



Auto-organisation des réseaux sans-fil multi-sauts dans les villes intelligentes

Tony Ducrocq

► To cite this version:

Tony Ducrocq. Auto-organisation des réseaux sans-fil multi-sauts dans les villes intelligentes. Réseaux et télécommunications [cs.NI]. Université de Lille 1, 2013. Français. NNT : 41202 . tel-00915269

HAL Id: tel-00915269

<https://theses.hal.science/tel-00915269>

Submitted on 6 Dec 2013

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

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

Auto-organisation des réseaux sans-fil multi-sauts dans les villes intelligentes

THÈSE

présentée et soutenue publiquement le 15 novembre 2013

en vue de l'obtention du

Doctorat de l'Université des Sciences et Technologies de Lille
(spécialité informatique)

par

Tony DUCROCQ

Composition du jury

Rapporteurs :

Eric FLEURY

Professeur, École Normale Supérieure de Lyon

Aline CARNEIRO VIANA

Chargée de recherche, Inria Saclay - Île-de-France

Examineurs :

Gilles GRIMAUD

Professeur, Université de Lille 1

Paul COUDERC

Chargé de recherche, Inria Rennes - Bretagne Atlantique

Directeur de thèse :

Nathalie MITTON

Chargée de recherche, Inria Lille - Nord Europe

Co-Encadrant de thèse :

Michaël HAUSPIE

Maître de conférence, Université de Lille 1

Je dédie cette thèse à ma femme et à mes parents.

Résumé

Les villes du futur et actuelles sont de plus en plus connectées. Les relèves de compteur électrique, de gaz et d'eau se font de plus en plus à distance et sans-fil. Les luminaires des villes deviennent intelligents et communiquent pour économiser l'énergie et offrir un éclairage adapté. Les engins de ramassage des ordures communiqueront bientôt avec les poubelles afin d'anticiper les tournées de ramassage et de mieux connaître le contenu des containers.

Pour des questions de coûts et de rapidité de déploiement, ces réseaux sont souvent sans infrastructure. Par ailleurs, pour les mêmes raisons, chaque nœud du réseau puise son énergie dans une batterie donc la capacité est limitée. Il convient donc pour ces réseaux particuliers, déployés dans un environnement contraint, de trouver des solutions d'organisation et de communication adaptées.

À travers cette thèse je propose d'étudier la problématique des réseaux sans-fil multi-sauts dans les villes intelligentes. J'étudie dans un premier temps l'importance et l'impact de la topologie sur les performances réseau. Plus précisément, au travers de simulations et d'études expérimentales, je démontre que le placement des nœuds impacte les performances des algorithmes et que, par conséquent, il est nécessaire d'étudier les algorithmes pour réseaux de capteurs sans-fil sur des topologies qui correspondent à l'application ciblée.

Je propose ensuite une famille d'algorithmes de *clustering* pour réseaux de capteurs sans-fil permettant d'optimiser la durée de vie du réseau. L'hypothèse clé de ces algorithmes est qu'un chef de *cluster* (*cluster-head*) consomme plus d'énergie qu'un nœud sans rôle particulier. En effet, le *cluster-head* se voit souvent attribuer des tâches supplémentaires telles que l'agrégation de données et la transmission à un nœud puits, plus lointain donc plus coûteux à joindre. Cette famille d'algorithmes nommée **BLAC** permet de créer des clusters multi-sauts dans lesquels chaque tête de *cluster* est la racine d'un arbre qui compose le cluster. Dans chaque cluster, les données sont acheminées vers le *cluster-head* grâce à un *gradient routing*. L'idée principale de ces algorithmes est que le rôle de *cluster-head* doit être attribué en fonction du niveau d'énergie des nœuds et de leur voisinage. Ces algorithmes ont été testés grâce à des simulations sur des topologies de villes réalistes avec des paramètres de simulation tirés du monde réel.

Enfin, je propose un algorithme de routage pour des réseaux de villes intelligentes à large échelle. Cet algorithme repose sur l'hypothèse que seulement certains nœuds du réseau ont une connaissance de leur position. Cette position peut être acquise parce que ces nœuds sont équipés d'une puce de localisation (telle que le GPS) ou parce qu'ils sont fixes et configurés au moment du déploiement avec leur position pré-programmée. Cet algorithme repose sur la combinaison de deux techniques de routage, classique et géographique : **AODV** et **Greedy** geographic routing. Je montre qu'il est possible d'obtenir des performances proches des algorithmes de routage géographique alors même qu'un sous ensemble des nœuds du réseau n'a pas connaissance de sa position.

Self-organisation of multi-hop wireless networks in smart cities

Abstract

In recent years, cities have become more and more connected. Readings of the electricity usage are being performed wirelessly. In order to reduce the energy consumption, lampposts became intelligent, while the garbage trucks communicate with dust bins in order to plan the garbage collection tours. These networks often lack the infrastructure and sensor nodes rely on a battery with a limited capacity. In this thesis, I analyze the issues of multi-hop wireless networks in smart cities.

First, I study the significance and impact of network topologies on network performance. More precisely, through simulations and an experimental study, I show that the placement of nodes impacts the performance of geographic routing algorithms.

Then, I propose a set of *clustering* algorithms for wireless sensor networks that optimize the lifetime of the network. The key hypothesis is that a *cluster-head* is consuming more energy than a regular node. This set of algorithms, named BLAC, creates multi-hop *clusters* in which each *cluster-head* is the root of a tree. The main idea is that the role of each *cluster-head* should be assigned regarding the remaining energy of nodes and their neighborhood. These algorithms have been tested through simulations based on realistic city topologies with simulation parameters resembling the real world.

In the end, I propose a routing algorithm for large scale smart cities that combines two geographic routing techniques. It relies on the hypothesis that only a fraction of nodes in the network are aware of their positions. I show that it is possible to achieve the performance close to classical routing algorithms, even under this hypothesis.

Table des matières

1	Introduction	15
1.1	Les réseaux	15
1.1.1	Le modèle en couches	15
1.1.2	Les réseaux sans-fil	17
1.2	Les réseaux de capteurs sans-fil	17
1.2.1	Apparition et évolution	17
1.2.2	Applications	19
1.2.3	Problématiques et solutions	19
1.3	Les villes intelligentes	21
1.4	Le projet <i>BinThatThinks</i>	21
1.5	Notations	24
1.6	Contribution et organisation du document	25
2	État de l’art	27
2.1	Les villes intelligentes	27
2.2	Les solutions pour la gestion des déchets	30
2.2.1	La RFID	30
2.2.2	Les autres technologies	32
2.3	Plates-formes de simulation et d’expérimentation	33
3	Impact de la topologie sur les performances	35
3.1	Introduction	35
3.2	État de l’art	36
3.3	Motivations	38
3.4	Algorithmes	39
3.5	Topologies	40
3.6	Simulations	42
3.6.1	Taux de livraison	43
3.6.2	Longueur des routes	46
3.6.3	Coûts d’émission	49
3.7	Expérimentations	52
3.8	Discussion	57
3.9	Conclusion	59
4	Clustering pour l’optimisation de la durée de vie	61
4.1	Introduction	61
4.2	Motivations et énoncé du problème	63
4.3	Notations et préliminaires	63
4.3.1	Métrique de densité	64

4.3.2	Réduction de topologie	64
4.4	État de l'art	66
4.5	Contribution	70
4.5.1	Algorithmes BLAC-b*	71
4.5.2	BLAC-r*	73
4.6	Résultats	74
4.6.1	Paramètres de simulation	74
4.6.2	Durée de vie du réseau	76
4.6.3	Taux de livraison	77
4.6.4	Propriétés des réseaux	78
4.7	Conclusion	81
5	Routage géographique	83
5.1	Introduction	83
5.2	État de l'art	84
5.3	Proposition	87
5.4	Simulations et résultats	90
5.4.1	Taux de livraison	92
5.4.2	Nombre de messages de contrôle	96
5.4.3	Taille de la table de routage	98
5.5	Auto-configuration du paramètre k	100
5.6	Conclusion	101
6	Conclusion et perspectives	103
6.1	Conclusion	103
6.2	Perspectives	104
6.3	Conclusion personnelle	105

Table des figures

1.1	Modèle OSI.	16
1.2	Un réseau de capteurs sans-fil.	18
1.3	Un tag RFID ©Maschinenjunge	23
1.4	Mutualisation des ressources GPRS dans BTT.	24
3.1	Comparaison des variantes de l'algorithme de routage géographique.	40
3.2	Exemple de topologie aléatoire.	40
3.3	Topologie <i>city</i> avec un exemple de route.	41
3.4	Topologie <i>city grid</i>	42
3.5	Taux de livraison pour <i>Greedy</i>	44
3.6	Taux de livraison pour <i>NFP</i>	44
3.7	Taux de livraison pour <i>MFR</i>	45
3.8	Taux de livraison pour <i>Compass</i>	45
3.9	Longueur moyenne des routes pour <i>Greedy</i>	46
3.10	Longueur moyenne des routes pour <i>NFP</i>	47
3.11	Longueur moyenne des routes pour <i>MFR</i>	47
3.12	Longueur moyenne des routes pour <i>Compass</i>	48
3.13	Coûts d'émission pour <i>Greedy</i>	49
3.14	Coûts d'émission pour <i>NFP</i> . (légende identique à la figure 3.13)	50
3.15	Coûts d'émission pour <i>MFR</i>	50
3.16	Coûts d'émission pour <i>Compass</i>	51
3.17	Position des nœuds sur la plate-forme de Rennes.	53
3.18	Topologie city sur la plate-forme de Rennes.	53
3.19	Taux de livraison pour <i>greedy</i> sur SensLAB	54
3.20	Taux de livraison pour <i>NFP</i> sur SensLAB	55
3.21	Taux de livraison pour <i>MFR</i> sur SensLAB	55
3.22	Taux de livraison pour <i>COMPASS</i> sur SensLAB	56
3.23	Comparaison de la longueur des routes pour trois topologies.	57
3.24	Comparaison des coûts énergétiques pour trois topologies.	58
4.1	Création des <i>clusters</i>	65
4.2	Construction du RNG	66
4.3	Épuisement de la batterie : cas du <i>clustering</i> densité.	67
4.4	Épuisement de la batterie : cas de BLAC-b*	68
4.5	Épuisement de la batterie : cas de BLAC-r*	69
4.6	Topologies réseau, chaque point rouge représente un nœud.	74
4.7	Durée de vie du réseau	76
4.8	Taux de livraison	77
4.9	Durée de vie du réseau	78

4.10	Stabilité du réseau	79
4.11	Excentricité moyenne	80
5.1	Exemple de coordonnées dans LTP.	86
5.2	Exemple de recherche de route du nœuds S vers le nœud D dans AODV.	87
5.3	Exemple de routage dans HGA.	91
5.4	Taux de livraison des différents algorithmes de routage avec 1% des nœuds ayant leur information de position pour HGA et Greedy.	93
5.5	Taux de livraison des différents algorithmes de routage avec 5% des nœuds ayant leur information de position pour HGA et Greedy.	93
5.6	Taux de livraison des différents algorithmes de routage avec 10% des nœuds ayant leur information de position pour HGA et Greedy.	94
5.7	Nombre de messages de contrôle des différents algorithmes de routage avec 1% des nœuds ayant leur information de position pour HGA.	96
5.8	Nombre de messages de contrôle des différents algorithmes de routage avec 5% des nœuds ayant leur information de position pour HGA.	97
5.9	Nombre de messages de contrôle des différents algorithmes de routage avec 10% des nœuds ayant leur information de position pour HGA.	97
5.10	Taille moyenne de la table de routage (pour HGA) ou de voisinage (pour greedy et Vcap) pour les différents algorithmes de routage avec 1% des nœuds ayant leur information de position pour HGA.	98
5.11	Taille moyenne de la table de routage ou de voisinage pour les différents algorithmes de routage avec 5% des nœuds ayant leur information de position pour HGA.	99
5.12	Taille moyenne de la table de routage ou de voisinage pour les différents algorithmes de routage avec 10% des nœuds ayant leur information de position pour HGA.	99

Liste des tableaux

3.1	Paramètres de simulation pour la mesure de l'impact de la topologie	43
4.1	Paramètres de simulation des algorithmes de <i>clustering</i>	75
5.1	Paramètres de simulation des algorithmes de routage géographique .	92

Introduction

Sommaire

1.1	Les réseaux	15
1.2	Les réseaux de capteurs sans-fil	17
1.3	Les villes intelligentes	21
1.4	Le projet <i>BinThatThinks</i>	21
1.5	Notations	24
1.6	Contribution et organisation du document	25

Grâce à l'évolution des technologies, à la miniaturisation des composants et la réduction des coûts, on a vu apparaître de nouveaux types de réseaux. Autonomes, avec des capacités d'auto organisation, les réseaux de capteurs sans-fil permettent de nouvelles applications dans tous les domaines de notre société moderne. Dans les villes de plus en plus peuplées, il est nécessaire de trouver des solutions, non seulement innovantes, mais surtout efficaces pour résoudre les différents problèmes qui s'accroissent avec l'augmentation de la population. Gestion du trafic, économies d'énergie, écologie et gestion des déchets sont quelques-uns des défis de demain que pourront aider à relever ces nouveaux réseaux que sont les réseaux de capteurs sans-fil.

Cette thèse est financée par le projet *ANR BinThatThinks*, dont le but est de proposer des solutions intelligentes pour la collecte des déchets. Les contributions de cette thèse permettent d'apporter des solutions aux problématiques du projet. Le projet *BinThatThinks* est décrit plus en détails dans la section 1.4.

1.1 Les réseaux

Pour comprendre au mieux les principes utilisés dans cette thèse, il est nécessaire d'introduire quelques notions autour des réseaux et des communications sans-fil.

1.1.1 Le modèle en couches

Dans les réseaux informatiques (filaires et sans-fil), un modèle en couches est utilisé. Chaque couche répond à un problème ou à une famille de problèmes des transmissions de données. Une couche s'appuie sur les services des couches inférieures du modèle pour fournir un service aux couches supérieures. Lorsqu'un message est envoyé de a vers b , il traverse les différentes couches de la machine a , transite

par le milieu physique (courants électriques, ondes électromagnétiques, ondes lumineuses...) et remonte les différentes couches de la machine *b*. Les couches hautes sont proches de l'utilisateur tandis que les couches basses se rapprochent du monde physique.

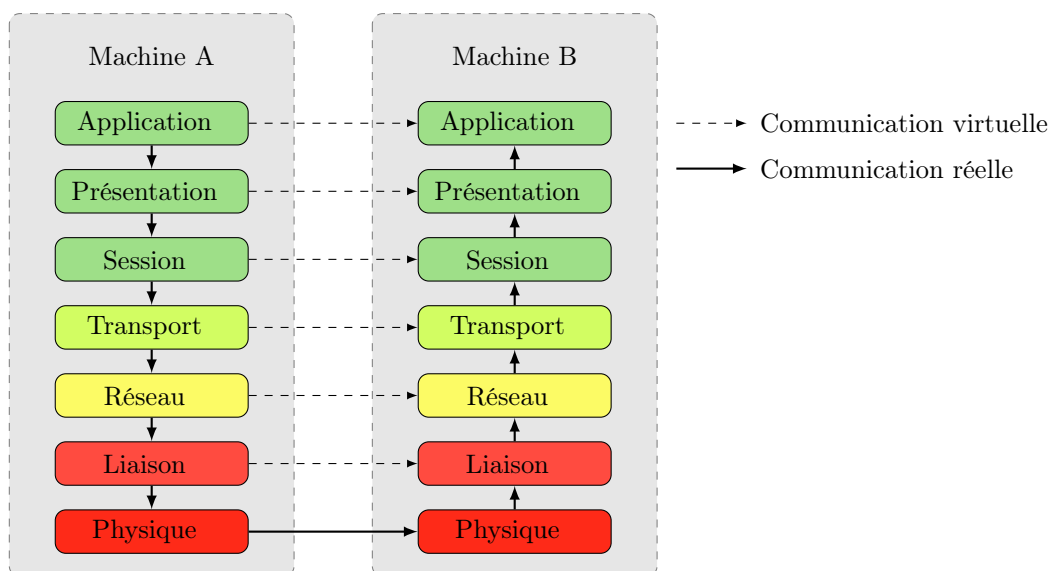


FIGURE 1.1 – Communications entre deux applications sur deux machines différentes d'un réseau en utilisant le modèle OSI.

Dans cette thèse, nous nous intéressons essentiellement à la couche réseau, et plus particulièrement à sa composante routage. Le routage, dans un réseau complexe, permet de déterminer le chemin entre deux machines. C'est une tâche essentielle qui permet à toute machine du réseau d'envoyer des messages à n'importe quelle autre machine. La tâche du routage dans un réseau incombe au protocole de routage qui se repose sur un ou des algorithmes¹ de routages.

La couche réseau s'appuie sur les services de la couche liaison de données dont la sous-couche MAC (Medium Access Control) est une composante. Dans la suite de cette thèse, nous parlerons de couche MAC. Sa principale tâche est de réguler l'accès au support de communication. Dans les réseaux sans-fil, qui partagent le même support, son rôle est capital. Lorsque deux machines émettent des données au même moment, la communication est brouillée et aucune des deux émissions n'est exploitable, on parle de collision de paquets. Le rôle de la couche MAC est de gérer et limiter, voire de supprimer ces collisions.

1. Wikipédia : *un algorithme est une suite finie et non ambiguë d'opérations ou d'instructions permettant de résoudre un problème.*

1.1.2 Les réseaux sans-fil

À la différence des communications câblées, les communications sans-fil utilisent toutes le même milieu de propagation. Le partage de cette ressource crée des collisions, ce qui nécessite une coopération entre les différentes machines du réseau. On rencontre aussi des problèmes d'interférences, beaucoup plus présents que dans les réseaux filaires. Ces interférences sont essentiellement gérées par la couche physique, dont nous parlerons peu dans cette thèse. Ces problèmes spécifiques aux réseaux sans-fil nécessitent des solutions adaptées.

Pour pallier les éventuelles pertes de messages, qui peuvent être causées par les interférences et les collisions, on utilise les accusés de réception. Appelés aussi ACK (de l'anglais *acknowledgement*), ils permettent à l'émetteur de savoir qu'un message est bien arrivé au destinataire. Après un délai d'attente prédéfini (*timeout*), si l'émetteur n'a pas reçu d'accusé de réception, il en déduit que le message est perdu. Il peut alors décider de ré-émettre le message ou de signaler le message comme perdu.

Afin de connaître leur voisinage, les stations sans-fil utilisent des messages particuliers, appelés *beacon* ou messages HELLO. En embarquant l'identifiant de leur émetteur, les messages HELLO permettent aux stations les recevant de connaître l'existence et l'identité de l'émetteur. Les stations recevant des messages HELLO peuvent maintenir une table de voisinage contenant l'identité de tout ou partie de leurs voisins.

1.2 Les réseaux de capteurs sans-fil

Les réseaux de capteurs sans-fil ont vu le jour en combinant des capacités de détection (*sensing* en anglais) et de transmission sans-fil. Leur essor a été possible lorsque les évolutions de la technologie ont permis leur miniaturisation et la réduction des coûts de fabrication.

On peut définir un réseau de capteurs sans-fil comme un ensemble d'appareils capables de mesurer des informations sur leur environnement et de communiquer grâce à une liaison radio. L'ensemble des capteurs, ou nœuds, définit la zone surveillée par le réseau. Les différents capteurs du réseau achemineront, la plupart du temps, leurs informations à un nœud spécial dans le réseau appelé *station de base* ou *puits*. La figure 1.2 montre un exemple de réseau de capteurs sans-fil envoyant une information à une station de base.

1.2.1 Apparition et évolution

Les premières communications sans-fil [1] sont apparues à la fin du XIX^e siècle, d'abord avec les travaux de Nikola Tesla [2] puis avec l'essor des Transmissions Sans Fil (TSF). La démocratisation et la diversification des communications sans-fil ont eu lieu dans les années 90 avec l'apparition de nouvelles technologies dans la téléphonie mobile, notamment le GSM (2G) [3]. Dans la même période, et au début

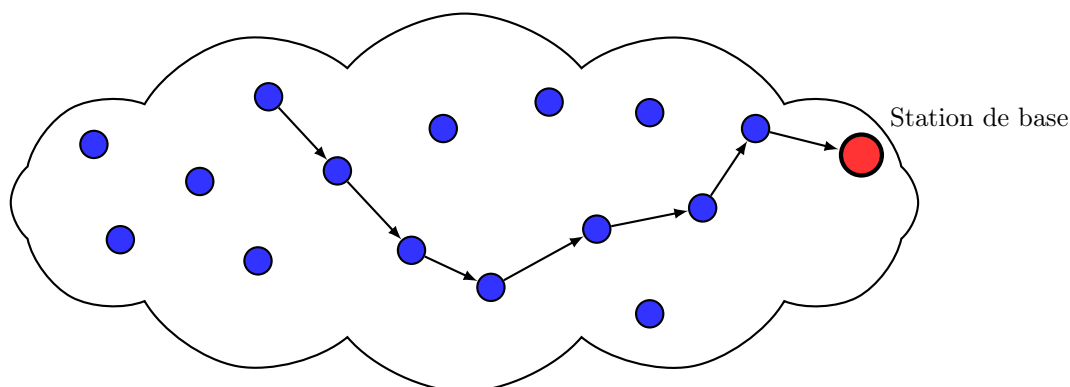


FIGURE 1.2 – Schéma d'un réseau de capteurs sans-fil acheminant une information à une station de base.

des années 2000, sont apparues des technologies à destination des réseaux privés grand public comme le Wi-Fi, le Bluetooth[®], etc.

Ces nouvelles technologies sans-fil autorisent les connexions *ad-hoc* [4] permettant à chaque nœud de se connecter directement aux nœuds à portée de communication. Grâce aux connexions *ad-hoc*, en formant des réseaux maillés, il est possible de créer des réseaux dans lesquels tous les nœuds ont un rôle équivalent. Chaque nœud peut émettre, recevoir et relayer des informations. Ce type de réseau est plus robuste et tolérant aux pannes que les réseaux centralisés, sensibles à la seule panne du nœud central.

Les techniques utilisées pour les réseaux *ad-hoc* combinées à la démocratisation et la miniaturisation des composants électroniques ont permis l'apparition des réseaux de capteurs sans-fil [5]. Chaque nœud du réseau associe une ou plusieurs capacités de détection de l'environnement (les capteurs) avec une unité de traitement de l'information (le processeur ou micro-contrôleur) et un module de communication sans-fil. Ils sont associés le plus souvent à une batterie leur assurant une réserve d'énergie afin de fonctionner de manière autonome.

Leur taille réduite, leur faible coût et la contrainte de fonctionner longtemps sur batterie implique un certain nombre de compromis quant aux performances des capteurs sans-fil. En effet la mémoire embarquée sur ces petits objets est très faible, de l'ordre de quelques kilo-octets. Le processeur, souvent 8 ou 16 bits, fonctionne à une fréquence de quelques dizaines de méga-hertz. À titre de comparaison les smartphones modernes embarquent plus d'un giga-octet de mémoire et possèdent un processeur 32 voire 64 bits fonctionnant à plus d'un giga-hertz. On peut ainsi dire qu'un smartphone est plus de 1000 fois plus puissant qu'un capteur sans-fil. Ces contraintes fortes nécessitent des algorithmes et des protocoles adaptés.

1.2.2 Applications

Les réseaux *ad-hoc* ne nécessitant pas d'infrastructure, ils sont particulièrement adaptés aux applications militaires [5]. Les autres propriétés de ces réseaux : déploiement rapide, auto-organisation et tolérance aux pannes leur permettent de répondre aux besoins de ces conditions spécifiques.

Les réseaux de capteurs sans-fil sont quant à eux particulièrement adaptés à la surveillance de zones critiques grâce notamment à leur autonomie et à leur petite taille. Parmi les applications possibles on notera la détection des feux de forêt [6], la prévention des inondations [7], la surveillance d'installations nucléaires [8], etc.

Ce type de réseau ouvre de nouvelles opportunités pour les villes de demain. La congestion du trafic routier, la gestion de l'énergie et des déchets sont les défis des grandes villes d'aujourd'hui et de demain. Les réseaux de capteurs sans-fil apportent des éléments de solution à ces problématiques. On trouve des solutions pour l'observation et la gestion du trafic avec notamment la gestion des feux tricolores [9], la régulation des lampadaires publics [10] ou encore la surveillance et la gestion de la collecte des déchets centralisées [11].

Dans notre quotidien, les applications de domotique et les bâtiments intelligents suscitent un engouement particulier. Les réseaux de capteurs sans-fil dans ces contextes apportent des solutions innovantes pour le confort, les économies d'énergie et la sécurité [12]. Pour ces applications, les technologies de l'alliance ZigBee® offrent une pile protocolaire sans-fil complète pour différents types d'applications.

Enfin les réseaux de capteurs sans-fil trouvent des applications pour la surveillance de mesures physiologiques. On peut par exemple surveiller les fonctions vitales des personnes âgées ou ayant des problèmes de santé afin de donner l'alerte par l'intermédiaire d'un réseau de capteurs sans-fil [13]. De même on peut analyser différentes mesures physiques recueillies sur un sportif, les possibilités offertes par le réseau de capteurs sans-fil permettant d'alléger le dispositif de collecte des données [14].

1.2.3 Problématiques et solutions

Bien qu'offrant des avantages certains, les réseaux de capteurs sans-fil apportent de nouvelles problématiques. En effet, les propriétés différentes de ce type de réseau impliquent une mise en œuvre adaptée.

La première difficulté que l'on peut mettre en avant dans les réseaux de capteurs sans-fil, en comparaison de leurs homologues filaires, est la faible fiabilité des liaisons radio. En effet, les transmissions sans-fil sont impactées directement par la distance entre l'émetteur et le récepteur, par les perturbations électromagnétiques et par le partage du médium radio. Zhou et al. [15] étudient l'impact des irrégularités des liaisons radio et proposent des solutions pour en limiter les effets, notamment sur les protocoles de routage.

La seconde contrainte, qui découle de leur déploiement autonome et sans infrastructure est celle de l'énergie. Bien que les technologies évoluent et offrent des bat-

teries de plus en plus endurantes (technologies Lithium Ion, Lithium Polymère...), des micro-contrôleurs et des circuits radio de plus en plus efficaces, les protocoles pour réseaux de capteurs doivent toujours prêter une attention toute particulière à la consommation énergétique. Ingelrest et al. [16] proposent un mécanisme pour adapter la fréquence des messages HELLO en fonction de la mobilité des nœuds, permettant ainsi un bon compromis entre consommation énergétique et exactitude de la table de voisinage. Les protocoles MAC qui assurent un partage du médium radio équitable entre les différents nœuds jouent aussi un rôle essentiel dans l'économie d'énergie. Le protocole X-MAC [17] par exemple, propose un mécanisme autorisant les nœuds en veille à couper complètement leur module radio. En effet, pour les applications dans lesquelles le nombre de messages échangés est faible, une grande partie de l'énergie est perdue par l'écoute du médium radio. Dans X-MAC les nœuds qui doivent émettre une donnée envoient des préambules courts contenant l'identifiant du nœud destinataire. Après chaque préambule l'émetteur écoute le médium dans l'attente de recevoir un accusé de réception (ACK), dans le cas contraire il ré-émet un nouveau préambule. Lorsqu'un ACK est reçu, cela signifie que le nœud destinataire est en écoute, l'émetteur peut alors envoyer les données.

Par ailleurs, les faibles ressources embarquées sur les capteurs sans-fil ne permettent pas l'utilisation de protocoles dont la complexité est élevée ou les besoins en mémoire trop grands.

Les réseaux de capteurs sans-fil reposent rarement sur une infrastructure existante, ils sont aussi souvent totalement décentralisés. Les nœuds du réseau doivent alors faire preuve d'auto-organisation, au contraire des réseaux filaires classiques, tous les nœuds du réseau sont équivalents. Des techniques de *clustering* proposent une approche décentralisée à l'auto-organisation des réseaux de capteurs. Dans DDR [18] une forêt d'arbres est construite, chaque arbre devient une zone (ou un *cluster*) qui s'apparente alors à un élément du réseau pour le routage des données. L'algorithme LEACH [19] propose une approche de *clustering* économe en énergie en équilibrant la charge des nœuds désignés comme chefs (les *cluster-heads*). Les nœuds *cluster-heads* se voyant assigner des tâches supplémentaires ils épuisent leur batterie plus rapidement que les autres nœuds, il est donc important que ce rôle soit attribué à tour de rôle aux différents nœuds. Ces aspects seront abordés plus en détails dans le chapitre 4.

Pour le routage des données, des algorithmes tels que AODV [20] ou OLSR [21] sont des solutions qui s'adaptent bien aux réseaux de capteurs sans-fil, bien qu'elles ne soient pas développées spécifiquement pour ceux-ci. Les solutions de routage géographique apportent aussi une réponse efficace lorsque les nœuds disposent de leur position. En effet, ces algorithmes ne nécessitent qu'une connaissance locale par chacun des nœuds pour router les informations [22]. Chaque nœud ayant une donnée à transmettre l'envoie à son voisin se rapprochant le plus de la destination à atteindre. Bose et al. ont proposé un algorithme de routage géographique avec garantie de livraison dans les réseaux connexes [23]. Ce type d'algorithme permet, dans les réseaux denses, d'assurer la livraison des données tout en bénéficiant des avantages du

routage géographique comme l'efficacité énergétique. Kalosha et al. [24] proposent un algorithme de routage géographique sans message HELLO et avec garantie de livraison. Cela permet de diminuer encore le nombre de messages, entraînant une diminution des coûts énergétiques et des délais ainsi qu'une augmentation des débits tout en conservant les propriétés précédemment citées.

Les réseaux de capteurs sans-fil offrent une technologie prometteuse, pour laquelle de nombreuses problématiques se sont vues offrir des solutions pertinentes. Malgré tout, il reste de nombreux problèmes non résolus ou pour lesquels les solutions existantes ne sont pas satisfaisantes. Nous les aborderons en partie au chapitre 2.

1.3 Les villes intelligentes

La notion de ville intelligente ou *smart city* est émergente et les définitions peuvent varier. La définition a évolué par le passé et continuera probablement à évoluer dans le futur. Dans le passé, l'infrastructure de communication était la notion principale, sinon la seule, de la définition d'une ville intelligente [25]. Aujourd'hui cette définition simple n'est plus suffisante et il semble difficile de définir précisément une ville intelligente. Les définitions modernes tiennent compte des infrastructures mais aussi de l'humain, du social, de l'environnement, du transport, de l'économie ou encore de l'administration. Les villes intelligentes sont en quelque sorte une extension des bâtiments intelligents qui sont eux-mêmes une extension des maisons intelligentes.

Les villes intelligentes offrent un vivier de possibilités pour améliorer le quotidien de leurs habitants. Les nouvelles applications qui émergent dans ces villes nécessitent des technologies adaptées. Les réseaux de capteurs sans-fil sont parmi les technologies les plus prometteuses pour répondre aux besoins des villes intelligentes. Toutefois seule une mise en œuvre efficace et transparente pour ses utilisateurs permettra une large adoption de ces technologies et donc une amélioration de la qualité de vie des citoyens. Nous décrivons au chapitre suivant, en section 2.1, l'état de l'art des travaux sur les villes intelligentes.

1.4 Le projet *BinThatThinks*

Le projet *BinThatThinks*, qui finance cette thèse, se place dans le contexte des villes intelligentes. *BinThatThinks* est un projet financé par l'Agence Nationale pour la Recherche (ANR) dans l'appel Ecotech 2010. Il a démarré en décembre 2010 pour une durée de 3 ans. Les différents partenaires du projet sont :

- **Inria**² avec les équipes ACES³ à Rennes et FUN⁴ à Lille,

2. <http://www.inria.fr>

3. <http://www.irisa.fr/aces>

4. <https://team.inria.fr/fun>

- Véolia propreté⁵ à Paris,
- Etineo⁶ à Lille.

L'un des enjeux des villes intelligentes est une gestion des déchets, du recyclage et de l'énergie sage et intelligente. Le but du projet *BinThatThinks* est de faciliter la collecte et le recyclage des déchets, d'améliorer la qualité du tri pour en réduire les coûts et réduire l'impact sur l'environnement. Pour réduire la distance parcourue par le camion de ramassage, limiter les manipulations effectuées par les opérateurs, détecter la proximité de produits incompatibles ou facturer les clients pour les déchets qu'ils jettent réellement, *BinThatThinks* ajoute de l'intelligence aux poubelles. Le projet se propose d'agir sur un certain nombre de facteurs permettant d'atteindre ces objectifs :

- améliorer la qualité des déchets retournés en améliorant la qualité du tri pour diminuer les refus en centre de tri,
- séparer les déchets au niveau du point de collecte grâce à une reconnaissance des déchets,
- planifier la collecte pour optimiser les trajets, réduire les coûts et l'empreinte carbone,
- éduquer de façon ciblée les usagers grâce aux informations obtenues sur les points de collecte et les containers.

Les points de collectes, qu'ils soient publics ou au sein même du foyer, permettent d'améliorer la qualité du tri en mettant en œuvre des aides pour l'utilisateur. Ils permettent aussi de quantifier les différents déchets collectés afin de mettre en place des redevances incitatives.

Le scénario principal du projet consiste à identifier l'ensemble des déchets recyclables ou non à l'aide de tags RFID. La technologie RFID⁷ est un élément essentiel du projet *BinThatThinks*. Une étiquette RFID est une petite puce électronique associée à une antenne qui contient des informations numériques. La particularité de ces étiquettes est qu'elles ne nécessitent pas de batterie, elles transforment les ondes émises par les lecteurs en énergie pour émettre à leur tour leur identifiant et d'éventuelles informations complémentaires. La figure 1.3 montre un exemple de tag RFID. Le projet définit les besoins dans cette technologie, tant au niveau radio-fréquences et protocoles qu'au niveau des informations stockées dans les tags RFID.

Grâce à cette technologie la poubelle de l'utilisateur le guide dans le geste de tri en identifiant les déchets et en lui indiquant le bon bac de destination. Lorsque l'utilisateur ferme le sac les identifiants des déchets contenus dans celui-ci sont enregistrés dans le tag du sac. Le sac est ensuite placé dans la poubelle de ramassage qui embarque un capteur sans-fil capable de lire les informations contenues dans les tags des sacs qu'elle contient. Les poubelles de ramassage peuvent ensuite communiquer avec le centre de tri pour remonter un certain nombre d'informations utiles. Ces informations peuvent être le taux de remplissage de la poubelle, l'exactitude du tri,

5. <http://www.veolia-proprete.com>

6. <http://www.etineo.fr>

7. De l'anglais *radio frequency identification*.

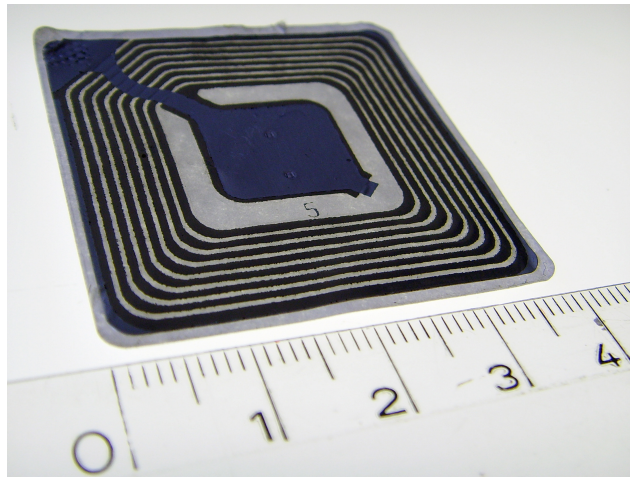


FIGURE 1.3 – Un tag RFID ©Maschinenjunge

les éventuels déchets incompatibles, l'adresse du propriétaire de la poubelle etc.

