la sollicitation). Par la suite, nous avons retenu cette dernière approche, car elle ne nécessite aucune

Événement déclancheur	**Opérations de LIFT**
Prochain saut inaccessible	- Stockage local ses messages jusqu'à S_{Stock}
Seuil de stockage S_{Stock} atteint	- Élection d'un voisin *Passif* comme relais - Retransmission de tous les messages au relais
Seuil de stockage du relais atteint	- Notification du nœud *Actif* - Élection d'un nouveau relais *Passif*
Aucun relais *Passif* disponible	- Simulation d'une panne

TABLE 10 – Résumé des opérations de stockage de LIFT.

information complémentaire et ne s'applique pas à un type de déploiement spécifique.

Comme nous l'avons mentionné plus haut, les nœuds composant le réseau disposent de capacités de stockage finies. Tout nœud *Passif* sollicité pour aider au stockage des messages de u peut alors émettre un message en retour, demandant de mettre fin à son processus de stockage. Cela déclenche en réaction l'élection d'un nouveau relais pour suppléer au stockage des messages de u. Notons que cette opération de stockage par les voisins *Passifs* est sans-état, puisqu'aucune information identifiant le relais responsable du stockage d'un message donné n'est enregistrée par le nœud *Actif* correspondant. Nous autorisons aussi plusieurs sélections d'un même nœud *Passif* par plusieurs nœuds *Actifs* différents.

Lorsqu'aucun voisin *Passif* ne répond plus aux sollicitations de stockage (i.e. soit du fait de la mobilité du nœud, soit parce que tous ont atteint leur seuil de stockage), u doit simuler sa panne en arrêtant d'acquitter les messages qu'il devrait retransmettre. Par ce biais, tout nœud *Actif* l'ayant pour prochain saut vers le puits déclanchera à son tour le fonctionnement de LIFT. Cette approche a pour avantage d'être économe, car elle ne nécessite aucun message de contrôle supplémentaire. Cette situation est illustrée dans la figure 33. Le nœud 3 ayant dépassé son seuil S_{Stock} et n'ayant pas de relais disponible, il simule sa panne. Ainsi, son fils (i.e. le nœud 5) enclenche à son tour le fonctionnement de LIFT.

Un résumé des opérations de stockage de LIFT est proposé dans le tableau 10.

6.2.3 *Gestion de la reprise*

Une fois la reprise du fonctionnement normal du réseau (i.e. le puits de collecte étant joignable à nouveau), tout message stocké doit alors être retransmis à un prochain saut valide. Cependant, selon le nombre de nœuds ayant participé au stockage, cette opération peut entraîner une congestion du réseau. Afin de contourner ce problème, plusieurs techniques d'ordonnancement des émissions ont été implémantées dans LIFT.

Ainsi, LIFT notifie graduellement les nœuds de la reprise du réseau. Tout nœud *Actif* situé en bordure de la zone de panne (i.e. ayant un prochain saut jusqu'alors en panne) initie en premier la retransmission des messages stockés localement. Ensuite, il diffuse une information de rétablissement du réseau à ses relais *Passifs*. Ces derniers lui transmettent alors à leur tour leurs messages stockés. Ainsi, les messages sont retransmis sans modification, comme si le relais *Passif* avait été un saut supplémentaire dans leur acheminement au puits.

Enfin, une fois que tous ses relais *Passifs* ont vidé leur stock de messages un nœud *Actif* simulant sa panne peut reprendre son comportement normal, soit en acquittant de nouveau les messages réguliers de détection de panne, soit en le signalant directement par l'intermédiaire d'un message. Pour déterminer précisément ce moment, les nœuds *Actifs* devraient tenir un état de leurs relais sollicités et des messages stockés. Néanmoins, cette approche ne respecterait pas la nature sans-état de LIFT. Nous avons donc préféré instaurer un délai d'attente fixe avant que les nœuds simulant leur panne puissent signaler leur reprise. Ce délai peut être déduit empiriquement (i.e. à partir de campagnes de test) ou estimé à partir du nombre de messages stockés et de l'espace mémoire des nœuds. Dès lors, leurs *fils* pourront à leur tour profiter de la reprise du réseau. LIFT fonctionne donc récursivement jusqu'à ce que tous les messages stockés soient acheminés à leur tour.

L'oronnancement de la reprise sur panne est résumé dans le tableau 11.

Notons qu'il est possible d'améliorer encore la phase de reprise, en faisant fonctionner LIFT en conjonction avec BAT-MAC (c.f. chapitre 5). En effet, les nœuds de stockage vident les messages en mémoire en les envoyant les uns directement à la suite des autres (i.e. en rafale), afin de permettre une reprise sur panne rapide. Or, comme nous l'avons vu dans le chapitre précédent, BAT-MAC permet dans ces circonstances de réduire l'occupation du canal radio et les délais des messages émis. Ainsi, BAT-MAC pourrait faciliter et accélérer la reprise du réseau, tout en limitant la congestion du canal radio. Nous avons dans un premier temps fait le choix de garder LIFT indépendant de la pile de

Événement déclancheur	**Opérations de LIFT**
Prochain saut accessible	- Transmission des messages stockés localement
Plus de message en stock	- Notification de reprise aux voisins *Passifs* - Les relais transmettent les messages stockés - Attente de la fin de transmission des relais
Délai d'attente écoulé	- Notification de reprise aux *fils*

TABLE 11 – Récapitulatif de l'ordonnancement de la reprise par LIFT.

communication, en l'évaluant sans adaptation de la couche MAC. Néanmoins, nous envisageons dans nos travaux ultérieurs d'évaluer en détail les gains pouvant être obtenus avec une telle collaboration.

LIFT n'interagit avec aucun protocole de contrôle topologique, de routage ou MAC, et est par cela complètement indépendant de la pile de communication. De plus, tant que le seuil S_{Stock} n'est pas atteint, les nœuds *Actifs* ne génèrent pas de messages supplémentaires. Ensuite, chaque élection requiert l'émission d'un message de sollicitation de stockage par le nœud *Actif* ainsi qu'un message par nœud *Passif* avoisinant. De manière similaire, la phase de reprise induit au plus un message de contrôle par nœud *Actif* ayant stocké des messages. Enfin, la transmission de messages entre nœuds *Actif* et *Passif* peut être vue comme un saut supplémentaire en direction du puits.

6.3 ÉVALUATION DE LIFT PAR SIMULATION

Afin d'évaluer les performances de LIFT, nous avons mené une série de simulations, avec le simulateur WSNet comme décrit dans la section 2.5.1. Pour cela, nous avons déployé un réseau et introduit un trafic de données initié par chaque nœud, en direction du puits. Après 10 minutes de fonctionnement, nous avons coupé les nœuds voisins du puits de collecte, afin de simuler une panne du réseau. Ainsi, nous avons pu mesurer le comportement du réseau avec et sans LIFT.

