



TER M1 INFORMATIQUE

Analyse et conception d'algorithmes économes en énergie dans les réseaux de capteurs

Auteurs :

Chloé DESDOUITS
Sofiane Zahir KALI
Rabah LAOUADI
Samuel ROUQUIE

Encadrante :

Anne-élisabeth BAERT

Table des matières

1	Introduction	5
1.1	Capteurs	5
1.2	Caractéristiques techniques des capteurs actuels	7
1.3	Réseau de capteurs sans fil (Wireless Sensors Network)	8
1.3.1	Architecture	8
1.3.2	Caractéristiques	9
1.3.3	Problématique et défis	9
2	État de l’art	11
2.1	Modélisation des réseaux de capteurs sans fil	11
2.1.1	Critères pratiques	11
2.1.2	Modèle d’un réseau de capteurs : un graphe	12
2.1.3	Modèle énergétique	14
	Energie d’un capteur	14
2.1.4	Durée de vie du réseau	15
	Problématique	15
	Définitions	16
2.2	Éléments de classification des algorithmes existants	16
2.2.1	Transmissions en broadcast vs transmissions en single-cast	16
	Broadcast	16
	Single-cast	17
2.2.2	Algorithme à rayon d’émission fixe vs variable	17

Déterminer le rayon optimal d'émission pour un broadcast	17
Rayon d'émission fixe pour tous les nœuds	18
Rayon d'émission variable	18
2.2.3 Algorithme global vs local	19
Catégories	19
2.2.4 Algorithme à balisage vs algorithme sans balisage (beaconless)	19
2.2.5 Algorithme déterministe vs probabiliste	19
2.3 Quelques algorithmes de routage	20
Flow augmentation [7]	20
Flow redirection [7]	20
Energy Aware Routing [20]	21
2.4 Quelques algorithmes de broadcast	22
Blind Flooding	22
Probabilistic Flooding	22
Area-based Beaconless Broadcasting Algorithms [18]	23
Broadcast Incremental-power Protocol [22]	23
Découverte des 1- et 2-voisinages	25
Localized Broadcast Incremental-power Protocol [15]	25
Dynamic Localized Broadcast Incremental-power Protocol [5]	26
LMST Broadcast Oriented Protocol(LBOP) et RNG Broadcast Oriented Protocol(RBOP) [3]	26
LMST&RNG Broadcast Oriented Protocol (LBOP & RBOP) [3]	27
Target Radius LMST Broadcast Oriented Protocol [13]	29
2.4.1 Synthèse	29
Tableau récapitulatif	29
3 Analyse et réflexion	30
3.1 Critique de l'existant	30
3.1.1 Consommation de l'énergie	30
3.2 Nos idées	30
3.2.1 Application du rayon optimal à DLBIP : TR-DLBIP	30
3.2.2 Algorithme qui préserve la connexité du graphe	31

4	Simulations et résultats	32
4.1	Outils de programmation	32
4.2	Paramètres de simulation	33
4.3	Topologie aléatoire	33
4.3.1	Modèle énergétique simple	33
	Time To First Fall	34
	PerCent Node	35
	Lose Connectivity	37
4.3.2	Modèle énergétique plus réaliste	37
	Time To First Fall	38
	PerCent Node	39
	Lose Connectivity	40
5	Conclusion	41
5.1	Bilan de notre travail de groupe	41
5.2	Bilan du TER	43
5.2.1	Problèmes rencontrés	43
5.2.2	Contributions	43

Chapitre 1

Introduction

Les réseaux sans fil font depuis plus d'une dizaine d'années partie intégrante de la vie quotidienne des entreprises, des particuliers, de l'industrie et d'autres organisations. Ils représentent aujourd'hui une des briques de base sur lesquelles vont se fonder les systèmes intelligents omniprésents qui vont constituer une des technologies de l'avenir. Cependant, la majeure partie de ces technologies sans fils, à commencer par le Wifi, elle est basée sur des infrastructures fixes, limitant la mobilité des utilisateurs. Pour faciliter cette mobilité, il existe un autre type de réseau, de plus en plus courant, qui permet aux nœuds du réseau de communiquer directement entre eux sans nécessiter d'infrastructure : ce sont les réseaux ad hoc.

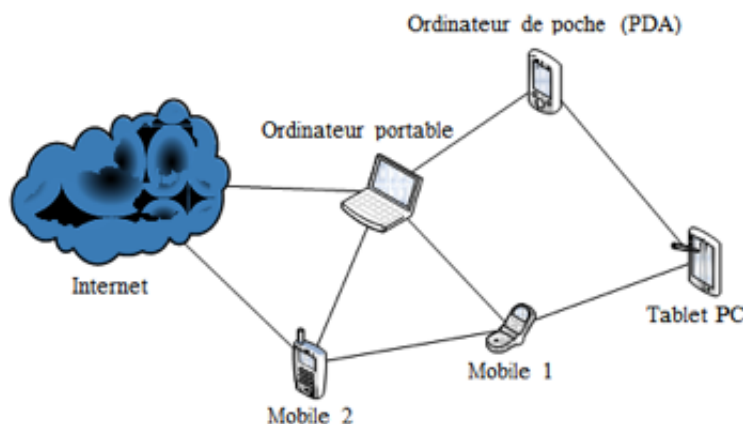


FIGURE 1.1 – Exemple de réseau ad hoc

On distingue donc deux principales classes de réseaux sans fils, les classiques structurés et les non structurés comme les réseaux ad hoc. Les réseaux ad hoc offrent la possibilité de connecter différents dispositifs sans avoir à préinstaller une infrastructure fixe comme dans les réseaux traditionnels. Dans les réseaux ad hoc, l'ensemble des nœuds communiquent directement entre eux (voir figure 1.1). Nous allons nous intéresser à un type particulier de réseau ad hoc : les réseaux de capteurs. Ces réseaux ont de nombreuses applications pratiques dans le médicale, la physique, la chimie, le multimédia, l'automobile, la climatologie...

1.1 Capteurs

Les capteurs sont des petites entités électroniques à faible coût qui ont pour but de récolter des informations dans leur environnement proche comme la température, la vitesse, le bruit, la pression,

le mouvement, la chaleur ou la lumière... La valeur mesurée est convertie dans une représentation analogique ou numérique.



FIGURE 1.2 – Capteur sans fil

Il existe des **capteurs intelligents** (Smart Sensors) dans lesquels coexistent le(s) capteur(s) et les circuits de traitement et de communication. Leurs relations avec des couches de traitement supérieures vont bien au-delà d'une simple « transduction de signal ». Les capteurs intelligents sont des « capteurs d'informations » et non pas simplement des capteurs et des circuits de traitement du signal juxtaposés. De plus, les « Smart Sensors » ne sont pas des dispositifs banalisés car chacun de leurs constituants a été conçu dans l'objectif d'une application bien spécifique.

Lorsque nous parlerons de capteur plus loin dans ce rapport, il s'agira d'un capteur intelligent. Un tel capteur contient quatre unités de base (voir Figure 1.3) :

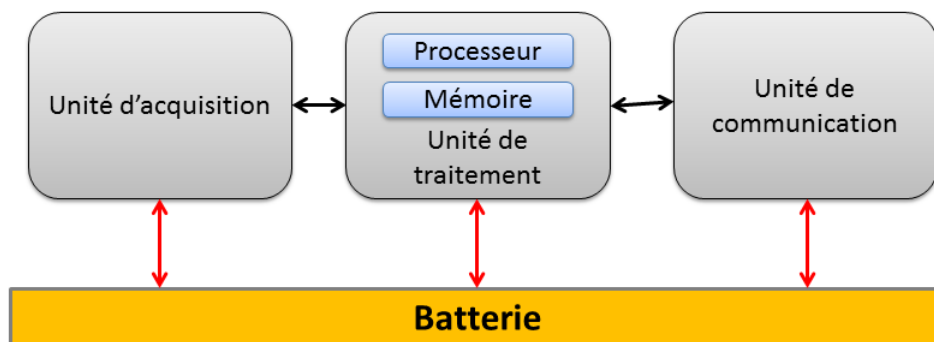


FIGURE 1.3 – Architecture d'un capteur

L'unité d'acquisition est composée d'un capteur qui obtient des mesures sur les paramètres environnementaux et d'un convertisseur Analogique/Numérique qui convertit l'information relevée et la transmet à l'unité de traitement. La perception d'un capteur est limitée par un rayon de sensation (R_s). La Figure 1.4 illustre ce principe.

L'unité de traitement est composée d'un processeur et d'une mémoire intégrant un système d'exploitation spécifique. Cette unité possède deux interfaces, une interface pour l'unité d'acquisition et une interface pour l'unité de communication. Elle acquiert les informations en provenance de l'unité d'acquisition et les envoie à l'unité de communication. Cette unité est chargée aussi d'exécuter les protocoles de communications qui permettent de faire collaborer le capteur avec d'autres capteurs. Elle peut aussi analyser les données captées.

L'unité de communication est l'unité responsable de toutes les émissions et réceptions de données via un support de communication radio. Elle peut être de type optique, ou de type radiofréquence. Fonctionnellement chaque capteur possède un rayon de communication (R_c). La figure 1.4 montre la zone dans laquelle le capteur peut communiquer. Certains capteurs peuvent moduler leur rayon de communication.

L'unité de contrôle d'énergie (batterie) sert à alimenter tous les composants. Cependant, à cause de la taille réduite du capteur, la batterie est limitée et généralement irremplaçable. Ainsi, l'énergie est la ressource la plus précieuse puisqu'elle influe directement sur la durée de vie des capteurs.

Selon son domaine d'application, un capteur peut contenir des modules supplémentaires comme le système de positionnement GPS (Global Positioning System) ou un système lui permettant de se déplacer.

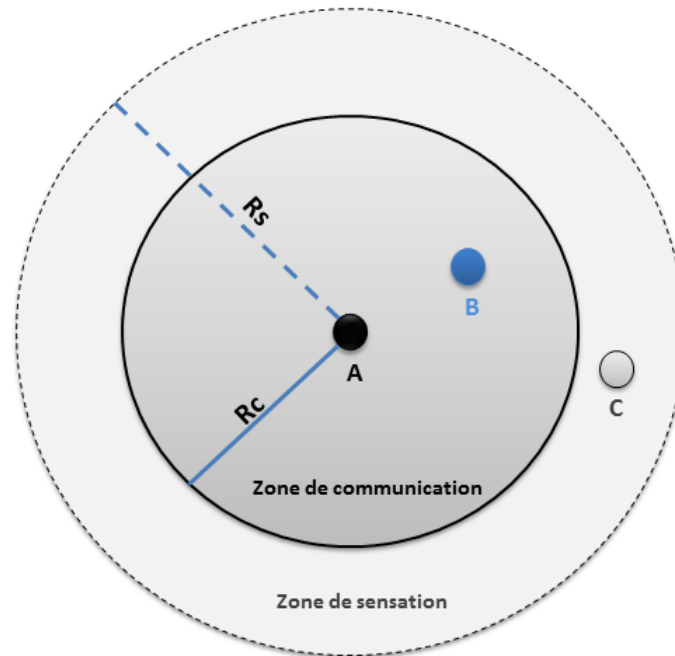


FIGURE 1.4 – Rayon de communication et de sensation

1.2 Caractéristiques techniques des capteurs actuels

Une faible puissance de calcul : quand les ordinateurs peuvent avoir jusqu'à 4 processeurs, chacun cadencé à 3GHz, ou quand les derniers Smartphones peuvent fonctionner jusqu'à 800MHz, un capteur actuel est à peine plus puissant qu'une calculatrice graphique produite dans les années 90.