Deux hypothèses sont envisagées quand à la remonté d'informations vers le centre de tri. La première consiste à équiper l'ensemble des nœuds sans-fil des poubelles de ramassage de deux modules de communication sans-fil. Le premier est un module basse consommation permettant des communications à courte distance entre les nœuds proches. Ce module pourrait être de type ZigBee® par exemple. Le second module est une puce GPRS, permettant de communiquer directement avec le centre de tri mais dont la consommation énergétique est bien plus élevée.

La seconde hypothèse consiste à équiper les nœuds avec uniquement une puce de communication basse consommation. Cela permet de réduire les coûts au niveau du matériel et des abonnements de données ainsi que les besoins en énergie. Cette hypothèse apporte toutefois des contraintes plus forte au niveau du routage des données puisque des sauts multiples devront être effectués pour atteindre le centre de tri. Dans cette hypothèse il est envisagé d'équiper tous les nœuds ou une partie des nœuds de puce GPS permettant de faciliter le routage des données vers le centre de tri.

Dans les deux hypothèses envisagées on notera que les poubelles de ramassage peuvent être mobiles. En effet les utilisateurs stockent bien souvent leur poubelle à un endroit différent de leur lieu de ramassage. Par ailleurs dans certains cas, notamment dans les habitats collectifs, les poubelles sont déplacées juste avant leur ramassage. De ce fait, même si la distance parcourue par les poubelles est faible, il peut être nécessaire aux algorithmes de remontée d'informations de s'adapter rapidement pour envoyer des données avant le ramassage de la poubelle.

Cette thèse permet d'apporter des solutions dans les réseaux de capteurs sans-fil appliquées au projet *BinThatThinks*. Nous étudions l'impact du placement des nœuds sur les algorithmes pour réseaux de capteurs sans-fil (Chapitre 3). Nous

proposons ensuite deux solutions d'organisation et de routage avec différentes hypothèses. La première apporte une solution de remontée d'informations dans l'hypothèse où l'ensemble des nœuds est équipé d'une puce GPRS (Chapitre 4). Elle permet aux poubelles de s'organiser et de communiquer avec un serveur central via une liaison GPRS tout en équilibrant l'énergie dépensée sur les différentes poubelles. La figure 1.4 illustre le cheminement des communications en utilisant cette organisation.

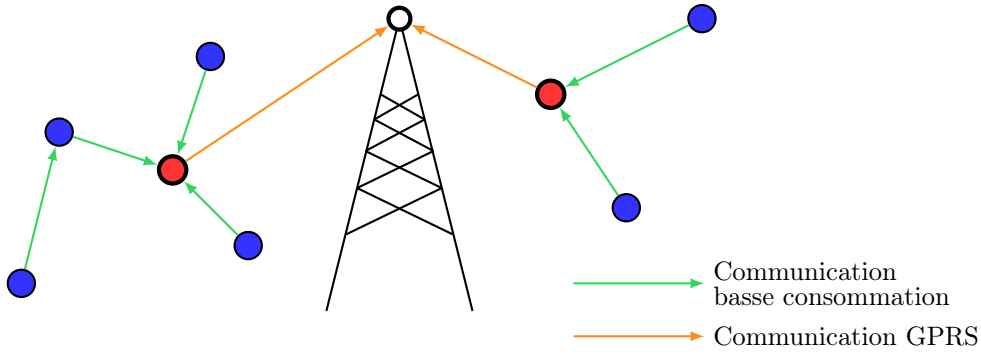


FIGURE 1.4 – Mutualisation des ressources GPRS dans *BinThatThinks*.

La seconde solution permet de router les données en utilisant l'information de position connue de certains nœuds du réseau uniquement (Chapitre 5). Appliqué au projet, cet algorithme permet aux poubelles d'utiliser les nœuds équipés de puce GPS pour router leur données.

1.5 Notations

Nous décrivons ici les différentes notations utilisées dans les différents chapitres de cette thèse, pour les notations spécifiques à un chapitre, il conviendra de se référer à la section *notations* correspondante au chapitre. Nous modélisons un réseau de capteurs sans-fil par un graphe $G = (V, E)$. V représente l'ensemble des nœuds du réseau. E représente l'ensemble des liaisons sans-fil uv entre chaque paire de nœuds u et v . Il existe un lien uv si u et v sont à portée de communication l'une de l'autre. Dans cette thèse nous considérons des liaisons bi-directionnelles, c'est-à-dire que si le lien uv existe alors u reçoit les messages de v et v reçoit les messages de u .

On note $\mathcal{N}(u)$ le voisinage à 1-saut d'un nœud u i.e. $\mathcal{N}(u) = \{\{v\} \mid uv \in E\}$. Le voisinage à 1-saut de u , qu'on appelle aussi simplement le voisinage, est l'ensemble des nœuds tel qu'il existe un lien entre u et ces nœuds.

Le nombre de nœuds dans le voisinage de u est le degré de u : $\delta(u) = |\mathcal{N}(u)|$. C'est aussi le nombre d'arêtes de u .

Enfin, la distance euclidienne entre deux nœuds u et v est notée $|uv|$.

1.6 Contribution et organisation du document

Comme décrit précédemment, cette thèse s'inscrit dans le projet *BinThatThinks*. Nous proposons donc dans ce contexte des analyses et algorithmes pour réseaux de capteurs sans-fil répondant aux contraintes du projet. Pour répondre à ces objectifs, dans la suite de cette thèse, nous nous efforçons de valider nos propositions grâce à des simulations intensives et des expérimentations en environnement réel.

Le chapitre 2 décrit les travaux annexes à cette thèse ainsi que ceux sur lesquels elle s'appuie. Les différentes contributions de cette thèse pouvant être abordées de manière isolée, nous avons fait le choix de décrire dans ce chapitre les travaux annexes qui s'appliquent à l'ensemble des contributions. Les travaux relatifs spécifiquement aux différentes contributions sont quant à eux décrits dans une section spécifique attachée aux différents chapitres de contributions.

Le chapitre 3 présente l'impact des topologies sur les performances des algorithmes de réseaux de capteurs sans-fil. Nous illustrons en analysant divers algorithmes de routage géographique sur plusieurs topologies différentes comme des topologies aléatoires, des topologies provenant de villes et des topologies régulières en grille. Nous montrons grâce à des simulations et des expérimentations sur plateforme que la topologie a un impact significatif sur les performances des algorithmes pour réseaux de capteurs sans-fil.

Une solution de *clustering* permettant l'auto-organisation des réseaux de capteurs sans-fil tout en préservant l'autonomie énergétique globale du réseau est présentée au chapitre 4. Cette solution apporte une réponse pertinente au problème d'énergie des grands réseaux en faisant jouer le rôle de chef de *cluster* à tour de rôle aux différents nœuds du réseau tout en préservant de bonnes propriétés pour le réseau,

Dans le chapitre 5 nous faisons l'hypothèse qu'un certain nombre de nœuds du réseau sont géolocalisés alors que d'autres ne le sont pas. Dans ce contexte, nous proposons une solution de routage hybride mixant routage géographique et routage réactif. Nous montrons que cette solution est adaptée dans certaines conditions de densité et de proportion de nœuds géolocalisés.

La conclusion de cette thèse ainsi que les perspectives seront développées en section 6.

État de l'art

Sommaire

2.1	Les villes intelligentes	27
2.2	Les solutions pour la gestion des déchets	30
2.3	Plates-formes de simulation et d'expérimentation	33

Dans cette thèse, les différents chapitres de contribution (chapitres 3, 4 et 5) abordent des techniques assez distinctes. Chacun de ces chapitres inclut donc une section état de l'art qui aborde les travaux liés au chapitre en question. Nous aborderons ici les sujets communs aux différents chapitres, notamment celui des villes intelligentes au sens général et les sujets liés au projet *BinThatThinks* qui soutient cette thèse.

2.1 Les villes intelligentes

L'architecture générale d'une ville intelligente est décrite par Al-Hader et al. [26]. Ils discutent des différents éléments essentiels à considérer dans le développement d'une ville intelligente. Il s'agit notamment de l'infrastructure, de la base de données, d'un système de positionnement précis, du système de gestion des bâtiments, etc. Cet article aborde les lignes directrices de la création et de la mise en place des différents éléments de la ville intelligente. Il soulève la nécessité de penser les infrastructures au plus tôt dans le développement de la ville intelligente. Les auteurs indiquent que l'infrastructure sans-fil est un élément clé de la ville intelligente mais qu'il ne s'agit là que d'un élément parmi d'autres qui ne sont pas non plus à négliger.

Filipponi et al. [27] proposent une plate-forme d'inter-opérabilité appelée SOFIA. Leur solution est un intergiciel permettant d'abstraire différents réseaux de capteurs hétérogènes et de les rendre accessibles via une interface reposant sur des ontologies¹. Sur le même sujet, IBM [28] propose des solutions permettant de connecter ensemble les différents éléments d'une ville intelligente. Ils proposent une solution pour gérer l'importante quantité de données générées par les différents composants de la ville intelligente et permettent d'obtenir une vue d'ensemble à destination des décideurs. Ces intergiciels proposent des solutions intéressantes dans le cadre de projets comme

1. Wikipédia : *une ontologie est l'ensemble structuré des termes et concepts représentant le sens d'un champ d'informations. On peut aussi dire que l'«ontologie est aux données ce que la grammaire est au langage ».*

BinThatThinks. Ces projets offrent des licences d'utilisation variées qui pourront satisfaire des besoins différents comme le support ou la possibilité d'accéder et de modifier l'intergiciel.

Denardin et al. [10] décrivent une solution complète pour la gestion des lampadaires dans le contexte d'une ville intelligente. L'idée générale proposée est d'équiper les lampadaires avec un système d'éclairage utilisant des diodes électroluminescentes (LED), une batterie et un système de communication à basse consommation. Avec cet équipement et la possibilité de communiquer, chaque lampadaire est en mesure de basculer son mode d'alimentation depuis le réseau électrique vers sa batterie. Cela permet de délester le réseau d'alimentation principal lors des pics de consommation. Les auteurs proposent un algorithme de routage géographique avec garantie de livraison. L'utilisation de stations de base permet la propagation d'un gradient² sur les nœuds du réseau. Ce gradient est utilisé pour éviter les impasses dans le routage, permettant ainsi d'offrir de meilleures performances énergétiques et un meilleur délai de livraison que les algorithmes utilisés habituellement comme GPSR [29]. Dans le cadre du projet *BinThatThinks*, l'algorithme pourrait être appliqué aux containers moyennant d'éventuelles modifications. Toutefois la présence d'une puce GPS sur chaque poubelle n'est pas systématique. De plus, la réception du signal satellite peut ne pas être suffisante dans certaines conditions, empêchant l'algorithme de fonctionner correctement. Il est aussi envisageable de faire cohabiter le réseau de lampadaires avec le réseau des poubelles. Les poubelles intelligentes, autonomes en énergie, pourraient simplement générer des données et utiliser le réseau de lampadaires pour router les informations, leur permettant ainsi d'atteindre une grande autonomie. Toutefois cela nécessite un partenariat ou une collaboration avec l'organisme en charge de la gestion des lampadaires ou une gestion globale des systèmes intelligents d'une ville, ce qui n'est pas le cas dans *BinThatThinks*.

Parmi les applications emblématiques pour les villes intelligentes, on retrouve la gestion efficace des réseaux électriques (*smart grids*) [30]. Il s'agit de gérer les différents éléments de ces réseaux pour faire face aux pannes ou aux pics de consommation par exemple. L'idée étant de gérer le réseau de façon automatique et décentralisée.

La ville intelligente fait aussi la part belle aux applications de gestion du trafic routier. Une gestion globale des feux tricolores est un moyen de réduire les embouteillages dans les villes, permettant de réduire les émissions de CO_2 et autres gaz polluants. Wiering et al. [31] présentent un algorithme de gestion des feux tricolores pour améliorer la fluidité du trafic routier.

Toujours dans le but d'améliorer le trafic routier, la gestion du stationnement permet de désencombrer les axes de circulation en aidant les automobilistes à trouver plus rapidement une place de stationnement [32]. Il est nécessaire pour cela de surveiller les emplacements de parking et d'informer les usagers de l'emplacement des places disponibles via un réseau de communication et des protocoles adaptés.

2. Un suite monotone croissante qui augmente à chaque saut.

Dohler et al. [33] proposent une RFC décrivant les besoins des protocoles de routage dans les réseaux urbains à basse consommation. Ce document discute notamment du déploiement des nœuds, de l’extensibilité, de la fiabilité des liens, de la latence, etc. Des considérations importantes sont indiquées sur le type de trafic de données rencontré dans ces réseaux. Les questions de sécurité sont également mentionnées. Ce document offre une source d’inspiration intéressante pour le développement d’algorithmes, protocoles ou applications pour les villes intelligentes. Les contributions de cette thèse reprennent des hypothèses discutées dans ce document. On notera notamment les besoins d’évolutivité des réseaux de part leur grande taille, les auteurs indiquent en effet des réseaux allant de 100 à 10 000 000 de nœuds. Le réseau est aussi susceptible de s’agrandir au gré du déploiement de nouveaux nœuds, les protocoles de routage doivent donc pouvoir s’accommoder d’une grande quantité de nœuds et d’augmentations significatives de leur nombre sans détérioration des performances. Par ailleurs, on notera que le trafic est fortement directionnel et que la configuration automatique des nœuds est vivement recommandée. Ces paramètres ont été soigneusement pris en compte dans les contributions de cette thèse.

Hanke et al. [34] proposent un aperçu de solutions de sensing pour villes intelligentes. Ils décrivent un large éventail de solutions allant des solutions sans-fil (ZigBee®, 4G, NFC, ...) aux projets de villes intelligentes (Songdo en Corée, Fujisawa au Japon, Santander en Espagne, ...) en passant par les plates-formes matérielles de capteurs sans-fil (*Mica2*, *Senspot*, *Waspmote*, ...) ainsi que les applications possibles (distribution d’eau et d’électricité, transports intelligents, monitoring environnemental, ...). Les auteurs concluent qu’un certain nombre de défis techniques restent à surmonter mais que les défis d’ordre sociologique doivent aussi être pris en compte. Dans *BinThatThinks*, l’acceptation des solutions envisagées est une donnée importante de la réflexion. Les solutions proposées se veulent faciles à utiliser, accessibles et respectant la vie privée des utilisateurs.

La ville de Santander au nord de l’Espagne est un laboratoire grandeur nature pour villes intelligentes [35, 36, 37, 38]. Plusieurs milliers d’appareils mesurent et interagissent avec leur environnement. Les différents périphériques déployés dans la ville permettent de surveiller l’environnement comme le bruit, les emplacements de parking, l’état du trafic ... Il permettent aussi de guider les utilisateurs vers des emplacements de parking gratuits grâce à des panneaux, de contrôler l’arrosage des jardins, etc. Les utilisateurs sont par ailleurs invités à participer au réseau à l’aide de leur smartphone. La ville de Santander fait partie du projet SmartSantander [39] qui regroupe désormais 4 villes européennes et dont le but est de déployer plus de 20 000 capteurs. La société Libelium [40, 41], partenaire du projet SmartSantander, développe des solutions matérielles pour le sensing dans les villes intelligentes disponibles pour tous. Les plates-formes comme SmartSantander offrent un outil de choix pour tester en environnement réel (matériel, conditions environnementales, ...) des applications pour les villes intelligentes.

La ville de Santander propose probablement le projet de ville intelligente le plus ouvert. Toutefois, il existe de nombreuses villes intelligentes à travers le monde.

En France, la ville de Lyon, avec Grand Lyon [42] entreprend de nombreux projets autour du thème de la ville intelligente, notamment autour de la gestion des déchets, des économies d'énergie et de l'environnement. On pourra aussi citer les exemples de Dubaï [43], Amsterdam [44], ... Le projet *BinThatThinks* apporte de nouvelles solutions pour ces projets, avec des réponses concrètes aux besoins de ces villes et des solutions technologiques fonctionnelles.

Nous avons présenté différentes solutions pour les villes intelligentes, allant du matériel aux intergiciels pour la gestion des données en passant par les protocoles de routage. Toutefois comme l'indiquent les articles des Al-Hader et al. et Hancke et al. [26, 34], les défis techniques sont encore nombreux et de nombreuses solutions restent à trouver. Afin de rendre les capteurs moins visibles et pour diminuer leur impact sur l'environnement il est important de continuer les avancées sur le matériel. Les avancées sur l'autonomie en énergie combinées à des algorithmes toujours plus performants énergétiquement permettront de déployer des capteurs n'importe où, sans se soucier de leur durée de vie. Enfin, des intergiciels performants permettant de stocker et d'analyser efficacement les données fournies par les différents éléments de la ville intelligente sont primordiaux pour une utilisation efficace des équipements.

2.2 Les solutions pour la gestion des déchets

Il existe un certain nombre de projets, de produits et d'articles dont l'objectif est de faciliter ou améliorer le recyclage des déchets grâce aux nouvelles technologies. L'idée derrière ces nouvelles technologies est d'améliorer le recyclage par les utilisateurs pour en diminuer les coûts, inciter à recycler plus et diminuer la pollution.

2.2.1 La RFID

La RFID est une technologie clé dans l'identification des objets. La gestion des déchets repose en grande partie sur l'identification, soit des déchets, soit des bacs de collecte ou des poubelles, la RFID est donc une technologie de choix pour cette tâche.

Dans une ville intelligente, une gestion responsable des déchets semble primordiale. Pour cela Chowdhury et al. [45] proposent d'équiper les poubelles de la ville avec des tags RFID pour permettre l'identification de leurs propriétaires. Le but de l'identification est de permettre une facturation individuelle et de limiter les vols. Les auteurs décrivent un modèle définissant les différents éléments du système de collecte et de consultation des données.

Betty-box [46] est une machine de collecte de déchets automatique dans laquelle le recyclage est effectué grâce à des capteurs, des balances et des détecteurs optiques. Les utilisateurs amènent leurs déchets au point de collecte. Ces déchets sont alors analysés automatiquement par la machine qui rémunère l'utilisateur en fonction des matériaux et de leur poids. Cette solution présente l'avantage d'inciter les utilisateurs à recycler, tout en étant simple. Toutefois, l'encombrement de la machine est

très important et l'utilisateur doit se déplacer jusqu'au point de collecte pour trier ses déchets, ce qui est contraignant et représente un frein à son utilisation.

Les déchets électroniques sont majoritairement très polluants, or ils sont souvent déposés dans des décharges où ils peuvent contaminer les sols et les nappes phréatiques. De plus, la quantité de ce type de déchets est en constante augmentation. C'est le constat que font Kahhat et al. [47] qui proposent un système de rachat des appareils électriques qui met en compétition différentes entreprises. L'étude étant axée pour le marché américain, plus sensible aux arguments économiques que législatifs, les auteurs proposent un système économiquement neutre pour les différents acteurs du cycle de vie du produit. La gestion du cycle de vie du produit s'appuie sur la technologie RFID pour suivre les flux économiques et matériels. Bien qu'intéressante, cette proposition ne considère que le problème des déchets électroniques, or le recyclage est un problème global qui mérite d'être traité dans son ensemble.

Binder et Domnitcheva [48] proposent d'utiliser la technologie RFID et ses atouts (petite taille, sans batterie, faibles coûts si produits en masse) pour taguer les déchets. La distinction est faite entre les objets durables et les objets à vie courte. Les objets à vie courte sont souvent composés d'un matériau et se verraient équipés d'un tag RFID. D'un autre côté les objets durables sont souvent composés de différents éléments nécessitant leur désassemblage. Un tag serait alors placé sur chaque élément constituant l'objet. Grâce aux techniques anti-collision des tags RFID, il serait alors possible de scanner l'ensemble des tags de l'objet pour en connaître la composition et procéder à son désassemblage et au recyclage des différents composants. Cet article propose une solution intéressante utilisant les caractéristiques de la technologie RFID, toutefois il ne discute pas des infrastructures nécessaires au fonctionnement du système. De plus l'équipement en tags RFID des objets est envisagé depuis de nombreuses années, or il n'est toujours pas effectif et prendrait vraisemblablement plusieurs autres années. Équiper les objets complexes de plusieurs tags prendrait probablement plus de temps encore, or il est nécessaire de trouver des solutions pour recycler efficacement ces produits en attendant le déploiement de la RFID.

Le projet *Bin it!* [49], lauréat du concours *The Sentient Future Competition*³ est une autre proposition consistant à équiper les objets de tags RFID en vue de leur recyclage. La particularité de ce projet est l'utilisation d'une carte de collecte contenant un crédit que l'utilisateur peut monnayer. Chaque fois que l'utilisateur recycle un objet correctement, il voit sa carte de collecte créditée du montant correspondant à l'objet collecté. Le projet considère que la carte de collecte puisse être anonyme pour garantir la vie privée des utilisateurs. Toutefois un modèle sans anonymat est aussi envisagé pour infliger une amende aux propriétaires d'objets jetés dans la nature. Ce modèle repose lui aussi sur un mode de récupération des objets recyclables dans des points de collecte spécifiques, ajoutant une contrainte pour l'utilisateur qui freine l'adoption du système.

3. <http://www.embedded-wisents.org/competition/competition.htm>

L'utilisation de la RFID dans la gestion des déchets est un sujet d'actualité. Un article de DiscoverRFID [50] indique que la RFID se développe pour la gestion du ramassage des ordures pour diminuer les travaux pénibles et augmenter la productivité du ramassage des ordures. Dans un autre article de *parole d'élus* [51], l'utilisation des RFID par le leader mondial de la conteneurisation des déchets est mise en avant pour la mise en place de redevances incitatives et la réduction des coûts de collecte de 20 à 30%. Par ailleurs la réglementation sur les déchets jouera un rôle important dans l'adoption de la RFID pour le suivi des déchets notamment pour les déchets dangereux [52, 53].

La disponibilité de solutions RFID clés en main, comme celles de *TAG Product* [54] ou de *RFIDHY Technology* [55], montrent une réelle maturité du marché des RFID pour la gestion des déchets. Toutefois leur intégration reste marginale et limitée à la simple identification des containers, *BinThatThinks* propose des solutions plus avancées et destinées à être déployées massivement.

Les solutions RFID pour la gestion des déchets sont nombreuses et parfois même déjà déployées en phase d'expérimentation ou de production. L'identification des containers est la solution la plus mature dans la gestion des déchets avec RFID. L'identification des déchets eux-mêmes reste une solution souvent décrite mais la mise en place difficile et coûteuse freine encore son adoption. Son déploiement nécessite des modifications sur l'ensemble des chaînes de fabrication, entraînant de longs délais dans sa mise en place. Les réglementations actuelles et futures [56] imposeront ou favoriseront probablement l'adoption des RFID, d'abord sur les déchets dangereux puis probablement sur les déchets communs.

2.2.2 Les autres technologies

Nuortio et al [57] proposent un algorithme pour l'optimisation des routes empruntées par les camions de collecte des déchets. Ils utilisent des traces GPS réelles de camions de ramassage en Finlande et mettent en évidence que le trajet des camions de collecte peut être diminué de 12% en moyenne. Par ailleurs, ils indiquent qu'un certain nombre de containers sont collectés alors qu'ils ne sont pas pleins. L'utilisation de ces travaux combinés à la possibilité offerte par le projet *BinThatThinks* d'obtenir des informations sur (entre autres choses) le niveau de remplissage des poubelles permettrait d'économiser une quantité non négligeable de carburant grâce à la diminution de la distance parcourue par le camion et au temps gagné lors de la collecte.

La start-up Finlandaise *Enovo* [58] propose un capteur sans-fil pouvant s'installer sur n'importe quel container ou poubelle. Grâce aux informations collectées par les capteurs et à des algorithmes de recherche de chemin performants il est possible d'optimiser la tournée de ramassage des ordures. *Etineo*⁴, partenaire du projet *BinThatThinks*, propose aussi des solutions matérielles servant de base au projet, notamment avec la solution *Etinode* de réseaux de capteurs intelligents.

4. <http://www.etineo.fr/>

La société *Ijinus* [59] propose des solutions de smart metering pour les silos, les containers ou pour la prévention des inondations. Les envois de données peuvent se faire sans-fil en utilisant la technologie GPRS par exemple. Les capteurs sont autonomes en énergie et ont une durée de vie de plusieurs années en effectuant plusieurs mesures par heure et un envoi de données par jour. *SigrenEa* [60, 61] propose des solutions clés en main pour les collectivités afin d'équiper les points de collecte de capteurs sans-fil pour mesurer le taux de remplissage des containers et améliorer les tournées de collecte. Ces solutions sont autonomes, chaque équipement émet ses données directement à une station de base via GPRS. Une telle solution est très coûteuse pour un déploiement à très large échelle et la multiplication des émissions sur le réseau GSM engendre des problèmes tels qu'une augmentation des émissions radio de forte puissance et des problèmes de saturation de bande passante.

Les différentes solutions exposées proposent des solutions innovantes pour le recyclage des déchets, toutefois une grande partie de ces solutions reposent sur une architecture centralisée. Une architecture centralisée est peu robuste car elle ne permet pas de s'adapter à des changements de dernière minute. Par ailleurs, en plus de devoir être connectés au réseau (sauf éventuellement pour *Bin It!*) les points de collectes sont situés à l'extérieur de l'habitation des utilisateurs. Cela limite l'adoption de ces solutions, certains utilisateurs préférant jeter les déchets recyclables avec les ordures plutôt que de se déplacer jusqu'aux points de collectes.

2.3 Plates-formes de simulation et d'expérimentation

Dans cette thèse, afin d'évaluer les solutions que nous proposons, nous utilisons les plates-formes WSNET et **SensLAB**. Dans cette section nous proposons une présentation de ces outils.

Pour développer de nouveaux algorithmes et protocoles et pour évaluer leurs performances, il est nécessaire d'utiliser des plates-formes de simulation et d'expérimentation. La plate-forme de simulation Worldsens [62] offre un simulateur haut niveau : WSNET et un simulateur bas niveau : WSIM pour réseaux de capteurs sans-fil. WSNET permet de simuler des applications et des protocoles réseau à haut niveau afin de vérifier le bon fonctionnement de l'application ou du protocole, d'en établir les performances ou d'en régler les paramètres. WSIM permet quant à lui de simuler fidèlement un nœud-capteur à bas-niveau. Il exécute du code compilé pour la plate-forme cible et émule le matériel afin de reproduire le comportement exact du nœud-capteur avec le code développé. WSNET et WSIM peuvent être utilisés conjointement pour itérer le développement haut-niveau et bas-niveau simultanément.

SensLAB [63, 64] est une plate-forme d'expérimentation pour réseaux de capteurs sans-fil ouverte dédiée à la recherche. **SensLAB** propose 1024 nœuds répartis sur 4 sites géographiques : Grenoble, Lille, Rennes et Strasbourg. Chaque nœud est équipé d'un processeur *Texas Instrument MSP430* [65] 16 bit avec 10 kbit de mé-

moire vive, 48 kbit de mémoire flash interne et 1 Mbit de mémoire flash externe. Les nœuds de Lille et Rennes sont équipés d'une interface radio IEEE 802.15.4TM *Texas Instrument cc2420* [66] fonctionnant dans la bande des 2.4 GHz. Les nœuds de Grenoble et Strasbourg sont équipés d'une interface radio IEEE 802.11TM *Texas Instrument cc1100* [67] fonctionnant dans la bande des 868 MHz. Une partie des nœuds des sites de Lille et de Strasbourg sont mobiles afin de tester des protocoles ou des applications impliquant la mobilité de certains nœuds.

Impact de la topologie sur les performances des réseaux de capteurs sans-fil

Sommaire

3.1	Introduction	35
3.2	État de l'art	36
3.3	Motivations	38
3.4	Algorithmes	39
3.5	Topologies	40
3.6	Simulations	42
3.7	Expérimentations	52
3.8	Discussion	57
3.9	Conclusion	59

3.1 Introduction

Les réseaux de capteurs sans-fil sont déployés dans des environnements très variés. Il peut s'agir d'un environnement très contrôlé comme un laboratoire ou, à l'opposé, d'un environnement extrême tel qu'un volcan actif ou la scène d'un tremblement de terre. Entre ces deux extrêmes on retrouve une multitude de possibilités différentes, comme le contrôle des lampadaires dans une ville intelligente ou encore la collecte intelligente des déchets. Le placement des nœuds pour chacun des scénarios peut-être contrôlé ou subi, déterministe ou aléatoire. Dans un laboratoire on pourra choisir l'emplacement exact des nœuds, on aura alors un environnement contrôlé. Dans le cas du contrôle de lampadaires dans les villes intelligentes, le placement des nœuds n'est pas contrôlé mais est connu à l'avance, il est déterministe. Enfin dans un scénario tel que la surveillance de volcan actif où les nœuds sont déployés depuis les airs, leur placement est à la fois non contrôlé et aléatoire. Il est en effet impossible de prévoir à l'avance le placement de chaque nœud.

Dans le cadre du projet *BinThatThinks* les nœuds sont déployés dans une ville, sur les poubelles et les bacs de collecte. Dans ce contexte, le placement des nœuds est spécifique et mérite une attention particulière.

Un telle diversité dans la topologie des réseaux de capteurs amène à penser qu'il est aussi nécessaire de diversifier les algorithmes de routage pour pouvoir s'adapter à ces différentes topologies. En effet, il est légitime de penser qu'un algorithme adapté au réseau de lampadaires d'une ville intelligente ne sera pas adapté au réseau de surveillance d'un volcan. Dans la littérature, les algorithmes pour réseaux de capteurs sans-fil sont souvent évalués sur une seule topologie réseau. Très souvent cette topologie est aléatoire.

Sur des topologies spécifique, notamment celles du projet *BinThatThinks*, il est intéressant d'observer si l'on doit faire face à des contraintes supplémentaires ou si, au contraire, il est possible de tirer des bénéfices d'un placement particulier des nœuds.

Pour toutes ces raisons, nous avons souhaité étudier l'impact de différentes topologies sur les performances des algorithmes pour réseaux de capteurs sans-fil. Dans ce chapitre nous étudions cet impact sur des algorithmes de routage géographique. Ces algorithmes utilisent la position des nœuds pour transmettre les informations d'un nœud à un autre. La destination d'un paquet de données est ici une position plutôt que l'adresse d'un nœud. Grâce aux messages HELLO, un nœud a connaissance de la position des nœuds qui sont ses voisins directs, c'est-à-dire qui sont à portée de communication. Ainsi un nœud qui souhaite envoyer une information vers une position donnée pourra choisir parmi ses voisins celui qui est le plus proche de la destination comme intermédiaire dans le routage.

3.2 État de l'art

Beaucoup de recherches et d'analyses de performances sont faites sur l'évaluation des protocoles ou l'analyse de l'efficacité énergétique. Cependant, peu de recherches ont été menées sur l'impact de la topologie des réseaux sur les performances des protocoles dans les réseaux de capteurs sans-fil. La plupart des recherches dans ce domaine se concentrent sur le placement efficace des nœuds dans le but d'obtenir des performances optimales mais pas sur l'impact du placement des nœuds.

Dhillon et Chakrabarty [68] se concentrent sur le placement efficace des nœuds pour la couverture de zone. Le but de leurs propositions est de minimiser le nombre de ressources nécessaires pour la couverture effective d'une zone définie. Ils utilisent un modèle de détection probabiliste dans lequel un capteur a une probabilité plus ou moins élevée de détecter un événement en fonction de sa distance à l'événement. Les points d'intérêt à surveiller sont placés sur une grille et la notion d'obstacle est considérée. Sur ces hypothèses, les auteurs proposent deux algorithmes pour couvrir chaque point d'intérêt avec un niveau de confiance minimum.

Dasgupta et al. [69] proposent la première solution pour le problème de maximisation de la durée de vie du réseau avec contraintes de couverture. Ils proposent l'algorithme *SPRING* qui apporte une solution sous-optimale, dont les performances, mesurées expérimentalement sont meilleures qu'un placement aléatoire.

Certains travaux étudient des algorithmes de placement et de mobilité dans le but d'optimiser la couverture de zone et la connexité du réseau comme Wang et al. [70] et Erdelj et al. [71].

Younis et Akkaya [72] ont compilé les recherches dans le domaine du placement des nœuds dans une étude couvrant différentes techniques pour la surveillance de zone, l'optimisation de la consommation énergétique et l'optimisation de la connectivité du réseau. Cette étude couvre le placement statique des nœuds ainsi que leurs transferts après leur placement initial. Les nœuds générateurs de données, les nœuds relais et les nœuds polyvalents sont différenciés dans cette étude. Les auteurs mettent en évidence que la position des nœuds a un impact significatif sur la durée de vie du réseau et sur la résistance aux pannes.

Ces travaux ne répondent pas à la question de l'impact du placement des nœuds mais apportent une première approche de son influence sur les performances du réseau, comme la durée de vie de celui-ci.

Un autre pan de recherche ne se concentre pas sur la topologie des réseaux de capteurs sans-fil mais sur les topologies rencontrées sur le réseau Internet [73, 74].

Bhardwaj et al. [75] étudient les limites théoriques de la durée de vie des réseaux de capteurs sans-fil. Ils analysent ensuite les performances des techniques d'agrégation de données face à ces limites. Dans cette étude chaque nœud possède une quantité d'énergie initiale qui diminue chaque fois qu'un nœud reçoit ou émet une donnée ou qu'il analyse son environnement. La durée de vie du réseau y est définie comme le temps pendant lequel toute la zone à observer est couverte. L'analyse des limites maximales théoriques de la durée de vie des réseaux permet d'obtenir un élément de comparaison pertinent pour la mise au point d'algorithmes visant à maximiser ladite durée de vie. Les auteurs étudient différents placements d'événements à reporter pour les nœuds-capteurs et mettent en avant des résultats différents. Cet article n'étudie pas spécifiquement le placement des nœuds dans le réseau mais le placement des événements. Les différences de performances indiquent que le placement des nœuds a aussi une influence sur les performances du réseau.

Ishizuka et Aida [76] étudient le comportement de réseaux de capteurs sans-fil sur des topologies aléatoires aux propriétés différentes. Les auteurs étudient trois topologies aléatoires différentes : une *diffusion simple*, un *placement constant* et un *placement r -aléatoire*. La topologie *diffusion simple* représente un placement de nœuds dispersés depuis les airs, la densité est donc plus élevée à proximité du point de déploiement. La topologie *placement constant* est telle que la densité est constante, elle est souvent utilisée dans les simulations pour réseau de capteurs. Enfin dans la topologie *placement r -aléatoire*, imaginée par les auteurs, les nœuds sont uniformément dispersés en fonction de la distance et de l'angle par rapport à la station de base. Ces travaux montrent que la topologie de *diffusion simple* a une faible probabilité de capter les événements alors que la topologie *placement constant* a une faible tolérance aux pannes. La topologie proposée par les auteurs, le *placement r -aléatoire*, permet une bonne tolérance aux pannes et une bonne couverture de détection, offrant un bon compromis. Cet article permet de mettre en avant des

écarts de performance entre différentes topologies aléatoires mais n'étudie pas les topologies que l'on peut rencontrer dans une ville qui ne sont pas aléatoires

Vassiliou et Sergiou [77] étudient l'impact de quatre topologies sur les performances de trois algorithmes de contrôle de congestion. Les topologies étudiées sont les mêmes que dans l'article de Ishizuka et Aida [76] (*diffusion simple*, *placement constant* et *placement r -aléatoire*) avec l'ajout d'un *placement en grille*. Les algorithmes étudiés sont *SenTCP*, *Directed Diffusion* et *HTAP*. Les auteurs montrent que les performances des algorithmes sont affectés par la topologie. Ils montrent que certains algorithmes sont affectés au niveau des délais de transmission alors que d'autres sont plutôt affectés au niveau du taux de livraison. L'impact de la topologie est à nouveau mis en avant dans cet article mais les topologies sont toujours simples et ne représentent pas la réalité de déploiement dans une ville intelligente.

