



Optimisation multi-objectif des communications dans les réseaux de capteurs hétérogènes

Karla Breschi

► To cite this version:

Karla Breschi. Optimisation multi-objectif des communications dans les réseaux de capteurs hétérogènes. Réseaux et télécommunications [cs.NI]. Université Bourgogne Franche-Comté, 2018. Français. NNT : 2018UBFCD045 . tel-02144035

HAL Id: tel-02144035

<https://theses.hal.science/tel-02144035>

Submitted on 29 May 2019

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.

THÈSE DE DOCTORAT DE L'ÉTABLISSEMENT UNIVERSITÉ BOURGOGNE FRANCHE-COMTÉ
PRÉPARÉE À L'UNIVERSITÉ DE FRANCHE-COMTÉ

École doctorale n°37
Sciences Pour l'Ingénieur et Microtechniques

Doctorat d'Informatique

par

KARLA BRESCHI

**Optimisation des communications dans les réseaux de capteurs
hétérogènes**

Thèse présentée et soutenue à Besançon, le 6 Décembre 2018

Composition du Jury :

SEVAUX MARC	Professeur à l'Université de Bretagne Sud	Rapporteur
PHAM CONGDUC	Professeur à l'Université de Pau	Rapporteur
HÉAM PIERRE-CYRILLE	Professeur à l'Université de Franche-Comté	Examineur
ADDIS BERNARDETTA	Maître de Conférences à l'Université de Lorraine	Examineur
PHILIPPE LAURENT	Professeur à l'Université de Franche-Comté	Directeur de thèse
BERNARD JULIEN	Maître de Conférences l'Université de Franche-Comté	Co-directeur de thèse

Titre : Optimisation des communications dans les réseaux de capteurs hétérogènes

Mots-clés : Réseaux de capteurs sans fil, Optimisation, Diffusion, Routage, Latence, Énergie, Cross-layer

Résumé :

Cette thèse étudie les problèmes d'optimisation des communications dans les réseaux de capteurs hétérogènes du point de vue de la portée de transmission. Le premier critère considéré est nécessairement l'énergie qu'il s'agit d'économiser pour maximiser la durée de vie du réseau. Le second critère, primordial dans le cas des applications en temps réel, est le délai d'acheminement des données.

Pour résoudre le problème d'optimalité de la diffusion, nous proposons une formulation en

nombre entiers, un algorithme d'approximation, une métaheuristique et une heuristique permettant de trouver ou s'approcher de l'optimal pour les objectifs considérés.

Pour le cas du routage, nous proposons une approche cross-layer via un programme linéaire en nombre entiers entre les couches MAC et Liaison de données permettant de calculer l'optimal et ainsi d'avoir un point de comparaison pour les futures heuristiques ayant les mêmes objectifs.

Title: Optimisation des communications dans les réseaux de capteurs hétérogènes

Keywords: Wireless sensor networks, Optimisation, Broadcast, Routing, Latency, Energy, Cross-layer

Abstract:

This thesis studies the problems of optimization of communications in heterogeneous wireless sensor networks from the point of view of the transmission range. The first criterion considered is necessarily the energy consumption that needs to be reduced in order to maximize the network lifetime. The second criterion is the data transmission delay, which is essential for real-time applications.

To solve the optimality problem of the broadcast, we propose an integer linear program, an approximation

algorithm, a metaheuristic and a heuristic. All these techniques allow us to find or to approximate the optimal solutions for the considered objectives.

For the routing case, we propose a cross-layer approach via an integer linear program between the MAC and Data Link layers. Our method calculates the optimal solution, which represents a point of comparison for future heuristics having the same goals.

REMERCIEMENTS

En premier lieu, je tiens à remercier mon directeur de thèse, Laurent Philippe, pour avoir accepté de diriger ma thèse et pour ses précieuses remarques me permettant de l'améliorer.

Je tiens également à remercier mes rapporteurs, les professeurs Marc Sevaux et Congduc Pham, pour avoir accepté d'évaluer ma thèse. Leurs remarques m'ont permis d'améliorer la qualité de ce manuscrit. Je remercie aussi le professeur Pierre-Cyrille Héam pour son aide au cours de la thèse mais aussi pour avoir accepté de faire partie du jury de cette thèse. Je voudrais également remercier Mme. Addis Bernardetta, Maître de Conférences, pour s'être intéressée à mes travaux et avoir accepté de faire partie du jury.

Je remercie Mme Violeta Felea pour ses encouragements, conseils et discussions au cours de ces trois années ainsi que M. Herve Guyennet pour ses suggestions.

Je tiens à remercier, tout particulièrement, mon encadrant, Julien Bernard, pour ses conseils et son aide tout au long de la thèse. En particulier, je tiens à le remercier pour son soutien, ses encouragements, sa compréhension et sa capacité à connaître les moments où je devais me détacher de la thèse et m'aérer l'esprit. Ses qualités humaines m'ont permis de mener à bien ce projet.

Je voudrais remercier ma famille pour m'avoir encouragée à entreprendre ce nouveau défi. Je tiens à remercier Thomas, mon compagnon, pour sa patience et son soutien dans les moments difficiles que j'ai pu rencontrer au cours de ces trois années.

Je remercie, tout particulièrement, Lemia et Lucas, mes amis et membres de l'équipe réseau de capteurs.

Je remercie, également, mes amis et collègues, doctorants au sein du DISC pour les bons moments passés ensemble et leur présence dans les moments de doute.

Enfin je voudrais remercier tous mes amis, notamment Sophie et Amal, pour leur présence à mes côtés.

SOMMAIRE

Introduction	1
I Les réseaux de capteurs sans fils	3
1 Réseaux de capteurs sans fil hétérogènes	5
1.1 Contexte	5
1.1.1 Nœud capteur	5
1.1.2 Réseau de capteurs sans fil	7
1.1.2.1 Généralités sur les réseaux de capteurs sans fil	7
1.1.2.2 Modèles de connexité	10
1.1.2.3 Modèles d'interférence	12
1.1.2.4 Challenges	14
1.2 Notations	14
1.3 Problèmes	21
1.3.1 Diffusion	21
1.3.2 Routage multi-couche	23
1.3.2.1 Le modèle en couches	23
1.3.2.2 Le cross-layering	24
1.3.2.3 Protocoles de la sous-couche MAC	25
1.4 Méthodes de résolution	26
1.5 Résultats obtenus	28
II Communications dans les réseaux sans fils	29
2 La diffusion dans les réseaux de capteurs sans fils	31
2.1 Ensemble dominant connexe	32
2.1.1 Graphe de disques unitaires	32
2.1.2 Graphe de disques avec liens bidirectionnels	33
2.1.3 Graphe de disques	34

2.1.4	Graphe général	37
2.1.5	Synthèse	38
2.2	Ensemble dominant connexe avec équilibrage de la charge	38
2.2.1	Graphe de disques unitaires	38
2.2.2	Synthèse	40
2.3	Ensemble dominant connexe et énergie	41
2.3.1	Graphe de disques unitaires	41
2.3.2	Graphe de disques	42
2.3.3	Synthèse	43
2.4	Ensemble dominant connexe et latence	44
2.4.1	Graphe de disques unitaires	44
2.4.2	Graphe Cordal	45
2.4.3	Graphe de disques	45
2.4.4	Graphe général	46
2.4.5	Synthèse	47
2.5	Ensemble dominant connexe tolérant aux pannes	47
2.5.1	Graphe de disques unitaires	48
2.5.2	Graphe de disques avec liens bidirectionnels	49
2.5.3	Graphe de disques	49
2.5.4	Graphe non orienté	51
2.5.5	Graphe général	52
2.5.6	Synthèse	52
2.6	Synthèse	53

III Contributions 55

3 Algorithme d'approximation pour la diffusion dans les réseaux de capteurs 57

3.1	Caractérisation du problème	57
3.1.1	Complexité	58
3.1.2	Programme linéaire en nombres entiers	59
3.2	Algorithme d'approximation BB	62
3.3	Étude préliminaire	63
3.3.1	Structure du chemin P_1 par rapport à P_2	64
3.3.2	Preuve du théorème 1	65
3.4	Algorithme Bounded Backbone (BB)	65

3.5	Preuve	67
3.6	Synthèse	70
4	Heuristiques pour la diffusion dans les réseaux de capteurs	71
4.1	Heuristique naïve	71
4.2	Heuristique Highly Bounded Backbone (HBB)	72
4.3	Recuit simulé	74
4.4	Expérimentations	76
4.4.1	Description	77
4.4.2	Résultats et analyses	78
4.4.3	Synthèse des expérimentations	82
4.5	Synthèse	82
5	Routage multi-couche	85
5.1	Protocoles cross-layer dans les WSN	86
5.1.1	Approches cross-layer inter-couche	86
5.1.2	Approches cross-layer par fusion de couches	88
5.2	Formalisation du problème	89
5.2.1	Définition du problème	89
5.2.2	Modélisation de la solution	90
5.3	Programme linéaire en nombres entiers	90
5.4	Algorithme IDeg-Routing&MAC	93
5.5	Exemple explicatif	93
5.6	Simulations	96
5.6.1	Description	96
5.6.2	Résultats	97
5.6.3	Discussion	98
5.7	Synthèse	99
	Conclusion	101
	Bibliographie	105
	Table des figures	112
	Liste des tables	114
	Liste des définitions	115

Listes des théorèmes

118

INTRODUCTION

CONTEXTE

Nous vivons dans une période où tout se mesure et s'analyse : le comportement des écosystèmes [52], les performances énergétiques de nos bâtiments [83], les déplacements dans nos villes [59], jusqu'aux constantes de nos corps [49].

Auparavant cela demandait l'intervention humaine afin d'effectuer le relevé des informations monitorées. Grâce au développement d'une technologie, nommée capteur, capable de détecter et de communiquer des informations, nous sommes à présent en mesure d'automatiser cette tâche. Les capteurs étant limités en portée de communication et la zone à superviser étant souvent vaste, un seul capteur n'est pas capable de la couvrir entièrement. Par conséquent, cela demande l'utilisation de plusieurs capteurs mis en réseau afin d'être capable d'acquérir les informations environnementales mais aussi de les acheminer vers l'utilisateur final.

Ces dernières années, les réseaux de capteurs ont été utilisés pour développer des villes intelligentes. Une ville intelligente est une ville munie de différents capteurs permettant la collecte de données constituant les informations essentielles à la gestion efficace des ressources. Il s'agit d'une ville intégrant toutes ses infrastructures et ses services dans un ensemble cohérent et utilisant des dispositifs intelligents de surveillance et de contrôle, pour assurer la durabilité et l'efficacité [59]. L'objectif est de répondre aux défis de la surpopulation dans les zones urbaines des années à venir. L'utilisation de ce type de réseau a été proposée, par exemple, pour l'éclairage urbain intelligent [87], la surveillance des réseaux de distribution de l'eau [84] ou de l'électricité [59].

De nombreux projets ont été financés dans ce but [59]. Par exemple, en Autriche, un projet sur les villes intelligentes a été lancé dans le but d'améliorer l'efficacité énergétique et la protection du climat en réduisant l'empreinte carbone. Un autre exemple est un projet réalisé en Espagne qui vise la mise en place d'une collecte intelligente des déchets et d'un système de stationnement et d'éclairage public intelligent.

PROBLÉMATIQUE

Les services fournis par les réseaux de capteurs reposent sur deux opérations fondamentales : la diffusion des demandes sur les informations à capter, et l'acheminement des informations recueillies depuis les nœuds capteurs jusqu'à un nœud de collecte. L'acheminement correspond à l'intérêt principal des réseaux de capteurs, celui de fournir les données extraites de l'environnement à l'utilisateur. Quant à l'opération de diffusion, elle est essentielle pour l'exploitation dynamique d'un réseau de capteurs, où la caractéristique de la donnée extraite par le réseau est variable dans le temps et dépend de la dynamique de l'environnement et des besoins des utilisateurs. Nous qualifions ce type de réseau d'interrogeable. Ces deux opérations sont coûteuses énergétiquement

et contraintes par l'aspect matériel du capteur (énergie, bande passante, mémoire, puissance cpu, défaillance) et du réseau (fiabilité de la connexion, portées de communication variables, liens unidirectionnels). L'objectif principal d'un tel système est la récupération de l'information, mais souvent le volet quantitatif n'est pas suffisant, c'est à dire qu'on ne s'intéresse pas uniquement à la quantité d'informations reçues.

Le volet qualitatif de l'information extraite, décliné le plus souvent sous la forme de la fiabilité et du délai, devient primordial. Assurer un niveau qualitatif important de la donnée engendre souvent une surconsommation énergétique.

Cette thèse se concentre sur l'optimisation de ces deux objectifs contradictoires à savoir le délai d'acheminement des données (latence) et le coût énergétique dans la diffusion et le routage. Nous nous positionnons dans le cas où les capteurs peuvent avoir des portées hétérogènes.

CONTRIBUTIONS ET ORGANISATION

Le but de notre travail est traiter l'optimisation dans les communications des réseaux de capteurs sans fils hétérogènes. En particulier, nous avons traité, dans un premier temps, l'optimisation de l'énergie et de la latence dans la diffusion. Pour cela, nous avons proposé différentes manières de résoudre ce problème d'optimisation. Certaines méthodes ne sont pas adaptées en raison du temps de calcul grandissant de manière exponentielle avec la taille du réseau. Pour des réseaux de petite taille, nous proposons une solution optimale. Pour des réseaux de taille moyenne à grande, des méthodes approchées.

Dans un deuxième temps, nous nous sommes concentrés sur l'optimisation de la latence et de l'énergie dans le cas du routage. Pour cela, nous avons proposé une solution optimale.

La suite de ce manuscrit s'organise de la manière suivante.

Dans le chapitre 1, nous présentons le contexte de la thèse, à savoir les réseaux de capteurs sans fils ainsi que leurs caractéristiques. De plus, nous donnons les différentes définitions utiles pour la suite du manuscrit.

Le chapitre 2 introduit les résultats trouvés dans la littérature pour la diffusion dans les réseaux de capteurs.

Le chapitre 3 présente l'optimisation de la diffusion par le biais d'un programme linéaire en nombres entiers ainsi que par un algorithme d'approximation.

Dans le chapitre 4, nous traitons l'optimisation de la diffusion en construisant une dorsale via une heuristique et le recuit simulé. En outre, ces méthodes sont comparées avec celles proposées dans le chapitre précédent à travers des simulations.

Pour finir, le chapitre 5 commence par un état de l'art du routage multi-couche dans les réseaux de capteurs. Puis il propose la résolution de ce problème via un programme linéaire en nombres entiers.



LES RÉSEAUX DE CAPTEURS SANS FILS

RÉSEAUX DE CAPTEURS SANS FIL HÉTÉROGÈNES

Sommaire

1.1 Contexte	5
1.1.1 Nœud capteur	5
1.1.2 Réseau de capteurs sans fil	7
1.2 Notations	14
1.3 Problèmes	21
1.3.1 Diffusion	21
1.3.2 Routage multi-couche	23
1.4 Méthodes de résolution	26
1.5 Résultats obtenus	28

Dans ce chapitre, nous introduisons le contexte dans lequel cette thèse s'inscrit. En particulier, nous définissons ce que représente un capteur et un réseau de capteurs sans fil. Nous présentons, ensuite, les notations utilisées tout au long du manuscrit. Nous continuons par la présentation des problèmes traités ainsi que les méthodes de résolution qui peuvent être utilisées pour ce type de problème. Pour finir, nous présentons les résultats obtenus au cours de la thèse.

1.1/ CONTEXTE

Dans cette section, nous définissons les concepts de base, à savoir ce que constitue un capteur et un réseau de capteurs sans fil. Nous définissons, également, les modes de collecte et de déploiement. Pour finir, nous présentons les différentes topologies, les modèles de connexité et les divers challenges auxquels nous devons faire face lorsque nous travaillons avec ce type de réseau.

1.1.1/ NŒUD CAPTEUR

Un nœud capteur est un dispositif électronique capable d'acquérir des données environnementales, de les traiter et de les communiquer à un utilisateur final.

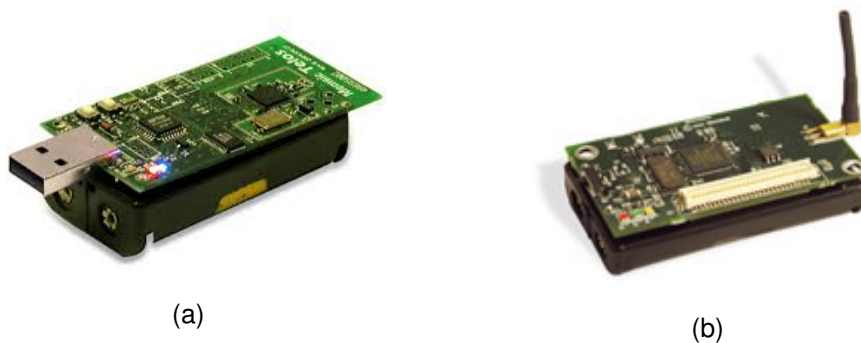


FIGURE 1.1 – Capteur TelosB (à gauche) et capteur MicaZ (à droite)

Dans le reste du manuscrit, nous utilisons les termes capteur ou nœud pour désigner un nœud capteur de manière abrégée.

Un nœud capteur est constitué des composants suivants :

- Un **processeur** dont la puissance de calcul est limitée. Il est utilisé pour le traitement des données captées et des données transmises par les autres capteurs.
- Une **mémoire** exploitée à la fois pour stocker des données recueillies ainsi que pour l'exécution d'un programme.
- Un **émetteur-récepteur radio** permettant les communications entre capteurs via un medium sans fil. La puissance de signal utilisée détermine la portée de communication. Toutefois, plus la puissance augmente, plus la consommation énergétique est importante.
- Une **unité de captage** permettant de mesurer des grandeurs physiques telles que la température, la luminosité ou encore l'humidité. Les données captées sont, ensuite, transformées en données numériques pouvant être exploitées par le processeur.
- Une **source d'énergie** fournissant l'énergie nécessaire au fonctionnement des autres composants. La ressource énergétique, la plus souvent utilisée, est la batterie. Toutefois, suivant les applications, les capteurs peuvent être branchés à une source énergétique continue.
- Un **GPS** (*Global Positioning System*) permettant de géolocaliser le capteur. Il s'agit d'un dispositif optionnel car les contraintes économiques et énergétiques limitent son utilisation.

Les différents composants que nous venons de citer sont représentés dans l'image 1.2.

Le tableau 1.1 considère deux types de capteurs qui peuvent être trouvés dans le commerce et illustre leurs principales caractéristiques. Nous pouvons constater que les principales contraintes des capteurs sont ces ressources limitées.

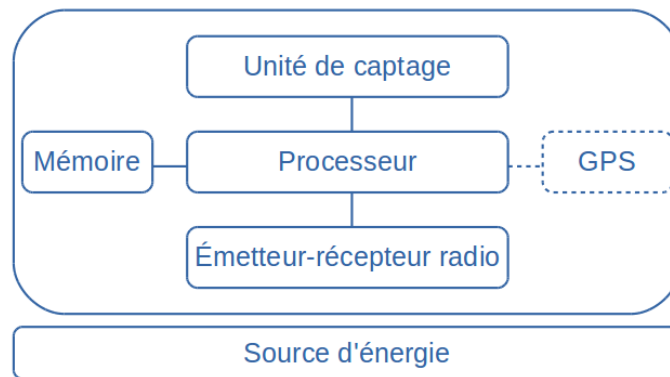


FIGURE 1.2 – Composants d'un capteur

<i>Caractéristique</i>	<i>MicaZ</i>	<i>TelosB</i>
Puissance de calcul (MHz)	7.37	8
Mémoire (kB)	512	1024
RAM (kB)	4	10
Énergie	2 Batteries AA	2 Batteries AA

TABLE 1.1 – Comparaison des caractéristiques des capteurs

Les opérations effectuées par un capteur utilisant de l'énergie sont l'acquisition des données environnementales, le traitement des données et la communication. Les opérations les plus coûteuses en termes énergétiques sont la communication et l'acquisition des données environnantes [43]. Dans le cadre de cette thèse, nous allons nous focaliser sur l'économie de l'énergie et, plus particulièrement, l'économie de l'énergie lors de la communication.

L'utilisation de capteurs en groupe forme un réseau de capteurs comme nous le verrons dans la prochaine section.

1.1.2/ RÉSEAU DE CAPTEURS SANS FIL

Dans cette section, nous présentons les réseaux de capteurs de manière générale. Puis, nous détaillons les divers modèles permettant de les modéliser suivant les caractéristiques à prendre en compte. Pour finir, nous expliquerons les différents challenges propres à ce type de réseau.

1.1.2.1/ GÉNÉRALITÉS SUR LES RÉSEAUX DE CAPTEURS SANS FIL

Nous définissons, dans cette section, ce que constitue un réseau de capteurs sans fil, leurs domaines d'utilisation, les différents modes de collecte et de déploiement et, pour finir les topologies.

Définition 1 : Réseau de capteurs sans fil

Un réseau de capteurs sans fil est un réseau ad hoc constitué d'un ensemble de nœuds capteurs mesurant et communiquant des données environnementales pour les transmettre à un ou plusieurs nœuds particuliers nommé nœud puits ou station de base. Les communications au sein de ce type de réseau sont effectuées via des ondes radio.

D'après la définition, un réseau de capteur se compose de deux types de nœuds : les nœuds capteurs et les nœuds puits. Les premiers ont pour objectif de relever des informations sur la zone de captage et, ensuite, de les transférer vers le point de collecte représenté par la station de base ou le puits. La station de base, quant à elle, récupère les informations qui lui sont transmises et les envoie au centre de traitement. Elle peut également envoyer un message à tous les nœuds capteurs afin de diffuser une information comme par exemple, la périodicité de l'acquisition des données environnementales.

Un réseau de capteurs peut se composer d'une ou plusieurs stations de base : nous parlons de réseau mono-station de base (*mono-sink*) ou de réseau multi-station de base (*multi-sink*).

Les réseaux de capteurs appartiennent à la famille des réseaux ad hoc. Au sein de ce type de réseau, la communication s'effectue par le biais d'ondes radio. Ils se caractérisent par leur capacité à s'organiser sans infrastructure définie au préalable.

Domaines d'utilisation :

Les domaines d'utilisation des réseaux de capteurs sont divers et variés [21, 42]. Dans la suite, nous donnons quelques exemples :

- La santé : un exemple est le réseau corporel (*Body Area Network*). Il s'agit d'un réseau sans fil permettant d'interconnecter sur, autour ou dans le corps humain, des minuscules dispositifs pouvant effectuer des mesures (capteurs) ou agir de façon active (actionneurs). Ces capteurs sont en mesure de communiquer avec un centre de service distant afin, par exemple, de communiquer l'état du patient sous observation. Des exemples de capteurs corporels sont les capteurs de glycémie ou encore de pression artérielle [49]. Notamment, l'implantation d'un biocapteur chez un patient fournit une méthode plus précise et moins invasive pour mesurer le taux de glucose et, en fonction du résultat, un dispositif complémentaire peut effectuer une injection automatique d'insuline.
- L'urbanisation et l'infrastructure : les capteurs sont utilisés, par exemple, dans le cadre des villes intelligentes. Federspiel et al. [9] étudient l'amélioration de la production, de la distribution et de la consommation d'énergie via l'utilisation de capteurs. Idwan et al. [80] proposent l'utilisation des réseaux de capteurs pour optimiser la gestion des déchets grâce à l'utilisation de bennes à ordures intelligentes équipées de capteurs de niveau de déchets. La connaissance du niveau pouvant être utilisée par les municipalités pour déployer le nombre suffisant de camions et de personnes pour la collecte des déchets. De plus, la planification du parcours et les routes à suivre peuvent être effectués pour minimiser le temps et le carburant nécessaires.

- L'environnement : les capteurs sont utilisés pour surveiller la nature. Par exemple, ils permettent la détection des inondations ou la présence de feu dans les forêts. Un autre exemple d'application peut être la prédiction du temps météorologique. L'un des exemples les plus représentatifs est le déploiement d'un réseau de capteurs composé de 32 nœuds sur l'île Great Duck [11]. Les capteurs utilisés détectaient la température, la pression barométrique et l'humidité. Le but était de surveiller l'environnement naturel d'un oiseau (pétrel de mer) et son comportement en fonction des changements climatiques.
- Le domaine militaire : les capteurs sont utilisés afin de détecter des explosifs ou encore dans un champs de bataille pour détecter les mouvements des troupes ennemies. Par exemple, l'université de Virginie propose de remplacer les mines par des milliers de capteurs dispersés qui détecteront une intrusion d'unités hostiles [17].

Les informations mesurées par les capteurs peuvent être recueillies de différentes manières. Ce point sera traité dans la sous-section suivante.

Modes de collecte :

La collecte des informations environnementales peut se faire suivant trois modes différents :

- Supervision (*Monitoring*) : Ce mode permet l'acquisition en continu de la grandeur à mesurer. Les capteurs envoient de manière périodique les données acquises.
- Collecte à la demande : Lorsque nous souhaitons avoir les informations, la station de base effectue une diffusion pour propager la demande. Les capteurs effectuent la mesure et remontent l'information mesurée à la station de base.
- Collecte suite à un événement : Lorsqu'un événement se produit (changement brusque de température, détection d'un mouvement, minuteur expiré, etc), les capteurs ayant détecté l'information la remontent jusqu'à la station de base.

Modes de déploiement :

Les réseaux de capteurs peuvent être déployés suivant une topologie pré-configurée ou dispersés de manière aléatoire [23].

Il existe des réseaux de capteurs sans fils mobiles dont le déploiement n'est pas fixe. Nous ne nous intéressons pas à l'aspect mobile dans nos travaux.

Une fois les capteurs déployés, une topologie logique peut être mise en place afin de palier au manque d'infrastructure physique des réseaux de capteurs sans fils.

Topologies :

Un réseau de capteur peut être organisé suivant différentes topologies : en étoile, en maille, en grille ou en arbre.

Dans la topologie en étoile dessinée dans la figure 1.3a, les nœuds sont tous reliés à un nœud central. Tous les nœuds sont donc à une distance d'un saut du nœud central.

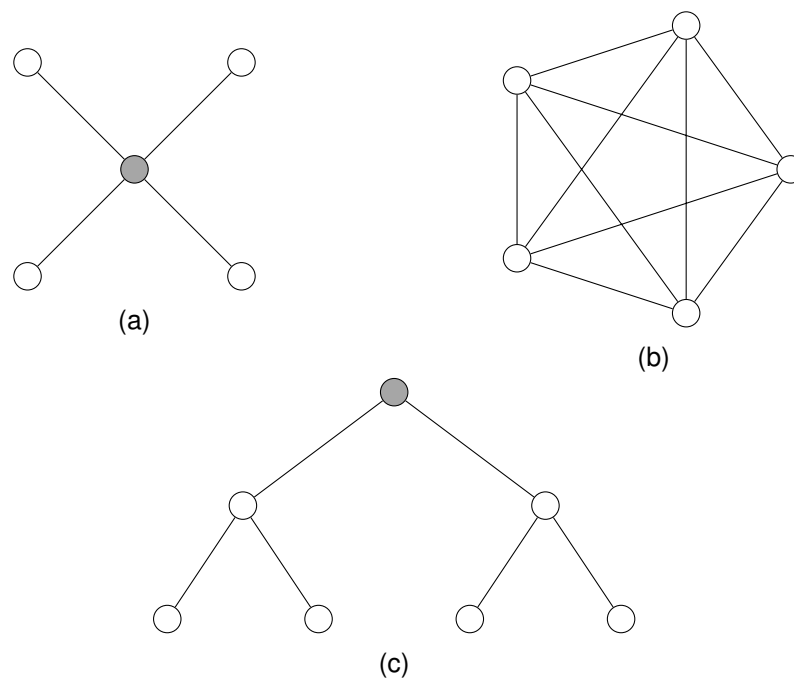


FIGURE 1.3 – Topologies d'un réseau de capteurs (a) en étoile (b) en maille (c) en arbre

Dans une topologie en maille, tous les nœuds sont connexes pair à pair sans la présence d'une hiérarchie centrale. Cette topologie est représentée dans l'image 1.3b.

Dans la topologie en arbre, une hiérarchie s'installe entre les différents nœuds. Le réseau se divise en niveaux. Le niveau le plus haut contient la station de base qui est connectée à plusieurs nœuds se trouvant à un niveau inférieur. Ces nœuds pouvant eux-mêmes être connectés à d'autres nœuds de niveau inférieur. Cela constitue un arbre comme le montre la figure 1.3c.

Pour abstraire les réseaux de capteurs, les chercheurs ont proposé différents modèles, comme nous le verrons dans les sections suivantes.

1.1.2.2/ MODÈLES DE CONNEXITÉ

Un réseau de capteurs peut être modélisé via un graphe. Dans ce cas, les capteurs sont les sommets et les connexions entre les capteurs sont représentées par des arêtes. Cette représentation est détaillée dans la section 1.2. L'avantage de l'utilisation de ce type de modélisation est de faciliter l'étude mathématique des algorithmes proposés, leur correction ainsi que leur performance. Le modèle de connexité le plus simple se nomme graphe général (*General Graph*). Il se représente par un graphe non orienté dépourvu de propriétés particulières. Ce type de modèle est considéré comme trop pessimiste car la connexité dans un réseau de capteurs est soumise à des règles géométriques que nous ne retrouvons pas dans les graphes généraux. C'est pour cela que dans la littérature, différents modèles ont été proposés suivant le niveau d'abstraction auquel nous voulons nous positionner ou suivant la propriété/contrainte étudiée. Schmid et Wattenhofer [32] et Yu et al. [70] présentent les modèles des connectivités les plus communément utilisés. Nous verrons dans la suite la description de certains d'entre eux.

Graphe de Disques Unitaires (UDG) :

Un graphe de disques unitaires (*Unit Disk Graph*) est un graphe non orienté dans lequel les nœuds sont déployés dans un plan bidimensionnel. Il représente une collection de disques ayant tous le même diamètre unitaire. En d'autres termes, deux nœuds sont voisins si, et seulement si, leur distance euclidienne est inférieure ou égale à 1. Dans ce cas, tous les nœuds ont la même portée de transmission. Dans ce modèle les nœuds sont supposés avoir une antenne radio omnidirectionnelle et être dispersés dans un environnement planaire dépourvu d'obstacles.

Quasi Graphe de Disques Unitaires (QUDG) :

Le quasi graphe de disques unitaires (*Quasi Unit Disk Graph*) est un modèle généralisant le graphe de disques unitaires. En effet, celui-ci permet de prendre en compte les antennes directionnelles ou la présence de petits obstacles. Comme dans les modèles précédents, les nœuds sont dispersés dans un plan à deux dimensions. Deux nœuds peuvent être voisins si leur distance euclidienne est d'au plus ρ où ρ appartient à l'intervalle $(0, 1]$. Deux cas peuvent se présenter. La distance euclidienne est comprise entre 0 et ρ ou elle appartient à l'intervalle $(\rho, 1]$. Dans le premier cas, les nœuds sont voisins. Dans le deuxième cas, les nœuds peuvent ou pas être voisins.

Graphe de Disques (DG) :

Dans le cas des graphes de disques (*Disk Graph*) [13], le réseau est modélisé par un graphe orienté. Les nœuds du graphe sont situés dans un plan euclidien et chaque nœud possède sa propre portée de transmission. Un nœud i est voisin d'un nœud j si, et seulement si, la distance euclidienne entre eux est inférieure ou égale à la portée de transmission du nœud i . Nous remarquons que dans ce cas de figure, il est possible que le nœud i ait le nœud j comme voisin mais que le nœud j n'ait pas le nœud i comme voisin. En effet, cela dépend de leur portée de transmission.

Graphe de Disques avec liens bidirectionnels (BDG) :

Ce modèle est un cas particulier du modèle de graphe de disques dans lequel seul un sous-ensemble des arêtes est considéré : les arêtes bidirectionnelles. Autrement dit, une arête existe entre deux nœuds u et v si la distance euclidienne entre eux est inférieure ou égale à la plus petite portée parmi r_u et r_v . Dans ce cas, le graphe de disques avec liens bidirectionnels est modélisé par un graphe non orienté.

Le modèle de graphe de disques unitaires permet de modéliser un réseau de capteurs homogène : tous les capteurs ont la même portée de communication. En revanche, les modèles de graphe de disques, de graphe de disques avec liens bidirectionnels et de quasi graphe de disques unitaires rendent possible la représentation d'un réseau de capteurs hétérogène. L'hétérogénéité est ici mise en œuvre au niveau de la portée de communication affectée aux différents capteurs du réseau. Toutefois, les graphes de disques avec liens bidirectionnels ne permettent pas de prendre en compte le cas où la relation de voisinage est unidirectionnelle, ce qui représente le cas le plus générique.

Les modèles que nous venons de voir permettent l'abstraction du réseau de capteur en mettant en avant la connectivité de celui-ci. En complément, il existe une autre catégorie de modèles permettant de représenter le réseau du point de vue de l'interférence présente entre les différents nœuds le constituant.

1.1.2.3/ MODÈLES D'INTERFÉRENCE

D'une manière générale, dans les réseaux sans fil et, en particulier, dans les réseaux de capteurs sans fil, le support de communication est partagé et, par conséquent, les transmissions sont exposées aux interférences. Concrètement, un nœud peut ne pas être en mesure de recevoir correctement un message d'un nœud adjacent si une transmission simultanée est en train d'être effectuée à proximité. Le rôle d'un modèle d'interférence est de mettre en évidence les transmissions qui, effectuées de manière simultanée, créent une perturbation de l'onde entraînant ainsi une mauvaise voire une non réception du message envoyé. Lorsque les transmissions ne sont pas effectuées durant le même créneau de temps, aucune interférence ne se produit. Un certain nombre de modèles permettant de représenter ces interférences ont été proposés. Schmid et al. [32] en énumèrent un certain nombre. Dans la suite, nous présentons le plus communément utilisé ainsi que le modèle simplifié que nous avons utilisé dans le cadre de la thèse.

Signal-to-Interference Plus Noise (SINR) :

Le modèle le plus connu est le modèle physique nommé SINR. Il est communément utilisé pour définir les interférences au niveau des couches basses. Ce modèle considère que la réception est réussie en fonction de la puissance du signal reçu, du niveau de bruit ambiant ainsi que des interférences provoquées par la transmission simultanée des nœuds.

De manière plus formelle, ce modèle considère qu'un message de source s est reçu par le nœud d si :

$$\frac{P_d}{N + I} \geq \beta$$

Où :

- P_d représente la puissance du signal reçu par le nœud d ,
- N est le niveau du bruit ambiant,
- I indique le niveau des interférences générées par les autres nœuds,
- β est une constante indiquant le rapport minimal entre le signal et l'interférence requis afin de recevoir un message avec succès.

Dans le cadre de la thèse, nous avons considéré un modèle plus simplifié que nous présentons dans la suite.

Modèle d'Interférence simplifié :

Dans notre modèle simplifié d'interférence, nous considérons qu'un nœud u peut créer une interférence pour un nœud v si v est dans la portée de transmission de u . Nous considérons deux types de collisions : la collision primaire ou explicite et la collision secondaire ou implicite. La collision primaire se produit lorsqu'un nœud émet et reçoit pendant le même créneau de temps. Ceci est représenté dans les figures 1.4 et 1.5. La figure 1.4 représente le cas où le nœud 2 reçoit un message venant du nœud 1 et envoie un message au nœud 3 pendant le même créneau. La figure 1.5 illustre le cas où le nœud 2 reçoit un message du nœud 1 et 3 pendant le même créneau. La collision secondaire se produit lorsqu'un nœud, qui reçoit une transmission qui lui est destinée, est perturbé par une autre transmission qui ne lui est pas destinée. Ceci est montré dans la figure 1.6, dans laquelle le nœud 3 envoie un message au nœud 4 en même temps que le nœud 1 transmet un message au nœud 2. Or, les nœuds 1 et 3 sont dans la portée de transmission l'un de l'autre, donc cela provoque une collision. Par conséquent, pour éviter les collisions, les créneaux de transmission des deux nœuds pouvant entraîner une collision doivent être ordonnancés à des intervalles de temps différents.

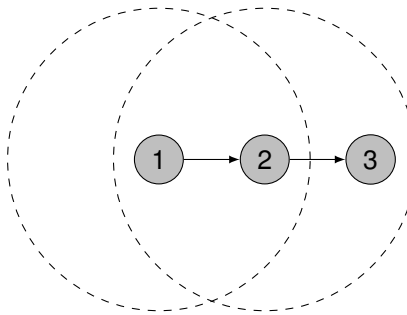


FIGURE 1.4 – Exemple 1 de conflit primaire

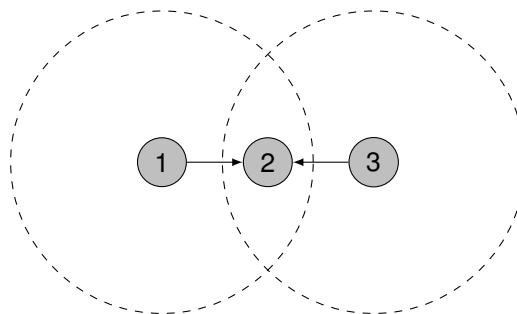


FIGURE 1.5 – Exemple 2 de conflit primaire (nœud caché)

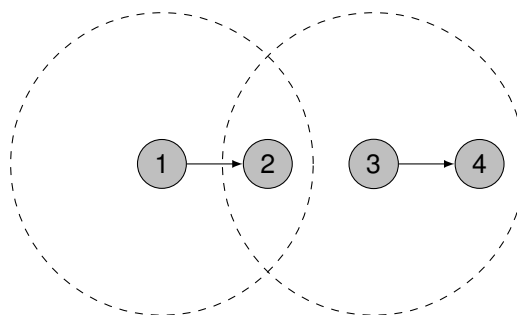


FIGURE 1.6 – Exemple de conflit secondaire (nœud exposé)

Pour comptabiliser les collisions ou interférences auxquelles un nœud peut être soumis, la notion de degré d'interférence est utilisée.