6.3.1 *Comportement général de LIFT*

La figure 34 détaille le déroulement d'une simulation typique. Le temps t y est représenté en abscisse, et le nombre de paquets émis par les nœuds, stockés,

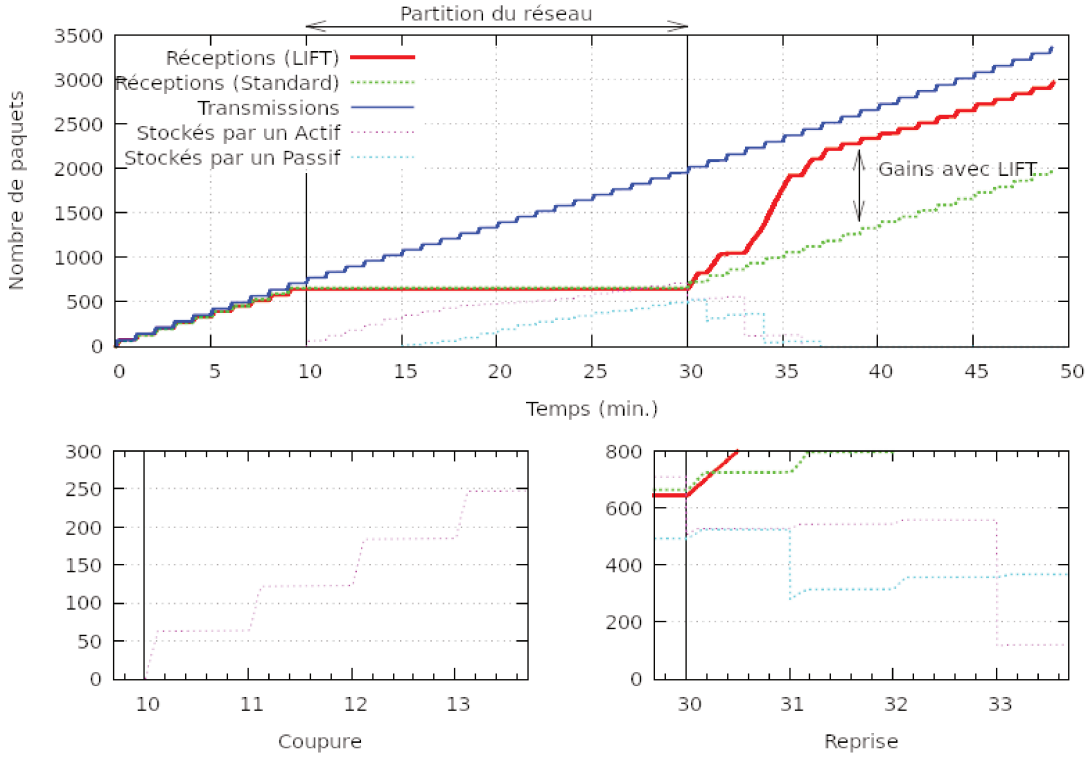
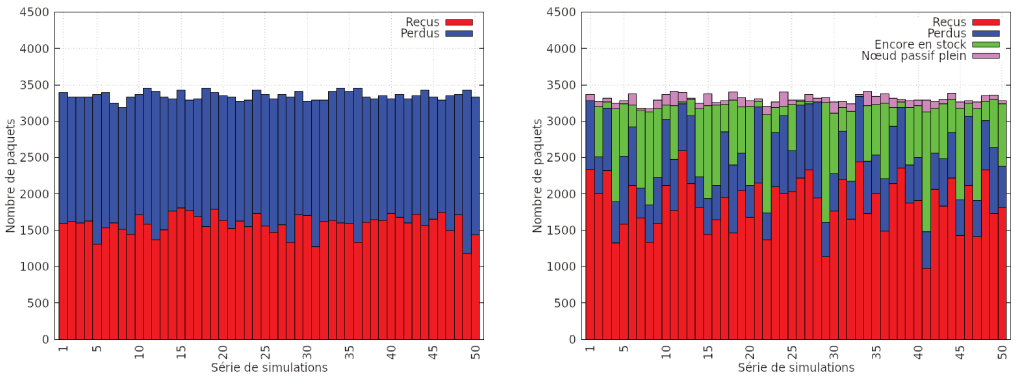


FIGURE 34 – Vue d'ensemble du fonctionnement de LIFT

ou reçus par le puits sont représentés en ordonnée. Ici, le réseau fonctionne normalement les 10 premières minutes, temps à partir duquel la partition prend effet. Comme nous pouvons le constater, quelques paquets sont perdus durant cette période, principalement du fait des collisions sur les liens. Ensuite, à $t = 10min$, la coupure du réseau prend effet, les nœuds voisins directs du puits cessant d'acquitter les messages. Comme nous sommes partis du principe que la détection de la joignabilité du prochain saut n'était pas opérée par LIFT mais par un mécanisme annexe, nous avons implémenté un système de notification automatique. Cela explique la réactivité de LIFT observée dans le détail de la coupure du réseau. À partir de cet instant, tous les nœuds *Actifs* notifiés de la rupture du réseau commencent alors à stocker leurs propres paquets ainsi que ceux qu'ils devraient relayer au puits.

À $t = 15,02min$, le premier nœud *Actif* atteint son seuil de stockage S_{Stock}, et sollicite donc l'assistance de ses voisins *Passifs*. Suite à cette requête, le premier paquet est stocké par un nœud *Passif* à $t = 15,04min$. Ensuite, les messages sont distribués entre nœuds *Actifs* et *Passifs* jusqu'à la reprise du fonctionnement normal du réseau à $t = 30min$. À cet instant, tous les nœuds *Actifs*



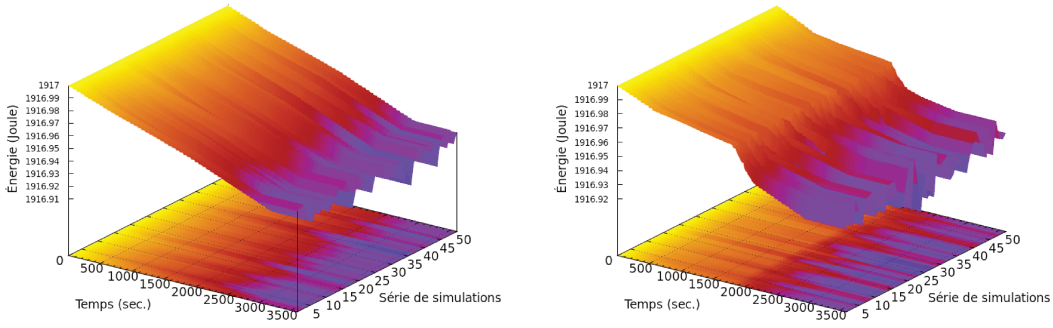
(a) Comportement standard (b) LIFT

FIGURE 35 – Étude des raisons conduisant aux pertes de messages

réactivés diffusent un message signifiant la reprise du réseau aux nœuds *Actifs* voisins. La réception d'un tel message déclenche alors la retransmission des paquets stockés. Une fois cette étape achevée, ces nœuds *Actifs* diffusent une notification à leurs voisins *Passifs* les ayant assistés pour le stockage. Cela explique le délai entre la retransmission des paquets stockés par les nœuds *Actifs* et *Passifs*. Ce délai limite aussi, jusqu'à un certain point, la congestion du médium radio induit par la retransmission de tous les messages stockés, comme le montre l'encart détaillant la reprise. De plus, nous pouvons constater que des paquets continuent à être stockés par des nœuds *Actifs* ou *Passifs* après la reprise. Cela est dû au délai avant la propagation du message de reprise plus loin dans le réseau.