Puccinelli et al. [78] proposent l'une des rares études incluant des résultats expérimentaux. Ils introduisent par ailleurs une métrique permettant de caractériser une topologie appelée *END* (Expected Network Delivery) qui permet d'obtenir le taux de livraison du réseau en fonction des liens entre les nœuds indépendamment du protocole utilisé. Les auteurs concluent en indiquant que l'étude des performances sans considération de la topologie ne donne qu'une image incomplète des performances du protocole étudié.

L'article de Karim et al. [79] met en évidence les différences de performances des protocoles en fonction du contrôle de topologie appliqué. Ces travaux n'étudient pas directement l'impact de la topologie sur les performances mais indirectement par l'utilisation du contrôle de topologie.

La littérature ne regorge pas d'études spécifiques à l'impact de la topologie sur les performances des algorithmes pour réseaux de capteurs sans-fil. Toutefois, les quelques articles s'y intéressant montrent que les performances semblent bien affectées par la topologie mais il est nécessaire d'approfondir les recherches, notamment sur l'impact de topologies que l'on peut rencontrer dans une ville intelligente.

3.3 Motivations

Nous avons vu dans l'état de l'art que l'étude de l'impact des topologies sur les réseaux de capteurs sans-fil n'était pas très courant. Un grand nombre d'articles se concentrent sur l'amélioration d'un aspect précis d'un protocole ou d'un algorithme. Seul quelques uns considèrent la topologie comme un aspect important du design des algorithmes pour les réseaux de capteurs sans-fil.

Dans ce chapitre nous montrons d'abord que la topologie des réseaux peut être de nature très variée. Même si elles ont des propriétés similaires tels que le nombre de nœuds, le degré, la surface, etc, ces topologies ne sont pas égales vis-à-vis des performances des algorithmes exécutés sur celles-ci. Nous montrons que ces différentes topologies impactent les performances des algorithmes pour réseaux de capteurs sans-fil, en particulier avec les algorithmes de routage géographique.

Nous avons choisi d'illustrer ces différences sur des algorithmes de routage géographique car ils offrent des propriétés intéressantes et sont à même d'être utilisés pour le routage dans les villes intelligentes. Les protocoles de routage géographique nécessitent peu de mémoire, et peu de ressources calculatoires. En effet un nœud doit uniquement mémoriser son voisinage et le calcul du prochain saut repose sur des algorithmes dont la complexité est linéaire en fonction du nombre de voisins. De plus ils sont totalement localisés, ce qui leur permet une grande extensibilité. Ces propriétés les rendent particulièrement adaptés aux réseaux de capteurs sans-fil, limités en ressources, dans le contexte des villes intelligentes à large échelle.

Les variantes choisies sont sans récupération en cas de minimum local pour être en mesure d'observer l'impact des topologies sur le taux de livraison. Le choix des topologies permet de comparer le placement des nœuds souvent utilisé dans l'évaluation des performances des algorithmes de la littérature avec des placements plus adaptés aux villes intelligentes.

Nous montrons spécifiquement que les caractéristiques comme le taux de livraison, la taille des routes et l'énergie consommée ont des comportements différents. Nous voulons mettre en avant que les algorithmes pour réseaux de capteurs sans-fil doivent être conçus avec l'application ciblée en tête.

3.4 Algorithmes

Les algorithmes étudiés dans ce chapitre sont différentes variantes du *Greedy Geographic Routing*. Il s'agit d'un routage géographique sans récupération en cas d'impasse. L'utilisation de différentes variantes sans récupération permet d'étudier le taux de livraison. En effet celui-ci serait de 100% pour toutes les variantes si la récupération était possible car les topologies considérées sont toutes connexes. La différence entre ces variantes se situe au niveau de la métrique utilisée pour choisir le prochain nœud dans le routage. La figure 3.1 résume le comportement des différents algorithmes présentés. Les variantes qui sont étudiées sont les suivantes :

- *Greedy* [22] : Choisit le nœud parmi les voisins qui minimise la distance euclidienne avec la destination (nœud **b** sur la figure 3.1).
- *MFR*¹ [80] : Choisit le nœud qui minimise la distance euclidienne entre la projection orthogonale du voisin sur la droite définie par la source et la destination, et la destination (nœud **a** sur la figure 3.1).
- *NFP*² [81] : Choisit le nœud n qui minimise la distance euclidienne avec la source et qui permet un progrès vers la destination ($|nD| < |SD|$) (nœud **c** sur la figure 3.1).
- *Compass* [82] : Choisit le nœud qui minimise l'angle voisin-source-destination ($\widehat{eSD}$ sur la figure 3.1).

1. Most Forwarding within Range : le meilleur progrès dans le voisinage.

2. Nearest with Forwarding Progress : le plus proche permettant un progrès.

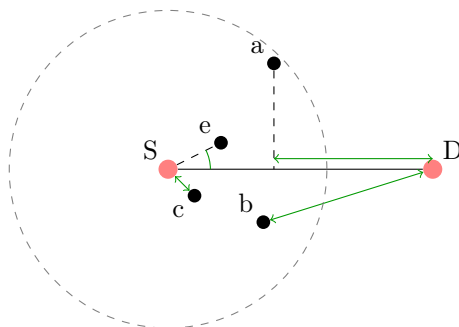


FIGURE 3.1 – Comparaison des variantes de l’algorithme de routage géographique. Le nœud a est choisi par la variante *MFR*, le nœud b par la variante *Greedy*, le nœud c par la variante *NFP* et le nœud e par la variante *Compass*.

3.5 Topologies

Des simulations sont exécutées sur cinq topologies différentes tandis que des expérimentations sont réalisées sur quatre topologies différentes. Les topologies sont choisies pour être suffisamment différentes les unes des autres. Elles sont soit représentatives des topologies que l’on retrouve dans la littérature pour l’évaluation de solutions pour réseaux de capteurs sans-fil, soit représentatives de topologies réalistes.

La première topologie utilisée est une topologie aléatoire. Elle est utilisée à la fois dans les simulations et les expérimentations. La figure 3.2 donne un exemple de topologie aléatoire. Dans les simulations, 1619 nœuds sont répartis de manière

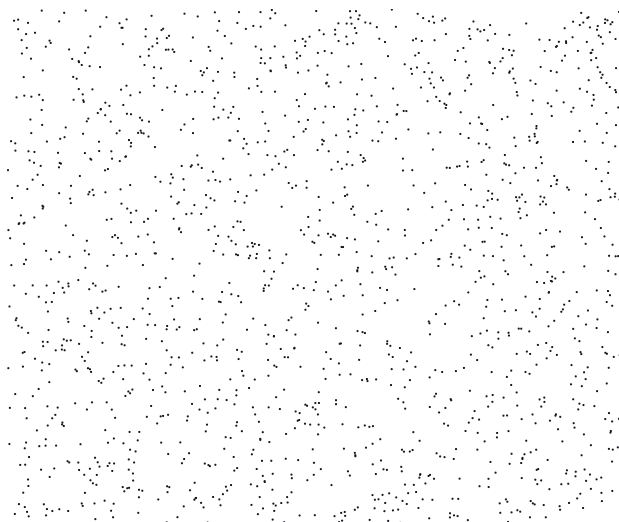


FIGURE 3.2 – Exemple de topologie aléatoire.

totalement aléatoire sur une grille de $600 \text{ m} \times 450 \text{ m}$. Pour chaque simulation, une nouvelle topologie est générée. Dans les expérimentations sur **SensLAB** [63] les nœuds ne peuvent pas être placés de manière totalement aléatoire. En effet il n'est pas possible de déplacer les nœuds. La solution choisie a donc été de sélectionner un sous ensemble de 60 nœuds parmi les 256 nœuds proposés par une plate-forme. Nous appelons cette topologie *random*.

La topologie *random hole* est une variante de la topologie *random*. Le placement des nœuds est similaire mais une zone circulaire, d'un diamètre de 100 m , aléatoirement placée, ne contient aucun nœud.

La troisième topologie, appelée *city*, est la représentation d'un déploiement de réseaux de capteurs sans-fil dans une ville européenne. La figure 3.3 montre la topologie *city*, on y voit les routes et les intersections de la ville.

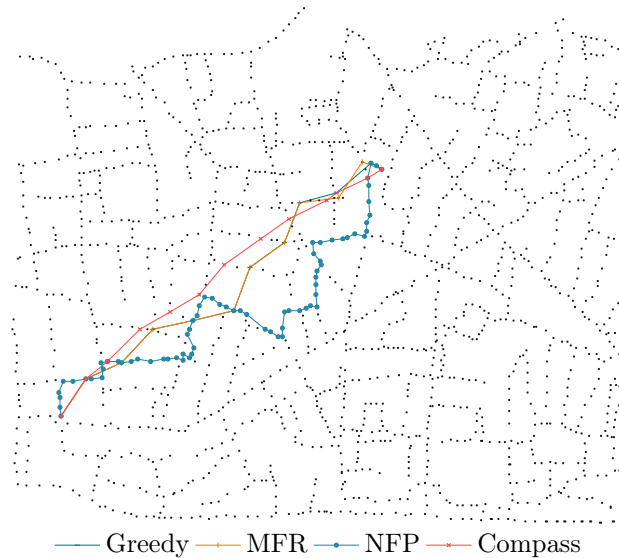
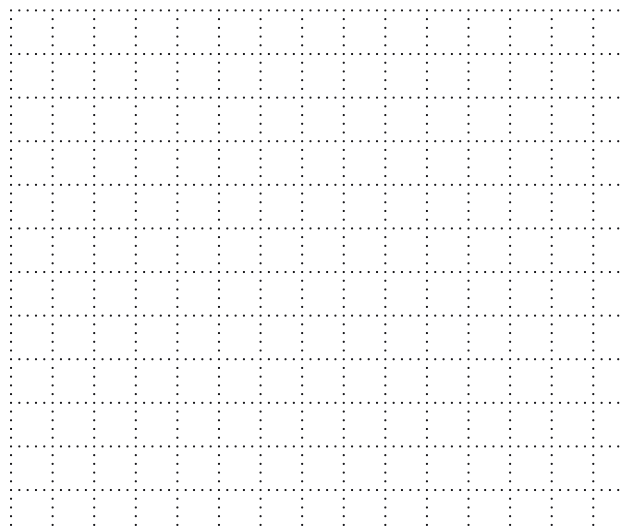


FIGURE 3.3 – Topologie *city* avec un exemple de route.

La topologie *city grid*, illustrée par la figure 3.4, est une représentation théorique d'une ville de type Manhattan. Dans cette topologie, les nœuds sont placés régulièrement en lignes et en colonnes, l'espacement entre deux nœuds consécutifs d'une ligne ou d'une colonne est constant.

Enfin la topologie *small city* est un sous ensemble de la topologie *city* limité à un domaine de $320 \text{ m} \times 320 \text{ m}$ et contenant 500 nœuds.

Chacune de ces topologies est connexe. C'est à dire que quels que soient les nœuds u et v appartenant au réseau, il existe un chemin entre u et v .

FIGURE 3.4 – Topologie *city grid*.

3.6 Simulations

Nous avons mesuré l'impact des topologies grâce au simulateur *WSNET*[83] décrit dans la section 2.3 du chapitre 2. Pour chaque combinaison de topologie et de variante de routage nous mesurons les paramètres suivants :

- *Taux de livraison* : le nombre de messages reçus divisé par le nombre de messages envoyés pour l'ensemble du réseau,
- *Taille moyenne des routes* : le nombre de sauts nécessaires à un message pour être routé de la source à la destination,
- *Coût d'émission des messages* : la somme des coûts pour tous les messages envoyés dans le réseau. Le coût d'un message dépend de la distance entre la source et la destination du message. On définit le coût d'une transmission E_t par $E_t = r^\alpha + C$ où r est la portée de communication en mètres, α et C sont des constantes et dépendent du matériel et du modèle de propagation comme décrit dans [84].

Pour faire varier le degré des nœuds, nous faisons varier leur portée de communication de 25 m à 50 m par étapes de 5 m.

Modèle de trafic de données

Le trafic de données simulé est le suivant : toutes les 10 secondes, chaque nœud choisit aléatoirement un destinataire parmi l'ensemble des nœuds du réseau excepté lui-même. Le nœud émetteur envoie alors un paquet de données ayant un entête de 88 octets et 10 octets de données.

Paramètres de simulation

Chaque combinaison des différentes variantes des algorithmes et les topologies est exécutée 50 fois. La couche *MAC* utilisée est idéale et aucune interférence n'est simulée. Ce choix a été fait pour isoler l'impact des algorithmes de routage uniquement. Par ailleurs les expériences sur **SensLAB**, décrites dans la section expérimentations (3.7), permettent de prendre en compte ces paramètres dans les résultats. Les barres d'erreur représentées sur les courbes de résultats symbolisent un intervalle de confiance de 95%. Le tableau 3.1 résume les paramètres de simulation.

Paramètre	Valeur
α	4
C	2^8
Durée (s)	60
Couche MAC	idealmac
Interférences	none
Taille des données (octets)	10
Taille de l'entête (octets)	88

TABLE 3.1 – Paramètres de simulation pour la mesure de l'impact de la topologie

3.6.1 Taux de livraison

Les figures 3.5, 3.6, 3.7 et 3.8 comparent les différentes topologies étudiées grâce à la métrique de taux de livraison pour les différentes variantes d'algorithmes. Elles montrent le taux de livraison en fonction du degré moyen des nœuds dans le réseau. On observe sur ces figures que pour de faibles degrés, inférieurs à 15, les topologies *city* et *city grid* montrent un écart de performance de 25% pour l'algorithme *NFP* (figure 3.5).

Le taux de livraison pour la topologie *city grid* est très haut, proche de 100% quels que soient la méthode de routage et le degré. Ce comportement s'explique par la quasi absence d'impasse dans cette topologie. En effet, un nœud trouvera presque toujours un voisin en direction de la destination. Le seul contre exemple est lorsqu'un nœud se situe aux limites du réseau, à l'extrémité d'une branche et qu'il souhaite envoyer une donnée à destination d'un nœud, lui aussi à la limite du réseau sur une autre branche. Cette situation est rare car le nombre de nœuds à l'extrémité des branches est faible. La partie droite et la partie basse de la figure 3.4 illustrent ces propos.

On observe aussi sur ces figures que plus le réseau est dense, plus le taux de livraison est élevé. Ces résultats, valables avec des couche MAC et physiques idéales, ne sont pas vrais dans des conditions réelles comme nous le montrerons à la section expérimentations (3.7). Par ailleurs, lorsque la densité de nœuds augmente, les

différences de performances entre les topologies s'amenuisent.

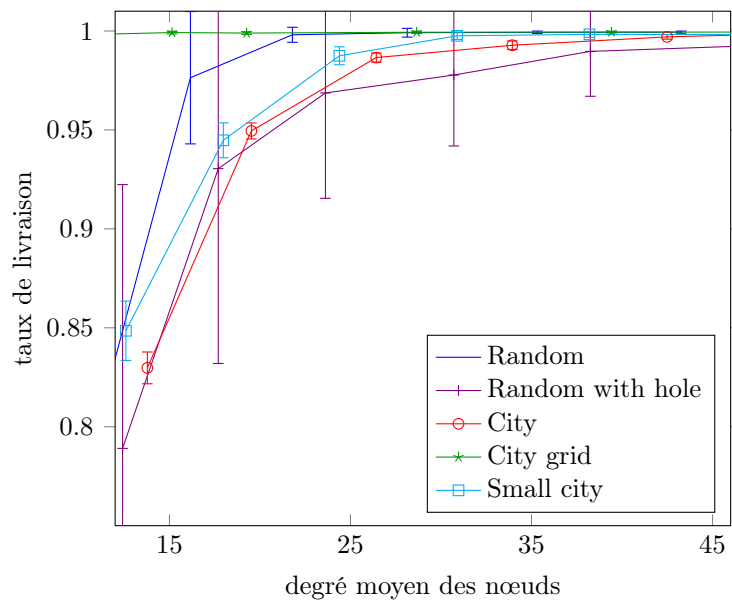


FIGURE 3.5 – Taux de livraison pour *Greedy*.

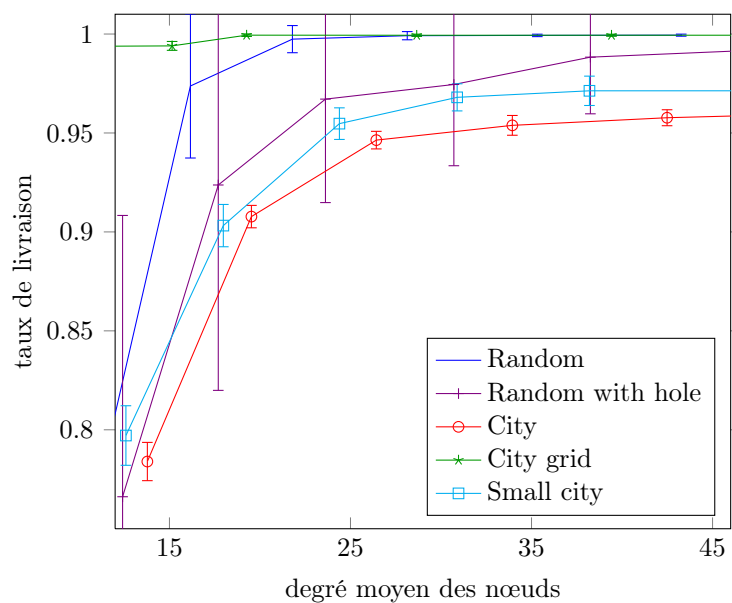
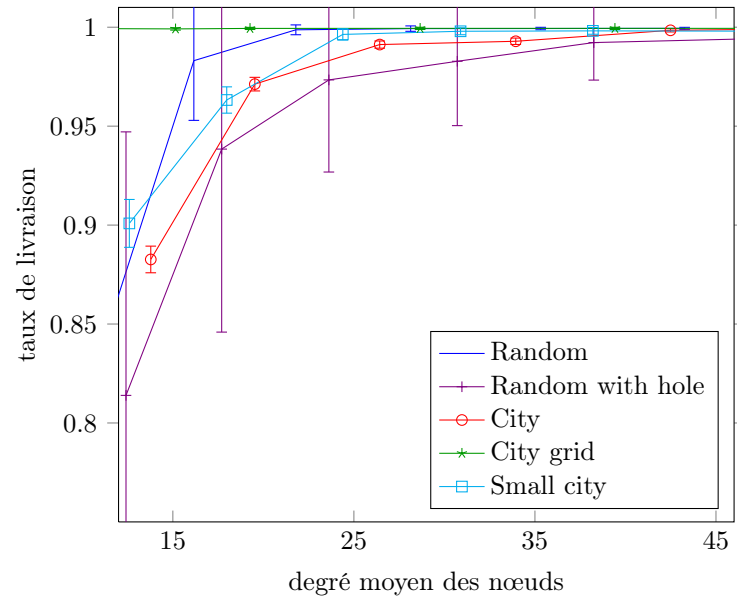
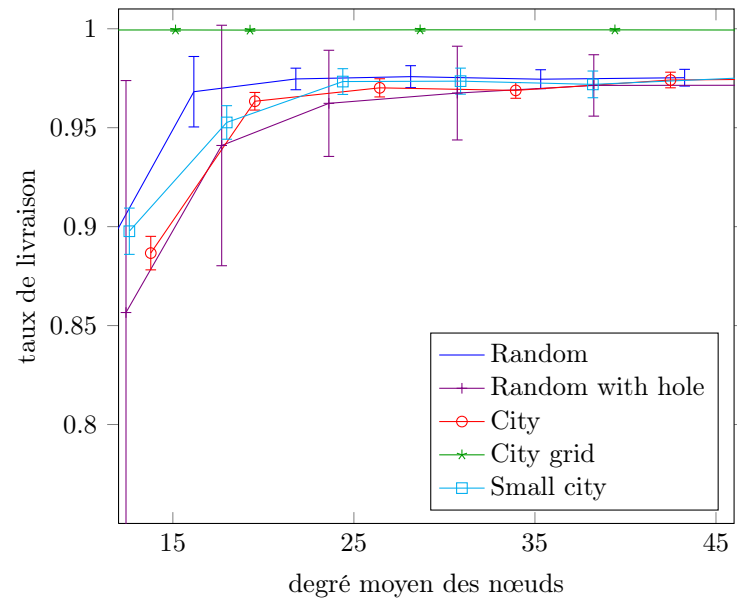


FIGURE 3.6 – Taux de livraison pour *NFP*.

FIGURE 3.7 – Taux de livraison pour *MFR*.FIGURE 3.8 – Taux de livraison pour *Compass*.

3.6.2 Longueur des routes

Les figures 3.9, 3.10, 3.11 et 3.12 représentent la moyenne des longueurs des routes des paquets de données acheminés avec succès. Pour les variantes *Greedy*, *MFR* et *Compass*, toutes les topologies, exceptée *small city*, montrent des résultats similaires. Les routes de la topologie *small city* sont plus courtes car le diamètre du réseau est plus petit.

On remarque que pour *NFP* (figure 3.10) la longueur moyenne des routes augmente alors que pour tous les autres algorithmes elle diminue lorsque le degré augmente. Cela s'explique car *NFP* favorise les nœuds qui sont les plus proches de la source pour aller vers la destination (sans jamais aller en arrière). Lorsque le degré augmente, la probabilité d'avoir un nœud plus proche de la source augmente, le chemin parcouru à chaque saut sera donc plus court et par conséquent, il faudra plus de sauts pour atteindre la destination.

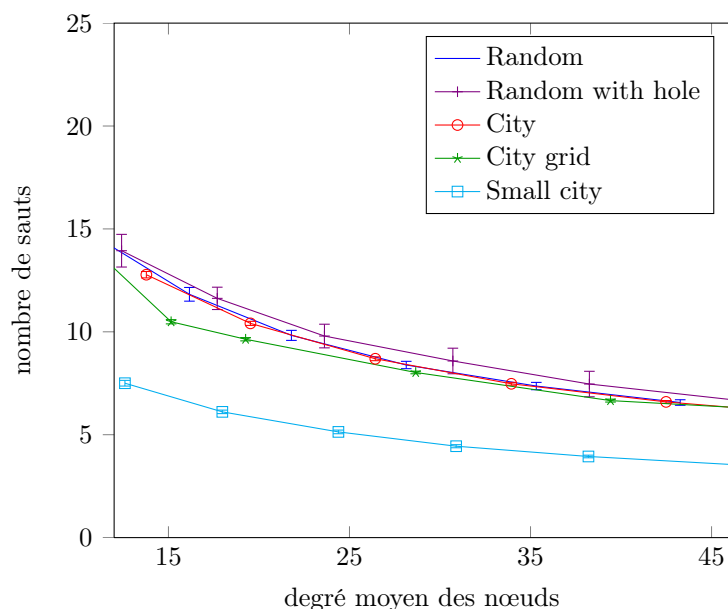
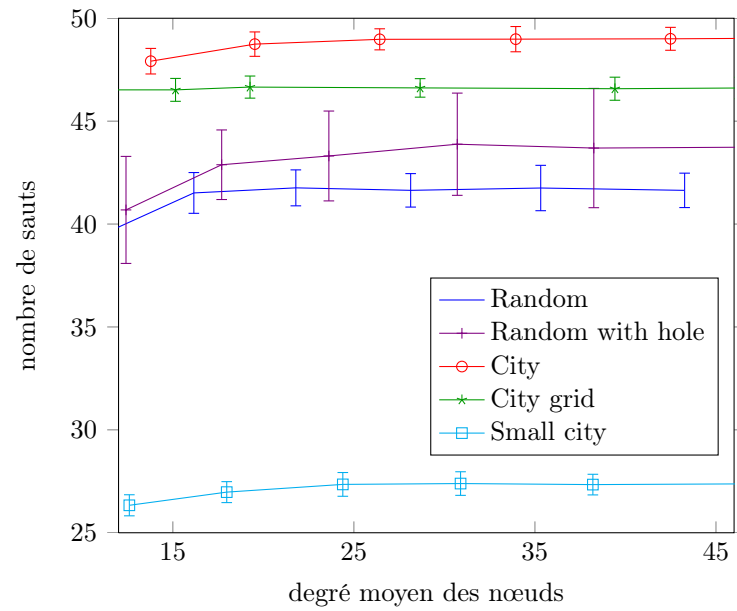
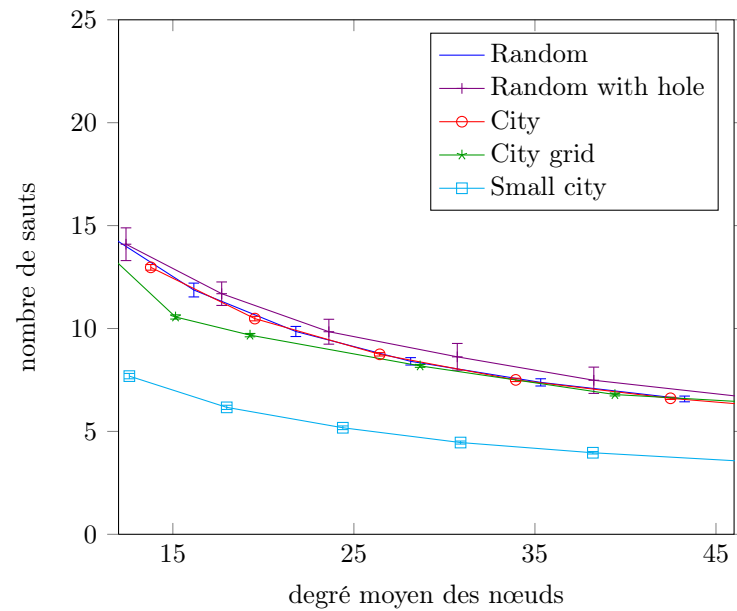
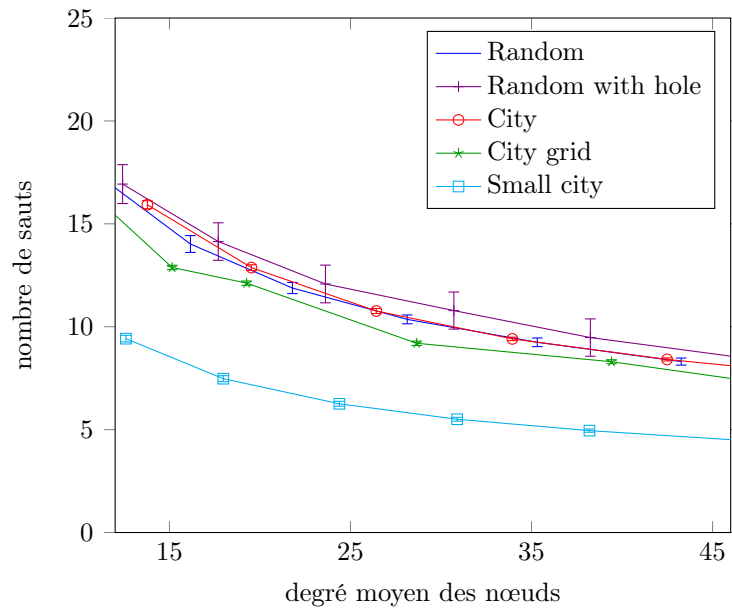


FIGURE 3.9 – Longueur moyenne des routes pour *Greedy*.

FIGURE 3.10 – Longueur moyenne des routes pour *NFP*.FIGURE 3.11 – Longueur moyenne des routes pour *MFR*.

FIGURE 3.12 – Longueur moyenne des routes pour *Compass*.

3.6.3 Coûts d'émission

Les coûts d'émission sont représentés sur les figures 3.13, 3.14, 3.15 et 3.16. En raison de son diamètre plus faible, on observe pour tous les algorithmes étudiés des différences de 150% à 300% entre la topologie *small city* et les autres topologies. Sur la figure 3.14, la variante *NFP* montre des écarts importants entre les différentes topologies concernant les coûts d'émission pouvant aller jusqu'à un facteur de 10. On y voit que le coût pour *city grid* est constant, cela est dû au fait que, sur cette topologie, le voisin le plus proche d'un nœud est toujours à la même distance, quels que soient les nœuds choisis. Pour la topologie *city*, contrairement à *city grid*, la distance entre les différents nœuds n'est pas constante, toutefois les variations de distance sont plus faibles que pour la topologie *random* et *random hole*. C'est pourquoi on observe que l'algorithme *NFP* est moins efficace sur les topologies *city* et *small city* mais malgré tout plus efficace que sur les topologies *random* et *random hole*.

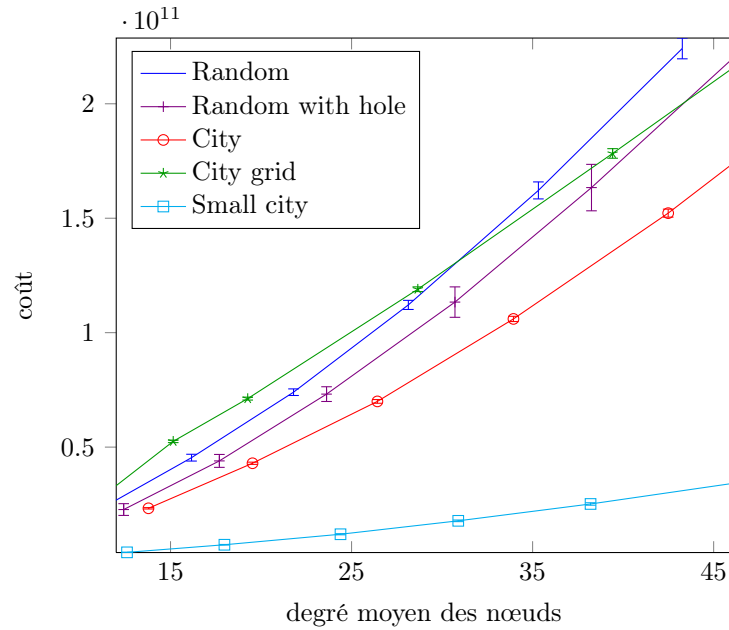
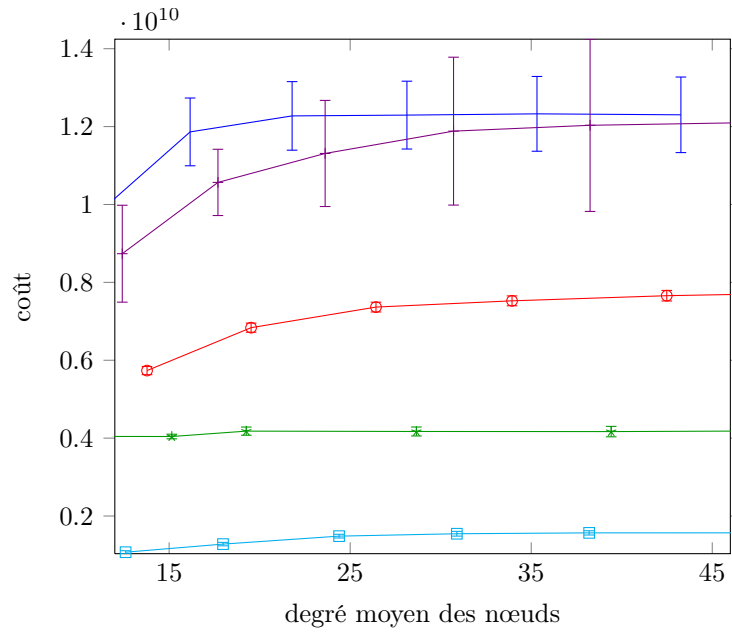
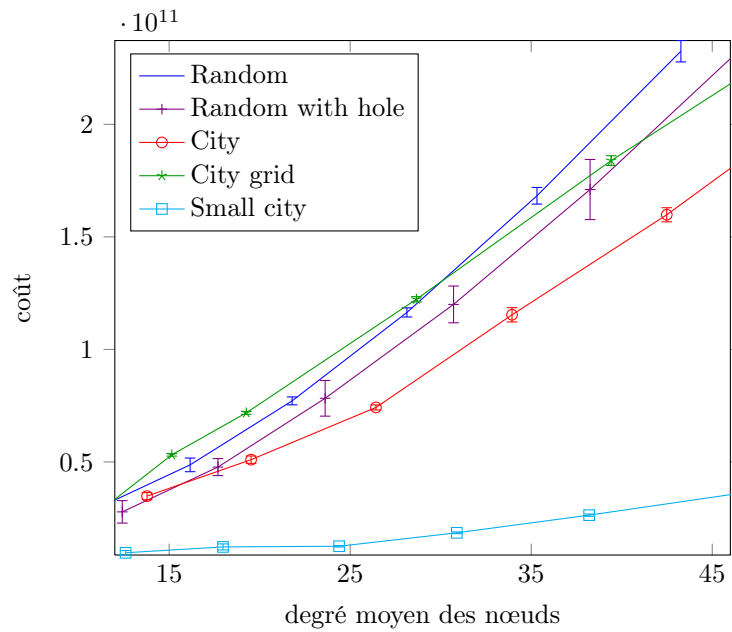
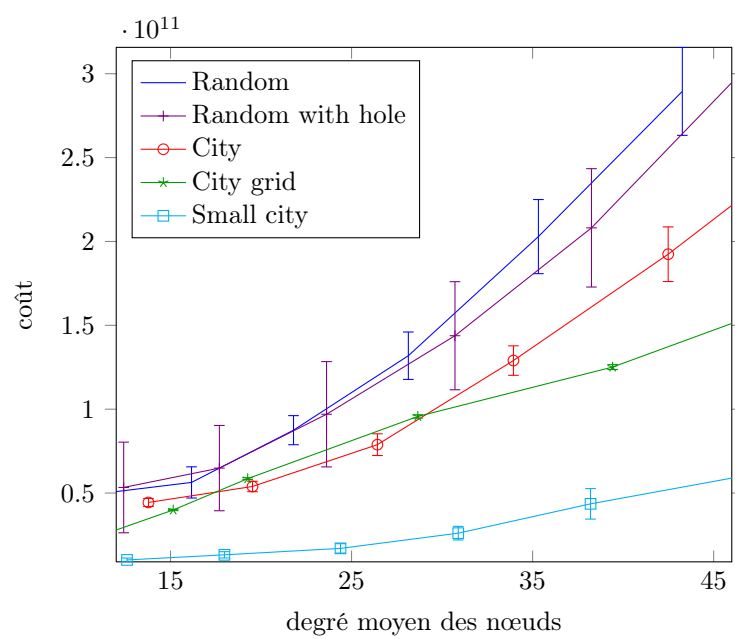


FIGURE 3.13 – Coûts d'émission pour *Greedy*.

FIGURE 3.14 – Coûts d'émission pour *NFP*. (légende identique à la figure 3.13)FIGURE 3.15 – Coûts d'émission pour *MFR*.

FIGURE 3.16 – Coûts d'émission pour *Compass*.

3.7 Expérimentations

Des expériences ont été réalisées sur des capteurs sans-fil grâce à la plate-forme **SensLAB** [63]. La section 2.3 du chapitre 2 donne plus de détails sur cette plate-forme.