Définition 2 : Degré d'interférence

Le degré d'interférence (*interference degree*) d'un nœud se définit comme la somme du nombre de ses voisins à un saut et à deux sauts.

Nous avons présenté les réseaux de capteurs sans fil ainsi que les modèles permettant de le représenter de manière abstraite. Nous allons, à présent, voir les différents challenges qui les caractérisent.

1.1.2.4/ CHALLENGES

Bien que les réseaux de capteurs partagent de nombreuses similitudes avec d'autres systèmes distribués, ils sont soumis à une variété de défis et de contraintes qui leurs sont propres. Cette section décrit les contraintes les plus importantes dans un réseau de capteurs.

Ressources limitées :

La contrainte la plus souvent associée au réseau de capteurs est l'énergie. En effet, les nœuds sont la plupart du temps alimentés par des batteries, qui doivent être remplacées ou rechargées (par exemple, en utilisant l'énergie solaire) lorsqu'elles sont épuisées. Dans certains cas de figure, les capteurs sont déployés dans un environnement hostile ou bien difficile d'accès, ce qui implique que lorsque leur source d'énergie est épuisée ils seront inutilisables. Un nœud inutilisable peut entraîner différents problèmes allant d'un manque de mesure dans la zone de captage à un partitionnement du réseau.

Auto-organisation :

La nature de nombreuses applications de réseau de capteurs exige que les capteurs soient dispersés dans des zones éloignées ou dans des environnements hostiles. Par conséquent, les capteurs doivent être capables de s'organiser de manière autonome et de s'adapter aux défaillances sans intervention humaine. Un autre aspect qui peut engendrer le besoin d'une ré-organisation du réseau peut être la présence de capteurs mobiles.

1.2/ NOTATIONS

Dans cette section, nous donnerons une série de définitions inhérentes à la théorie des graphes qui nous permettront de modéliser un réseau de capteurs ainsi que ses caractéristiques intrinsèques.

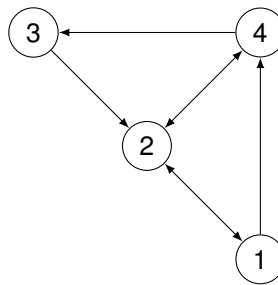


FIGURE 1.7 – Exemple de graphe orienté

Définition 3 : Graphe orienté [15]

Un graphe orienté G est représenté par un couple (V, E) , où V est un ensemble fini et E une relation binaire sur V . L'ensemble V est appelé ensemble des sommets de G et l'ensemble E constitue l'ensemble des arcs de G .

La figure 1.7 représente un graphe orienté G dont l'ensemble des sommets V est constitué des nœuds 1, 2, 3, et 4. L'ensemble E contient les arcs suivants : $(1, 2)$, $(1, 4)$, $(2, 1)$, $(2, 4)$, $(3, 2)$, $(4, 2)$, $(4, 3)$.

Un réseau de capteurs peut être modélisé comme un graphe orienté $G = (V, E)$. Dans ce cas, l'ensemble des sommets V représente les capteurs. Un arc entre les sommets i et j existe si, et seulement si, j se situe dans la portée de i (c'est-à-dire que i peut envoyer un message à j). L'ensemble des arcs ainsi définis constitue l'ensemble E du graphe G . L'ensemble V^* se compose de tous les nœuds du réseau sauf la station de base.

Définition 4 : Sommet adjacent ou voisin

Dans un graphe $G = (V, E)$ non orienté (respectivement orienté), deux sommets u et v appartenant à V sont dits adjacents ou voisins si l'arête (respectivement arc) (u, v) appartient à E .

Dans l'exemple de la figure 1.7, nous pouvons voir que le nœud 4 est un voisin du nœud 1 alors que le nœud 1 n'est pas un voisin du nœud 4.

Nous remarquons que dans un graphe orienté la relation d'adjacence peut être asymétrique. En effet, il est possible qu'un nœud u ait comme voisin un nœud v mais que le nœud v n'ait pas u comme voisin.

Appliquée au réseau de capteurs, la relation d'adjacence traduit l'aptitude d'envoyer un message.

Définition 5 : Voisinage ouvert

Dans un graphe non orienté $G = (V, E)$, le voisinage ouvert $N(v)$ d'un sommet v appartenant à V est l'ensemble des sommets adjacents à v . Formellement, la définition est :

$$N(v) = \{w | (v, w) \in E\}$$

Définition 6 : Voisinage fermé

Dans un graphe non orienté G , le voisinage fermé $N[v]$ d'un sommet v est constitué du voisinage ouvert plus le sommet lui-même :

$$N[v] = N(v) \cup \{v\}$$

En revanche, si le graphe G est orienté, une distinction s'effectue entre voisin entrant et voisin sortant. Par conséquent, il existe un voisinage entrant et un voisinage sortant. Le **voisinage sortant ouvert** $N^+(v)$ et le **voisinage entrant ouvert** $N^-(v)$ se définissent comme suit : $N^+(v) = \{w | (v, w) \in E\}$ et $N^-(v) = \{w | (w, v) \in E\}$.

De manière analogue, nous pouvons définir le voisinage entrant $N^-[v]$ et sortant $N^+[v]$ fermé dans le cas d'un graphe orienté.

Lorsque le réseau de capteurs est modélisé par un graphe orienté, le voisinage sortant d'un nœud v représente les nœuds auxquels v peut envoyer un message. Tandis que le voisinage entrant symbolise les nœuds qui peuvent envoyer un message au nœud v .

Dans l'exemple de la figure 1.7, le voisinage ouvert sortant du nœud 2 se compose des nœuds 4 et 1. En revanche, son voisinage ouvert entrant est composé des nœuds 1, 3 et 4.

Définition 7 : Degré

Dans un graphe G , le degré d'un sommet v représente le nombre d'arêtes qui ont pour extrémité v , autrement dit le nombre de voisins de v .

Dans le cas des graphes orientés, nous distinguons deux types de degrés : le degré entrant et le degré sortant. Le **degré entrant** constitue le nombre de voisins entrants de v . Le **degré sortant** représente le nombre de voisins sortants de v .

Dans les cas des réseaux de capteurs modélisés par un graphe non orienté, le degré représente le nombre de nœuds auxquels un nœud peut envoyer ou desquels il peut recevoir un message. Pour un graphe orienté, le degré entrant et le degré sortant d'un nœud u représentent, respectivement, le nombre de voisins pouvant envoyer un message à u et le nombre de voisins auxquels le nœud u peut envoyer un message.

Considérons à nouveau, le graphe de la figure 1.7. Le degré entrant du nœud 2 est 3 alors que son degré sortant est de 2.

Il existe deux autres définitions inhérentes au degré d'un graphe : le degré maximum et minimum.

Définition 8 : Degré maximum

Le degré maximum d'un graphe G , noté Δ est le maximum des degrés des sommets du graphe.

Définition 9 : Degré minimum

Le degré minimum d'un graphe G , noté δ est le minimum des degrés des sommets du graphes.

Définition 10 : Chemin

Dans un graphe $G = (V, E)$, un chemin allant du nœud u_0 vers le nœud u_l , noté P , est une suite $(u_0, u_1, \dots, u_l)$ tels que pour tout k appartenant à $[0, l]$, (u_k, u_{k+1}) est une arête de G .

Dans un réseau de capteur, un chemin $P(u, v)$ représente les nœuds devant retransmettre l'information afin que celle-ci soit envoyée depuis le nœud u vers le nœud v .

Dans l'exemple de la figure 1.7, un chemin pour aller du nœud 2 au nœud 3 est : $P(2, 3) = \{2, 4, 3\}$.

Définition 11 : Longueur d'un chemin

Étant donné un graphe $G = (V, E)$. La longueur $L(P(u, v))$ d'un chemin $P(u, v)$, où u et v sont des nœuds du graphe G , constitue le nombre d'arêtes appartenant au chemin.

Définition 12 : Plus court chemin

Étant donné un graphe $G = (V, E)$. Le plus court chemin entre deux nœuds u et v est le chemin de longueur minimale parmi tous les chemins reliant u à v .

Définition 13 : Excentricité

L'excentricité $e(s)$ d'un sommet s est la distance maximale parmi les plus courts chemins entre le sommet s et tout autre sommet du graphe. L'excentricité correspond au nombre $e(s)$ défini de la manière suivante :

$$e(s) = \max\{L(P(s, s')) | s' \in V\}$$

En considérant le graphe de l'image 1.7, nous constatons que :

- $L(3, 1) = 2$
- $L(3, 2) = 1$
- $L(3, 4) = 2$

L'excentricité du nœud 3 est donc de 2.

Définition 14 : Diamètre

Le diamètre d'un graphe est l'excentricité maximale de ses sommets :

$$\text{diam}(G) = \max\{e(s) | s \in V\}$$

Dans le graphe de la figure 1.7, le diamètre est égale à 2.

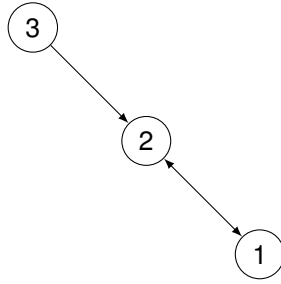


FIGURE 1.8 – Exemple de sous-graphe induit

Définition 15 : Sous-graphe induit

Soit $G = (V, E)$ un graphe. Soit A un sous-ensemble de V . Le sous-graphe induit $G[A]$ de G est un graphe ayant comme nœuds, les nœuds appartenant à A et pour arêtes, les arêtes de E ayant comme extrémités les nœuds de A . Mathématiquement cela se traduit ainsi :

$$G[A] = (A, \{(u, v) | u, v \in A \text{ et } (u, v) \in E\})$$

Le sous-graphe induit par l'ensemble $A = \{1, 2, 3\}$ du graphe de la figure 1.7 est illustré dans la figure 1.8.

Définition 16 : Graphe pondéré

Un graphe pondéré est un triplet $G = (V, E, W)$ où (V, E) définit un graphe et W est une fonction qui associe à chaque arc un poids.

Dans le cas des réseaux de capteurs, la fonction W permet d'associer à chaque capteur le coût énergétique dépensé pour, par exemple, la transmission d'un message.

Nous avons choisi de pondérer les sommets du graphe au lieu des arcs. Ces deux types de pondération sont équivalentes, en effet, pondérer un sommet est équivalent à pondérer toutes les arêtes incidentes à un nœud (dans le cas d'un graphe non orienté) ou tous les arcs sortant d'un nœud (dans le cas d'un graphe orienté).

Définition 17 : Poids d'un chemin

Étant donné un graphe $G = (V, E)$. Le poids $w(P)$ d'un chemin $P = (u_1, u_2, \dots, u_k)$, où les u_i sont des nœuds du graphe G , se calcule de la manière suivante :

$$w(P) = \sum_{i=1}^{k-1} w(u_i, u_{i+1}) = w(u_i)$$

Définition 18 : Graphe connexe

Un graphe G non orienté est dit connexe si, pour chaque paire de sommets u et v l'un des deux cas suivants est réalisé :

- $u = v$
- $u \neq v$ et il existe une chaîne de u vers v



FIGURE 1.9 – Exemple de graphe complémentaire

Un graphe connexe permet de modéliser un réseau de capteurs connexe, autrement dit, ne contenant pas de nœud isolé.

Définition 19 : Graphe fortement connexe

Un graphe G orienté est fortement connexe si pour toute paire de sommets u et v distincts, il existe un chemin de u vers v et un chemin de v vers u .

Un graphe fortement connexe spécifie un réseau de capteurs fortement connexe, c'est-à-dire un réseau dans lequel tout nœud peut envoyer un message à n'importe quel autre nœud.

Définition 20 : Graphe complémentaire $\bar{G}$

Un graphe $\bar{G}$ est dit complémentaire du graphe $G = (V, E)$, lorsqu'il est défini de la manière suivante : $\bar{G} = (V, V \times V \setminus E)$.

Définition 21 : Graphe transposé G^T

Soit $G = (V, E)$ un graphe orienté. Un graphe G^T est dit transposé du graphe G si son ensemble de sommets est égal à V et son ensemble d'arêtes se définit de la manière suivante : $E^T = \{(y, x) | (x, y) \in E\}$.

Définition 22 : Arbre de Steiner

Soit $G = (V, E)$ un graphe et S un sous-ensemble de nœuds de V . Un arbre de Steiner est un arbre reliant tous les nœuds de S avec des arêtes de E . Les sommets de l'ensemble S se nomment les sommets terminaux et les autres sommets de V utilisés pour construire l'arbre Steiner sont appelés sommets de Steiner.

La construction d'un arbre de Steiner est également étudiée dans le cas des graphes pondérés. Il s'agit de trouver un arbre Steiner de poids minimal.

L'image 1.9 illustre un graphe (à gauche) et son graphe complémentaire (à droite).

Toutes les définitions que nous venons d'énoncer ainsi que les notations respectives sont résumés dans le tableau 1.2.

Tableau des notations	
Notation	Explication
V	Ensemble des sommets
V^*	Ensemble des sommets dépourvu de la station de base
E	Ensemble des arêtes
$G = (V, E)$	Graphe G composé de l'ensemble des sommets V et des arêtes E
n	Cardinal de l'ensemble V
m	Cardinal de l'ensemble E
$N(i)$	Ensemble des voisins du sommet i
(i, j)	Arête joignant de i à j
d_{ij}	Distance euclidienne entre les sommets i et j
$N^+(x)$	Successeurs de x
$N^-(x)$	Prédécesseurs de x
$N^+[x]$	Voisinage fermé sortant de x . $N^+[x] = N^+(x) \cup \{x\}$
$N^-[x]$	Voisinage fermé entrant de x . $N^-[x] = N^-(x) \cup \{x\}$
$d^+(i)$	Degré sortant de i
$d^-(i)$	Degré entrant de i
$d(i)$	Degré de i , c'est-à-dire la somme de $d^+(i)$ et $d^-(i)$
Δ	Degré maximum du graphe
δ	Degré minimum du graphe
$diam(G)$	Diamètre du graphe G
$\overline{G}$	Graphe complémentaire de G
$G[U]$	Sous-graphe induit de G
G^T	Graphe transposé de G

TABLE 1.2 – Tableau des notations

Dans notre cas, le réseau de capteur est modélisé par un graphe orienté pour représenter le fait que les différents nœuds peuvent avoir différentes portées. Nous utilisons, donc, le modèle de Graphe de Disque présenté dans la section 1.1.2.2. De plus, nous souhaitons modéliser le coût énergétique que chaque nœud dépense pour chaque transmission. Pour cela nous associons un poids à chaque nœud correspondant à ce coût.

<i>Graphe $G = (V, E, W)$</i>	<i>Réseau de capteurs</i>
$V = \{0, \dots, n - 1\}$ Ensemble des sommets	Capteurs sans fil
$E \subseteq V \times V$ Ensemble des arêtes	Liens de communication sans fil entre les capteurs
$W : V \rightarrow R$ Ensemble des poids associés aux sommets	Énergie dépensée pour la transmission d'un message à tous les voisins

TABLE 1.3 – Modélisation d'un réseau de capteur via la théorie des graphes

Le tableau 1.3 résume les correspondances entre réseau de capteurs et graphe.

1.3/ PROBLÈMES

Dans cette section nous présentons les problèmes étudiés dans le cadre de cette thèse : la diffusion et le routage multi-couche dans les réseaux de capteurs sans fil. Comme vu dans précédemment, nous modélisons le réseau de capteurs par un graphe pondéré de disques. Dans la suite, nous donnons la définition de la diffusion et du routage à l'aide de la théorie des graphes.

1.3.1/ DIFFUSION

La diffusion est une opération communément utilisée dans les réseaux filaires ou non. Elle permet de répandre une information à l'ensemble du réseau. La définition formelle est la suivante.

Définition 23 : Diffusion

La diffusion (*Broadcast*) consiste à envoyer le même message à tous les nœuds du réseau.

Une méthode naïve pour effectuer la diffusion est de charger chaque capteur de ré-émettre le message reçu. Le problème de cette méthode est que tous les capteurs effectuent une transmission. Par conséquent, le coût de la diffusion est donné par la somme des coûts de tous les nœuds constituant le réseau. Cela veut dire que tous les nœuds du réseau sont sollicités et donc que nous consommons l'énergie de chacun d'entre eux. Dans la littérature, différentes approches ont été proposées pour effectuer la diffusion de manière plus efficace. L'idée est de définir un sous ensemble des capteurs qui est en charge de la re-transmission. Il existe différentes approches : probabilistes, par aires et dorsales.

Définition 24 : Dorsale

Une dorsale (*Backbone*) est l'ensemble des nœuds chargés de transmettre le message lors d'une diffusion.

Dans la théorie des graphes, une dorsale se définit par un ensemble dominant connexe. Tout d'abord, définissons un ensemble dominant :

Définition 25 : Ensemble dominant (DS)

Soit $G = (V, E)$ un graphe. Un sous-ensemble S de V est un ensemble dominant (*Dominating Set*) si $\forall v \in V \setminus S, \exists u \in S$ tel que $(u, v) \in E$. Autrement dit, tout nœud du réseau est soit dans l'ensemble dominant soit il est voisin d'un nœud appartenant à l'ensemble dominant.

Pour que le message se propage depuis la station de base, nous avons besoin que les nœuds qui retransmettent soient voisins entre eux, d'où la nécessité d'ajouter la notion de connexité présentée dans la prochaine définition.

Définition 26 : Ensemble dominant connexe (CDS)

Soit $G = (V, E)$ un graphe. L'ensemble dominant connexe (*Connected Dominating Set*) est un sous-ensemble S de V tel que S est dominant et le graphe induit $G[S]$ par S est connexe.

Parfois, il est intéressant de construire une dorsale composée du moins de nœuds possibles. Par exemple, si on considère que tous les nœuds ont la même portée de transmission C , moins de nœuds retransmettent, moins d'énergie est dépensée. Pour ce faire, nous construisons un ensemble dominant connexe de taille minimale se définissant comme suit.

Définition 27 : Ensemble dominant connexe de taille minimale (MCDS)

Soit $G = (V, E)$ un graphe. Un ensemble $S \subseteq V$ est un ensemble dominant connexe de taille minimale (*Minimum Connected Dominating Set*) si S est un ensemble dominant connexe et s'il est de taille minimale [6], c'est à dire que la suppression de n'importe quel nœud de S rend S non dominant et/ou non connexe.

Construire un ensemble dominant connexe de taille minimale est un problème NP-Difficile [4].

Pour prendre en compte des nœuds de poids différents tout en minimisant l'énergie, nous construisons un ensemble dominant connexe de poids minimal. La définition formelle est la suivante :

Définition 28 : Ensemble dominant connexe de poids minimal (MWCDs)

Soit $G = (V, E, W)$ un graphe. Un sous-ensemble S de V est un ensemble dominant connexe de poids minimal (*Minimum Weighted Connected Dominating Set*) si S est un ensemble dominant connexe et s'il est de poids minimal. [6]

La figure 1.10 montre un exemple de réseau de capteurs. L'image 1.10b représente le réseau de capteurs considéré, modélisé par un graphe orienté. Il se compose de l'ensemble des nœuds V contenant les nœuds A, B, C, D, E, F, G et H . Le tableau 1.10a montre le poids associé à chaque nœud. Ce poids représente le coût énergétique que chaque nœud dépense pour effectuer une transmission. Un ensemble dominant connexe du réseau de l'image 1.10b se constitue des nœuds A, B et G . Cette dorsale a une taille de 3 et a un poids de 7. Un ensemble dominant connexe de taille minimale du même réseau est l'ensemble composé des nœuds A et C . Cet ensemble se caractérise par une taille de 2 et un poids de 5. Pour finir, un exemple d'ensemble dominant connexe de poids

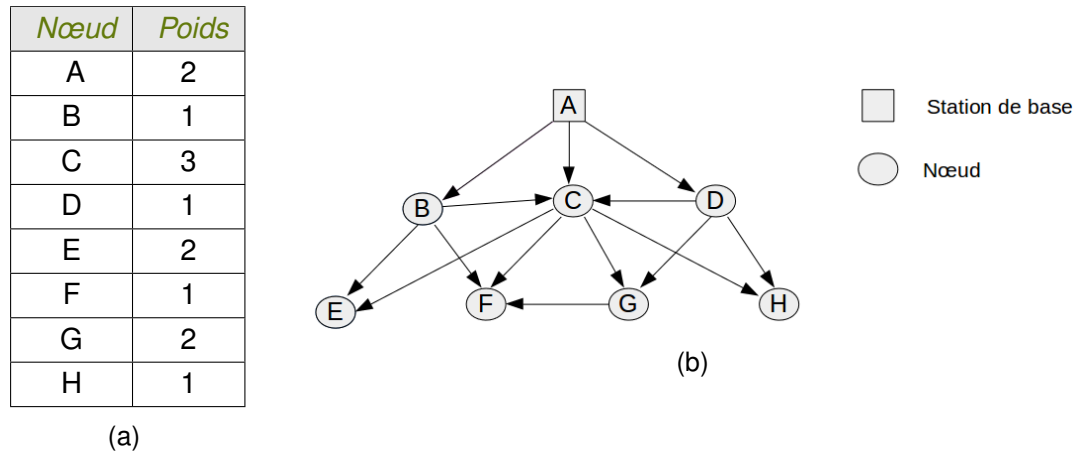


FIGURE 1.10 – Exemple d'ensemble dominant connexe

minimal est l'ensemble constitué des nœuds *A*, *B* et *D*. Dans ce cas, la dorsale a une taille de 3 et un poids de 4.

1.3.2/ ROUTAGE MULTI-COUCHE

Au cours de la thèse, nous nous sommes intéressés au routage dans les réseaux de capteurs. En particulier, au routage multi-couche entre les couches Réseau et Liaison des données. Pour préciser le problème traité, nous présentons d'abord les concepts de base, à savoir le modèle en couche et la méthode du cross-layering.

1.3.2.1/ LE MODÈLE EN COUCHES

Généralement, les processus communicants d'un réseau sont divisés en sept couches suivant le modèle OSI (*Open Systems Interconnection*). Ce modèle a été proposé par l'organisme ISO (*International Organisation for Standardization*). Étant donné les caractéristiques propres aux réseaux de capteurs, seulement cinq couches sont utilisées [51, 57] : la couche physique, la couche de liaison de données, la couche réseau, la couche de transport et la couche application, comme indiqué dans la figure 1.11.

La couche physique est responsable de transmission et réception de signaux sur un support de communication physique, y compris la génération de fréquence, la modulation du signal, la transmission et la réception, le chiffrement des données, etc.

La couche liaison de données est principalement responsable du multiplexage du flux de données, de la transmission et réception de trames de données, de l'accès au medium ainsi que du contrôle des erreurs. Elle est composée de deux sous-couches : la sous-couche de contrôle d'accès au canal (*Media Access Control* ou *MAC*) et la sous-couche de contrôle de la liaison logique (*Logical Link Control* ou *LLC*). La première sous-couche détermine, pour un capteur, la possibilité et le moment pour accéder au canal de communication. Les protocoles permettant la gestion de cette sous-couche sont appelés protocoles MAC. La deuxième contrôle le flux de données ainsi que les erreurs



FIGURE 1.11 – Modèle en couche pour les réseaux de capteurs

permettant ainsi la fiabilité des protocoles MAC.

La troisième couche se nomme couche Réseau. La couche Réseau est responsable principalement de l'acheminement et du routage des paquets dans le réseau.

La quatrième est la couche Transport. Elle garantit le transfert de paquets ou d'octets bruts sans corruption, perte, ré-ordonnancement ou duplication. Elle est responsable de gérer les communications de bout en bout.

La couche Application contient une variété de protocoles propres aux diverses applications du réseau de capteurs. Elle constitue une interface entre l'utilisateur final et le réseau de capteurs permettant une interaction transparente avec celui-ci.

La majorité des protocoles proposés pour les réseaux de capteurs suivent l'architecture de protocole en couches traditionnelle présentée dans la figure 1.11. Melodia et al. [25] soulignent que ces types de protocoles atteignent de très hautes performances pour les métriques relatives à chacune de ces couches individuelles mais, ils ne sont pas optimisés conjointement pour maximiser les performances globales du réseau tout en minimisant la dépense énergétique. C'est pour cela, que les recherches se sont tournées vers l'optimisation conjointe inter-couches présentée dans la section suivante.

1.3.2.2/ LE CROSS-LAYERING

La conception cross-layer indique que les paramètres de deux ou plusieurs couches peuvent être récupérés et/ou modifiés pour atteindre un objectif d'optimisation. Le concept de cross-layer a été proposé pour la première fois pour les réseaux TCP/IP, lorsque des liaisons sans fil étaient déployées [27]. Étant donné que la pile TCP/IP a été proposée pour les connexions filaires, l'utilisation de cette nouvelle approche a entraîné une diminution des performances. Dernièrement, la recherche cross-layering est un domaine qui fait l'objet de plus d'attention dans la recherche sur le réseau de capteurs sans fils.

Différentes solutions ont déjà été proposées dans la littérature, et au moins dans le cadre de simulations, elles se sont avérées plus performantes que leurs homologues en couches [56].

D'après Fedor et al. [55], il existe deux types d'approches cross-layer.

La première est une approche inter-couche. Considérons deux couches : Couche

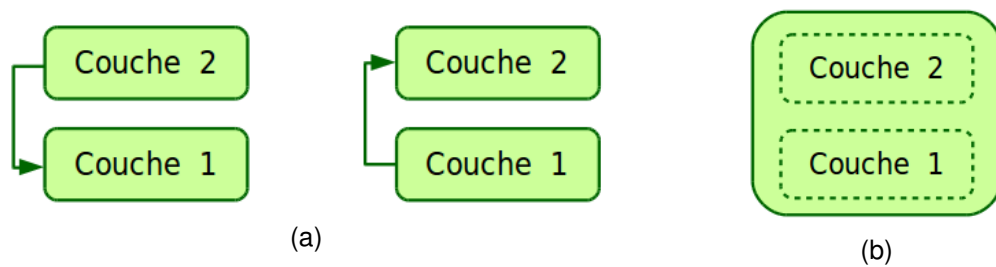


FIGURE 1.12 – Cross-layering inter-couche (à gauche) et par fusion de couche (à droite)

1 et Couche 2. Dans ce premier type de cross-layering le protocole de la Couche 2 utilise les informations de la Couche 1 ou inversement, comme le montre la figure 1.12a. Par exemple, considérons la couche Réseau et la sous-couche Liaison de données. Une forme de cross-layering peut être la suivante : la sous-couche MAC utilise les informations de la couche Réseau, c'est-à-dire que les décisions temporelles sont prises en fonction des décisions spatiales.

Le deuxième type est une approche par fusion de couches. Il s'agit de définir une super-couche capable d'assurer les fonctionnalités des couches la constituant. Reprenons l'exemple de la couche Réseau et de la sous-couche Liaison de données. Dans ce cas, les décisions temporelles et spatiales des couches respectives sont prises de manière simultanée. Ceci est représenté dans la figure 1.12b.

Ces deux types de cross-layering peuvent être réalisés entre deux ou plusieurs couches et ils peuvent être basés sur une combinaison quelconque de couches [39].

1.3.2.3/ PROTOCOLES DE LA SOUS-COUCHE MAC

D'après Pal et al. [73], dans les réseaux de capteurs sans fils, les protocoles MAC peuvent être divisés en deux catégories : les protocoles avec ou sans contention. Dans la première catégorie, nous retrouvons, par exemple, ALOHA (*Additive Link On-Line Hawaii System*) et CSMA (*Carrier Sense Multiple Access*).

Dans les protocoles MAC sans contention, l'accès des nœuds au medium est divisé par rapport au temps (*Time Division Multiple Access* ou *TDMA*), ou la fréquence (*Frequency Division Multiple Access* ou *FDMA*) ou le code (*Code Division Multiple Access* ou *CDMA*). Ces méthodes permettent aux différents nœuds d'accéder au medium sans interférer les uns avec les autres et évite ainsi efficacement les collisions. Le mécanisme de contrôle d'accès au medium le plus populaire pour les réseaux ad-hoc utilise le CSMA (*Carrier Sense Multiple Access*). Toutefois, dans le cas des réseaux sans fils, il présente de nombreux inconvénients, tels que un coût énergétique élevé ou encore un délai d'accès accru. Pour ces raisons, le TDMA (*Time Division Multiple Access*) semble plus approprié pour ce type de réseaux [79]. C'est pour cela que nous nous concentrons sur le protocole TDMA.

Le nombre d'intervalles de temps dans chaque trame TDMA est appelé la longueur de la trame. Un intervalle de temps a une longueur de temps unitaire requise pour qu'un paquet soit transmis entre nœuds adjacents. Lorsque les nœuds sont proches, des collisions peuvent se produire en cas de transmissions simultanées. Par conséquent, l'utilisation de TDMA en tant que protocole MAC dans les réseaux multihop sans fil nécessite

un ordonnancement des nœuds afin d'éviter les collisions.

Un protocole TDMA a pour objectif de définir un ordonnancement. Cela consiste à déterminer un nombre d'intervalles de temps ou créneaux (*slot*). Chaque nœud se voit associer un créneau pour transmettre ses données et un certain nombre de créneaux pour recevoir les données de ses voisins.

Tout au long de la thèse, nous nous concentrons sur l'optimisation de l'énergie ainsi que la latence. C'est pour cela que dans le chapitre 5, nous proposons une approche multi-couche entre la couche Réseau et la sous-couche MAC utilisant le protocole TDMA.

Dans la section suivante, nous présentons les différentes méthodes d'optimisation applicables pour la résolution des problèmes abordés.

1.4/ MÉTHODES DE RÉOLUTION

Avant de commencer à résoudre un problème, nous devons nous intéresser à la difficulté pour obtenir des solutions. En informatique, des efforts sont effectués afin de proposer des algorithmes capables de les calculer en un temps raisonnable. Pour cela, il faut proposer des algorithmes efficaces.

Un algorithme est dit efficace s'il a une complexité en temps de la forme $O(n^k)$ où n est un entier représentant la taille de l'entrée du problème traité et k est une constante. Nous disons que l'algorithme a une complexité en temps polynomial.

La classe des problèmes pouvant être résolus en temps polynomial est la **Classe P**. Plus précisément, les problèmes de la classe P peuvent être résolus en temps polynomiale sur une machine de Turing déterministe.

Les problèmes que nous traitons dans le cadre de cette thèse sont NP-difficile, c'est-à-dire qu'ils sont plus durs que n'importe quel problème appartenant à la classe NP. Les problèmes de la **classe NP** sont les problèmes qui peuvent être résolus en temps polynomiale sur une machine de Turing non-déterministe.

La question de savoir si $P = NP$ est encore une question ouverte à ce jour. Ceci signifie qu'on ne connaît pas d'algorithme efficace permettant de résoudre un problème de la classe NP, à fortiori pour des problèmes NP-difficiles.

Par conséquent, pour résoudre les problèmes traités, nous utilisons différentes méthodes suivant la taille du problème. Pour des petites instances, une méthode exacte est utilisée : programme linéaire en nombres entiers. Dans cette méthode, le problème est modélisé par un objectif à minimiser ou maximiser (on parle de fonction objectif) et par des variables soumises à des contraintes définies par des relations linéaires. Pour les instances de taille importante, nous cherchons une solution approchée, calculable en temps polynomial. Pour cela, il existe trois méthodes utilisables :

- heuristiques
- métaheuristiques
- algorithmes d'approximation

Une heuristique permet de calculer une solution mais ne donne aucune garantie sur la qualité du résultat. D'autre part, un algorithme d'approximation donne une solution dont

la qualité par rapport à l'optimal est bornée mathématiquement. Quant aux métaheuristiques, elles peuvent être considérées comme des méthodologies générales pouvant être utilisées pour résoudre des problèmes d'optimisation. Les métaheuristiques, tout comme les heuristiques, donnent une solution approchée mais ne garantissent pas l'optimalité des solutions obtenues.

Parmi les algorithmes d'approximation, il existe une famille nommée schéma d'approximation en temps polynomial (*Polynomial-Time Approximation Scheme* ou *PTAS*). Ce sont des algorithmes d'approximation prenant en argument une instance d'un problème d'optimisation et un paramètre $\epsilon \geq 0$. Ils produisent en temps polynomial, en la taille n de l'instance traitée, une solution ayant un facteur d'approximation de $1 + \epsilon$ de la solution optimale.

Les métaheuristiques peuvent être classées en deux familles : la famille des métaheuristiques avec une seule solution et celle avec des populations de solutions. Dans le cadre de nos travaux, nous utiliserons uniquement la première famille représentée par la recherche locale. Ce type de métaheuristique commence avec une solution donnée et l'améliore itérativement en choisissant une solution voisine permettant l'amélioration de la fonction objectif. La recherche est arrêtée lorsque toutes les solutions voisines ne permettent pas d'amélioration. Le principal problème avec cette méthode est que cela peut conduire à un optimum local. Pour éviter la convergence vers un optimum local, deux méthodes de recherche locale, la recherche tabou (*Tabu Search*) et le recuit simulé (*Simulated Annealing*), acceptent parfois des solutions voisines qui détériorent la fonction objectif.

Le recuit simulé est une méthode s'inspirant du recuit physique. Ce dernier est un processus utilisé en métallurgie pour améliorer la qualité d'un solide. Pour cela, il alterne des cycles de chauffage (recuit) et refroidissement. L'objectif est d'atteindre un état d'énergie minimum correspondant à une structure du métal stable. Le processus commence à une température très élevée pour laquelle le matériau est liquide. Ensuite, suit une phase de décroissance progressive de la température permettant au matériau de retrouver son état solide. Le recuit simulé a été développé par trois chercheurs d'IBM, S. Kirkpatrick et al. en 1983 [2] et indépendamment par V. Černý en 1985 [3].

Le recuit simulé commence avec une configuration arbitrairement choisie. Ensuite, une configuration voisine est choisie au hasard. Cette configuration est conservée suivant la valeur de la fonction objectif qui lui est associée. Si cette valeur améliore la fonction objectif, la nouvelle configuration est conservée. Sinon, la nouvelle configuration est acceptée selon une probabilité donnée par la formule suivante (distribution de Boltzmann) :

$$e^{\left(-\frac{\Delta E}{T}\right)}$$

Où ΔE est la différence entre les valeurs de la fonction objectif de la configuration actuelle et de la configuration voisine et T , la température actuelle du système.

Les résultats obtenus à l'aide des méthodes citées sont présentés dans la section suivante.

1.5/ RÉSULTATS OBTENUS

Ce manuscrit présente les résultats obtenus pour deux problèmes de communication connus dans les réseaux de capteurs : la diffusion et le routage multi-couche.

Dans les deux cas, le réseau traité est hétérogène. L'hétérogénéité intervient ici au niveau des capteurs, et plus particulièrement au niveau de leur puissance de transmission. Le réseau est donc modélisé par un graphe de disques dans lequel les nœuds sont pondérés par leur coût énergétique de transmission.

La diffusion est traitée via la construction d'une dorsale caractérisée par une énergie minimale et une latence bornée. Pour cela, nous proposons un programme linéaire en nombres entiers, un algorithme d'approximation, une heuristique et l'utilisation du recuit simulé.

Pour le routage, nous proposons une approche multi-couche de type fusion de couches via un programme linéaire en nombre entiers permettant de calculer la solution optimale. Pour cela, nous combinons les décisions temporelles prises par la sous-couche MAC en utilisant le protocole TDMA et les décisions spatiales prises par la couche Liaison de données.

La suite de ce manuscrit s'organise de la manière suivante. Le chapitre 2 introduit les résultats trouvés dans la littérature pour la diffusion dans les réseaux de capteurs sans fils. Le chapitre 3 présente la construction d'une dorsale pour la diffusion par le biais d'un programme linéaire en nombres entiers ainsi qu'un algorithme d'approximation. Le chapitre 4 traite la résolution de la diffusion en construisant une dorsale via une heuristique et le recuit simulé. En outre, ces méthodes sont comparées avec celles proposées dans le chapitre 3 à travers des simulations. Les résultats de littérature et la résolution du problème du routage multi-couche seront décrites dans le chapitre 5. Le dernier chapitre résume les différentes contributions présentées dans cette thèse et donne des perspectives d'amélioration possibles.



COMMUNICATIONS DANS LES RÉSEAUX SANS FILS

LA DIFFUSION DANS LES RÉSEAUX DE CAPTEURS SANS FILS

Sommaire

2.1 Ensemble dominant connexe	32
2.1.1 Graphe de disques unitaires	32
2.1.2 Graphe de disques avec liens bidirectionnels	33
2.1.3 Graphe de disques	34
2.1.4 Graphe général	37
2.1.5 Synthèse	38
2.2 Ensemble dominant connexe avec équilibrage de la charge	38
2.2.1 Graphe de disques unitaires	38
2.2.2 Synthèse	40
2.3 Ensemble dominant connexe et énergie	41
2.3.1 Graphe de disques unitaires	41
2.3.2 Graphe de disques	42
2.3.3 Synthèse	43
2.4 Ensemble dominant connexe et latence	44
2.4.1 Graphe de disques unitaires	44
2.4.2 Graphe Cordal	45
2.4.3 Graphe de disques	45
2.4.4 Graphe général	46
2.4.5 Synthèse	47
2.5 Ensemble dominant connexe tolérant aux pannes	47
2.5.1 Graphe de disques unitaires	48
2.5.2 Graphe de disques avec liens bidirectionnels	49
2.5.3 Graphe de disques	49
2.5.4 Graphe non orienté	51
2.5.5 Graphe général	52
2.5.6 Synthèse	52
2.6 Synthèse	53

Ce chapitre présente l'état de l'art de la diffusion dans les réseaux de capteurs. La diffusion au sein d'un réseau de capteurs peut se faire suivant différents types d'approches : probabilistes, par dorsales et par aires. Les approches probabilistes déterminent les nœuds chargés de transmettre en fonction d'une certaine probabilité. Ces

approches se distinguent en deux groupes suivant que la probabilité soit fixe au cours du temps ou bien évolutive : on parle d'approche statique ou adaptative. Les approches dorsales utilisent le problème de l'ensemble dominant connexe pour définir les nœuds qui sont chargés de la retransmission. Dans cette approche, il s'agit de définir les nœuds faisant partie de la dorsale. Il est possible de définir soit une seule dorsale soit plusieurs qui peuvent être utilisées à tour de rôle. Les approches par aires consistent à diviser le réseau de capteurs en plusieurs espaces réguliers. Au sein d'un espace, un seul capteur sera chargé de la transmission des informations à tous les membres de celle-ci.