6.3.2 *Étude des performances de LIFT*

Nous pouvons observer que même en utilisant LIFT, le nombre de paquets reçus par le puits diffère du nombre de paquets générés par les nœuds. La figure 35 illustre l'état de chaque paquet généré à l'issue de la simulation, que le réseau implémante LIFT ou non. Dans cette figure, les résultats sont classés par série de simulation. En effet, les valeurs moyennes ne sont pas pleinement représentatives dans ce cas, du fait des changements topologiques entre plusieurs séries de simulations. En moyenne, 3000 paquets sont générés pour chaque simulation. Dans le cas du comportement standard du réseau (i.e. sans implémenter LIFT), le nombre de paquets reçus par le puits est compris entre 1547 et 1624 pour 95% des valeurs. De toute évidence, tous les paquets générés lors de la partition du réseau (1100 paquets en moyenne) sont définitivement perdus par les nœuds voisins de la zone de coupure. En revanche, LIFT permet



(a) Comportement standard (b) LIFT

FIGURE 36 – Consommation énergétique du réseau

au puits de recevoir entre 1774 et 1974 paquets pour 95% des valeurs. Ceci est en deça de nos prévisions, mais peut être expliqué par plusieurs facteurs.

Tout d'abord, il est possible que des nœuds *Passifs* ne reçoivent pas le message de notification de reprise du réseau, et conservent alors leurs messages stockés indéfiniment. Ce comportement est illustré dans la figure 35 par les messages encore en stock à la fin de la simulation. Bien que ce problème puisse être facilement atténué en envoyant les notifications de reprise en unicast indépendamment à chaque voisin *Passif*, cela signifierait aussi que les nœuds *Actifs* doivent tenir une liste des identifiants de chacun d'entre eux, mettant ainsi en question le caractère sans-état de LIFT. Le second facteur vient quant à lui du fait que lorsqu'un nœud *Passif* a atteint son seuil de stockage, il doit en avertir son voisin *Actif*. Or, là encore, ce paquet peut être perdu ou subir des collisions, de sorte que ce dernier peut ne pas parvenir à sa destination. Dans ce cas, le nœud *Actif* continue de retransmettre ses paquets à ce nœud *Passif*. Ce problème est assez marginal, comme l'illustre la figure 35 concernant les messages perdus du fait d'un nœud passif plein. Enfin les paquets de données peuvent être perdus lors de la reprise du réseau. En effet, à ce moment, la charge de trafic augmente soudainement, et conduit donc à un accroissement des pertes de messages au niveau MAC dues à la congestion et l'engorgement du canal radio.

Enfin, nous avons évalué l'impact énergétique de LIFT en présentant en figure 36 la consommation du réseau avec et sans ce mécanisme. Comme nous pouvons le constater, la consommation énergétique augmente à partir de $t = 1800s$, ce qui correspond à la période de reprise du réseau et résulte de l'augmentation soudaine de la charge de trafic. Nous pouvons observer que la

consommation énergétique introduite par LIFT est similaire à celle induite par le comportement standard, pour un taux de perte plus faible.

6.4 CONCLUSION

Dans ce chapitre, nous avons présenté un mécanisme de tolérance aux pannes dénommé LIFT. Ce dernier permet de stocker de manière distribuée les messages envoyés ou générés dans le réseau lorsque le puits de collecte n'est plus joignable. Pour cela, LIFT permet dans un premier temps de mettre localement en mémoire les messages. Puis, une fois un seuil de stockage atteint, un nœud voisin *Passif* est sélectionné comme relais, et les messages suivants lui seront retransmis afin d'y être conservés. Un nouveau relais est sélectionné tour à tour lorsque la mémoire du précédent est pleine. Enfin, lorsqu'aucun voisin *Passif* ne peut plus participer au stockage, le nœud *Actif* simule sa propre panne. Ses fils déclenchent alors récursivement le fonctionnement de LIFT.

LIFT permet aussi une reprise sur panne fluidifiée, grâce à un mécanisme d'ordonnancement des émissions. Ainsi, les nœuds *Actifs* vident leur stock en premier, avant de notifier leurs voisins *Passifs*. Ceux-ci peuvent alors à leur tour lui retransmettre les messages stockés. Enfin, tout nœud *Actif* ayant simulé sa panne notifie ses fils de sa reprise, de sorte que celle-ci s'opère elle aussi de manière récursive.

Dans la conception de LIFT, nous avons conservé une solution sans-état. En effet, notre mécanisme ne garde pas trace des messages stockés ou des nœuds ayant servi de relais. De plus, LIFT est totalement indépendant de la pile de communication (i.e. il ne repose en aucun cas sur les couches MAC, routage, etc.), et peut donc s'intégrer facilement à n'importe quelle solution existante. Enfin, LIFT permet de favoriser la reprise du réseau, en diminuant de près d'un tiers les messages perdus, sans augmenter significativement la consommation énergétique des nœuds.

7

CONCLUSION GÉNÉRALE ET PERSPECTIVES

"La perfection est atteinte non pas parce qu'il n'y a plus rien à ajouter, mais lorsqu'il n'y a plus rien à retirer."

— Antoine de Saint-Exupéry

Dans le domaine médical, des réseaux de capteurs sans-fil peuvent être déployés pour faciliter et améliorer le suivi des patients ou encore pour faciliter les conditions de vie des personnes dépendantes [1, 10]. Grâce à une collecte fréquente et sur le long-terme de données environnementales ou physiologiques concernant le patient, les capteurs facilitent le diagnostic de certaines pathologies (e.g. tachycardie, bradycardie, œdème pulmonaire) et peuvent détecter rapidement des accidents graves (e.g. chutes, arrêts cardiaques).

Les conditions dans lesquelles ces réseaux sont amenés à évoluer sont souvent variables et peu prévisibles. En effet, la qualité des liens radio peut changer au cours du temps [19] et une partie des nœuds peut se déplacer [22] ou tomber en panne (e.g. batteries épuisées, problème matériel). Or, la plupart des déploiements actuels sont configurés une fois pour toute lors de leur installation, en fonction des conditions rencontrées initialement et des objectifs fixés par l'application (e.g. durée de vie, criticité des données) [1, 24, 25, 26].

7.1 CONTRIBUTIONS DE LA THÈSE

Tout au long de ce travail de thèse, nous avons donc étudié différentes approches pour permettre aux réseaux de capteurs sans-fil de se configurer et s'adapter automatiquement et dynamiquement, en fonction de leurs conditions de déploiement. En particulier, nous avons cherché à réaliser cette auto-adaptation et auto-configuration à trois niveaux. D'une part nous nous sommes intéressés à la gestion de la topologie logique, via un ordonnancement de l'activité des capteurs. Ensuite, nous avons étudié la manière de régir les accès dans cette topologie, afin d'obtenir un réseau auto-configuré, connecté et efficace terme de consommation énergétique. Notre dernière contribution permet

en outre à ce dernier de s'adapter à la dynamique du réseau, notamment aux pannes. Dans un souci de réalisme, nous avons amené nos contributions de la simulation jusqu'aux expérimentations.

7.1.1 *Ordonnancement d'activité*

Dans les réseaux de capteurs sans-fil, les déploiements sont réalisés de manière à produire une topologie dense, fortement redondante [30]. Cette approche se justifie par le fait que ces nœuds, étant généralement alimentés par batterie ou disposant de faibles dispositifs de collecte d'énergie, disposent de ressources énergétiques limitées. En effet, il est possible d'accroître la durée de vie du réseau en ordonnançant l'activité des nœuds le composant. Dans ce contexte, seul un sous-ensemble de ces derniers, dénomés nœuds *Actifs*, participe activement à son fonctionnement. Les nœuds restants entrent ainsi dans un état dit *Passif*, économe en énergie. La durée de vie du réseau est alors accrue en homogénéisant le temps de participation active des nœuds, de manière à les utiliser équitablement.