Un espace de stockage mémoire limité à quelques kilo-octets ou quelques méga-octets impose l'utilisation d'algorithmes distribués, localisés et collaboratifs.

Une puissance radio limitée : l'ordre de grandeur des portées actuellement atteignables par les principaux capteurs est d'une centaine de mètres en extérieur et de quelques dizaines de mètres en intérieur. Cette portée est largement dépendante de la fréquence utilisée et de l'environnement. Elle nécessite un routage multi-saut pour l'acheminement des données vers une entité de collecte : le puits. Les capteurs ne peuvent communiquer qu'avec leur voisinage direct qui va relayer les communications.

Un débit faible : les composants radio d'un capteur sont limités à quelques centaines de kilo-octets par seconde.

Une réserve d'énergie réduite : même s'il existe des mécanismes de recharge d'énergie, la durée de vie d'un capteur reste directement liée au niveau de sa batterie. Cette réserve d'énergie est partagée par chaque unité d'un capteur mais l'unité de communication va en consommer près de 95% lors du fonctionnement actif du capteur. Les enjeux actuels portent donc sur :

- l'augmentation des capacités des batteries
- les dispositifs de transmission radio ultra-basse consommation
- les architectures basse consommation
- des mécanismes d'endormissement
- des protocoles de communication spécifiques

1.3 Réseau de capteurs sans fil (Wireless Sensors Network)

Les réseaux de capteurs sans fil sont un type particulier de réseau ad-hoc. Ces réseaux sont formés d'une multitude de capteurs, capables de s'auto-organiser et ainsi de travailler pour la collecte, le partage et le traitement coopératif des informations sur leur environnement ; le tout sans intervention humaine. Ces dispositifs sont peu coûteux, mais peu performants. Depuis quelques décennies, le besoin croissant d'observer et de contrôler des phénomènes physiques tels que la température, la pression ou encore la luminosité a conduit au déploiement de nombreux réseaux de capteurs.

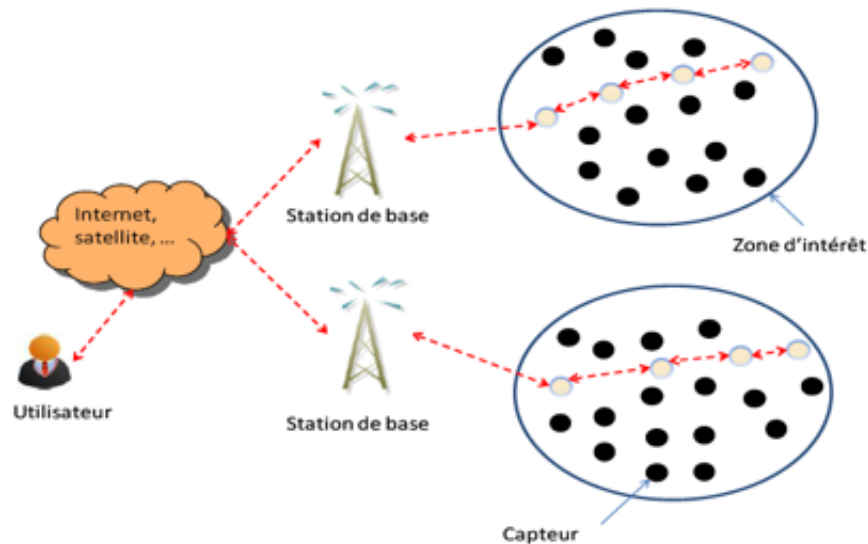


FIGURE 1.5 – Réseaux de capteurs

Dans l'exemple de la figure 1.5, les capteurs sont déployés d'une manière aléatoire dans une zone d'intérêt, et une station de base, située à l'extrémité de cette zone, est chargée de récupérer les données collectées par les capteurs. Lorsqu'un capteur détecte un événement pertinent, un message d'alerte est envoyé à la station de base par le biais d'une communication entre les capteurs. Les données collectées sont traitées et analysées par des machines puissantes.

1.3.1 Architecture

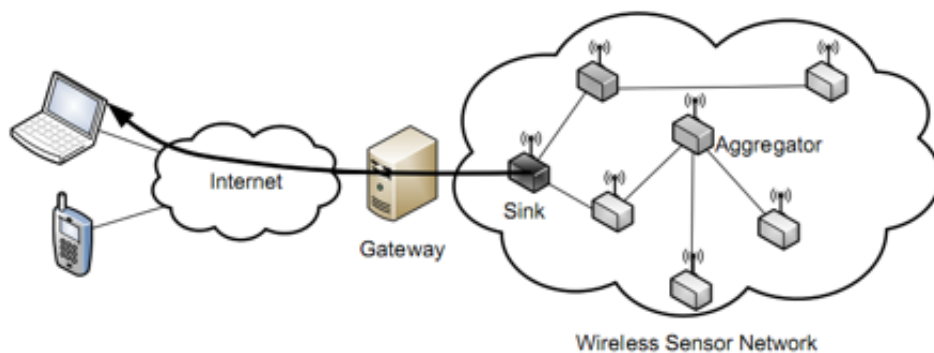


FIGURE 1.6 – architecture d'un réseau WSN

Les réseaux de capteurs sans fils sont construits autour des quatre principales entités suivantes :

Les capteurs décrits précédemment.

L'agrégateur est en charge d'agréger les messages qu'il reçoit de plusieurs capteurs puis de les envoyer en un seul message au puits. Cette opération a pour principal but de limiter le trafic sur le réseau et donc de prolonger la durée de vie globale du réseau de capteur.

Le puits est le nœud final du réseau. C'est à lui qu'est envoyé l'ensemble des valeurs mesurées par le réseau. Il peut arriver qu'il y ait plusieurs puits sur un même réseau de capteurs.

La passerelle est un dispositif qui a la particularité d'avoir deux interfaces réseau. Elle permet de relier le réseau de capteurs sans fils à un réseau plus traditionnel, typiquement l'internet. Habituellement le réseau de capteurs ne sert qu'à faire remonter les mesures, les applications traitant ces informations étant exécutées sur la machine de l'utilisateur final.

1.3.2 Caractéristiques

L'absence d'infrastructure préexistante et de tout genre d'administration centralisée.

Des interférences : les liens radio ne sont pas isolés, deux transmissions simultanées sur une même fréquence, ou utilisant des fréquences proches, peuvent interférer.

Une taille importante : un réseau de capteurs peut contenir des milliers de nœuds.

L'hétérogénéité des nœuds : plusieurs types de capteurs différents connectés entre eux.

Une topologie dynamique : les capteurs peuvent être attachés à des objets mobiles qui se déplacent d'une façon libre et arbitraire rendant ainsi la topologie du réseau fréquemment changeante.

Des contraintes énergétiques : la caractéristique la plus critique dans les réseaux de capteurs est la modestie de ses ressources énergétiques (batterie).

La capacité de stockage et la puissance de calcul sont limitées dans un capteur.

Une bande passante limitée en raison des caractéristiques techniques des radios.

Le faible coût du matériel qui facilite une redondance des liens pour assurer une connexité du réseau en cas de panne d'un ou plusieurs capteurs.

L'impossibilité de remplacer manuellement les nœuds dans le cas où leur position est inconnue (déploiement rapide dans des conditions difficiles ; absence de puce GPS...).

Le caractère aléatoire de la topologie du réseau : celui-ci est déployé en fonction des zones d'intérêt ou aléatoirement.

1.3.3 Problématique et défis

Les caractéristiques particulières des réseaux de capteurs modifient les critères de performance par rapport aux réseaux sans fil traditionnels. Dans les réseaux locaux filaires ou les réseaux cellulaires, les critères les plus pertinents sont le débit, la latence et la qualité de service. En effet, les nouvelles activités telles que le transfert d'images, le transfert de vidéos, et la navigation sur Internet requièrent de bonnes performances selon ces trois critères.

En revanche, dans les réseaux de capteurs conçus pour surveiller une zone d'intérêt, la longévité du réseau est fondamentale. De ce fait, la conservation de l'énergie est devenue un critère de performance prépondérant et se pose en premier lieu tandis que les autres critères comme le débit ou l'utilisation de la bande passante sont devenus secondaires.

Les perspectives d'application des réseaux de capteurs sont enthousiasmantes mais les défis qu'elles posent n'en sont pas moins nombreux et complexes. Parmi les problématiques cruciales, nous pouvons citer :

L'énergie : cette contrainte impose de concevoir des protocoles économes en énergie.

Le routage : le problème de routage consiste à déterminer un acheminement optimal des paquets à travers le réseau au sens d'un certain critère de performance (l'énergie par exemple).

La sécurité : la puissance de calcul limitée d'un capteur ouvre de véritables défis pour concevoir des algorithmes de cryptages distribués et des politiques de confiance spécifiques.

La collecte de données : récupérer les données des capteurs et les assembler.

L'auto-configuration : une partie des applications visées appartient au domaine des applications domestiques. Il est donc important que le routage, l'intégration et l'adaptation à l'environnement soient transparents pour l'utilisateur.

L'autoréparation : les capteurs sont parfois inaccessibles (intégrés dans un mur, installés chez un particulier, déployés dans une zone dangereuse, etc.) et de conception peu sûre (faible coût de production). Une solution complète doit donc gérer efficacement la perte ou l'ajout d'un nœud dans le réseau.

La localisation : il s'agit de concevoir des mécanismes de localisation réalistes vis-à-vis des contraintes et des applications propres aux réseaux de capteurs. Les solutions actuellement proposées sont soit imprécises, soit coûteuses en énergie ou en matériel.

Dans la section 2, nous ferons un état de l'art des algorithmes de routage dans les réseaux de capteur sans fil. Puis en section 3, nous établirons un modèle et analyserons quelques algorithmes existants. En section 4, nous présenterons des résultats de simulations sur la plateforme WSNET. Enfin nous concluons ce rapport.

Chapitre 2

État de l'art

2.1 Modélisation des réseaux de capteurs sans fil

Dans les différents articles du domaine, les réseaux sans fils sont modélisés sous forme de graphe. Cependant, d'autres paramètres (comme les capteurs) sont modélisés de façon différentes par chaque auteur. Nous allons présenter quelques modélisations possibles pour le problème d'économie d'énergie dans les réseaux de capteurs sans fils.

Tout d'abord, nous allons aborder les paramètres pratiques liés aux réseaux de capteurs. Nous formaliserons ensuite la modélisation d'un réseau par les graphes. Puis nous aborderons les différents modèles énergétiques. Enfin, nous nous pencherons sur les définitions de durée de vie d'un réseau. Une table récapitulative des notations est disponible à la fin de cette partie.