Pour les approches probabilistes, étant donné leur nature non déterministe, le calcul d'une solution optimale n'est pas aisé. Les approches par aires répondent à des considérations plus pratiques et, de plus, elles conduisent à la construction d'une dorsale hiérarchique. Par conséquent, dans le cadre de la thèse, nous avons choisi de nous concentrer que sur les approches par dorsale.

Dans la suite, nous présentons les différentes modélisations de dorsales proposées dans la littérature. Elles varient suivant la métrique à optimiser. Les articles sont classés suivant le problème abordé puis la modélisation du réseau utilisée.

2.1/ ENSEMBLE DOMINANT CONNEXE

Les réseaux de capteurs sans fils se caractérisent par une absence d'infrastructure fixe. Pour pallier ce manque, la construction d'une dorsale a été proposée. Nous présentons ici tous les travaux construisant une dorsale via la construction d'un ensemble dominant connexe regroupés suivant le modèle utilisé pour représenter le réseau de capteurs.

2.1.1/ GRAPHE DE DISQUES UNITAIRES

Guha et al. [6] traitent différentes variantes de l'ensemble dominant connexe. Ils modélisent le réseau par un graphe $G = (V, E)$. Dans cette section, nous ne verrons que les problèmes utilisant un graphe non pondéré. Les autres seront présentés dans la section 2.3. Guha et al. présentent deux algorithmes permettant la résolution du problème de l'ensemble dominant connexe de taille minimale. Le premier est un algorithme glouton dont l'idée est de construire un arbre T à partir du sommet de degré maximal. À chaque étape, un sommet v ou une paire de sommet adjacents (u, v) de T sont choisis et scannés. Le choix se fait suivant le nombre de nœuds couverts, c'est-à-dire selon celui qui donne le rendement le plus élevé. Lorsqu'un sommet est scanné, les arêtes reliant v à ses voisins sont ajoutées à T . Quand une paire (u, v) est scannée, cela ajoute dans T tous les arêtes partant de u et de v . Scanner une paire effectue, d'une certaine manière, une vision anticipée (*look-ahead*) d'un sommet supplémentaire. À la fin, ce processus construit un arbre de recouvrement T . L'ensemble dominant connexe sera constitué des nœuds non feuille de l'arbre T .

Les auteurs prouvent que cet algorithme a une borne d'approximation de $2(1+H(\Delta))$ où H représente la fonction Harmonique.

Ils proposent également un deuxième algorithme dont la borne d'approximation est de $\ln(\Delta) + 3$. Cet algorithme se compose de deux phases : la première consiste à trouver

un ensemble dominant et la deuxième à connecter les nœuds dominants choisis lors de la première phase. Lors de la première phase, un nœud est choisi à chaque itération de manière gloutonne. Le nœud choisi est celui qui permet de maximiser la diminution du nombre de nœuds non couverts et/ou des composants dominants connexes. À la fin de la première phase, nous avons une collection de composants dominants connexes. Dans la deuxième phase, ces composants sont reliés récursivement par paire via une chaîne de deux sommets, jusqu'à l'obtention d'une unique composante connexe. Les deux algorithmes proposés ont une complexité polynomiale.

Guha et al. traitent également le problème de l'ensemble dominant connexe qui domine uniquement un sous-ensemble S des sommets donnés de taille k . Pour le cas des graphes non pondérés, ils proposent deux algorithmes l'un avec un facteur d'approximation de $3 \ln k$ et l'autre de $(\eta + 1)H(\delta) + \eta - 1$ où δ correspond au minimum entre Δ et k . Le premier algorithme se compose de deux étapes :

1. Construire un ensemble dominant A pour l'ensemble S en utilisant un algorithme glouton pour résoudre le problème de couverture par ensemble. L'ensemble correspondant au sommet v sera constitué de v et de ses voisins.
2. Associer à chaque élément de A un élément de S . Soit $R(A)$ cet ensemble. Ensuite, il suffit d'appliquer un algorithme d'approximation pour le problème de l'arbre de Steiner avec des nœuds de pondération unitaire permettant de couvrir $R(A)$.

Le deuxième algorithme est une version modifiée du premier. La première modification consiste à construire un ensemble B de nœuds. Chaque élément de cet ensemble domine au plus un seul nœud de l'ensemble S . Les nœuds de S restant constituent l'ensemble B' . La deuxième modification consiste à utiliser un algorithme résolvant le problème de l'arbre de Steiner avec des arêtes pondérées dans lequel l'ensemble requis sera $B \cup B'$.

Cheng et al. [12] proposent un schéma d'approximation en temps polynomial pour le calcul de l'ensemble dominant connexe de taille minimale dans un graphe de disques unitaires.

2.1.2/ GRAPHE DE DISQUES AVEC LIENS BIDIRECTIONNELS

Thai et al. [34] proposent un algorithme centralisé permettant de construire un ensemble dominant connexe pour le routage. Le réseau est modélisé par un graphe de disques avec des liens bidirectionnels. Les auteurs s'appuient sur les stables maximums définis de la manière suivante.

Définition 29 : Stable maximum

Soit $G = (V, E)$ un graphe non orienté. Un sous-ensemble S de V est un stable maximum (*Maximum independent set*) de G si les nœuds de ce sous-ensemble ne sont pas voisins et s'il est de cardinal maximum.

Thai et al. construisent d'abord un stable maximum S respectant la propriété suivante : chaque paire d'ensemble complémentaire du stable maximum a une distance de deux sauts. Pour cette première étape, les auteurs utilisent l'algorithme présenté par Wan et al. [20]. Ensuite, ils connectent les nœuds de l'ensemble S obtenu en construisant un arbre de Steiner de taille minimale. Le ratio d'approximation de l'algorithme proposé

est de $(K + 2 + \ln K)$ où $K = 5$ dans le cas des graphes de disques unitaires, sinon $K = 6(3\lceil \log_2 c \rceil + 2)$ où c représente le ratio entre les portées maximales et minimales.

Wightman et al. [53] prennent en considération un graphe $G = (V, E)$ de disques géométriques aléatoires tel que deux nœuds soient voisins si la distance entre eux est inférieure à R . Le graphe considéré est un graphe entièrement connexe et tous les liens sont bidirectionnels.

Les auteurs définissent la propriété de dominance via le problème de l'ensemble dominant total de taille minimale (*Minimum Total Dominating Set*). Celui-ci ajoute la contrainte suivante au problème de l'ensemble dominant : chaque nœud, y compris les dominants, doivent avoir au moins un dominant dans leur voisinage. Cette contrainte permet de ne pas avoir de dominants isolés. La formulation du programme linéaire se divise en deux : la définition des contraintes de dominance et la définition des contraintes permettant de construire un arbre en se basant sur la formulation du flot mono-commodité.

Pour évaluer le programme linéaire proposé, les auteurs le comparent à trois heuristiques. Pour cela, trois scénarios de tests sont utilisés. Les deux premiers visent à analyser le comportement des heuristiques et du programme linéaire face au changement de la densité du réseau. En particulier, le premier modifie le degré des nœuds en changeant la portée pour un nombre de nœuds donné. Le deuxième change la densité du réseau en augmentant le nombre total de nœuds tout en gardant fixe la portée.

Le troisième scénario utilise une topologie idéale en grille dans laquelle tous les nœuds ont le même nombre de voisins. D'après les simulations, le programme linéaire donne toujours des meilleurs résultats que les heuristiques. En moyenne, les heuristiques calculent un ensemble dominant connexe de taille minimale avec un taux d'erreur moyen de 64% par rapport à la solution optimale.

En ce qui concerne les temps de calcul, le programme linéaire obtient une solution optimale dans une durée de temps ayant la même ordre de grandeur que les heuristiques, lorsque le réseau présente peu de nœuds ou quand le degré des nœuds est élevé. Dans les autres cas de figure, le temps de calcul du programme linéaire dépasse, parfois, les trois heures de calcul.

2.1.3/ GRAPHE DE DISQUES

Du et al. [29] modélisent le réseau de capteurs par un graphe de disques. Ils considèrent que le graphe d'entrée est fortement connexe. La dorsale est modélisée par une ensemble dominant fortement connexe défini comme suit :

Définition 30 : Ensemble dominant fortement connexe

Soit $G = (V, E)$ un graphe orienté. Un ensemble dominant fortement connexe (*Strongly Connected Dominating Set*) de G est sous-ensemble S de V tel que S est un ensemble dominant et $G[S]$ est fortement connexe.

Les auteurs soulignent que construire un ensemble dominant fortement connexe est un problème NP-Difficile car construire un ensemble dominant connexe de taille minimale dans les graphes de disques unitaires est un problème NP-Difficile.

Du et al. proposent deux algorithmes ayant un facteur d'approximation constant : l'algorithme CDS-BFS (Connected Dominating Set using the Breath First Search tree)

et CDS-MSN (Connected Dominating Set using the Minimum Nodes Steiner tree). Les deux algorithmes se divisent en deux phases, la première étant commune aux deux. La première phase consiste à trouver l'ensemble dominant S en utilisant une méthode gloutonne. À chaque itération, le nœud ayant la plus grande portée est choisi parmi les nœuds non encore traités. Ce nœud est ajouté dans S et tous les nœuds de son voisinage sortant fermé sont enlevés de l'ensemble de nœuds non traités. On ré-itére tant que tous les nœuds n'ont pas été traités, c'est-à-dire que tout nœud est soit dominant soit dominé.

La deuxième étape de l'algorithme CDS-BFS consiste à construire deux arbres orientés :

1. $T^f(s)$: On choisit comme nœud de départ s , le nœud ayant la plus grande portée parmi les nœuds de l'ensemble S . Un arbre de racine s est construit en utilisant le parcours en largeur.
2. $T^b(s)$: On construit un graphe G' ayant les mêmes sommets que G et ayant comme arêtes, les arêtes inversées de G . On effectue le même traitement que celui décrit pour $T^f(s)$; L'arbre $T^b(s)$ est obtenu en inversant à nouveau les arêtes.

L'union de ces deux arbres constitue l'ensemble dominant fortement connexe.

La deuxième étape de l'algorithme CDS-MSN essaye de diminuer la taille de l'ensemble dominant fortement connexe en diminuant le nombre de nœuds nécessaires pour connecter les nœuds de l'ensemble dominant S . Pour ce faire, ils construisent l'arbre de Steiner orienté avec un nombre minimal de nœuds de Steiner. Un nœud de Steiner est un nœud de l'arbre qui n'est pas une feuille. On choisit un nœud de départ s ayant la plus grande portée. On construit deux arbres $T^f(s)$ et $T^b(s)$ de racine s en utilisant, cette fois-ci, l'arbre de Steiner pour connecter l'ensemble S au lieu d'utiliser un parcours en largeur comme pour l'algorithme CDS-BFS.

L'algorithme CDS-BFS construit un ensemble dominant fortement connexe dont la taille est bornée par $12(c + \frac{1}{2})^2 \text{OPT} + 18.5(c + \frac{1}{2})^2$. L'algorithme CDS-MSN construit un ensemble dominant fortement connexe de taille au plus $(2.4(c + \frac{1}{2})^2 + 2 + 4 \ln(2c + 1))\text{OPT} + 3.7(c + \frac{1}{2})^2$. Où c représente le ratio entre les portées maximales et minimales et OPT désigne la taille de la dorsale optimale.

Li et al. [48] se concentrent sur la recherche d'un ensemble dominant fortement connexe de taille minimale. Étant donné qu'un ensemble dominant fortement connexe de taille minimale est une généralisation de l'ensemble dominant connexe de taille minimale, il est aussi difficile à approximer que celui-ci.

Les auteurs proposent un algorithme d'approximation ayant comme ratio d'approximation $(3H(n-1) - 1)$ où H représente la fonction harmonique. Pour ce faire, les auteurs introduisent le problème SAFIN (Spanning Arborescence with Fewest Internal Nodes) se définissant de la manière suivante :

Définition 31 : Arborescence de recouvrement avec nombre de nœuds internes minimal

Soit $G = (V, E)$ un graphe orienté et $s \in V$ un nœud source. Une arborescence de recouvrement avec nombre de nœuds internes minimal (*Spanning Arborescence with Fewest Internal Nodes* ou *SAFIN*) est une s -arborescence couvrante T minimisant le nombre des nœuds ayant un degré sortant non nul.

Li et al. proposent un algorithme glouton apportant une réponse à ce problème et ayant un facteur d'approximation de $(1.5H(n-1) - 0.5)$. L'algorithme proposé calcule une

arborescence couvrante et de racine s à partir du graphe G . Ensuite, le graphe G^R est calculé à partir de G en inversant la direction de chaque arc. Pour finir, ils calculent une deuxième arborescence couvrante et de racine s à partir du graphe G^R . L'union de ces deux arborescences couvrantes constitue un ensemble dominant fortement connexe de taille minimale.

Thai et al. [41] présentent les algorithmes BFS-SCDS (*Strongly Connected Dominating Set using Breadth First Search*) et MSN-SCDS (*Strongly Connected Dominating Set using Minimum number of Steiner Nodes*) permettant la construction d'un ensemble dominant fortement connexe dans un réseau par un graphe de disques. Le premier algorithme a déjà été présenté dans l'article [29]. Le deuxième est similaire à la différence que lors de la connexion des nœuds dominants en construisant les deux arbres de Steiner $T^f(s)$ et $T^b(s')$, il est possible de choisir deux nœuds racine différents pour les deux arbres. L'ensemble dominant fortement connexe final est obtenu par l'union des arbres $T^f(s)$ et $T^b(s')$ et du plus court chemin entre les nœuds s et s' .

La différence réside également sur le fait que ces algorithmes sont utilisés sur un graphe n'ayant pas que des liens bidirectionnels.

Pour construire une dorsale pour le routage dans un graphe de disque, il faut ajouter une caractéristique à la définition de l'ensemble dominant fortement connexe. Cela se définit de la manière suivante :

Définition 32 : Ensemble dominant fortement connexe et absorbant

Soit $G = (V, E)$ un graphe orienté. Un sous-ensemble S de V est un ensemble dominant fortement connexe et absorbant (*Strongly Connected Dominating and Absorbing Set*) de G si S est un ensemble dominant fortement connexe et s'il est absorbant.

Un ensemble est absorbant s'il respecte la définition suivante.

Définition 33 : Ensemble absorbant

Soit $G = (V, E)$ un graphe orienté. Un sous-ensemble S est dit absorbant si tout nœud n'appartenant pas à S possède au moins un voisin sortant appartenant à l'ensemble S .

Les auteurs proposent une adaptation de l'algorithme MSN-SCDS proposé dans l'article [29] afin de construire un ensemble dominant fortement connexe et absorbant. Ce nouvel algorithme se nomme Ext-SCDS (*Extended Strongly Connected Dominating Set*). Il consiste à construire, dans un premier temps, un ensemble dominant fortement connexe C en utilisant l'algorithme MSN-SCDS. Dans un second temps, il ajoute des nœuds supplémentaires de manière itérative jusqu'à l'obtention d'un ensemble dominant fortement connexe et absorbant.

Les auteurs évaluent les algorithmes MSN-SCDS et BFS-SCDS par le biais de simulations. Ils évaluent la taille respective entre la dorsale construite par MSN-SCDS et BFS-SCDS en faisant varier la densité du réseau et la rapport entre la portée de transmission maximale et minimale. Pour modifier la densité du réseau, les auteurs ont fait varier le nombre de nœuds dans un espace fixe ou varier la taille de l'espace en gardant un nombre de nœuds fixe. Dans tous les scénarios étudiés, la taille de la dorsale construite par l'algorithme MSN-SCDS est toujours plus petite que celle de l'algorithme BFS-SCDS.

À travers des simulations les auteurs évaluent le nombre de nœuds nécessaires pour étendre un ensemble dominant fortement connexe afin qu'il devienne également absorbant. Pour cela, d'après les scénarios étudiés, la taille du nouvel ensemble construit est d'au plus deux fois celle d'ensemble obtenu par l'algorithme MSN-SCDS.

Park et al. [36] se concentrent sur le problème de l'ensemble dominant et absorbant fortement connexe de taille minimale dans les graphes de disques. Ce problème est NP-Difficile. Les auteurs proposent un algorithme d'approximation ainsi que deux heuristiques pour la résolution de ce problème. Les auteurs considèrent que le graphe d'entrée est fortement connexe.

L'algorithme d'approximation se base sur la construction d'un arbre couvrant dominant et absorbant. Pour ce faire, ils construisent un arbre "outgoing" et un arbre "incoming" qui sont couvrants et possédant comme racine un nœud arbitraire r à partir d'un graphe G donné. Après avoir effectué l'union de ces deux arbres, l'ensemble dominant et absorbant fortement connexe est donné par les nœuds non feuille résultants de cette union. La taille de la dorsale produite est borné par $9.6(c + 1/2)^2 \text{OPT} + 14.8(c + 1/2)^2$ où c représente le ratio entre la portée de transmission maximale et minimale et OPT désigne la solution optimale.

Les deux heuristiques proposées fonctionnent de manière analogue : elles construisent d'abord un ensemble dominant et absorbant S puis ajoutent des nœuds de façon gloutonne pour rendre l'ensemble S connexe.

La première heuristique choisit le nœud ayant la plus grande portée dans l'ensemble S qui constituera la racine de l'arbre. Ils construisent une arborescence (*outconnected tree*) $A_o(1)$ de racine s . Puis, ils renversent le sens des arcs dans le graphe et, ensuite, ils reconstruisent une arborescence $A_o(2)$ de racine s . Après cela, ils renversent à nouveau les arcs dans $A_o(2)$, en obtenant ainsi une anti-arborescence (*inconnected tree*) $A_i(2)$. L'union des arbres $A_o(1)$ et $A_i(2)$, en excluant les feuilles, donne un ensemble dominant et absorbant fortement connexe.

La deuxième heuristique, après avoir construit l'ensemble S , fusionne deux composantes fortement connexes en choisissant celles générant le coût minimal. Le coût considéré pour le choix glouton, est défini comme le nombre de nœuds à ajouter pour les connecter.

2.1.4/ GRAPHE GÉNÉRAL

Fan et al. [58] proposent une formulation en programmation entière mixte pour les problèmes de l'ensemble dominant de taille minimale et l'ensemble dominant connexe de taille minimale. Les auteurs illustrent quatre contraintes pouvant assurer la connectivité des graphes induits par les ensembles dominants.

Les quatre contraintes permettant la connectivité sont les suivantes :

- Les contraintes de Miller-Tucker-Zemlin [58] : L'idée est de vérifier si le graphe induit par l'ensemble dominant est connexe en vérifiant si ce graphe admet un arbre de recouvrement orienté. Cette approche contient $O(|E| + |V|)$ contraintes et autant de variables de décisions.
- Les contraintes de Martin [58] : L'idée est de trouver un arbre couvrant non orienté. Cette approche contient $O(|V|^3)$ contraintes et autant de variables de décisions.

- Les contraintes sur les flots mono-commodité [58] : Cette approche consiste à fixer un nœud racine qui aura la responsabilité d'envoyer autant d'unité de "flot" qu'il y a de nœuds dans l'ensemble dominant moins un. Chaque sommet dans l'ensemble dominant consomme une unité de "flot" sauf le racine. Cette approche contient $O(|E| + |V|)$ contraintes et autant de variables de décisions.
- Les contraintes sur les flots multi-commodités [58] : Cette méthode consiste à utiliser les flots multi-commodités. Il s'agit d'associer une commodité ou un produit distinct à chaque sommet.

2.1.5/ SYNTHÈSE

Le diagramme 2.1 synthétise les travaux présentés précédemment classés suivant la modélisation du réseau. Dans chaque catégorie, les articles sont triés par ordre croissant de date de publication. Les modèles sont présentés du plus spécifique (à gauche) au plus générique (à droite).

Pour identifier les différents modèles de connexité, nous utilisons les acronymes suivants :

- UDG : Graphe de disques unitaires (*Unit Disk Graph*)
- BDG : Graphe de disques avec liens bidirectionnels (*Disk Graphs with Bidirectional links*)
- DG : Graphe de disques (*Disk Graph*)
- GG : Graphe général (*General Graph*)

2.2/ ENSEMBLE DOMINANT CONNEXE AVEC ÉQUILIBRAGE DE LA CHARGE

Les travaux présentés jusqu'à présent visent à construire une dorsale (ou ensemble dominant connexe) composée du moins de nœuds possible afin de limiter les nœuds sollicités. Toutefois, lorsque la charge de travail de chaque nœud dominant n'est pas équilibrée, les nœuds dominants très chargés épuisent leur énergie plus rapidement. Cela peut conduire à la mort d'un nœud qui pourrait entraîner une déconnexion du réseau. Autrement dit, cela peut réduire la durée de vie du réseau. Dans cette section, nous présentons les travaux prenant en compte l'équilibrage de la charge lors de la construction d'une dorsale.

2.2.1/ GRAPHE DE DISQUES UNITAIRES

He et al. [65] visent à construire une dorsale en tenant compte de la distribution de la charge ainsi que de sa taille. Ce problème est NP-Complet. Pour le résoudre, les auteurs traitent le problème en deux étapes. La première étape consiste à trouver l'ensemble indépendant maximal tel que le degré maximal de tous les nœuds indépendants soit minimisé. Cette phase est effectuée par un programme linéaire relaxé ayant un facteur de d'approximation de $O(\ln(n))$. La deuxième étape consiste à chercher un ensemble

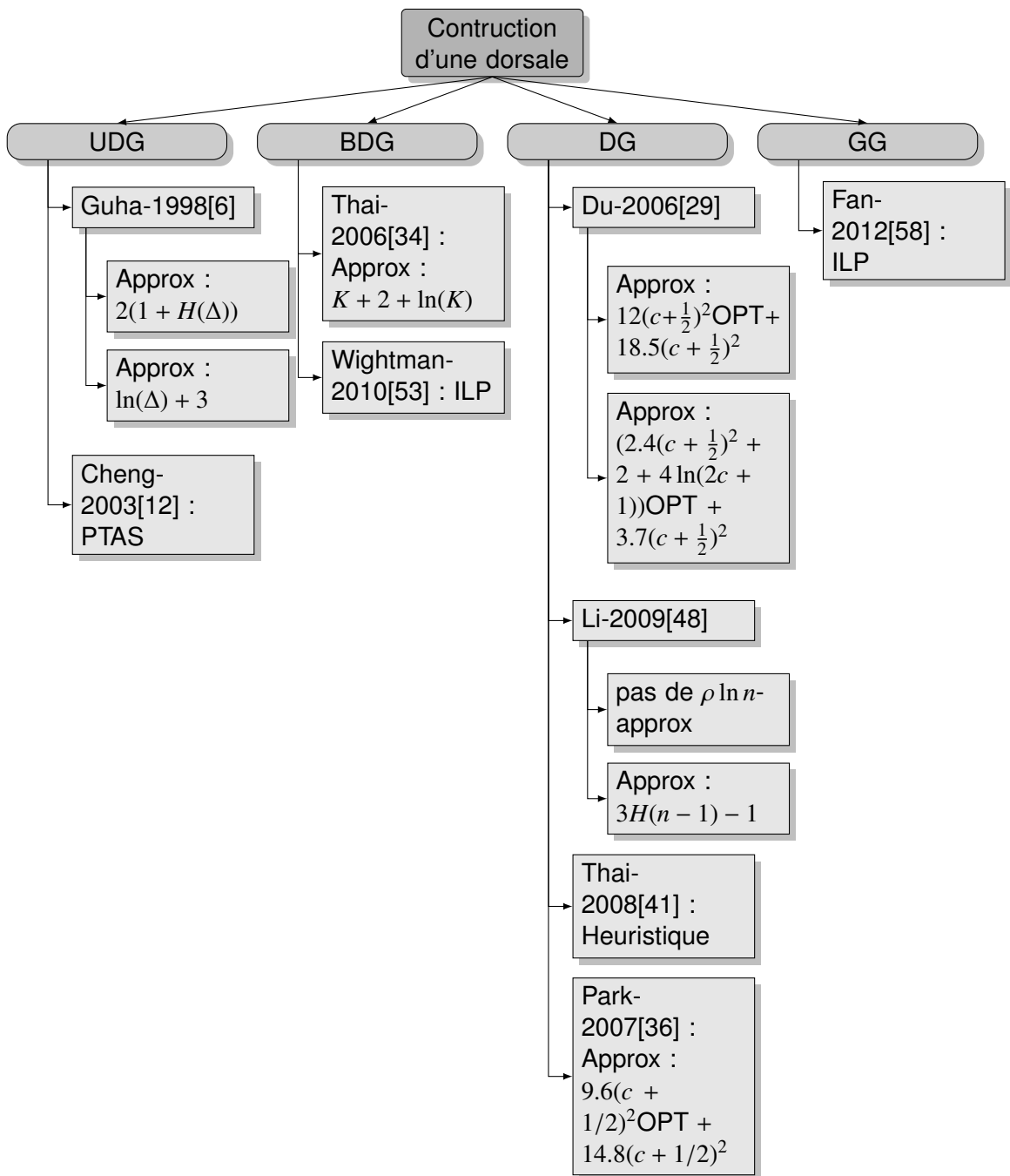


FIGURE 2.1 – Résumé des travaux proposant la construction d'une dorsale

minimal de nœuds permettant la connexion de l'ensemble précédemment trouvé. L'union de ces deux ensembles donne une dorsale équilibrée en charge et de taille minimale. À la fin, la dorsale sera composée d'au plus deux fois le nombre de nœuds de l'ensemble indépendant maximal précédemment construit. Ce qui donne une $O(\ln(n))$ -approximation.

Pour finir, les auteurs étudient le problème de l'allocation des nœuds n'appartenant pas à la dorsale de sorte à minimiser le degré valide. Le degré valide d'un nœud de la dorsale représente le nombre des nœuds n'appartenant pas à la dorsale et qui lui sont

affectés. Ils proposent un programme linéaire pour le résoudre. Par le biais des méthodes de relaxation et d'arrondi d'un programme linéaire, les auteurs prouvent qu'il a un facteur de d'approximation de $O(\ln(n))$.

He et al. proposent dans un nouvel article [72], un algorithme glouton permettant la construction d'un ensemble dominant connexe et équilibré en termes de charge.

Le problème abordé dans cet article se définit comme indiqué ci-dessous :

Définition 34 : Ensemble dominant connexe et équilibré en charge

Soit $G = (V, E)$ un graphe. Un sous-ensemble S de V est un ensemble dominant connexe et équilibré en charge (*Load-balanced Connected Dominating Set*) de G si S est un ensemble dominant connexe et si l'écart entre le degré de tout nœud dominant et le degré moyen est minimisé.

Afin d'évaluer la charge d'un nœud, les auteurs proposent la p -norme d'un vecteur $X = (x_1, x_2, \dots, x_n)$. Elle se définit de la manière suivante :

$$|X|_p = \left(\sum_{i=1}^n |x_i|^p \right)^{1/p} \quad (2.1)$$

En fixant $p = 2$, elle permet d'évaluer l'équilibrage de la charge des n nœuds. Plus la p -norme est petite, plus la distribution de la charge est équilibrée.

Leur idée est de construire dans un premier temps une dorsale équilibrée en charge via un algorithme glouton se basant sur la p -norme dans laquelle on utilise $|d_i - \bar{d}|$ comme information du vecteur. d_i étant le degré du nœud i et $\bar{d}$ le degré moyen. Une fois la dorsale construite, les auteurs essayent de distribuer de manière équilibrée les nœuds dominés aux divers nœuds dominants. Ils proposent pour cela, un algorithme centralisé et un algorithme distribué. Pour l'affectation des nœuds dominés aux divers dominants, les auteurs utilisent toujours la p -norme. Dans ce cas, l'information du vecteur X est donnée par l'écart entre le nombre de dominés reliés à un dominant donné et le nombre de dominés attendus par dominant. Grâce aux simulations, les auteurs démontrent que leurs méthodes prolongent la durée de vie du réseau jusqu'à 80% par rapport à l'algorithme, proposé par Wan et al. [50], auquel ils se comparent.

2.2.2/ SYNTHÈSE

Le diagramme 2.2 synthétise les travaux proposant la construction d'une dorsale équilibrée en charge. Ils sont classés suivant la modélisation du réseau et par ordre croissant de date de publication. Les modèles sont présentés du plus spécifique (à gauche) au plus générique (à droite).

Dans le diagramme, l'acronyme UDG identifie le modèle de connexité des graphes de disques unitaires (*Unit Disk Graph*).

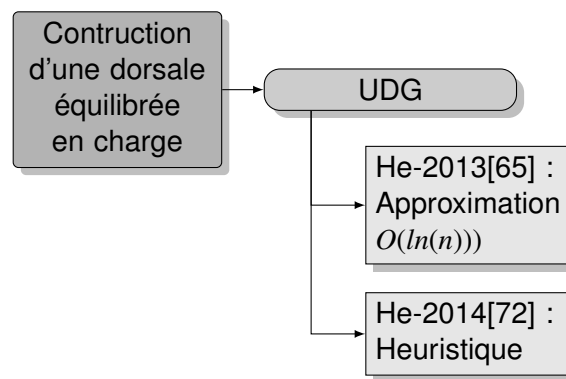


FIGURE 2.2 – Résumé des travaux proposant la construction d'une dorsale équilibrée en charge

2.3/ ENSEMBLE DOMINANT CONNEXE ET ÉNERGIE

Nous allons, à présent, exposer les travaux étudiant la construction d'une dorsale dans un graphe pondéré. L'objectif est de construire une dorsale de coût minimal. Dans les articles présentés, le coût modélise tantôt la puissance énergétique tantôt le coût énergétique pour la communication. Étant donné que le coût énergétique pour la communication est fortement lié à la puissance de transmission, nous considérons que tous les travaux présentés par la suite permettent de manière directe ou indirecte la minimisation du coût énergétique de la dorsale construite.

2.3.1/ GRAPHE DE DISQUES UNITAIRES

Guha et al. [6] proposent un algorithme recherchant un ensemble dominant de poids minimal dans un graphe de disques unitaires avec des nœuds pondérés. L'algorithme se compose des deux phases suivantes :

1. Construire un ensemble dominant D . Cette étape est effectuée en utilisant un algorithme d'approximation pour le problème de couverture par ensembles pondérés. Dans ce cas, un élément sera représenté par un sommet et chaque sommet correspondra à un ensemble constitué par le sommet lui-même ainsi que tous les nœuds appartenant à son voisinage. L'algorithme utilisé est un algorithme glouton choisissant les éléments en fonction du rapport entre leur poids et le nombre de sommets qu'ils couvrent.
2. Connecter les nœuds de l'ensemble dominant D . La connexion se fait via l'algorithme d'approximation proposé dans l'article [5] pour la construction de l'arbre de Steiner avec des nœuds pondérés.

Guha et al. prouvent que cet algorithme a un facteur d'approximation de $3 \ln n$.

Huang et al. [45] proposent un algorithme d'approximation pour le problème de l'ensemble dominant connexe minimal avec des sommets pondérés dans les graphes de disques unitaires.

L'algorithme se compose de deux phases. La phase 1 permet de calculer un ensemble dominant à l'aide de la programmation dynamique. La phase 2 connecte l'en-

semble obtenu dans la première phase via la construction d'un arbre couvrant de taille minimale. Les auteurs démontrent que la phase 1 a un ratio d'approximation de $6 + \gamma$ où γ est un nombre positif arbitraire et la phase 2 a un ratio de 4. Par conséquent, leur algorithme a un facteur d'approximation de $(10 + \gamma)$.

2.3.2/ GRAPHE DE DISQUES

Li et al. [40] traitent le problème de la construction d'un ensemble dominant fortement connexe de puissance minimale. Pour prendre en compte la puissance lors de la construction de la dorsale, les auteurs utilisent un graphe pondéré. Ce qui mène à la définition du problème suivant :

Définition 35 : Ensemble dominant fortement connexe de puissance minimale

Soit $G = (V, E)$ un graphe orienté dans lequel chaque nœud v est pondéré par sa puissance de transmission $p(v)$. Un sous-ensemble S de V est un ensemble dominant fortement connexe de puissance minimale (*Minimum Power Strongly Connected Dominating Set*) de G si S est un ensemble dominant fortement connexe et si la somme des puissances des nœuds appartenant à S est minimisée.

Étant donné qu'il s'agit d'un problème NP-Difficile, les auteurs proposent un algorithme d'approximation dont la borne est $O(\ln n)$. Pour résoudre ce problème, un autre problème est utilisé : le Minimum Energy Broadcast Routing Problem.

Définition 36 : Minimum Energy Broadcast Routing Problem

Étant donné une demande de diffusion de source s , le Minimum Energy Broadcast Routing Problem consiste à trouver un arbre de diffusion de racine s tel que le coût de la puissance totale de l'arbre soit minimisé.

Soit T un arbre orienté de diffusion de racine s . La puissance totale $P(T)$ de l'arbre T est égale à $\sum_{v \in NL(T)} p(v)$ où $NL(T)$ représente l'ensemble des nœuds de l'arbre T qui ne sont pas des feuilles.

L'algorithme proposé est le suivant :

1. Choisir de manière aléatoire le nœud source v
2. Générer un arbre de diffusion T enraciné dans le nœud v en utilisant un algorithme permettant de résoudre le problème du Minimum Energy Broadcast Routing en se basant sur l'arbre de Steiner.
3. Construire un graphe $\overleftarrow{G}$ à partir du graphe G en inversant le sens des arcs. Ensuite, créer un arbre de diffusion $\overleftarrow{T}$ dont la racine est v à partir de $\overleftarrow{G}$ en utilisant le même algorithme utilisé dans l'étape précédente pour résoudre le problème du Minimum Energy Broadcast Routing.
4. Effectuer l'union de $NL(T)$ avec $NL(\overleftarrow{T})$ pour obtenir l'ensemble dominant fortement connexe de puissance minimale de G .

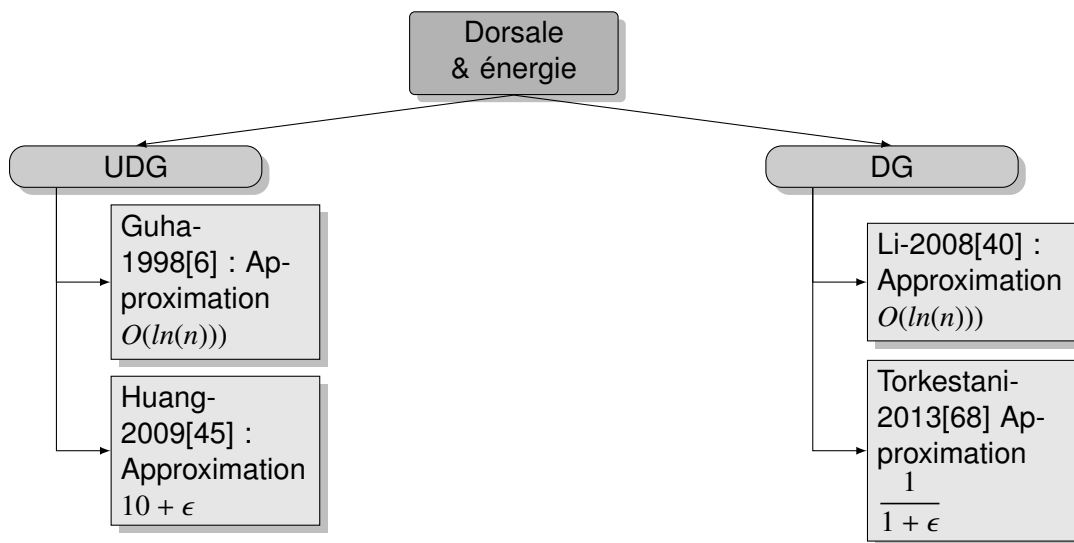


FIGURE 2.3 – Résumé des travaux proposant la construction d'une dorsale minimisant l'énergie

Torkestani [68] étudie le problème de la construction d'une dorsale pour le routage énergiquement efficace et avec une latence bornée. Pour ce faire, ils résolvent le problème de l'ensemble dominant connexe de poids minimal et de degré borné. Ensuite, ils proposent l'utilisation d'une heuristique basée sur les automates d'apprentissage. Ils associent l'énergie restante à chaque nœud via la pondération. L'objectif étant de trouver la meilleure borne/contrainte permettant d'allonger la durée de vie de la dorsale tout en minimisant le délai. À noter que le délai de la dorsale est inversement proportionnel à la contrainte sur le degré. Alors que l'énergie de la dorsale est directement proportionnelle à la contrainte sur le degré. Les auteurs ont approximé le temps d'exécution nécessaire pour que l'algorithme calcule une dorsale avec un facteur d'approximation de $\frac{1}{1-\epsilon}$ de l'optimal. La performance de l'algorithme concernant la durée de vie, la latence ainsi que la taille de la dorsale est étudiée de manière expérimentale par rapport à deux algorithmes existants. Les simulations montrent que l'approche proposée est meilleure que les algorithmes auxquels elle est comparée.

2.3.3/ SYNTHÈSE

Le diagramme 2.3 synthétise les travaux proposant la construction d'une dorsale de coût minimal. Ils sont classés suivant la modélisation du réseau ainsi que par ordre croissant de date de publication. Les modèles sont présentés du plus spécifique (à gauche) au plus générique (à droite).