À partir de là, nous avons montré qu'un sous-ensemble des nœuds *Actifs* est nécessaire à la réalisation des tâches dévolues au réseau (i.e. critère dicté par l'application, la surveillance d'une zone d'intérêt par exemple), sans participer au maintien de la connectivité. Nous avons alors introduit un nouvel état d'activité intermédiaire. Celui-ci se trouve à mi-chemin entre les états *Actif* et *Passif* déterminés par la plupart des mécanismes d'ordonnancement d'activité issus de la littérature. Ce nouvel état d'activité, dénommé *Sensing-Only*, permet aux nœuds concernés d'éteindre leur radio en dehors de leurs propres émissions de manière à préserver leurs ressources énergétiques, et vise à réduire encore davantage la proportion de nœuds *Actifs*, afin d'accroître la durée de vie du réseau. Nous avons proposé deux méthodes pour identifier les nœuds *Sensing-Only*, soit par l'intermédiaire d'un LMST, soit par celui d'un gradient. Nous avons aussi généralisé cette identification aux mécanismes d'ordonnancement par couches d'activité multiples.

Nous avons ensuite évalué par simulation ainsi qu'expérimentalement plusieurs mécanismes d'ordonnancement d'activité ainsi que l'impact de l'introduction de l'état *Sensing-Only*. Nous avons montré jusqu'à quel point ces techniques permettent d'accroître la durée de vie du réseau tout en lui laissant la possibilité de réaliser les tâches lui étant dévolues (i.e. critère dicté par l'application, la surveillance d'une zone d'intérêt par exemple), sans risquer une partition du réseau (i.e. le critère de connectivité).

7.1.2 *Contrôle d'accès au médium*

Les ressources énergétiques d'un capteur sans-fil sont limitées, et la radio en est le composant le plus énergivore [23]. Il convient donc de limiter son utilisation autant que possible. L'usage de la radio étant géré par la couche de contrôle d'accès au médium (MAC), celle-ci a fait l'objet de nombreux travaux visant spécifiquement les réseaux de capteurs. En particulier, les protocoles dits à "échantillonnage de canal" (c.f. section 2.3.2) ont acquis une popularité croissante ces dernières années du fait de leurs performances, leur simplicité et surtout leur faculté à fonctionner sur des réseaux de toutes tailles [50].

Dans ce contexte, la période d'échantillonnage du canal radio porte une grande influence sur les performances des communications (e.g. délais d'envoi des messages, taux de perte, consommation énergétique). Dans la vaste majorité des déploiements actuels, cette période d'échantillonnage est homogène au sein du réseau (i.e. identique pour tous les nœuds) et statique (i.e. invariante au cours du temps). Plusieurs études [29] ont néanmoins démontré que des économies d'énergie supplémentaires pouvaient être réalisées en permettant à plusieurs périodes d'échantillonnage différentes de cohabiter au sein d'un même réseau. Cependant, il est difficile de réaliser une telle configuration hétérogène sans engendrer de déconnexion (e.g. partitionnement du réseau, noeuds isolés) et détériorer les performances du réseau [59].

Dans un premier temps, nous avons alors adapté des solutions d'ordonnancement d'activité, en donnant vie au concept de passivité des nœuds. Pour cela, nous avons affecté à chaque état d'activité (i.e. *Actif*, *Passif*, *Sensing-Only*) une configuration de la période d'échantillonnage spécifique correspondante. Nous avons aussi proposé deux méthodes originales et indépendantes d'autoconfiguration de la couche MAC [89]. La première divise les nœuds en plusieurs profondeurs de sommeil, chacune associée à une configuration spécifique, selon les critères applicatif et de connectivité. La seconde permet d'estimer localement la charge de trafic que chaque nœud devra traiter, par l'intermédiaire d'une analyse de la topologie de routage, et ainsi lui affecter une configuration de la période d'échantillonnage correspondante. Enfin, nous avons proposé une solution pour adapter localement et à la volée la configuration en cas de variations rapides et significatives de la charge de trafic traitée par ces derniers, par l'introduction du protocole MAC BAT-MAC.

7.1.3 *Tolérance aux pannes*

Dans les applications médicales, les informations relayées par les capteurs sont de première importance. Il est donc indispensable de pouvoir garantir leur collecte et leur acheminement, quoi qu'il advienne. Ainsi, lorsqu'une panne survient dans le réseau (i.e. une partie des nœuds n'est plus à même de joindre le puits), les messages doivent être stockés, jusqu'à une reprise totale de son fonctionnement. Cependant les capteurs disposent de capacités de stockage très limitées. Afin de répondre à ces contraintes, nous avons complété nos solutions par un mécanisme de stockage distribué, dénommé LIFT [87].

Lors d'une partition du réseau (le puits n'étant plus joignable), les nœuds en bordure de la zone impactée initient un processus de stockage des données. Ainsi, chacun d'entre eux place dans sa mémoire les informations qu'il récolte aussi bien que les messages reçus de ses voisins (et qu'il devrait en temps normal retransmettre en direction du puits). Lorsque la limite de stockage est atteinte, les nœuds voisins *Passifs* (par exemple issus de l'ordonnancement d'activité) sont mis à contribution pour stocker les informations. Ainsi, l'un d'eux est élu comme relais, et les données lui seront alors retransmises. Une fois tous les relais potentiels saturés ou indisponibles, le nœud impacté par la rupture du réseau simule sa propre panne en cessant de répondre aux messages lui étant destinés. Ses fils déclenchent alors à leur tour le fonctionnement de LIFT. Enfin, lorsque la liaison avec le puits est rétablie, les données sont récoltées par les nœuds *Actifs* auprès de leur voisins *Passifs* pour être acheminées normalement.

Ainsi, nous avons montré par simulation que l'acheminement des informations issues des capteurs n'est pas interrompu, et que ce mécanisme garantit donc la possibilité de leur collecte à long terme de manière efficace et sans rupture. De plus, LIFT n'a pas d'impact significatif sur la consommation énergétique du réseau. Pour toutes ces raisons, LIFT se distingue des autres mécanismes de tolérence aux pannes, qui s'intéressent plus à restaurer la connectivité (e.g choix d'une nouvelle route au niveau routage), mais engendrent invariablement de fortes pertes de messages lors de ce processus. À ce sujet, notons que LIFT peut être combiné à ces mécanismes pour permettre à la fois une reprise plus rapide et une réduction des pertes causées par la panne.

7.2 PERSPECTIVES

Tout au long de nos travaux, nous nous sommes focalisés sur des mécanismes d'auto-configuration et d'auto-adaptation pour lesquels le comportement des nœuds est ajusté en fonction des conditions courantes du déploiement, sans mémoire sur son évolution. En effet, ces solutions ont pour avantage de ne pas nécessiter de période de rôdage (i.e. récupération des conditions de déploiement pendant un intervalle de temps suffisamment long pour pouvoir en tirer des conclusions pertinentes), ce qui leur permet d'être immédiatement opérationnelles. De plus, une telle approche permet d'adapter plus efficacement les réseaux pour lesquels les conditions fluctuent de manière peu prévisible. Cependant, les performances de nos solutions pourraient être améliorées tout en évitant toute période de rôdage, grâce à la mémorisation de certains paramètres. En particulier, une connaissance du trafic moyen et de son évolution au cours du temps permettrait d'ajuster les configurations des nœuds dans notre solution d'estimation du trafic par exemple. De même, l'ensemble des configurations MAC utilisées par nos algorithmes pourrait être ajusté à la volée, en prédiction des conditions de déploiement.