2.1.1 Critères pratiques

La modélisation générale d'un réseau de capteur sans fil M_1 prends en compte :

- *Singularité.* Tout les capteurs ne sont pas forcément identiques (batterie, portée, capacité de calcul...)[2].
- *Connexité.* Le réseau est initialement connexe (chaque capteur est lié directement ou indirectement à tous les autres).
- *Dimension.* Le réseau est réparti dans un environnement en 3 dimensions.
- *Mobilité.* Les capteurs sont peuvent changer de position au cour du temps [12].
- *Uniformité.* Tout les capteurs sont identiques (batterie, portée, capacité de calcul...).
- *L'ajout de capteurs.* le réseau comprend un nombre variable de capteurs n . Les ajouts de capteurs en cours de fonctionnement sont possibles.
- *Panne.* Les capteurs peuvent tomber en panne en raison de divers facteurs. Une loi de probabilité modélise ce phénomène.
- *Interférences* Les transmissions radio peuvent être perturbées par des interférences dues à l'environnement ou à des conflits internes au réseau [1].
- *Modèle de consommation énergétique* Un modèle énergétique complexe prenant en compte l'énergie de capture, de traitement, de transmission ainsi que de la taille des messages. [8]

Cependant, un tel modèle est extrêmement complexe à étudier. C'est pourquoi nous nous ramènerons à un modèle plus simple mais moins réaliste choisi dans la beaucoup d'articles [4, 13]. Notre modélisation de base des réseaux de capteurs sans fil M_1 est la suivante :

- *Uniformité.* Tout les capteurs sont identiques (batterie, portée, capacité de calcul...).

- *Connexité*. Le réseau est initialement connexe (chaque capteur est lié directement ou indirectement à tous les autres).
- *Planarité*. Les capteurs sont placés dans un plan euclidien à deux dimensions (distance euclidienne).
- *Staticité*. Pas de mobilité des capteurs : nous supposons que les capteurs sont immobiles.
- *Sans ajout de capteurs*. Le réseau comprend un nombre fixe de capteurs n . Aucun ajout de capteurs en cours de fonctionnement n'est possible.
- *Transmission idéale*. Dans des conditions de transmission de message idéales : aucune interférence entre les messages, pas de perturbations des ondes, système d'identifiants uniques.
- *Fiabilité*. Les capteurs sont fiables, aucune panne n'est possible.
- *Energie initiale fixée*. Chaque capteur a une énergie initiale β donnée. Un modèle décrit la consommation énergétique. Un capteur est éliminé lorsqu'il n'a plus d'énergie ou que son énergie restante ne permet plus aucun envoi de message.
- *Egalité*. Chaque site peut à tout moment débiter une procédure de transmission de données. Une loi de probabilité modélise ce phénomène (loi de poisson).
- *GPS*. Chaque site connaît sa position absolue.

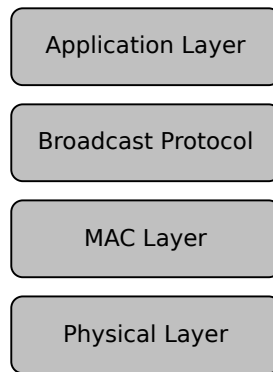


FIGURE 2.1 – Les couches dans les réseaux de capteurs sans fil

Pour l'étude et la conception d'algorithmes dans les réseaux de capteurs sans fil (réseau de capteur sans fil), il est indispensable de définir le problème précis, d'établir un cadre rigoureux, formel et sans ambiguïtés. Au vu de la figure 2.1, notre étude se placera dans la couche 'Broadcast Protocols'. Nous ne développerons pas la couche MAC ni la couche application.

Dans le cadre de notre travail d'étude et de recherche, nous utiliserons, sauf mention du contraire, la modélisation de capteurs simplifiée M_2 (cf 2.1.1) avec pour modèle énergétique celui décrit ci dessous.

2.1.2 Modèle d'un réseau de capteurs : un graphe

Un réseau de capteur sans fil peut être représenté par un graphe $G = (V, E, \gamma)$ où :

- V est un ensemble de noeuds (capteurs)
- $E \subseteq V^2$ est l'ensemble des arêtes représentant les communications possibles entre les capteurs : (u, v) appartient à E signifie que v est à portée de u
- γ est le rayon d'émission maximum

Notations 1. Soit $n = |V|$ la taille du réseau de capteur sans fil. Notons que les éléments de E dépendent de la position des capteurs ainsi que de leur portée. Tous les capteurs ont la même portée maximale notée γ . Soit ij l'arête allant de i à j . Soit $d(u, v)$ la distance euclidienne dans \mathbb{R}^2 entre u et v

Nous avons :

$$E = \{(u, v) \in V^2 \mid d(u, v) \leq \gamma\}$$

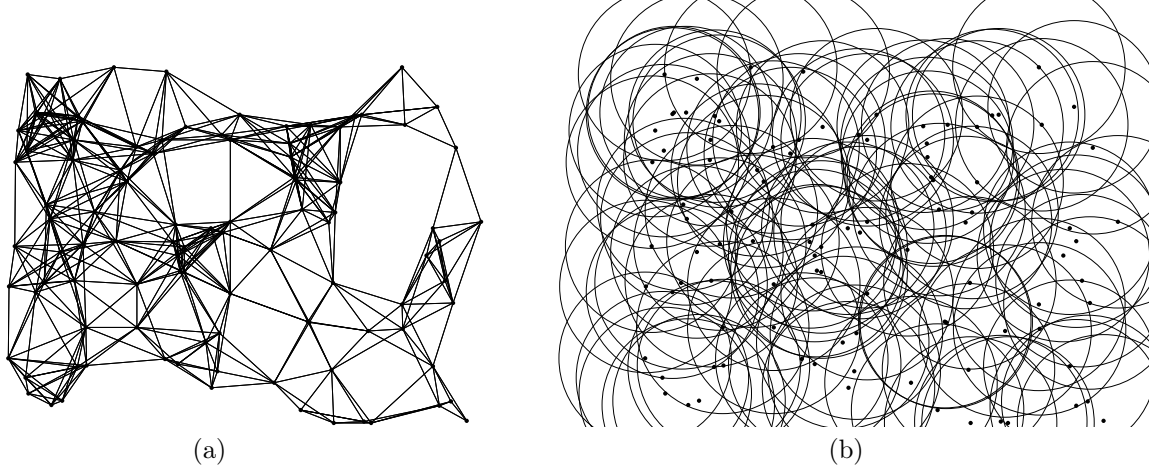


FIGURE 2.2 – Graphe Unité (a) et Graphe Disque (b)

Définition 1. Soit $d_G(u, v)$ la *distance unitaire* entre u et v : $d_G(u, v)$ est la taille du plus court un chemin $c(u, v)$ en terme du nombre d'arêtes parcourues.

Définition 2. Soit $G = (V, E, \gamma)$ le *graphe Unité* du réseau de capteur sans fil et γ son rayon de communication (cf fig.2.2 (a)).

Définition 3. Nous définissons le *graphe Disque* comme $G = (V, D, \gamma)$ où $D = \{Cercle(u, \gamma), u \in V\}$ (cf fig.2.2 (b)).

Définition 4. Soit $N_1(u)$ le 1-voisinage de u : $N_1(u) = \{v \in V \mid (u, v) \in E\}$.

Définition 5. Soit $N_1(u)$ le 2-voisinage de u : $N_2(u) = \{v \in V \mid \exists w \in V : \{(u, w); (w, v)\} \in E^2\}$.

Définition 6. Soit $N_1(u)$ le k -voisinage de u , $k \in \mathbb{N}$: $N_k(u) = \{v \in V \mid d_G(u, v) = k\}$.

Vocabulaire 1. Nous parlerons de 1- 2- et k -voisins de i pour désigner des noeuds appartenant respectivement à $N_1(i), N_2(i), N_k(i)$.

Définition 7. Le voisinage de $A \subseteq V$ est l'ensemble $N(A) = \{v \in V \setminus A \mid \forall u \in A, (u, v) \in E\}$.

Définition 8. Le degré de u est le nombre $N(u) = |N_1(u)|$.

Définition 9. Le diametre de G est $diametre_G = \max_{i, j \in [1, n], i < j} (d_G(i, j))$.

Définition 10. Le degré de G est la moyenne des degrés de chaque sommet :

$$N_G = \sum_{i=1}^n \frac{1}{n} N(i)$$

Définition 11. La densité de G est le nombre $D_G = N_G / diametre_G$.

Définition 12. La distance unitaire de G est la moyenne des distances entre toutes paires de sommets :

$$d_G = \sum_{i, j \in [1, n], i < j} \frac{1}{n} d_G(i, j)$$

Définition 13. La distance euclidienne de G est la moyenne des distances euclidienne entre toutes paires de sommets :

$$d(G) = \sum_{i, j \in [1, n], i < j} \frac{1}{n} d(i, j)$$

Notations 2. Nous notons $MST(G)$ un des arbres couvrants de poids minimum de G .

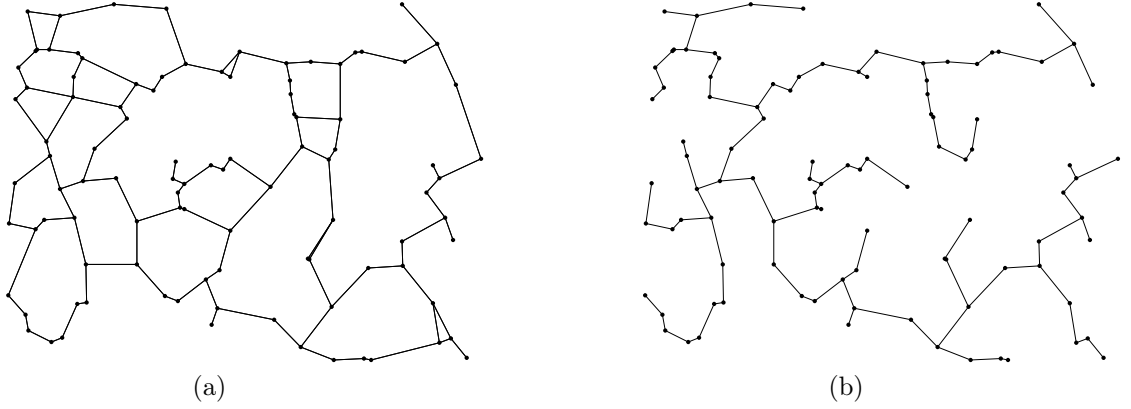


FIGURE 2.3 – Graphe RNG(a) et Graphe MST(b)

Définition 14. Soit $RNG(G)$ (pour Relative Neighborhood Graph) le graphe de voisinage . $RNG(G) = (V, E_{rng})$ avec

$$E_{rng} = \{(u, v) \in E \mid \nexists w \in N_1(u) \cap N_1(v) : d(u, w) < d(u, v) \wedge d(v, w) < d(u, v)\}$$

Le graphe de voisinage est noté $RNG(G)$ (pour Relative Neighborhood Graph).

Remarque 1. De façon intuitive, pour construire le graphe RNG, pour chaque noeud u , (u, v) est ajouté à E_{rng} si u n'a pas de voisin w plus proche de v .

Propriété 1. Nous avons la propriété suivante :

$$MST \subseteq RNG(G) \subseteq G$$

2.1.3 Modèle énergétique

Energie d'un capteur

Dans [10], Dong présente les deux modèles de consommation énergétique communément utilisés.