Dans le diagramme, nous identifions les modèles de connexité par les acronymes suivants :

- UDG : Graphe de disques unitaires (*Unit Disk Graph*)
- DG : Graphe de disques (*Disk Graph*)

2.4/ ENSEMBLE DOMINANT CONNEXE ET LATENCE

Dans les sections précédentes, nous nous sommes intéressés aux travaux visant à optimiser la durée de vie du réseau en visant l'équilibrage de la charge ou l'énergie de la dorsale. Ici, nous présentons les articles se concentrant sur l'efficacité du réseau du point de vue de la latence.

2.4.1/ GRAPHE DE DISQUES UNITAIRES

Dans l'article [35], Li et al. considèrent la construction d'un ensemble dominant connexe de taille et de diamètre minimaux. Pour répondre à la problématique traitée, ils proposent un algorithme d'approximation. L'algorithme commence par choisir une racine r parmi les nœuds du graphe. Ensuite, il construit des sous ensembles V_k de V constitués des nœuds qui sont à une distance en nombre de sauts d'au plus k où k est un entier positif. L'étape suivante consiste à construire un ensemble indépendant maximal I_k sur le graphe induit par V_k pour tous les k pairs. Ensuite, l'ensemble I est calculé en effectuant l'union de tous les ensembles I_k . La dorsale sera, au départ constituée des nœuds de l'ensemble I . Des nouveaux nœuds sont ajoutés à la dorsale s'il reste des nœuds non couverts. Pour finir, les nœuds de la dorsale sont connexes via l'ajout de nouveaux nœuds.

L'algorithme proposé a les bornes d'approximation suivantes :

- au niveau de la taille de la dorsale : $11,4\text{OPT} + 1,6$
- du point de vue du diamètre de la dorsale : $3\text{diam}(\text{OPT}) + 7$

Où OPT représente la solution optimale.

Sa complexité en temps est de $O(n^2)$.

Kim et al. [47] s'intéressent à la construction d'un ensemble dominant connexe dans un graphe de disques unitaires en prenant en compte les métriques suivantes : la taille, le diamètre et la distance moyenne en hop (*Average Backbone Path Length*). Ils introduisent cette dernière métrique car le diamètre prend en compte uniquement le pire des cas.

Ils proposent deux algorithmes centralisés et une version distribuée du deuxième. L'idée commune à ces algorithmes est de construire un arbre via un parcours en profondeur, de rechercher un ensemble indépendant maximal I dans cet arbre, et pour finir, connecter les nœuds de l'ensemble I .

Le premier algorithme centralisé construit une dorsale dont le facteur d'approximation est le suivant :

- la taille est d'au plus $10,359\text{OPT} + 22,879$
- le diamètre est d'au plus $3\text{diam}(\text{OPT}) + 5$

Où OPT désigne la solution optimale.

Le deuxième algorithme centralisé construit d'abord un ensemble indépendant maximal et connecte l'ensemble obtenu via un arbre de recouvrement. Son facteur d'approximation, pour la dorsale obtenue, est le suivant :

- la taille est d'au plus $6.90OPT + 15.582$
- le diamètre est d'au plus $4diam(OPT) + 6$

Pour la version distribué, seules les complexités en temps $O(n^{1.6} + |Diam(G)|)$ et en messages $O(m + n^{1.6} + \Delta \times n)$ sont données. L'algorithme est évalué via des simulations montrant qu'il est meilleur que les algorithmes auxquels il est comparé.

2.4.2/ GRAPHE CORDAL

Schaudt [66] prouve que pour tout $k \geq 1$, décider si un graphe donné admet un ensemble dominant dont le graphe induit a un diamètre d'au plus k est un problème NP-Complet. Il se tourne donc vers les graphes cordaux car, cette classe de graphes admet un tel ensemble. Un graphe cordal se définit de la manière suivante.

Définition 37 : Graphe Cordal

Soit $G = (V, E)$ un graphe. Le graphe G est dit cordal si tout cycle de longueur supérieure ou égale à quatre possède une corde.

Un corde est une arête reliant deux sommets non-adjacents d'un cycle. Pour cette classe de graphe, il propose un algorithme ayant une complexité de $O(mn)$ permettant de calculer le plus petit k pour lequel un graphe G ayant n sommets et m arêtes admet un ensemble dominant dont le graphe induit a un diamètre d'au plus k . Pour tout $k \geq 3$ un graphe cordal G admet un ensemble dominant de diamètre au plus $k - 2$ si et seulement si G a un diamètre d'au plus k . De plus, l'ensemble dominant D peut être choisi tel que chaque plus petit chemin entre deux nœuds de l'ensemble dominant soit entièrement contenu dans l'ensemble D : un tel ensemble est dit *convexe*. Schaudt prouve également qu'un ensemble dominant connexe et convexe de diamètre k peut être calculé en temps polynomial. Un graphe cordal G avec un diamètre $diam(G) \geq 3$ admet toujours un ensemble dominant connexe et convexe dont le diamètre est $diam(G) - 2$.

2.4.3/ GRAPHE DE DISQUES

Li et al. [31] se posent comme objectif la prise en compte simultanée de la consommation énergétique totale et de la latence dans les arborescences de diffusion. Ils définissent formellement le problème abordé de la manière suivante :

Définition 38 : Strongly Connected Broadcast Arborescence with bounded Transmission delay

Étant donné un ensemble de nœuds dans un réseau sans fils, construire une arborescence de diffusion fortement connexe SBA (*Strongly Connected Broadcast Arborescence*) telle que $d(SBA)$ soit borné par une borne spécifiée par l'utilisateur et $p(SBA)$ soit minimisé. $d(SBA)$ et $p(SBA)$ définissent respectivement la longueur totale et le poids total de l'arborescence SBA.

Pour résoudre ce problème, les auteurs proposent un algorithme centralisé en temps linéaire ainsi que sa version distribuée. Les contraintes que l'arborescence de diffusion SBA doit respecter sont les suivantes :

- borner la consommation totale d'énergie de tous les nœuds,
- borner la latence entre le nœud source et tous les autres nœuds du réseau,
- préserver la connectivité du réseau, c'est-à-dire construire une arborescence fortement connexe.

Le réseau sans fils est modélisé via un graphe orienté fortement connexe et pondéré $G = (V, E, W)$. Le poids est associé aux arêtes. Le poids d'une arête (u, v) représente la puissance nécessaire pour transmettre depuis u vers v .

Pour construire une arborescence de diffusion fortement connexe respectant les contraintes énergétique et de latence, les auteurs utilisent l' (α, β) -arborescence qui se définit comme suit : Étant donné $\alpha \geq 1$ et $\beta \geq 1$, une arborescence couvrante A de G est une (α, β) -arborescence de racine r , si elle respecte les deux conditions suivantes :

- pour chaque sommet u , la distance entre r et u dans l'arborescence A est au plus α fois la plus courte distance entre r et u dans le graphe G ,
- la puissance de l'arborescence A est au plus β fois la puissance d'une solution optimale.

De plus l' (α, β) -arborescence doit être fortement connexe.

L'algorithme proposé pour construire une (α, β) -arborescence se compose des étapes suivantes :

1. construire deux arborescences couvrantes de poids minimal A et A' . L'arborescence A contient les chemins depuis la racine r et tout autre nœud du réseau et l'arborescence A' contient les chemins allant depuis tout nœud du réseau vers la racine r .
2. superposer les arborescences A et A'
3. construire l'arbre SPT des plus courts chemins de G
4. Parcourir A en comparant les différentes distances entre un nœud et la racine. Si cette distance est supérieure à αOPT alors remplacer ce chemin par celui dans l'arbre SPT. Faire de même pour A' .

La complexité en temps de l'algorithme proposé est de $O(n^2)$. La borne d'approximation est de $\left(1 + n \left(\frac{2}{\alpha-1}\right)^2\right)(\Delta + 1)$.

2.4.4/ GRAPHE GÉNÉRAL

Buchanan et al. [71] se concentrent sur la construction d'un ensemble connexe dominant ayant un diamètre borné à s en utilisant la notion de s -club afin de borner la distance que l'information peut traverser dans le réseau. Un s -club se définit de la manière suivante :

Définition 39 : s -club

Un s -club est un ensemble dominant minimal induisant un graphe connexe ayant un diamètre d'au plus s .

L'avantage d'un s -club est que la transmission se fait plus rapidement, ce qui permet de diminuer le délai. Les auteurs démontrent que la détermination de l'existence d'un s -club dans un graphe est un problème NP-complet même lorsque le diamètre du graphe

est de $s + 1$. Ils étudient également un cas particulier du s -club : la recherche d'une clique ou 1-club dans un graphe de diamètre 2. Ils montrent qu'il s'agit aussi d'un problème NP-complet. Les auteurs proposent un programme linéaire pour résoudre le problème du s -club dominant minimal. Le nombre de sauts séparant deux sommets quelconques dans un graphe, après la mise en place d'un s -club dominant, est d'au plus $2 + s$ sauts.

Ils soulignent qu'approximer le problème du s -club dominant minimal avec un facteur de $\nu \log \Delta$ (où ν est une constante) est NP-difficile, même lorsque nous connaissons l'existence d'un tel ensemble.

L'article souligne qu'il est possible de résoudre le problème de l'ensemble dominant connexe de taille minimale en résolvant le problème du s -club lorsque le nombre s dépend d'une donnée d'entrée comme par exemple le nombre de sommets n du graphe. Cette approche donne des résultats plus performants que les méthodes classiques.

2.4.5/ SYNTHÈSE

Le diagramme 2.4 synthétise les travaux proposant la construction d'une dorsale de latence minimale classés suivant la modélisation du réseau. Les modèles sont présentés du plus spécifique (à gauche) au plus générique (à droite). Dans chaque catégorie, les articles sont classés par ordre croissant de date de publication.

Pour identifier les différents modèles de connectivité, nous utilisons les acronymes suivants :

- UDG : Graphe de disques unitaires (*Unit Disk Graph*)
- CG : Graphe Cordal (*Chordal Graph*)
- DG : Graphe de disques (*Disk Graph*)
- GG : Graphe général (*General Graph*)

2.5/ ENSEMBLE DOMINANT CONNEXE TOLÉRANT AUX PANNES

L'épuisement de l'énergie d'un nœud peut entraîner le partitionnement du réseau de capteurs. La conséquence étant qu'une partie du réseau ne sera plus atteignable et/ou ne pourra plus atteindre le reste du réseau. Considérons le cas où un réseau de capteurs est utilisé pour la surveillance d'une forêt afin de détecter le démarrage d'un incendie. Si une partie du réseau est non connexe, cela entraîne une perte des informations récoltées sur une partie de la forêt. Par conséquent, on ne pourra pas détecter si un incendie s'est déclaré dans cette zone. Afin de palier ce problème, on construit une dorsale tolérante aux pannes. Cela se traduit par le fait que la dorsale construite doit rester dominante et connexe même si un nœud tombe en panne. Dans la suite, nous présentons les travaux construisant une dorsale tolérante aux pannes.

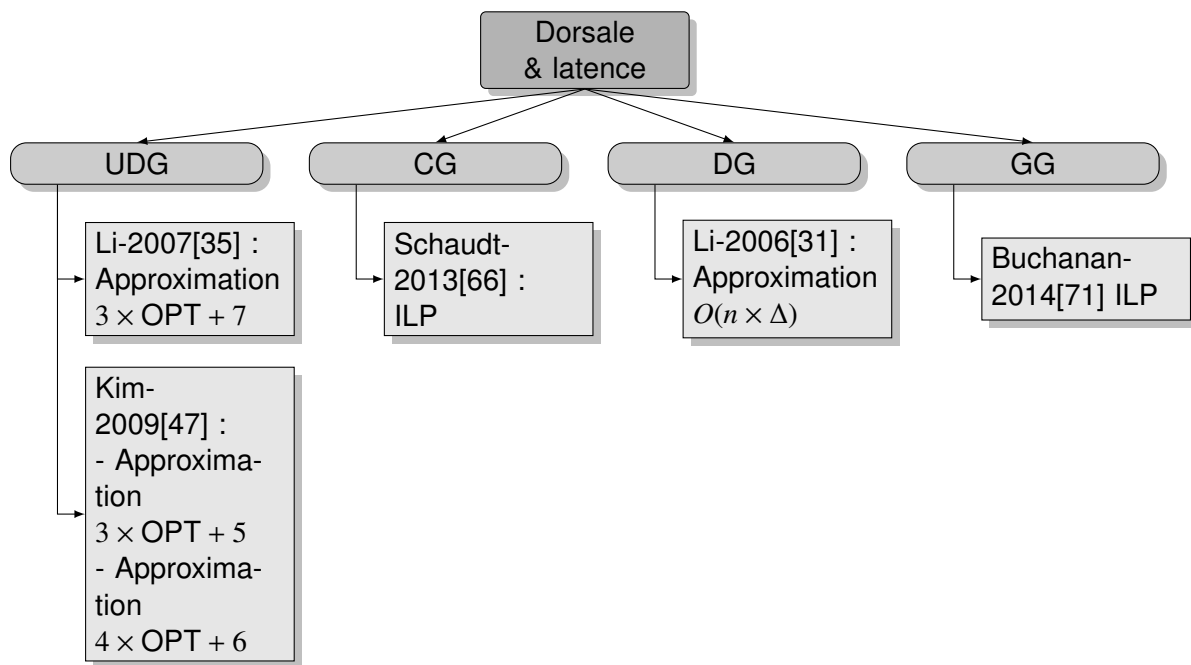


FIGURE 2.4 – Résumé des travaux proposant la construction d'une dorsale minimisant la latence

2.5.1/ GRAPHE DE DISQUES UNITAIRES

Dai et al. [28] étudient le problème de la construction d'une dorsale dans les graphes de disques unitaires qui soit tolérante aux pannes. Pour cela, ils introduisent la notion d'ensemble k -dominant et k -connexe défini de la manière suivante :

Définition 40 : ensemble k -dominant et k -connexe

Étant donné un graphe $G = (V, E)$ non orienté. Un sous-ensemble S de V est un ensemble k -dominant et k -connexe si :

- S est k -dominant : si tout nœud u de V appartient à l'ensemble S ou possède k voisins dans l'ensemble S
- S est k -connexe : si le graphe induit par S est k -sommet connexe, c'est-à-dire si en supprimant $k - 1$ nœuds de S le graphe $G[S]$ reste connexe.

D'après la définition, la dorsale construite par le biais de la construction d'un ensemble k -dominant et k -connexe tolère jusqu'à k pannes. Un ensemble k -dominant et k -connexe permet, non seulement une meilleure tolérance aux pannes, mais aussi une plus grande flexibilité de routage. Pour répondre à ce problème, ils proposent quatre algorithmes localisés qui sont des extensions de protocoles existants construisant un ensemble dominant connexe. Les algorithmes sont les suivants :

- Algorithme k -Gossip : chaque nœud devient un nœud de la dorsale suivant une probabilité donnée p_k . L'algorithme demande peu d'entête mais la probabilité p_k dépend de la taille du réseau ainsi que de sa densité.
- Algorithme k -Grid : il réduit la taille de la dorsale en choisissant B_k nœuds appartenant à la dorsale dans le voisinage de chaque nœud. Le nombre B_k dépend de la taille du réseau.

- Algorithme k -couverture : au début, tous les nœuds appartiennent à la dorsale. Ensuite, un nœud peut être enlevé de la dorsale si la condition de k -couverture est satisfaite. La condition de k -couverture est la suivante : le nœud v n'appartient pas à la dorsale si pour tout voisins u et w de v , il existe des k chemins nœud-disjoints de remplacement connectant u et w . Deux chemins sont dit nœud-disjoints si l'ensemble de leur nœuds internes sont disjoints.
- Algorithme hybride : il se base sur la coloration de graphe. D'abord il sépare le réseau en k sous-réseaux. Ensuite, il construit un ensemble dominant connexe par sous-réseau. L'union de ces k ensembles dominants connexes forme un ensemble k -dominant et k -connexe.

2.5.2/ GRAPHE DE DISQUES AVEC LIENS BIDIRECTIONNELS

Thai et al. [37] se concentrent sur le problème de la construction d'un ensemble m -dominant et k -connexe. Il s'agit d'une généralisation d'un ensemble k -dominant et k -connexe car les nombres m et k ne sont pas forcément égaux. Le réseau considéré est un graphe de disques avec liens bidirectionnels. Les auteurs proposent deux algorithmes d'approximation : un pour l'ensemble m -dominant et 1-connexe et un pour l'ensemble k -dominant et k -connexe. Sur la base de ces algorithmes, les auteurs proposent un algorithme pour construire un ensemble m -dominant et k -connexe.

L'algorithme CDSMIS (*Connected Dominating Set by Maximal Independent Sets*), construisant l'ensemble m -dominant et 1-connexe, se comporte de la manière suivante : il construit un ensemble dominant connexe de taille minimale (en utilisant la notion d'ensemble indépendant maximal) et ensuite il ajoute des nœuds afin de respecter la m -dominance. Le ratio d'approximation est de $(K + m + \ln K + 2)$ où K est la borne supérieure du nombre de voisins indépendants d'un nœud dans graphe de disques bidirectionnel (donné par l'article [34]).

En ajoutant des nœuds à l'ensemble m -dominant et 1-connexe, les auteurs construisent l'ensemble k -dominant et k -connexe. Cela est fait via l'algorithme CDSAN (*Connected Dominating Set by Adding Nodes*) dont la borne d'approximation est de $(K + \ln(K) + k + 2)(2k - 1)$. Les auteurs proposent la construction de l'ensemble m -dominant et k -connexe par le biais des étapes suivantes :

1. Utiliser l'algorithme CDSMIS pour obtenir un ensemble m -dominant et 1-connexe A
2. Utiliser l'algorithme CDSAN pour ajouter des nœuds à l'ensemble A jusqu'à obtenir un ensemble m -dominant et k -connexe

Dans le cas particulier où $k = m + 1$, les auteurs prouvent que leur algorithme construit un ensemble m -dominant et k -connexe de taille au plus $(2k + 2K - 1)(K + \ln(K) + k + 1)OPT$ où OPT représente la solution optimale.

2.5.3/ GRAPHE DE DISQUES

Tiwari et al. [38] se concentrent sur la construction d'une dorsale tolérante aux pannes. Pour cela, ils adaptent la définition d'ensemble m -dominant et k -connexe de la manière suivante :

Définition 41 : ensemble k -dominant et absorbant et k -fortement connexe

Étant donné un graphe $G = (V, E)$ orienté et deux entiers positifs k et m . Un sous-ensemble S de V est un ensemble m -dominant et absorbant et k -fortement connexe (*k -strongly connected m -dominating and absorbing set*) si :

- S est un ensemble dominant et absorbant fortement connexe,
- le graphe induit par S est k -connexe,
- tout nœud u n'appartenant pas à S est dominé par au moins m nœuds de C ,
- tout nœud u n'appartenant pas à S est absorbé par au moins m nœuds de C . Autrement dit, le nœud u possède au moins m voisins sortant appartenant à C .

Les auteurs proposent deux heuristiques pour résoudre les problèmes suivants :

- problème 1 : Construction d'un ensemble m -dominant et absorbant et 1-fortement connexe
- problème 2 : Construction d'un ensemble 1-dominant et absorbant et k -fortement connexe

En combinant ces deux algorithmes, les auteurs construisent l'ensemble m -dominant et absorbant et k -fortement connexe pour n'importe quelle valeur de k et de m .

L'algorithme répondant au problème 1 se compose des phases suivantes :

1. Phase 1 : Construire un ensemble dominant fortement connexe C et l'étendre afin qu'il devienne également absorbant.
2. Phase 2 : Rechercher de manière itérative un $i^{\text{ème}}$ ensemble dominant et l'ajouter à C . Étendre C afin qu'il devienne un ensemble i -dominant et absorbant fortement connexe. Cette phase est répétée $m - 1$ fois.

L'algorithme résolvant le problème 2 se compose des phases suivantes :

1. Phase 1 : Construire un ensemble dominant et absorbant fortement connexe C .
2. Phase 2 : Répéter de manière itérative $k - 1$ fois :
 - exécuter le parcours en profondeur sur G et supprimer toutes les arêtes traversées connectant les nœuds dans C ,
 - renverser le sens des arcs dans G pour obtenir G^T ,
 - exécuter le parcours en profondeur sur G^T et supprimer toutes les arêtes traversées connectant les nœuds dans C ,
 - renverser à nouveau le sens des arcs dans G^T pour obtenir G ,
 - chercher un ensemble dominant fortement connexe dans G en ajoutant des nouveaux nœuds dans C et l'étendre pour qu'il devienne absorbant.

L'ensemble C obtenu est un ensemble 1-dominant et absorbant et k -fortement connexe.

Les auteurs mettent en évidence les performances de leurs propositions via des simulations.

Dans l'article [63], Tiwari et al., étudient la construction d'un dorsale tolérante aux pannes dans un graphe de disques par le biais de la construction de l'ensemble m -dominant et absorbant et k -fortement connexe.

Les auteurs proposent un algorithme d'approximation pour la construction d'un tel ensemble pour tout entier positif k et m .

Pour obtenir un ensemble m -dominant et absorbant et k -fortement connexe, les auteurs génèrent d'abord un ensemble m -dominant et absorbant et 1-fortement connexe et ensuite ils le rendent k -connexe.

La construction d'ensemble m -dominant et absorbant et 1-fortement connexe se fait en deux phases. La première phase consiste à construire un ensemble dominant et absorbant fortement connexe C . La deuxième phase est itérative et consiste à ajouter des nœuds dans l'ensemble C afin qu'il soit m -dominant et absorbant. Cette dernière phase sera effectué $m - 1$ fois.

Une fois l'ensemble m -dominant et absorbant et 1-fortement connexe construit, pour chaque paire ordonnée de nœuds dans C , l'algorithme identifie $k - 1$ chemins disjoints de nœuds et les ajoute à C . Pour que l'algorithme soit valide, il faut que le graphe de départ soit k -connexe avec $m \geq k$.

Les auteurs énoncent une conjecture quant au ratio de performance de leur algorithme. Ce ratio vaut : $2(2k + 1)(K_{SI} + m)$ où K_{SI} est la borne supérieure du nombre de voisins semi-indépendants pour chaque nœud u du graphe. Nous introduisons la notion de voisin semi-indépendant par le biais de la définition suivante.

Définition 42 : Voisin semi-indépendant

Soit u et v deux nœuds appartenant au sous-ensemble SI de V . L'ensemble SI est dit semi-indépendant du graphe G si, et seulement si, soit les arêtes (u, v) et (v, u) n'appartiennent pas à E soit si (u, v) appartient à E alors (v, u) n'appartient pas à E et vice-versa. On dit que les nœuds u et v sont semi-indépendant entre eux.

2.5.4/ GRAPHE NON ORIENTÉ

Afin d'obtenir une dorsale plus tolérante aux pannes, Buchanan et al. [75] construisent un ensemble d -dominant k -connexe. Les auteurs considèrent un réseau modélisé par un graphe non orienté. Ils proposent trois formulations en programmation linéaire pour résoudre le problème dans les trois cas suivants : $k = d$, $k < d$ et $k > d$. Ils se basent sur les points d'articulations définis de la manière suivante.

Définition 43 : Point d'articulation

Dans un graphe, un point d'articulation est un sommet du graphe qui, si on le retire du graphe, augmente le nombre de composantes connexes.

Les formulations proposées ont un nombre exponentiel de contraintes. Toutefois, leurs relaxations peuvent être résolues en temps polynomial.

2.5.5/ GRAPHE GÉNÉRAL

Ahn et al. [76] proposent un programme linéaire en nombres entiers ainsi qu'un algorithme optimal du problème de l'ensemble m -dominant k -connexe minimal. Le réseau est modélisé par un graphe général. Les auteurs s'appuient sur la coupe $s - t$.

Définition 44 : Coupe $s - t$

Étant donné un graphe $G = (V, E)$ et deux sommets s et t de V , une coupe $s - t$ (*st-vertex cut*) est un sous-ensemble $S \subseteq V \setminus \{s, t\}$ tel que la suppression de l'ensemble des nœuds de S de G déconnecte les nœuds s et t .

Le nombre maximal k pour lequel G est k -connexe se nomme le *vertex connectivity* de G et se note $k(G)$.

Comme le programme linéaire proposé par les auteurs génère un nombre exponentiel de contraintes, ils proposent l'algorithme suivant pour obtenir une solution optimale. La première étape consiste à utiliser un sous-ensemble des contraintes définies dans le programme linéaire proposé et d'en calculer le résultat. L'étape suivante comporte la construction d'un graphe G' . Ce graphe $G' = (V', E')$ est constitué de l'ensemble V' égal à l'ensemble V privé des nœuds dont la variable de décision indiquant s'ils sont dominant est à zéro. L'ensemble E' est égal à l'ensemble E privé des arêtes incidentes aux nœuds précédemment supprimés. Ensuite, on teste si, pour toute paire de nœuds distincts, la taille de la coupe $s - t$ minimale est supérieure ou égale à $k(G)$. Si ce n'est pas le cas, la contrainte permettant de contraindre la taille d'un st-vertex cut est ajoutée pour tous les sommets s et t ayant échoués au test. De même, on teste si la taille minimale du st-vertex-cut est inférieure ou égale à $\min(m, k)$. Si le test échoue, alors on ajoute l'équation permettant de contraindre la taille d'un st-vertex cut afin que celle-ci possède une valeur d'au moins $\min(m, k)$. Les auteurs étudient également la relaxation du programme linéaire proposé. Leur algorithme est évalué via des simulations par rapport à une heuristique précédemment proposée.

2.5.6/ SYNTHÈSE

Le diagramme 2.5 synthétise les travaux proposant la construction d'une dorsale tolérante aux pannes classés suivant la modélisation du réseau. Les modèles sont présentés du plus spécifique (à gauche) au plus générique (à droite). Dans chaque catégorie, les articles sont classés par ordre croissant de date de publication.

Dans le diagramme, nous identifions les modèles de connexité par les acronymes suivants :

- UDG : Graphe de disques unitaires (*Unit Disk Graph*)
- BDG : Graphe de disques avec liens bidirectionnels (*Disk Graphs with Bidirectional links*)
- DG : Graphe de disques (*Disk Graph*)
- UG : Graphe non orienté (*Undirected graph*)
- GG : Graphe général (*General Graph*)

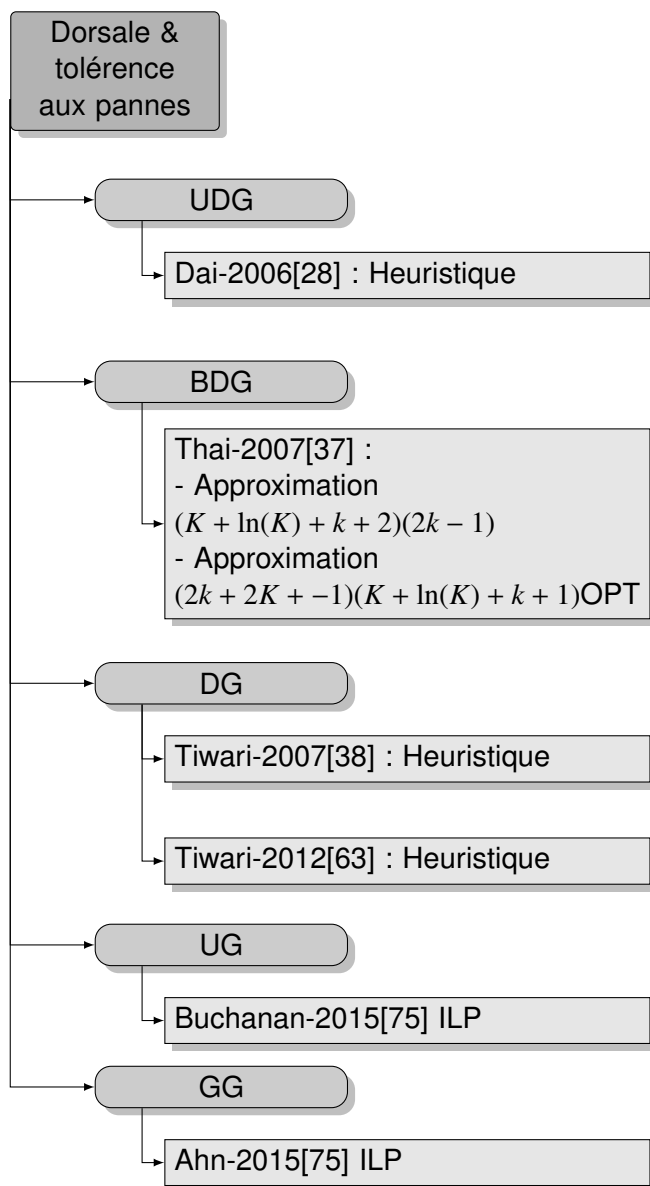


FIGURE 2.5 – Résumé des travaux proposant la construction d'une dorsale minimisant la latence

2.6/ SYNTHÈSE

Dans cette section, nous présentons un schéma illustrant les différents travaux présentés précédemment. Ceux-ci seront organisés suivant le modèle, le problème abordé et la date de publication.

Un des objectifs de cette thèse consiste à prendre en compte les réseaux de capteurs sans fil hétérogènes pour ce qui est de la portée de transmission. Pour cela, nous avons choisi de nous concentrer sur le modèle nous permettant de prendre cela en compte, à savoir les graphes de disques. Ce modèle a l'avantage de représenter un réseau dont les capteurs sont dotés de portées différentes, mais aussi de prendre en compte le fait que la connexité d'un tel réseau est assujettie à des règles géométriques. D'après le diagramme 2.5 nous constatons que, dans la littérature, les problèmes traités

sont la tolérance aux pannes, la minimisation de l'énergie et de la latence. Nous identifions, parmi les problèmes ouverts dans le cas de ce modèle, l'étude de la construction d'une dorsale avec équilibrage de la charge et la combinaison de l'optimisation de deux ou plusieurs métriques à la fois. C'est dans cette dernière direction que nos travaux s'orientent. En effet, nous avons choisi d'optimiser la latence et l'énergie dans le cas des graphes de disques comme le présentent les chapitres 3 et 4.

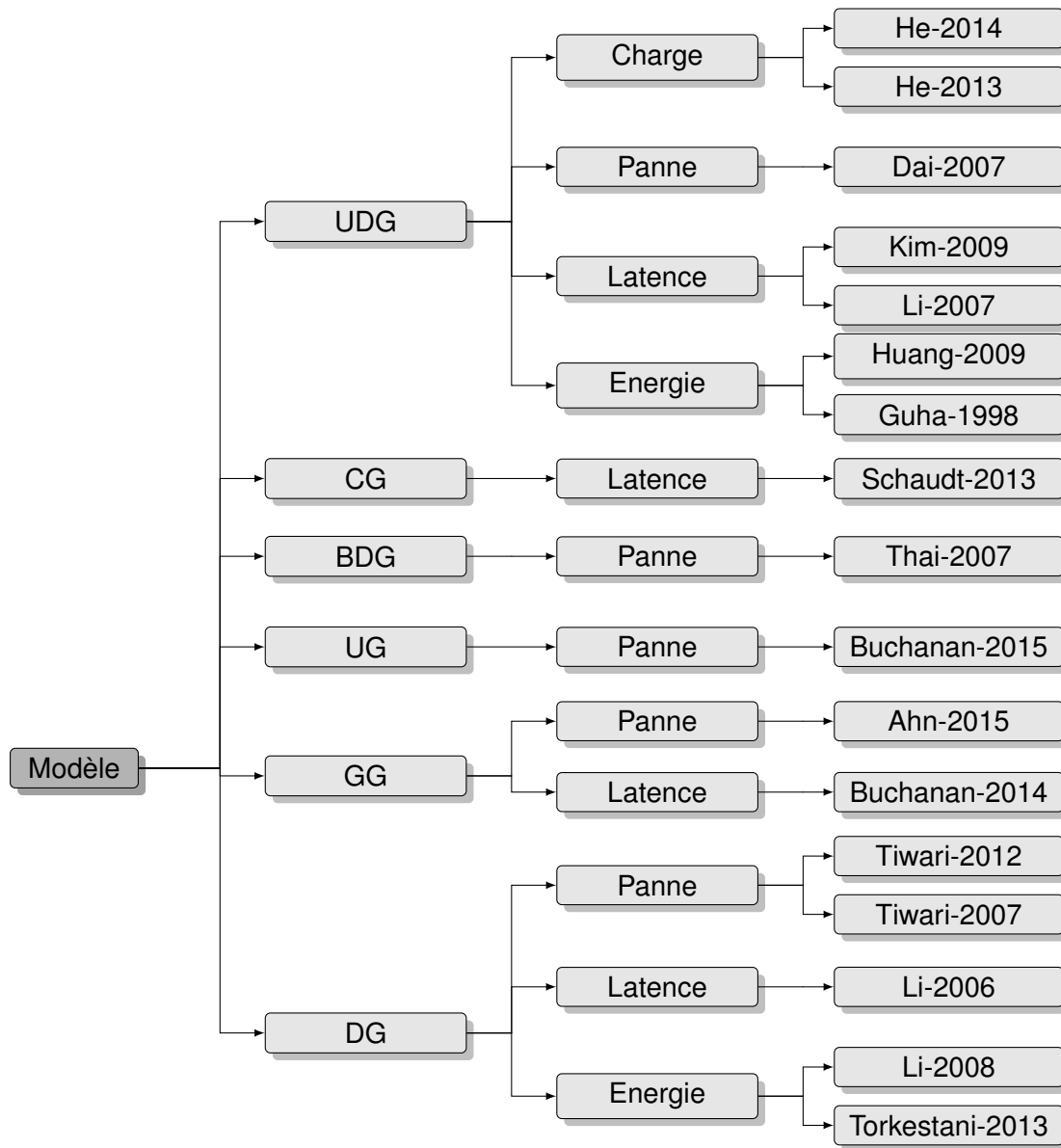


FIGURE 2.6 – Résumé des travaux proposant la construction d'une dorsale



CONTRIBUTIONS

ALGORITHME D'APPROXIMATION POUR LA DIFFUSION DANS LES RÉSEAUX DE CAPTEURS

Sommaire

3.1	Caractérisation du problème	57
3.1.1	Complexité	58
3.1.2	Programme linéaire en nombres entiers	59
3.2	Algorithme d'approximation BB	62
3.3	Étude préliminaire	63
3.3.1	Structure du chemin P_1 par rapport à P_2	64
3.3.2	Preuve du théorème 1	65
3.4	Algorithme Bounded Backbone (BB)	65
3.5	Preuve	67
3.6	Synthèse	70

Dans ce chapitre, nous nous intéressons à la diffusion dans les réseaux de capteurs hétérogènes. Pour rappel, nous nous focalisons sur le cas où la station de base doit envoyer un message à tous les nœuds du réseau de capteurs. Ce chapitre débute en effectuant la caractérisation du problème traité et se termine par la présentation d'un algorithme d'approximation.

3.1/ CARACTÉRISATION DU PROBLÈME

Le réseau considéré est hétérogène au niveau des puissances de transmission affectées aux capteurs. Le coût énergétique W , nécessaire pour effectuer une transmission, dépend de la puissance du nœud. Dans notre cas, nous considérons que W appartient à l'intervalle $[W_{\min}; W_{\max}]$ où $W_{\max} = \Omega * W_{\min}$. Pour effectuer la diffusion, nous utilisons une approche par dorsale. Une dorsale apporte de nombreux avantages : elle supprime les liaisons de transmission inutiles, réduit les entêtes de communication, réduit la redondance et, en diminuant la consommation d'énergie, augmente la durée de vie du réseau. Les caractéristiques recherchées dans la construction d'une dorsale dépendent du problème à résoudre. Dans notre cas, l'objectif est d'optimiser deux métriques en même

temps : diminuer le coût énergétique lors de la diffusion du message et diminuer le délai de transmission (nommé aussi latence). Le délai de transmission étant proportionnel au nombre de sauts, nous calculons un ensemble dominant connexe avec un poids minimal et dont la distance en sauts depuis la station de base est bornée par une constante $h_{\max}$. De cette manière, en bornant le nombre de sauts depuis la station de base, nous bornons le délai de transmission, c'est à dire la latence. Nous remarquons que $h_{\max}$ doit avoir une valeur supérieure ou égale à l'excentricité ε pour atteindre le nœud avec le chemin le plus long depuis la station de base. De plus, l'ensemble dominant connexe construit doit contenir la station de base car nous considérons le cas où la station de base effectue la diffusion. Nous appelons ce problème l'ensemble dominant connexe de poids minimal et de distance en nombre de sauts bornée.

Définition 45 : Ensemble dominant connexe de poids minimal et de distance en nombre de sauts bornée (MWCDs-BHD)

Soit $G = (V, E, W)$ un graphe. Un ensemble $S \subseteq V$ est un ensemble dominant connexe de poids minimal et de distance en nombre de sauts bornée (*Minimum Weighted Connected Dominating Set with Bounded Hop Distance*) si S est un ensemble dominant connexe de poids minimal et si la distance en nombre de saut depuis la station de base est bornée par une constante $h_{\max}$. [86].

Comme nous le verrons dans la suite, ce problème est NP-Difficile. Par conséquent, l'utilisation d'un algorithme exact pour les grandes instances n'est pas concevable en raison du temps de calcul requis pour obtenir une solution. Toutefois, cela nous permet d'obtenir la borne inférieure du coût d'une dorsale et de pouvoir, par la suite, effectuer une comparaison avec les solutions proposées dans le chapitre 4 pour des petites instances.

Dans cette partie, nous proposons la résolution de la construction d'une dorsale pour la diffusion d'énergie minimale et ayant un délai de transmission bornée via une méthode donnant une solution exacte : un programme linéaire en nombres entiers. Mais d'abord, nous étudions la complexité en temps du problème traité.

3.1.1/ COMPLEXITÉ

Pour déterminer la complexité du problème de l'ensemble dominant connexe de poids minimal et de distance en nombre de sauts bornée, nous allons nous baser sur la complexité d'un problème proche du nôtre et en déduire sa complexité. Pour cela, nous considérons les travaux de Li et al [18]. Ils prouvent que la construction d'un arbre de diffusion minimisant le coût énergétique total est NP-Difficile.