Certaines études récentes [29, 57] se sont penchées sur ce problème en établissant une collecte régulière d'informations sur l'état des liens (e.g. taux de perte, occupation du canal radio, symétrie). Celles-ci sont ensuite envoyées au puits afin de calculer en temps réel la configuration optimale du réseau. Cependant, du fait de leur fonctionnement centralisé, ces solutions s'avèrent complexes et coûteuses en terme de messages de contrôle. Une décision locale, réalisée à partir des informations concernant les nœuds voisins (e.g. en ajoutant aux messages de données l'information sur l'état des liens en amont dans la structure de routage) pourrait permettre d'approcher la configuration optimale du réseau de manière localisée et avec une complexité réduite.

De manière plus générale, un des grands défis à venir pour les réseaux de capteurs sans-fil vient du traitement des données collectées. En effet, les déploiements à large échelle (i.e. plusieurs centaines de nœuds) sont devenus une réalité et permettent de mesurer un large éventail de paramètres simultanément [11, 10]. Cela implique la collecte à haute fréquence d'une large quantité de données devant être stockées ou analysées en temps réel. Les capteurs eux mêmes ne peuvent pas y parvenir du fait de leurs ressources limitées (i.e. capacité de calcul et mémoire) et de leur connaissance partielle (i.e. chaque capteur ne dispose que de ses propres mesures, alors qu'une vue plus globale peut être nécessaire). Le traitement de l'information doit donc être fait en amont, après sa collecte par le puits.

Cette situation soulève de nouveaux défis, auxquels l'informatique dématérialisée et les systèmes distribués peuvent répondre [90]. Pour y parvenir, ces systèmes doivent être adaptés aux spécificités des réseaux de capteurs sans-fil (e.g. collecte à haute fréquence d'une grande quantité d'informations peu volumineuses) afin d'effectuer le stockage ou l'analyse des données de manière suffisamment efficace (i.e. en temps réel) [91, 92]. Alors, les services ainsi offerts pourraient aider à la réalisation d'un grand nombre d'objectifs, allant du stockage sécurisé et robuste [93, 94] à la détection automatisée de pathologies chez des patients suivis à distance [95, 96].

Nos travaux initiés dans le cadre du projet Tamias [1] s'inscrivent dans cette optique en proposant un système de stockage distribué, robuste et sécurisé. En particulier, notre évaluation de ce système nous a permis de montrer que les fonctionnalités offertes par ce dernier (e.g. temps de stockage accéléré pour les données de taille réduite) en faisaient un parfait candidat pour le stockage des données issues d'un réseau de capteurs sans-fil.

1. Projet Tamias - https://tamias.iijlab.net/

LISTE DES ACRONYMES

LIFT Layer Independant Fault Tolerance mechanism

WISEMAC Wireless Sensor MAC

NSF National Science Foundation

PO Positive-Only

PR Positive-Only

DGA Distributed Greedy Algorithm

WSN-HEAP Wireless Sensor Networks Powered by Ambient Energy Harvesting

DAG Directed Acyclic Graph - Graphe acyclique orienté

MST Minimum Spanning Tree - Arbre couvrant de poids minimal

LMST Local Minimum Spanning Tree

LPL Low-Power-Listening

BAT-MAC Burst Adaptative Transmission MAC

B-MAC Berkeley MAC

CSMA/CA Carrier Sense Multiple Access / Collision Avoidance

IETF Internet Engineering Task Force

MAC Medium Access Control - Contrôle d'accès au médium

RPL IPv6 Routing Protocol for Low-Power and Lossy Networks

TDMA Time Division Multiple Access - Accès multiple à répartition dans le temps

LISTE DES FIGURES ET TABLES

TABLE DES FIGURES

LISTE DES TABLEAUX

PUBLICATIONS

REVUES INTERNATIONALES À COMITÉ DE RELECTURE

Thorough Empirical Analysis of X-MAC over a Large Scale Internet of Things Testbed.
Julien Beaudaux, Antoine Gallais, Julien Montavont, Thomas Noël, Damien Roth et Erkan Valentin
Dans *IEEE Sensors Journal*, 2013.

Reality-Proof Activity Scheduling for Energy-Efficient Wireless Sensor Networks.
Julien Beaudaux, Antoine Gallais et Thomas Noël
Dans *InderScience International Journal of Ad Hoc and Ubiquitous Computing*, 2013.

Heterogeneous MAC Duty-Cycling for Energy-Efficient Internet of Things Deployments.
Julien Beaudaux, Antoine Gallais et Thomas Noël
Dans *Springer Networking Science Journal*, 2013.

CONFÉRENCES INTERNATIONALES À COMITÉ DE RELECTURE

Adding value to WSN simulation using the IoT-LAB experimental platform.
Georgios Z. Papadopoulos, Julien Beaudaux, Antoine Gallais et Thomas Noël
Dans *WiMob '13 : 9th IEEE International Conference on Wireless and Mobile Computing, Networking and Communications*, Octobre 2013.

Localized MAC Duty-Cycling Adaptations for Global Energy-Efficiency in Wireless Sensor Networks.
Julien Beaudaux, Antoine Gallais et Thomas Noël
Dans *ISCC '13 : 18th IEEE Symposium on Computers and Communications*, Juillet 2013.

LIFT : Layer Independent Fault Tolerance Mechanism for Wireless Sensor Networks.
Julien Beaudaux, Antoine Gallais, Julien Montavont et Thomas Noël
Dans *VTC '11 : 73rd IEEE Vehicular Technology Conference*, Mai 2011.

CASINO : Creating Alea with a Sensor-based Interactive Network.
Julien Beaudaux, Antoine Gallais, Romain Kuntz, Julien Montavont, Thomas Noël, Damien Roth, Fabrice Theoleyre et Erkan Valentin.
Dans *SenSys '10 : ACM Conference on Embedded Networked Sensor Systems*, Novembre 2010.

Multiple Coverage with Controlled Connectivity in Wireless Sensor Networks.
Julien Beaudaux, Antoine Gallais et Tahiry Razafindralambo
Dans *PE-WASUN '10 : 7th ACM International Symposium on Performance Evaluation of Wireless Ad Hoc, Sensor, and Ubiquitous Networks*, Octobre 2010.

CONFÉRENCES NATIONALES À COMITÉ DE RELECTURE

Auto-configuration Proactive de la Couche MAC pour les Réseaux de Capteurs Sans Fil à Trafic Variable.
Julien Beaudaux, Georgios Z. Papadopoulos, Antoine Gallais et Thomas Noël
Dans *UbiMob '13 : 9èmes Journées Francophones Mobilité et Ubiquité*, Juin 2013.

BIBLIOGRAPHIE

[1] O. Chipara, C. Lu, T. Bailey, and G. Roman, "Reliable clinical monitoring using wireless sensor networks : Experiences in a step-down hospital unit," in *ACM Conference on Embedded Networked Sensor Systems (Sensys)*, pp. 155–168, 2010.

[2] A. Sakurai, M. Nakamura, and J. Nakamura, "Self-organizing map analysis of sensor networks for human movement tracking," *Sensors and Actuators*, vol. 166, no. 1, pp. 141–148, 2011.

[3] S. D. Glaser, "Some real-world applications of wireless sensor nodes," in *SPIE Symposium on Smart Structures and Materials*, vol. 344–355, 2004.

[4] L. Krishnamurthy, R. Adler, P. Buonadonna, J. Chhabra, M. Flanigan, N. Kushalnagar, L. Nachman, and M. Yarvis, "Design and deployment of industrial sensor networks : experiences from a semiconductor plant and the north sea," in *International Conference on Embedded Networked Sensor Systems*, pp. 64–75, 2005.

[5] J. Polastre, R. Szewczyk, and D. Culler, "Telos : enabling ultra-low power wireless research," in *Proceedings of the 4th International Symposium on Information Processing in Sensor Networks (IPSN)*, pp. 364–369, 2005.