The Packet based model. Nous utiliserons pour notre analyse le modèle de consommation d'énergie idéal suivant : nous considérerons que chaque capteur i a une énergie initiale $E_{init} = \beta$. L'envoi de message est le seul facteur de perte d'énergie. L'énergie consommée lors de la réception de message, l'acquisition et traitement des informations sera considérée comme négligeable. Tous les capteurs i offrant les mêmes caractéristiques, ils peuvent modifier leur rayon d'émission r_i en respectant l'inégalité suivante : $0 \leq r_i \leq \gamma$. Nous noterons E_i l'énergie restante de i . L'envoi d'un message de i avec un rayon r coûte

$$E(r) = \begin{cases} r^\alpha + c & \text{si } i \neq j \\ 0 & \text{sinon} \end{cases}$$

L'envoi d'un message de i à j ($d(i, j) \leq \gamma$) coûte $E_{ij} = E(d(i, j))$.

The Time based model. Ce modèle plus réaliste prend en compte l'énergie de réception des messages, de traitement ainsi que d'écoute inactive du réseau (mode veille). En effet, dans [16], Kasten souligne le fait que souvent, la réception, l'écoute et le traitement consomment en moyenne autant d'énergie que la transmission de message. Dans de nombreuses topologies, si la fréquence des transmissions est faible, beaucoup de capteurs vont être à court d'énergie avant même d'avoir pu transmettre des messages. Dans beaucoup d'applications, la densité du réseau étant élevée, le coût propre aux transmissions de message est relativement faible tandis que le coût de réception des messages est élevé puisque chaque capteur traite les messages de son voisinage (qui en l'occurrence est grand).

Energie du réseau.

Définition 15. L'énergie potentielle de G est la somme des énergies des capteurs :

$$E_G = \sum_{i=1}^n E_i$$

Définition 16. La consommation de G est :

$$C_G = n\beta - E_G$$

Définition 17. Le coût moyen de transmission de G est :

$$c_G = E(d(G))$$

Définition 18. Le cout moyen d'une transmission est $C(1)$.
Le cout moyen de k transmissions est $C(k)$.

2.1.4 Durée de vie du réseau

Problématique

Dans un réseau de capteurs sans fil, la contrainte majeure est l'efficacité de l'algorithme utilisé en matière de consommation énergétique. En effet, la principale caractéristique des capteurs est leur petite taille et leur micro-batterie. Dans la majeure partie des cas, remplacer les batteries est impossible. Cependant la durée de vie d'un réseau de capteur sans fil est difficile à définir et à mesurer. Il n'y a pas de définition absolue. Elle indique combien de temps le réseau sera 'efficace' par rapport à l'application donnée (nombre de transmissions effectuées, connexité du réseau, pourcentage de noeuds vivants...).

Dans la littérature, deux approches apparaissent clairement en matière de maximisation de la durée de vie des réseau de capteur sans fils. Une approche indirecte consiste à minimiser la consommation d'énergie de façon locale tandis que l'autre a pour but de maximiser la directement la durée de vie du réseau de façon plus globale. Bien que l'approche indirecte puisse améliorer la durée de vie du réseau, elle ne suffit pas à elle seule à appréhender le problème de la durée de vie. Parmi ces approches figurent par exemple le fait de maximiser le nombre de transmissions effectuées avant qu'un capteur ne meurt.

Dans [17], Liang prouve que le problème 'THE MINIMUM-ENERGY BROADCAST TREE PROBLEM' est NP-complet par réduction à (3-CNF SAT) :

Théorème 1. Soient un réseau de capteur sans fil G dans lequel chaque noeud a k rayons de transmission possibles, une source s et un entier positif w . Déterminer s'il existe un arbre de broadcast de source s tel que la somme des coûts de transmission aux noeuds relais (qui ne sont pas des feuilles) soit inférieure à w est NP-Complet.

Ainsi, pour un broadcast donné il n'existe pas d'algorithme centralisé polynomial pour trouver l'arbre de diffusion optimale. Il est évident qu'il n'en existe pas de distribués. Dans [10], Dong prouve que le problème BROADCAST LIFETIME est NP-Complet par réduction à (3DM) :

Théorème 2. Soient un réseau de capteur sans fil G , une source s et un entier positif k . Déterminer si G a assez d'énergie pour broadcaster k messages à partir de s est NP-Complet.

De façon formelle, il est donc souvent impossible de calculer le nombre total de broadcasts réussis étant donné un réseau et un protocole (sauf quand le protocole est très simple). Nous analyserons donc *les performances moyennes* des algorithmes quand cela est possible. Sinon, les simulations nous permettront de mesurer les performances en fonction de différentes topologies.

Définitions

Les définitions et critères de durée de vie d'un réseau de capteur sans fil sont tirés des articles [9],[6],[11].

LIFETIME 1. Nombre moyen de transmissions réussies avant qu'un capteur n'ai plus de batterie (TTFF).

LIFETIME 2. Nombre moyen de transmissions réussies avant que le réseau ne perde sa connectivité (LC).

LIFETIME 3. Nombre moyen de transmissions réussies jusqu'à ce qu'il ne reste que $X\%$ de noeuds vivants (PCN).

$G(V, E, \gamma)$	Graphe unité
n	Nombre de capteurs
γ	Rayon d'émission maximum
α	Constante de consommation énergétique
c	Constante de consommation énergétique
β	Energie initiale des capteurs
E_i	Energie restante de i
E_{ij}	Coût d'envoi d'un message de i à j
$N(u)$	Degré de u
$d_G(i, j)$	Distance dans G entre i et j
$d(i, j)$	Distance euclidienne entre i et j
$N_k(u)$	Nombre de k -voisins de u
N_G	Degré de G
d_G	Distance de G
$d(G)$	Distance euclidienne de G
$diametre_G$	Diametre de G
D_G	Densité de G
$C(k)$	Coût de k broadcasts

TABLE 2.1 – Notations

2.2 Eléments de classification des algorithmes existants

Les éléments de classification cités ci-dessous sont inspirés des articles [21], [14], [23].

2.2.1 Transmissions en broadcast vs transmissions en single-cast

Dans un réseau $G(V, E, \gamma)$, deux capteurs peuvent communiquer directement uniquement s'ils sont 1-voisins dans G . A cause l'atténuation du signal wifi, le rayon de transmission est relativement limité, c'est pourquoi les communications doivent se faire par multi-sauts. Tout ceci est géré par un algorithme de routage. Les algorithmes de routage peuvent gérer le broadcast (un nœud vers tous les autres) et/ou le single-cast (un nœud vers un autre). Chacune des approches a des avantages dans un contexte précis.

Broadcast

La procédure de broadcast consiste à transmettre une information à tous les autres nœuds du graphe. C'est un mécanisme fondamental pour la propagation des données ainsi que pour la découverte de routes.

En effet le broadcast est utilisé pour acquérir une connaissance du voisinage des capteurs. Ce mode de communication est naturel pour des périphériques en wifi avec antenne omnidirectionnelle car les ondes sont réparties uniformément dans un cercle de rayon donné. Tout le voisinage est donc touché et le coût en énergie est celui d'une seule transmission.

Le routage des broadcasts nécessite de s'assurer que le message ne soit pas indéfiniment retransmis mais que tous les nœuds du graphe soient touchés. Parfois même si le destinataire est à distance 1 dans le graphe unité, il est nécessaire que le message soit relayé par un nœud tiers pour des raisons énergétiques.

Single-cast

La transmission par single-cast est le moyen classique de transmettre les messages : un expéditeur fait parvenir un message à un destinataire. Ce moyen de communication est valable lorsqu'on ne cherche qu'à atteindre un seul nœud. En effet, le routage d'un single-cast coûte moins cher que le broadcast (en moyenne, moins de transmissions sont nécessaires).

2.2.2 Algorithme à rayon d'émission fixe vs variable

Dans les réseaux de capteurs, les transmissions s'effectuent sans fils. Un bon moyen de réduire la quantité d'énergie utilisée est donc de réduire la portée d'une transmission. L'idée des algorithmes de broadcast est de choisir les nœuds relais de façon à ce que leur ensemble se rapproche au maximum d'un maillage optimal. Certains algorithmes choisissent de fixer un rayon égal pour tous les nœuds ; d'autres utilisent un rayon variable pour chaque nœud.

Notons que les algorithmes qui émettent en single-cast ont, par définition, un rayon d'émission variable.

Déterminer le rayon optimal d'émission pour un broadcast

Théorème 3. Sans chevauchement et sans vide, un plan ne peut être découpé de façon uniforme que par des polygones réguliers de type triangle, carrés ou hexagones.

Démonstration. Soit m le nombre de sommets d'un m -polygone et n le nombre de m -polygones nécessaires pour couvrir 2π degrés. La somme des angles internes étant égale au produit d'un angle interne par le nombre de côtés m , elle sera égale à $\pi m - 2\pi$ radian. Un angle interne vaut donc $\frac{(m-2)\pi}{m}$. Nous avons :

$$\frac{(m-2)n\pi}{m} = 2\pi \Leftrightarrow (m-2)(n-2) = 4$$

Comme n et m sont des entiers, les seules solutions sont :

$$(m-2; n-2) \in \{(1, 4), (2, 2), (4, 1)\} \Leftrightarrow m \in \{3, 4, 6\}$$

□

Théorème 4. Le quadrillage en hexagone est celui offrant le moins de chevauchements du point de vue réseau de capteur sans fil.

Démonstration.

□

Soit P un plan sur lequel n capteurs sont placés. Chaque capteur peut émettre des messages avec un rayon compris entre 0 et γ . Nous allons déterminer pour tous les nœuds un même rayon d'émission fixe R . Etant donnée une source s et un message M à broadcaster, nous devons placer les nœuds relais de façon à minimiser leur nombre m . Bien sûr, leur nombre dépend directement du rayon R .

Dans un réseau hexagonal, nous avons les résultats suivants : Soit S l'aire du plan rectangulaire P . Connaissant R , nous pouvons calculer facilement le nombre de sommets nécessaires pour couvrir le plan. Soit h le nombre d'hexagones couvrant P , on a :

$$h \simeq \frac{Surface(P)}{Surface(hexagone)} = \frac{2S}{3R^2\sqrt{3}}$$

Comme il faut deux nœuds par hexagone,

$$\begin{aligned} n &= 2h \\ &= \frac{4S}{3R^2\sqrt{3}} \\ &= \frac{k}{R^2} \end{aligned}$$

Ainsi, le coût d'un broadcast vaut :

$$\begin{aligned} C(1)(R) &= 2h \\ &= n \cdot E(R) \\ &= \frac{k}{R^2} \cdot E(R) \\ &= \frac{k}{R^2} \cdot (R^\alpha + c) \end{aligned}$$

Nous cherchons le rayon optimal minimisant $C(1)(R)$. Comme $\alpha \geq 2$, $c \geq 0$ et $R > 0$, nous avons 4 cas possibles :

- $\alpha = 2$, $c = 0$: $C(1)(R) = k$. R_{opt} ne dépend pas de r .
- $\alpha = 2$, $c \neq 0$: $C(1)(R) = k(1 + cR^{-2})$. Le rayon doit être le plus grand possible donc : $R_{opt} = \gamma$.
- $\alpha > 2$, $c = 0$: $C(1)(R) = kR^{\alpha-2}$. $R_{opt} = 0$
- $\alpha > 2$, $c \neq 0$: $C(1)(R) = k(R^{\alpha-2} + cR^{-2})$.