Supposons que nous disposions d'un algorithme A_h qui résout notre problème de l'ensemble dominant connexe de poids minimal et de distance en nombre de sauts h bornée. Nous pouvons alors résoudre le problème de l'ensemble dominant de poids minimal en faisant appel à A_n , c'est-à-dire en fixant h au maximum possible dans un graphe : le nombre de nœuds n . Cette réduction étant clairement polynomiale, ceci permet de montrer que notre problème est NP-Difficile.

3.1.2/ PROGRAMME LINÉAIRE EN NOMBRES ENTIERS

Dans cette section, nous présentons un programme linéaire en nombres entiers (PLNE) permettant la modélisation du problème MWCDs-BHD. Ce programme construit une arborescence telle que le coût énergétique des nœuds non-feuilles est minimisé et leurs profondeurs dans l'arborescence est bornée par $h_{\max} - 1$ en nombre d'arcs.

Définition 46 : Profondeur d'un nœud

La profondeur d'un nœud ou d'une feuille dans une arborescence est le nombre d'arêtes qu'il faut parcourir pour atteindre ce nœud à partir de la racine de l'arbre. Cela correspond à la longueur du chemin depuis la racine vers ce nœud.

Afin d'effectuer la formulation du programme linéaire en nombres entiers, nous devons, dans un premier temps, définir un certain nombre de variables de décisions.

La première est une variable de décision binaire x_i indiquant si le nœud i appartient ($x_i = 1$) ou pas ($x_i = 0$) à la dorsale. Afin d'assurer que la dorsale construite est connectée, nous construisons une arborescence de racine le nœud 0 ou station de base.

Les nœuds, appartenant à l'arborescence obtenue par le programme linéaire et qui ne sont pas des feuilles, constituent la dorsale.

La variable de décision $y_{i,j}$, $(i, j) \in E$ exprime si l'arête (i, j) appartient à l'arborescence constituée des nœuds de la dorsale. En d'autres termes, si $y_{i,j} = 1$, alors les nœuds i et j font partie de la dorsale et, de plus, le nœud i est le père du nœud j dans l'arborescence obtenue.

Pour finir, la variable de décision entière u_i , $i \in V$ garantit l'absence de cycle dans la dorsale.

La fonction objectif de notre programme linéaire vise à minimiser le coût énergétique total W de la dorsale construite. Mathématiquement, cela se traduit comme suit : $W = \sum_{i \in V} x_i W_i$. Les contraintes du programme linéaire sont les suivantes :

$$\text{Minimiser} \quad W = \sum_{i \in V} x_i W_i \quad (3.1)$$

Sous les contraintes

$$\forall i \in V^*, \quad \sum_{\{j \mid j \in N^-(i)\}} x_j \geq 1 \quad (3.2)$$

$$\forall i \in V^*, \quad \sum_{\{j \mid i \in N^+(j)\}} y_{j,i} = x_i \quad (3.3)$$

$$\forall (i, j) \in E, \quad x_i + x_j - 2y_{i,j} \geq 0 \quad (3.4)$$

$$\forall (i, j) \in E, \quad (n+1)y_{i,j} + u_i - u_j + (n-1)y_{j,i} \leq n \quad (3.5)$$

$$\forall i \in V, \quad u_i \leq h_{\max} \quad (3.6)$$

$$\sum_{i \in V} x_i - \sum_{(i,j) \in E} y_{i,j} = 1 \quad (3.7)$$

$$x_0 = 1 \quad (3.8)$$

$$u_0 = 1 \quad (3.9)$$

L'équation (3.2) permet d'assurer la propriété de dominance : chaque nœud i de V^* appartient au voisinage d'au moins un nœud appartenant au backbone.

L'équation (3.3) spécifie que les nœuds du backbone ont exactement un père dans l'arbre.

L'équation (3.4) assure que si une arête appartient à la dorsale, alors ses deux extrémités font aussi partie de la dorsale.

L'équation (3.5) s'inspire de ce qui est proposé par [1] où elle est utilisée pour éviter la présence des cycles dans le problème du voyageur de commerce. Cette équation est aussi utilisée dans [58] pour le même objectif mais dans le cas d'arbre couvrant. L'équation permet d'affecter un nombre u_i strictement positif à chaque nœud de la dorsale. Ce nombre représente la distance (plus un) dans l'arbre depuis la station de base. Si le nœud i et j sont dans la dorsale et que $y_{i,j} = 1$, alors cela assure que $u_j = u_i + 1$. Si un des nœuds i ou j n'appartient pas à la dorsale, l'équation garantie que la différence entre u_i et u_j est bornée par la différence maximale possible.

L'équation (3.6) assure que la distance en termes de nombre de sauts est bornée par $h_{\max}$. La variable u_i modélise la profondeur du nœud i dans l'arbre représentant la dorsale. Borner la profondeur de chaque nœud de la dorsale garantie que la hauteur de l'arbre est également bornée.

L'équation (3.7) stipule que le nombre de nœuds de la dorsale est égal au nombre d'arêtes appartenant à la dorsale plus un. Cela constitue une caractéristique propre aux arbres.

Les deux dernières équations (3.8) et (3.9) permettent l'affectation des valeurs de la station de base.

Ce programme linéaire peut être utilisé pour trouver une solution optimale. Malheureusement, le temps nécessaire pour le résoudre est considérable, même pour des réseaux de taille moyenne car il s'agit d'un problème NP-Difficile comme vu dans la section 3.1.1.

Le programme linéaire proposé est constitué de $2|V| + |E|$ variables et $3|V| + 2|E| + 3$ équations.

Exemple de résultat et d'encodage obtenu par le PLNE

Dans cette partie, nous présentons un exemple afin d'illustrer l'instanciation du programme linéaire défini précédemment. Dans l'exemple, nous considérons un réseau composé de cinq nœuds où le nœud 0 représente la station de base. La figure 3.1a montre le graphe représentant le réseau de capteurs considéré.

L'ensemble de nœuds V est composé des nœuds 0, 1, 2, 3, 4. L'ensemble des arêtes E est égal à l'ensemble suivant : $\{(0, 1), (1, 0), (1, 4), (4, 1), (1, 3), (3, 1), (3, 2), (3, 4)\}$. Les poids des nœuds sont : $W_0 = 5.77$, $W_1 = 10.17$, $W_2 = 7.53$, $W_3 = 6.30$, $W_4 = 7.2$. Nous considérons que $h_{\max}$ a pour valeur 3. Dans la suite, nous donnerons les contraintes obtenues pour le programme linéaire pour l'exemple considéré.

En appliquant l'inégalité (3.2) à notre exemple, nous obtenons les contraintes suivantes :

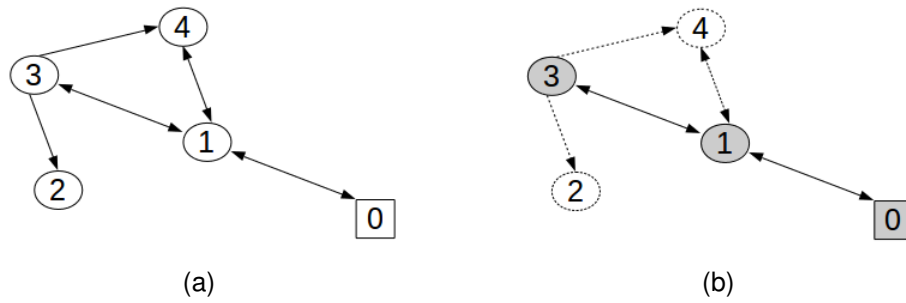


FIGURE 3.1 – Réseau de capteur considéré dans l'exemple (à gauche) et exemple de dorsale construite par le PLNE (à droite)

$$\text{Nœud 1 : } x_0 + x_4 + x_3 \geq 1$$

$$\text{Nœud 3 : } x_1 \geq 1$$

$$\text{Nœud 2 : } x_3 \geq 1$$

$$\text{Nœud 4 : } x_1 + x_3 \geq 1$$

La contrainte (3.3) donne :

$$\text{Nœud 1 : } y_{0,1} + y_{4,1} + y_{3,1} = x_1$$

$$\text{Nœud 3 : } y_{1,3} = x_3$$

$$\text{Nœud 2 : } y_{3,2} = x_2$$

$$\text{Nœud 4 : } y_{3,4} + y_{1,4} = x_4$$

Dans le cas de la contrainte (3.4), nous obtenons :

$$\text{Arête (0, 1) : } x_0 + x_1 - 2 \times y_{0,1} \geq 0$$

$$\text{Arête (1, 4) : } x_1 + x_4 - 2 \times y_{1,4} \geq 0$$

$$\text{Arête (1, 3) : } x_1 + x_3 - 2 \times y_{1,3} \geq 0$$

$$\text{Arête (3, 2) : } x_3 + x_2 - 2 \times y_{3,2} \geq 0$$

$$\text{Arête (1, 0) : } x_1 + x_0 - 2 \times y_{1,0} \geq 0$$

$$\text{Arête (4, 1) : } x_4 + x_1 - 2 \times y_{4,1} \geq 0$$

$$\text{Arête (3, 1) : } x_3 + x_1 - 2 \times y_{3,1} \geq 0$$

$$\text{Arête (3, 4) : } x_3 + x_4 - 2 \times y_{3,4} \geq 0$$

L'instanciation de la containte (3.5) donne :

$$\text{Arête (0, 1) : } 6 \times y_{0,1} + u_0 - u_1 + 4 \times y_{1,0} \leq 5$$

$$\text{Arête (1, 0) : } 6 \times y_{1,0} + u_1 - u_0 + 4 \times y_{0,1} \leq 5$$

$$\text{Arête (1, 4) : } 6 \times y_{1,4} + u_1 - u_4 + 4 \times y_{4,1} \leq 5$$

$$\text{Arête (4, 1) : } 6 \times y_{4,1} + u_4 - u_1 + 4 \times y_{1,4} \leq 5$$

$$\text{Arête (1, 3) : } 6 \times y_{1,3} + u_1 - u_3 + 4 \times y_{3,1} \leq 5$$

$$\text{Arête (3, 1) : } 6 \times y_{3,1} + u_3 - u_1 + 4 \times y_{1,3} \leq 5$$

$$\text{Arête (3, 2) : } 6 \times y_{3,2} + u_3 - u_2 + 4 \times y_{2,3} \leq 5$$

$$\text{Arête (3, 4) : } 6 \times y_{3,4} + u_3 - u_4 + 4 \times y_{4,3} \leq 5$$

En appliquant l'inégalité (3.6) à notre exemple, nous obtenons les inégalités suivantes :

$$\text{Nœud 0 : } u_0 \leq h_{\max}$$

$$\text{Nœud 3 : } u_3 \leq h_{\max}$$

$$\text{Nœud 1 : } u_1 \leq h_{\max}$$

$$\text{Nœud 4 : } u_4 \leq h_{\max}$$

$$\text{Nœud 2 : } u_2 \leq h_{\max}$$

L'équivalence (3.7) donne :

$$x_0 + x_1 + x_2 + x_3 + x_4 - y_{0,1} - y_{1,0} - y_{1,4} - y_{4,1} - y_{1,3} - y_{3,1} - y_{3,2} - y_{3,4} = 1$$

Pour finir, la fonction objectif à minimiser est la suivante :

$$x_0 W_0 + x_1 W_1 + x_2 W_2 + x_3 W_3 + x_4 W_4$$

Variable	u_0	u_1	u_2	u_3	u_4	x_0	x_1	x_2	x_3
Valeur	1	2	0	3	0	1	1	0	1
Variable	x_4	$y_{0,1}$	$y_{1,0}$	$y_{1,3}$	$y_{3,1}$	$y_{3,2}$	$y_{3,4}$	$y_{1,4}$	$y_{4,1}$
Valeur	0	1	0	1	0	0	0	0	0

TABLE 3.1 – Exemple de programme linéaire obtenu

En réunissant toutes les contraintes obtenues, nous pouvons déduire les valeurs de nos variables de décisions comme le montre le tableau 3.1. De plus, la fonction objectif a pour valeur 22.24.

Le résultat obtenu par la résolution du PLNE en utilisant GLPK est illustré dans la figure 3.1b. Les nœuds en gris constituent l'ensemble des nœuds de la dorsale construite. Les nœuds en blanc représentent les feuilles de l'arborescence obtenue par la résolution du programme linéaire : il s'agit des nœuds qui n'effectuerons pas d'envoi du message et, par conséquent, qui ne consommerons pas d'énergie pour la transmission. Nous verrons dans le chapitre 4 section 4.4 la taille maximale du réseaux que nous pouvons traiter avec le PLNE.

3.2/ ALGORITHME D'APPROXIMATION BB

Dans cette section, nous proposons un algorithme d'approximation permettant d'obtenir une solution approchée et garantie du problème du MWCDs-BHD.

Principe

Étant donné un graphe orienté et pondéré, l'algorithme calcule, dans un premier temps, deux arbres couvrants $\mathcal{A}_1$ et $\mathcal{A}_2$.

L'arbre couvrant $\mathcal{A}_1$ est obtenu en calculant les plus courts chemins depuis la station de base pour tout nœud du graphe. Dans ce cas, on considère que la pondération des nœuds est unitaire. De cette manière, les chemins présents dans $\mathcal{A}_1$ permettent de minimiser le nombre de sauts entre la station de base et tout autre nœud.

L'arbre couvrant $\mathcal{A}_2$ est construit en calculant les plus courts chemins pondérés depuis la station de base. Dans ce cas, la pondération associée à chaque nœud i correspond à l'énergie W_i consommée pour la transmission d'un message par le nœud i .

Afin de prendre en compte l'énergie consommée par un nœud donné, nous avons étiqueté toutes ces arêtes sortantes par leur poids. Les chemins de $\mathcal{A}_2$ minimisent le coût énergétique nécessaire pour atteindre le nœud cible.

Le principe de l'algorithme est de choisir itérativement un nœud k et, ensuite, analyser $P_1(k)$ et $P_2(k)$, respectivement le chemin entre la station de base et le nœud k dans l'arbre couvrant $\mathcal{A}_1$ et $\mathcal{A}_2$. Comme $h_{\max}$ est supérieure ou égale à ε , la longueur de $P_1(k)$ est inférieure ou égale à $h_{\max}$. Puisque $P_2(k)$ est le chemin minimisant le coût énergétique, pour connecter le nœud k à la station de base, nous considérons dans un premier temps le chemin $P_2(k)$. Si sa longueur est inférieure ou égale à $h_{\max}$ alors le chemin $P_2(k)$ est retenu. Dans le cas contraire, c'est le chemin $P_1(k)$ qui est choisi. Ce processus assure que tous les chemins retenus auront une longueur inférieure ou égale à $h_{\max}$. Cependant, la question se pose de savoir quel est le coût énergétique qui est ajouté lorsque le chemin choisi est P_1 au lieu de P_2 .

Algorithm 1 Algorithme Bounded Backbone

Input : Graphe G (WSN), Distance maximale en nombre de sauts $h_{\max}$

Output : Dorsale B

```

 $B \leftarrow \emptyset$ 
 $U \leftarrow V$ 
 $D \leftarrow \emptyset$ 
while  $U \neq \emptyset$  do
   $k \leftarrow$  nœud le plus distant de la station de base dans  $\mathcal{A}_1$  non encore couvert
  if  $L(P_2(k)) \leq h_{\max}$  then
     $P \leftarrow P_2(k)$ 
  else
     $P \leftarrow P_1(k)$ 
  end if
   $B \leftarrow B \cup \{x \mid x \in P \wedge x \neq k\}$ 
   $U \leftarrow U \setminus \{x \mid x \in P\}$ 
   $D \leftarrow D \cup \bigcup_{\substack{x \in P \\ x \neq k}} N^+(x)$ 
  for  $x \in P \wedge x \neq k$  do
    //Phase de mise à zéro de l'énergie
     $W_x \leftarrow 0$ 
  end for
  Calculer  $\mathcal{A}_2$ 
end while
return  $B$ 

```

3.3/ ÉTUDE PRÉLIMINAIRE

Dans cette section, nous effectuons une étude permettant de borner le coût énergétique supplémentaire lorsque le chemin choisi n'est pas celui minimisant l'énergie mais celui minimisant la latence. Pour cette preuve, nous considérons un cas particulier dans lequel les nœuds ne peuvent avoir que deux énergies possibles : $W_{\min}$ et $W_{\max}$ avec $W_{\max} = \Omega W_{\min}$.

Théorème 1 : Coût du chemin borné

Le chemin P choisi par l'algorithme Bounded Backbone a un coût borné :

$$w(P) \leq \Omega w(P_2)$$

Pour ce faire, nous allons d'abord étudier la structure du chemin P_1 par rapport au chemin P_2 .

3.3.1/ STRUCTURE DU CHEMIN P_1 PAR RAPPORT À P_2

Dans cette section, nous prouvons le lemma 1 afin de prouver le théorème 1.

Lemma 1 : Structure du chemin P_1 par rapport à P_2

Supposons que P_2 est constitué de u nœuds d'énergie $W_{\min}$ et v nœuds d'énergie $W_{\max}$, alors P_1 est constitué de $u - \Delta u$ nœuds d'énergie $W_{\min}$ et $v + \Delta v$ nœuds d'énergie $W_{\max}$ et

$$0 \leq \Delta v \leq \Delta u \leq \left(\frac{W_{\max}}{W_{\min}} \right) \Delta v$$

Démonstration. Comme P_1 est le plus court chemin en termes de nombre de sauts, nous savons que :

$$u - \Delta u + v + \Delta v \leq u + v$$

Par conséquent, nous en déduisons que :

$$\Delta v \leq \Delta u \tag{3.10}$$

Sachant que P_2 minimise l'énergie, on déduit ce qui suit :

$$(u - \Delta u) * W_{\min} + (v + \Delta v) * W_{\max} \geq u * W_{\min} + v * W_{\max}$$

Ainsi, nous obtenons :

$$\Delta u \leq \left(\frac{W_{\max}}{W_{\min}} \right) \Delta v \tag{3.11}$$

Quatre cas peuvent se produire :

1. $\Delta u < 0$ et $\Delta v \geq 0$. Impossible à cause de l'équation 3.10. En effet, cela impliquerait que Δu est une variable positive.
2. $\Delta u \geq 0$ et $\Delta v \leq 0$. Ce cas est impossible à cause de l'équation 3.11, car cela signifierait que Δu serait une variable négative.
3. $\Delta u < 0$ et $\Delta v \leq 0$. Donc nous avons $-\Delta v \leq -\left(\frac{W_{\min}}{W_{\max}}\right) \Delta u < -\Delta u$. Ceci est en contradiction avec l'équation 3.10.
4. $\Delta u \geq 0$ et $\Delta v \geq 0$. Ce cas est le seul possible et il implique l'inégalité suivante :

$$0 \leq \Delta v \leq \Delta u \leq \left(\frac{W_{\max}}{W_{\min}} \right) \Delta v = \Omega \Delta v \tag{3.12}$$

□

3.3.2/ PREUVE DU THÉORÈME 1

Avec les résultats obtenus précédemment, nous pouvons compléter la preuve du théorème 1.

Démonstration. Le coût énergétique du chemin P_1 peut s'écrire comme suit :

$$\begin{aligned} w(P_1) &= (u - \Delta u)W_{\min} + (v + \Delta v)W_{\max} \\ &= uW_{\min} - \Delta uW_{\min} + vW_{\max} + \Delta vW_{\max} \end{aligned}$$

D'après l'équation 3.12, nous savons que $\Delta v \leq \Delta u$. Ce qui donne :

$$w(P_1) \leq uW_{\min} - \Delta uW_{\min} + vW_{\max} + \Delta uW_{\max}$$

En remplaçant $W_{\max}$ par $\Omega * W_{\min}$, nous obtenons :

$$\begin{aligned} w(P_1) &\leq uW_{\min} - \Delta uW_{\min} + v\Omega W_{\min} + \Delta u\Omega W_{\min} \\ &= W_{\min}(u - \Delta u + v\Omega + \Delta u\Omega) \\ &= W_{\min}(u + v\Omega + \Delta u(\Omega - 1)) \end{aligned}$$

Puisque $u - \Delta u \geq 0$, nous pouvons maximiser Δu par u :

$$\begin{aligned} w(P_1) &\leq W_{\min}(u + v\Omega + u(\Omega - 1)) \\ &= W_{\min}(u + v\Omega - u + u\Omega) \\ &= W_{\min}(v\Omega + u\Omega) \end{aligned}$$

Comme $W_{\max} = \Omega W_{\min}$, nous obtenons :

$$\begin{aligned} w(P_1) &\leq v(W_{\max}) + u(\Omega W_{\min}) \\ &\leq \Omega(vW_{\max} + uW_{\min}) \\ &= \Omega w(P_2) \end{aligned}$$

Ce qui nous permet de conclure :

$$w(P_2) \leq w(P_1) \leq \Omega w(P_2).$$

Comme le chemin P est égal soit à P_1 soit à P_2 , on peut déduire que son coût énergétique est égal soit à $w(P_1)$ ou $w(P_2)$. Donc $w(P) \leq \Omega w(P_2)$.

□

3.4/ ALGORITHME BOUNDED BACKBONE (BB)

Dans cette section, nous allons expliquer en détails le fonctionnement de l'algorithme d'approximation décrit dans l'algorithme 1. Pour décrire l'algorithme *BB*, nous introduisons la définition d'un certain nombre d'ensembles de nœuds. L'ensemble B représente l'ensemble des nœuds faisant partie de la dorsale. L'ensemble U constitue l'ensemble des nœuds non encore traités. Initialement, U est équivalent à V . D est l'ensemble

des nœuds dominés. Au début de chaque itération, l'équation suivante est toujours vraie : $V = B \cup D \cup U$. De plus $B \cap U = \emptyset$ et $D \cap U = \emptyset$.

Comme nous pouvons le voir, au début, l'algorithme initialise les ensembles B , U et D . L'idée principale est de sélectionner itérativement un nœud k non encore couvert par la diffusion i.e. appartenant à l'ensemble U . Le nœud k choisi par l'algorithme BB constitue le nœud le plus distant non encore couvert dans l'arbre couvrant $\mathcal{A}_1$ i.e. la distance est considérée en termes de nombre de sauts. Une fois le nœud k choisi, l'étape suivante est de choisir le chemin pour l'atteindre entre le chemin $P_1(k)$ et $P_2(k)$ comme décrit précédemment.

Ensuite, les nœuds appartenant au chemin retenu sont ajoutés à la dorsale et enlevés de l'ensemble U . Pour finir, l'ensemble des nœuds dominés est mis à jour. À la fin de chaque itération, le coût énergétique des nœuds ajoutés à B est mis à zéro. Ensuite, puisque les plus courts chemins pondérés changent suite à cette mise à jour, l'arbre couvrant $\mathcal{A}_2$ est recalculé. Le processus expliqué ci-dessus est répété jusqu'à ce que l'ensemble U soit vide. La solution est donnée par l'ensemble B obtenu par ce processus.

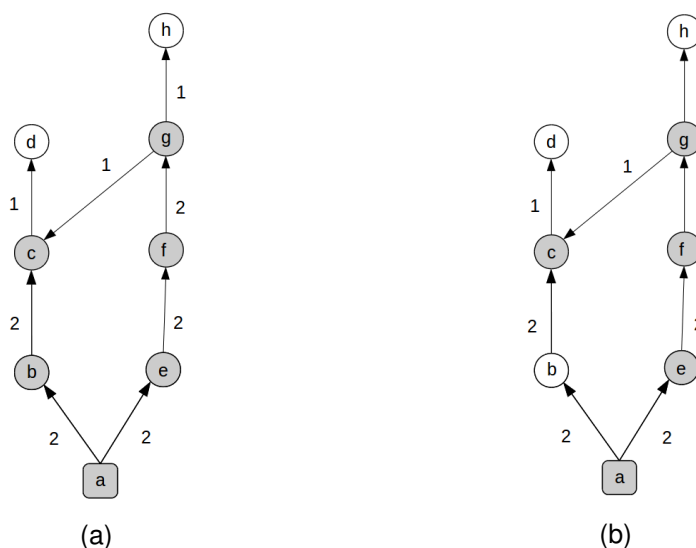


FIGURE 3.2 – Explication de la mise à zéro de l'énergie dans l'algorithme Bounded Backbone. Dorsale obtenue sans la mise à zéro de l'énergie (à gauche) et dorsale obtenue avec mise à zéro de l'énergie (à droite).

La mise à zéro de l'énergie des nœuds ajoutés à la dorsale nous permet de choisir en priorité les chemins contenant des nœuds appartenant déjà à la dorsale. En effet, puisqu'ils sont déjà dans celle-ci, ils ne vont pas augmenter son coût énergétique. Pour mieux expliquer la mise à zéro de l'énergie, considérons l'exemple représenté dans la figure 3.2.

Dans cette figure, nous supposons que $h_{\max} = 5$. La dorsale de la figure 3.2a représente la dorsale obtenue sans la phase de mise à zéro des énergies. La dorsale est obtenue de la manière suivante. À la première itération, BB choisit comme nœud à traiter le nœud h . Le chemin choisi est $P = (a, e, f, g, h)$. Par conséquent, les nœuds a , e , f et g sont ajoutés à l'ensemble B et sont supprimés de l'ensemble U . Les nœuds h , c et b sont, quant à eux, ajoutés à l'ensemble D .

Lors de la deuxième itération, le nœud à traiter est d . Le chemin choisi est $P = (a, b, c, d)$. Les nœuds b et c sont ajoutés à l'ensemble B constituant la dorsale et supprimés de U . Pour finir, d est ajouté à l'ensemble D . L'algorithme se termine car l'ensemble U est vide. À la fin, la dorsale est constituée des nœuds a, b, c, e, f , et g et son coût énergétique est de 10.

La dorsale représentée dans la figure 3.2b est moins coûteuse en termes énergétique. En effet, elle se compose des nœuds a, c, e, f , et g et a un coût énergétique de 8. Au niveau de la deuxième itération, la mise à zéro des énergies des nœuds appartenant à la dorsale nous permet de considérer le chemin $P = (a, e, f, g, c, d)$ et de le choisir car son coût énergétique est inférieur à celui du chemin $P = (a, b, c, d)$. En effet, $W_a = W_e = W_f = W_g = 0$. Avec cette méthode, nous obtenons la dorsale de la figure 3.2b. De plus, en mettant à zéro l'énergie, l'algorithme Bounded Backbone choisit les chemins qui minimisent le coût énergétique à condition que ceux-ci respectent la contrainte de longueur maximale donnée par $h_{\max}$.

La complexité de l'algorithme Bounded Backbone dépend de l'algorithme choisi pour le calcul des arbres couvrants $\mathcal{A}_1$ et $\mathcal{A}_2$. Dans notre cas, nous avons choisi d'utiliser l'algorithme de Dijkstra. Ce dernier est utilisé dans la boucle principale pour recalculer $\mathcal{A}_2$. Par conséquent, la complexité de l'algorithme Bounded Backbone est $O(|V||E| + |V|^2 \log_2 |V|)$ ce qui constitue une complexité polynomiale.

3.5/ PREUVE

Dans cette section, une étude théorique de l'algorithme Bounded Backbone est effectuée. Cette section se divise en deux parties : la première vise à montrer la preuve de la correction de l'algorithme Bounded Backbone et la deuxième à effectuer la preuve de sa borne d'approximation.

La correction de l'algorithme Bounded Backbone s'effectue via les Lemme 2 et 3.

Lemma 2 : Ensemble dominant connexe

L'ensemble B construit par l'algorithme Bounded Backbone est un ensemble dominant connexe.

Démonstration. L'ensemble B est un ensemble dominant connexe si, et seulement si, B est un ensemble dominant et que le graphe induit par B est connecté. Par construction, l'ensemble B est un ensemble connecté. En effet, l'ensemble B est construit par ajouts itératifs de chemins qui sont tous connectés à la station de base.

Pour prouver que B est dominant, nous procédons par l'absurde. Supposons que B n'est pas dominant. Cela implique qu'il existe un nœud $v \in V \setminus B$ tel que pour tout $u \in B$, (u, v) n'est pas dans E . Il y a deux cas possibles : soit v est dans D ou v est dans U .

- Cas où $v \in D$. Par construction de l'ensemble D , nous savons qu'il existe un nœud $d \in B$ tel que $v \in N^+(d)$, donc $(d, v) \in E$. Ceci est en contradiction avec notre hypothèse.
- Cas où $v \in U$. Puisque $v \in U$ alors $U \neq \emptyset$. Ceci est impossible car dans ce cas la condition d'arrêt de la boucle n'est pas respectée.

Nous pouvons conclure que l'ensemble B est dominant. Par conséquent, l'ensemble B est bien un ensemble dominant connexe. $\square$

Lemma 3 : Profondeur bornée

Dans l'ensemble B , la distance, exprimée en nombre de sauts, entre la station de base et tout autre nœud du graphe est au plus de $h_{\max}$.

Démonstration. Soit u un nœud dans V . Il y a deux possibilités :

- Le nœud u est l'un des nœuds k choisis par l'algorithme BB . Pour connecter u à la station de base, l'algorithme BB choisit un chemin entre $P_1(u)$ et $P_2(u)$. Si le chemin choisi est $P_1(u)$ alors nous pouvons affirmer que $L(P_1(u)) \leq \varepsilon$. Par hypothèse, nous savons que $\varepsilon \leq h_{\max}$, par conséquent $L(P_1(u)) \leq h_{\max}$. Si le chemin choisi est $P_2(u)$, alors nous savons que $L(P_2(u)) \leq h_{\max}$ autrement ce chemin n'aurait pas pu être retenu.
- Le nœud u n'est pas l'un des nœuds k sélectionnés par l'algorithme BB . Dans ce cas, nous avons deux possibilités :
 - $\square$ Le nœud u est un nœud interne d'un chemin choisi pour atteindre un nœud k . Supposons que u est un nœud interne au chemin choisi pour relier k_i à la station de base. Nous savons que $L(P(k_i)) \leq h_{\max}$, par conséquent $L(P(u)) \leq L(P(k_i)) \leq h_{\max}$.
 - $\square$ Le nœud u est une feuille de l'arbre obtenu qui n'a pas été choisi par l'algorithme BB comme nœud k à traiter. Puisqu'il n'a pas été choisi, nous pouvons en déduire que tous les nœuds nécessaires pour le connecter à la station de base étaient déjà présents dans la dorsale. De plus, u est dans $N^+(\{x \mid x \in B\})$. Comme démontré précédemment, pour tout nœud k choisi par l'algorithme BB , la longueur du chemin retenu pour l'atteindre est d'au plus $h_{\max}$. Donc, tout nœud x appartenant à B a une distance en nombre de sauts depuis la station de base d'au plus $h_{\max} - 1$. Supposons que y soit le père de u . Alors la distance en nombre de sauts de u est égale à la distance en nombre de sauts de y plus un. Nous en déduisons que y est à $h_{\max}$ nombre de sauts à partir de la station de base.

$\square$

Maintenant, nous pouvons prouver le Théorème 2 en considérant que les nœuds peuvent avoir une énergie appartenant à l'intervalle $[W_{\min}; W_{\max}]$.

Théorème 2 : Borne d'approximation

La dorsale obtenue par l'algorithme BB a une borne d'approximation égale à $\Omega h_{\max}$.

Démonstration. Supposons que L^* est l'ensemble des feuilles dans un arbre optimal. Nous assignons à chaque nœud l^* de L^* les nœuds appartenant au chemin entre l^* et la station de base en commençant par le nœud le plus distant à chaque fois que nous traitons un nouvel ensemble. Si un nœud appartient à plusieurs chemins, nous

l'assignons au premier nœud k dans lequel il apparaît. Ce processus construit l'ensemble $S^*(l^*)$.

Ensuite, nous assignons à chaque nœud d'un ensemble la charge suivante :

$$\frac{w^*(S^*(l^*))}{|S^*(l^*)|}$$

Ainsi, la somme des charges de tous les nœuds d'un ensemble l^* donne son énergie totale optimale notée $w^*(S^*(l^*))$. En effet, nous avons :

$$\begin{aligned} \sum_{k \in S^*(l^*)} w^*(k) &= \sum_{k \in S^*(l^*)} \frac{w^*(S^*(l^*))}{|S^*(l^*)|} \\ &= |S^*(l^*)| \times \frac{w^*(S^*(l^*))}{|S^*(l^*)|} \\ &= w^*(S^*(l^*)) \end{aligned}$$

Grâce à cette manière de construire ces ensembles, la somme de la charge de tous les nœuds des ensembles construits donne le coût de l'arbre optimal. Pour calculer la borne d'approximation, nous devons calculer la charge que chaque nœud d'un ensemble construit peut prendre dans l'arbre construit par BB . Puis, comme auparavant, en additionnant toutes les charges de tous les ensembles, nous obtenons le coût de l'arbre construit par BB . On note $w(S^*(l^*))$ le coût de l'ensemble $S^*(l^*)$ obtenu en ajoutant la charge que les nœuds dans $S^*(l^*)$ peuvent prendre dans l'arbre construit par BB .

Nous savons que la plus grande charge qu'un nœud peut prendre est $W_{\max}$:

$$\forall i \in V, w_i \leq W_{\max}$$

De plus, par construction, le nombre de nœuds dans un chemin est au plus $h_{\max}$.

Par conséquent, nous pouvons en déduire que le coût de $w(S^*(l^*))$ est d'au plus :

$$w(S^*(l^*)) \leq W_{\max} h_{\max}$$

Nous en concluons que le rapport entre l'énergie W de notre algorithme et de l'énergie optimale W^* est :

$$\begin{aligned} \frac{W}{W^*} &= \frac{\sum_{l^* \in L^*} w(S^*(l^*))}{\sum_{l^* \in L^*} w^*(S^*(l^*))} \\ &\leq \frac{|L^*| (W_{\max} h_{\max})}{|L^*| W_{\min}} \\ &\leq \Omega h_{\max} \end{aligned}$$

En conclusion, l'algorithme a un rapport d'approximation de $\Omega h_{\max}$.

□

3.6/ SYNTHÈSE

Dans ce chapitre, nous avons abordé l'optimisation de l'énergie et la latence dans les réseaux de capteurs sans fils. En particulier, nous avons considéré le cas où le réseau est composé de capteurs caractérisés par des portées différentes. Afin d'optimiser la diffusion, nous avons proposé deux approches par dorsale. En particulier, nous avons proposé un programme linéaire en nombres entiers ainsi qu'un algorithme d'approximation. Des études théoriques ont été présentées afin de prouver l'exactitude de l'algorithme proposé ainsi que sa borne d'approximation.

Dans le chapitre suivant, nous proposons de traiter le même problème par le biais d'une nouvelle heuristique et du recuit simulé. De plus, nous effectuerons les simulations afin de comparer les performances des méthodes proposées.

HEURISTIQUES POUR LA DIFFUSION DANS LES RÉSEAUX DE CAPTEURS

Sommaire

4.1	Heuristique naïve	71
4.2	Heuristique Highly Bounded Backbone (HBB)	72
4.3	Recuit simulé	74
4.4	Expérimentations	76
4.4.1	Description	77
4.4.2	Résultats et analyses	78
4.4.3	Synthèse des expérimentations	82
4.5	Synthèse	82

Dans ce chapitre, nous présentons la résolution du problème d'ensemble dominant connexe de poids minimal et de distance en nombre de sauts bornée via deux heuristiques et le recuit simulé. Ce chapitre commence en présentant une heuristique naïve, puis une heuristique représentant une amélioration de l'algorithme d'approximation présenté dans le chapitre 3. Ensuite, le recuit simulé adapté à notre problème est décrit. Pour finir, le chapitre présente les simulations permettant d'évaluer les différentes méthodes proposées.

4.1/ HEURISTIQUE NAÏVE

Dans cette section, nous proposons une heuristique naïve permettant de définir une borne supérieure lorsque la diffusion est effectuée de manière naïve. Toutefois, même s'il s'agit d'une méthode naïve, son coût énergétique reste inférieur au coût généré lors d'une inondation (*flooding*). En effet, dans la méthode d'inondation, tous les nœuds retransmettent le message à diffuser. Dans ce cas de figure, le coût énergétique d'une diffusion est égal à la somme des coûts énergétiques dépensés par tous les nœuds appartenant au réseau de capteurs. L'algorithme 2 illustre une heuristique naïve permettant de calculer une dorsale respectant les contraintes d'énergie et de profondeur du problème MWCDs-BHD. Au début, la dorsale se compose de tous les nœuds de l'ensemble V . L'idée de cette heuristique est d'essayer de supprimer tous les nœuds de la dorsale un

par un et, si la nouvelle dorsale respecte les contraintes alors le nœud est définitivement enlevé. La complexité de l'heuristique naïve est polynomiale : $O(|V|^2 + |V||E|)$.

Algorithm 2 Heuristique naïve

Input : Graphe G (WSN), Distance maximale en nombre de sauts $h_{\max}$

Output : Dorsale B

```

 $B \leftarrow V$ 
for  $x$  in  $B$  do
  if  $(is\_cds(B \setminus \{x\})) \wedge (\varepsilon_{B \setminus \{x\}}(0) \leq h_{\max})$  then
     $B \leftarrow B \setminus \{x\}$ 
  end if
end for
return  $B$ 

```

4.2/ HEURISTIQUE HIGHLY BOUNDED BACKBONE (HBB)

L'heuristique Highly Bounded Backbone se compose de deux étapes principales. La première est la construction d'un nouveau graphe G' à partir du graphe G représentant le réseau de capteurs. La seconde consiste à appliquer une version modifiée de l'heuristique Bounded Backbone à G' . Le graphe G' constitue le graphe contenant tous les chemins de longueur au plus $h_{\max}$ depuis la station de base. Nous remarquons que dans le graphe G' , il existe au moins un chemin pour chaque nœud dans G car $h_{\max}$ est supérieur ou égal à l'excentricité ϵ à la station de base. De plus, le plus petit chemin pondéré ayant une profondeur bornée par $h_{\max}$ est inclus dans G' .