Nous avons choisi d'utiliser la plate-forme de Rennes qui offre une topologie plate. Nous utilisons un sous-ensemble des nœuds offerts par la plate-forme pour créer différentes topologies. La topologie initiale de la plate-forme de Rennes est représentée sur la figure 3.17.

La disposition de la plate-forme étant limitée nous n'avons pas conduit d'expériences sur la topologie *random hole* car il n'était pas possible d'obtenir une topologie connexe avec une zone sans nœud suffisamment significative. La topologie *random* est constituée de 60 nœuds choisis aléatoirement parmi les nœuds disponibles. La topologie *city grid* utilise une ligne sur deux et une colonne sur deux de la partie droite de la plate-forme. L'ensemble est constitué de 77 nœuds. La topologie *city* est constituée de 117 nœuds tandis que le sous-ensemble *city small* est constitué de 57 nœuds. Ces topologies tentent de reproduire le schéma d'une ville avec des routes et des intersections formés par les nœuds. La figure 3.18 montre la topologie *city* sur la plate-forme de Rennes.

La connexité des topologies ne peut pas être garantie dans les expériences. En effet l'environnement n'est pas maîtrisé et il n'est pas possible de garantir que deux nœuds communicant à un instant puisse communiquer à un autre instant. Nous avons choisi les nœuds de manière à ce que la topologie générée soit connectée de manière théorique en fonction de la portée de communication mesurée expérimentalement mais elle ne peut être garantie.

Les nœuds de la plate-forme de Rennes sont équipés de la puce de communication radio *cc2420* de *Texas Instrument*. Les couches MAC et physique sont de type 802.15.4. Les expérimentations ont été effectuées sur les quatre variantes de routage, pour toutes les puissances de transmission disponibles, sur chacune des quatre topologies présentée en section 3.4.

Conduire des expériences sur une plate-forme réelle permet de valider les résultats avec l'ensemble de la couche réseau. De plus les phénomènes induits par l'environnement, difficilement simulables, sont ici pris en compte, de même que les couches physique et MAC. Les résultats ainsi obtenus permettent de se situer au plus proche des conditions réelles que l'on pourra rencontrer dans les applications. On notera que la conduite d'expérimentations n'offre pas la souplesse des simulations. En effet le débogage des capteurs n'est pas aussi souple que celui des simulateurs, le déploiement du code est plus contraignant et le temps d'exécution est incompréhensible contrairement aux simulateurs. Pour toutes ces raisons, le nombre d'itérations de chaque expérience est plus faible que pour les simulations. Il est ici ramené à 5 itérations par jeu de paramètres. Du fait du nombre réduit d'itérations nous n'avons pas représenté l'intervalle de confiance sur les courbes des résultats pour les expérimentations car il ne serait pas représentatif.

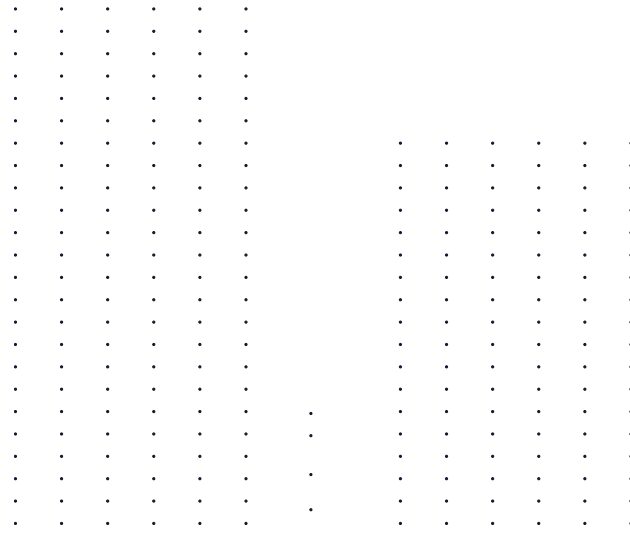


FIGURE 3.17 – Position des nœuds sur la plate-forme de Rennes.

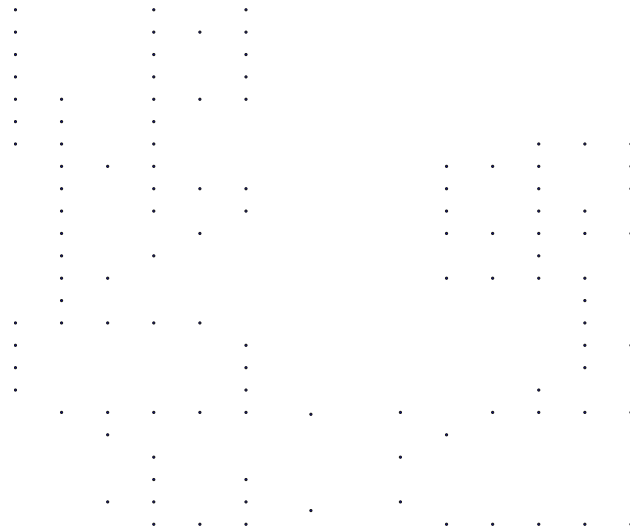


FIGURE 3.18 – Topologie city sur la plate-forme de Rennes.

Les résultats sont visibles sur les figures 3.19, 3.20, 3.21 et 3.22. Alors que l'on a pu observer un taux de livraison strictement croissant en fonction du degré sur les simulations, on peut observer sur les expérimentations des résultats différents. En effet, pour la plupart des résultats expérimentaux, le taux de livraison croît puis décroît en fonction du degré des nœuds. Ce résultat s'explique par une augmentation du nombre de collisions entre les paquets, impliquée par une plus grande portée de communication. L'utilisation d'une couche MAC idéale dans les simulations ne permettait pas de mettre en évidence ce comportement.

En raison du nombre de nœuds et de l'espace limité proposé par **SensLAB**, les différentes topologies testées ne sont pas aussi différentes les unes des autres que dans les simulations. Néanmoins, on observe des variations entre les différentes topologies pour chaque méthode. Sur la figure 3.19, on peut observer une différence entre les topologies *random* et *city* de plus de 20% pour les degrés inférieurs à 10.

Il est par ailleurs intéressant d'observer que les topologies *city small* et *random* atteignent des performances très similaires, le profil des courbes étant lui aussi comparable. Ces deux topologies sont similaires à cause du nombre limité de possibilités offertes par le placement des nœuds sur la plate-forme **SensLAB**.

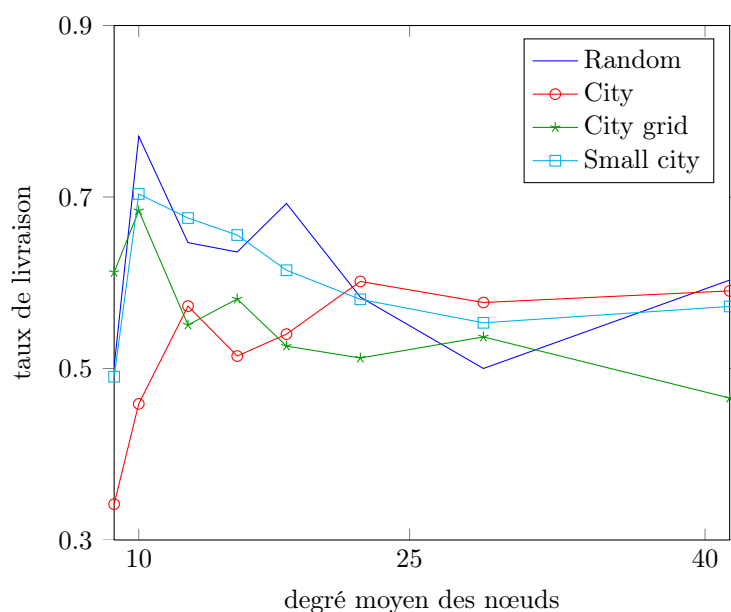
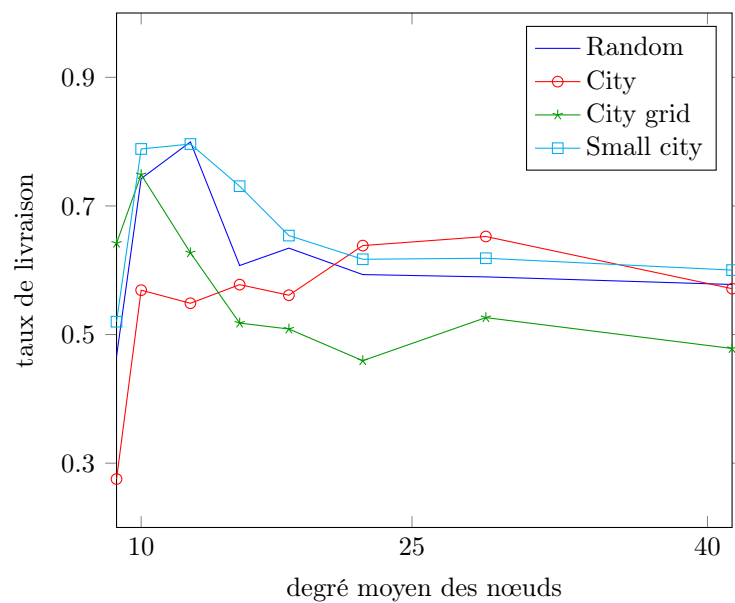
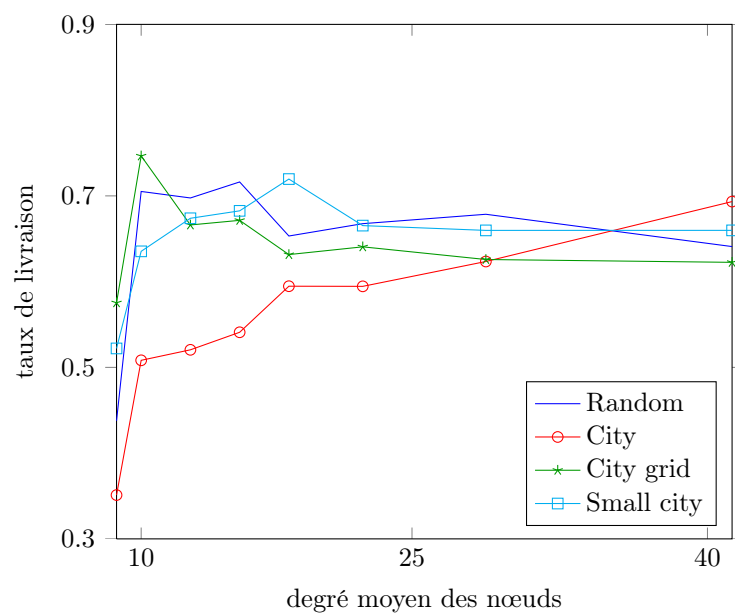
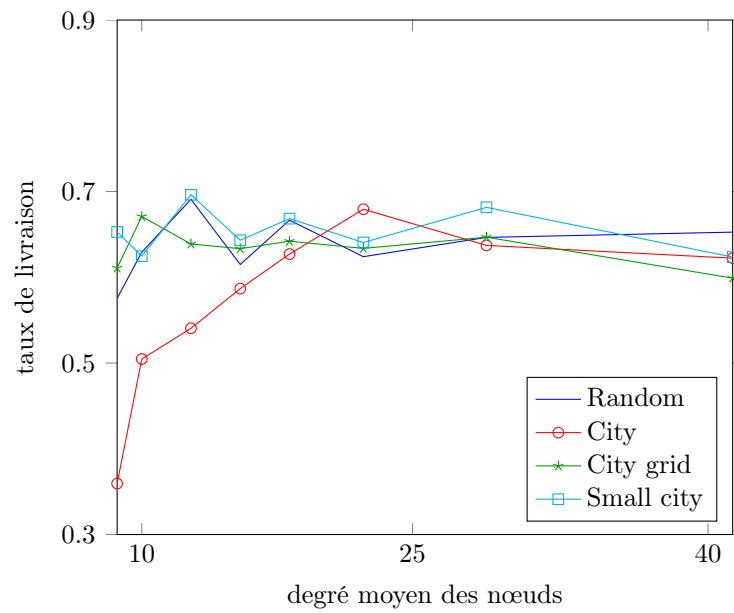


FIGURE 3.19 – Taux de livraison pour *greedy* sur **SensLAB**.

FIGURE 3.20 – Taux de livraison pour *NFP* sur **SensLAB**.FIGURE 3.21 – Taux de livraison pour *MFR* sur **SensLAB**.

FIGURE 3.22 – Taux de livraison pour *COMPASS* sur **SensLAB**.

3.8 Discussion

Dans cette section nous corrélons le délai de livraison avec la longueur des routes étudiée en section 3.6.2. En effet, si l'on considère l'hypothèse raisonnable selon laquelle le temps de traitement d'un paquet sur chaque nœuds est constant et similaires entre les différents algorithmes, de même que le temps de transmission entre deux nœuds, le délai de livraison est alors proportionnel au nombre de sauts. Pour cette raison nous faisons le parallèle entre nombre de sauts et délai de livraison.

Étant donnés les résultats analysés dans les précédentes sections, on peut dire qu'en fonction de la topologie de l'application ciblée et des propriétés désirées (consommation énergétique, taux de livraison), le meilleur algorithme n'est pas toujours le même. Si on considère le délai d'acheminement des paquets comme étant une propriété importante pour notre application, on éliminera l'algorithme *NFP* car la longueur des routes générée par cet algorithme est plus de trois fois plus élevée que pour les autres algorithmes étudiés (fig. 3.23).

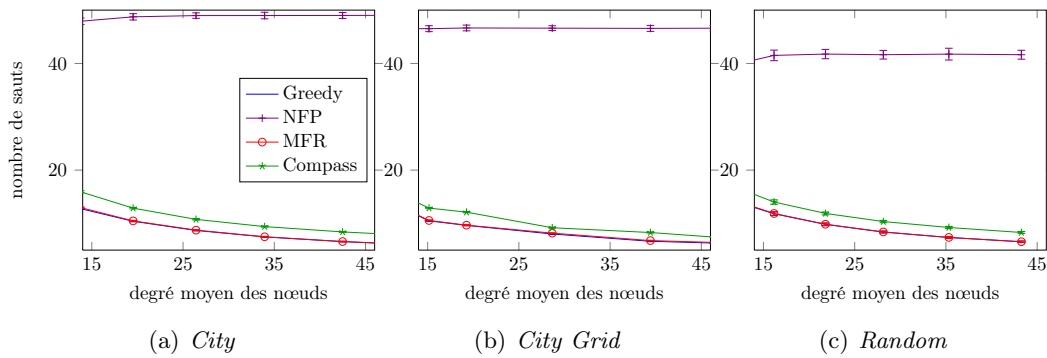


FIGURE 3.23 – Comparaison de la longueur des routes pour trois topologies.

Dans ce contexte, si l'on souhaite maintenant optimiser la consommation énergétique, il faudra considérer la topologie car on observe que le meilleur algorithme (en excluant *NFP*) n'est pas le même selon la topologie utilisée. En effet, pour la topologie *city*, le meilleur choix est *greedy*, par contre, pour la topologie *city grid*, *compass* est alors le meilleur choix. Cette comparaison est illustrée figure 3.24. On peut considérer que les taux de livraison de *greedy* et de *compass* sont similaires pour une même topologie. En effet, pour un degré supérieur à 25, le taux de livraison de *greedy* est de 98% sur la topologie *city* et de 100% sur la topologie *city grid*. Le taux de livraison de *compass* est lui de 96% sur *city* et de 100% sur *city grid*. Il sera plus intéressant d'effectuer d'occasionnelles retransmissions pour favoriser la consommation énergétique.

Lors de la mise au point de leurs algorithmes, les concepteurs choisissent souvent d'utiliser une topologie aléatoire. Ce choix peut être établi pour des raisons de commodité. Il est en effet aisé de générer ce type de topologie dans un simulateur.

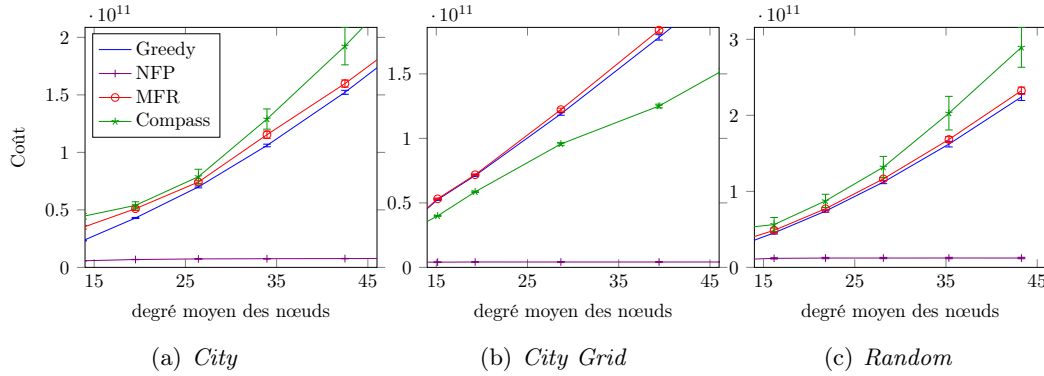


FIGURE 3.24 – Comparaison des coûts énergétiques pour trois topologies.

Ce peut être parce que les concepteurs ne suspectent pas qu'il puisse y avoir de grandes différences de performances avec d'autres topologies. Enfin, ce choix peut être fait parce que l'application ciblée n'est pas clairement définie, les topologies de celle-ci sont donc inconnues. Il est important dans le processus de conception de considérer l'application cible pour guider le développement vers les performances souhaitées et la topologie ciblée.

Nous donnons un exemple concret pour lequel nous devons déployer deux réseaux de capteurs sans-fil dans deux contextes différents. Dans les deux cas nous considérons que le délai de livraison des paquets est critique. Le premier exemple est la surveillance d'une zone critique (volcan, feu de forêt, tremblement de terre ...) pour laquelle les capteurs sont déployés par voie aérienne. Le second exemple est le contrôle des lampadaires sur l'île de Manhattan à New-York où l'on souhaite contrôler l'allumage et l'arrêt des lampadaires en fonction de la présence de passants. Le délai est dans cet exemple important car la sécurité des passants la nuit est liée à l'allumage des lampadaires. Dans le premier exemple la topologie est aléatoire, la position des capteurs lancés depuis les airs n'est pas contrôlée. Dans le deuxième exemple par contre, les capteurs seront espacés d'une distance relativement constante, dans des rues en forme de grille, la topologie sera donc très proche de *city grid*. Si nous regardons les résultats (fig. 3.24(b) et 3.24(c)), l'algorithme *NFP* est le plus efficace en terme de consommation énergétique pour les topologies *random* et *city grid*. Cependant, étant donné le délai non satisfaisant engendré par cet algorithme nous décidons de l'écarter (fig. 3.23(b) et 3.23(c)). Dans ce contexte on observe que *Compass* est l'algorithme le plus efficace sur la topologie *city grid* mais, sur la topologie *random* c'est l'algorithme *greedy* qui est le plus efficace (fig. 3.24(b) et 3.24(c)). On choisira donc l'algorithme *greedy* pour la surveillance de zones critiques tandis que l'on choisira plutôt *compass* pour le contrôle des lampadaires à Manhattan.

3.9 Conclusion

Nous avons vu dans ce chapitre que l'impact de la topologie ne devrait pas être négligé dans les réseaux de capteurs sans-fil. En effet nous avons mis en évidence des différences de performance pouvant aller jusqu'à 25% concernant le taux de livraison et la taille des routes et des différences pouvant aller jusqu'à 100% pour le coût énergétique total. Au delà de l'illustration sur des algorithmes de routage géographiques, ces résultats peuvent être élargis à la conception d'algorithmes pour réseaux de capteurs sans-fil en général.

Ces travaux montrent que la topologie choisie a un impact significatif sur les performances des réseaux de capteurs sans-fil. La conception d'algorithmes efficaces pour réseaux de capteurs sans-fil devrait toujours considérer la topologie et l'application ciblée.

Dans *BinThatThinks*, le réseau est amené à être déployé dans des villes et des villages. Les topologies pourront être de natures différentes et, au vu des résultats des travaux de ce chapitre, il est important de trouver des solutions adaptées aux différentes topologies rencontrées.

Clustering pour l'optimisation de la durée de vie des réseaux de capteurs sans-fil

Sommaire

4.1	Introduction	61
4.2	Motivations et énoncé du problème	63
4.3	Notations et préliminaires	63
4.4	État de l'art	66
4.5	Contribution	70
4.6	Résultats	74
4.7	Conclusion	81

4.1 Introduction

Le *clustering* dans les réseaux de capteurs sans-fil est un moyen efficace de structurer et d'organiser le réseau. L'organisation en *clusters* consiste à identifier des sous-ensembles de nœuds dans le réseau et à leur attribuer un *chef* (un *cluster-head*). Ce *chef* devient alors en charge de tâches spécifiques comme le regroupement de données en provenance des autres nœuds du *cluster*. Le *cluster-head* peut ensuite transmettre ces données à un puits grâce à un moyen de communication longue distance. Ce mode de fonctionnement induit un épuisement plus rapide de l'énergie du *cluster-head* par rapport aux nœuds ordinaires.

Dans ce chapitre, nous proposons un nouvel algorithme de *clustering* pour les réseaux de capteurs sans-fil adapté au contexte de l'application du projet *BinThatThinks* [85]. Comme nous l'avons vu au chapitre précédent, il est important de cibler l'application et ses topologies potentielles afin d'obtenir les performances les plus réalistes possibles. Dans cette application, les *cluster-heads* collectent les données depuis les nœuds-capteurs appartenant à leur *cluster* en utilisant une liaison point à point basse consommation telle que IEEE 802.15.4TM. Les *cluster-heads* transmettent ensuite les données recueillies à un puits au moyen d'une liaison GPRS. Les communications GPRS sont beaucoup plus énergivores que les communications point

à point, c'est pour cette raison que les nœuds ordinaires n'utilisent pas la liaison GPRS. De même, le *cluster-head*, qui utilise à la fois la liaison GPRS et la liaison point à point consommera bien plus d'énergie qu'un nœud ordinaire. Pour cette raison, les nœuds devraient être *cluster-head* à tour de rôle afin de permettre au réseau de fonctionner aussi longtemps que possible avec tous les nœuds opérationnels. L'organisation en *clusters* devrait se faire en limitant l'augmentation des communications et les modifications de structure des *clusters*. L'algorithme de *clustering* devrait aussi fournir un bon compromis entre le nombre de *clusters* (plus il y a de *clusters*, plus il y a de nœuds avec GPRS actif) et le diamètre des *clusters* (plus le diamètre est grand, plus il y a de communications intra-cluster). Le réseau devrait avoir une durée de vie aussi grande que possible. Nous définissons la durée de vie comme la durée pendant laquelle tous les nœuds du réseau sont opérationnels, et donc, ont encore de l'énergie. Pour cela, la quantité d'énergie devrait être équilibrée entre les différents nœuds, de façon à ce qu'ils épuisent leur batterie au même moment. Dans ce chapitre, nous répondons à cette problématique en présentant **BLAC**¹, une nouvelle famille d'algorithmes de *clustering* adaptée à l'équilibre énergétique.

À notre connaissance, **BLAC** est la première famille d'algorithmes de *clustering* distribués qui fournit des *clusters* non-recouvrants, multi-sauts ayant des préoccupations énergétiques. Les solutions de la littérature ne considèrent pas la durée de vie du réseau comme le temps jusqu'à épuisement du premier nœud. Il en résulte que ces solutions offrent une durée de vie jusqu'à épuisement de tous les nœuds plus longue, mais le temps jusqu'à épuisement du premier nœud est réduit. Par ailleurs, les algorithmes de la littérature ne considèrent pas l'équilibre énergétique entre les différents nœuds comme un élément important de la création des *clusters*. D'autres solutions de *clustering* ont des caractéristiques non adaptées à notre application comme des *clusters* recouvrant ou à 1-saut. **BLAC** considère le niveau de batterie des nœuds combiné à une autre métrique (la *densité* ou le *degré*) pour élire un *cluster-head*. Les nœuds changent alors naturellement de *cluster-head* en fonction du niveau de batterie à une fréquence limitée afin de fournir de la stabilité à la structure. Les changements de *cluster-head* sont liés au niveau de batterie et à la topologie du réseau. Ainsi ils n'ont lieu que si les nœuds échangent des données (ce qui consomme de l'énergie) où s'il y a des changements dans la topologie (mobilité des nœuds, nœuds hors-services, interférences ...). De cette manière la fréquence de changement des *cluster-heads* s'adapte totalement à l'activité du réseau. **BLAC** se décline en quatre variantes, **BLAC-bg** combine le niveau de batterie avec le *degré*, **BLAC-bs** combine le niveau de batterie avec la *densité*. Les variantes **BLAC-rg** et **BLAC-rs** fonctionnent en deux phases. Une planarisation du graphe est d'abord appliquée avant la phase de construction des *clusters*. Comme nous le verrons dans les sections suivantes, chacune de ces variantes présente des propriétés spécifiques qui les rendent adaptées à différentes situations. Les résultats de simulation montrent

1. **B**attery-**L**evel **A**ware **C**lustering

que **BLAC** étend la durée avec tous les nœuds en vie jusqu'à 300% en comparaison des solutions de la littérature.

4.2 Motivations et énoncé du problème

Dans le cadre du projet *BinThatThinks*, décrit au chapitre introduction (section 1.4), dans la seconde hypothèse envisagée, nous souhaitons mettre en place un réseau de capteurs dont les nœuds sont équipés d'un module radio point à point et d'un autre module pour des communications longue distance. Une des hypothèses du projet consiste en effet à équiper les poubelles avec :

- Un module sans-fil basse consommation *IEEE* 802.15.4TM pour les communications pair à pair entre les nœuds.
- Un module sans-fil GPRS pour les communication longue distance avec la station de base.

L'équipement de chaque nœud de GPRS peut être coûteux en matériel et abonnements, toutefois il existe des abonnements spécifiques pour les communications *machines-to-machines*. Étant donné que le GPRS consomme beaucoup plus d'énergie que la liaison point à point, il n'est pas judicieux de l'utiliser continuellement, il est donc intéressant d'étudier des méthodes pour réduire l'utilisation du module GPRS lorsque c'est possible.

Dans *BinThatThinks*, de part leur nature, tous les nœuds du réseau envoient des données importantes. Il n'est donc pas souhaitable qu'un nœud épuise sa batterie avant les autres et qu'il ne soit plus en mesure de transmettre ses informations à la station de base. Il est ainsi préférable de maximiser la durée pendant laquelle l'ensemble des nœuds est actif, c'est pourquoi nous utilisons cette notion pour la durée de vie du réseau. Pour atteindre cet objectif il est nécessaire d'équilibrer la consommation énergétique des différents nœuds afin que l'un d'eux ne soit pas hors service avant les autres, compromettant l'ensemble du réseau. **BLAC** propose un compromis, entre la taille et le nombre de *clusters*, efficace d'un point de vue énergétique. De plus il apporte l'équilibre énergétique nécessaire à l'application ciblée.

Afin d'apporter les réponses les plus adaptées au projet, nous étudions le comportement des algorithmes que nous proposons sur des topologies de villes, plus ou moins denses, dans lesquelles *BinThatThinks* pourrait être déployé. Par ailleurs, nous utilisons des valeurs de consommation énergétique réalistes, issues de composants du commerce, qui pourraient être utilisés dans le cadre du projet (voir 4.6.1).

4.3 Notations et préliminaires

La famille d'algorithmes **BLAC** s'appuie sur plusieurs travaux issus de la littérature. Par soucis de cohérence et de clarté, cette section introduit les travaux sur lesquels **BLAC** repose ainsi que les notations qui y sont associées.

Dans un arbre on note $P(u)$ le père du nœud u . On note $\mathcal{H}(u)$ le *cluster-head* du *cluster* auquel u appartient.

4.3.1 Métrique de densité

Dans l'article de Mitton et al. [86], les *clusters* sont construits comme suit : chaque nœud u élit comme père le nœud dans son voisinage ayant la plus forte densité. Si u possède la plus forte densité dans son voisinage, il devient alors *cluster-head*. Les égalités sont levées en choisissant le nœud dont l' Id est le plus petit. Si un nœud v est voisin de deux *cluster-heads*, il envoie un message d'avertissement et le *cluster-head* avec la plus faible densité choisit v comme père. De cette façon on assure un minimum de 3 sauts entre les différents *cluster-heads*. La densité $\rho(u)$ est définie comme le rapport entre le nombre de liens entre et avec les voisins de u sur le degré de u :

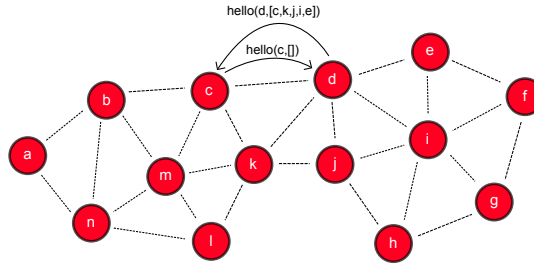
$$\rho(u) = \frac{|\{(v, w) \in E \mid v \in \{u, \mathcal{N}(u)\}, w \in \mathcal{N}(u)\}|}{\delta(u)}$$

La figure 4.1(b) montre un exemple de réseau sur lequel on calcule la densité des nœuds. Le nœud a par exemple, a un degré $\delta(a)$ égal à 2 et le nombre de liens entre les voisins de a est 1. Ainsi $\rho(a) = \frac{2+1}{2} = 1.5$.

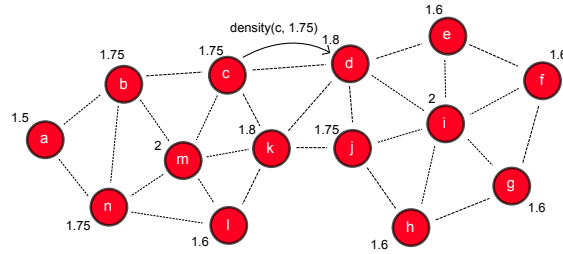
4.3.2 Réduction de topologie

Deux variantes de **BLAC** (**BLAC-rg** et **BLAC-rs**) fonctionnent en deux étapes. La première étape consiste en une réduction de graphe prenant en compte le niveau de batterie des nœuds introduite par Radak et al. [87]. La réduction de graphe est exécutée en construisant le *RNG* (**R**elative **N**eighborhood **G**raph) [88]. La construction d'un *RNG* consiste à supprimer virtuellement un côté de chaque triangle du graphe G , celui dont la métrique calculée est la plus faible. Il est à noter que le *RNG* conserve la connexité. Radak et al. [87] introduisent une nouvelle métrique appelée le *Power factor*. Le *Power factor* d'un lien est déterminé en fonction du niveau de batterie des nœuds formant le lien. Si les deux nœuds ont un niveau d'énergie suffisant le *Power factor* est 0. Si l'un des deux nœuds a un niveau de batterie critique alors le *Power factor* vaut 1. Enfin, si les deux nœuds ont un niveau de batterie critique, le *Power factor* vaut 2. Un seuil est défini pour décider si le niveau d'énergie d'un nœud est suffisant ou critique. Ce seuil doit être défini lors du paramétrage des nœuds. Des métriques habituellement utilisées pour la construction du **RNG** sont utilisées pour rompre les égalités, il s'agit du *RSSI* (**R**eceived **S**ignal **S**trength **I**ndicator) et de l' Id des nœuds. Cet algorithme de réduction de graphe est appelé *bl-RNG* pour **B**attery-**L**evel aware **RNG**.

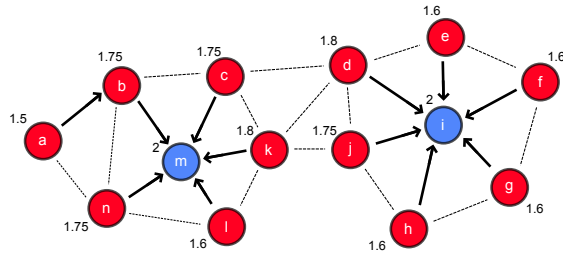
L'algorithme 1 décrit comment est calculé le *RNG* tandis que la figure 4.2 illustre sa création sur un exemple. On note sur cet exemple que la plupart des liens entre deux nœuds dont le niveau d'énergie est critique sont inutilisés après application du *RNG*. Par exemple, le lien entre les nœuds n et m a été écarté, de même que le lien



(a) Chaque nœud envoie un message *hello* à ses voisins contenant son *Id*. Le message *hello* suivant contient la liste des voisins, ainsi chaque nœud connaît son voisinage à deux sauts.



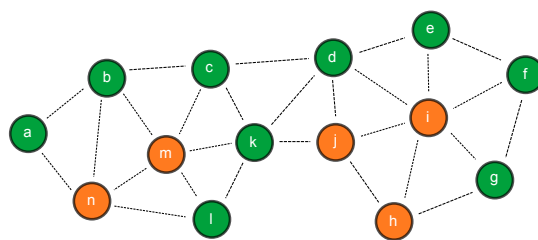
(b) Les nœuds peuvent calculer leur *densité* grâce aux informations qu'ils ont collectées avec les messages *hello*, ils envoient ensuite leur densité dans les prochains messages *hello*.



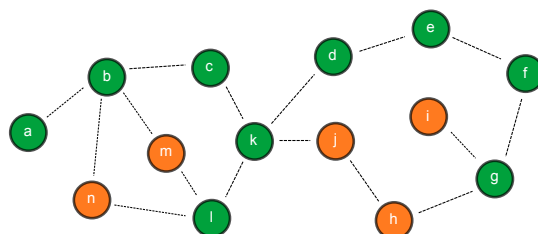
(c) Finalement les nœuds choisissent comme père le nœud dont la *densité* est la plus élevée dans leur voisinage. Les nœuds ayant la plus grande *densité* dans leur voisinage deviennent *cluster-heads*.

FIGURE 4.1 – Création des *clusters*.

entre *i* et *j*. Le lien entre *j* et *h* est conservé car il n'existe pas de lien entre *j* et un voisin de *h* ayant un meilleur *Power factor* qu'entre *j* et *h*. Les nœuds dont le niveau de batterie est élevé sont préférés pour les communications.



(a) Les nœuds dont le niveau de batterie est critique sont représentés en orange, ceux avec un niveau de batterie suffisant sont représentés en vert.



(b) Un **bl-RNG** est appliqué, supprimant les liens entre les nœuds de faible énergie.

FIGURE 4.2 – Construction du **RNG**.

4.4 État de l'art

Les chercheurs ont proposé différentes techniques pour la formation des *clusters* et la sélection des *cluster-heads*. Chacune des solutions permet d'identifier un sous-ensemble de nœuds dans le réseau pour lui attribuer un leader (un *cluster-head*) mais le but recherché est souvent différent du nôtre.