[6] C. B. des Roziers, G. Chelius, T. Ducrocq, E. Fleury, A. Fraboulet, A. Gallais, N. Mitton, T. Noel, and J. Vandaele, "Using senslab as a first class scientific tool for large scale wireless sensor network experiments," in *IFIP Networking*, pp. 147–159, 2011.

[7] J. W. Ng, B. P. Lo, O. Wells, M. Sloman, N. Peters, A. Darzi, C. Toumazou, and G.-Z. Yang, "Ubiquitous monitoring environment for wearable and implantable sensors (ubimon)," in *Proceedings of the 6th International Conference on Ubiquitous Computing (UbiComp)*, 2004.

[8] D. Malan, T. Fulford-jones, M. Welsh, and S. Moulton, "Codeblue : An ad hoc sensor network infrastructure for emergency medical care," in *International Workshop on Wearable and Implantable Body Sensor Networks*, 2004.

[9] A. Friggeri, G. Chelius, E. Fleury, A. Fraboulet, F. Mentre, and J. Lucet, "Reconstructing social interactions using an unreliable wireless sensor network," *Computer Communications*, vol. 34, no. 5, pp. 609–618, 2011.

[10] H. Cruz-Sanchez, L. Havet, M. Chehaider, and Y.-Q. Song, "Mpigate : A solution to use heterogeneous networks for assisted living applications," *Proceedings of the 9th International Conference on Ubiquitous Intelligence and Computing and Autonomic and Trusted Computing (UIC/ATC)*, pp. 104–111, 2012.

[11] P. Lopez, D. Fernandez, A. J. Jara, and A. F. Skarmeta, "Survey of internet of things technologies for clinical environments," in *Proceedings of the 27th International Conference on Advanced Information Networking and Applications Workshops (WAINA)*, pp. 1349–1354, 2013.

[12] J. Ko, J. Lim, Y. Chen, R. Musvaloiu, A. Terzis, G. Masson, T. Gao, W. Destler, L. Selavo, and R. Dutton, "Medisn : Medical emergency detection in sensor networks," in *ACM Transactions on Embedded Computing Systems (TECS)*, 2010.

[13] R. S. Dilmaghani, H. Bobarshad, M. Ghavami, S. Choobkar, and C. Wolfe, "Wireless sensor networks for monitoring physiological signals of multiple patients," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 5, no. 4, pp. 347–356, 2011.

[14] S. Erdogan and T. Bilgin, "A data mining approach for fall detection by using k-nearest neighbour algorithm on wireless sensor network data," *IET Communications Journal*, vol. 6, no. 18, pp. 3281–3287, 2012.

[15] J. Olivo, S. S. Ghoreishizadeh, S. Carrara, and G. D. Micheli, "Electronic implants : Power delivery and management," in *Conference on Design, Automation and Test in Europe*, pp. 1540–1546, 2013.

[16] A. J. Jara, M. A. Zamora, and A. F. Skarmeta, "An internet of things-based personal device for diabetes therapy management in ambient assisted living (aal)," *Personal and Ubiquitous Computing*, vol. 15, no. 4, pp. 431–440, 2011.

[17] A. Gaddam, S. C. Mukhopadhyay, and G. S. Gupta, "Elder care based on cognitive sensor network," *IEEE Sensors Journal*, vol. 11, no. 3, pp. 574–581, 2011.

[18] T. W. Hnat, V. Srinivasan, J. Lu, T. I. Sookoor, R. Dawson, J. Stankovic, and K. Whitehouse, "The hitchhiker's guide to successful residential sensing deployments," in *ACM Conference on Embedded Networked Sensor Systems (Sensys)*, pp. 232–245, 2011.

[19] C. A. Boano, J. Brown, Z. He, U. Roedig, and T. Voigt, "Low-power radio communication in industrial outdoor deployments : The impact of weather conditions and atex-compliance," *Sensor Applications, Experimentation, and Logistics*, vol. 29, pp. 159–176, 2010.

[20] S. Lohs, R. Karnapke, and J. Nolte, "Link stability in a wireless sensor network - an experimental study," *Sensor Systems and Software*, vol. 102, no. 2, pp. 146–161, 2012.

[21] H. Hongwei, S. Wei, X. Youzhi, and Z. Hongke, "The effect of human activites on 2.4 ghz radio propagation at home environment," in *Proceedings of the 2nd IEEE International Conference on Broadband Network and Multimedia Technology (IC-BNMT)*, pp. 95–99, 2009.

[22] J. Caldeira, J. Rodrigues, and P. Lorenz, "Toward ubiquitous mobility solutions for body sensor networks on healthcare," *IEEE Communications Magazine*, vol. 50, no. 5, pp. 108–115, 2012.

[23] G. Terrasson, *Contribution à la conception d'un émetteur-récepteur pour microcapteurs autonomes*. PhD thesis, Université de Bordeaux 1, 2008.

[24] G. Barrenetxea, F. Ingelrest, G. Schaefer, M. Vetterli, O. Couach, and M. Parlange, "Sensorscope : Out-of-the-box environmental monitoring," in *Proceedings of the ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, pp. 332–343, 2008.

[25] V. Dyo, S. A. Ellwood, D. W. Macdonald, A. Markham, C. Mascolo, B. Pásztor, S. Scellato, N. Trigoni, R. Wohlers, and K. Yousef, "Evolution and sustainability of a wildlife monitoring sensor network," in *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems (SenSys)*, pp. 127–140, 2010.

[26] A. Dunkels, J. Eriksson, N. Finne, F. Osterlind, N. Tsiftes, J. Abeille, and M. Durvy, "Low-power ipv6 for the internet of things," in *Proceedings of the 9th International Conference on Networked Sensing Systems (INSS)*, pp. 1–6, 2012.

[27] R. Jurdak, P. Baldi, and C. V. Lopes, "Adaptive Low Power Listening for Wireless Sensor Networks," *IEEE Transactions on Mobile Computing (TMC)*, vol. 6, no. 8, pp. 988–1004, 2007.

[28] R. Kuntz, A. Gallais, and T. Noël, "From versatility to auto-adaptation of the medium access control in wireless sensor networks," *Journal of Parallel and Distributed Computing*, vol. 71, no. 9, pp. 1236–1248, 2011.

[29] M. Zimmerling, F. Ferrari, L. Mottola, T. Voigt, and L. Thiele, "ptunes : Runtime parameter adaptation for low-power mac protocols," in *ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, pp. 173–184, 2012.

[30] S. Al-Omari and W. Shi, "Toward highly-available wsns for assisted living," in *Proceedings of the ACM SIGMOBILE International Workshop on Systems and Networking Support for Healthcare and Assisted Living Environments*, pp. 31–36, 2007.

[31] S. Shin, T. Kwon, G.-Y. Jo, Y. Park, and H. Rhy, "An experimental study of hierarchical intrusion detection for wireless industrial sensor networks," *IEEE Transactions on Industrial Informatics*, vol. 6, no. 4, pp. 744–757, 2010.

[32] O. Landsiedel, E. Ghadimi, S. Duquennoy, and M. Johansson, "Low power, low delay : Opportunistic routing meets duty cycling," in *Proceedings of the ACM/IEEE Conference on Information Processing in Sensor Networks (IPSN)*, 2012.

[33] Y. Zhang, N. Meratnia, and P. Havinga, "Outlier detection techniques for wireless sensor networks : A survey," *IEEE Communications Surveys and Tutorials*, vol. 12, no. 2, 2010.