Pour construire G' , l'heuristique Highly Bounded Backbone détermine les nœuds accessibles pour chaque niveau n_i avec i variant de 0 à $h_{\max}$. Pour chaque nœud x du niveau n_i , ses voisins sortants sont ajoutés dans le graphe G' ainsi que les arêtes correspondantes. L'ensemble des voisins sortants de tous les nœuds de niveau n_i constituent l'ensemble des nœuds accessibles au niveau n_{i+1} . Nous procédons itérativement jusqu'à construire le niveau $n_{h_{\max}}$.

Une fois le graphe G' obtenu, nous pouvons appliquer la version modifiée de l'heuristique Bounded Backbone. La différence avec la version originale est que nous avons besoin seulement de l'arborescence $\mathcal{A}_2$ des plus courts chemins pondérés et le chemin choisi pour atteindre le nœud k en cours de traitement est nécessairement le plus court chemin pondéré car tous les chemins ont une longueur d'au plus $h_{\max}$. De plus, une autre différence est la façon dont l'heuristique choisit le nœud k à traiter à chaque étape. Dans l'heuristique Bounded Backbone, le nœud k est choisi dans l'arborescence $\mathcal{A}_1$, alors que dans l'heuristique Highly Bounded Backbone il sera choisi dans $\mathcal{A}_2$. En d'autres termes, k est le nœud le plus distant du point de vue du coût énergétique. Il ne serait pas utile de choisir le nœud le plus distant en nombre de sauts car tous les chemins sont bornés par $h_{\max}$ par construction.

L'algorithme 3 illustre l'heuristique Highly Bounded Backbone.

Exemple

Algorithm 3 Heuristique Highly Bounded Backbone**Input :** Graphe $G = (V, E)$, Distance en nombre de sauts maximale $h_{\max}$ **Output :** Dorsale B n_i ensemble des nœuds atteignables avec i sauts $id_{G'}$ indice des nœuds dans G' $corrTab[|V|]$ table de correspondance entre les indices des nœuds dans G et G' $n_0 \leftarrow 1$ // Étape 1 : calcul du graphe G' **for** $h \in \{0, \dots, h_{\max} - 1\}$ **do** **for** $id \in n_h$ **do** **for** $neighbour$ in $N^+(getNodeInG(id))$ **do** $n_{h+1} \leftarrow n_{h+1} \cup \{id_{G'}\}$ $V' \leftarrow V' \cup \{id_{G'}\}$ $E' \leftarrow E' \cup \{(id, id_{G'})\}$ $corrTab[neighbour] \leftarrow id_{G'}$ $id_{G'} \leftarrow id_{G'} + 1$ **end for** **end for****end for** $B \leftarrow \emptyset$ $U \leftarrow V$ $D \leftarrow \emptyset$

//Étape 2 : version modifiée de l'algorithme BB

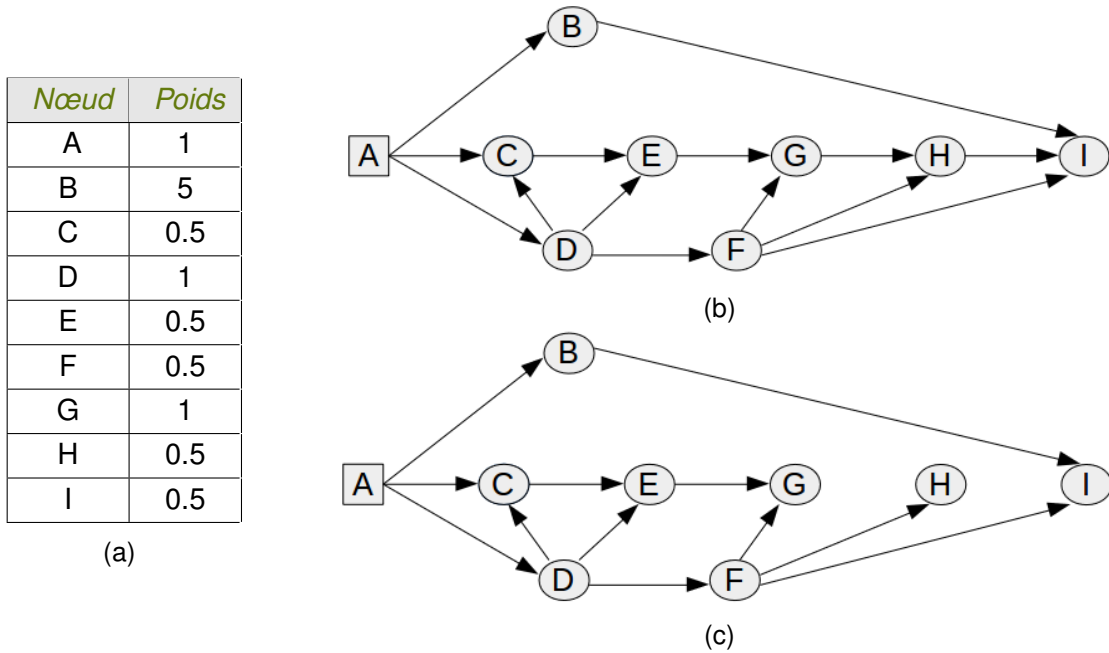
while $U \neq \emptyset$ **do** $k \leftarrow$ le nœud le plus distant dans $\mathcal{A}_2$ non encore traité $P \leftarrow P_2(k)$ $B \leftarrow B \cup \{x \mid x = getNodeInG(x') \wedge x' \in P \wedge x' \neq k\}$ $D \leftarrow D \cup \bigcup_{(x' \in P) \wedge (x' \neq k)} N^+(x')$ $U \leftarrow U \setminus (\{x \mid x = getNodeInG(x') \wedge x' \in P\}) \cup (\bigcup_{(x' \in P) \wedge (x' \neq k)} N^+(x'))$ **for** x tel que $x \in P \wedge x \neq k$ **do**

//Phase de mise à zéro des énergies

 $W_x \leftarrow 0$ **end for** Recalculer $\mathcal{A}_2$ **end while****return** B

Dans cette section, nous présentons un exemple permettant de comprendre le fonctionnement de l'heuristique Highly Bounded Backbone. Il permet de mettre en évidence pourquoi la construction du graphe G' , en amont de l'utilisation de l'heuristique Bounded Backbone, permet un gain énergétique.

Cet exemple permet de voir quelle est l'amélioration effectuée par l'heuristique Highly Bounded Backbone par rapport à l'heuristique Bounded Backbone par le biais de l'exemple 4.1. La figure 4.1b représente le réseau de capteurs considéré. Le tableau 4.1a montre les pondérations des différents nœuds. Pour l'exemple, nous considérons que

FIGURE 4.1 – Exemple de construction du graphe G'

$h_{\max}$ a une valeur de 3. Pour atteindre le nœud I , l'heuristique Bounded Backbone calcule $P_2(I)$ et $P_1(I)$. Le chemin $P_2(I)$ se compose des nœuds A, C, E, G et H et son poids est de 3. Le chemin $P_1(I)$ est constitué des nœuds A et B , et son poids est de 6. L'heuristique Bounded Backbone choisit le chemin $P_1(I)$ car $P_2(I)$ ne respecte pas les contraintes de profondeur bien qu'il soit moins coûteux. L'heuristique Highly Bounded Backbone construit dans un premier temps le graphe G' représenté dans la figure 4.1c. Ensuite, pour atteindre le nœud I elle peut choisir entre les chemins suivants :

- $P_1(I)$ constitué des nœuds A et B , et ayant un coût énergétique de 6,
- $P_2(I)$ composé des nœuds A, D et F , et ayant un coût énergétique de 4.

Nous remarquons que la suppression des arêtes (G, H) et (H, I) a permis à l'heuristique Highly Bounded Backbone de considérer un nouveau chemin P_2 dont le coût énergétique est plus faible que P_1 et qui respecte la contrainte de longueur. Pour conclure, la construction du graphe G' permet à l'heuristique Highly Bounded Backbone de considérer des chemins, s'ils existent, qui sont un compromis entre les chemins P_1 et P_2 .

Nous venons de présenter l'heuristique Highly Bounded Backbone. Dans la section suivante nous abordons la résolution du problème MWCDs-BHD en utilisant le recuit simulé.

4.3/ RECUIT SIMULÉ

L'objectif du recuit simulé sera de trouver un ensemble dominant connexe de coût énergétique minimale et de profondeur maximale bornée par $h_{\max}$.

Normalement, le recuit simulé prend comme paramètre la température initiale et la configuration initiale. Nous avons créé une fonction pour calculer la température initiale. Cette fonction prend comme paramètre le taux initial d'acceptation ρ_0 des configurations voisines qui dégradent la fonction objectif puis effectue un nombre prédéfini de changements aléatoires. Un changement entre deux configurations est un ajout ou une suppression d'un nœud. La moyenne $\overline{\Delta E}$ des écarts énergétique $|\Delta E|$ entre deux configurations est calculée. La température initiale est calculée en utilisant la formule suivante :

$$T_0 = \frac{\ln(\overline{\Delta E}/W_{\max}) \times \ln(|V|)}{\ln \rho_0}.$$

Comme nous pouvons le voir, la température initiale T_0 est directement proportionnelle au nombre de nœuds et au gain énergétique moyen possible à chaque perturbation effectuée. En revanche, elle est inversement proportionnelle au taux d'acceptation ρ_0 . Cela permet de commencer à une température initiale de plus en plus élevée lorsque le nombre de nœuds et le gain énergétique moyen augmentent. Plus le taux d'acceptation augmente, plus le dénominateur $\ln \rho_0$ diminue et, par conséquent, plus la température augmente. Une température élevée augmente la probabilité d'accepter des configurations dégradant la fonction objective et, donc, de sortir avec une plus grande probabilité des extrema locaux.

La configuration initiale du recuit simulé ou dorsale initiale B_0 est constituée de tous les nœuds du réseau de capteurs. L'algorithme du recuit simulé adapté à la construction d'une dorsale est décrit par l'algorithme 4.

À chaque itération, le recuit simulé calcule une nouvelle configuration voisine via la fonction *computeNewConfiguration()*. Ensuite, le ΔE entre la dorsale actuelle et la dorsale nouvellement établie est calculé. Si le ΔE est inférieur ou égal à zéro alors la nouvelle dorsale est conservée. Sinon, elle peut toujours être acceptée avec une probabilité

$$p = \exp \left(- \frac{\left(\frac{|\Delta E|}{E_{tot}} \right)}{T} \right)$$

où E_{tot} correspond à l'énergie totale du réseau et T à la température actuelle du système.

Nous considérons qu'une configuration est une configuration voisine d'une configuration c si une seule modification élémentaire de la dorsale est nécessaire pour passer de la configuration c à la configuration voisine. À chaque itération, un nœud u est choisi aléatoirement. Si u est dans la dorsale, alors u est retiré de la dorsale. Sinon, u est ajouté à celle-ci. Ce processus correspond à une modification élémentaire. Le calcul d'une configuration voisine est fait par la fonction *computeNewConfiguration()*. Lorsque *computeNewConfiguration()* construit une configuration voisine, cette fonction vérifie que la nouvelle configuration respecte les contraintes de connexité et de profondeur.

À la fin de chaque itération, la température est mise à jour. Comme indiqué dans l'algorithme 4, nous avons un palier par température. La température diminue selon le schéma de refroidissement après $100 \times N$ configurations rejetées.

L'espace de recherche exploré par le recuit simulé est connexe. En effet, considérons deux états représentés par deux dorsales B_i et B_j . Il est possible de passer de

B_i à B_j et inversement à travers la dorsale B_0 composée de tous les nœuds du graphe. Considérons le cas où nous voulons passer de B_i à B_j . Cela peut être fait en ajoutant les nœuds du graphe pas encore présents dans B_i jusqu'à ce que tous les nœuds appartiennent à la dorsale. Ensuite, il reste à supprimer les nœuds "supplémentaires" afin d'atteindre la dorsale B_j . Cela est possible car les liens entre les différentes dorsales sont symétriques.

Algorithm 4 Métaheuristique recuit simulé

Input : Graphe G (WSN), Distance maximale en nombre de sauts $h_{\max}$, Taux d'acceptation ρ_0

Output : Dorsale B

$B_0 \leftarrow V$

$T \leftarrow \text{computeInitialTemperature}(\rho_0)$

$B \leftarrow B_0$

while $T > 0.01$ **do**

while ($|\text{Configurations refusées}| < 100 * N \vee (\text{random}(0, 1) \leq p)$) **do**

$B' \leftarrow \text{computeNewConfiguration}(B)$

$\Delta E \leftarrow E_B - E_{B'}$

if $\Delta E \leq 0$ **then**

$E_B \leftarrow E_{B'}$

$B \leftarrow B'$

else

$E_B \leftarrow E_{B'}, B \leftarrow B'$

end if

end while

$T \leftarrow 0.9 \times T$

end while

return B

Maintenant que les différentes méthodes proposées pour résoudre le MWCDs-BHD ont été présentées, nous pouvons les évaluer grâce à des simulations.

4.4/ EXPÉRIMENTATIONS

Dans cette section, nous présentons quelques expérimentations pour évaluer les performances des différentes méthodes proposées pour la résolution du MWCDs-BHD. Deux approches différentes sont utilisées : pour les petites instances de réseaux, il est possible de comparer les résultats donnés par les heuristiques avec le résultat optimal calculé par le programme linéaire en nombres entiers proposé dans la section 3.1.2. Cependant, pour les instances de taille moyenne, cette comparaison devient impossible car le temps nécessaire pour obtenir un résultat avec le programme linéaire entier est excessif (plus de 15 minutes). Dans le cas d'instances de taille moyenne, les heuristiques sont évaluées et comparées au recuit simulé. Pour les grandes instances, nous effectuons seulement une comparaison des heuristiques Bounded Backbone, Highly Bounded Backbone et l'heuristique naïve.

4.4.1/ DESCRIPTION

Nous considérons les graphes composés de n nœuds ayant une portée appartenant à l'intervalle $[r_{\min}; r_{\max}]$ où $r_{\min} = c \times r_{\max}$ et c est une constante entière positive prédéfinie que nous choisissons supérieure à 1. Les positions des nœuds sont établies aléatoirement et uniformément dans un carré de taille L .

Le déploiement des nœuds de capteurs est influencé par la portée de communication r et la densité du réseau. La formule calculant la densité δ , c'est-à-dire le nombre moyen de nœuds dans le voisinage, est la suivante :

$$\delta = \pi \left(\frac{n \times r^2}{L^2} \right)$$

Puisque, dans notre cas, r est une variable aléatoire définie sur l'intervalle $[r_{\min}; r_{\max}]$ qui suit la loi uniforme, nous devons calculer l'espérance de r . Par conséquent, la densité δ se calcule comme indiqué ci-dessous :

$$\delta = \pi \left(\frac{n \times \mathbb{E}[r^2]}{L^2} \right)$$

Où $\mathbb{E}[r^2]$ se détermine à l'aide de la formule suivante :

$$\begin{aligned} \mathbb{E}[r^2] &= \int_{r_{\min}}^{r_{\max}} r^2 \times \frac{1}{r_{\max} - r_{\min}} dr \\ &= \frac{r_{\max}^2 + r_{\max} \times r_{\min} + r_{\min}^2}{3} \end{aligned}$$

Pour les simulations, la densité et la portée de transmission sont fixées. Ainsi, il est possible de déduire la longueur du côté L . Comme indiqué précédemment, le modèle de connexité utilisé est le modèle de graphe disque. Plus précisément, si r_i est la portée de transmission du nœud i , et d_{ij} est la distance euclidienne entre le nœud i et le nœud j alors j est un voisin de i si, et seulement si, $d_{ij} \leq r_i$.

Pour simplifier les calculs, nous calculons l'énergie dépensée par un nœud u lors de la transmission de la manière suivante : $w_u = r_u^2$. Ce calcul correspond à une simplification du First Order Radio Model [7].

Dans les scénarios présentés dans les figures 4.2 et 4.3, le nombre de nœuds a été fixé à 25 et la densité évolue de 7 à 15. Dans ces conditions, il est possible d'évaluer les heuristiques proposées par rapport à la valeur optimale obtenue en résolvant le programme linéaire.

Dans les scénarios présentés dans les figures 4.4 et 4.5, l'objectif est de considérer les réseaux de taille moyenne et de comparer les heuristiques avec le recuit simulé. Pour ces scénarios, une densité de 15 est considérée et le nombre de nœuds évolue de 20 à 100.

Dans le dernier scénario présenté dans les figures 4.6 et 4.7, l'objectif est d'évaluer les heuristiques. Dans ce cas, la densité du réseau est de 15 et le nombre de nœuds augmente de 100 à 1000.

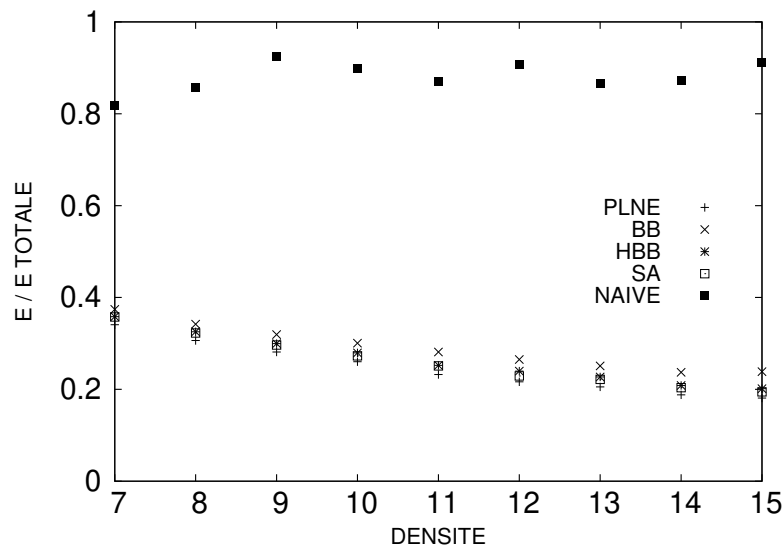


FIGURE 4.2 – Consommation énergétique de la solution optimale (ILP), recuit simulé (SA), heuristique naïve (NAIVE), heuristique Highly Bounded Backbone (HBB) et heuristique Bounded Backbone (BB) lorsque $h_{\max}$ est égal à ε et lorsque le nombre de nœuds est 25.

Dans tous les scénarios, le rayon de communication $r_{\min}$ est égal à 20 et $r_{\max}$ à 40. Nous considérons que $h_{\max}$ est équivalent à l'excentricité de la station de base ε . Pour chaque scénario, 100 réseaux aléatoires sont calculés. La position de chaque nœud est tirée uniformément dans l'espace considéré. La portée est tirée uniformément dans l'intervalle $[r_{\min}; r_{\max}]$.

GNU Linear Programming Kit (GLPK) est utilisé pour résoudre le programme linéaire en nombres entiers. L'énergie prise en compte dans les simulations est l'énergie consommée lors de la diffusion. L'énergie nécessaire pour construire la dorsale n'est pas prise en compte. Nous supposons que la construction de la dorsale est centralisée sur une ressource puissante. Les résultats de ces expériences sont donnés dans la section suivante.

4.4.2/ RÉSULTATS ET ANALYSES

Dans cette section, nous présentons les résultats des différents scénarios. D'abord nous effectuons la comparaison entre le programme linéaire en nombres entiers, le recuit simulé et les heuristiques. Ensuite, nous comparons les heuristiques et le recuit simulé. Pour finir, nous effectuons les simulations pour comparer les heuristiques entre elles.

Nous nous attendons à voir que l'heuristique Highly Bounded Backbone est meilleure que l'heuristique Bounded Backbone. Cela est dû au fait que l'heuristique Highly Bounded Backbone est une amélioration de l'heuristique Bounded Backbone. En effet, grâce à son prétraitement, elle est capable de considérer, et donc de choisir, des chemins respectant la contrainte sur la longueur mais dont le coût est inférieur à celui du plus court chemin en nombre de sauts. De plus, le recuit simulé devrait être meilleur que les heuristiques Highly Bounded Backbone et Bounded Backbone. Cela s'explique par le fait qu'il est capable de considérer tous les chemins reliant la station de base aux autres

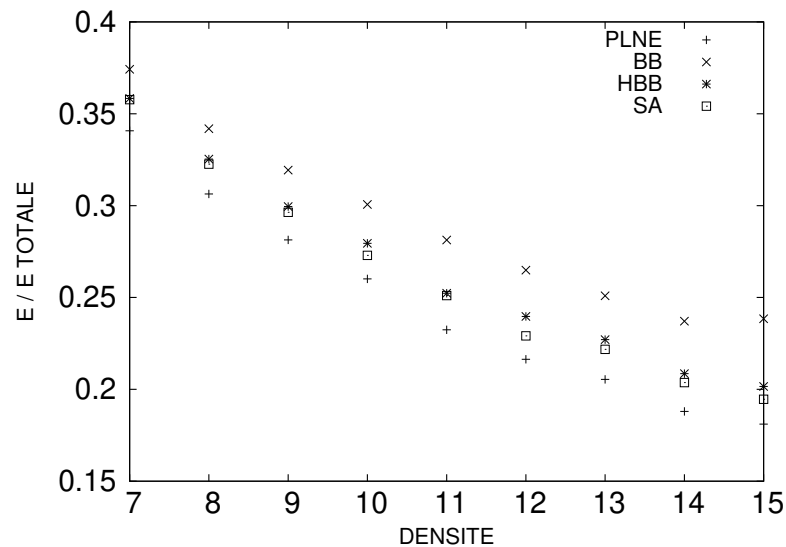


FIGURE 4.3 – Consommation énergétique de la solution optimale (ILP), recuit simulé (SA), heuristique Highly Bounded Backbone (HBB) et heuristique Bounded Backbone (BB) lorsque $h_{\max}$ est égal à ε et lorsque le nombre de nœuds est 25.

nœuds.

Les figures 4.2 et 4.3 illustrent l'évolution du rapport entre l'énergie utilisée par la dorsale construite par les heuristiques et l'énergie totale du réseau en fonction de la densité du réseau lorsque $h_{\max}$ est égal à l'excentricité de la station de base. La figure 4.3 est un zoom de la figure 4.2 dans lequel on a supprimé l'heuristique Naïve.

Dans la figure 4.2, il est possible d'observer que l'heuristique naïve construit une dorsale dont l'énergie est approximativement comprise entre 82% et 93% de l'énergie totale. Alors que les autres solutions ont trouvé une dorsale avec une énergie comprise entre 20% et 37% du coût énergétique total du réseau.

Nous observons que plus la densité augmente, plus le coût énergétique de la dorsale diminue par rapport à l'énergie totale du réseau. Ceci s'explique par le fait que l'augmentation de la densité signifie l'augmentation du nombre de voisins. Par conséquent, un nœud peut dominer plus de nœuds et donc moins de nœuds sont nécessaires pour dominer l'ensemble du réseau.

Dans la figure 4.3, nous pouvons voir le comportement du programme linéaire, de l'heuristique Bounded Backbone, de l'heuristique Highly Bounded Backbone et du recuit simulé. Nous remarquons que le rapport entre l'énergie utilisée par la dorsale et l'énergie totale du réseau décroît de manière quasi linéaire avec l'augmentation de la densité. De plus, nous pouvons voir que le recuit simulé trouve la solution la plus proche de la solution optimale. Comme prévu, l'heuristique Highly Bounded Backbone est meilleure que l'heuristique Bounded Backbone : le gain est d'environ 2%. En regardant le rapport entre les solutions trouvées par les heuristiques et le programme linéaire en nombres entiers, nous notons que l'heuristique Highly Bounded Backbone calcule des solutions d'au plus 1,1 fois l'énergie de la dorsale optimale. L'heuristique Bounded Backbone trouve des solutions dont le coût énergétique est au plus de 1,3 la valeur de la solution optimale. Enfin, le recuit simulé calcule une dorsale dont le coût en énergie est d'au plus 1,08 fois l'énergie de la dorsale optimale.

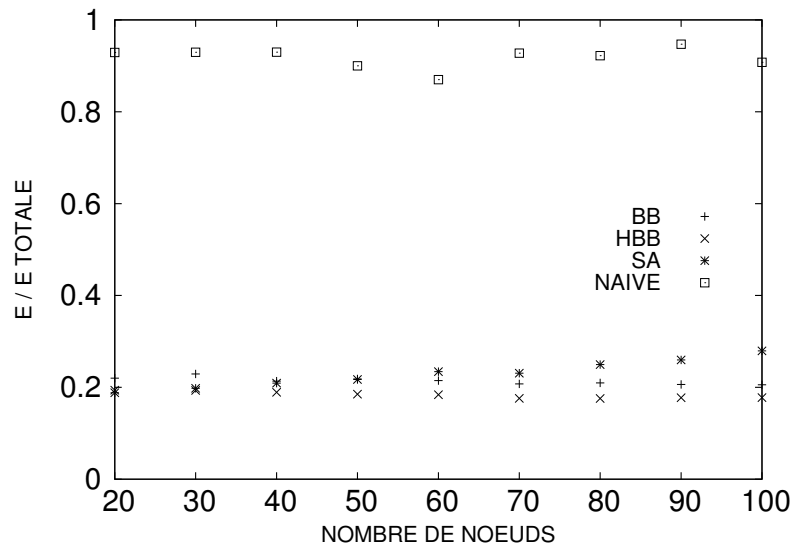


FIGURE 4.4 – Consommation énergétique du recuit simulé (SA), heuristique naïve (NAIVE), heuristique Highly Bounded Backbone (HBB) et heuristique Bounded Backbone (BB) lorsque $h_{\max}$ est égal à ε et la densité est égale à 15.

Les figures 4.4 et 4.5 représentent l'évolution du rapport entre l'énergie utilisée par la dorsale construite par les heuristiques et l'énergie totale du réseau en fonction du nombre de nœuds du réseau. Dans ce scénario, le cas où $h_{\max}$ est égal à l'excentricité de la station de base est considéré. La figure 4.5 est un zoom de la figure 4.4 privée de l'heuristique Naïve.

Dans la figure 4.4, l'heuristique naïve construit une dorsale dont le coût est compris entre 87% et 95% de l'énergie totale du réseau. Le recuit simulé et les heuristiques Highly Bounded Backbone et Bounded Backbone construisent une dorsale dont l'énergie représente 18% à 28% de l'énergie totale du réseau. Nous pouvons voir un gain d'au moins 77% en utilisant le recuit simulé ou les heuristiques Highly Bounded Backbone et Bounded Backbone à la place de l'heuristique naïve.

Dans la figure 4.5, nous voyons que le recuit simulé donne respectivement de meilleurs résultats que l'heuristique Highly Bounded Backbone/Bounded Backbone jusqu'à 20/40 nœuds. Ensuite, le coût énergétique de la dorsale construite par le recuit simulé devient progressivement supérieur à celle des heuristiques Bounded Backbone et Highly Bounded Backbone jusqu'à atteindre une différence de 10% d'énergie supplémentaire. Ce résultat suggère que les paramètres utilisés pour le recuit simulé, à savoir la température initiale, le nombre d'itérations par palier de température et le schéma de refroidissement sont adaptés jusqu'à 40 nœuds. Au-delà, il faudrait, par exemple, utiliser une autre formule pour calculer la température initiale afin qu'elle soit plus élevée et, par conséquent, que le recuit simulé ait plus d'itérations avant de converger. Une autre façon pourrait être d'augmenter le nombre d'itérations par palier de température. Cependant, tous ces changements entraîneraient une augmentation du temps de calcul du recuit simulé. Nous devons donc faire un compromis entre le temps de calcul et la qualité de la solution. Comme précédemment, dans la figure 4.5, on peut voir que l'heuristique Highly Bounded Backbone est meilleure que l'heuristique Bounded Backbone d'au moins 2,4% (cas de 40 nœuds) à 3,6% (cas de 30 nœuds). Notez que plus le nombre de nœuds augmente, plus la différence entre les heuristiques Highly Bounded Backbone et Bounded

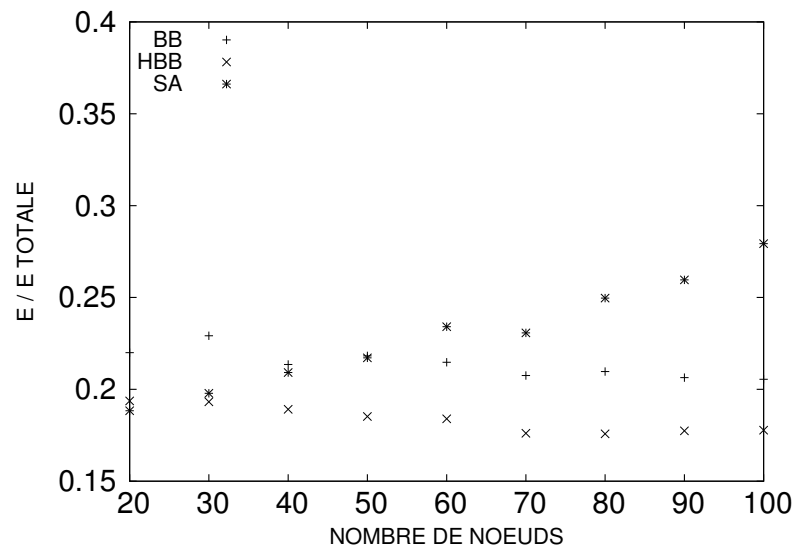


FIGURE 4.5 – Consommation d'énergie du recuit simulé (SA), heuristique Highly Bounded Backbone (HBB) et heuristique Bounded Backbone (BB) lorsque $h_{\max}$ est égal à ε et la densité est égale à 15.

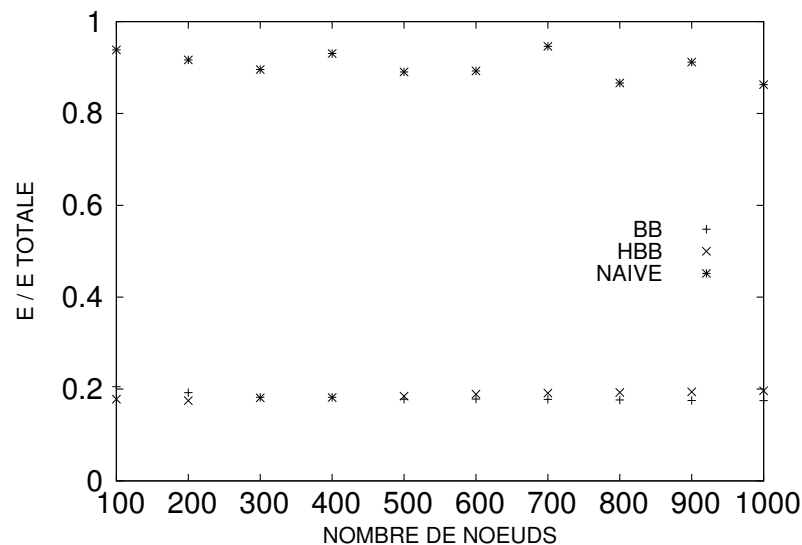


FIGURE 4.6 – Consommation énergétique des heuristiques naïve (NAIVE), Highly Bounded Backbone (HBB) et Bounded Backbone (BB) lorsque $h_{\max}$ est égal à ε et la densité est égale à 15.

Backbone diminue.

Les figures 4.6 et 4.7 affichent l'évolution du rapport entre l'énergie utilisée par le dorsale des heuristiques et l'énergie totale du réseau en fonction du nombre de nœuds du réseau. Nous considérons le cas où $h_{\max}$ est égal à l'excentricité de la station de base.

Dans la figure 4.6, nous pouvons remarquer que l'heuristique naïve construit une dorsale dont le coût en énergie est égal à 86% et 94% de l'énergie totale du réseau. Le coût de la dorsale obtenue par les heuristiques Highly Bounded Backbone et Bounded Backbone est de 17% à 20% de l'énergie totale du réseau.

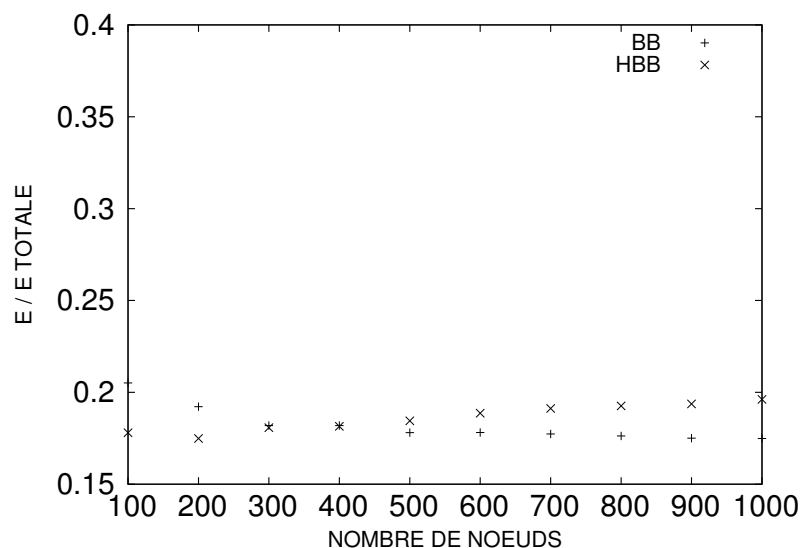


FIGURE 4.7 – Consommation énergétique des heuristiques Highly Bounded Backbone (HBB) et Bounded Backbone (BB) lorsque $h_{\max}$ est égal à ε et la densité est égale à 15.

En regardant la figure 4.7, nous remarquons que jusqu'à 400 nœuds, l'heuristique Highly Bounded Backbone donne de meilleurs résultats que l'heuristique Bounded Backbone. Après 400, l'heuristique Bounded Backbone calcule les meilleures solutions. La différence varie entre 0,6% (500 nœuds) et 2,1% (1 000 nœuds). L'heuristique Bounded Backbone peut choisir des chemins qui coûtent plus cher que ceux choisis par l'heuristique Highly Bounded Backbone par conception. Mais après 400 nœuds, l'heuristique Bounded Backbone choisit moins de chemins pour dominer l'ensemble du graphe et ainsi obtenir de meilleurs résultats.

4.4.3/ SYNTHÈSE DES EXPÉRIMENTATIONS

Le programme linéaire, nous permet d'obtenir la solution optimale mais nous avons pu obtenir des résultats uniquement pour des réseaux de petite taille. En comparant nos propositions à la solution optimale, il est possible d'établir que, pour les paramètres choisis, le recuit simulé est le plus proche de la solution optimale pour les réseaux d'une taille de 25 ou moins. Pour les réseaux de taille comprise entre 30 et 400, le meilleur est l'heuristique Highly Bounded Backbone. Pour les réseaux de taille supérieure à 400, l'heuristique Bounded Backbone donne les meilleures solutions. Ces conclusions sont résumées dans la figure 4.8.

4.5/ SYNTHÈSE

Ce chapitre a présenté l'étude de l'optimisation de la diffusion dans les réseaux de capteurs sans fils et hétérogènes du point de vue de la portée de communication. Nous avons présenté deux méthodes pour résoudre ce problème : une heuristique nommée Highly Bounded Backbone et le recuit simulé. Des simulations ont également été conduites afin de comparer les performances des algorithmes proposés dans ce chapitre et dans le chapitre 3.

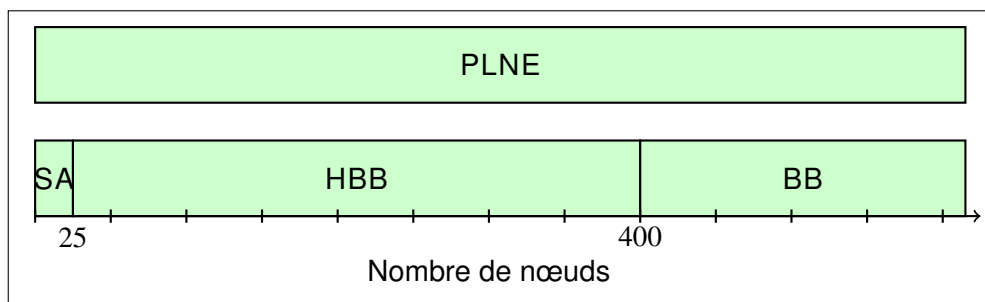


FIGURE 4.8 – Synthèse des performances des heuristiques en fonction du nombre de nœuds du réseau

ROUTAGE MULTI-COUCHE

Sommaire

5.1 Protocoles cross-layer dans les WSN	86
5.1.1 Approches cross-layer inter-couche	86
5.1.2 Approches cross-layer par fusion de couches	88
5.2 Formalisation du problème	89
5.2.1 Définition du problème	89
5.2.2 Modélisation de la solution	90
5.3 Programme linéaire en nombres entiers	90
5.4 Algorithme IDeg-Routing&MAC	93
5.5 Exemple explicatif	93
5.6 Simulations	96
5.6.1 Description	96
5.6.2 Résultats	97
5.6.3 Discussion	98
5.7 Synthèse	99

Cette thèse s'intéresse à l'optimisation de l'énergie et de la latence dans les réseaux de capteurs sans fils. C'est pour cela, que nous considérons les approches cross-layer. En effet, elles nous permettent d'atteindre des meilleurs résultats globaux pour ces deux métriques par rapport aux approches en couches classiques. En particulier, ces approches permettent une consommation d'énergie moindre [42], ce qui est très approprié dans le cas du réseau de capteurs sans fil.

Ce chapitre se structure comme suit. La section 5.1 expose certains protocoles cross-layer par fusion de couches ou inter-couche impliquant les couches MAC et Liaison de données trouvés dans la littérature. Les travaux présentés supposent la présence d'une seule station de base au sein du réseau de capteurs sans fils. La section 5.2 donne la définition du problème traité dans ce chapitre. La section 5.3 présente notre proposition qui est un programme linéaire utilisant la méthode de cross-layering. La section 5.4 détaille l'heuristique que nous évaluons par rapport au PLNE proposé et la section 5.5 montre le comportement de cette heuristique ainsi que du programme linéaire par le biais d'un exemple. La section 5.6 présente les paramètres et les résultats des simulations. Enfin, la section 5.7 conclut ce chapitre.