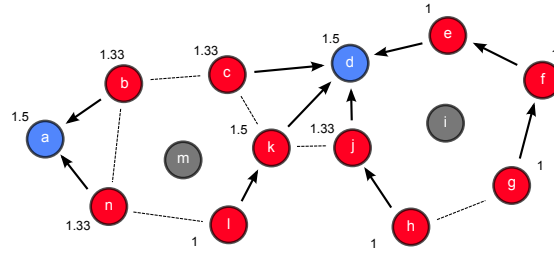
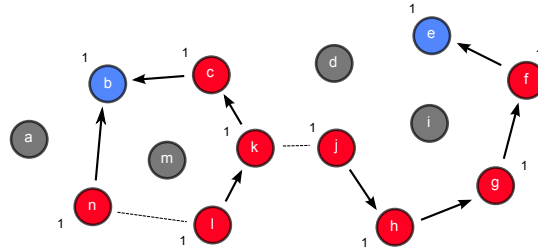
Les premières solutions, comme *LCA* [89], proposé par Baker et Ephremides, et *HCC* [90], par Gerla et Tsai, offrent une structure de *clustering* similaire, les différences se trouvant dans la métrique utilisée. Chaque nœud choisit un père dans son voisinage tel que la métrique de celui-ci soit la plus élevée parmi tous ses voisins. Si un nœud a la métrique la plus élevée dans son voisinage, il devient *cluster-head*. *LCA* utilise l'identifiant (*Id*) unique du nœud comme métrique tandis que *HCC* utilise le degré puis l'*Id* pour lever les égalités. *LCA* et *HCC* créent des *clusters* à un saut, recouvrants, leur taille est donc limitée. Un *cluster* est constitué d'un *cluster-head* et de tous ses voisins à un saut. Deux *cluster-heads* ne peuvent pas être voisins et les nœuds ordinaires appartiennent au moins à un *cluster*. Comme les *clusters* sont recouvrants (un nœud peut appartenir à plusieurs *clusters*), ils sont sujets aux réactions en chaîne en cas de changement. De plus, ils ne sont pas efficaces sur le plan énergétique car cette donnée n'est pas prise en compte lors de la construction des *clusters*. De tels algorithmes ne sont donc pas adaptés aux grands

Algorithme 1: Calcule le *RNG* pour le nœud u .

```

1 begin
2    $N_{RNG}(u) \leftarrow N(u)$  ;
3   CalculatePowerFactor( $u$ ) ;
4   {Le nœud  $u$  calcule le Power factor de chacun des liens de son voisinage.}
5   foreach  $v, w \in N(u)$  do
6     if  $PF(uv) < PF(vw) < PF(uw) \vee PF(vw) < PF(uv) < PF(uw)$  then
7        $N_{RNG}(u) \leftarrow N_{RNG}(u) \setminus \{w\}$  ;
8       Le lien  $uw$  est supprimé du RNG.
9     else
10      if  $PF(uw) < PF(vw) < PF(uv) \vee PF(vw) < PF(uw) < PF(uv)$  then
11         $N_{RNG}(u) \leftarrow N_{RNG}(u) \setminus \{v\}$  ;
12        {Le lien  $uv$  est supprimé du RNG.}
13      else
14        {Le lien  $vw$  est supprimé du RNG.}
15  Return  $N_{RNG}(u)$  ;

```

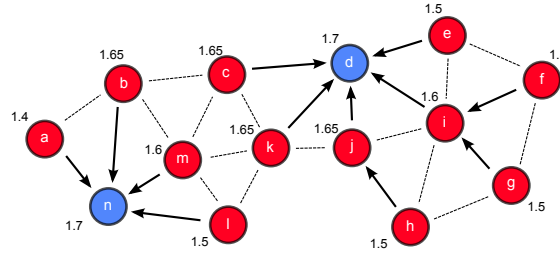
(a) Les *cluster-heads* épuisent leur énergie plus rapidement que les autres nœuds, ce qui crée des trous dans le réseau.

(b) Après quelque temps, d'autres nœuds sont hors service.

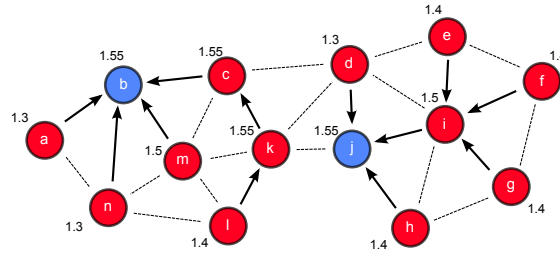
FIGURE 4.3 – Épuisement de la batterie : cas du *clustering* densité.

réseaux dynamiques.

WCA [91] considère de multiples paramètres comme la position, la mobilité et l'énergie pour la construction de ses *clusters*. Cet algorithme est centralisé, ce qui le rend difficilement évolutif pour passer à l'échelle. Il calcule une somme pondérée des



(a) Le réseau est initialisé, les *cluster-heads* consomment plus d'énergie que les nœuds ordinaires.



(b) Après quelque temps les *cluster-heads* changent, les niveaux d'énergie s'équilibrent.

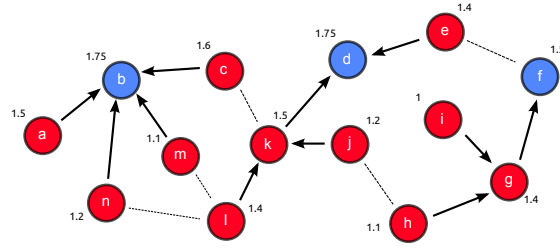
FIGURE 4.4 – Épuisement de la batterie : cas de **BLAC-b***.

paramètres cités ci-dessus et utilise cette métrique pour élire les *cluster-heads*. *WCA* est l'un des premiers algorithmes de *clustering* ayant des considérations énergétiques. Toutefois, il crée des *clusters* à un saut, ce qui engendre plus de *cluster-heads* et donc, à cause de leur puce GPRS active, augmente la consommation énergétique globale du réseau.

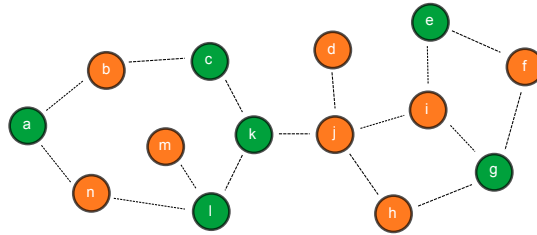
Low et al. [92] proposent une solution de *clustering* pour minimiser les risques de saturation de bande passante des passerelles (*cluster-heads*). Dali et Chan [93] décrivent une solution permettant d'équilibrer l'énergie des différents *clusters* du réseau et de minimiser la consommation énergétique au sein de chaque *cluster* en positionnant le *cluster-head* proche du centre du *cluster*. Cette proposition permet des performances énergétiques intéressantes mais elle est centralisée et ne permet donc pas de passer à l'échelle.

D'autres se concentrent sur des analyses avant déploiement [94, 95] et donc ne peuvent pas passer à l'échelle. En effet, ces propositions supposent un déploiement manuel, ou tout au moins contrôlé, ce qui rend le passage à l'échelle difficile et limite fortement les capacités d'extension d'un tel réseau.

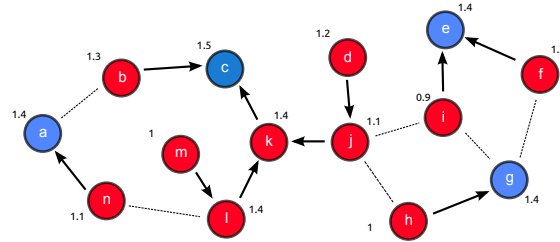
Peu ont des considérations énergétiques [94, 96, 97, 98, 99, 100, 101, 102, 103], lorsque c'est le cas, elles nécessitent des caractéristiques qui ne sont pas toujours disponibles sur les nœuds (comme la position GPS). Elles requièrent parfois des échanges de données nombreuses ou un puits déclaré [100].



(a) Le réseau est initialisé, les *cluster-heads* consomment plus d'énergie que les nœuds ordinaires.



(b) Après quelque temps, les *Power factor* évoluent, les liens du *RNG* aussi.



(c) Basés sur ces nouveaux liens, de nouveaux *cluster-heads* sont élus, permettant d'équilibrer les niveaux d'énergie.

FIGURE 4.5 – Épuisement de la batterie : cas de **BLAC-r***.

L'algorithme de *clustering LEACH* [19] considère l'énergie à la base de sa conception. Chaque nœud devient *cluster-head* en fonction d'une probabilité p et si il a déjà été *cluster-head* précédemment. Le paramètre p définit le ratio de *cluster-head* dans le réseau. Il est possible dans *LEACH* qu'un nœud ayant un faible niveau d'énergie soit élu *cluster-head*. Ce nœud épuisera alors rapidement l'énergie qui lui reste, le rendant hors service. La durée de vie du réseau est alors diminuée, et si le réseau est toujours fonctionnel, une nouvelle sélection de *cluster-head* sera nécessaire pour les nœuds devenus orphelins. De plus, le paramètre p doit être réglé hors ligne, ce qui ne permet pas une adaptation dynamique aux changements dans le réseau. Lorsque les *cluster-heads* se sont fait connaître dans leur voisinage, les nœuds s'attachent

au *cluster-head* qui nécessite le moins d'énergie pour être atteint, formant ainsi des *clusters* à un saut. Ensuite, chaque *cluster-head* synchronise les nœuds de son *cluster* et leur assigne un intervalle de temps pendant lequel le nœud est autorisé à émettre ses données. En raison de la manière dont les *cluster-heads* sont élus, il est possible qu'un nœud n'ait pas de *cluster-head* à portée de communication, cette probabilité augmente lorsque la densité du réseau diminue ou que le paramètre p est trop petit. Cela entraîne des pertes de messages et donc une baisse du taux de livraison.

Dans ce chapitre, nous nous concentrons sur les *clusters* multi-sauts, distribués, non recouvrants et efficaces en énergie. Ce sont les propriétés qui répondent le mieux aux besoins de notre application dans le contexte du projet *BinThatThinks*. Les *clusters* n'ont pas de taille prédéfinie afin de s'adapter au mieux à la topologie sous-jacente et pour être robuste aux petits changements dans la topologie. Les algorithmes que nous proposons sont dérivés de *DDR* [18] et de *Density-based* [86]. Il s'agit d'algorithmes distribués proposant des *clusters* multi-sauts non recouvrants. Leurs algorithmes sont proches, ils diffèrent essentiellement dans la métrique utilisée. *DDR* [18] utilise le degré tandis que *Density-based* [86] introduit une nouvelle métrique appelée *densité*. Plus de détails ont été donnés sur cet algorithme et la métrique de *densité* en section 4.3.1. Bien que ces algorithmes proposent des propriétés intéressantes en ce qui concerne la taille des *clusters* et leur capacité à s'adapter à différentes topologies et à leur changement, les deux propositions ne considèrent pas la consommation énergétique des nœuds. **BLAC** est ainsi basé sur un algorithme similaire mais diffère dans la métrique utilisée. La métrique que nous utilisons permet de rendre l'organisation des *clusters* dynamique vis-à-vis de l'évolution de la consommation énergétique de chaque nœud. Cela permet de maintenir l'ensemble du réseau disponible le plus longtemps possible.

Contrairement aux solutions de la littérature, **BLAC** construit des *clusters* dynamiques, efficaces en énergie, multi-sauts, de manière distribuée. Son objectif principal est d'étendre la durée du vie du réseau² dans le contexte des applications définies dans la section 4.2.

4.5 Contribution

Pour maximiser la durée de vie des réseaux organisés en *clusters* nous proposons une nouvelle famille d'algorithmes : **BLAC**. Quatre variantes sont proposées, chacune étant distribuée et locale. Les algorithmes proposés sont distribués de manière à pouvoir être utilisés sur des réseaux de très grande taille, comme dans les *villes intelligentes*. L'objectif des algorithmes **BLAC** est de conserver la totalité des nœuds en état de fonctionnement aussi longtemps que possible. Le rôle de *cluster-head* est assuré par chaque nœud à tour de rôle afin d'équilibrer l'énergie consommée sur l'ensemble des nœuds sans pour autant nécessiter de grosses modifications ou de

2. où la durée de vie est considérée comme la durée pendant laquelle l'ensemble des nœuds est en service

messages de contrôle supplémentaires. Les différentes variantes de **BLAC** utilisent un algorithme similaire à *Density-based* [86] avec une métrique différente.

L'idée principale de **BLAC** est d'utiliser le *niveau de batterie* dans la métrique, associée à un autre paramètre comme le *degré* ou la *densité*. Pour cela on définit $B(u)$ pour le nœud u tel que :

$$B(u) = \lfloor \frac{batt(u) \cdot 10}{battcap} \rfloor$$

où *battcap* est la capacité initiale de la batterie d'un nœud (la même pour chaque nœud) et *batt(u)* est le niveau de batterie actuel du nœud u . $B(u)$ est un entier variant de 0 à 10 qui diminue moins fréquemment que le niveau réel de la batterie, évitant ainsi des changements intempestifs dans la métrique et donc des modifications dans le *clustering*.

BLAC est décliné en quatre variantes : **BLAC-bg** et **BLAC-bs** d'une part (détaillés dans la section 4.5.1), **BLAC-rg** et **BLAC-rs** d'autre part (détaillés dans la section 4.5.2).

4.5.1 Algorithmes BLAC-b*

BLAC-bg et **BLAC-bs** appliquent le même algorithme mais différent dans la métrique utilisée. Nous détaillons d'abord les métriques utilisées.

4.5.1.1 Degré et énergie

La première variante est appelée **BLAC-bg** et signifie **B**attery-**L**evel **A**ware **C**lustering - **B**attery de**G**ree. Cette variante associe le niveau de batterie avec le degré des nœuds. L'algorithme 2 est exécuté sur chaque nœud avec $h(u) = \delta(u)$. Le degré est une métrique qui nécessite uniquement la connaissance du voisinage à un saut, le niveau de batterie étant local, cette variante ne nécessite qu'une connaissance à un saut et se stabilise donc rapidement. Cependant, la structure du *clustering* ainsi formée est sensible à l'apparition et la disparition de nœuds puisque chaque changement a un impact direct sur le degré (démonstré par Mitton et al. [86]).

4.5.1.2 Densité et énergie

La seconde variante est appelée **BLAC-bs** et signifie **B**attery **L**evel **A**ware **C**lustering - **B**attery den**S**ity. Elle utilise cette fois la densité $\rho(u)$ décrite en section 4.3.1. L'algorithme 2 est exécuté avec $h(u) = \rho(u)$. Cette variante calcule la structure du *clustering* grâce à la connaissance du voisinage à deux sauts. La stabilité du *clustering* est améliorée (au détriment d'une stabilisation légèrement plus lente) car un simple nœud a moins d'impact sur son voisinage (démonstré par Mitton et al. [104]).

4.5.1.3 BLAC-b*

Une fois la métrique calculée et celles de ses voisins obtenues, chaque nœud exécute l'algorithme 2 avec $h(u)$ étant, soit le degré ($\delta(u)$) pour **BLAC-bg**, soit la densité ($\rho(u)$) pour **BLAC-bs**. Ces deux variantes utilisant l'énergie associée directement à la métrique, on a aussi $g(u) = B(u)$.

Algorithme 2: Algorithme **BLAC** exécuté sur chaque nœud u .

```

1 begin
2    $C_h \leftarrow 0$  ;
3    $Id_{min} \leftarrow +\infty$  ;
4   if BLAC-r* then
5      $\mathcal{R}(u) \leftarrow RNG(\mathcal{N}(u))$ ;
6      $g(u) = 1$ ;
7   else
8      $\mathcal{R}(u) \leftarrow \mathcal{N}(u)$ ;
9      $g(u) = B(u)$ ;
10  for  $i \in \mathcal{R}(u)$  do
11    if  $C_h < h(i) \times g(i) \vee (C_h = h(i) \times g(i) \wedge Id(i) < Id_{min})$  then
12       $P(u) \leftarrow i$ ;
13       $C_h \leftarrow h(i) \times g(i)$ ;
14       $Id_{min} \leftarrow Id(i)$ 
15  if  $h(u) \times g(u) > C_h \vee (h(u) \times g(u) = C_h \wedge Id(u) > Id_{min})$  then
16    /*  $u$  devient cluster-head */
17     $\mathcal{H}(u) \leftarrow u$ ;

```

La figure 4.1 illustre l'algorithme 2 exécuté sur chaque nœud de la façon suivante : les nœuds envoient dans leur voisinage un message *hello* contenant leur *Id* (figure 4.1(a)). Chaque nœud, après réception d'un message *hello* par chacun des nœuds de son voisinage, prend connaissance de ses voisins. Les nœuds envoient un nouveau message *hello* contenant à nouveau leur *Id* ainsi que les *Ids* de leurs voisins (figure 4.1(a)). Avec ces informations, chaque nœud peut calculer sa *densité* et la transmettre à ses voisins dans un nouveau message *hello*³ (figure 4.1(b)).

Finalement, chaque nœud choisit un père (sur la figure 4.1(b) le nœud a choisit le nœud b , le nœud b choisit m , etc.) ou devient *cluster-head* s'il a la métrique la plus élevée dans son voisinage (sur la figure 4.1(c) les nœuds m et i deviennent *cluster-heads*). On notera que les *clusters* sont ainsi construits suivant un arbre, a choisit b qui à son tour choisit m qui se choisit lui même et devient la racine de l'arbre. Nous utilisons l'arbre ainsi formé pour router les données vers le *cluster-head*.

Au début de la vie du réseau, juste après le déploiement, tous les nœuds ont le même niveau de batterie, l'algorithme agit donc de la même façon que DDR (ou Density-based). La batterie des nœuds commence à s'épuiser, plus ou moins vite, en fonction de la quantité de données que le nœud doit acheminer, de son activité

3. Le message *hello* contient toujours l'*Id* du nœud et ceux de ses voisins

propre et s'il est un *cluster-head* ou non. Par exemple sur la figure 4.1(c), $B(m)$ diminue plus rapidement que $B(n)$ (puisque m est *cluster-head*), m laissera alors son rôle de *cluster-head* à n (figure 4.4(a)).

La figure 4.3 montre que l'algorithme *Density-based*⁴ crée des trous dans le réseau. D'abord, en tant que *cluster-heads*, les nœuds m et l épuisent leur batterie et deviennent hors-service (figure 4.3(a)). Après qu'une réorganisation ait eu lieu, les nœuds n et d sont les nouveaux *cluster-heads*⁵ qui vont à leur tour devenir hors-service (Figure 4.3(b)) créant plus de trous dans le réseau. Par ailleurs, les nœuds n'ayant jamais été *cluster-heads* auront une réserve d'énergie bien plus grande.

D'autre part, avec l'algorithme **BLAC-bs** (figure 4.4) avant d'épuiser complètement leur batterie, les nœuds m et l vont abandonner leur rôle de *cluster-heads* au profit de n et d . L'énergie est donc équilibrée entre les différents nœuds du réseau. À leur tour, n et d laisseront leur rôle de *cluster-heads* à b et l , permettant l'équilibre énergétique dans tout le réseau.

4.5.2 BLAC-r*

Battery-Level Aware Clustering - RNG deGree (BLAC-rg) et **Battery-Level Aware Clustering - RNG density (BLAC-rs)** sont des variantes des deux premiers algorithmes. La plus grande différence est que l'algorithme s'exécute en deux étapes. Avant de calculer la métrique (le degré ou la densité), un *RNG* [88] est appliqué sur le graphe pour ne conserver qu'un sous-ensemble des liens existants (lignes 4-5 de l'algorithme 2). Le *RNG* appliqué est *bl-RNG* et il utilise le *Power factor* comme discriminant pour éliminer les liens du graphe [87].

L'algorithme de *clustering* s'exécute alors sur le nouveau graphe sans considération énergétique puisque le niveau de batterie a déjà été pris en compte lors de la réduction de graphe ($g(u) = 1$).

La figure 4.5 montre l'évolution de la structure du *clustering* au dessus du *RNG* pour les deux variantes *BLAC-r**. On peut voir à la première étape que les nœuds b , d et f sont choisis comme *cluster-heads* (figure 4.5(a)). Après un certain temps, les *cluster-heads* ont consommé plus d'énergie que les autres nœuds conduisant à une nouvelle construction du graphe (figure 4.5(b)). Sur les figures 4.5(b) et 4.5(c), on peut voir par exemple que les liens entre les nœuds a et b et entre les nœuds b et n ne seront plus utilisés.

4. qui ne considère pas le niveau de batterie

5. à noter qu'un *cluster-head* peut-être remplacé par plusieurs *cluster-heads* ou plusieurs *cluster-heads* remplacés par un seul ou encore un *cluster-head* existant absorber un *cluster* dont le *cluster-head* vient de faillir

4.6 Résultats

4.6.1 Paramètres de simulation

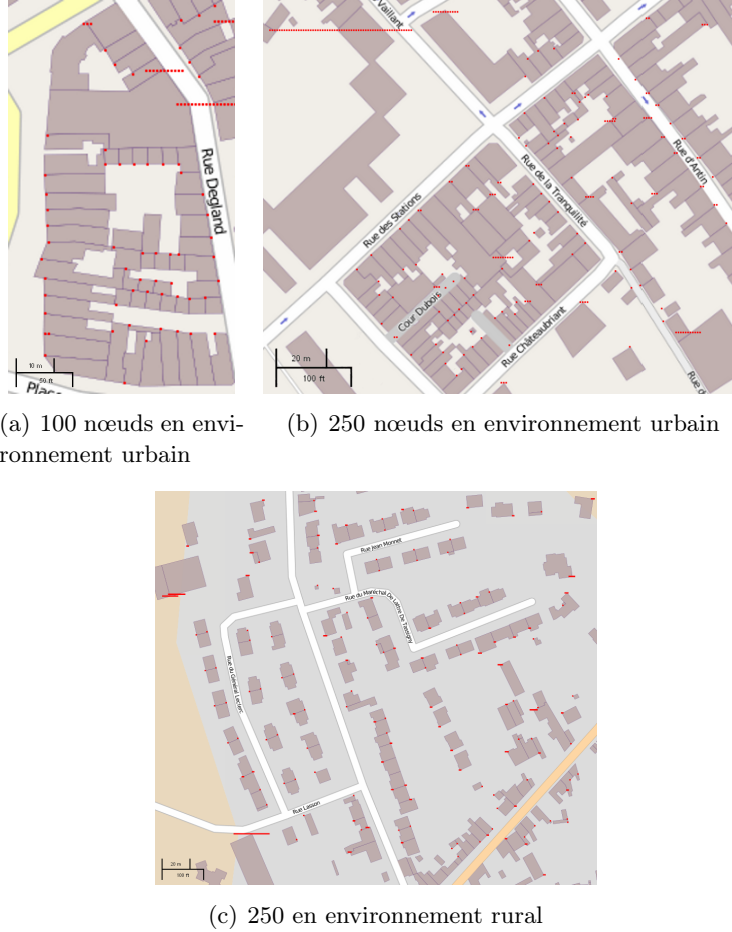


FIGURE 4.6 – Topologies réseau, chaque point rouge représente un nœud.

Pour évaluer les performances des différentes versions de **BLAC**, nous avons réalisé des simulations à l'aide de *WSNET* [83]. La section 2.3 du chapitre 2 donne plus de détails sur le simulateur. Nous comparons les quatre variantes de **BLAC** proposées avec trois approches issues de la littérature. DDR [18] et density-based [86] ont été choisis pour leurs similitudes avec **BLAC**. LEACH [19] a été choisi pour ses préoccupations concernant l'énergie dans le *clustering*. Afin d'observer différents comportements pour LEACH, nous avons choisi d'étudier ses performances pour trois valeurs différentes du paramètre p : $p = 5\%$, $p = 10\%$ et $p = 20\%$ ⁶. Le tableau 4.1 résume les différents paramètres de simulation.

Les topologies sont créées en utilisant des cartes réalistes issues de quartiers de

6. pour rappel p est le pourcentage moyen de *cluster-head* (voir section 4.4)

la ville de Lille (environnement urbain) et de petites villes du nord de la France (environnement rural). Les nœuds sont placés dans les rues de la ville, à proximité des bâtiments, grâce aux données du projet *OpenStreetMap* [105]. Cela permet de reproduire des topologies proches des topologies de l'application ciblée afin d'obtenir des résultats réalistes (cf chapitre 3). La figure 4.6 montre les topologies que nous avons utilisées dans les simulations.

paramètre	valeur
propagation	freespace
mac	802.15.4 CSMA-CA
CC2420 idle	0.77 mW
CC2420 Rx	35.46 mW
CC2420 Tx	31.32 mW
CC2420 data rate	250 kbit/s
GPRS idle	6.4 mW
GPRS Rx/Tx	1.25 W
GPRS Tx data rate	26.8 kbit/s
Capacité des batteries	32 mWh
Seuil de batterie critique	50%
Taille Hello	48 bit + 24 bit/ voisin

TABLE 4.1 – Paramètres de simulation des algorithmes de *clustering*

Toujours dans un souci de réalisme, nous avons utilisé des consommations énergétiques données par les fiches techniques de puce de communication radio utilisées par des nœuds capteurs. Nous avons considéré la puce *Texas Instruments CC2420 ZigBee®* [66] pour les communications inter-nœuds et le module *u-blox LEON-G100 GSM/GPRS* [106] pour les communications avec la station de base. La puce *TI CC2420* consomme 0.77 mW lorsqu'elle est en veille (idle), 35.46 mW en réception de données (Rx), et 31.32 mW en émission de données (Tx). La puce GPRS consomme 6.4 mW en veille (idle) et 1.25 W en réception (Rx) et en émission (Tx). La batterie simulée dispose de 31 mWh, ce qui représente moins de 1% d'une batterie réelle. La capacité de la batterie a été diminuée de façon à conserver une durée de simulation raisonnable. En effet, les simulations s'exécutent jusqu'à l'épuisement des batteries de tous les nœuds, ce qui peut être considérablement long. Un modèle de trafic est simulé. Chaque nœud génère 16 kbit de données périodiquement (toutes les 5 secondes) et les transmet à son père. Le *cluster-head* simule l'envoi de données à une station de base. Lorsqu'un nœud reçoit des données d'un autre nœud, il les stocke provisoirement en attendant de devoir émettre des données à son tour. Il agrège alors ses données et celles qu'il a reçues pour les envoyer à son père. Par exemple, sur la figure 4.1(c), après que le nœud *a* a généré ses 16 kbit de données, il les envoie à *b* qui les stocke en attendant de générer ses propres 16 kbit de données. Ensuite *b* envoie 32 kbit à *m* qui attend la génération de ses données pour envoyer

48 kbit à la station de base en utilisant le module GPRS.

Au démarrage, chaque nœud possède la même quantité d'énergie initialisée à 32 mWh. Lorsqu'un nœud est en veille, qu'il envoie un paquet de données ou qu'il reçoit un paquet de données, il déduit de son énergie restante, la quantité d'énergie correspondante à l'opération effectuée (Tab. 4.1) en fonction du type de lien radio utilisé (GPRS si le nœud est un *cluster-head*, *CC2420* sinon) et de la durée de l'opération. Le seuil de batterie critique est fixé à 50% pour l'algorithme *bl-RNG*.

4.6.2 Durée de vie du réseau

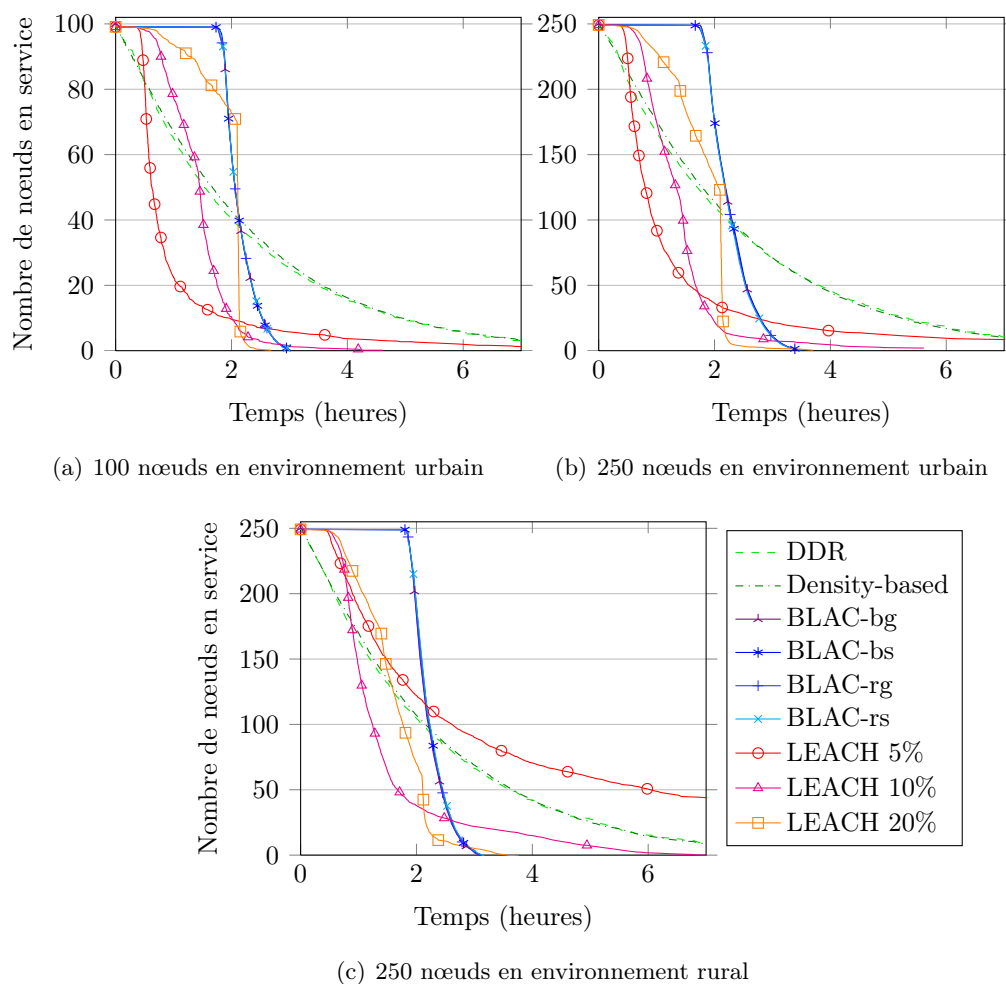


FIGURE 4.7 – Durée de vie du réseau

La figure 4.7 illustre la durée de vie du réseau exécutant les différents algorithmes simulés. Les graphiques représentent le nombre de nœuds en état de marche en fonction du temps

Nous pouvons voir dans les solutions qui ne considèrent pas le niveau de batterie, comme *Density-based* et *DDR*, que les *cluster-head* épuisent leur batterie rapidement. Lorsque les *cluster-head* sont hors service, d'autres nœuds prennent leur place, jusqu'à ce qu'ils épuisent leur énergie à leur tour, et ainsi de suite, jusqu'à ce qu'il n'y ai plus de nœud en état de marche. L'algorithme *LEACH* offre de meilleures performances, en améliorant grandement la durée avec tous les nœuds opérationnels en comparaison avec les algorithmes *DDR* et *Density-based*. Néanmoins, ses performances dépendent grandement du nombre de *cluster-heads* définis pendant la phase de configuration. *BLAC* maintient l'ensemble des nœuds en service jusqu'à trois fois plus longtemps par rapport aux autres algorithmes, comme prévu (voir section 4.2).

On notera que le coût de synchronisation des nœuds nécessaire pour *LEACH* n'a pas été pris en compte.

4.6.3 Taux de livraison

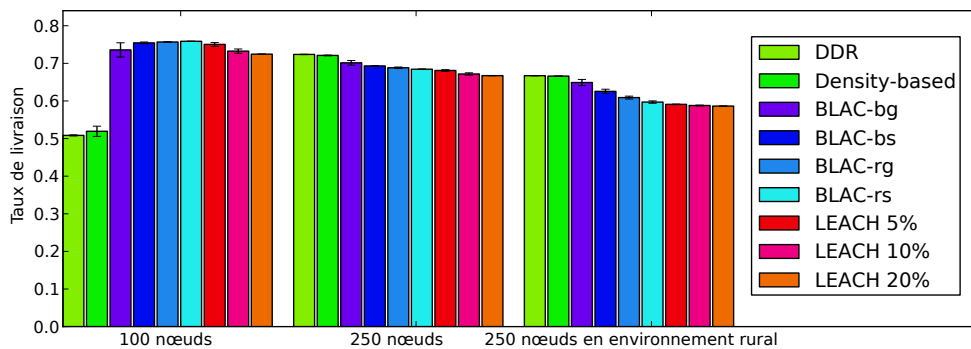


FIGURE 4.8 – Taux de livraison

La figure 4.8 montre le taux de livraison pour chaque algorithme, pour trois topologies simulées. Le taux de livraison est calculé comme la quantité de données reçue divisée par la quantité de données générée par l'ensemble des nœuds. On observe que les différentes variantes de *LEACH* perdent légèrement plus de données que les algorithmes *BLAC*, quelque soit le nombre de *cluster-heads* paramétrés. En effet, avec *LEACH*, les nœuds non attachés à un *cluster-head* ne peuvent pas émettre leurs données qui sont alors perdues.

On observe que les algorithmes *DDR* et *Density-based* perdent beaucoup plus de données dans les cas de simulations avec 100 nœuds. Comme on peut l'observer sur la figure 4.9, le nombre de *cluster-heads* est très faible alors que pendant le même temps, des nœuds deviennent hors-services continuellement (voir figure 4.7). Un faible nombre de *cluster-heads* implique que le chemin pour joindre un *cluster-head* est long, combiné à la disparition très fréquente de ces mêmes *cluster-heads*, cela conduit à des pertes importantes dans l'acheminement des messages.

Lorsque le nombre de nœuds du réseau augmente, on observe pour tous les algorithmes une perte plus importante de données (excepté pour *DDR* et *Density-based* comme expliqué précédemment). Pour les **BLAC** cela est dû au fait que lorsqu'un nœud intermédiaire devient hors-service, les données qu'il devait acheminer vers le *cluster-head* sont perdues. Avec *LEACH*, dont les *clusters* sont uniquement à un saut, peu de pertes ont lieu alors qu'un nœud intermédiaire devient hors service puisque le *cluster-head* est directement atteint par l'ensemble des nœuds du *cluster*.