[34] A. Gallais, J. Carle, D. Simplot-Ryl, and I. Stojmenovic, "Localized Sensor Area Coverage with Low Communication Overhead," *IEEE Transactions on Mobile Computing (TMC)*, vol. 7, no. 5, pp. 661–672, 2008.

[35] H. Gupta, Z. Zhou, S. Das, and Q. Gu, "Connected sensor cover : self-organization of sensor networks for efficient query execution," *IEEE/ACM Transactions on Networking (ToN)*, vol. 14, no. 1, pp. 55–67, 2006.

[36] Z. Zhou, S. Das, and H. Gupta, "Variable radii connected sensor cover in sensor networks," *ACM Transactions on Sensor Networks*, vol. 5, no. 1, pp. 1–36, 2009.

[37] H. Kdouh, H. Farhat, G. Zaharia, C. Brousseau, G. Grunfelder, and G. E. Zein, "Performance analysis of a hierarchical shipboard wireless sensor network," in *IEEE 23rd International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, pp. 781–786, 2012.

[38] B. Yener, M. Magdon-Ismail, and F. Sivrikaya, "Joint problem of power optimal connectivity and coverage in wireless sensor networks," *ACM Wireless Networks (WINET)*, vol. 13, no. 4, pp. 537–550, 2007.

[39] B. A. Bash and P. J. Desnoyers, "Exact distributed voronoi cell computation in sensor networks," in *ACM/IEEE Proceedings of Information Processing in Sensor Networks*, 2007.

[40] X.-Y. Li, Y. Wang, and W.-Z. Song, "Applications of k-local mst for topology control and broadcasting in wireless ad hoc networks," *IEEE Transactions on Parallel and Distributed Systems (TPDS)*, vol. 15, no. 12, pp. 1057–1069, 2004.

[41] W. Choi, G. Ghidini, and S. K. Das, "A novel framework for energy-efficient data gathering with random coverage in wireless sensor networks," *ACM Transactions on Sensor Networks (ToSN)*, vol. 8, no. 4, pp. 36 :1–36 :30, 2012.

[42] O. Yang and W. Heinzelman, "A better choice for sensor sleeping," in *European Conference on Wireless Sensor Networks (EWSN)*, pp. 134–149, 2009.

[43] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless Sensor Networks : a Survey," *Computer Networks*, vol. 38, no. 4, pp. 393–422, 2002.

[44] A. Bachir, M. Dohler, T. Watteyne, , and K. Leung, "Mac essentials for wireless sensor networks," *IEEE Communications Surveys Tutorials*, vol. 12, no. 2, pp. 222–248, 2010.

[45] V. Cionca, T. Newe, and V. Dadarlat, "Tdma protocol requirements for wireless sensor networks," in *Proceedings of the 2nd International Conference on Sensor Technologies and Applications (SENSORCOMM)*, pp. 30–35, 2008.

[46] V. Rajendran, K. Obraczka, and J. J. Garcia-Luna-Aceves, "Energy-Efficient, collision-free medium access control for wireless sensor networks," *Wireless Networks*, vol. 12, pp. 63–78, 2006.

[47] K. S. J. Pister and L. Doherty, "TSMP : Time synchronized mesh protocol," in *Proceedings of the IEEE/IFIP International Symposium on Distributed Sensor Networks (DSN)*, pp. 391–398, 2008.

[48] W. Ye, J. Heidemann, and D. Estrin, "Medium access control with coordinated adaptive sleeping for wireless sensor networks," *IEEE/ACM Transactions on Networking*, vol. 12, pp. 493–506, 2004.

[49] T. van Dam and K. Langendoen, "An adaptive energy-efficient MAC protocol for wireless sensor networks," in *Proceedings of the ACM International Conference on Embedded Networked Sensor Systems (SenSys)*, pp. 171–180, 2003.

[50] C. Cano, B. Bellalta, A. Sfairopoulou, and M. Oliver, "Low energy operation in wsns : A survey of preamble sampling mac protocols," *Computer Networks*, vol. 55, no. 15, pp. 3351—3363, 2011.

[51] J. Polastre, J. Hill, and D. Culler, "Versatile low power media access for wireless sensor networks," in *International Conference on Embedded Networked Sensor Systems (SenSys)*, pp. 95–107, 2004.

[52] M. Buettner, G. V. Yee, E. Anderson, and R. Han, "X-mac : a short preamble mac protocol for duty-cycled wireless sensor networks," in *International Conference on Embedded Networked Sensor Systems (Sensys)*, pp. 307–320, 2006.

[53] O. Yang and W. B. Heinzelman, "Modeling and performance analysis for duty-cycled mac protocols with applications to s-mac and x-mac," in *IEEE Transactions on Mobile Computing (TMC)*, vol. 11, pp. 905–921, 2012.

[54] D. Moss and P. Levis, "Box-macs : Exploiting physical and link layer boundaries in low-power networking," Tech. Rep. SING-08-00, Stanford Information Networks Group, 2008.

[55] A. Dunkels, "The contikimac radio duty cycling protocol," Tech. Rep. 5128, Swedish Institute of Computer Science (SICS), 2011.

[56] A. El-Hoiydi and J.-D. Decotignie, "Wisemac : An ultra low power mac protocol for multi-hop wireless sensor networks," *Algorithmic Aspects of Wireless Sensor Networks*, vol. 3121, pp. 18–31, 2004.

[57] X. Wang, X. Wang, G. Xing, and Y. Yao, "Dynamic duty cycle control for end-to-end delay guarantees in wireless sensor networks," in *International Workshop on Quality of Service (IWQoS)*, pp. 1–9, 2010.

[58] H. Yoo, M. Shim, and D. Kim, "Dynamic duty-cycle scheduling schemes for energy-harvesting wireless sensor networks," in *IEEE Communications Letters*, 2012.

[59] C. Merlin and W. Heinzelman, "Duty cycle control for low-power-listening mac protocols," *IEEE Transactions on Mobile Computing (TMC)*, vol. 9, no. 11, pp. 1508–1521, 2010.

[60] P. Jiang, "A new method for node fault detection in wireless sensor networks," *Sensors*, vol. 9, pp. 1282–1294, 2009.

[61] X. Chen, D. Man, Y. Wu, W. Wang, and S. Xuan, "A routing failure node detection method of wireless sensors based on binary encoding," in *Proceedings of the International Conference on Computer Science Service System (CSSS)*, pp. 1130–1133, 2012.

[62] T. Narten, E. Nordmark, and W. Simpson, "Neighbor discovery for ip version 6 (ipv6)." Internet Engineering Task Force Request for Comments (RFC) 2461, 1998.

[63] J. Chen, S. Kher, and A. Somani, "Distributed fault detection of wireless sensor networks," in *Proceedings of the Workshop on Dependability Issues in Wireless Ad-hoc Networks and Sensor Networks (DIWANS)*, pp. 65–72, 2006.

[64] L. Paradis and Q. Han, "A survey of fault management in wireless sensor networks," *Journal of Network and Systems Management*, vol. 15, no. 2, pp. 171–190, 2007.

[65] K. Piotrowski, P. Langendoerfer, and S. Peter, "tinydsm : A highly reliable cooperative data storage for wireless sensor networks," in *Proceedings of the International Symposium on Collaborative Technologies and Systems (CTS)*, pp. 225–232, 2009.

[66] R. Sokullu and O. Karaca, "Fault management for smart wireless sensor networks," in *9th International Conference on Ubiquitous Intelligence and Computing, Autonomic and Trusted Computing (UIC/ATC)*, pp. 382–387, 2012.