En dérivant, nous obtenons : $C'(1)(R) = k((\alpha - 2)R^{\alpha-3} - 2cR^{-3})$.

Cette fonction admet un minimum en

$$R_{opt} = \sqrt[\alpha]{\frac{2c}{\alpha - 2}}$$

Bien sûr, en pratique il est impossible, sur une topologie quelconque, d'extraire un tel maillage hexagonal de rayon R_{opt} . Cependant il est possible de s'en approcher le plus possible.

Rayon d'émission fixe pour tous les nœuds

Certains algorithmes de broadcast les plus primitifs (cf 2.4, 2.4) utilisent un rayon d'émission fixe. Les performances énergétiques associées ne sont pas très bonnes car réduire le rayon d'émission et réduire le nombre de transmissions sont les deux seuls moyens d'économiser de l'énergie lors du routage d'un broadcast.

Rayon d'émission variable

La plupart des algorithmes de broadcast utilisent un rayon d'émission variable pour les raisons citées ci-dessus. Tous les algorithmes de routage de single-cast utilisent un rayon d'émission variable car ils ajustent leur transmission pour toucher uniquement le nœud destinataire.

2.2.3 Algorithme global vs local

Le fait qu'un algorithme soit global ou local dépend du type d'informations qu'il utilise. Si un capteur connaît dès son arrivée des informations sur ses voisins, l'algorithme est global. À l'inverse, un algorithme distribué fournit aux capteurs uniquement les informations que leurs perceptions peuvent leur procurer.

Cependant, la distinction entre global et local n'est pas toujours évidente. Des algorithmes centralisés peuvent être implémentés d'une manière distribuée en décidant par exemple d'un noeud possédant la connaissance globale du réseau. À l'inverse, un algorithme distribué peut simuler une connaissance globale : au travers d'un échange séquentiel et très localisé d'informations (1 ou 2-voisinage par exemple), chaque noeud peut combiner sa connaissance avec celle de ses voisins et ainsi obtenir une vision globale du réseau. Cependant, une telle phase de propagation coûte très cher en terme de nombre de messages échangés et de temps.

Si le réseau est dynamique (mobilité des capteurs), maintenir une connaissance locale du réseau devient plus complexe tandis que tenir à jour la topologie globale de celui-ci devient impossible par le moyen cité précédemment. Ainsi, la quantité et la nature des informations nécessaires au déroulement de l'algorithme sont une bonne mesure de la capacité d'adaptation du protocole à un environnement dynamique.

Catégories

Global : protocole de routage, centralisé ou distribué nécessitant une connaissance globale du réseau.

Quasi-global : protocole distribué de broadcast nécessitant une connaissance quasi-globale du réseau.

Quasi-local : protocole distribué nécessitant une connaissance du réseau principalement locale et occasionnellement globale (ex : Cluster networks : tandis que les groupes peuvent être construits de manière locale, des réactions en chaînes peuvent arriver).

Local : protocole distribué nécessitant une connaissance très locale du réseau. Tous les algorithmes de 1 ou 2-voisinage appartiennent à cette catégorie.

Remarquons que pour qu'un protocole soit extensible, il doit être avant tout distribué et local : le comportement de chaque noeud, bien qu'il nécessite qu'une connaissance locale du réseau, permet d'atteindre l'objectif global. Il est facile d'admettre que l'extensibilité d'un réseau de capteur sans fil est inversement proportionnelle à la localité du protocole.

2.2.4 Algorithme à balisage vs algorithme sans balisage (beaconless)

Les algorithmes "beaconless" ne nécessitent pas de phase d'initialisation ni de mise à jour. Les nouveaux capteurs sont immédiatement opérationnels mais n'ont aucune connaissance de leur environnement et notamment de leur voisinage dans G .

Dans les algorithmes avec balisage, tout capteur, lorsqu'il est connecté au réseau, commence par une procédure d'initialisation et stocke en mémoire un certain nombre d'informations (voisins, groupe, topologie locale...). Au cœur de l'algorithme, chaque noeud met périodiquement à jour ces informations. Il est possible, par exemple, que chaque site envoie régulièrement à ses voisins un message de type 'Hello' contenant par exemple son Id, sa position, sa dominante connexe, son degré, ses voisins, etc.

2.2.5 Algorithme déterministe vs probabiliste

Un algorithme déterministe est un algorithme qui, confronté à un choix, prendra toujours la même décision : celle prévue par le concepteur. Chaque noeud prendra donc, dans des conditions similaires, la même décision.

À l'inverse, un protocole de routage peut utiliser des fonctions probabilistes pour prendre certaines décisions de manière aléatoire. Il en existe deux types : l'un qui prend toujours la décision optimale mais au bout d'un temps qui peut être long ; l'autre qui décide rapidement mais dont la décision peut être mauvaise.

2.3 Quelques algorithmes de routage

Tous les algorithmes présentés ci-dessous utilisent la modélisation $M2$ (2.1.1) des réseaux avec pour modèle énergétique le 'Packet based model' (2.1.3).

Flow augmentation [7]

Description : L'algorithme $FA(x_1, x_2, x_3)$ est un algorithme d'augmentation de flot. L'idée consiste à utiliser un algorithme de plus court chemin existant (comme l'algorithme distribué de Bellman-Ford) et de lui passer trois paramètres (x_1 , x_2 et x_3) qui détermineront sa fonction de poids sur les arcs. Ces paramètres coefficienteront les trois facteurs suivants : le coût de transmission du nœud i au nœud j (e_{ij}) ; l'énergie résiduelle du nœud i (E_i), et l'énergie initiale du nœud i (E_i). Voici la formule qui donne le poids de l'arc (i, j) en fonction des paramètres :

$$c_{ij} = e_{ij}^{x_1} \frac{E_i^{-x_2}}{E_i^{x_3}}$$

Algorithm 1 $FA(x_1, x_2, x_3)$

Require: a network $G(N, A)$

```

while each node  $i$  has some energy do
  for each commodity  $c \in C$  do
    for each origin node  $o \in O^{(c)}$  do
      compute the shortest cost path from  $o$  to  $D^{(c)}$ 
      augment the flow on this path by an amount of  $\lambda Q_i^{(c)}$ 
    end for
  end for
end while
return the total flow  $q$  in the network

```

Durée de vie :

$$TTF(q) = \min_{i \in N} \frac{E_i}{\sum_{j \in S_i} e_{ij} \sum_{c \in C} q_{ij}^{(c)}}$$

avec :

- q le flot
- N l'ensemble des nœuds du graphe (les capteurs)
- E_i la quantité d'énergie initiale du nœud i
- S_i l'ensemble des voisins du nœud i
- e_{ij} l'énergie requise pour transmettre une unité d'information du nœud i au nœud j
- C l'ensemble des composantes du réseau
- $q_{ij}^{(c)}$ le taux d'information de la composante c transmise du nœud i au nœud j

Flow redirection [7]

Description : Cet algorithme est basé sur le fait que le flot trouvé est optimal ssi pour tous les chemins de l'origine à la destination portants un flot positif, la durée de vie minimum est la même. Le principe consiste donc à rediriger une partie du flot de chaque nœud à travers un autre chemin vers la destination tel que la durée de vie de chaque chemin portant du flot augmente (ou au moins reste stable). Le flot initial a pour valeur la quantité totale d'informations générées par le réseau.

Pour chaque nœud dans le réseau, on compare les chemins sortants de ce nœud vers la destination. On construit un ordre total sur ces chemins en les classant en fonction de leur nœud qui possède la plus courte durée de vie (puis par nombre de nœuds).

Si le nœud dont la durée de vie est la plus courte sur le chemin est le nœud source (i), alors on a deux choix. On regarde les voisins sortants de (i). Soit on redirige un des flots qui passe par le voisin (j) dont le coût de transmission e_{ij} est le plus grand vers un des chemins qui passe par le voisin (j) dont le coût de transmission e_{ij} est le moindre. Soit on redirige un flot quelconque vers le chemin le plus petit dans l'ordre total parmi ceux qui passe par le voisin (j) dont le coût de transmission e_{ij} est le moindre.

Si, en revanche, le nœud dont la durée de vie est la plus courte sur le chemin n'est pas le nœud source, on a également deux possibilités. Dans les deux cas on redirige le flot le plus petit dans l'ordre total. Soit on redirige vers le chemin le plus grand dans l'ordre total. Soit on redirige vers le chemin qui passe par le voisin (j) dont le coût de transmission e_{ij} est le moindre parmi les chemins supérieurs dans l'ordre total au chemin redirigé.

Une fois que l'algorithme a défini pour un nœud donné les deux chemins, il détermine la quantité de flot à rediriger.

Algorithm 2 $FR()$

Require: a network $G(N, A)$

add to G an imaginary super destination node $\tilde{d}^{(c)}$

for all $d \in D^{(c)}$ **do**

$\tilde{d}^{(c)} \in S_d$

$e_{d\tilde{d}^{(c)}} = 0$

end for

let q be the initial flow

for all $o \in O^{(c)}$ **do**

$q_{o\tilde{d}^{(c)}} = Q_o^{(c)}$

end for

for each $c \in C$ **do**

for each $i \in N - D^{(c)}$ **do**

(Determine the Two Paths)

(Calculate $\epsilon_i^{(c)}$)

(Redirect the flow)

end for

end for

return

Durée de vie :

$$TTFF(q) = \min_{i \in N} \frac{E_i}{\sum_{j \in S_i} e_{ij} \sum_{c \in C} q_{ij}^{(c)}}$$

Energy Aware Routing [20]

Description : Ce protocole est réactif (càd qu'il ne découvre un chemin entre deux nœuds qu'en cas de besoin) et les requêtes sont initiées depuis la destination. Il se décline en trois phases : la phase d'initialisation qui permet de trouver tous les chemins de la source à la destination ; la phase de communication de données ; et la maintenance des routes. Les deux premières phases sont détaillées dans les algorithmes ci-dessous.

Algorithm 3 Setup phase of EAR

```
Cost( $N_D$ ) = 0
{Flood the network from  $N_D$  to  $N_S$  :}
for each intermediate node  $N_i$  do
  for each neighbor  $N_j$  of  $N_i$  do
    if  $d(N_i, N_S) \geq d(N_j, N_S)$  and  $d(N_i, N_D) \leq d(N_j, N_D)$  then
      Forward the request from  $N_i$  to  $N_j$ 
    end if
  end for
end for

for each intermediate node  $N_i$  do
  for each neighbor  $N_j$  of  $N_i$  do
    On receiving a request :
     $C_{N_j, N_i} = \text{Cost}(N_i) + \text{Metric}(N_j, N_i)$ 
     $FT_j = \{i | C_{N_j, N_i} \leq \alpha(\min_k C_{N_j, N_k})\}$ 


$$P_{N_j, N_i} = \frac{\frac{1}{C_{N_j, N_i}}}{\sum_{k \in FT_j} \frac{1}{C_{N_j, N_k}}}$$


$$\text{Cost}(N_j) = \sum_{i \in FT_j} P_{N_j, N_i} C_{N_j, N_i}$$

  end for
end for
```

Algorithm 4 Data communication phase of EAR

```
 $N_S$  chooses randomly a number
This number points a neighbor in the forwarding table  $FT_S$  out
Send the packet from  $N_S$  to this neighbor  $N_j$ 
Do the same thing for all intermediate nodes until the packet reaches the destination
```

2.4 Quelques algorithmes de broadcast

Blind Flooding

Description : le "blind flooding" ou "broadcast aveugle" est un algorithme glouton de broadcast sans balisage. Lors de la réception d'un message par un noeud, si c'est la première fois qu'il le reçoit, il le broadcast à ses voisins (avec le rayon maximum), sinon il ne fait rien.