4.6.4 Propriétés des réseaux

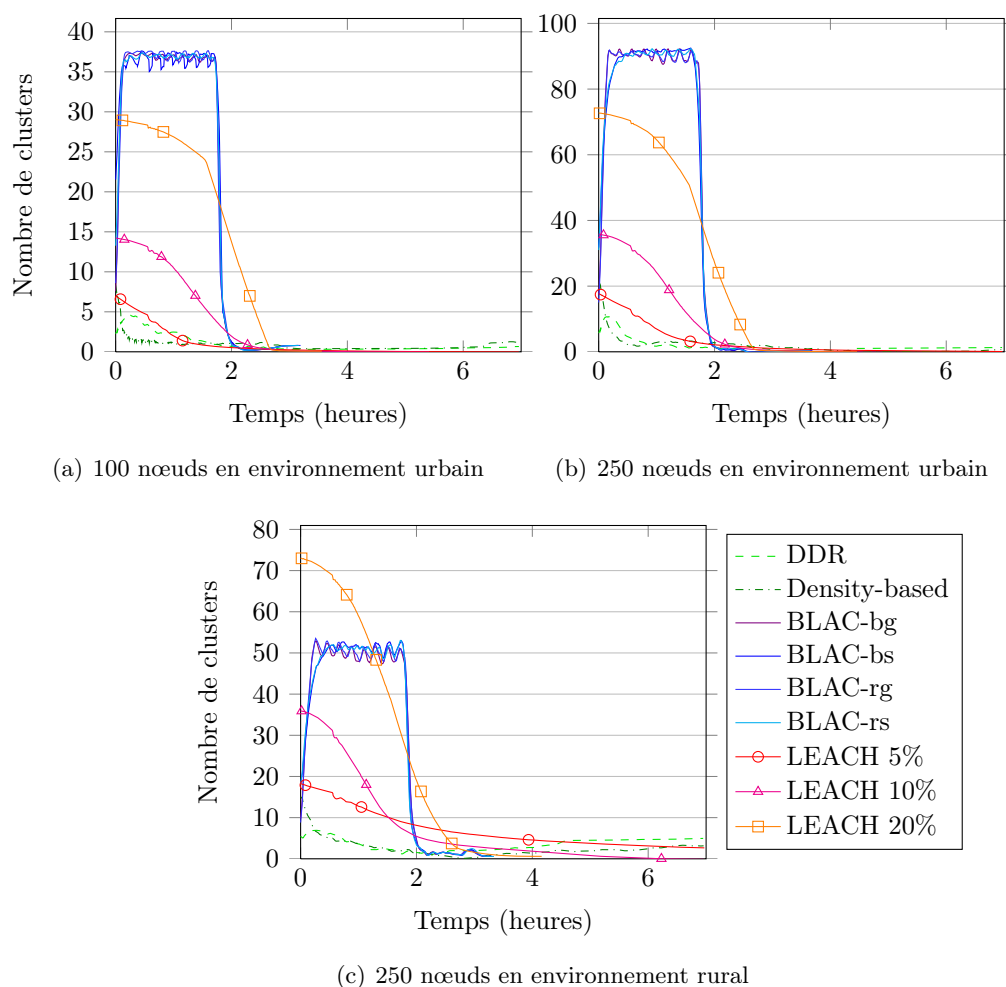
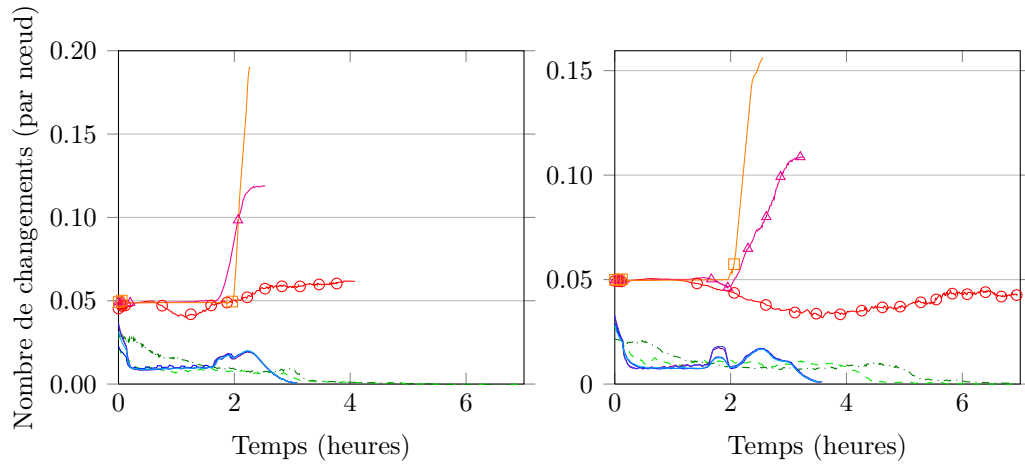


FIGURE 4.9 – Durée de vie du réseau

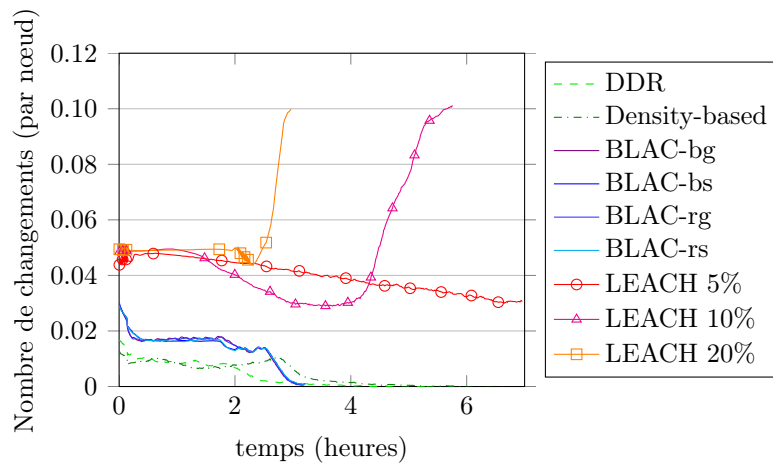
Le nombre de *clusters* dans le réseau est illustré en figure 4.9. Le nombre de *clusters* correspond au nombre de *cluster-heads* (un *cluster* possède un seul *cluster-head*). L'algorithme *LEACH* produit un nombre de *clusters* en relation avec le paramètre p . De plus, le nombre de *clusters* diminue au fur et à mesure que le nombre

de nœuds en fonctionnement diminue, comme prévu. Cela met en avant le besoin de paramétrage préliminaire du nombre de *clusters*, ce qui est une tâche complexe si on considère que le réseau peut être déployé sur des topologies différentes. Il est en effet préférable d'avoir un algorithme capable de s'adapter seul au contexte.



(a) 100 nœuds en environnement urbain

(b) 250 nœuds en environnement urbain



(c) 250 nœuds en environnement rural

FIGURE 4.10 – Stabilité du réseau

La figure 4.10 compare la stabilité du réseau en mesurant le nombre de changements de père à chaque pas de simulation. Sur cette figure, le tracé des courbes s'arrête lorsque tous les nœuds ont épuisé leur batterie. On peut voir que les algorithmes ayant une métrique qui n'est pas basée sur l'énergie (*DDR*, *Density-based*) ont une activité plus épisodique. Avec ces algorithmes, un changement de père a lieu lorsque le père est hors-service, obligeant les nœuds à trouver un nouveau père. Lorsque cela arrive, le *degré* ou la *densité* dans le voisinage du nœud hors-service

change, amenant à des restructurations.

Pour **BLAC**, qui intègre le niveau de batterie, des restructurations ont lieu plus continuellement car le niveau de batterie change au cours du temps. Lorsque le niveau de batterie d'un *cluster-head* diminue, un nœud dans son voisinage peut avoir une métrique plus intéressante que l'actuel *cluster-head*, le voisin devient alors automatiquement le nouveau *cluster-head* et l'ancien s'y attache, permettant ainsi de soulager énergétiquement l'ancien *cluster-head* au détriment du nouveau. Les variantes RNG sont proches des variantes non-RNG pour la stabilité, ce qui permet de faire le choix de la variante en fonction des besoins de l'application.

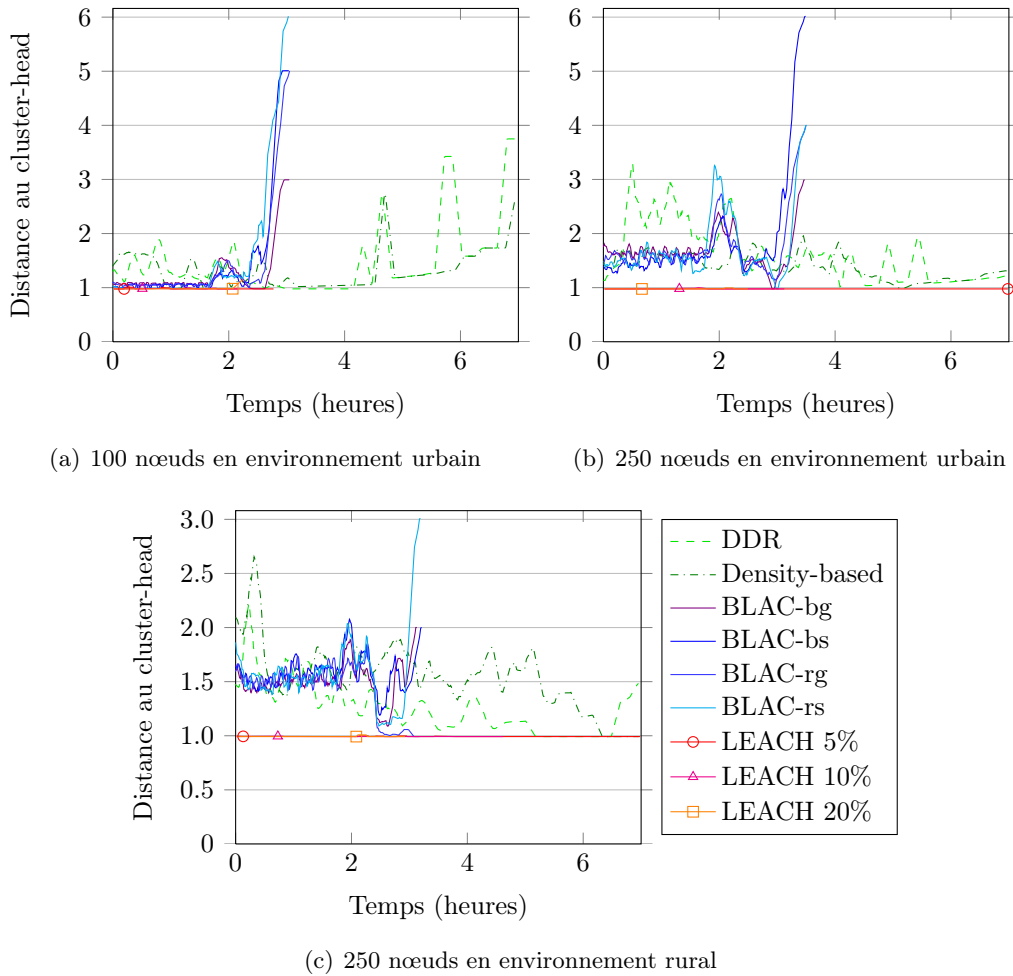


FIGURE 4.11 – Excentricité moyenne

La figure 4.11 montre l'éloignement moyen des nœuds par rapport à leur *cluster-head* en nombre de sauts. On observe une corrélation entre l'excentricité et le nombre de *clusters*. Au regard de la figure 4.9, on peut voir que plus les *clusters* sont profonds et plus ils contiennent de nœuds. Des *clusters* plus profonds impliquent alors plus

de communications intra-*cluster* à basse consommation pour joindre le *cluster-head*. Il s'agit donc de trouver le meilleur compromis entre le nombre de communications point à point et le nombre de communications GPRS pour optimiser la durée de vie du réseau. On constate que les algorithmes **BLAC** atteignent un compromis intéressant grâce à leur adaptation automatique.

Discussion

Nous avons vu dans l'analyse des résultats que les quatre versions de **BLAC** sont très proches. Les variantes RNG offrent un meilleur taux de livraison pour les réseaux de faibles densités alors que cette tendance s'inverse pour les réseaux de plus fortes densités. Le choix de la variante de **BLAC** doit être établi en fonction des besoins au niveau de la taille des *clusters*, de la stabilité et de la qualité de service demandée pour déterminer le meilleur compromis entre la consommation énergétique, la perte de messages et la stabilité. Si les nœuds sont mobiles par exemple, **BLAC-bs** est le meilleur choix car il offre la meilleure stabilité face à la mobilité. En effet les *clustering* basés sur la densité sont plus robustes à la mobilité. Pour les applications dans lesquelles la densité de nœuds est faible et les nœuds sont statiques, **BLAC-rg** est le meilleur choix car il offre le meilleur taux de livraison. Il pourra être intéressant d'étudier des valeurs différentes pour les consommations point à point et GPRS en fonction des technologies radio choisies pour l'application ciblée.

4.7 Conclusion

Dans ce chapitre nous introduisons une nouvelle famille d'algorithmes de *clustering*. Deux variantes combinent le niveau de batterie avec le degré ou la densité comme métrique pour la création des *clusters*. Les deux autres variantes appliquent une réduction de graphe basée sur le niveau de batterie avant la formation des *clusters*. En intégrant complètement la métrique de niveau d'énergie pour élire les *cluster-heads*, **BLAC** équilibre la consommation des nœuds sur les différents nœuds du réseau et maximise sa durée de vie. Ces algorithmes sont distribués et les modifications de structure sont gérées localement, permettant un passage à l'échelle sans problème. Les résultats montrent que notre proposition améliore la durée de vie du réseau jusqu'à trois fois, ce qui est très utile pour des applications comme celles rencontrées dans le projet *BinThatThinks*. Par ailleurs, nous avons testé nos propositions dans des environnements tels que ceux qui pourront être rencontrés dans *BinThatThinks*. Pour des travaux futurs il serait intéressant d'étendre la comparaison de **BLAC** avec d'autres algorithmes comme *WCA* [91]. D'autres modèles énergétiques pourraient aussi être étudiés (taille des paquets de données, consommation des puces radio ...)

La conduite d'expériences sur des plate-formes réelles telle que **SensLAB** [63] ou *Wisebed* [107, 108] serait aussi très instructive. Un autre domaine d'étude qui

pourrait donner des résultats intéressants concernant la consommation énergétique est l'ajustement de portée de communication des nœuds.

Routage géographique

Sommaire

5.1	Introduction	83
5.2	État de l'art	84
5.3	Proposition	87
5.4	Simulations et résultats	90
5.5	Auto-configuration du paramètre k	100
5.6	Conclusion	101

5.1 Introduction

Nous avons vu dans le chapitre 3 que les protocoles de routage géographiques ont de bonnes propriétés pour les réseaux de capteurs sans-fil. Ils sont sans état, passent facilement à l'échelle et offrent de bonnes performances. Le principal inconvénient de ce type d'algorithmes est évidemment qu'ils nécessitent que chaque nœud ait connaissance de sa position. Pour cela, il est possible d'équiper chaque nœud d'un récepteur GPS ou de configurer leur position au moment du déploiement, s'ils sont statiques.

D'un autre côté, nous avons les algorithmes de routage non géographiques. Ils n'utilisent pas la position des nœuds pour router les données mais uniquement les informations de voisinage. Ces algorithmes sont moins efficaces sur les grands réseaux et leur performance énergétique est moindre mais ils ne nécessitent pas de matériel ou de configuration supplémentaire.

Dans le contexte des villes intelligentes il est probable que le réseau soit hétérogène [26]. Certains nœuds peuvent connaître leur position tandis que d'autres non. Dans *BinThatThinks*, les nœuds doivent envoyer des informations au centre de tri qui est fixe, il est possible de configurer les nœuds avant leur déploiement pour connaître la position de cette destination afin de router géographiquement les paquets de données vers cette destination. Certaines poubelles publiques et les points de collecte volontaires sont fixes, il est donc possible de configurer les nœuds qui équipent ces poubelles pour qu'elles connaissent leur position. Dans cette situation, comme évoqué au chapitre 1, il peut-être intéressant de tirer partie des propriétés de ces nœuds pour router les informations des autres nœuds. Même si tous les nœuds d'une ville sont équipés d'un récepteur GPS, il est possible que certains d'entre eux

ne reçoivent pas le signal en raison des contraintes de l'environnement. Par exemple, deux quartiers d'une ville peuvent être reliés par un tunnel, dans lequel le signal GPS ne peut être reçu. La possibilité d'équiper tous les nœuds de GPS est aussi envisagée dans le projet *BinThatThinks*, toutefois les aléas dans la réception du signal satellite ne doivent pas mettre en péril les performances du protocole de routage. Nous proposons une solution applicable au projet *BinThatThinks*, dans lequel on considère que la réception du signal GPS ne sera pas optimale sur l'ensemble des nœuds. Par ailleurs, dans le contexte où le camion de ramassage fait partie du réseau, il convient de considérer que certains nœuds pourront être mobiles. De même les poubelles des usagers ne sont pas fixes, leur position peut alterner entre leur lieu de stockage et leur lieu de ramassage. De plus certains utilisateurs positionnent leur poubelle sur leur lieu de ramassage juste avant celui-ci. Pour ces raisons une réactivité du protocole de routage sera bénéfique à l'application *BinThatThinks*.

Dans de tels contextes, il est intéressant d'avoir un algorithme de routage qui tire partie des informations de positions lorsqu'elles sont disponibles mais qui puisse aussi fonctionner lorsque ces informations ne sont pas disponibles pour diverses raisons. Pour cela, nous proposons l'algorithme HGA, pour Hybrid Greedy-AODV. L'algorithme fonctionne de manière similaire à l'algorithme de routage géographique Greedy [22] lorsque c'est possible et, lorsque le voisinage direct d'un nœud ne permet plus de progrès, un message RREQ (route request) est envoyé dans le voisinage, à l'instar de l'algorithme AODV [20], afin de trouver un nouveau chemin.

5.2 État de l'art

A notre connaissance, dans la littérature, les algorithmes permettant le routage lorsqu'une partie ou tous les nœuds n'ont pas connaissance de leur position sont des algorithmes de routage utilisant des coordonnées virtuelles. Dans la section 3.4 du chapitre 3, nous décrivons différents algorithmes de routage géographique, notamment l'algorithme Greedy [22] qui sert de base à notre proposition.

Certains algorithmes permettent de déduire la position de nœuds ne la connaissant pas à partir d'autres nœuds du réseau qui la connaissent [109, 110]. Toutefois ces solutions ont des performances médiocres à cause de la difficulté pour déterminer la distance entre deux nœuds sans dispositif spécialisé [111]. Par ailleurs Seada et al. [112] ont montré qu'une petite erreur sur la position pouvait engendrer un grand nombre de pertes dans les protocoles de routage géographiques.

Les algorithmes de routage utilisant des coordonnées virtuelles permettent d'utiliser les avantages des algorithmes de routage géographique sans que les nœuds n'aient besoin de connaître leur position géographique réelle. L'inconvénient majeur de ces solutions réside dans la mise en place du système de coordonnées virtuelles. En effet il est nécessaire d'inonder totalement le réseau de messages de contrôle plusieurs fois au démarrage. La solution Vcap [113] proposée par Caruso et al. repose sur des coordonnées virtuelles et un routage Greedy. Des nœuds particuliers

sont choisis pour leurs propriétés intéressantes pour servir de points de départs aux différentes coordonnées du système. Ces nœuds sont appelés les ancres. Le système de coordonnées repose sur la distance en nombre de sauts entre un nœud et les différentes ancres. Un nœud racine prédéfini est utilisé pour initialiser l'élection des ancres. Le nœud racine, en diffusant un message à travers tout le réseau, permet l'élection d'un nœud proche des limites du réseau. Ce nœud diffuse à son tour un message dans tout le réseau qui permet l'élection de la première ancre. À son tour l'ancre diffuse un message pour élire la seconde ancre, et ainsi de suite. Les ancres sont choisies de manière à être proches des limites du réseau et éloignées les unes des autres pour limiter les doublons dans le système de coordonnées. Dans Vcap le réseau est inondé au minimum 5 fois. À cause de l'absence d'unicité des coordonnées, la livraison ne peut être garantie.

Stojmenovic [114] introduit la notion de *cost over progress* dont l'idée est d'optimiser le ratio entre le coût d'une transmission sur le progrès effectué par cette transmission. VCost [115] s'appuie sur cette idée et sur le système de coordonnées de VCap. Cette combinaison permet de réduire les besoins énergétiques de l'algorithme pour le routage.

L'algorithme LTP [116] proposé par Chávez et al. utilise une racine et un système de coordonnées basé uniquement sur cette racine plutôt que plusieurs ancres. La racine doit être unique, elle peut être élue à l'initialisation du réseau ou prédéfinie avant le déploiement du réseau. Les auteurs ne décrivent pas de mécanisme permettant de choisir la racine. La racine diffuse un message "discovery" dans son voisinage auquel chacun de ses voisins répond avec son Id par un message «Tag Request». La racine ordonne alors par leur Id les nœuds ayant répondu et attribue à chacun des nœuds un label (R1, R2, R3, ...). Lorsqu'un nœud a reçu un label, il répète le même processus. Seuls les nœuds n'ayant pas encore de label répondent, ils se voient alors attribuer un label qui contient le label du nœud père ainsi qu'un nombre unique (R11, R12, R13, etc. si le nœud père est R1). Le processus continue jusqu'à ce que tous les nœuds du réseau aient un label. Les labels ainsi créés sont uniques et permettent d'établir une route entre chaque paire de nœuds du réseau en remontant l'arbre jusqu'à un label commun entre la source et la destination puis en redescendant l'arbre vers la destination. La livraison est donc garantie si le réseau est connecté et si il n'y a pas de modification du chemin pendant la transmission d'un message. La figure 5.1 montre un exemple de labels créés par l'algorithme LTP et un exemple de route entre les nœuds R411 et R421.

HECTOR [117] combine les systèmes de coordonnées de LTP et de Vcap. Lorsqu'un nœud doit router un paquet il sélectionne les nœuds dans son voisinage qui permettent de faire un progrès positif ou nul dans le système de coordonnées de LTP. Parmi ces nœuds il applique une sélection *cost-over-progress* en utilisant le système de coordonnées issue de Vcap pour choisir le nœud permettant de minimiser la consommation énergétique. HECTOR permet donc de garantir la livraison des paquets tout en réduisant la consommation énergétique pour le routage.

D'autres solutions de routage utilisant des coordonnées virtuelles ont été pro-

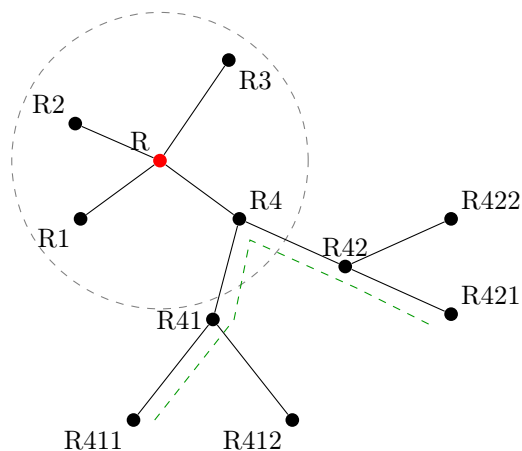


FIGURE 5.1 – Exemple de coordonnées dans LTP. Les pointillés verts représentent le chemin parcouru par un paquet allant du nœud R411 au nœud R421.

posées, certaines sans garantie de livraison [118, 119, 120], d'autre avec garantie de livraison [116, 121]. Toutefois, les algorithmes utilisant des coordonnées virtuelles ont l'inconvénient de ne pas supporter la mobilité des nœuds dans le réseau. Si les nœuds sont mobiles, il est alors nécessaire de générer à nouveau le système de coordonnées, ce qui augmente les coûts énergétiques. De plus il est nécessaire d'inonder totalement le réseau au démarrage, ce qui peut introduire une très forte latence au démarrage lorsque le réseau est très grand.

Notre proposition utilise des techniques de *route request* et *route reply* similaires à celles introduites dans AODV [20]. Nous introduisons ici leur fonctionnement. La figure 5.2 montre une recherche de route dans l'algorithme AODV.

AODV repose sur un routage non géographique totalement réactif. En effet, les routes sont établies uniquement à la demande, lorsque celles-ci sont nécessaires (au contraire de OLSR [21]). Lorsqu'un nœud source souhaite établir une route vers un nœud destination, il diffuse dans son voisinage un paquet RREQ (*route request*). Chaque nœud recevant la RREQ conserve une trace de celle-ci jusqu'à un délai d'expiration (défini à 3000 ms dans AODV). La mémorisation des RREQ permet de créer le *reverse path*, comme le montre la figure 5.2(a), qui sera utilisé lorsque la destination aura été trouvée. Si un nœud ne connaît pas de route vers la destination, qu'il n'est pas la destination et qu'il n'a pas déjà reçu cette RREQ, il la diffuse à son tour dans son voisinage. Lorsqu'un nœud recevant une RREQ connaît une route vers la destination, ou s'il est lui-même la destination, il envoie un paquet RREP (*route reply*) au nœud lui ayant envoyé la RREQ. Le paquet RREP suit alors le *reverse path* et chaque nœud sur cette route enregistre la source de la RREP dans sa table de routage afin de créer le *forward path*. La figure 5.2(b) montre la mise en place du *forward path* et l'expiration des *reverse path*.

Nous avons vu dans cet état de l'art que des solutions existent pour router des

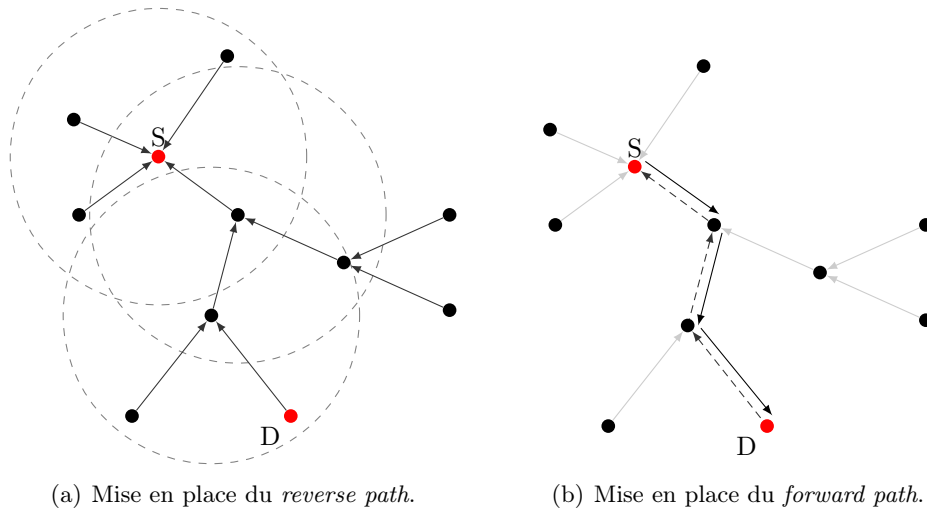


FIGURE 5.2 – Exemple de recherche de route du nœuds S vers le nœud D dans AODV.

données dans les réseaux de capteurs sans-fil. Toutefois aucune solution ne considère l'hypothèse qu'une partie des nœuds seulement a connaissance de sa position. Par ailleurs les solutions proches passent difficilement à l'échelle et ne supportent pas la mobilité des nœuds.

5.3 Proposition

L'idée générale de HGA est d'appliquer un routage géographique greedy lorsque cela est possible. Quand le routage greedy n'est pas possible une recherche de route est envoyée dans le voisinage à k -sauts du nœud. Le paramètre k est la profondeur maximum de recherche de route et détermine la variante de HGA, qu'on notera HGA- k . Lorsqu'un nœud doit router un paquet et qu'aucun de ses voisins ne permet de faire un progrès vers la destination, une recherche de route similaire à AODV [20] est effectuée.

L'ensemble des nœuds ayant connaissance de leur position est contenue dans P . Si le nœud u a connaissance de sa position alors $u \in P$. L'entête d'un paquet de données contient la position de la destination, ainsi que la dernière position connue par laquelle le paquet a transité.

Pour router un paquet de données, un nœud u commence par vérifier si un nœud parmi ses voisins permet un progrès vers la destination. Si u a connaissance de sa propre position, la notion de progrès dépendra alors de la distance de u par rapport à la destination. Si u n'a pas connaissance de sa position alors le progrès dépendra de la dernière position connue contenue dans le paquet de données si elle est disponible, sinon n'importe quel nœud ayant connaissance de sa position sera considéré comme

présentant un progrès. Si aucun voisin de u ne permet un progrès alors la table des routes R est parcourue à la recherche d'une route permettant un progrès. Si aucune route ne permet de router le paquet, une *route request* (RREQ) est envoyée dans le voisinage de u et une nouvelle recherche dans la table des routes de u est effectuée après un délai *RREQ_TIMEOUT*. L'algorithme 3 résume la recherche de route sur un nœud.

Lorsqu'un nœud u reçoit une RREQ, il peut la transmettre, y répondre par une *route reply* (RREP) ou ne rien faire. Le nœud u commence d'abord par analyser sa table des routes R afin de vérifier, en comparant le numéro de séquence et la source de la RREQ à celui des routes, si u a déjà reçu cette RREQ. Si c'est le cas, la recherche s'arrête et la RREQ reçue est ignorée, sinon u recherche parmi ses voisins si un nœud est plus proche de la destination que la source de la RREQ. Si une route est trouvée parmi les voisins de u , une *route reply* (RREP) est envoyée au nœud ayant transmis la RREQ à u . La RREP contient la source et le numéro de séquence issus de la RREQ, elle contient aussi la position du voisin de u permettant d'atteindre la destination. Si aucune route n'est trouvée, la RREQ est réémise à condition que la profondeur maximale pour la RREQ (*MAX_HOP_RREQ*) n'a pas été atteinte. Lorsqu'une RREQ est émise, une nouvelle entrée dans la table de routage est créée avec un drapeau "en attente" permettant de savoir qu'une RREQ a été envoyée mais qu'il n'y a pas encore de route pour cette destination. L'algorithme 4 décrit les opérations effectuées lors de la réception d'une RREQ, l'ensemble W contient les routes en attente.

Lorsqu'une RREP est reçue par un nœud u , la table de routage est mise à jour à la recherche de la route "en attente" associée. Le drapeau "en attente" est alors supprimé et le nœud voisin permettant d'atteindre la destination est mis à jour. De même, la position de la destination est mise à jour, en effet la RREP peut indiquer une position permettant de s'approcher de la destination mais pas nécessairement la position exacte de la destination. En enregistrant la position du nœud contenue dans la RREQ les informations de la table de routage sont plus précises.

HGA ne permet pas de garantir la livraison des paquets, en effet si aucun nœud n'a connaissance de sa position dans le k -voisinage d'un nœud, celui-ci ne pourra pas router les paquets. Toutefois, le mécanisme de recherche de route permet de contourner les impasses en recherchant une nouvelle route dans la limite de k -sauts. La figure 5.3 montre un exemple de routage du nœud S qui n'a pas connaissance de sa position jusqu'au nœud D qui a connaissance de sa position. Sur la figure 5.3(a) S effectue une *route request* avec une profondeur de 3 maximum. Les deux nœuds recevant la RREQ connaissant leur position répondent et S choisit celui qui permet la meilleure progression vers la destination. Une fois le paquet routé vers la destination intermédiaire, une nouvelle RREQ est effectuée, le paquet continue sa progression puis est routé en utilisant la technique du routage géographique car la destination intermédiaire possède un voisin qui lui permet un progrès vers la destination 5.3(b). Enfin, sur la figure 5.3(c) une dernière RREQ est effectuée qui permet d'atteindre la destination.

Algorithme 3: Réception d'un paquet de données au nœud u vers D avec l étant le dernier nœud ayant transmis le paquet dont la position est connue.

```

1  if  $u = D$  then
2     $\leftarrow$  exit /* succès */ ;
3   $next \leftarrow -1$  ;
4  if  $u \in P$  then
5     $\leftarrow$   $dist \leftarrow |uD|$  ;
6     $\leftarrow$   $l \leftarrow u$  ;
7  else if  $l \neq -1$  then
8     $\leftarrow$   $dist \leftarrow |lD|$  ;
9  else
10    $\leftarrow$   $dist \leftarrow +\infty$  ;
11  for  $v \in N(u)$  do
12    if  $|vD| < dist$  then
13       $\leftarrow$   $next \leftarrow v$  ;
14       $\leftarrow$   $dist \leftarrow |vD|$  ;
15  if  $next < 0$  then
16    /* pas de route trouvée dans le voisinage direct */
17     $\leftarrow$   $broadcast(RREQ)$  ;
18     $\leftarrow$   $R \leftarrow R \cup RREQ$  ;
19     $\leftarrow$   $W \leftarrow W \cup RREQ$  ;
20     $\leftarrow$   $wait(RREQ\_TIMEOUT)$  ;
21    if  $\neg(\exists r, r \in R \wedge |rD| < dist) \wedge r \notin W$  then
22       $\leftarrow$   $R \leftarrow R \setminus RREQ$  ;
23      /* pas de route trouvée */
24       $\leftarrow$   $drop(data)$  ;
25       $\leftarrow$  exit /* échec */ ;
26  else
27    for  $r \in R$  do
28      /* Pour chaque route connue */
29      if  $|rD| < dist$  then
30         $\leftarrow$   $dist \leftarrow |rD|$  ;
31         $\leftarrow$   $next \leftarrow r$  ;
32   $\leftarrow$   $send(data, next)$  /* données transmises */ ;

```

Algorithme 4: Réception d'une RREQ initiée par S avec le numéro de séquence $seqnum$ sur le nœud u pour la destination D .

```

1 if  $(S, seqnum) \in R$  then
2    $\lfloor exit$  /* RREQ déjà envoyée */;
3    $next \leftarrow \emptyset$ ;
4    $dist \leftarrow |SD|$ ;
5   for  $v \in N(u)$  do
6     if  $|vD| < dist$  then
7        $next \leftarrow v$ ;
8        $dist \leftarrow |vD|$ ;
9   if  $next = \emptyset$  then
10     $hop \leftarrow hop - 1$ ;
11     $R \leftarrow R \cup RREQ$ ;
12    if  $hop > 0$  then
13       $\lfloor broadcast(RREQ)$ ;
14  else
15     $\lfloor send(RREP, u, next)$ ;

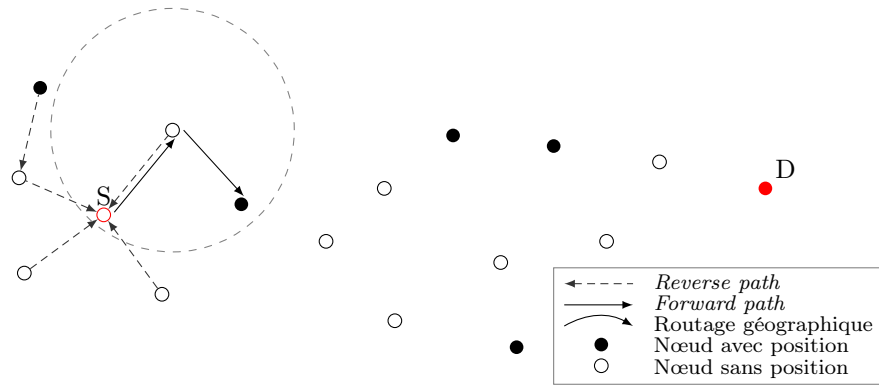
```

5.4 Simulations et résultats

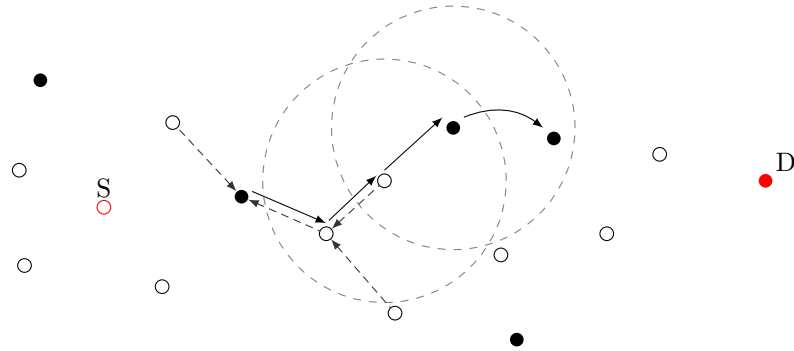
Nous avons choisi de comparer notre algorithme HGA à l'algorithme Vcap qui permet un routage proche des algorithmes de routage géographique mais utilise des coordonnées virtuelles. Nous avons aussi simulé l'algorithme greedy, légèrement adapté pour pouvoir fonctionner avec des nœuds n'ayant pas connaissance de leur position. Dans cette variante, les nœuds qui n'ont pas connaissance de leur position n'envoient pas de message HELLO, ainsi ils ne participent pas au routage mais peuvent émettre des paquets de données. Lorsqu'un nœud doit transférer un paquet de données (parce qu'il en est la source ou parce qu'il doit le transmettre), il choisit parmi ses voisins ayant connaissance de leur position celui qui est le plus proche de la destination permettant de faire un progrès. Si le nœud qui émet un paquet de données ne connaît pas sa position, la notion de progrès n'est pas mesurable, le voisin le plus proche de la destination sera alors choisi.