[67] Y. Ren, V. Oleshchuk, and F. Li, "A scheme for secure and reliable distributed data storage in unattended wsns," in *Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM)*, pp. 1–6, 2010.

[68] S. Aly, K. Zhenning, and E. Soljanin, "Fountain codes based distributed storage algorithms for large-scale wireless sensor networks," in *Proceedings of International Conference on Information Processing in Sensor Networks (IPSN)*, pp. 171–182, 2008.

[69] C. Jardak, E. Osipov, and P. Mahonen, "Distributed information storage and collection for wsns," in *Proceedings of the IEEE Internatonal Conference on Mobile Adhoc and Sensor Systems (MASS)*, pp. 1–10, 2007.

[70] A. Fraboulet, G. Chelius, and E. Fleury, "Worldsens : Development and prototyping tools for application specific wireless sensors networks," in *International Symposium on Information Processing in Sensor Networks (IPSN)*, pp. 176–185, 2007.

[71] F. Osterlind, A. Dunkels, J. Eriksson, N. Finne, and T. Voigt, "Cross-level sensor network simulation with cooja," in *Proceedings of the IEEE Conference on Local Computer Networks*, pp. 641–648, 2006.

[72] A. Dunkels, B. Gronvall, and T. Voigt, "Contiki - a lightweight and flexible operating system for tiny networked sensors," in *Proceedings of the 29th Annual IEEE International Conference on Local Computer Networks*, pp. 455–462, 2004.

[73] C. B. Margi, B. T. de Oliveira, G. T. de Sousa, M. A. S. Jr, P. S. L. M. Barreto, T. C. M. B. Carvalho, M. Näslund, and R. Gold, "Impact of operating systems on wireless sensor networks (security) applications and testbeds," in *International Conference on Computer Communications and Networks (ICCCN)*, pp. 1–6, 2010.

[74] T. Watteyne, D. B. Kris Pister, M. Dohler, and I. Auge-Blum, "Implementation of Gradient Routing in Wireless Sensor Networks," in *Proceedings of the IEEE Global Telecommunications Conference (Globecom)*, pp. 1–6, 2009.

[75] W. K. Sheah, Z. A. Eu, and H.-P. Tan, "Wireless sensor networks powered by ambient energy harvesting (wsn-heap) - survey and challenges," in *Proceedings of the 1st International Conference on Wireless Communication, Vehicular Technology, Information Theory and Aerospace and Electronic Systems Technology (Wireless VITAE)*, pp. 1–5, 2009.

[76] J. H. Abawajy, S. Nahavandi, and F. Al-Neyadi, "Sensor node activity scheduling approach," in *Proceedings of the IEEE International Conference on Multimedia and Ubiquitous Engineering (MUE)*, pp. 72–77, 2007.

[77] P. Merindol and A. Gallais, "Path diversity in energy-efficient wireless sensor networks," in *Proceedings of the IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, pp. 2280–2284, 2009.

[78] G. Galbiati, F. Maffiolib, and A. Morzenti, "A short note on the approximability of the maximum leaves spanning tree problem," *Information Processing Letters*, vol. 52, no. 1, pp. 45–49, 1994.

[79] C. Intanagonwiwat, R. Govindan, D. Estrin, J. Heidemann, and F. Silva, "Directed diffusion for wireless sensor networking," *IEEE/ACM Transactions on Networking (ToN)*, vol. 11, no. 1, pp. 2–16, 2003.

[80] F. Khadar and T. Razafindralambo, "Performance evaluation of gradient routing strategies for wireless sensor networks," in *IFIP Networking*, vol. 5550, pp. 535–547, 2009.

[81] T. Winter, P. Thubert, A. Brandt, J. Hui, R. Kelsey, P. Levis, K. Pister, R. Struik, J. Vasseur, and R. Alexander, "Rpl : Ipv6 routing protocol for low-power and lossy networks." Internet Engineering Task Force Request for Comments (RFC) 6550, 2012.

[82] S. Guha and S. Khuller, "Approximation algorithms for connected dominating sets," *Algorithmica*, vol. 20, no. 4, pp. 374–387, 1998.

[83] J. Beaudaux, A. Gallais, and T. Razafindralambo, "Multiple coverage with controlled connectivity in wireless sensor networks," in *Proceedings of the 7th ACM International Symposium on Performance Evaluation of Wireless Ad Hoc, Sensor, and Ubiquitous Networks (PE-WASUN)*, pp. 9–16, 2010.

[84] G. Halkes and K. Langendoen, "Practical considerations for wireless sensor network algorithms," *Wireless Sensor Network*, vol. 2, pp. 441–446, 2010.

[85] A. Gallais and J. Carle, "An adaptive localized algorithm for multiple sensor area coverage," in *Proceedings of the 21st IEEE International Conference on Advanced Information Networking and Applications (AINA)*, pp. 525–532, 2007.

[86] R. Kuntz, A. Gallais, and T. Noel, "Medium access control facing the reality of wsn deployments," *ACM Computer Communication Review (CCR)*, vol. 39, no. 3, 2009.

[87] J. Beaudaux, A. Gallais, J. Montavont, and T. Noel, "Lift : Layer independent fault tolerance mechanism for wireless sensor networks," in *Proceedings of the IEEE Vehicular Technology Conference (VTC)*, pp. 1–5, 2011.

[88] I. Demirkol, F. Alagoz, H. Delic, and C. Ersoy, "Wireless sensor networks for intrusion detection : packet traffic modeling," *IEEE Communications Letters*, vol. 10, no. 1, pp. 22 – 24, 2006.

[89] J. Beaudaux, A. Gallais, and T. Noel, "Heterogeneous mac duty-cycling for energy-efficient internet of things deployments," *Springer Networking Science Journal*, pp. 1–9, 2013.

[90] S. K. Dash, S. Mohapatra, and P. K. Pattnaik, "A survey on applications of wireless sensor network using cloud computing," *International Journal of Computer Science and Emerging Technologies*, vol. 1, no. 4, pp. 50–55, 2010.

[91] K. Ahmed and M. Gregory, "Integrating wireless sensor networks with cloud computing," in *Proceedings of the 7th International Conference on Mobile Ad-hoc and Sensor Networks (MSN)*, pp. 364–366, 2011.

[92] R. Liu and I. J. Wassell, "Opportunities and challenges of wireless sensor networks using cloud services," in *Proceedings of the workshop on Internet of Things and Service Platforms (IoTSP)*, pp. 1–7, 2011.

[93] Z. Wilcox-O'Hearn and B. Warner, "Tahoe : the least-authority filesystem," in *Proceedings of the 4th ACM international workshop on Storage security and survivability (StorageSS)*, pp. 21–26, 2008.

[94] D. Slamanig and C. Hanser, "On cloud storage and the cloud of clouds approach," in *Proceedings of the International Conferece for Internet Technology And Secured Transactions*, pp. 649–655, 2012.

[95] A. Lounis, A. Hadjidj, A. Bouabdallah, and Y. Challal, "Secure and scalable cloud-based architecture for e-health wireless sensor networks," in *Proceedings of the 21st International Conference on Computer Communications and Networks (ICCCN)*, pp. 1–7, 2012.

[96] X. H. Le, S. Lee, P. T. H. Truc, L. T. Vinh, A. Khattak, M. Han, D. V. Hung, M. Hassan, M. Kim, K.-H. Koo, Y.-K. Lee, and E.-N. Huh, "Secured wsn-integrated cloud computing for u-life care," in *Proceedings of the 7th IEEE Consumer Communications and Networking Conference (CCNC)*, pp. 1–2, 2010.