Consommation : le coût global d'un broadcast est exactement le coût moyen puisque chaque noeud i envoie un unique message $\langle M, \gamma \rangle$ par broadcast. Le coût global d'un broadcast est $C(1) = n \cdot E(\gamma) = n \cdot \gamma^\alpha + n \cdot c$. Le coût global de k broadcasts est $C(k) = k \cdot n \cdot E(\gamma) = k \cdot n \cdot \gamma^\alpha + k \cdot n \cdot c$.

Durée de vie :

$$TTFF(\text{BlindFlooding}) = \lfloor \frac{\beta}{n \cdot \gamma^\alpha + n \cdot c} \rfloor$$

Probabilistic Flooding

Description : Afin d'éviter le redondance et les collisions, une idée est que chaque noeud retransmette le message suivant une probabilité P lorsqu'il le reçoit pour la première fois. Si $P = 1$ cela équivaut

au blind flooding. *Consommation* : Chaque noeud ayant une probabilité P de retransmettre le message $\langle M, \gamma \rangle$, en moyenne $P \cdot n$ noeuds envoient ce message lors d'un broadcast. Le coût moyen d'un broadcast est $C(1) = P \cdot n \cdot E(\gamma)$. Le coût global de k broadcasts est $C(k) = P \cdot k \cdot n \cdot E(\gamma)$. *Durée de vie* :

$$TTFP(ProbabilisticFlooding) = \lfloor \frac{\beta}{P \cdot (n \cdot \gamma^\alpha + n \cdot c)} \rfloor$$

Area-based Beaconless Broadcasting Algorithms [18]

Description : Nous supposons que les noeuds n'ont aucune connaissance de leur voisinage. Cependant ceux-ci connaissent leur position géographique (GPS par exemple). Chaque capteur a un rayon d'émission fixe R et couvre une zone circulaire.

L'idée maitresse de ABBA est assez simple. A la première réception d'un message, le noeud initialise un timer. Avant expiration du timer, le noeud peut recevoir d'autres copies du même message. Si un noeud u reçoit le même message de différentes sources et que ces sources couvrent sa zone, cela signifie que chacun de ses voisins potentiels a déjà reçu le message. Alors u ne retransmet pas le message et quand que son timer expire, il ne fait rien. Sinon si le voisinage de u n'est pas couvert alors u retransmet.

Comme chaque couverture est un cercle de rayon R , le critère de couverture peut être simplifié : au lieu de s'intéresser au disque entier, un noeud vérifie uniquement si le périmètre π de sa zone est couvert par ses voisins lui ayant envoyé le message.

Algorithm 5 ABBA

Require: : Un noeud source s , un message M
 s envoie $\langle M, s, position_s, R \rangle$;
Réception par u de $\langle M, v, position_v, R \rangle$
 $z_u \leftarrow ensemble_{vide}$;
Calculer la zone z couverte par v lors de la transmission;
Déclencher le timer;
repeat
 Attendre la réception d'une autre copie de M ou que le timer expire;
 if Une copie de M est reçue **then**
 Mettre à jour z_u ;
 réinitialiser le timer;
 end if
until timer expire;
if $\pi \not\subseteq z_u$ **then**
 Retransmettre;
end if
Ignorer les autres copies de M ;

Broadcast Incremental-power Protocol [22]

Description : Rappelons les désavantages des transmissions longue portée : interférences, cout énergétique (le modèle de consommation énergétique des noeuds est non linéaire par rapport au rayon à cause de l'atténuation du signal radio). C'est pourquoi il est nécessaire de trouver le bon compromis entre le rayon de transmission et le nombre de messages circulant. En effet, un large rayon de transmission coûte cher mais atteint beaucoup de noeuds; un court rayon coûte très peu cher mais augmente le nombre de messages.

BIP est glouton et centralisé. Il se base sur l'algorithme de Prim [19] : un algorithme permettant de construire un arbre couvrant de poids minimum à partir d'un sommet d'un graphe. Le principe de l'algorithme de Prim est de construire l'arbre couvrant minimal arête par arête. Pour ajouter une arête

à un arbre partiellement construit, il considère l'ensemble des arêtes dont une extrémité est connectée à l'arbre déjà construit, et l'autre extrémité ne l'est pas. Il choisit dans cet ensemble une arête de poids minimal qu'il ajoute à l'arbre. L'algorithme commence avec un arbre contenant un nœud et pas d'arêtes, et ajoute successivement $n - 1$ arêtes.

La formation d'un arbre couvrant de poids minimum dans BIP suit le même principe dans le sens où les arêtes sont une par une ajoutées à l'arbre. En fait, BIP utilise l'algorithme de Prim avec une différence fondamentale : au lieu d'utiliser des coûts fixes P_{ij} sur les arêtes (demeurant inchangés au cours de la procédure), BIP actualise dynamiquement ces couts P'_{ij} à chaque étape (*i.e* à chaque ajout d'une arête).

Posons $P(i)$ le poids du nœud i qui correspond au coût de broadcast de i dans l'arbre (*i.e* : son rayon d'émission actuel). À chaque étape de BIP, les poids des nœuds changent : en effet, à chaque étape on ajoute une arête et on change potentiellement les arêtes déjà choisies. Chaque nœud dans l'arbre de BIP (dont celui nouvellement ajouté) change de poids s'il devient père d'un nouveau nœud qui l'oblige à augmenter son rayon d'émission. On a donc :

$$\begin{cases} P(i) = 0 & \text{si } i \text{ est une feuille ou que } i \text{ n'est pas encore dans l'arbre} \\ P(i) = \max_{j \in N_1(i) \cap BIP} (P_{ij}) & \text{sinon} \end{cases}$$

Le coût d'ajout d'une arête dépend des nœuds déjà dans l'arbre de la façon suivante :

$$\forall i \in BIP, \forall j \notin BIP, P'_{ij} = P_{ij} - P(i)$$

où P_{ij} est le coût réel de transmission (poids de l'arête). P'_{ij} représente donc le coût d'ajout de j comme fils du nœud i . La paire $\{i, j\}$ minimisant P'_{ij} avec $i \in BIP$ et $j \notin BIP$ est sélectionnée et i ajuste son rayon pour transmettre à j . Ainsi, une nouvelle arête est ajoutée à chaque étape de l'algorithme.

Algorithm 6 Procédure de construction du BIP-Tree

ENTREES : $G = (V, E)$ un graphe connexe, s une source, des coûts P_{ij} sur les arêtes
 SORTIE : Arbre BIP de racine s
 B : ENSEMBLE des arêtes de l'arbre
 $B \leftarrow \text{arbre}_{vide}$
 $P_s = \min_{j \in N_1(s)} P_{sj}$
 Créer les nouveaux poids : $\forall j \notin BIP, P'_{sj} = P_{sj} - P(s) = P_{sj}$
while il existe un sommet non marqué adjacent à un sommet marqué **do**
 Mettre à jour les poids : $\forall i \in BIP, \forall j \notin BIP, P'_{ij} = P_{ij} - P(i)$
 Sélectionner un sommet j non marqué adjacent à un sommet marqué i tel que (i, j) est l'arête sortante de plus faible poids P'_{ij}
 $B := B \cup (i, j)$
 Marquer j
end while
 Retourner $B = (V, B)$

Contrairement à Prim, qui garantie l'optimalité de l'arbre couvrant en termes de coût total, BIP ne construit pas forcément un arbre de poids minimum. Cependant, contrairement à Prim, BIP exploite l'avantage notoire du broadcast dû aux transmissions radio. Un fois cet arbre construit, le broadcast se fait naturellement via celui-ci. Chaque sommet connaît l'arbre global, sa position et ses fils si il en a.

Algorithm 7 BIP

ENTREES : $G = (V, E)$ un graphe connexe, s une source, un message M
SORTIE : message broadcasté
 s envoie M à ses fils dans son BIP-tree
Lors de réception de M par un nœud i :
if i n'est pas une feuille de l'arbre **then**
 Retransmettre M à ses fils
else
 ne rien faire
end if

Découverte des 1- et 2-voisinages

Description : la connaissance du 2-voisinage est un très bon compromis conservant la localité du protocole tout en minimisant le nombre de messages.

Algorithm 8 Découverte 2-voisinage

for chaque nœud i **do**
 Broadcaster un message de type $\langle \text{HELLO}, i \rangle$ avec un rayon de transmission γ
end for
A la réception de $\langle \text{HELLO}, j \rangle$, ajouter j à son 1-voisinage.
Attendre Δ
for chaque nœud i **do**
 Broadcaster $\langle \text{HELLO}, i, N_1(i) \rangle$ contenant $N_1(i)$ le 1-voisinage de i
 A la réception de $\langle \text{HELLO}, j, N_1(j) \rangle$, ajouter $N_1(j)$ à son 2-voisinage.
end for

Consommation : Pour une 2-découverte, $2n$ messages de rayon γ circulent, soit un consommation de $2n \cdot \gamma^\alpha + n \cdot c$. Pour une 1-découverte, $2n$ messages de rayon γ circulent, soit un consommation de $n \cdot \gamma^\alpha + n \cdot c$.

Localized Broadcast Incremental-power Protocol [15]

Description : LBIP est l'application locale de BIP : au lieu de construire l'arbre de diffusion BIP de façon centralisée, il le construit de façon locale. Chaque site, lorsqu'il reçoit un message à retransmettre, calcul son BIP-tree sur son 2-voisinage à partir des informations du paquet et diffuse le message à ses voisins dans l'arbre.

Algorithm 9 LBIP

ENTREES : $G = (V, E)$ un graphe connexe, s une source, un message M
SORTIE : message broadcasté
REQUIERE : connaissance du 2-voisinage
 s calcul son arbre de BIP avec comme entrée son graphe de 2-voisinage
 s ajoute dans le paquet les identifiants des nœuds qui devront retransmettre le message
 s ajoute également les identifiants des nœuds qui devront être atteints par ceux qui retransmettent
 s broadcast $\langle M, s \rangle$ avec un rayon suffisant pour atteindre tous ses fils dans l'arbre
if u reçoit $\langle M, v \rangle$ **then**
 if le paquet contient des instructions pour u **then**
 soit v le nœud que u doit atteindre d'après les instructions du paquet
 u construit son BIP-tree à partir des informations du paquet :
 l'arbre est construit à partir du 2-voisinage de u auquel on enlève :
 -le nœud s source du broadcast
 -les voisins de s non compris dans le rayon de transmission que le paquet donne à u
 -le nœud v qui a redirigé le paquet vers u
 -les voisins de v non compris dans le rayon de transmission que le paquet donne à u
 -les voisins de u non compris dans le rayon de transmission que le paquet donne à u
 pour chaque paire de nœuds (r_i qui doit retransmettre, t_i qui doit être atteint)
 dans la construction de l'arbre de BIP on initialise $P_{r_i} = P_{r_i t_i}$
 u remplace dans le paquet les instructions pour les nœuds voisins
 u broadcast $\langle M, s \rangle$ avec un rayon suffisant pour atteindre tous ses fils dans l'arbre
 end if
end if

Dynamic Localized Broadcast Incremental-power Protocol [5]

Description : DLBIP est une amélioration de LBIP. Le principe est de répartir l'énergie consommée lors d'un broadcast en empreintant des chemins (arbre couvrants) différents en fonction de l'énergie restantes des noeuds qui vont servir de relais. A chaque broadcast, l'arbre de diffusion est recalculé sur le même graphe mais avec de nouveaux poids dépendants de l'énergie de communication entre les noeuds ainsi que de l'énergie restante de chaque noeud.