Nous avons simulé notre algorithme HGA avec des profondeurs de recherche de 1, 2, 3 et 5 sauts. L'algorithme VCap a été simulé avec 3 ancres et une variante avec 5 ancres.

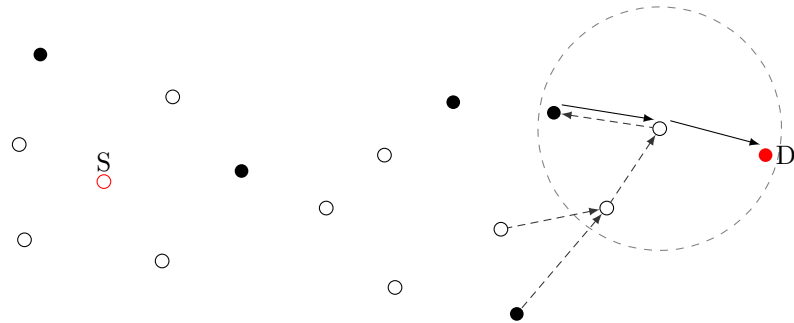
Les simulations sont effectuées avec le simulateur WSNET [62] (voir section 2.3 du chapitre 2 pour plus de détails). Chaque simulation a une durée de 10 min. Nous générons des topologies aléatoires connectées de 500, 750, 1000, 1500, 2000, 2500 et 3000 nœuds dans un domaine de 500 m $\times$ 500 m pour obtenir des réseaux de degrés moyens 10, 15, 20, 30, 40, 50 et 60. Les nœuds ont une portée de communication de 40 m. Le réseau ainsi obtenu a un diamètre minimum de 12 sauts. Chaque combinaison (algorithme, degré moyen) est exécutée 50 fois sur des topologies dif-



(a) Première RREQ.



(b) Deuxième RREQ et routage géographique.



(c) Troisième RREQ.

FIGURE 5.3 – Exemple de routage dans HGA.

férentes. La même topologie est utilisée pour la même combinaison (degré moyen, itération) pour les différents algorithmes. Ainsi la première simulation pour l'algorithme HGA-1 utilise la même topologie que pour l'algorithme VCAP-3, GREEDY, etc.

Pour simuler un trafic de données, toutes les 10 secondes, chaque nœud choisit

aléatoirement un destinataire ayant connaissance de sa position. Le nœud source envoie alors le paquet de données suivant l'algorithme qu'il exécute. Les nœuds sont désynchronisés pour que les émissions de données ne se fassent pas toutes au même moment.

Nous mesurons le taux de livraison, le nombre de messages de contrôle et la mémoire utilisée. Le taux de livraison est le rapport entre le nombre de messages reçus et le nombre de messages envoyés. Les messages de contrôle sont les messages HELLO, RREQ, RREP et les messages d'initialisation du système de coordonnées. Enfin, la mémoire utilisée est comptabilisée comme étant la moyenne du nombre de voisins enregistrés plus le nombre de routes connues pour chaque nœud. Les paramètres de simulation sont récapitulés dans le tableau 5.1

Paramètre	Valeur
Durée (m)	10
Couche MAC	idealmac
Interférences	none
Taille des données (octets)	10
Taille de l'entête (octets)	88
taille du domaine	500 m × 500 m
Portée de communication	40 m
Nombre de nœuds	500, 750, 1000, 1500, 2000, 2500, 3000
Degré	10, 15, 20, 30, 40, 50, 60
Itérations	50

TABLE 5.1 – Paramètres de simulation des algorithmes de routage géographique

Pour plus de clarté, nous ferons référence dans la suite à HGA-1 pour l'algorithme HGA avec une profondeur de recherche de 1 saut ($k = 1$). De même HGA-2 (resp. HGA-3 et HGA-5) se réfèrent à l'algorithme HGA avec une profondeur de recherche de 2 (resp. 3 et 5) sauts. Enfin VCAP-3 et VCAP-5 désignent l'algorithme VCAP avec 3 et 5 ancres. Les courbes VCAP-3 et VCAP-5 ont été dupliquées sur les figures avec 1%, 5% et 10% de nœuds ayant connaissance de leur position afin de comparer plus facilement les résultats avec les autres algorithmes (car Vcap n'utilise pas les données de position et donc est indépendant de ces paramètres).

5.4.1 Taux de livraison

Les figures 5.4, 5.5 et 5.6 montrent le taux de livraison obtenu pour les quatre variantes de HGA, les deux variantes de Vcap et l'algorithme Greedy simulés. On observe sur la figure 5.4 que quel que soit le degré moyen du réseau, avec 1% des nœuds ayant la connaissance de leur position, HGA-5 a un taux de livraison supérieur aux deux variantes de Vcap et à Greedy. On observe aussi qu'à partir d'un

degré moyen de 40, HGA-5 atteint un taux de livraison proche de 100%. Comme on peut s'y attendre HGA-1 offre de moins bonnes performances que HGA-2, qui lui-même offre de moins bonnes performances que HGA-3 tandis que HGA-5 obtient les meilleures performances. De même, Vcap est meilleur avec 5 ancres qu'avec 3. Cette hiérarchie est aussi observée avec 5% et 10% de nœuds ayant connaissance de leur position comme le montrent les figures 5.5 et 5.6.

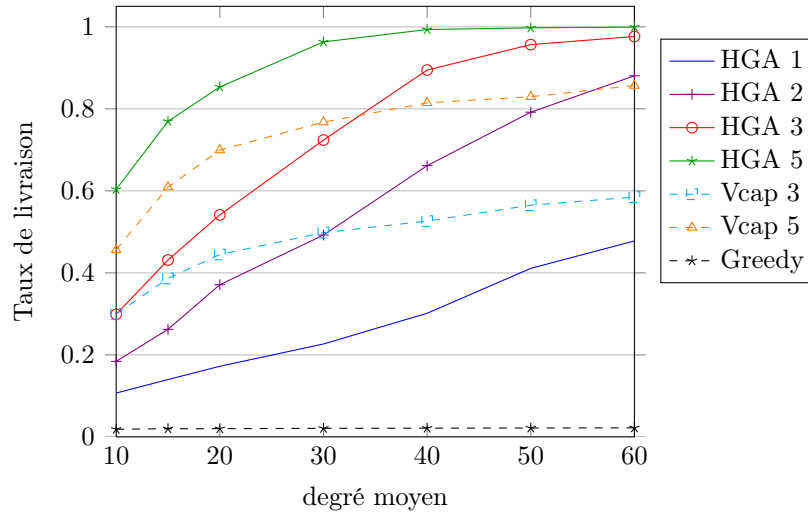


FIGURE 5.4 – Taux de livraison des différents algorithmes de routage avec **1%** des nœuds ayant leur information de position pour HGA et Greedy.

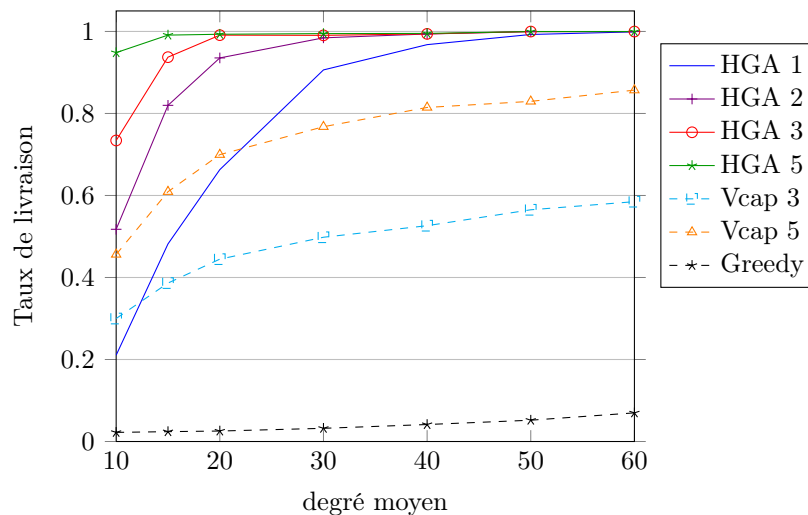


FIGURE 5.5 – Taux de livraison des différents algorithmes de routage avec **5%** des nœuds ayant leur information de position pour HGA et Greedy.

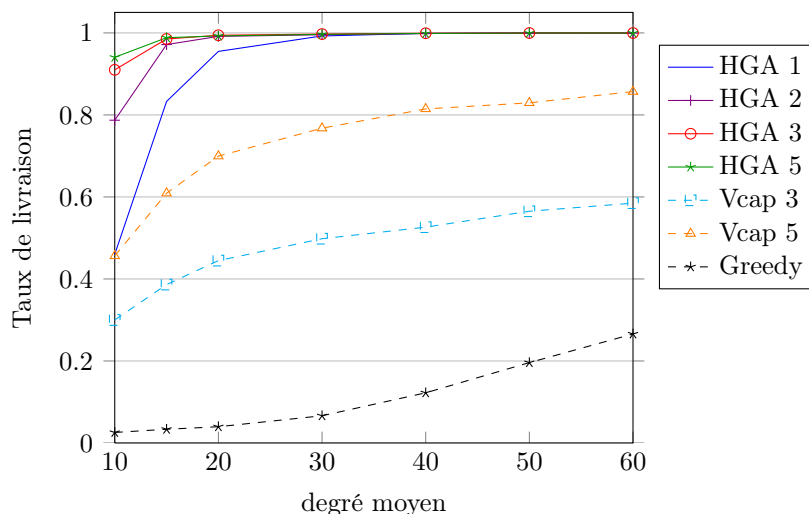


FIGURE 5.6 – Taux de livraison des différents algorithmes de routage avec **10%** des nœuds ayant leur information de position pour HGA et Greedy.

On peut aussi observer que HGA-1 obtient de bien meilleures performances que greedy. HGA-1 a un taux de livraison 5 fois plus élevé pour un degré de 10 et quasiment 15 fois plus élevé pour un degré de 40 alors que seulement 1% des nœuds connaissent leur position. Cela montre qu'avec une recherche de route d'une profondeur de 1 seulement, on obtient des performances très intéressantes. Il sera intéressant d'observer quel est le coût en nombre de messages de cette recherche de route dans la section suivante.

On remarque que les algorithmes HGA tirent mieux parti d'une densité plus élevée des nœuds. En effet pour 1% de nœuds ayant connaissance de leur position (figure 5.4), la croissance du taux de livraison pour des degrés allant 10 à 20 est similaire pour les différentes variantes de HGA et de Vcap. À partir d'un degré de 20, par contre, la croissance ralentit plus fortement pour Vcap que pour HGA. Cela permet à HGA-2 de passer devant Vcap-3 à partir d'un degré moyen de 25, HGA-3 passe devant Vcap-5 à partir d'un degré moyen de 35.

Avec 5% des nœuds ayant connaissance de leur position (figure 5.5), les performances de HGA sont améliorées. Ce comportement est normal car une RREQ a plus de chances de trouver un nœud connaissant sa position et donc un nœud dans la direction de la destination. Il est intéressant de noter que la croissance du taux de livraison est plus élevée qu'avec 1% de nœuds ayant connaissance de leur position. Cela permet à toutes les variantes de HGA de dépasser les deux variantes de Vcap dès qu'un degré de 25 est atteint. Ce comportement s'observe aussi en s'accroissant d'avantage pour une connaissance de la position de 10% (5.6). Ces résultats montrent que HGA tire mieux parti d'une densité de nœuds élevée que Vcap. En effet dans Vcap le nombre de nœuds ayant les mêmes coordonnées ne diminue pas nécessairement avec l'augmentation du degré des nœuds alors que les RREQ de

HGA ont plus de chances de trouver un nœud dans la direction de la destination comme expliqué précédemment.

5.4.2 Nombre de messages de contrôle

Les figures 5.7, 5.8 et 5.9 montrent le nombre moyen de messages de contrôle envoyés par nœud. Pour 1% des nœuds connaissant leur position (figure 5.7), on observe pour HGA, que plus la profondeur de recherche de route est élevée, plus le nombre de messages de contrôle envoyés est élevé. De même, plus le degré moyen est élevé, plus le nombre de messages envoyés est élevé. Cela s'explique car plus de nœuds reçoivent les RREQ lorsque le degré est plus élevé, chaque nœud recevant une RREQ la ré-émet potentiellement, ce qui explique l'augmentation exponentielle du nombre de messages émis avec le degré moyen du réseau. Cela explique aussi pourquoi l'augmentation est plus forte lorsque la profondeur de recherche de route est plus grande. On voit que le nombre de messages envoyés par l'algorithme greedy est très faible, cela s'explique par le fait que seuls les nœuds connaissant leur position émettent des messages de contrôle. Par ailleurs, pour greedy comme pour Vcap, le nombre de messages de contrôle envoyés est constant car il ne dépend pas du degré moyen. Pour un faible degré moyen, le nombre de messages envoyés par les différents algorithmes est proche, l'écart se creuse de plus en plus lorsque le degré augmente. Pour des degrés faibles, il sera intéressant de privilégier HGA-5 qui offre un taux de livraison bien supérieur aux autres alternatives, tout en ajoutant une quantité raisonnable de messages de contrôle. Lorsque le degré moyen du réseau augmente l'algorithme Vcap est plus approprié car il offre un taux de livraison proche, voire plus élevé que certaines variantes de HGA tout en générant beaucoup moins de messages de contrôle.

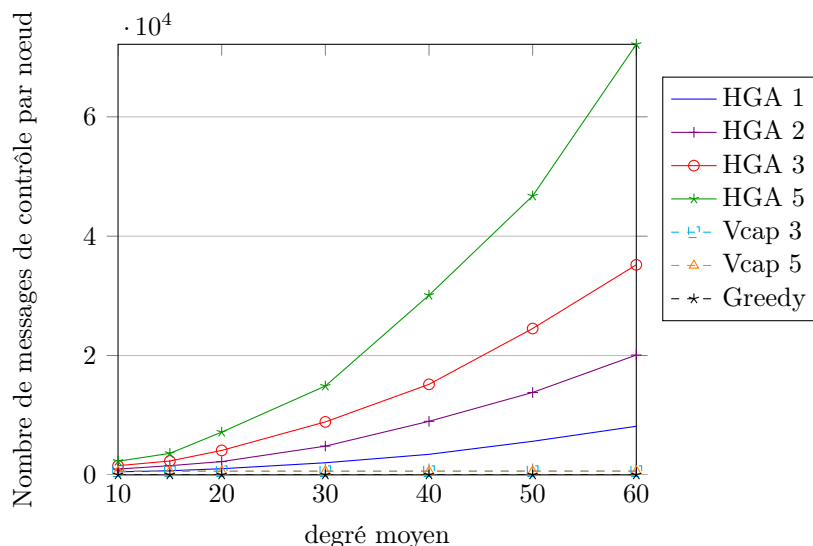


FIGURE 5.7 – Nombre de messages de contrôle des différents algorithmes de routage avec 1% des nœuds ayant leur information de position pour HGA.

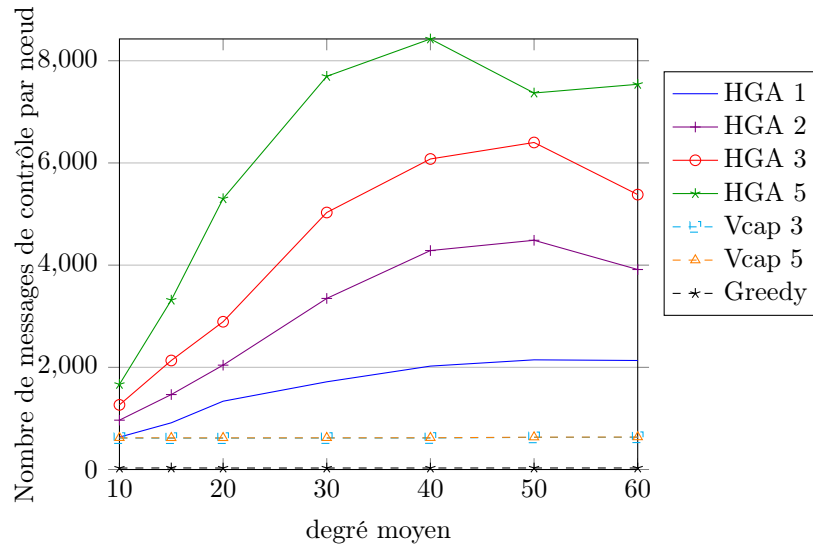


FIGURE 5.8 – Nombre de messages de contrôle des différents algorithmes de routage avec 5% des nœuds ayant leur information de position pour HGA.

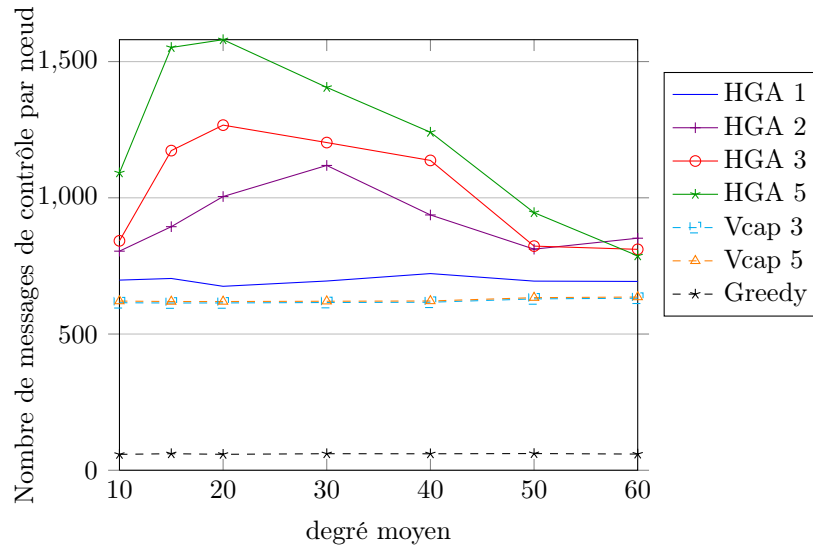


FIGURE 5.9 – Nombre de messages de contrôle des différents algorithmes de routage avec 10% des nœuds ayant leur information de position pour HGA.

Lorsque le nombre de nœuds connaissant leur position augmente à 5% le nombre de messages de contrôle diminue fortement pour HGA, notamment les variantes dont la profondeur de recherche est élevée. Ce comportement s'explique parce que la probabilité de trouver une route lors d'une RREQ augmente lorsque le nombre de

nœuds connaissant leur position augmente. Cette probabilité augmente aussi lorsque le degré moyen du réseau augmente (il y a plus de nœuds dans le voisinage d'un nœud pouvant satisfaire la demande), ce qui explique la tendance à l'inversion de la croissance pour les différentes variantes de HGA. Cette inversion de croissance devient même une décroissance lorsque le nombre de nœuds connaissant leur position passe à 10% (figure 5.9). Toutefois, l'intérêt des variantes ayant une profondeur de recherche élevée n'est plus justifiée car le taux de livraison atteint 100% rapidement dans ce cas.

5.4.3 Taille de la table de routage

Les figures 5.10, 5.11 et 5.12 montrent l'occupation mémoire des différents algorithmes. On observe que les besoins en mémoire sont beaucoup plus élevés pour HGA que pour Greedy ou Vcap pour des réseaux dont 1% des nœuds ont connaissance de leur position (figure 5.10). Pour 5% des nœuds connaissant leur position (figure 5.11), les écarts concernant les besoins en mémoire sont plus faibles pour HGA et sont même très proches des besoins de Vcap lorsque le degré moyen augmente. Sur la figure 5.12, avec 10% de nœuds ayant connaissance de leur position, les besoins en mémoire sont beaucoup plus faibles pour HGA et sont même inférieurs à ceux de Vcap lorsque le degré dépasse 40.

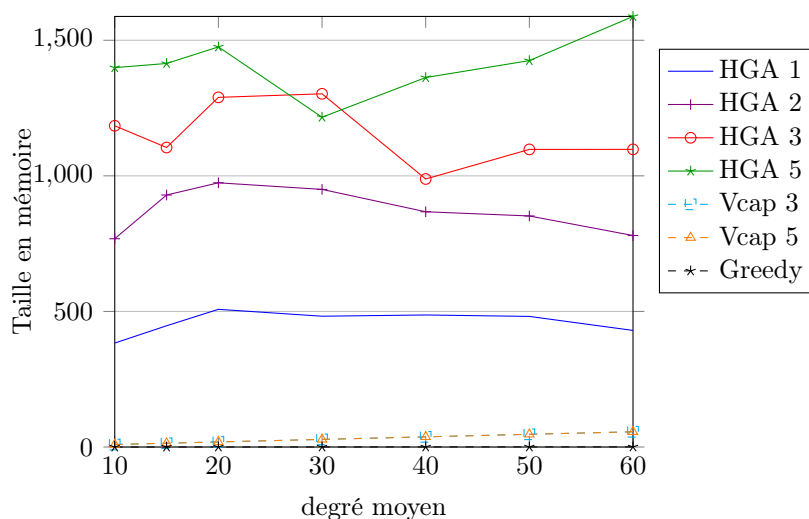


FIGURE 5.10 – Taille moyenne de la table de routage (pour HGA) ou de voisinage (pour greedy et Vcap) pour les différents algorithmes de routage avec 1% des nœuds ayant leur information de position pour HGA.

Pour Greedy, la taille en mémoire représente uniquement le nombre de voisins ayant connaissance de leur position pour chaque nœud. La taille mémoire importante nécessaire à HGA s'explique par le fait que dans les simulations, toutes les routes

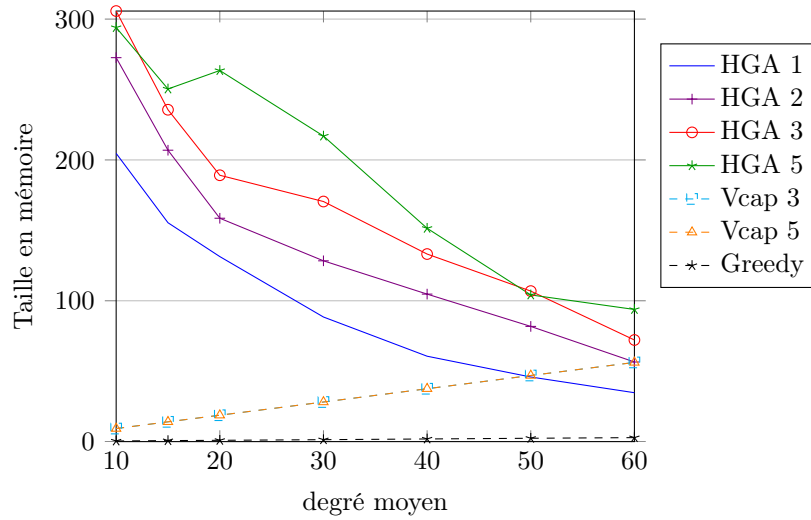


FIGURE 5.11 – Taille moyenne de la table de routage ou de voisinage pour les différents algorithmes de routage avec **5%** des nœuds ayant leur information de position pour HGA.

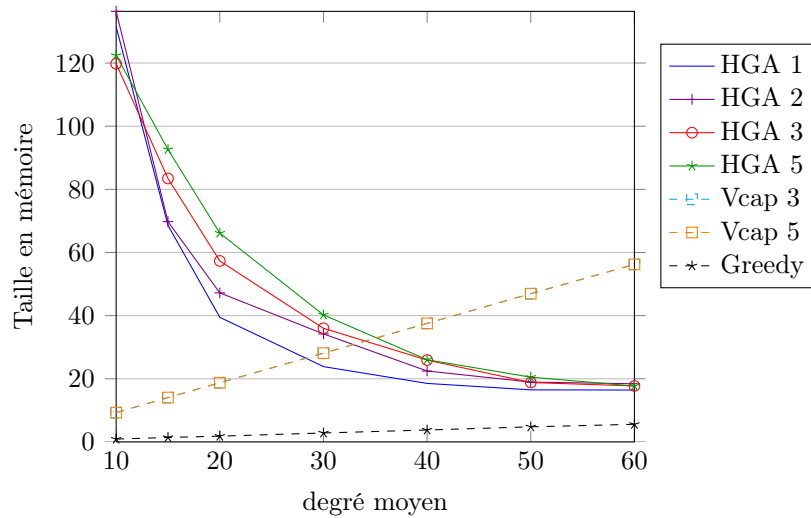


FIGURE 5.12 – Taille moyenne de la table de routage ou de voisinage pour les différents algorithmes de routage avec **10%** des nœuds ayant leur information de position pour HGA.

sont sauvegardées jusqu'à la fin de la simulation. Il sera possible de limiter la taille mémoire en "oubliant" les routes les plus anciennes avec, comme contrepartie, un trafic plus élevé en messages de contrôle. On pourra aussi noter que les conditions d'expérimentations sont exigeantes en ce qui concerne les besoins en mémoire. En

effet, les sources et destinations de paquets de données sont choisies aléatoirement et la fréquence d'émission est élevée. Dans une application courante, les destinations des messages sont souvent limitées voire uniques dans le cas de collecte de données. Le nombre de sources peut aussi être limité ou unique dans le cas d'envoi d'ordres à des actionneurs.

5.5 Auto-configuration du paramètre k

Nous avons vu dans la section 5.4 que le paramètre k , correspondant à la profondeur de recherche de routes de l'algorithme HGA, pouvait influencer le taux de livraison. Le nombre de messages de contrôle et par conséquent l'énergie consommée par les nœuds est aussi affectée par ce paramètre. Le paramètre k n'est pas le seul responsable des performances du réseau, en effet les propriétés du réseau déterminent elles aussi les performances finales. Il n'est pas toujours facile, ou possible, de connaître les propriétés du réseau avant son déploiement. De même, on peut souhaiter utiliser la même application sur des réseaux de nature différentes, comme des villes denses d'une part et d'autres moins denses d'autre part, dans le cadre des villes intelligentes par exemple. Dans ce contexte on pourra s'intéresser à l'auto-configuration du paramètre k afin d'obtenir l'auto-organisation du réseau.

On peut noter que si la valeur du paramètre k est supérieure à la distance entre la source et la destination, le paquet sera assurément délivré (hors pertes dues au lien physique). La mesure du diamètre du réseau pourrait permettre de déduire la distance moyenne entre source et destination, ou la distance maximale entre source et destination avec un indice de confiance. L'indice de confiance permettrait de déduire un taux de livraison théorique pour le réseau. Grâce à des techniques similaires à celles utilisées par Vcap [113] décrit en section 5.2 il est possible de déterminer le diamètre du réseau. De cette manière il serait possible de déduire une valeur pour le paramètre k adaptée au réseau. Toutefois cette méthode nécessite d'inonder plusieurs fois le réseau au démarrage or, comme décrit en section 5.2, nous souhaitons l'éviter pour garantir le passage à l'échelle.

Une autre possibilité serait de corréler le degré moyen en nombre de nœuds positionnés au taux de livraison. Grâce à un jeu de simulation plus ciblé sur cette mesure il serait possible de définir le degré moyen des nœuds positionnés pour un taux de livraison donné. Par exemple, si on vise un taux de livraison supérieur à 90% et qu'un nœud souhaitant envoyer un paquet de données connaît 2 voisins positionnés, on pourra déduire de la figure 5.5 que la variante HGA-1 est suffisante. La figure 5.6 permet de déduire les mêmes conclusions. En effet dans un réseau dont 5% des nœuds sont positionnés, si le degré moyen est 40, un nœud aura en moyenne 2 voisins positionnés. De même dans un réseau dont 10% de nœuds sont positionnés, avec un degré moyen de 20, un nœud aura aussi en moyenne 2 voisins positionnés. Or on observe sur la figure 5.5 (resp. 5.6) que le taux de livraison de HGA-1 est supérieur à 90% pour un degré de 40 (resp. 20) et plus.

Lorsque le nombre de nœuds positionnés dans le voisinage du nœuds source est faible, voire nul, il peut être difficile de déduire la profondeur la plus adaptée. Pour cela il peut être intéressant de prendre la décision de transmettre ou non le paquet RREQ sur les nœuds intermédiaires. Le paquet RREQ contiendrait alors quelques informations supplémentaires permettant de calculer le degré moyen en nombre de nœuds positionnés. Grâce à cette information la retransmission de la RREQ serait réévaluée sur chaque nœuds en fonction du taux de livraison souhaité.

5.6 Conclusion

Nous proposons dans ce chapitre une solution de routage géographique inédite, permettant d'utiliser la position des nœuds du réseau quand elle est disponible et de s'en passer lorsqu'elle ne l'est pas. À notre connaissance, il n'existe pas de solution dans la littérature reposant sur les mêmes hypothèses. En effet, on retrouve des solutions de routage géographique dans lesquelles l'ensemble des nœuds connaissent leur position et des solutions utilisant des coordonnées virtuelles où aucun nœud ne connaît sa position. Nous proposons une solution intermédiaire, tirant parti de la position réelle des nœuds lorsqu'elle est disponible.

La solution que nous proposons, HGA, offre des performances nettement supérieures à la solution de routage géographique Greedy si une partie des nœuds n'a pas connaissance de sa position. Nous comparons aussi HGA avec une solution de routage utilisant des coordonnées virtuelles, Vcap, et montrons que pour différents scénarios d'utilisation, notre solution offre de meilleures performances, avec un surcoût limité.

Pour les travaux futurs, la validation de nos résultats avec des couches physiques et MAC réalistes est envisagée. L'utilisation de plates-formes réalistes comme FIT ou SmartSantander permettrait de tester ces aspects en plus d'offrir un environnement réaliste. Par ailleurs la validation des résultats sur des topologies ciblant l'application telle que définie dans *BinThatThinks* est aussi prévue.

L'étude de la mobilité des nœuds est aussi un aspect intéressant, elle permettrait de caractériser notamment le délai nécessaire avant expiration des routes en fonction de la vitesse de déplacement des nœuds. La comparaison avec les algorithmes de routage utilisant des coordonnées virtuelles devrait aussi apporter de nouveaux arguments en faveur de HGA puisque ces algorithmes nécessiteraient une reconstruction fréquente du système de coordonnées virtuelles, et donc, une augmentation du nombre de messages échangés dans le réseau.

Nous pourrions étendre notre proposition en supprimant l'usage des messages HELLO, comme proposé dans l'algorithme GDBF [122], ce qui permettrait de réduire le nombre de messages échangés dans le réseau et donc d'augmenter sa durée de vie notamment lorsque le trafic est faible.

Nous avons proposé une approche réactive, permettant de faire face à des changements dans la topologie du réseau. Ainsi notre proposition est applicable lorsque

certaines nœuds du réseau sont mobiles. Il serait intéressant de comparer notre proposition avec une variante proactive dans laquelle chaque nœud maintient une connaissance de son voisinage à k -sauts en lieu et place de l'utilisation de RREQ à k -sauts. Nous pourrions étudier le nombre de messages de contrôle de l'une et l'autre des solutions et trouver le seuil pour lequel il est plus intéressant d'utiliser l'une ou l'autre variante. Nous pourrions aussi mesurer le temps de réactions des différentes variantes et son impact sur le taux de livraison.

L'algorithme que nous proposons offre une solution intéressante dans le projet *BinThatThinks* et plus généralement dans les villes intelligentes. L'utilisation d'un réseau de capteurs fixes connaissant leur position, sur les poubelles publiques et les points de collecte par exemple, pourrait offrir les informations de position nécessaires aux capteurs disposés sur les poubelles. Cela permettrait de réduire les coûts (aucun récepteur GPS nécessaire) tout en offrant de bonnes performances pour le routage des données.

Conclusion et perspectives

Sommaire

6.1	Conclusion	103
6.2	Perspectives	104
6.3	Conclusion personnelle	105

6.1 Conclusion

J'ai, à travers cette thèse, proposé des solutions pour l'auto-organisation des réseaux de capteurs sans-fil dans le but de les appliquer au contexte des villes intelligentes. Je propose d'abord une analyse de l'impact de différentes topologies sur les performances des algorithmes pour réseaux de capteurs sans-fil. En analysant différents algorithmes de routage géographique sur différentes topologies, je montre une incidence non négligeable de la topologie sur les performances des réseaux de capteurs sans-fil. Ces travaux mettent en avant la nécessité de faire les bons choix dans la mise en place de protocoles pour les villes intelligentes. Ils ont été validés en analysant différents paramètres pertinents et communément utilisés dans le cadre des réseaux de capteurs sans-fil pour les villes intelligentes et par de nombreuses simulations et des expériences sur plate-forme.

En me basant sur cette analyse j'ai ensuite proposé une solution adressant le problème de la remontée d'informations dans un réseau de capteurs sans-fil équipés de GPRS. L'optimisation de la durée de vie du réseau a été la principale préoccupation dans la recherche de cette solution. Je propose une famille d'algorithmes, **BLAC**, qui surpasse les solutions de la littérature en utilisant des paramètres réalistes. En effet, grâce à la conclusion issue de l'étude sur les topologies, je compare notre proposition à celles de la littérature sur des topologies réalistes et en utilisant des paramètres de consommation issues du matériel existant. Notre proposition apporte des réponses adaptées et directement applicables à la remontée d'informations dans un réseau de capteurs sans-fil pour les villes intelligentes.

Enfin, je propose une solution de routage hybride, toujours adaptée au contexte des villes intelligentes, permettant d'être utilisable à très large échelle. Équiper chaque capteur sans-fil d'une ville d'un récepteur GPS est difficilement envisageable, de même que paramétrer chacun d'eux avec leur position. Par ailleurs, même si on envisageait ces solutions, il est idéaliste de penser que chaque capteur d'une ville

sera en mesure de recevoir correctement un signal GPS. Pour cela je propose l'algorithme **HGA** permettant de répondre à cette contrainte. **HGA** offre des performances meilleures ou équivalentes aux solutions offertes par la littérature, selon les propriétés du réseau. Toutefois, l'étude des performances de cet algorithme devrait être poursuivie en analysant d'autres topologies, proches de celles rencontrées dans une ville, et en conduisant des expérimentations sur plate-forme et en conditions réelles.

À travers ces travaux, j'ai essayé d'apporter une solution adressant différents problèmes pouvant survenir dans les réseaux de capteurs appliqués aux villes intelligentes.

6.2 Perspectives

Les contributions présentées dans cette thèse peuvent bénéficier d'améliorations, d'extensions et d'élargissements. Les travaux liés à l'impact de la topologie pourraient être étendus à des topologies plus variées encore, des algorithmes différents, voire éventuellement à d'autres couches de la pile de communication. Il serait aussi pertinent d'étendre les expérimentations sur des plates-formes de simulation plus vastes. La combinaison des travaux de *clustering* et de routage présentés aux chapitres 4 et 5 pourrait apporter une réponse intéressante au routage dans certaines conditions dans les villes intelligentes. En effet, dans un contexte où l'ensemble des nœuds est équipé d'un récepteur GPS, sachant que ce périphérique est gourmand en énergie, il est possible de mutualiser cet équipement pour étendre la durée de vie du réseau. Combiner les capacités de *clustering* de l'algorithme **BLAC**, appliqué au GPS plutôt qu'au GPRS, avec l'algorithme de routage **HGA**, utilisant des informations partielles de positions, apporterait une solution pour l'autonomie et le routage dans les réseaux de capteurs sans-fil appliqué aux villes intelligentes.

Les contributions de cette thèse apportent des solutions à la mise en œuvre du réseau de capteurs sans-fil envisagée dans le projet *BinThatThinks*. La mise en pratique de ces contributions dans un démonstrateur grandeur nature dans une ville, voire pour un déploiement national, serait une concrétisation pour ces travaux.