5.1/ PROTOCOLES CROSS-LAYER DANS LES WSN

De nombreux travaux dans la littérature traitent des approches cross-layer MAC et Liaison de données. Ces travaux peuvent être classés suivant le type de cross-layer utilisé. Dans le premier type, deux catégories existent. Dans la première, la couche MAC exploite les informations de l'arbre de routage afin d'établir les décisions temporelles. Dans la seconde, la couche de Liaison de données utilise les décisions temporelles pour définir l'arbre de routage. Dans le second type, les couches MAC et Liaison de données sont combinées en une seule couche assurant les deux fonctionnalités.

Dans la suite, nous présentons quelques travaux existants dans la littérature traitant les deux types de cross-layer. Tous les travaux qui seront cités dans les sections suivantes considèrent la présence d'une seule station de base dans le réseau. Toutefois, il existe des travaux tels que [67, 77, 85] proposant des approches cross-layer inter-couches entre MAC et Liaison de données mais dans des réseaux de capteurs sans fil avec multiples stations de base.

5.1.1/ APPROCHES CROSS-LAYER INTER-COUCHE

Au sein de cette catégorie, il est possible de diviser les travaux en deux classes : MAC-aware routing et Routing-aware MAC qui seront présentées dans la suite.

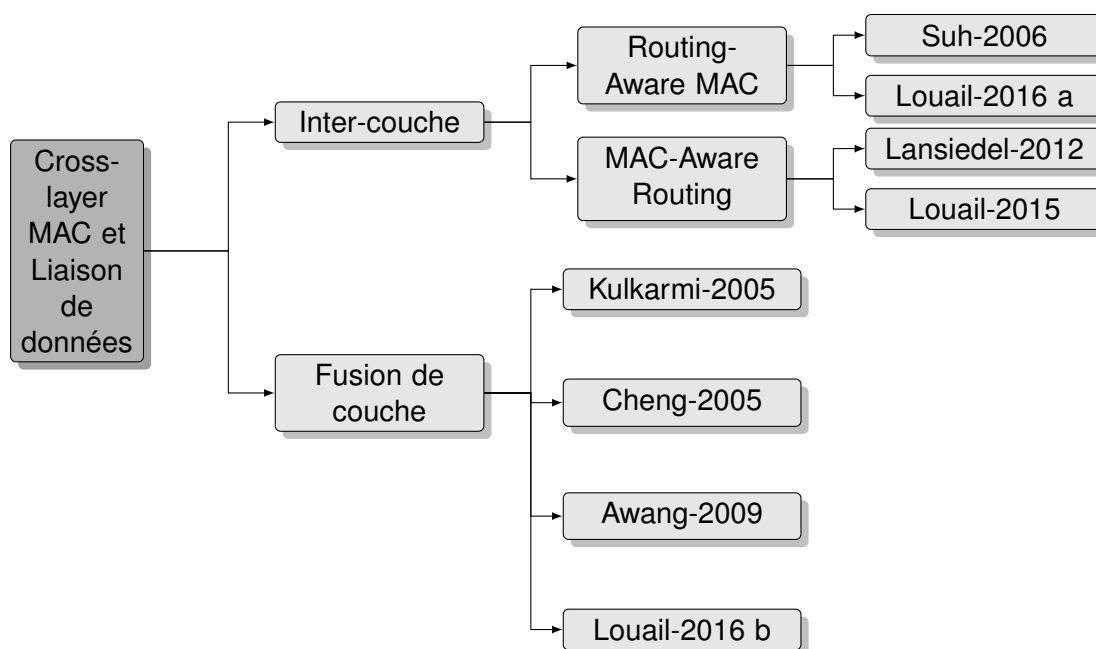


FIGURE 5.1 – Résumé des travaux cross-layer entre couche MAC et Liaison de données

Classe MAC-aware routing

Dans ce type de cross-layering, les décisions temporelles du protocole MAC sont

données à la couche de Liaison de données afin qu'elle puisse en tirer parti et choisir un chemin approprié entre chaque nœud et le destinataire. ORW [61] *Opportunistic Routing in Wireless sensor networks* et MAR-WSN [78] sont deux exemples de ce type d'approches où le TDMA est utilisé par la couche Liaison de données pour choisir le prochain saut pour chaque nœud en prenant en compte les créneaux de transmission des nœuds.

Plus précisément, ORW adapte le concept de routage opportuniste aux besoins et challenges propres aux réseaux de capteurs. En effet, il se concentre sur l'optimisation de l'énergie tout en tenant compte du cycle de fonctionnement (*duty-cycle*) des nœuds. Lorsqu'un paquet doit être transmis, celui-ci est envoyé par le premier voisin réveillé qui le reçoit avec succès et s'approchant de la destination finale. Ce qui permet l'amélioration considérable de l'efficacité énergétique tout en réduisant les délais de bout en bout. Les simulations montrent que ORW réduit en moyenne le *duty-cycle* de 50% et le délai de 30% à 90% par rapport aux approches auxquelles il a été comparé. De plus, il double l'efficacité énergétique dans le cas des réseaux denses.

MAR-WSN est une heuristique prenant en entrée un ordonnancement TDMA. Dans l'article, les auteurs choisissent l'utilisation d'un TDMA aléatoire. Les auteurs supposent que tous les nœuds connaissent leur position géographique, celle de leurs voisins à un et deux sauts ainsi que celle de la station de base. Le principe de cette approche est de choisir le prochain saut en fonction de la métrique WDD (*Weighted Distance Delay*) calculée pour le nœud courant pour tout couple de voisins à un et deux sauts par rapport à leurs informations sur le TDMA et leur distance à la station de base. Le prochain saut sera donc celui permettant de minimiser la métrique WDD. MAR-WSN a été évalué via des simulations en utilisant les UDG pour modéliser le réseau. Les simulations montrent que pour les réseaux ayant une densité moyenne ou haute, MAR-WSN donne des meilleurs résultats pour le délai et la consommation d'énergétique. Cependant, pour les réseaux de faible densité, les résultats sont proches de ceux obtenus par l'heuristique à laquelle il a été comparé.

Classe Routing-aware MAC

Les approches existantes de ce type peuvent également être classées en deux catégories : les approches distribuées et les approches centralisées.

La plupart des approches distribuées utilisent les informations de l'arbre de routage afin de prendre des décisions temporelles dans la couche MAC basée sur la contention. Par exemple, MAC-CROSS [33] exploite les informations de routage afin d'être efficace en termes énergétique via la minimisation du nombre de nœuds réveillés en maximisant leur durée de sommeil. Pour cela, la couche MAC exploite les informations de la couche de Liaison de données à savoir la route empruntée par un paquet afin que les nœuds ne faisant pas partie de celle-ci puissent rester endormis et donc économiser de l'énergie. Suh et al. ont implémenté le MAC-CROSS sur les capteurs Mica ainsi que sur le simulateur NS-2 [46]. Les résultats expérimentaux démontrent que leur heuristique économise au minimum 16.7% d'énergie.

Dans les approches centralisées, une couche MAC sans contention utilise les informations de l'arbre de routage. Louail et al. [81] proposent des approches TDMA uti-

lisant l'arbre de routage du réseau pour optimiser la latence des communications dans les réseaux de capteurs sans fil. Les auteurs définissent les ordonnancements TDMA en utilisant les informations de l'arbre de routage. Leur première contribution consiste en une méthode de backtracking pour calculer la solution optimale. Ils proposent, de plus, différentes heuristiques calculant l'ordonnancement TDMA minimisant la latence de tous les nœuds. Deux types d'ordonnancement TDMA cross-layer par inter-couches sont proposés. Le premier est basé sur différents parcours d'arborescence de routage : cela consiste à attribuer un créneau à un nœud en fonction de sa position dans l'arbre de routage. Deux heuristiques de ce type ont été proposées. Dans Depth-ReLO, la première heuristique, un nœud est ordonnancé en fonction de l'ordre donné par une traversée ascendante (en partant des feuilles et en allant vers la station de base) de l'arbre de routage. Dans la deuxième heuristique, nommée BFS, une traversée descendante est utilisée par le biais d'un parcours en largeur de l'arbre de routage. Le second type d'ordonnancement TDMA cross-layer par inter-couches est basé sur les informations du graphe du réseau et de l'arbre de routage. Lors de l'attribution des créneaux, l'heuristique proposée, ordonnance suivant la position dans l'arbre de routage mais aussi en fonction du nombre de nœuds voisins à un et deux sauts. Les simulations effectuées pour évaluer les heuristiques proposées utilisent le UDG comme modèle de connectivité. Dans les simulations les heuristiques proposées ont été comparées à des approches existantes. Elles montrent que les approches proposées améliorent la latence de 23% à 29%.

5.1.2/ APPROCHES CROSS-LAYER PAR FUSION DE COUCHES

Dans ce type d'approches, les décisions temporelles et spatiales sont prises par une seule couche afin d'obtenir de meilleures performances du réseau. AIMRP [30], RBF [44] et SIF [22] sont trois approches similaires utilisant un protocole MAC basé sur la contention tout en acheminant progressivement les paquets vers des nœuds plus proches du récepteur.

AIMRP [30] est conçu pour une classe d'applications consistant à détecter et signaler un événement à la station de base. La particularité de ce travail, autre que l'utilisation du cross-layering, est de proposer une diminution du coût énergétique tout en gardant des bonnes performances en latence. Afin de diminuer les entêtes des paquets, le protocole considère que les nœuds sont organisés en tiers autour de la station de base. Le tiers i sera constitué de l'ensemble de nœuds qui sont à une portée de communication de i par rapport à la station de base. Les nœuds n'ont donc pas une adresse unique mais se voient affecter l'identifiant de leur niveau. Lors de la transmission, le choix du prochain saut est fait en considérant le *duty-cycle* des nœuds. Ceci permet de diminuer la consommation d'énergie ainsi que la latence. Les paquets sont transférés progressivement niveau par niveau en s'approchant du destinataire final, à savoir la station de base. AIMRP a été évalué par le biais de simulations montrant qu'il surpasse le protocole S-MAC en termes de consommation d'énergie moyenne tout en gardant des performances, en termes de latence, similaires.

L'objectif du protocole cross-layer RBF [44] est de diminuer les délais d'acheminement des paquets vers la station de base. La connaissance du positionnement des nœuds est une opération coûteuse. Par conséquent, RBF n'utilise aucune information sur les positions géographiques des nœuds. Afin de décider le prochain saut, il exploite l'indicateur de puissance du signal reçu RSSI (*Received Signal Strength Indicator*). Le

prochain saut sera un nœud dont le RSSI est supérieur à celui du nœud ayant transféré le message. De cette manière, RBF assure que le message avance dans la direction de la station de base. Les auteurs donnent les délais de transmission obtenus par le protocole RBF sur le scénario étudié. Ces résultats montrent la pertinence de leur algorithme pour le cas d'un réseau de capteurs sans fils utilisant un mode de collecte événementiel.

SIF (*State-free Implicit Forwarding*) [22] est basé sur la position géographique des nœuds et la sélection du prochain saut via la concurrence entre les récepteurs. Les auteurs supposent, donc, que les capteurs connaissent la distance les séparant de la station de base. Lors de la sélection du prochain saut, un message est envoyé et tous les nœuds ayant une distance de la station de base plus petite que celle du nœud courant seront choisis comme potentiels transmetteurs. À ce stade, tous les candidats définissent un temps d'attente devant s'écouler avant de répondre au message reçu. Ce temps d'attente est tel que les capteurs les plus proches de la station de base et ayant le plus d'énergie résiduelle ont une plus grande probabilité d'être le prochain saut. Les résultats des simulations montrent que SIF fonctionne mieux que certains protocoles précédemment proposés en termes de taux de remise des paquets et de délai de transmission moyen. Le réseau est modélisé par un graphe ayant des liens symétriques.

Louail et al. [82] proposent l'heuristique IDeg-Routing&MAC. Il s'agit d'une approche construisant un ordonnancement TDMA et un arbre de routage en même temps afin de minimiser la latence des communications dans les réseaux de capteurs sans fil. Le routage utilisé dans IDeg-Routing&MAC est un routage géographique dans lequel, un nœud choisit le prochain saut parmi ses voisins. Le nœud choisi est celui qui le rapproche le plus de la station de base en termes de distance géographique. L'ordonnancement TDMA est établi en même temps que le routage. La principale stratégie, afin de minimiser la latence, est d'attribuer les créneaux les plus proches possible entre nœuds pères et nœuds fils. Le fonctionnement de IDeg-Routing&MAC sera détaillé dans la section 5.4.

5.2/ FORMALISATION DU PROBLÈME

Dans cette section, nous présentons la problématique traitée ainsi que la modélisation de la solution attendue.

5.2.1/ DÉFINITION DU PROBLÈME

Le problème traité dans ce document est défini comme suit.

Étant donné un graphe orienté et pondéré $G = (V, E, W)$, définir pour chaque message la route à emprunter afin d'arriver à la station de base. La route sera définie comme un ensemble de triplets (m, i, t) où i représente le nœud auquel le message m est à l'instant t . Cela correspond à définir à la fois le routage et l'ordonnancement.

Pour cela, nous considérons les hypothèses suivantes :

- chaque nœud ne peut transférer qu'un seul message à la fois,
- tous les nœuds, sauf la station de base, génèrent un message à l'instant 0,
- tous les messages ont pour destination la station de base.

Un exemple sera donné dans la section 5.5.

5.2.2/ MODÉLISATION DE LA SOLUTION

Le résultat du protocole TDMA peut être représenté comme un ordonnancement. Dans cet ordonnancement, chaque nœud a un créneau de temps assigné pour l'envoi de données à ses voisins. Les créneaux restants peuvent être utilisés pour recevoir des messages des voisins ou ils ne sont pas utilisés du tout.

Pour modéliser un ordonnancement TDMA nous utilisons une matrice $Schedule[N, L]$ où le nombre de lignes N représente le nombre de nœuds et le nombre de colonnes L modélise la longueur de l'ordonnancement. Chaque élément de la matrice représente un créneau de temps. $Schedule[i, j] = i$ signifie que le créneau j est réservé pour la transmissions du nœud i . $Schedule[i, j] = k$ indique que le créneau j est réservé au nœud i pour la réception des données de son voisin k . Pour finir, si la case $Schedule[i, j]$ est vide alors le créneau j n'est pas réservé pour le nœud i ni pour émettre ni pour recevoir. Nous notons que les entiers i et k appartiennent à l'intervalle $[1; N]$ et l'entier j appartient à l'intervalle $[1; L]$. Un créneau réservé n'implique pas forcément qu'il est réellement utilisé car le nœud peut ne pas avoir de messages à transmettre. Un ordonnancement optimal vise, entre autre, à diminuer le nombre de créneaux réservés mais non utilisés.

Le routage est modélisé par une anti arborescence *inconnected tree*.

Nous autorisons la possibilité d'affecter plusieurs créneaux consécutifs aux mêmes nœuds. Cela peut être intéressant, en particulier, pour les voisins de station de base qui doivent acheminer les messages du reste du réseau.

5.3/ PROGRAMME LINÉAIRE EN NOMBRES ENTIERS

Dans cette section, nous introduisons un programme linéaire en nombres entiers (PLNE) modélisant le problème traité. Pour créer un ordonnancement MAC sans contention ainsi qu'une arborescence de routage, le PLNE affecte à chaque nœud un créneau de temps et un nœud père. Dans les sections suivantes, nous présentons la formulation du PLNE.

Définition du PLNE

Pour définir le programme linéaire en nombres entiers, nous spécifions les variables suivantes. $y_{m,i,j,t}$ est une variable binaire indiquant si le message m passe à travers l'arc (i, j) pendant le créneau de temps t . La variable binaire $x_{m,i,t}$ symbolise si le message m est présent dans le nœud i au début du créneau t . $T_{\max}$ est la latence maximale dans l'intervalle de temps autorisé. L_m est le créneau d'arrivée du message m . w_i modélise le coût énergétique pour l'envoi d'un message depuis le nœud i . $A_{i,j,t}$ désigne la présence d'un message dans l'arc (i, j) pendant le créneau t . $X_{i,t}^+$ indique si un message est envoyé par le nœud i pendant le créneau t . $X_{i,t}^-$ indique si un message est envoyé au nœud i pendant le créneau t . L'énergie maximale autorisée est représentée par la constante E . Elle nous permet de borner l'énergie dépensée lors de la collecte des informations afin de contrôler sa consommation. Le nœud 0 représente la station de base du réseau de capteurs sans fil. L'ensemble V^* représente l'ensemble V privée de la station de base.

La fonction objectif est de minimiser la latence maximale L . Pour définir le problème que nous voulons résoudre, nous devons définir des contraintes sur l'énergie, le flux, les interférences et la latence.

La contrainte énergétique est représentée par l'équation 5.1. Elle permet de compter l'énergie dépensée pour envoyer tous les messages générés et transmis à la station de base. Nous ne considérons que l'énergie dépensée pour la transmission. L'énergie dépensée pour la réception ne sera pas prise en compte.

$$\sum_{m \in V^*} \sum_{(i,j) \in E} \sum_{t \in [1; T_{\max}]} w_i \times y_{m,i,j,t} \leq E \quad (5.1)$$

La première contrainte portant sur le flux permet l'initialisation de la variable $x_{m,i,1}$ au créneau 1. Un message m est généré au nœud i lorsque m est égal à i et i n'est pas la station de base. Cette contrainte est illustrée par l'inégalité 5.2.

$$\forall m \in V^*, \forall i \in V, \quad x_{m,i,1} = \begin{cases} 1 & \text{si } m = i \text{ et } i \neq 0 \\ 0 & \text{sinon.} \end{cases} \quad (5.2)$$

La seconde contrainte de flux représente le routage du message m au début du créneau $t + 1$ en fonction de la réception ou de l'envoi dudit message au créneau t . Cette contrainte est représentée par l'inégalité 5.3.

$$\begin{aligned} & \forall m \in V^*, \forall i \in V, \forall t \in [1; T_{\max}], \\ x_{m,i,t+1} &= x_{m,i,t} + \sum_{j \in N^-(i)} y_{m,j,i,t} - \sum_{j \in N^+(i)} y_{m,i,j,t} \end{aligned} \quad (5.3)$$

La troisième contrainte sur le flux est donnée par l'inégalité 5.4. Elle assure que tous les messages m arrivent à la station de base (nœud 0) au maximum de temps autorisé $T_{\max}$.

$$\forall m \in V^*, \quad \sum_{t \in [1; T_{\max}]} x_{m,0,t} \geq 1 \quad (5.4)$$

L'équation 5.5 modélise qu'un message m peut être sur un seul nœud i à un instant donné t .

$$\forall m \in V^*, \forall t \in [1; T_{\max}], \quad \sum_{i \in V} x_{m,i,t} = 1 \quad (5.5)$$

L'équation 5.6 définit la valeur de la variable $A_{i,j,t}$ en fonction des variables $y_{m,i,j,t}$.

$$\forall (i, j) \in E, \forall t \in [1; T_{\max}], \quad A_{i,j,t} = \sum_{m \in V^*} y_{m,i,j,t} \quad (5.6)$$

Les équations 5.7 et 5.8 sont utilisées pour définir les variables $X_{i,t}^-$ et $X_{i,t}^+$ selon la variable $A_{i,j,t}$.

$$\forall i \in V, \forall t \in [1; T_{\max}], X_{i,t}^- = \sum_{j \in N^-(i)} A_{j,i,t} \quad (5.7)$$

$$\forall i \in V, \forall t \in [1; T_{\max}], X_{i,t}^+ = \sum_{j \in N^+(i)} A_{i,j,t} \quad (5.8)$$

L'équation 5.9 garantit que tout nœud i peut soit transmettre soit recevoir pendant un intervalle de temps donné t . Cette équation permet la modélisation de la collision primaire.

$$\forall i \in V, \forall t \in [1; T_{\max}], X_{i,t}^+ + X_{i,t}^- \leq 1 \quad (5.9)$$

L'équation 5.10 garantit que s'il y a un flux dans l'arc (i, j) pendant le créneau de temps t alors aucun père du nœud i ne peut envoyer de messages pendant cet intervalle de temps. De même, l'équation 5.11 garantit qu'aucun père du nœud j , à l'exception du nœud i , ne peut envoyer de messages. La collision secondaire est modélisée par les inégalités 5.10 et 5.11.

$$\forall (i, j) \in E, \forall t \in [1; T_{\max}], \forall k \in N^-(i), \\ A_{i,j,t} + X_{k,t}^+ \leq 1 \quad (5.10)$$

$$\forall (i, j) \in E, \forall k \in N^-(j) \setminus \{i\}, \forall t \in [1; T_{\max}], \\ A_{i,j,t} + X_{k,t}^+ \leq 1 \quad (5.11)$$

Pour finir, nous définissons deux contraintes sur la latence. La première, modélisée par l'équation 5.12, définit la latence L_m pour tout message m du réseau.

$$\forall m \in V^*, L_m = \sum_{j \in N^-(0)} \sum_{t \in [1; T_{\max}]} y_{m,j,0,t} \times t \quad (5.12)$$

La seconde borne toutes les latences L_m par la latence maximale autorisée L . Cette contrainte est définie par l'inégalité 5.13.

$$\forall m \in V^*, L_m \leq L \quad (5.13)$$

Ce programme linéaire est constitué de $O(|E| \times |\Delta| \times T_{\max})$ contraintes et de $O(|V| \times |E| \times T_{\max})$ variables. La constante Δ représente le nombre maximal de voisins d'un nœud du réseau de capteurs.

5.4/ ALGORITHME IDEG-ROUTING&MAC

Dans ce travail, nous nous intéressons aux approches centralisées combinant les couches MAC et Liaison de données en une seule couche, nous détaillons donc le concept d'IDeg-Routing&MAC [82] afin de comparer ses performances par rapport à la solution optimale.

Définition de l'algorithme

Le concept de IDeg-Routing&MAC est le suivant :

1. Initialement, l'arbre de routage et l'ordonnancement TDMA sont vides et la longueur de la TDMA est égale au degré maximale Δ du graphe, plus un,
2. Un arbre BFS (*Breadth First Search*) est construit en appliquant un parcours en largeur sur le réseau à partir de la station de base,
3. Dans l'arbre BFS, sélectionnez la feuille présentant le degré d'interférence le plus élevé du réseau,
4. Exécuter les étapes suivantes pour la feuille sélectionnée,
 1. Choisir comme prochain saut, son voisin le plus proche de la station de base en termes de distance géographique,
 2. Ajouter le couple (feuille, prochain saut) à l'arbre de routage,
 3. Si cette feuille a déjà été choisie en tant que prochain saut par un autre nœud, trouvez un emplacement compatible dans l'ordonnancement pour cette feuille à partir du dernier emplacement parmi ses emplacements enfants en effectuant une recherche circulaire.
 4. Si cette feuille n'a jamais été choisie comme prochain saut, recherchez un emplacement compatible à partir du début de l'ordonnancement.
 5. Supprimez la feuille courante de l'arbre BFS.
5. Répéter ces étapes depuis la 3 jusqu'à ce que l'arbre BFS soit vide.

IDeg-Routing&MAC construit une solution pour le routage géographique ainsi qu'un ordonnancement TDMA.

Dans l'heuristique, le résultat du routage est modélisé par un arbre permettant d'acheminer les données vers la station de base (*inconnected tree*). Le résultat de l'ordonnancement TDMA est modélisé par une matrice dont les lignes correspondent aux nœuds et les colonnes au créneaux. Dans cette matrice, chaque nœud apparaît une seule fois. Par conséquent, d'une manière générale, pour que tous les messages atteignent la station de base, il faudra répéter cette matrice (à l'exception, par exemple, du cas du réseau en étoile).

5.5/ EXEMPLE EXPLICATIF

Afin d'illustrer les résultats donnés par le PLNE défini ci-dessus et l'heuristique IDeg-Routing&MAC, nous considérons le réseau représenté dans la figure 5.2.

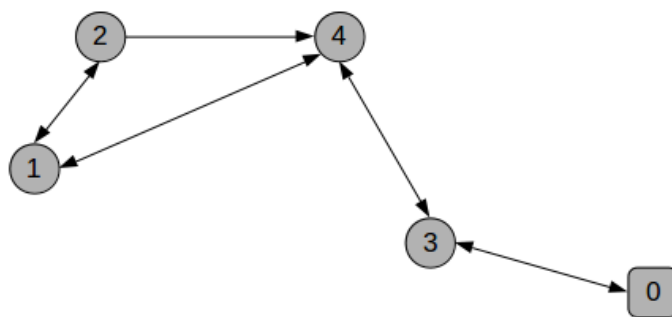


FIGURE 5.2 – Exemple de réseau de capteurs sans fils

Ce réseau est constitué de cinq nœuds numérotés de 0 à 4. Le nœud 0 représente la station de base. Nous considérons que les nœuds 1, 2, 3 et 4 génèrent un message avant t_1 .

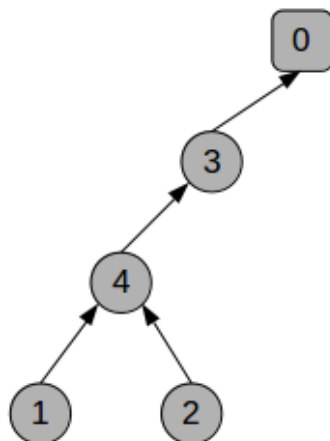


FIGURE 5.3 – Arbre de routage obtenu par IDeg-Routing&MAC pour le réseau de la figure 5.2

TABLE 5.1 – Ordonnancement obtenu par IDeg-Routing&MAC pour le réseau de la figure 5.2

	t1	t2	t3	t4	t5	t6	t7	t8	t9	t10	t11	t12	t13	t14	t15	t16
Node 0				3				3				3				3
Node 1	1				1				1				1			
Node 2		2				2				2				2		
Node 3			4	3			4	3			4	3			4	3
Node 4	1	2	4		1	2	4		1	2	4		1	2	4	

La figure 5.3 représente l'arbre de routage obtenu en appliquant l'heuristique IDeg-Routing&MAC sur le réseau considéré. Le tableau 5.1 représente l'ordonnancement obtenu par cette même heuristique.

La matrice *Schedule* obtenue par le programme linéaire en nombres entiers est donnée dans le tableau 5.2 et l'arbre de routage associé est représenté dans la figure 5.4.

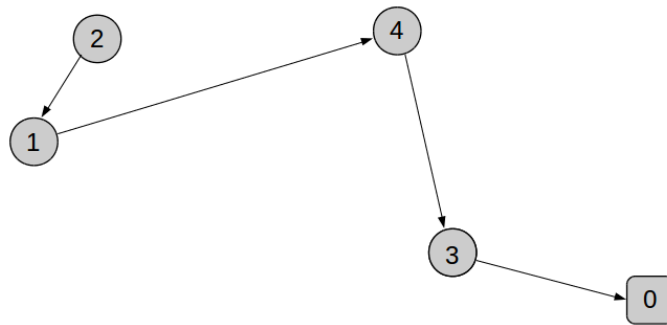


FIGURE 5.4 – Arbre de routage obtenu par le PLNE pour le réseau de la figure 5.2

TABLE 5.2 – Exemple d'ordonnancement et routage obtenus par le PLNE pour le réseau de la figure 5.2

	t1	t2	t3	t4	t5	t6	t7	t8	t9
Node 0		3				3		3	3
Node 1		2	1	1					
Node 2		2							
Node 3	4	3			4	3	4	3	3
Node 4	4		1	1	4		4		

Dans cet exemple, si chaque nœud du réseau a un paquet à envoyer avant t_1 , sachant qu'il ne peut envoyer qu'un seul paquet dans son créneau de transmission, la latence obtenue par IDeg-Routing&MAC pour que tous les paquets arrivent à la station de base est de 16 créneaux de temps. Le PLNE n'a besoin que de 9 intervalles de temps pour obtenir le même résultat.

Ceci est dû au fait que l'heuristique IDeg-Routing&MAC n'autorise pas l'attribution de plusieurs créneaux à un nœud donné et que l'heuristique considère que deux nœuds peuvent être en interférence s'ils sont voisins (entrant ou sortant) à un ou deux sauts. Par exemple, dans l'ordonnancement obtenu par IDeg-Routing&MAC, représenté dans le tableau 5.1, nous pouvons voir que les nœuds 2 et 3 ne peuvent pas avoir le même créneau de transmission dans IDeg-Routing&MAC tandis que le PLNE leur permet d'utiliser le même créneau de temps pour envoyer des messages à condition que la destination de ceux-ci n'interfèrent pas entre eux.

Nous pouvons remarquer que les nœuds 1, 2 et 4 ne sont pas des voisins entrants de la station de base. Donc, ils ne peuvent pas lui envoyer directement leur message. Ils doivent, par conséquent, envoyer les messages au nœud 3 pour que celui-ci les relaye à la station de base. C'est pourquoi le PLNE attribue trois intervalles de temps au nœud 4 et quatre intervalles de temps au nœud 3 pour transmettre ces messages. En effet, le nœud 4 transmet son message et relaye les messages des nœuds 1 et 2 et le nœud 3 transmet son message et relaye les messages des nœuds 1, 2 et 4.

5.6/ SIMULATIONS

Dans cette section, nous présentons les simulations effectuées pour comparer notre PLNE et IDeg-Routing&MAC.

5.6.1/ DESCRIPTION

Nous considérons des graphes avec N nœuds dont la portée de transmission appartient à l'intervalle $[r_{\min}; r_{\max}]$. Les positions des nœuds sont calculées de manière aléatoire et uniforme dans un carré de taille L .

Comme décrit dans le chapitre 4 section 4.4.1, dans notre cas la densité δ se calcule de la manière suivante :

$$\delta = \pi \left(\frac{n \times \mathbb{E}[r^2]}{L^2} \right)$$

Où $\mathbb{E}[r^2]$ se calcule à l'aide de la formule suivante :

$$\mathbb{E}[r^2] = \frac{r_{\max}^2 + r_{\max} \times r_{\min} + r_{\min}^2}{3}$$

Pour les simulations, la densité et la portée de transmission sont fixes. Ainsi, il est possible de déduire la valeur de L . Comme indiqué précédemment, le modèle de connectivité utilisé est le graphe de disque. Plus précisément, si r_i est dans la portée de transmission du nœud i et que d_{ij} est la distance euclidienne entre le nœud i et le nœud j alors le nœud j est un voisin sortant de i si $d_{ij} \leq r_i$.

L'énergie dépensée par un nœud u lors de la transmission est calculée de la manière suivante : $w_u = r_u^2$.

Dans le scénario présenté dans la figure 5.5 et 5.6, le nombre de nœuds a été fixé à 15 et la densité évolue de 3 à 9. Nous considérons que la densité prend en compte à la fois les voisins entrant et sortant d'un nœud. Les portées de transmission $r_{\min}$ et $r_{\max}$ sont fixées respectivement à 20 et 40. Nous n'utilisons que 15 nœuds dans les simulations car les temps d'exécution du programme linéaire augmente de façon exponentielle tandis que les temps d'exécution de l'heuristique IDeg-Routing&MAC est négligeable. Pour le scénario considéré, 70 réseaux aléatoires sont générés par densité.

Le modèle du problème traité étant plus grand, nous avons décidé d'utiliser le solveur Gurobi Optimizer au lieu de GLPK car il est plus efficace dû au fait qu'il est capable de paralléliser les calculs et possède des meilleures heuristiques pour explorer l'espace des solutions. L'énergie prise en compte dans les simulations est l'énergie consommée lors de la transmission des messages. L'énergie requise pour construire l'ordonnancement TDMA et l'arbre de routage n'est pas prise en compte, nous supposons que ce calcul est centralisé sur une ressource puissante.

La figure 5.5 montre l'évolution de la latence par rapport à la densité du réseau.

La figure 5.6 illustre l'évolution du rapport de l'énergie dépensée pour acheminer tous les messages vers la station de base et l'énergie totale du réseau par rapport à la densité du réseau.

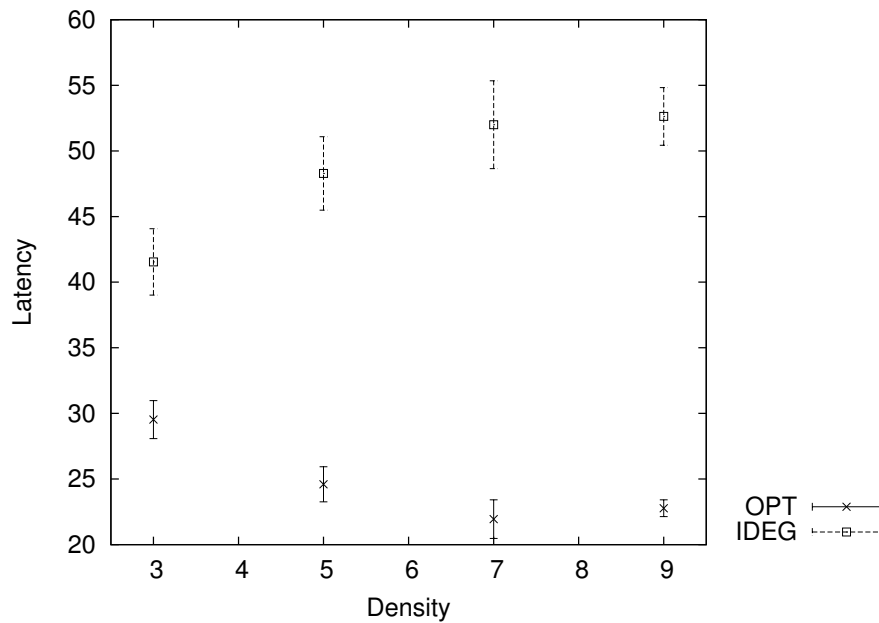


FIGURE 5.5 – Latence de la solution optimale (OPT) et IDeg-Routing& MAC (IDEG) en fonction de la densité des réseaux

5.6.2/ RÉSULTATS

Dans la figure 5.5, nous observons que la latence moyenne obtenue par notre PLNE est de 41% (densité égale à 3) à 153% (densité égale à 9) inférieure à la latence obtenue par IDeg-Routing&MAC.

En outre, nous remarquons que la latence moyenne obtenue par le PLNE diminue à mesure que la densité augmente. Ce résultat est logique car lorsque la densité augmente, le nombre de sauts entre les nœuds et la station de base diminue.

Comparé à IDeg-Routing&MAC, le PLNE donne de meilleures performances car il permet d'ordonnancer le même nœud plusieurs fois sur des intervalles de temps consécutifs. C'est intéressant, en particulier, pour les nœuds proches de la station de base car ils doivent relayer les messages provenant du reste du réseau. Alors que IDeg-Routing&MAC n'autorise pas l'attribution de plusieurs créneaux à chaque nœud, étant donné que le TDMA sera utilisé de manière circulaire. Cela augmente la latence pour IDeg-Routing&MAC car certains nœuds doivent attendre le tour suivant de la TDMA pour pouvoir transmettre.

Dans la figure 5.6, nous constatons que l'énergie dépensée par IDeg-Routing&MAC est, en moyenne, inférieure à celle consommée par le PLNE. Cependant, comme les intervalles de confiance se chevauchent, la différence n'est pas significative. De plus, nous observons que le coût énergétique du routage des messages vers la station de base diminue avec la densité. La raison est la même qu'avant, le nombre de sauts diminue avec l'augmentation de la densité.

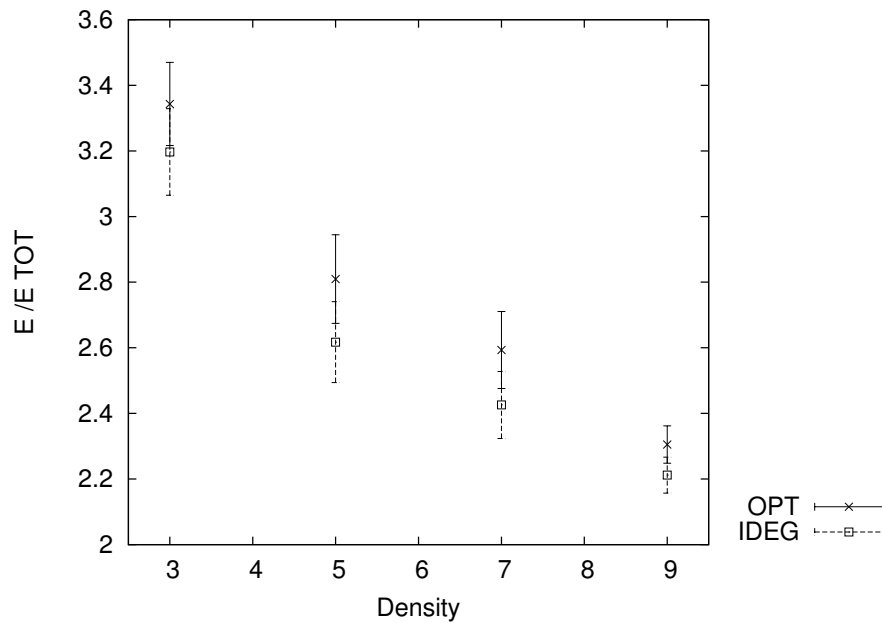


FIGURE 5.6 – Consommation énergétique de la solution optimale (OPT) et IDeg-Routing&MAC (IDEG) en fonction de la densité des réseaux

5.6.3/ DISCUSSION

Le programme linéaire, tel qu'il est conçu, permet de borner la consommation énergétique par une constante E prise en paramètre. Étant donné que les simulations montrent que le programme linéaire calcule une solution qui est meilleure en termes de nombre de créneaux mais qui coûte plus cher du point de vue énergétique, un test intéressant pourrait être d'attribuer comme valeur à E , le coût énergétique trouvé par l'algorithme IDeg-Routing&MAC. De cette manière, nous vérifions l'optimalité de la longueur de l'ordonnancement trouvé pour ce coût énergétique.