Ce protocole permet que le niveau d'énergie propre à chaque noeud baisse globalement de façon identique et ainsi retarder un maximum la panne d'un capteur par manque d'énergie. DLBIP actualise dynamiquement ces coûts P'_{ij} à chaque broadcast :

$$\forall i, j \in V, P'_{ij} = \frac{P_{ij}}{E_i}$$

ou P_{ij} est le coût réel de transmission et E_i l'énergie restante du nœud i .

Il existe deux méthode pour permettre d'actualiser les E_i : soit périodiquement en même temps que la mise à jour des informations locales, soit en les ajoutant dans les messages diffusés lors du broadcast.

Durée de vie : L'avantage de DLBIP est de limiter le phénomènes de chemins de fourmis lorsque les sources des broadcast sont toujours les mêmes.

LMST Broadcast Oriented Protocol(LBOP) et RNG Broadcast Oriented Protocol(RBOP) [3]

Description : Le protocole LBOP se base sur le contrôle de topologie pour réduire le nombre d'arcs, donc le nombre de nœuds inondés par la diffusion ainsi que le rayon de transmission tout en maintenant la connectivité du réseau. Pour cela, il prend comme graphe de départ le graphe dit Graphe LMST obtenu on appliquant l'algorithme LMST.

Algorithme LMST : Le but est de construire un arbre couvrant minimum local (LMST). Chaque nœud connaît la position de ses voisins à 1 saut et chaque nœud calcule son propre MST (Minimum Spanning Tree) sur son voisinage. Puis, si un nœud u est dans le LMST calculé par un nœud v et que le nœud v est dans le LMST du nœud u alors le lien (u, v) est retenu dans le LMST global. La construction du LMST global est ainsi basée sur la construction locale d'un MST par chaque nœud.

Algorithme MST :

Algorithm 10 LBIP

ENTREES : $G = (V, E)$ un graphe connexe, s une source, un message M
 SORTIE : message broadcasté
 REQUIERE : connaissance du 1-voisinage
 s calcul son arbre de BIP avec comme entrée son graphe de 2-voisinage
 s ajoute dans le paquet les identifiants des nœuds qui devront retransmettre le message
 s ajoute également les identifiants des nœuds qui devront être atteints par ceux qui retransmettent
 s broadcast $\langle M, s \rangle$ avec un rayon suffisant pour atteindre tous ses fils dans l'arbre
if u reçoit $\langle M, v \rangle$ **then**
 if le paquet contient des instructions pour u **then**
 soit v le nœud que u doit atteindre d'après les instructions du paquet
 u construit son BIP-tree à partir des informations du paquet :
 l'arbre est construit à partir du 2-voisinage de u auquel on enlève :
 le nœud s source du broadcast
 les voisins de s non compris dans le rayon de transmission que le paquet donne à u
 le nœud v qui a redirigé le paquet vers u
 les voisins de v non compris dans le rayon de transmission que le paquet donne à u
 les voisins de u non compris dans le rayon de transmission que le paquet donne à u
 pour chaque paire de nœuds $(r_i$ qui doit retransmettre, t_i qui doit être atteint)
 dans la construction de l'arbre de BIP on initialise $P_{r_i} = P_{r_i t_i}$
 u remplace dans le paquet les instructions pour les nœuds voisins
 u broadcast $\langle M, s \rangle$ avec un rayon suffisant pour atteindre tous ses fils dans l'arbre
 end if
end if

LMST&RNG Broadcast Oriented Protocol (LBOP & RBOP) [3]

Description : LBOP et RBOP se basent sur le contrôle de topologie pour réduire le nombre d'arcs, donc le nombre de nœuds inondés par la diffusion ainsi que le rayon de transmission tout en maintenant la connectivité du réseau. LBOP et RBOP sont des protocoles de diffusion s'appuyant respectivement sur l'arbre couvrant de coût minimum local (LMST) et le graphe de voisinage (RNG).

-Le graphe RNG peut être obtenu rapidement de façon locale et nécessite la connaissance du 1-voisinage. Etant donné que chaque site connaît sa position et celle de son 1-voisinage, il peut construire son graphe RNG sur celui-ci. La réunion de tous ses sous-graphes est exactement RNG(G). Une fois ce graphe construit, le message est diffusé via celui-ci, chaque nœud transmettant à 'ses voisins les plus proches' n'ayant pas encore reçu le message.

-LMST est la version distribuée pour calculer un arbre couvrant de coût minimum. Cependant LMST n'est pas forcément de coût minimum. Le but est de construire un arbre couvrant minimum local. Chaque nœud connaît la position de ses 1-voisins et chaque nœud calcule son propre MST (Minimum Spanning Tree) sur ce voisinage. Pour calculer le graphe LMST, chaque site u construit $LMST(N_1(u))$:
 $(u, v) \in LMST(N_1(u)) \Leftrightarrow v \in N_{LMST(N_1(u))}(u)$ et $u \in N_{LMST(N_1(v))}(v)$
 Avec $N_{LMST(N_1(u))}(v)$ les voisins de v dans l'arbre couvrant calculé sur son 1-voisinage (avec PRIM par

Algorithm 11 RNG

```
for chaque noeud  $u$  do
  Appel de Découverte de 1-voisinage
end for
for chaque noeud  $u$  do
   $RNG(u) \leftarrow \text{graphe}_{vide}$ 
  for chaque noeud  $v \in N_1(u)$  do
    if  $\nexists w \in N_1(u) \cap N_1(v) : d(u, w) < d(u, v) \wedge d(v, w) < d(u, v)$  then
      Ajouter  $v$  à  $RNG(u)$ 
    end if
  end for
end for
return  $\bigcap_{u \in G} RNG(u)$ 
```

Algorithm 12 LMST

```
for chaque noeud  $u$  do
  Appel de Découverte de 1-voisinage
   $u$  calcul  $MST(N_1(u))$  et l'envoi à ses voisins
end for
for chaque noeud  $u$  do
  for chaque noeud  $v \in N_1(u)$  do
    if  $v \in N_{MST(N_1(u))}(u)$  et  $u \in N_{MST(N_1(v))}(v)$  then
      Ajouter  $v$  à  $LMST(u)$ 
    end if
  end for
end for
return  $\bigcap_{u \in G} (LMST(u))$ 
```

exemple).

Une fois l'arbre construit, le message est diffusé via celui-ci, chaque nœud père transmettant à ses fils.

Algorithm 13 RBOP

```
ENTREES :  $G = (V, E)$  un graphe connexe,  $s$  une source, un message  $M$ 
SORTIE : message broadcasté
REQUIERE : connaissance du 1-voisinage
 $s$  calcul  $RNG(U)$  et envoie  $M$  avec pour rayon  $\max_{v \in N_{RNG(s)}} (d(s, v))$ 
if  $u$  reçoit  $\langle M, v \rangle$  then
  if  $M$  est un nouveau message then
    -si  $v \in RNG(u) : RNG(u) \leftarrow RNG(u) \setminus (RNG(u) \cap RNG(v))$   $u$  envoie  $M$  avec pour rayon
       $\max_{w \in N_{RNG(u)}} (d(u, w))$ 
    -sinon : attendre  $\Delta$ , si  $RNG(u)$  est non vide  $u$  envoie  $M$  avec pour rayon  $\max_{w \in N_{RNG(u)}} (d(u, w))$ 
  end if
  if  $M$  a déjà été reçu par  $u$  then
    -si  $u$  a déjà retransmit ce message :  $M$  est ignoré
    - $RNG(u) \leftarrow RNG(u) \setminus (RNG(u) \cap RNG(v))$  :
      -si  $RNG(u)$  est vide :  $M$  est ignoré
      -si  $RNG(u)$  est non vide :  $u$  envoie  $M$  avec pour rayon  $\max_{w \in N_{RNG(u)}} (d(u, w))$ 
  end if
end if
```

En remplaçant $RNG(u)$ par $LMST(u)$ dans l'algorithme de RBOP, nous obtenons LBOP.

Comme $LMST(G)$ est un sous graphe de $RNG(G)$, LBOP a de meilleures performances que RBOP.

Target Radius LMST Broadcast Oriented Protocol [13]

Description : TRLBOP est une amélioration de LBOP. Le principe est de choisir les noeuds relais pour qu'ils se rapprochent le plus possible d'un maillage hexagonal de rayon optimal théorique $R_{opt} = \sqrt[\alpha]{\frac{2c}{\alpha-2}}$. Au lieu d'envoyer le message M avec pour rayon $\max_{w \in N_{LMST}(u)} (d(u, w))$, le rayon choisi est la distance entre u et un voisins non encore couverts par le broadcast qui est le plus proche de R_{opt} .

Soient :

$$L(u) = N_{LMST}(u)$$

$$L'(u) = N_1(u) \setminus L(u)$$

$$D_L = \max(d(u, v) \mid v \in L(u))$$

$$D_{L'} = (d(u, v) \mid v \in L(u) \cup L'(u) \wedge \delta_{uv} = \min(\delta_{uv} \mid w \in L(u) \cup L'(u))) \text{ avec } \delta_{uv} = |d(u, v) - R_{opt}|$$

Ainsi, un sommet u retransmettra avec pour rayon d'émission $\max(D_L, D_{L'})$.