En élargissant le concept de ville intelligente, on peut se poser la question d'un *pays intelligent* et même d'une *planète intelligente*. Dans une époque où la mondialisation règne, cette question peut sembler légitime. En prenant en compte la seule problématique des transports, on comprend aisément qu'un problème technique sur un train à Lyon peut entraîner un retard sur son arrivée à Paris, engendrant à son tour des retards sur les trains en correspondance. On peut étendre cette réflexion aux vols internationaux et leurs impacts d'un pays à l'autre. On se souvient de l'éruption du volcan Islandais Eyjafjöll en 2010, qui paralysa les avions d'une partie de la planète¹. Les technologies pour la surveillance de volcans grâce à des réseaux de capteurs sans-fil multi-sauts existent, leur intégration dans un système plus large

1. www.lexpress.fr/actualite/environnement/nuage-de-cendres-sur-1-europe_891427.html

permettra une utilisation globale et automatique des données qu'il génèrent. D'autre part, aujourd'hui la Suède achète des déchets² à ses pays voisins, déchets qui sont issus de produits fabriqués de l'autre côté de la planète. Si les technologies sont disponibles pour la traçabilité des produits et déchets, leur mise en place est plus problématique. *BinThatThinks* apportera des solutions pour la gestion de la collecte grâce à des réseaux de capteurs mais la gestion de la traçabilité globale pose d'autres problématiques.

Dans ce contexte, peut-on apporter des solutions à l'échelle mondiale pour réduire la pollution, diminuer l'utilisation des ressources et faire profiter l'économie ? Si de tels concepts voyaient le jour, ce seraient les défis qu'ils auraient à relever.

6.3 Conclusion personnelle

Faire de la recherche dans le cadre du projet *BinThatThinks* avec des partenaires impliqués directement dans les solutions qui équiperont les villes de demain m'a permis de me forger ma définition de la ville intelligente et de ce que seront les villes du futur. Évidemment les nouvelles technologies joueront un rôle clé dans les villes du futur et les réseaux de capteurs sans-fil y occuperont une place de choix. Ces équipements modernes permettront d'améliorer le quotidien de tous. Cette amélioration du quotidien, passe avant tout, à mon sens, par une consommation plus raisonnée des différentes énergies disponibles et par le développement et la maîtrise des énergies renouvelables.

Les habitations à énergie positive donnent un bon exemple et offrent une analogie intéressante de ce que seront les villes intelligentes. Elles permettent à leurs habitants d'être autonomes au niveau énergétique, offrent des conditions de vie agréables et ont un faible impact sur l'environnement. Elles sont réalisables grâce aux progrès des matériaux isolants, aux techniques d'architecture innovantes, aux technologies telles que les ventilations mécaniques à double flux, etc. Le bon fonctionnement des différents équipements de ce type d'habitation est rendu possible grâce à l'utilisation de solutions de domotique, jouant un véritable rôle de chef d'orchestre pour toutes ces solutions. En effet, la domotique permet de contrôler la ventilation, d'orienter automatiquement des brises-soleil, de surveiller la température et bien d'autres tâches encore.

Je pense que les villes intelligentes peuvent être une extension de ce concept, toutefois les contraintes sont bien plus fortes. Les villes de demain, peuvent être, à l'image de ces habitations, autonomes en énergie, agréables à vivre et avoir un faible impact sur l'environnement. Cependant, l'échelle entre une ville et une habitation est radicalement différente, de plus les villes doivent concilier avec l'existant, au contraire des habitations souvent construites à partir de rien. Si l'utilisation de technologies sans-fil n'est pas systématique dans les habitations à énergie positive,

2. <http://ecologie.blog.lemonde.fr/2012/09/22/a-force-de-trop-recycler-la-suede-doit-importer-des-dechets/>

elles seront beaucoup plus présentes à l'échelle d'une ville où il est plus délicat de déployer de nouvelles infrastructures. D'autre part, la taille des villes (Los Angeles mesure 71 km du nord au sud) rend l'utilisation de protocoles multi-sauts quasiment inévitable. Enfin, dans une habitation, le nombre d'acteurs est souvent limité (le propriétaire dans le cas d'une maison individuelle) or, dans une ville, les acteurs sont nombreux. Entre les habitants, la municipalité, les fournisseurs de service (fournisseurs d'accès internet, compagnie d'électricité, ramassage de ordures...) il est nécessaire de créer une synergie pour tirer au mieux parti des différentes installations.

Pour que le concept des villes intelligentes puisse exploiter toutes ses possibilités, il est nécessaire que les différents acteurs de celles-ci collaborent, en partageant leurs ressources. Il est, selon moi, nécessaire de favoriser, voire obliger via une législation adaptée, les différents acteurs à utiliser les équipements existants plutôt que d'en déployer de nouveaux. De même, ils devraient mettre leurs ressources à disposition des autres acteurs. La création d'un réseau ouvert, peut-être à l'image d'Internet, me semble être la voie à suivre. Les initiatives gouvernementales comme l'Open Data³ peuvent fournir un élément de réponse, de même que l'interopérabilité. Le détail de ces éléments sort largement du cadre de cette thèse, les problématiques qui pourront-être rencontrées étant d'ordre politique, économique ou encore relevant de la sécurité en plus des problématiques techniques.

Le projet *BinThatThinks* entend apporter une brique dans la construction de la ville intelligente. *BinThatThinks* fournit une architecture matérielle et logicielle pour améliorer le recyclage, réduire la quantité de déchets, les besoins énergétiques et diminuer l'empreinte carbone. Une question légitime peut se poser : cette démarche est-elle la bonne ? En effet plutôt que de fabriquer de nouveaux appareils pour équiper les poubelles de demain, n'est-il pas plus judicieux de concentrer les efforts pour réduire la quantité de déchets en amont ? En consommant moins et mieux ? Plus généralement, plutôt que de fournir de nouveaux services pour réduire notre impact sur l'environnement n'est-il pas préférable de réduire nos besoins ? Une fois de plus, ces questions dépassent largement le cadre de cette thèse et pourront même relever de débats de société.

3. <http://www.data.gouv.fr>

Publications

Journaux

1. Tony DUCROCQ, Michaël HAUSPIE et Nathalie MITTON : Balancing energy consumption in clustered wireless sensor networks. *ISRN Sensor Networks*, septembre 2013.

Conférences internationales

1. Tony DUCROCQ, Nathalie MITTON et Michaël HAUSPIE : Energy-based Clustering for Wireless Sensor Network Lifetime Optimization. *In Proceedings of the Wireless Communications and Networking Conference, WCNC*, Shanghai, China, avril 2013. IEEE.
2. Clément BURIN DES ROSIERS, Guillaume CHELIUS, Tony DUCROCQ, Eric FLEURY, Antoine FRABOULET, Antoine GALLAIS, Nathalie MITTON, Thomas NOËL et Julien VANDAËLE : Using SensLAB as a First Class Scientific Tool for Large Scale Wireless Sensor Network Experiments. *In Proceedings of the 10th International IFIP TC 6 Networking Conference, NETWORKING*, pages 241–253, Valencia, Spain, avril 2011. IFIP. (Publication en dehors du cadre de la thèse).

Conférences nationales

1. Tony DUCROCQ, Nathalie MITTON et Michaël HAUSPIE : Clustering pour l'optimisation de la durée de vie des réseaux de capteurs sans fil. *In Proceedings of the 14èmes Rencontres Francophones sur les Aspects Algorithmiques des Télécommunications*, AlgoTel, La Grande Motte, France, mai 2012.

Autres publications

1. Tony DUCROCQ, Michaël HAUSPIE et Nathalie MITTON : On the Impact of Network Topology on Wireless Sensor Networks Performances - Illustration with Geographic Routing. *In Proceedings of the 10th International Symposium on Performance Evaluation of Wireless Ad Hoc, Sensor, and Ubiquitous Networks, PE-WASUN*, Barcelona, Spain, novembre 2013. ACM. Poster.
2. Clément BURIN DES ROSIERS, Guillaume CHELIUS, Tony DUCROCQ, Eric FLEURY, Antoine FRABOULET, Antoine GALLAIS, Nathalie MITTON, Thomas NOËL, Erkan VALENTIN et Julien VANDAËLE : Two demos using SensLAB : Very Large Scale Open WSN Testbed. *In Proceedings of the 7th International*

Conference on Distributed Computing in Sensor Systems, DCOSS, Barcelona, Spain, juin 2011. IEEE. Démo, (Publication en dehors du cadre de la thèse).

3. Tony DUCROCQ, Julien VANDAËLE, Nathalie MITTON et David SIMPLOT-RYL : Large scale geolocalization and routing experimentation with the SensLAB test-bed. *In Proceedings of the 7th International Conference on Mobile Ad-hoc and Sensor Systems*, MASS, San Francisco, CA, USA, novembre 2010. IEEE. Démo, (Publication en dehors du cadre de la thèse).

Références

- [1] T. S. RAPPAPORT *et al.*, *Wireless communications : principles and practice*, vol. 2. Prentice Hall PTR New Jersey, 1996.
- [2] T. C. MARTIN, *The inventions, researches and writings of Nikola Tesla*. Lindsay Publications, 1988.
- [3] G. PUJOLLE, O. SALVATORI et J. NOZICK, *Les réseaux*. Editions Eyrolles, 2010.
- [4] C. PERKINS *et al.*, *Ad hoc networking*. Addison-Wesley Professional, 2001.
- [5] I. AKYILDIZ, W. SU, Y. SANKARASUBRAMANIAM et E. CAYIRCI, « Wireless sensor networks : a survey », *Computer Networks*, vol. 38, no. 4, p. 393 – 422, 2002.
- [6] L. YU, N. WANG et X. MENG, « Real-time forest fire detection with wireless sensor networks », in *Proceedings of the International Conference on Wireless Communications, Networking and Mobile Computing, WCNC*, (New Orleans, LA, USA), p. 1214–1217, IEEE, 2005.
- [7] E. A. BASHA, S. RAVELA et D. RUS, « Model-based monitoring for early warning flood detection », in *Proceedings of the 6th ACM conference on Embedded network sensor systems*, SenSys'08, (Raleigh, NC, USA), p. 295–308, ACM, 2008.
- [8] R. LIN, Z. WANG et Y. SUN, « Wireless sensor networks solutions for real time monitoring of nuclear power plant », in *Proceedings of the 5th World Congress on Intelligent Control and Automation*, WCICA, (Hangzhou, China), p. 3663–3667, IEEE, 2004.
- [9] M. TUBAISHAT, Y. SHANG et H. SHI, « Adaptive traffic light control with wireless sensor networks », in *Proceedings of 4th Annual IEEE Consumer Communications and Networking Conference*, CCNC, (Las Vegas, NV, USA), p. 187–191, IEEE, 2007.
- [10] G. DENARDIN, C. BARRIQUELLO, A. CAMPOS, R. PINTO, M. DALLA COSTA et R. do PRADO, « Control network for modern street lighting systems », in *Proceedings of International Symposium on Industrial Electronics*, ISIE, (Gdansk, Poland), p. 1282–1289, IEEE, 2011.
- [11] W. ARBAB, F. IJAZ, T. M. YOON et C. LEE, « A USN based Automatic waste collection system », in *Proceedings of the 14th International Conference on Advanced Communication Technology*, ICACT, (PyeongChan, South Korea), p. 936–941, 2012.
- [12] A. WHEELER, « Commercial Applications of Wireless Sensor Networks Using ZigBee », *IEEE Communications Magazine*, vol. 45, no. 4, p. 70–77, 2007.

- [13] A. MILENKOVIĆ, C. OTTO et E. JOVANOVIĆ, « Wireless sensor networks for personal health monitoring : Issues and an implementation », *IEEE Computer Communications*, vol. 29, no. 13–14, p. 2521 – 2533, 2006.
- [14] G. CHELIUS, C. BRAILLON, M. PASQUIER, N. HORVAIS, R. GIBOLLET, B. ESPIAU et C. COSTE, « A Wearable Sensor Network for Gait Analysis : A Six-Day Experiment of Running Through the Desert », *IEEE/ASME Transactions on Mechatronics*, vol. 16, no. 5, p. 878–883, 2011.
- [15] G. ZHOU, T. HE, S. KRISHNAMURTHY et J. A. STANKOVIC, « Impact of radio irregularity on wireless sensor networks », in *Proceedings of the 2nd international conference on Mobile systems, applications, and services*, MobiSys, (New York, NY, USA), p. 125–138, ACM, 2004.
- [16] F. INGELREST, N. MITTON et D. SIMPLOT-RYL, « A Turnover based Adaptive HELLO Protocol for Mobile Ad Hoc and Sensor Networks », in *Proceedings of the 15th International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems*, MASCOTS, (Istanbul, Turkey), p. 9–14, 2007.
- [17] M. BUETTNER, G. V. YEE, E. ANDERSON et R. HAN, « X-MAC : a short preamble MAC protocol for duty-cycled wireless sensor networks », in *Proceedings of the 4th international conference on Embedded networked sensor systems*, SenSys '06, (New York, NY, USA), p. 307–320, ACM, 2006.
- [18] N. NIKAEIN, H. LABIOD et C. BONNET, « DDR-Distributed Dynamic Routing algorithm for mobile ad hoc networks », in *Proceedings of the 1st ACM international symposium on Mobile ad hoc networking & computing*, MobiHoc, (Boston, MA, USA), p. 19–27, ACM, 2000.
- [19] W. HEINZELMAN, A. CHANDRAKASAN et H. BALAKRISHNAN, « Energy-efficient communication protocol for wireless microsensor networks », in *Proceedings of the 33rd Hawaii International Conference on System Sciences*, HICSS, (Maui, HI, USA), p. 3005–3014, Computer Society Press, 2000.
- [20] C. PERKINS et E. ROYER, « Ad-hoc on-demand distance vector routing », in *Proceedings of the 2nd Workshop on Mobile Computing Systems and Applications*, WMCSA, (New Orleans, LA, USA), p. 90–100, IEEE, 1999.
- [21] T. CLAUSEN, P. JACQUET, C. ADJIH, A. LAOUITI, P. MINET, P. MUHLETHALER, A. QAYYUM et L. VIENNOT, « Optimized Link State Routing Protocol (OLSR) », rap. tech., 2003. Network Working Group.
- [22] G. G. FINN, « Routing and addressing problems in large metropolitan-scale internetworks », rap. tech., DTIC Document, 1987.
- [23] P. BOSE, P. MORIN, I. STOJMENOVIC et J. URRUTIA, « Routing with guaranteed delivery in ad hoc wireless networks », *Wireless Networks*, vol. 7, no. 6, p. 609–616, 2001.
- [24] H. KALOSHA, A. NAYAK, S. RUHRUP et I. STOJMENOVIC, « Select-and-Protest-Based Beaconless Georouting with Guaranteed Delivery in Wireless

- Sensor Networks », in *Proceedings of the 27th IEEE Conference on Computer Communications*, INFOCOM, (Phoenix, AZ, USA), p. 995–1003, IEEE, 2008.
- [25] A. CARAGLIU, C. DEL BO, P. NIJKAMP *et al.*, *Smart cities in Europe*. Vrije Universiteit, Faculty of Economics and Business Administration, 2009.
- [26] M. AL-HADER, A. RODZI, A. SHARIF *et al.*, « Smart City Components Architecture », in *Proceedings of the International Conference on Computational Intelligence, Modelling and Simulation*, CSSim, (Brno, Czech Republic), p. 93–97, IEEE, 2009.
- [27] L. FILIPPONI, A. VITALETTI, G. LANDI, V. MEMEO, G. LAURA *et al.*, « Smart City : An Event Driven Architecture for Monitoring Public Spaces with Heterogeneous Sensors », in *Proceedings of the 4th International Conference on Sensor Technologies and Applications*, SENSORCOMM, (Venice, Italy), p. 281–286, CPS, 2010.
- [28] http://www.ibm.com/smarterplanet/us/en/smarter_cities/overview/.
- [29] B. KARP *et al.*, « GPSR : greedy perimeter stateless routing for wireless networks », in *Proceedings of the 6th annual international conference on Mobile computing and networking*, MobiCom, (New York, NY, USA), p. 243–254, ACM, 2000.
- [30] S. AMIN *et al.*, « Toward a smart grid : power delivery for the 21st century », *IEEE Power and Energy Magazine*, vol. 3, no. 5, p. 34–41, 2005.
- [31] M. WIERING, J. v. VEENEN, J. VREEKEN *et al.*, « Intelligent traffic light control », *UU-CS*, no. 2004-029, 2004.
- [32] T. DELOT, S. ILARRI, S. LECOMTE *et al.*, « Sharing with caution : Managing parking spaces in vehicular networks », *Mobile Information Systems*, vol. 9, no. 1, p. 69–98, 2013.
- [33] M. DOHLER, D. BARTHEL, T. WATTEYNE *et al.*, « Routing requirements for urban low-power and lossy networks », 2009. RFC 5548.
- [34] G. P. HANCKE, B. d. C. e. SILVA *et al.*, « The Role of Advanced Sensing in Smart Cities », *Sensors*, vol. 13, no. 1, p. 393–425, 2012.
- [35] L. SANCHEZ, J. GALACHE, V. GUTIERREZ, J. HERNANDEZ, J. BERNAT, A. GLUHAK *et al.*, « SmartSantander : The meeting point between Future Internet research and experimentation and the smart cities », in *Proceedings of the Conference of Future Network Mobile Summit*, FutureNetw, (Warsaw, Poland), p. 1–8, IIMC Ltd Ireland, 2011.
- [36] <http://www.sensorsmag.com/wireless-applications/smart-city-project-santander-11152>.
- [37] <http://www.npr.org/blogs/parallels/2013/06/04/188370672/Sensors-Transform-Old-Spanish-Port-Into-New-Smart-City>.

- [38] <http://www.smartplanet.com/blog/report/santander-test-bed-for-smart-cities-and-open-data-policies/2001>.
- [39] <http://www.smartsantander.eu>.
- [40] <http://www.cooking-hacks.com/index.php/shop/waspmote/waspmote-smart-cities-sensor-kit.html>.
- [41] http://www.libelium.com/downloads/documentation/waspmote_technical_guide.pdf.
- [42] <http://www.economie.grandlyon.com/smart-city-lyon-france.346.0.html>.
- [43] <http://www.smartcity.ae/>.
- [44] <http://amsterdamsmartcity.com/>.
- [45] B. CHOWDHURY et M. CHOWDHURY, « RFID-based real-time smart waste management system », in *Proceedings of the Australasian Telecommunication Networks and Applications Conference*, ATNAC, (Christchurch, New Zealand), p. 175–180, 2007.
- [46] <http://www.betty-box.com/>.
- [47] R. KAHNAT, J. KIM, M. XU, B. ALLENBY, E. WILLIAMS et P. ZHANG, « Exploring e-waste management systems in the United States », *Resources, Conservation and Recycling*, vol. 52, no. 7, p. 955 – 964, 2008.
- [48] C. BINDER et S. DOMNITCHEVA, « When Waste Becomes Intelligent : Assessing the Environmental Impact of Microchip Tagging », in *Proceedings of the SETAC/ISIE/Swiss Discussion Forum 2003*, (Lausanne, Switzerland), 2003.
- [49] D. SCHOCH et M. SALA, « BIN IT!—The Intelligent Waste Management System », *Wireless Sensor Network*, p. 71, 2005.
- [50] <http://www.discoverrrfid.org/fr/ce-qui-est-possible/travailler-dans-de-meilleures-conditions/la-rfid-dans-les-services-de-ramassage-des-ordures-et-les-egouts.html>.
- [51] <http://www.parolesdelus.com/le-magazine/dossier/une-petite-revolution-lintegration-du-nfc-dans-les-telephones-mobiles/rfid-integ>.
- [52] <http://rfid.comprendrechoisir.com/comprendre/tracabilite-des-dechets>.
- [53] <http://www.actu-environnement.com/ae/news/recyclage-marquage-triman-consignes-tri-REP-projet-loi-19373.php4>.
- [54] http://www.solutions-rfid.com/collecte_bac_dechet_materiel_rfid.php?idmat=2.
- [55] <http://www.rfidhy.com/Applications/waste-management/>.
- [56] <http://energie.sia-partners.com/20090510/les-telecommunications-au-service-de-la-gestion-des-dechets/>.

- [57] T. NUORTIO, J. KYTÖJOKI, H. NISKA et O. BRÄYSY, « Improved route planning and scheduling of waste collection and transport », *Expert Systems with Applications*, vol. 30, no. 2, p. 223 – 232, 2006.
- [58] <http://www.journaldunet.com/economie/services/enevo.shtml>.
- [59] <http://www.ijinus.com/fr/presentation/1142592934456>.
- [60] <http://www.01net.com/editorial/542128/optimiser-la-collecte-des-dechets-avec-des-capteurs-radio/>.
- [61] <http://www.sigrenea.com>.
- [62] A. FRABOULET, G. CHELIUS et E. FLEURY, « Worldsens : Development and Prototyping Tools for Application Specific Wireless Sensors Networks », in *Proceedings of the 6th International Symposium on Information Processing in Sensor Networks*, IPSN, (Cambridge, MA, USA), p. 176–185, ACM, 2007.
- [63] C. Burin des ROZIERES, G. CHELIUS, T. DUCROCQ, E. FLEURY, A. FRABOULET, A. GALLAIS, N. MITTON, T. NOEL et J. VANDAELE, « Using SensLAB as a First Class Scientific Tool for Large Scale Wireless Sensor Network Experiments », in *Proceedings of the 10th International IFIP TC 6 Networking Conference*, NETWORKING, (Valencia, Spain), p. 147–159, Springer Berlin Heidelberg, 2011.
- [64] <http://www.senslab.info>.
- [65] <http://www.ti.com/product/msp430f1611>.
- [66] <http://www.ti.com/product/cc2420>.
- [67] <http://www.ti.com/product/cc1100>.
- [68] S. DHILLON et K. CHAKRABARTY, « Sensor placement for effective coverage and surveillance in distributed sensor networks », in *Proceedings of Wireless Communications and Networking*, WCNC, (New Orleans, LA, USA), p. 1609–1614, IEEE, 2003.
- [69] K. DASGUPTA, M. KUKREJA et K. KALPAKIS, « Topology-aware placement and role assignment for energy-efficient information gathering in sensor networks », in *Proceedings of the 8th International Symposium on Computers and Communication*, ISCC, (Kemer, Turkey), p. 341–348, IEEE, 2003.
- [70] Y. WANG, C. HU et Y. TSENG, « Efficient Placement and Dispatch of Sensors in a Wireless Sensor Network », *IEEE Transactions on Mobile Computing*, vol. 7, no. 2, p. 262 –274, 2008.
- [71] M. ERDELJ, T. RAZAFINDRALAMBO et D. SIMPLOT-RYL, « Covering Points of Interest with Mobile Sensors », *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 1, p. 32–43, 2013.
- [72] M. YOUNIS et K. AKKAYA, « Strategies and techniques for node placement in wireless sensor networks : A survey », *Ad Hoc Networks*, vol. 6, no. 4, p. 621–655, 2008.

- [73] P. RADOSLAVOV, H. TANGMUNARUNKIT, H. YU, R. GOVINDAN, S. SHENKER et D. ESTRIN, « On Characterizing Network Topologies and Analyzing Their Impact on Protocol Design », Rap. tech. 00-731, University of Southern California, 2000.
- [74] Z. LI et P. MOHAPAIRA, « The impact of topology on overlay routing service », in *Proceedings of Computer and Communications Societies*, INFOCOM, (Hong Kong), p. 408–418, IEEE, 2004.
- [75] M. BHARDWAJ, T. GARNETT et A. P. CHANDRAKASAN, « Upper bounds on the lifetime of sensor networks », in *Proceedings of International Conference on Communications*, ICC, (Helsinki, Finland), p. 785–790, IEEE, 2001.
- [76] M. ISHIZUKA et M. AIDA, « Performance study of node placement in sensor networks », in *Proceedings of the 24th International Conference on Distributed Computing Systems Workshops*, ICDCS, (Tokyo, Japan), p. 598–603, IEEE, 2004.
- [77] V. VASSILIOU et C. SERGIOU, « Performance Study of Node Placement for Congestion Control in Wireless Sensor Networks », in *Proceedings of the 3rd International Conference on New Technologies, Mobility and Security*, NTMS, (Cairo, Egypt), p. 1–8, IEEE, 2009.
- [78] D. PUCCINELLI, O. GNAWALI, S. YOON, S. SANTINI, U. COLESANTI, S. GIORDANO et L. GUIBAS, « The impact of network topology on collection performance », in *Wireless Sensor Networks*, vol. 6567 in *Lecture Notes in Computer Science*, p. 17–32, Springer Berlin Heidelberg, 2011.
- [79] L. KARIM, T. E. SALT, N. NASSER et Q. H. MAHMOUD, « The significant impact of a set of topologies on wireless sensor networks », *EURASIP Journal on Wireless Communications and Networking*, vol. 2012, no. 1, p. 1–13, 2012.
- [80] H. TAKAGI et L. KLEINROCK, « Optimal Transmission Ranges for Randomly Distributed Packet Radio Terminals », *IEEE Transactions on Communications*, vol. 32, no. 3, p. 246–257, 1984.
- [81] T.-C. HOU et V. LI, « Transmission Range Control in Multihop Packet Radio Networks », *IEEE Transactions on Communications*, vol. 34, no. 1, p. 38–44, 1986.
- [82] E. KRANAKIS, H. SINGH et J. URRUTIA, « Compass Routing on Geometric Networks », in *Proceedings of the 11th Canadian Conference on Computational Geometry*, CCCG, (Vancouver, Canada), p. 51–54, 1999.
- [83] G. CHELIUS, A. FRABOULET et E. FLEURY, « WorldSens : system tools for embedded sensor networks », in *Proceedings of 27th Real-Time Systems Symposium*, RTSS, (Rio de Janeiro, Brazil), p. 4, WIP Session, 2006.
- [84] V. RODOPLU et T. MENG, « Minimum energy mobile wireless networks », *IEEE Journal on Selected Areas in Communications*, vol. 17, no. 8, p. 1333–1344, 1999.

- [85] <http://binthatthink.inria.fr>.
- [86] N. MITTON, B. SERICOLA, S. TIXEUIL, E. FLEURY et I. GUÉRIN LASSOUS, « Self-stabilization in Self-organized Wireless Multihop Networks », *Ad Hoc and Sensor Wireless Networks*, vol. 11, no. 1-2, 2011.
- [87] J. RADAK, N. MITTON et D. SIMPLOT-RYL, « Using Battery Level as Metric for Graph Planarization », in *Proceedings of the 10th International Conference on Ad Hoc Networks and Wireless*, ADHOC-NOW, (Paderborn, Germany), 2011.
- [88] G. T. TOUSSAINT, « The Relative Neighbourhood Graph of a Finite Planar Set », *Pattern Recognition*, vol. 12, 1980.
- [89] D. BAKER et A. EPHREMIDES, « The Architectural Organization of a Mobile Radio Network via a Distributed Algorithm », *IEEE Transactions on Communications*, vol. 29, no. 11, 1981.
- [90] M. GERLA et J. TZU-CHIEH TSAI, « Multiclustet, mobile, multimedia radio network », *Wireless Networks*, vol. 1, 1995.
- [91] M. CHATTERJEE, S. K. DAS et D. TURGUT, « WCA : A Weighted Clustering Algorithm for Mobile Ad Hoc Networks », *Cluster Computing*, vol. 5, 2002.
- [92] C. LOW, C. FANG, J. NG et Y. ANG, « Efficient Load-Balanced Clustering Algorithms for wireless sensor networks », *Computer Communication*, vol. 31, 2008.
- [93] W. DALI et H. CHAN, « Clustering Algorithm to Balance and to Reduce Power Consumptions for Homogeneous Sensor Networks », in *Proceedings of Wireless Communications, Networking and Mobile Computing*, WiCom, (Shanghai, China), IEEE, 2007.
- [94] T. SHU, M. KRUNZ et S. VRUDHULA, « Power balanced coverage-time optimization for clustered wireless sensor networks », in *Proceedings of the International symposium on Mobile ad-hoc networking and computing*, MobiHoc, (Urbana-Champaign, IL, USA), p. 111–120, ACM, 2005.
- [95] T. KAUR et J. B., « A strategic deployment and cluster-header selection for wireless sensor networks », *IEEE Transactions on Consumer Electronics*, vol. 55, no. 4, 2009.
- [96] Y. YAO et G. GIANNAKIS, « Energy-Efficient Scheduling for Wireless Sensor Networks », *IEEE Transaction on Communications*, vol. 53, no. 8, 2005.
- [97] O. YOUNIS, M. KRUNZ et S. RAMASUBRAMANIAN, « Node clustering in wireless sensor networks : recent developments and deployment challenges », *IEEE Network*, vol. 20, no. 3, 2006.
- [98] X. MIN, S. WEIREN, J. CHANG-JIANG et Z. YING, « Energy efficient clustering algorithm for maximizing lifetime of wireless sensor networks », *International Journal of Electronics and Communications*, vol. 64, no. 4, 2010.

- [99] O. BUYANJARGAL et Y. KWON, « AEEC-Adaptive and Energy Efficient Clustering Algorithm for Content Based Wireless Sensor Networks », in *Proceedings of the 2nd International Conference on Computer Science and its Applications*, CSA, (Jeju, Korea), p. 1–6, IEEE, 2009.
- [100] N. DIMOKAS, D. KATSAROS et Y. MANOLOPOULOS, « Energy-efficient distributed clustering in wireless sensor networks », *Journal of Parallel and Distributed Computing*, vol. 70, no. 4, 2010.
- [101] K. DONGHYUN, W. YIWEI, L. YINGSHU, Z. FENG et D. DING-ZHU, « Constructing Minimum Connected Dominating Sets with Bounded Diameters in Wireless Networks », *IEEE Transactions on Parallel and Distributed Systems*, vol. 20, no. 2, p. 147–157, 2009.
- [102] T. ANKER, D. BICKSON, D. DOLEV et B. HOD, « Efficient Clustering for Improving Network Performance in Wireless Sensor Networks », in *Proceedings of the 5th European conference on Wireless sensor networks*, EWSN, (Bologna, Italy), p. 221–236, Springer Berlin Heidelberg, 2008.
- [103] A. A. ABBASI et M. YOUNIS, « A survey on clustering algorithms for wireless sensor networks », *Computer Communications*, vol. 30, no. 14–15, p. 2826 – 2841, 2007.
- [104] N. MITTON, *Auto-Organisation des Réseaux sans Fil Multi-Sauts à Grande Echelle*. Thèse de doctorat, INSA Lyon, 3.7.1, 2006.
- [105] <http://www.openstreetmap.org>.
- [106] <http://www.u-blox.com/en/wireless-modules/gsm-gprs-modules/leon-gsm-module-family.html>.
- [107] I. CHATZIGIANNAKIS, S. FISCHER, C. KONINIS, G. MYLONAS et D. PFISTERER, « WISEBED : An Open Large-Scale Wireless Sensor Network Testbed », in *Proceedings of the 1st International Conference On Sensor Applications, Experimentation, and Logistics*, SENSAPPEAL, (Athens, Greece), p. 68–87, Springer Berlin Heidelberg, 2010.
- [108] <http://www.wisebed.eu>.
- [109] S. ČAPKUN, M. HAMDI et J.-P. HUBAUX, « GPS-free Positioning in Mobile Ad Hoc Networks », *Cluster Computing*, vol. 5, no. 2, p. 157–167, 2002.
- [110] E. ERMEL, A. FLADENMULLER, G. PUJOLLE et A. COTTON, « On Selecting Nodes to Improve Estimated Positions », in *Proceedings of the 7th international Mobile and Wireless Communication Networks*, MWCN, (Marrakech, Morocco), p. 449–460, Springer US, 2005.
- [111] K. BENKIC, M. MALAJNER, P. PLANINSIC et Z. CUCEJ, « Using RSSI value for distance estimation in wireless sensor networks based on ZigBee », in *Proceedings of the 15th International Conference on Systems, Signals and Image Processing*, IWSSIP, (Bratislava, Slovakia), p. 303–306, 2008.

- [112] K. SEADA, A. HELMY et R. GOVINDAN, « On the effect of localization errors on geographic face routing in sensor networks », in *Proceedings of the 3rd international symposium on Information processing in sensor networks*, IPSN, (New York, NY, USA), p. 71–80, ACM, 2004.
- [113] A. CARUSO, S. CHESSA, S. DE et A. URPI, « GPS free coordinate assignment and routing in wireless sensor networks », in *Proceedings of the 24th Annual Joint Conference of the IEEE Computer and Communications Societies*, INFOCOM, (Miami, FL, USA), p. 150–160, IEEE, 2005.
- [114] I. STOJMENOVIC, « Localized network layer protocols in wireless sensor networks based on optimizing cost over progress ratio », *IEEE Network*, vol. 20, no. 1, p. 21–27, 2006.
- [115] E. H. ELHAFSI, N. MITTON et D. SIMPLOT-RYL, « Cost over Progress Based Energy Efficient Routing over Virtual Coordinates in Wireless Sensor Networks », in *Proceedings of International Symposium on a World of Wireless, Mobile and Multimedia Networks*, WoWMoM, (Helsinki, Finland), p. 1–6, IEEE, 2007.
- [116] E. CHÁVEZ, N. MITTON et H. TEJEDA, « Routing in Wireless Networks with Position Trees », in *Ad-Hoc, Mobile, and Wireless Networks* (E. KRANAKIS et J. OPATRYNY, éd.), ADHOC-NOW, (Cancun, Mexico), p. 32–45, Springer Berlin Heidelberg, 2007.
- [117] N. MITTON, T. RAZAFINDRALAMBO, D. SIMPLOT-RYL et I. STOJMENOVIC, « Hector is an Energy Efficient Tree-Based Optimized Routing Protocol for Wireless Networks », in *Proceedings of the 4th International Conference on Mobile Ad-hoc and Sensor Networks*, MSN, (Wuhan, China), p. 31–38, IEEE, 2008.
- [118] F. BENBADIS, J.-J. PUIG, M. D. de AMORIM, C. CHAUDET, T. FRIEDMAN et D. SIMPLOT-RYL, « Jumps : Enhancing hop-count positioning in sensor networks using multiple coordinates », *International Journal on Ad Hoc and Sensor Wireless Networks*, vol. abs/cs/0604105, 2006.
- [119] Q. FANG, J. GAO, L. GUIBAS, V. de SILVA et L. ZHANG, « GLIDER : gradient landmark-based distributed routing for sensor networks », in *Proceedings of the 24th Annual Joint Conference of the IEEE Computer and Communications Societies*, INFOCOM, (Miami, FL, USA), p. 339–350, IEEE, 2005.
- [120] D. NICULESCU et B. NATH, « Ad hoc positioning system (APS) », in *Proceedings of the Global Telecommunications Conference*, GLOBECOM, (San Antonio, TX, USA), p. 2926–2931, IEEE, 2001.
- [121] K. LIU et N. ABU-GHAZALEH, « Stateless and guaranteed geometric routing on virtual coordinate systems », in *Proceedings of the 5th International Conference on Mobile Ad Hoc and Sensor Systems*, MASS, (Atlanta, GA, USA), p. 340–346, IEEE, 2008.

- [122] M. CHAWLA, N. GOEL, K. KALAICHELVAN, A. NAYAK et I. STOJMENOVIC, « Beaconless Position Based Routing with Guaranteed Delivery for Wireless Ad-Hoc and Sensor Networks », in *Proceedings of the 19th World Computer Congress, TC-6, IFIP Interactive Conference on Ad-Hoc Networking*, Ad-Hoc Networking, (Santiago, Chile), p. 61–70, Springer US, 2006.