L'objectif de l'heuristique IDeg-Routing&MAC, ainsi que du programme linéaire que nous proposons, est de trouver un ordonnancement TDMA et un routage afin d'effectuer une récolte des données le plus rapidement possible tout en minimisant/bornant le coût énergétique.

Une manière pour optimiser l'heuristique IDeg-Routing&MAC, pour son objectif final, est de construire un ordonnancement unique débutant dès le début de l'envoi des messages jusqu'à la réception de ceux-ci par la station de base. Contrairement à ce qui est fait par IDeg-Routing&MAC qui construit un ordonnancement qui doit être répété jusqu'à ce que tous les messages soient reçus par la station de base. Le fait de répéter l'ordonnancement augmente le nombre de créneaux car certains intervalles de temps sont alloués à des nœuds n'ayant plus de messages à transmettre.

Par exemple en supprimant tous les créneaux attribués mais non utilisés, nous réduisons le nombre de créneaux comme le montre le tableau 5.3. Après suppression, le nombre de créneaux nécessaires pour récolter les données est de 9. Il y avait, donc, 7 créneaux non utilisés. Cela correspond à la même taille trouvée par le programme linéaire. Une perspective de notre travail pourrait être le développement d'un algorithme capable d'effectuer un post-traitement sur l'ordonnancement obtenu par IDeg-

TABLE 5.3 – Amélioration de l'ordonnement obtenu par IDeg-Routing&MAC pour le réseau de la figure 5.2

	t1	t2	t3	t4	t5	t6	t7	t8	t9	t10	t11	t12	t13	t14	t15	t16
Node 0				3				3				3				3
Node 1	1				1				1				1			
Node 2		2				2				2				2		
Node 3			4	3			4	3			4	3			4	3
Node 4	1	2	4		1	2	4		1	2	4		1	2	4	

Routing&MAC afin de supprimer tous les créneaux non utilisés.

5.7/ SYNTHÈSE

Ce chapitre présente un état de l'art pour les méthodes cross-layer entre les couches MAC et Liaison de données. D'après l'état de l'art proposé, nous constatons que pour les deux couches que nous avons choisies d'étudier les différents types de cross-layering ont été abordés. Toutefois, les modèles de connectivité utilisés pour modéliser le réseau considèrent que les capteurs sont homogènes. De plus, tous ces travaux proposent des heuristiques donc les différentes propositions sont évaluées de manière empirique entre elles. Par conséquent, nous ne connaissons pas l'écart des solutions trouvées par rapport à une solution optimale et donc la qualité de celles-ci. Pour cela nous proposons un programme linéaire afin de calculer une solution optimale permettant ainsi d'un point d'évaluation. Nous considérons des réseaux de capteurs sans fil hétérogènes où les capteurs peuvent avoir différentes portées de transmission. L'évaluation d'une heuristique existante, nommée IDeg-Routing&MAC, est réalisée via des simulations. Les résultats de la simulation montrent que notre proposition génère toujours une latence inférieure à celle de l'heuristique IDeg-Routing&MAC. Cependant, en moyenne, IDeg-Routing&MAC consomme moins d'énergie que notre proposition bien que l'écart soit négligeable.

CONCLUSION

RÉSULTATS

Cette thèse se situe dans le domaine des réseaux de capteurs, et plus particulièrement dans l'optimisation des communications au sein de ceux-ci. Dans un premier temps nous nous sommes intéressés à la diffusion, opération permettant l'exploitation dynamique d'un réseau interrogeable. Puis, nous avons abordé le routage, opération indispensable dans le cas de tous les réseaux de capteurs afin de remplir leur rôle : la récolte des informations monitorées.

Dans les deux cas, nous avons visé l'optimisation de deux métriques. De plus, nous avons choisi de considérer la présence de capteurs ayant des portées de transmission différentes car il s'agit d'un cas plus générique. Nous considérons, donc, un réseau de capteurs hétérogènes. La première métrique considérée est l'énergie qui est une contrainte liée aux capteurs. En minimisant l'énergie nous augmentons la durée de vie du réseau de capteurs. La deuxième métrique constitue un aspect qualitatif du réseau : il s'agit de la latence.

Nous avons commencé par présenter les réseaux de capteurs, leurs caractéristiques, les problèmes d'optimisation que nous avons choisi d'aborder ainsi que les différentes méthodes possibles pour les résoudre.

Puis, nous avons présenté et classifié les travaux de la littérature suivant les métriques optimisées puis suivant le modèle utilisé pour représenter le réseau de capteurs. Nous avons choisi de nous concentrer sur le modèle de graphe de disques car il nous permet de considérer la présence de capteurs de portées différentes. De plus, nous avons pu remarquer le manque d'une proposition traitant à la fois l'optimisation de l'énergie et la latence. L'objectif de nos contributions est de pallier ce manque. Nos travaux se divisent en deux suivant l'opération de communication abordée. En premier lieu, nous avons abordé la diffusion. Pour cette opération, nous avons proposé un programme linéaire en nombres entiers permettant de calculer une solution optimale. Cette proposition fournit un point de comparaison et donc d'évaluation pour les autres méthodes abordant le même problème. En particulier, cela est très utile pour les heuristiques dont on ne connaît pas la distance à l'optimale et que par conséquent on évalue uniquement de manière expérimentale. Nous avons également proposé l'algorithme d'approximation *BB* dont la borne est de $\Omega \times h_{\max}$. Tel qu'il est conçu l'algorithme donne des meilleures performances dans le cas des graphes de moyenne à haute densité. Pour finir, nous avons proposé l'utilisation du recuit simulé pour la résolution de notre problème et une version améliorée de *BB*.

D'après l'état de l'art du routage multi-couche entre la couche MAC et la couche liaison de données, les deux types de routages ont été abordés, à savoir le cross-layer inter couche et par fusion de couche. Toutefois, le modèle utilisé était celui des graphes de disques unitaires et l'optimisation de deux métriques à la fois n'était pas abordée. De plus, les propositions trouvées dans la littérature étaient des heuristiques. Nous avons

par conséquent proposé un programme linéaire en nombre entiers afin de pouvoir comparer les heuristiques avec la solution optimale. Notre solution considère un cas plus générique : le réseau de capteurs se compose de capteurs pouvant avoir des portées différentes. De plus elle permet de minimiser la latence tout en bornant le coût énergétique.

PERSPECTIVES

Après avoir résumé nos travaux, nous présentons les possibles perspectives de notre travail.

Équilibrage en charge

Une perspectives intéressante pourrait être la construction d'une dorsale équilibrée en charge dans le cas des réseaux de capteurs hétérogènes. En effet, nous n'avons pas trouvé de travaux sur l'équilibrage de la charge dans le cas des graphes de disques. Un équilibre dans la distribution de la charge lors du routage est très important afin d'épuiser les réserves énergétiques de manière homogène et ainsi prolonger la durée de vie du réseau.

Modélisation de l'énergie

Lors du calcul du coût nécessaire à la collecte ou à la diffusion de données, il serait plus intéressant de tenir compte de l'énergie résiduelle lors de la construction des chemins. En effet, considérer seulement l'énergie dépensée conduit à choisir à chaque fois le chemin le moins coûteux. Cela entraîne un épuisement de l'énergie des nœuds constituant ce chemin alors que les autres ne consomment presque pas d'énergie. Une manière de considérer à la fois le coût énergétique et l'énergie résiduelle des nœuds serait de pondérer les nœuds avec le ratio entre l'énergie dépensée pour la transmission et leur énergie résiduelle, puis calculer le plus court chemin pondéré. Une autre amélioration possible, pour ce qui est de la modélisation de l'énergie, serait de tenir compte également du coût de l'énergie de réception. Cette modification permettrait d'avoir un modèle plus proche de la réalité.

Multiples stations de base

Une possible perspective serait d'adapter notre travail à la présence de multiple stations de base. La présence de plusieurs stations de base permettrait d'améliorer les performances pour les métriques que nous avons choisies d'optimiser, à savoir l'énergie et la latence. En effet, si les stations sont distribuées de manière homogène dans le réseau, cela diminuera la distance entre les nœuds et les stations de base. En plus cela entraînera une sollicitation moindre des nœuds autour de la station de base car ils seront responsables de la retransmission des messages, non plus de l'ensemble du réseau mais seulement d'un sous-ensemble.

Mobilité des nœuds

Lorsqu'un véhicule détecte un accident il doit effectuer une diffusion afin de communiquer aux autres véhicules la présence de celui-ci. Nos travaux sur la diffusion pourraient s'adapter à cette application en ajoutant la prise en compte de la mobilité. Pour qu'une application de ce type fonctionne, nous avons besoin à la fois d'une partie fixe et d'une partie mobile. Toute l'autoroute doit être munie de capteurs pouvant détecter un accident et diffuser l'information en cas d'absence de véhicule. C'est ce qui va constituer la partie fixe. La partie mobile s'ajoute dans les sections de l'autoroute dans lesquelles des véhicules sont présents. Nos propositions pour la diffusion répondent aux besoins de la partie fixe. Pour exploiter nos travaux pour ce type d'application, il faudrait ajouter la partie mobile.

Amélioration de l'algorithme IDeg-Routing&MAC

Comme introduit dans la section 5.6.3 du chapitre 5, une perspective de notre travail serait d'implémenter le post-traitement permettant de réduire la taille de l'ordonancement obtenu par l'heuristique IDeg-Routing&MAC. Cela permettrait de réduire la latence et de s'approcher davantage de l'optimale tout en gardant le même coût énergétique de l'heuristique IDeg-Routing&MAC.

LISTE DES PUBLICATIONS PERSONNELLES

ARTICLES DANS REVUES INTERNATIONALES AVEC COMITÉ DE LECTURE RÉPERTORIÉES

- Solving the Minimum Energy Broadcast Backbone with Bounded Hop Distance Problem in Heterogeneous Wireless Sensor Networks
Breschi Jimenez Ramirez Karla, Bernard Julien, Héam Pierre-Cyrille, Ménier Victor, Bourquin Clément
Soumis au Journal of Heuristics le 19 Mars 2018
- A SAW wireless sensor network platform for industrial predictive maintenance
Gombé Bérenger Ossété, Goavec-Merou Gwenhaël, Breschi Jimenez Ramirez Karla, Guyennet Hervé, Friedt Jean-Michel, Felea Violeta, Medjaher Kamal
Journal of Intelligent Manufacturing, jul. 2017
- A self-adaptable distributed CBR version of the EquiVox system
Henriet Julien, Lang Christophe, Muthada pottayya Ronnie, Breschi Jimenez Ramirez Karla
Biomedical Engineering : Applications, Basis and Communications, vol. 28 , 4, n°1650028 (16 pages), sep. 2016

COMMUNICATIONS AVEC ACTES DANS UN CONGRÈS INTERNATIONAL

- Construction of a Minimum Energy Broadcast Backbone with Bounded Delay in Heterogeneous Wireless Sensor Networks
Breschi Jimenez Ramirez Karla, Bernard Julien
IEEE Symposium on Computers and Communications (ISCC), Heraklion, Greece, pp. 1306 - 1311, jul. 2017
- Radiofrequency Transceiver for Probing SAW Sensors and Communicating Through a Wireless Sensor Network
Droit Christophe, Friedt Jean-Michel, Goavec-Merou Gwenhaël, Martin Gilles, Balandras Sylvain, Breschi Jimenez Ramirez Karla, Bernard Julien, Guyennet Hervé
The Sixth International Conference on Sensor Technologies and Applications (SENSORCOMM), Roma, Italy, aug. 2012

BIBLIOGRAPHIE

- [1] MILLER, C. E., TUCKER, A. W., AND ZEMLIN, R. A. **Integer programming formulation of traveling salesman problems.** *Journal of the ACM (JACM)* 7, 4 (1960), 326–329.
- [2] KIRKPATRICK, S., GELATT, C. D., AND VECCHI, M. P. **Optimization by simulated annealing.** *science* 220, 4598 (1983), 671–680.
- [3] CERNY, V. **Minimization of continuous functions by simulated annealing.** Tech. rep., 1984.
- [4] GAREY, M. R., AND JOHNSON, D. S. **Computers and Intractability ; A Guide to the Theory of NP-Completeness.** W. H. Freeman & Co., New York, NY, USA, 1990.
- [5] KLEIN, P., AND RAVI, R. **A nearly best-possible approximation algorithm for node-weighted steiner trees.** *Journal of Algorithms* 19, 1 (1995), 104–115.
- [6] GUHA, S., AND KHULLER, S. **Approximation algorithms for connected dominating sets.** *Algorithmica* (1998).
- [7] HEINZELMAN, W. R., CHANDRAKASAN, A., AND BALAKRISHNAN, H. **Energy-efficient communication protocol for wireless microsensor networks.** In *33rd Annual Hawaii International Conference on System sciences* (2000), IEEE, pp. 10–pp.
- [8] WIESELTHIER, J. E., NGUYEN, G. D., AND EPHREMIDES, A. **On the construction of energy-efficient broadcast and multicast trees in wireless networks.** In *Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)* (2000), vol. 2, IEEE, pp. 585–594.
- [9] FEDERSPIEL, R. A., RABAEY, J., ARENS, E., FEDERSPIEL, C., GADGIL, A., MESSERSCHMITT, D., NAZAROFF, W., PISTER, K., AND OREN, S. **Smart energy distribution and consumption : Information technology as an enabling force.** In *Center for* (2001), Citeseer.
- [10] KRUMME, D. W., AND FRAGOPOULOU, P. **Minimum eccentricity multicast trees.** *Discrete Mathematics & Theoretical Computer Science* 4, 2 (2001), 157–172.
- [11] MAINWARING, A., CULLER, D., POLASTRE, J., SZEWCZYK, R., AND ANDERSON, J. **Wireless sensor networks for habitat monitoring.** In *Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications* (2002), Acm, pp. 88–97.
- [12] CHENG, X., HUANG, X., LI, D., WU, W., AND DU, D.-Z. **A polynomial-time approximation scheme for the minimum-connected dominating set in ad hoc wireless networks.** *Networks* 42, 4 (2003), 202–208.
- [13] FISHKIN, A. V. **Disk graphs : A short survey.** In *International Workshop on Approximation and Online Algorithms* (2003), Springer, pp. 260–264.
- [14] HILL, J. L. **System architecture for wireless sensor networks.** PhD thesis, University of California, Berkeley, 2003.

- [15] CORMEN, T. H., LEISERSON, C. E., RIVEST, R. L., AND STEIN, C. **Introduction à l'algorithmique**. Dunod, Paris, 2004.
- [16] DAI, F., AND WU, J. **An extended localized algorithm for connected dominating set formation in ad hoc wireless networks**. *IEEE transactions on parallel and distributed systems* 15, 10 (2004), 908–920.
- [17] HE, T., KRISHNAMURTHY, S., STANKOVIC, J. A., ABDELZAHER, T., LUO, L., STOLERU, R., YAN, T., GU, L., HUI, J., AND KROGH, B. **Energy-efficient surveillance system using wireless sensor networks**. In *Proceedings of the 2nd international conference on Mobile systems, applications, and services* (2004), ACM, pp. 270–283.
- [18] LI, D., JIA, X., AND LIU, H. **Energy efficient broadcast routing in static ad hoc wireless networks**. *Transactions on Mobile Computing* 3, 2 (2004), 144–151.
- [19] RUAN, L., DU, H., JIA, X., WU, W., LI, Y., AND KO, K.-I. **A greedy approximation for minimum connected dominating sets**. *Theoretical Computer Science* 329, 1 (2004), 325–330.
- [20] WAN, P.-J., ALZOUBI, K. M., AND FRIEDER, O. **Distributed construction of connected dominating set in wireless ad hoc networks**. *Mobile Networks and Applications* 9, 2 (2004), 141–149.
- [21] ARAMPATZIS, T., LYGEROS, J., AND MANESIS, S. **A survey of applications of wireless sensors and wireless sensor networks**. In *Intelligent Control, 2005. Proceedings of the 2005 IEEE International Symposium on, Mediterrean Conference on Control and Automation* (2005), IEEE, pp. 719–724.
- [22] CHEN, D., DENG, J., AND VARSHNEY, P. K. **A state-free data delivery protocol for multihop wireless sensor networks**. In *Wireless Communications and Networking Conference* (2005), vol. 3, IEEE, pp. 1818–1823.
- [23] KRISHNAMACHARI, B. **Networking wireless sensors**. Cambridge University Press, 2005.
- [24] LI, Y., THAI, M. T., WANG, F., YI, C.-W., WAN, P.-J., AND DU, D.-Z. **On greedy construction of connected dominating sets in wireless networks**. *Wireless Communications and Mobile Computing* 5, 8 (2005), 927–932.
- [25] MELODIA, T., VURAN, M. C., AND POMPILI, D. **The state of the art in cross-layer design for wireless sensor networks**. In *International workshop of the EuroNGI network of excellence* (2005), Springer, pp. 78–92.
- [26] MONTEMANNI, R., GAMBARDILLA, L. M., AND DAS, A. K. **The minimum power broadcast problem in wireless networks : a simulated annealing approach**. In *Wireless Communications and Networking Conference* (2005), vol. 4, IEEE, pp. 2057–2062.
- [27] SRIVASTAVA, V., AND MOTANI, M. **Cross-layer design : a survey and the road ahead**. *Communications Magazine* 43, 12 (2005), 112–119.
- [28] DAI, F., AND WU, J. **On constructing k-connected k-dominating set in wireless ad hoc and sensor networks**. *Journal of parallel and distributed computing* 66, 7 (2006), 947–958.
- [29] DU, D.-Z., THAI, M. T., LI, Y., LIU, D., AND ZHU, S. **Strongly connected dominating sets in wireless sensor networks with unidirectional links**. In *Asia-Pacific Web Conference* (2006), Springer, pp. 13–24.

- [30] KULKARNI, S., IYER, A., AND ROSENBERG, C. **An address-light, integrated mac and routing protocol for wireless sensor networks**. *IEEE/ACM Transactions on networking* 14, 4 (2006), 793–806.
- [31] LI, Y., THAI, M. T., WANG, F., AND DU, D.-Z. **On the construction of a strongly connected broadcast arborescence with bounded transmission delay**. *Mobile Computing, IEEE Transactions on* 5, 10 (2006), 1460–1470.
- [32] SCHMID, S., AND WATTENHOFER, R. **Algorithmic models for sensor networks**. In *20th International Parallel and Distributed Processing Symposium (IPDPS)* (2006), IEEE, pp. 11–pp.
- [33] SUH, C., KO, Y.-B., AND SON, D.-M. **An energy efficient cross-layer mac protocol for wireless sensor networks**. In *Asia-Pacific Web Conference* (2006), Springer, pp. 410–419.
- [34] THAI, M. T., AND DU, D.-Z. **Connected dominating sets in disk graphs with bidirectional links**. *Communications Letters, IEEE* 10, 3 (2006), 138–140.
- [35] LI, Y., KIM, D., ZOU, F., AND DU, D.-Z. **Constructing connected dominating sets with bounded diameters in wireless networks**. In *International Conference on Wireless Algorithms, Systems and Applications (WASA)* (2007), IEEE, pp. 89–94.
- [36] PARK, M. A., WILLSON, J., WANG, C., WU, W., AND FARAGO, A. **A dominating and absorbent set in a wireless ad-hoc network with different transmission ranges**. In *Proceedings of the 8th ACM international symposium on Mobile ad hoc networking and computing* (2007), ACM, pp. 22–31.
- [37] THAI, M. T., ZHANG, N., TIWARI, R., AND XU, X. **On approximation algorithms of k-connected m-dominating sets in disk graphs**. *Theoretical Computer Science* 385, 1 (2007), 49–59.
- [38] TIWARI, R., MISHRA, T., LI, Y., AND THAI, M. T. **K-strongly connected m-dominating and absorbing set in wireless ad hoc networks with unidirectional links**. In *Wireless Algorithms, Systems and Applications, 2007. WASA 2007. International Conference on* (2007), IEEE, pp. 103–112.
- [39] AKYILDIZ, I. F., AND WANG, X. **Cross-layer design in wireless mesh networks**. *Transactions on Vehicular Technology* 57, 2 (2008), 1061.
- [40] LI, D., DU, H., WAN, P.-J., GAO, X., ZHANG, Z., AND WU, W. **Minimum power strongly connected dominating sets in wireless networks**. In *ICWN* (2008), pp. 447–451.
- [41] THAI, M. T., TIWARI, R., AND DU, D.-Z. **On construction of virtual backbone in wireless ad hoc networks with unidirectional links**. *Mobile Computing, IEEE Transactions on* 7, 9 (2008), 1098–1109.
- [42] YICK, J., MUKHERJEE, B., AND GHOSAL, D. **Wireless sensor network survey**. *Computer networks* 52, 12 (2008), 2292–2330.
- [43] ANASTASI, G., CONTI, M., FRANCESCO, M. D., AND PASSARELLA, A. **Energy conservation in wireless sensor networks : A survey**. *Ad Hoc Networks* 7, 3 (2009), 537 – 568.
- [44] AWANG, A., LAGRANGE, X., AND ROS, D. **A cross-layer medium access control and routing protocol for wireless sensor networks**. *10èmes Journées Doctorales en Informatique et Réseaux* (2009), 2–4.

- [45] HUANG, Y., GAO, X., ZHANG, Z., AND WU, W. **A better constant-factor approximation for weighted dominating set in unit disk graph.** *Journal of Combinatorial Optimization* 18, 2 (2009), 179–194.
- [46] ISSARIYAKUL, T., AND HOSSAIN, E. **Introduction to Network Simulator 2 (NS2).** Springer US, Boston, MA, 2009, pp. 1–18.
- [47] KIM, D., WU, Y., LI, Y., ZOU, F., AND DU, D.-Z. **Constructing minimum connected dominating sets with bounded diameters in wireless networks.** *Transactions on Parallel and Distributed Systems* 20, 2 (2009), 147–157.
- [48] LI, D., DU, H., WAN, P.-J., GAO, X., ZHANG, Z., AND WU, W. **Construction of strongly connected dominating sets in asymmetric multihop wireless networks.** *Theoretical Computer Science* 410, 8 (2009), 661–669.
- [49] PERVEZ KHAN, M., HUSSAIN, A., AND KWAK, K. S. **Medical applications of wireless body area networks.** *International Journal of Digital Content Technology and its Applications* 3, 3 (2009), 185–93.
- [50] WAN, P.-J., HUANG, S. C.-H., WANG, L., WAN, Z., AND JIA, X. **Minimum-latency aggregation scheduling in multihop wireless networks.** In *Proceedings of the tenth ACM international symposium on Mobile ad hoc networking and computing* (2009), ACM, pp. 185–194.
- [51] ZHENG, J., AND JAMALIPOUR, A. **Wireless sensor networks : a networking perspective.** John Wiley & Sons, 2009.
- [52] CORKE, P., WARK, T., JURDAK, R., HU, W., VALENCIA, P., AND MOORE, D. **Environmental wireless sensor networks.** *Proceedings of the IEEE* 98, 11 (2010), 1903–1917.
- [53] WIGHTMAN, P., FABREGASY, A., AND LABRADORZ, M. **An optimal solution to the mcdds problem for topology construction in wireless sensor networks.** In *Communications (LATINCOM), 2010 IEEE Latin-American Conference on* (Sept 2010), pp. 1–6.
- [54] BEN-OTHTMAN, J., BESSAOUD, K., BUI, A., AND PILARD, L. **Self-stabilizing algorithm for energy saving in wireless sensor networks.** In *Symposium on Computers and Communications (ISCC)* (2011), IEEE, pp. 68–73.
- [55] FEDOR, S., COLLIER, M., AND SREENAN, C. J. **Crosslayer routing and time synchronisation in wireless sensor networks.** *International Journal of Sensor Networks* 10, 3 (2011), 143–159.
- [56] MENDES, L. D., AND RODRIGUES, J. J. **A survey on cross-layer solutions for wireless sensor networks.** *Journal of Network and Computer Applications* 34, 2 (2011), 523–534.
- [57] ALKHATIB, A. A. A., AND BAICHER, G. S. **Wireless sensor network architecture.** In *International Conference on Computer networks and Communication Systems (ICNCS 2012)* (2012), vol. 35, pp. 11–15.
- [58] FAN, N., AND WATSON, J.-P. **Solving the connected dominating set problem and power dominating set problem by integer programming.** In *International Conference on Combinatorial Optimization and Applications* (2012), Springer, pp. 371–383.
- [59] HANCKE, G., SILVA, B., HANCKE JR, G., AND OTHERS. **The role of advanced sensing in smart cities.** *Sensors* 13, 1 (2012), 393–425.

- [60] KIM, J., AND LEE, J.-H. **Vitamin : A virtual backbone tree algorithm for minimal energy consumption in wireless sensor network routing.** In *Information Networking (ICOIN), 2012 International Conference on* (Feb 2012), pp. 144–149.
- [61] LANDSIEDEL, O., GHADIMI, E., DUQUENNOY, S., AND JOHANSSON, M. **Low power, low delay : opportunistic routing meets duty cycling.** In *Proceedings of the 11th international conference on Information Processing in Sensor Networks* (2012), ACM, pp. 185–196.
- [62] MAMUN, Q. **A qualitative comparison of different logical topologies for wireless sensor networks.** *Sensors* 12, 11 (2012), 14887–14913.
- [63] TIWARI, R., AND THAI, M. **On enhancing fault tolerance of virtual backbone in a wireless sensor network with unidirectional links.** In *Sensors : Theory, Algorithms, and Applications*, V. L. Boginski, C. W. Commander, P. M. Pardalos, and Y. Ye, Eds., vol. 61 of *Springer Optimization and Its Applications*. Springer New York, 2012, pp. 3–18.
- [64] ZHANG, H., BAI, G., AND LIU, C. **A broadcast path choice algorithm based on simulated annealing for wireless sensor network.** In *International Conference on Automation and Logistics (ICAL)* (2012), IEEE, pp. 310–314.
- [65] HE, J. S., JI, S., PAN, Y., AND CAI, Z. **Approximation algorithms for load-balanced virtual backbone construction in wireless sensor networks.** *Theoretical Computer Science* 507 (2013), 2–16.
- [66] SCHAUDT, O. **On dominating sets whose induced subgraphs have a bounded diameter.** *Discrete Applied Mathematics* 161, 16 (2013), 2647–2652.
- [67] SOUA, R., LIVOLANT, E., AND MINET, P. **Musika : A multichannel multi-sink data gathering algorithm in wireless sensor networks.** In *9th International Wireless Communications and Mobile Computing Conference (IWCMC)* (2013), IEEE, pp. 1370–1375.
- [68] TORKESTANI, J. A. **Energy-efficient backbone formation in wireless sensor networks.** *Computers & Electrical Engineering* 39, 6 (2013), 1800–1811.
- [69] XENAKIS, A., FOUKALAS, F., STAMOULIS, G., AND KHATTAB, T. **Energy-aware joint power, packet and topology optimization by simulated annealing for wsns.** In *7th Conference and Exhibition (GCC)* (2013), IEEE, pp. 17–21.
- [70] YU, J., WANG, N., WANG, G., AND YU, D. **Connected dominating sets in wireless ad hoc and sensor networks—a comprehensive survey.** *Computer Communications* 36, 2 (2013), 121–134.
- [71] BUCHANAN, A., SUNG, J. S., BOGINSKI, V., AND BUTENKO, S. **On connected dominating sets of restricted diameter.** *European Journal of Operational Research* 236, 2 (2014), 410–418.
- [72] HE, J. S., JI, S., PAN, Y., AND LI, Y. **Greedy construction of load-balanced virtual backbones in wireless sensor networks.** *Wireless Communications and Mobile Computing* 14, 7 (2014), 673–688.
- [73] PAL, P. K., AND CHATTERJEE, P. **A survey on tdma-based mac protocols for wireless sensor network.** *International Journal of Emerging Technology and Advanced Engineering* 4, 6 (2014), 219–230.
- [74] SARAVANAN, M., AND MADHESWARAN, M. **A hybrid optimized weighted minimum spanning tree for the shortest intrapath selection in wireless sensor network.** *Mathematical Problems in Engineering* 2014 (2014).

- [75] A BUCHANAN, JS SUNG, S. B., AND PASILIAO, E. L. **An integer programming approach for fault-tolerant connected dominating sets.** *INFORMS Journal on Computing* (January 2015).
- [76] AHN, N., AND PARK, S. **An optimization algorithm for the minimum k-connected m-dominating set problem in wireless sensor networks.** *Wireless Networks* 21, 3 (2015), 783–792.
- [77] CHATTERJEE, P., AND DAS, N. **Multiple sink deployment in multi-hop wireless sensor networks to enhance lifetime.** In *Applications and Innovations in Mobile Computing (AIMoC)* (2015), IEEE, pp. 48–54.
- [78] LOUAIL, L., FELEA, V., BERNARD, J., AND GUYENNET, H. **Mac-aware routing in wireless sensor networks.** In *International Black Sea Conference on Communications and Networking (BlackSeaCom)* (2015), IEEE, pp. 225–229.
- [79] SGORA, A., VERGADOS, D. J., AND VERGADOS, D. D. **A survey of tdma scheduling schemes in wireless multihop networks.** *Computing Surveys (CSUR)* 47, 3 (2015), 53.
- [80] IDWAN, S., ZUBAIRI, J. A., AND MAHMOOD, I. **Smart solutions for smart cities : Using wireless sensor network for smart dumpster management.** In *Collaboration Technologies and Systems (CTS), 2016 International Conference on* (2016), IEEE, pp. 493–497.
- [81] LOUAIL, L., AND FELEA, V. **Latency optimization through routing-aware time scheduling protocols for wireless sensor networks.** *Computers & Electrical Engineering* 56 (2016), 418–440.
- [82] LOUAIL, L., AND FELEA, V. **Routing and tdma joint cross-layer design for wireless sensor networks.** In *International Conference on Ad-Hoc Networks and Wireless* (2016), Springer, pp. 111–123.
- [83] RAJARAM, M. L., KOUIGIANOS, E., MOHANTY, S. P., AND SUNDARAVADIVEL, P. **A wireless sensor network simulation framework for structural health monitoring in smart cities.** In *Consumer Electronics-Berlin (ICCE-Berlin), 2016 IEEE 6th International Conference on* (2016), IEEE, pp. 78–82.
- [84] RONG, D. **Wireless Sensor Networks in Smart Cities : The Monitoring of Water Distribution Networks Case.** PhD thesis, KTH Royal Institute of Technology, 2016.
- [85] SHA, C., QIU, J.-M., LI, S.-Y., QIANG, M.-Y., AND WANG, R.-C. **A type of low-latency data gathering method with multi-sink for sensor networks.** *Sensors* 16, 6 (2016), 923.
- [86] BRESCHI, K., AND BERNARD, J. **Construction of a minimum energy broadcast backbone with bounded delay in heterogeneous wireless sensor networks.** In *Symposium on Computers and Communications (ISCC)* (2017), IEEE, pp. 1306–1311.
- [87] KOSPENDA, A. **Wireless sensor network for smart street lighting.** PhD thesis.

TABLE DES FIGURES

1.1	Capteur TelosB (à gauche) et capteur MicaZ (à droite)	6
1.2	Composants d'un capteur	7
1.3	Topologies d'un réseau de capteurs (a) en étoile (b) en maille (c) en arbre .	10
1.4	Exemple 1 de conflit primaire	13
1.5	Exemple 2 de conflit primaire (nœud caché)	13
1.6	Exemple de conflit secondaire (nœud exposé)	13
1.7	Exemple de graphe orienté	15
1.8	Exemple de sous-graphe induit	18
1.9	Exemple de graphe complémentaire	19
1.10	Exemple d'ensemble dominant connexe	23
1.11	Modèle en couche pour les réseaux de capteurs	24
1.12	Cross-layering inter-couche (à gauche) et par fusion de couche (à droite) .	25
2.1	Résumé des travaux proposant la construction d'une dorsale	39
2.2	Résumé des travaux proposant la construction d'une dorsale équilibrée en charge	41
2.3	Résumé des travaux proposant la construction d'une dorsale minimisant l'énergie	43
2.4	Résumé des travaux proposant la construction d'une dorsale minimisant la latence	48
2.5	Résumé des travaux proposant la construction d'une dorsale minimisant la latence	53
2.6	Résumé des travaux proposant la construction d'une dorsale	54
3.1	Réseau de capteur considéré dans l'exemple (à gauche) et exemple de dorsale construite par le PLNE (à droite)	61
3.2	Explication de la mise à zéro de l'énergie dans l'algorithme Bounded Backbone. Dorsale obtenue sans la mise à zéro de l'énergie (à gauche) et dorsale obtenue avec mise à zéro de l'énergie (à droite).	66
4.1	Exemple de construction du graphe G'	74

4.2	Consommation énergétique de la solution optimale (ILP), recuit simulé (SA), heuristique naïve (NAIVE), heuristique Highly Bounded Bakcbone (HBB) et heuristique Bounded Bakcbone (BB) lorsque $h_{\max}$ est égal à ε et lorsque le nombre de nœuds est 25.	78
4.3	Consommation énergétique de la solution optimale (ILP), recuit simulé (SA), heuristique Highly Bounded Bakcbone (HBB) et heuristique Bounded Bakcbone (BB) lorsque $h_{\max}$ est égal à ε et lorsque le nombre de nœuds est 25.	79
4.4	Consommation énergétique du recuit simulé (SA), heuristique naïve (NAIVE), heuristique Highly Bounded Bakcbone (HBB) et heuristique Bounded Bakcbone (BB) lorsque $h_{\max}$ est égal à ε et la densité est égale à 15.	80
4.5	Consommation d'énergie du recuit simulé (SA), heuristique Highly Bounded Bakcbone (HBB) et heuristique Bounded Bakcbone (BB) lorsque $h_{\max}$ est égal à ε et la densité est égale à 15.	81
4.6	Consommation énergétique des heuristiques naïve (NAIVE), Highly Bounded Bakcbone (HBB) et Bounded Bakcbone (BB) lorsque $h_{\max}$ est égal à ε et la densité est égale à 15.	81
4.7	Consommation énergétique des heuristiques Highly Bounded Bakcbone (HBB) et Bounded Bakcbone (BB) lorsque $h_{\max}$ est égal à ε et la densité est égale à 15.	82
4.8	Synthèse des performances des heuristiques en fonction du nombre de nœuds du réseau	83
5.1	Résumé des travaux cross-layer entre couche MAC et Liaison de données	86
5.2	Exemple de réseau de capteurs sans fils	94
5.3	Arbre de routage obtenu par IDeg-Routing&MAC pour le réseau de la figure 5.2	94
5.4	Arbre de routage obtenu par le PLNE pour le réseau de la figure 5.2	95
5.5	Latence de la solution optimale (OPT) et IDeg-Routing& MAC (IDEG) en fonction de la densité des réseaux	97
5.6	Consommation énergétique de la solution optimale (OPT) et IDeg-Routing&MAC (IDEG) en fonction de la densité des réseaux	98

LISTE DES TABLES

1.1	Comparaison des caractéristiques des capteurs	7
1.2	Tableau des notations	20
1.3	Modélisation d'un réseau de capteur via la théorie des graphes	21
3.1	Exemple de programme linéaire obtenu	62
5.1	Ordonnancement obtenu par IDeg-Routing&MAC pour le réseau de la figure 5.2	94
5.2	Exemple d'ordonnancement et routage obtenus par le PLNE pour le réseau de la figure 5.2	95
5.3	Amélioration de l'ordonnancement obtenu par IDeg-Routing&MAC pour le réseau de la figure 5.2	99

LISTE DES DÉFINITIONS

1	Définition : Réseau de capteurs sans fil	8
2	Définition : Degré d'interférence	14
3	Définition : Graphe orienté [15]	15
4	Définition : Sommet adjacent ou voisin	15
5	Définition : Voisinage ouvert	15
6	Définition : Voisinage fermé	16
7	Définition : Degré	16
8	Définition : Degré maximum	16
9	Définition : Degré minimum	16
10	Définition : Chemin	17
11	Définition : Longueur d'un chemin	17
12	Définition : Plus court chemin	17
13	Définition : Excentricité	17
14	Définition : Diamètre	17
15	Définition : Sous-graphe induit	18
16	Définition : Graphe pondéré	18
17	Définition : Poids d'un chemin	18
18	Définition : Graphe connexe	18
19	Définition : Graphe fortement connexe	19
20	Définition : Graphe complémentaire $\overline{G}$	19
21	Définition : Graphe transposé G^T	19
22	Définition : Arbre de Steiner	19
23	Définition : Diffusion	21
24	Définition : Dorsale	21
25	Définition : Ensemble dominant (DS)	22
26	Définition : Ensemble dominant connexe (CDS)	22
27	Définition : Ensemble dominant connexe de taille minimale (MCDS)	22
28	Définition : Ensemble dominant connexe de poids minimal (MWCDs)	22

29	Définition : Stable maximum	33
30	Définition : Ensemble dominant fortement connexe	34
31	Définition : Arborescence de recouvrement avec nombre de nœuds internes minimal	35
32	Définition : Ensemble dominant fortement connexe et absorbant	36
33	Définition : Ensemble absorbant	36
34	Définition : Ensemble dominant connexe et équilibré en charge	40
35	Définition : Ensemble dominant fortement connexe de puissance minimale	42
36	Définition : Minimum Energy Broadcast Routing Problem	42
37	Définition : Graphe Cordal	45
38	Définition : Strongly Connected Broadcast Arborescence with bounded Transmission delay	45
39	Définition : s -club	46
40	Définition : ensemble k -dominant et k -connexe	48
41	Définition : ensemble k -dominant et absorbant et k -fortement connexe	50
42	Définition : Voisin semi-indépendant	51
43	Définition : Point d'articulation	51
44	Définition : Coupe $s - t$	52
45	Définition : Ensemble dominant connexe de poids minimal et de distance en nombre de sauts bornée (MWCDS-BHD)	58
46	Définition : Profondeur d'un nœud	59

LISTE DES THÉORÈMES

1	Théorème : Coût du chemin borné	64
2	Théorème : Borne d'approximation	68