2.4.1 Synthèse

Tableau récapitulatif

algorithme	mode d'émission		portée		rayon d'émission		balisage		décision	
	broadcast	single-cast	globale	locale	fixe	variable	avec	sans	déterministe	probabiliste
FA [7]	✗	✓	✗	✓	✗	✓	✗	✓	✓	✗
FR [7]	✗	✓	✓	✗	✗	✓	✗	✓	✓	✗
EAR [20]	✗	✓	✗	✓	✗	✓	✓	✗	✗	✓
blind flooding	✓	✗	✗	✓	✓	✗	✗	✓	✓	✗
probabilistic flooding	✓	✗	✗	✓	✓	✗	✗	✓	✗	✓
ABBA [18]	✓	✗	✗	✓	✓	✗	✗	✓	✓	✗
BIP [22]	✓	✗	✓	✗	✗	✓	✓	✗	✓	✗
LBIP [13]	✓	✗	✗	✓	✗	✓	✓	✗	✓	✗
DLBIP [5]	✓	✗	✗	✓	✗	✓	✓	✗	✓	✗
LBOP [3]	✓	✗	✗	✓	✗	✓	✓	✗	✓	✗
RBOP [3]	✓	✗	✗	✓	✗	✓	✓	✗	✓	✗
TR-LBOP [13]	✓	✗	✗	✓	✗	✓	✓	✗	✓	✗

Chapitre 3

Analyse et reflexion

3.1 Critique de l'existant

3.1.1 Consommation de l'énergie

Le modèle énergétique choisi dans beaucoup d'articles (dont ceux présentant les algorithmes cités dans l'état de l'Art) n'est pas du tout réaliste. Hors, la conception d'algorithmes a pour but de maximiser la durée de vie des réseau de capteurs sur le terrain. Pour être fidèle au contraintes réelle associées à la nature des capteurs, il faudrait avant tout choisir un type de capteur (Ultra Wideband, Wi-fi) dépendant de l'application voulu et, en se basant sur ses données constructeur prendre en considération :

- l'énergie de capture : traitement de signal et activation de la sonde de capture.
- L'énergie de traitement : l'énergie d'alimentation et l'énergie de veille. L'énergie d'alimentation est déterminée par la tension d'alimentation de l'unité de traitement. L'énergie de veille correspond au courant de fuite lorsque le capteur est en mode veille. à l'énergie consommée lorsque l'unité de calcul n'effectue aucun traitement.
- L'énergie de communication : l'énergie de réception et l'énergie de l'émission. Cette énergie est déterminée par la taille des message à communiquer et la distance de transmission, ainsi que par les propriétés physiques du module radio. L'émission d'un signal est caractérisée par sa puissance.

3.2 Nos idées

3.2.1 Application du rayon optimal à DLBIP : TR-DLBIP

DLBIP offre de bonne performances puisqu'il se base sur BIP qui est à ce jour l'algorithme le plus efficace en matiere de broadcast. Nous l'avons modifié légèrement en lui appliquant l'idée du rayon optimale.

Algorithm 14 TR-DLBIP

ENTREES : $G = (V, E)$ un graphe connexe, s une source, un message M

SORTIE : message broadcasté

REQUIERE : connaissance du 2-voisinage

$$R_{opt} = \sqrt[\alpha]{\frac{2c}{\alpha-2}}$$

Mise à jour des couts des messages : $\forall i, j \in V, P'_{ij} = \frac{P_{ij}}{E_i}$

s calcul $LBIP(s) = BIP - Tree(s, N_2(s))$

s ajoute dans le paquet les identifiants des nœuds qui devront retransmettre le message

s ajoute également les identifiants des nœuds qui devront être atteints par ceux qui retransmettent

s broadcast $\langle M, s \rangle$ avec un rayon suffisant pour atteindre tous ses fils dans l'arbre

if u reçoit $\langle M, v \rangle$ de w **then**

if le paquet contient des instructions pour u **then**

 soit v le nœud que u doit atteindre d'après les instructions du paquet

u construit $LBIP(u) = BIP - Tree(s, N_2(u) \setminus \{w, \text{les sommets atteints par } w, \text{les sommets plus loin que } d_e(u, v)\})$:

u remplace dans le paquet les instructions pour les nœuds voisins

u broadcast $\langle M, s \rangle$ avec pour rayon :

- R_{opt} si $\max_{v \in N_u(1) \cap LBIP(u)} < R_{opt}$
- $\max_{v \in N_u(1) \cap LBIP(u)} (d_e(u, v))$ sinon.

end if

end if

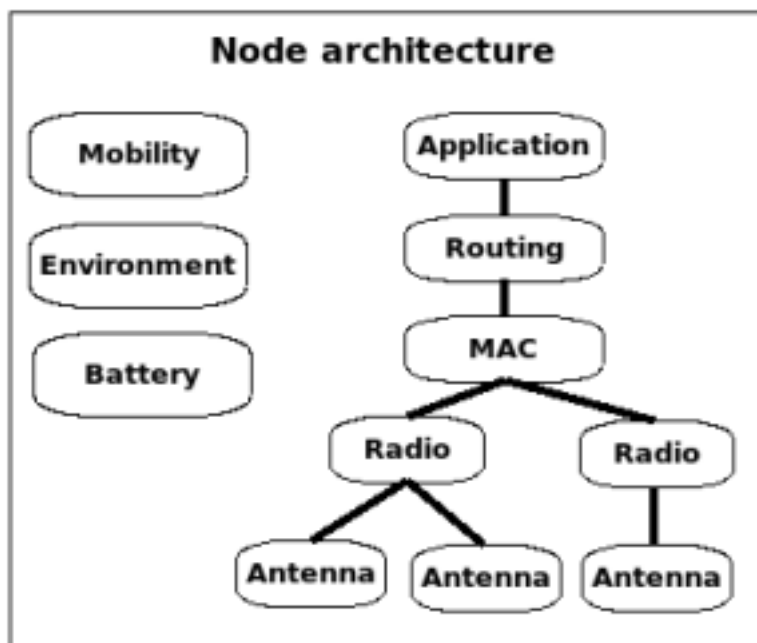
3.2.2 Algorithme qui préserve la connexité du graphe

Chapitre 4

Simulations et résultats

4.1 Outils de programmation

Nos contraintes d'implémentation lors de ce TER étaient d'utiliser le simulateur de réseaux [WSNET](#). Ce simulateur (décrit comme un "simulateur évènementiel pour les grands réseaux de capteurs sans fils") est constitué d'un ensemble de modules en langage *C*. Ces modules correspondent aux différentes couches réseaux (voir figure 4.1). Cette bibliothèque est développée par trois chercheurs lyonnais : Guillaume



Chelius, (chercheur INRIA), Antoine Fraboulet (maitre de conférence) et Elyes Ben Hamida (doctorat).

Dans le cadre de notre TER, nous avons développé onze modules supplémentaires pour WSNET :

- un module utilitaire contenant des structures de données
- un module gérant la consommation énergétique
- un module gérant la couche liaison (protocole mac) sans interférences mais qui appelle le module de consommation d'énergie
- sept modules de routage (le cœur de notre TER)
 - uni-cast : Flow Augmentation [7]
 - diffusion aveugle
 - RBOP [3]

- LBOP [3]
- BIP [22]
- LBIP [15]
- DLBIP [5]
- un module gérant la couche application et permettant la diffusion depuis plusieurs sources

Nous avons également écrit différents scripts *bash* et *C++* afin d'automatiser les phases de tests ainsi que la collecte des résultats. Enfin, nous avons développé un visualisateur de graphes sous *Qt* afin de pouvoir vérifier la pertinence de nos résultats sur une topologie donnée.

4.2 Paramètres de simulation

À l'aide de tous ces outils, nous avons pu nous lancer dans les simulations des différents algorithmes de routage. Pour cela, nous avons fixé différents paramètres :

nombre de capteurs (nœuds du graphe)	100
rayon d'émission maximum	$30m$
taille de la zone où sont répartis les capteurs	$1000 \times 1000m^2$
durée de simulation	$10000s$
temps entre deux diffusions	$2s$

Ces paramètres sont communs à toutes nos simulations. Cependant, certains autres paramètres varient : la topologie du réseau, les constantes α et c du modèle énergétique et la quantité d'énergie initiale des nœuds.

4.3 Topologie aléatoire

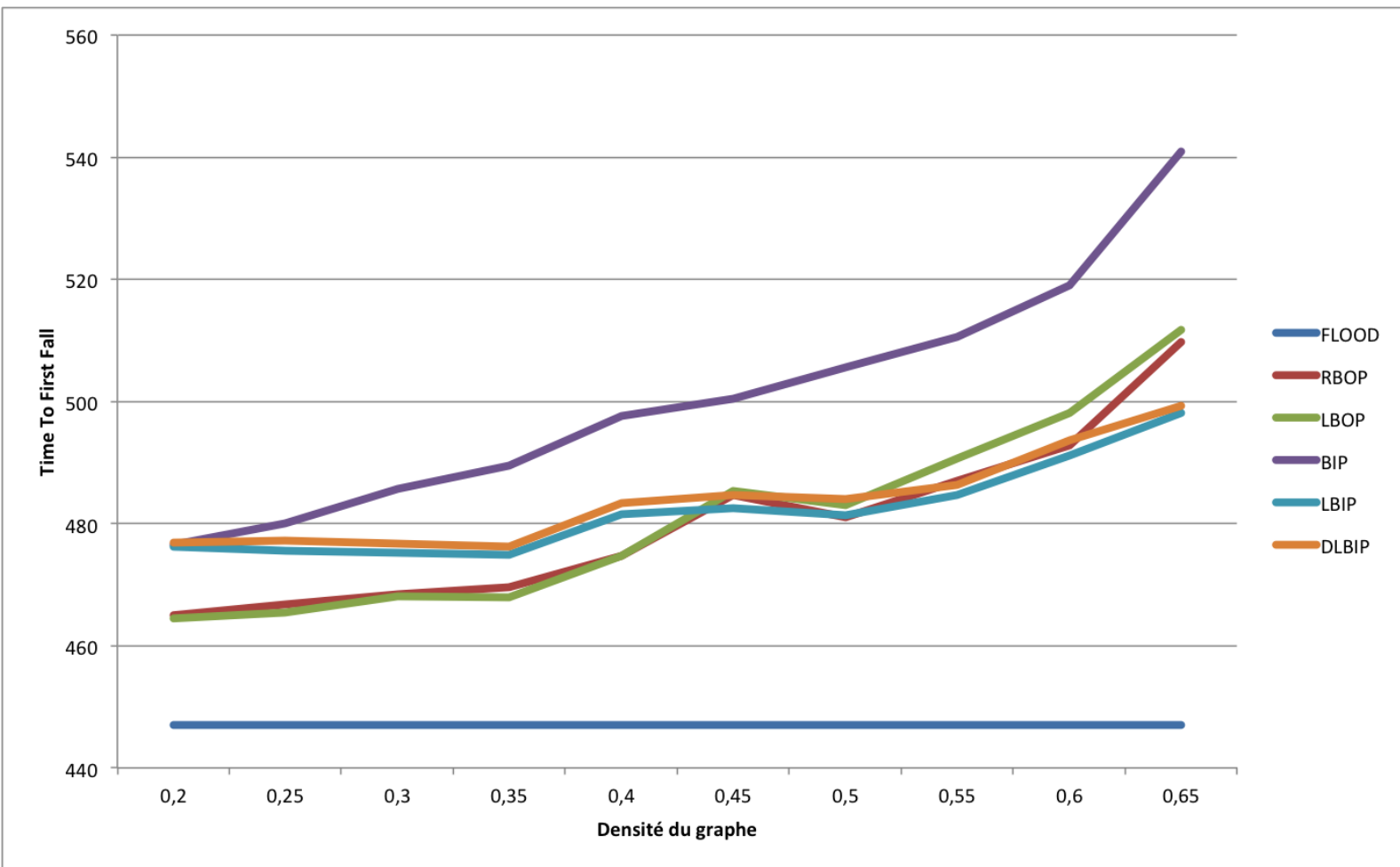
Ces phases de test ont été effectuées en utilisant un générateur de graphes aléatoires connexes que nous avons écrit en *C++*. Cette topologie permet de se rapprocher au mieux du contexte général des réseaux de capteurs.

Le modèle énergétique que nous avons choisi d'utiliser fixe la consommation énergétique lors de l'envoi d'information à $r^\alpha + c$. Nous avons lancé deux phases de tests avec des valeurs différentes pour les constantes α et c .

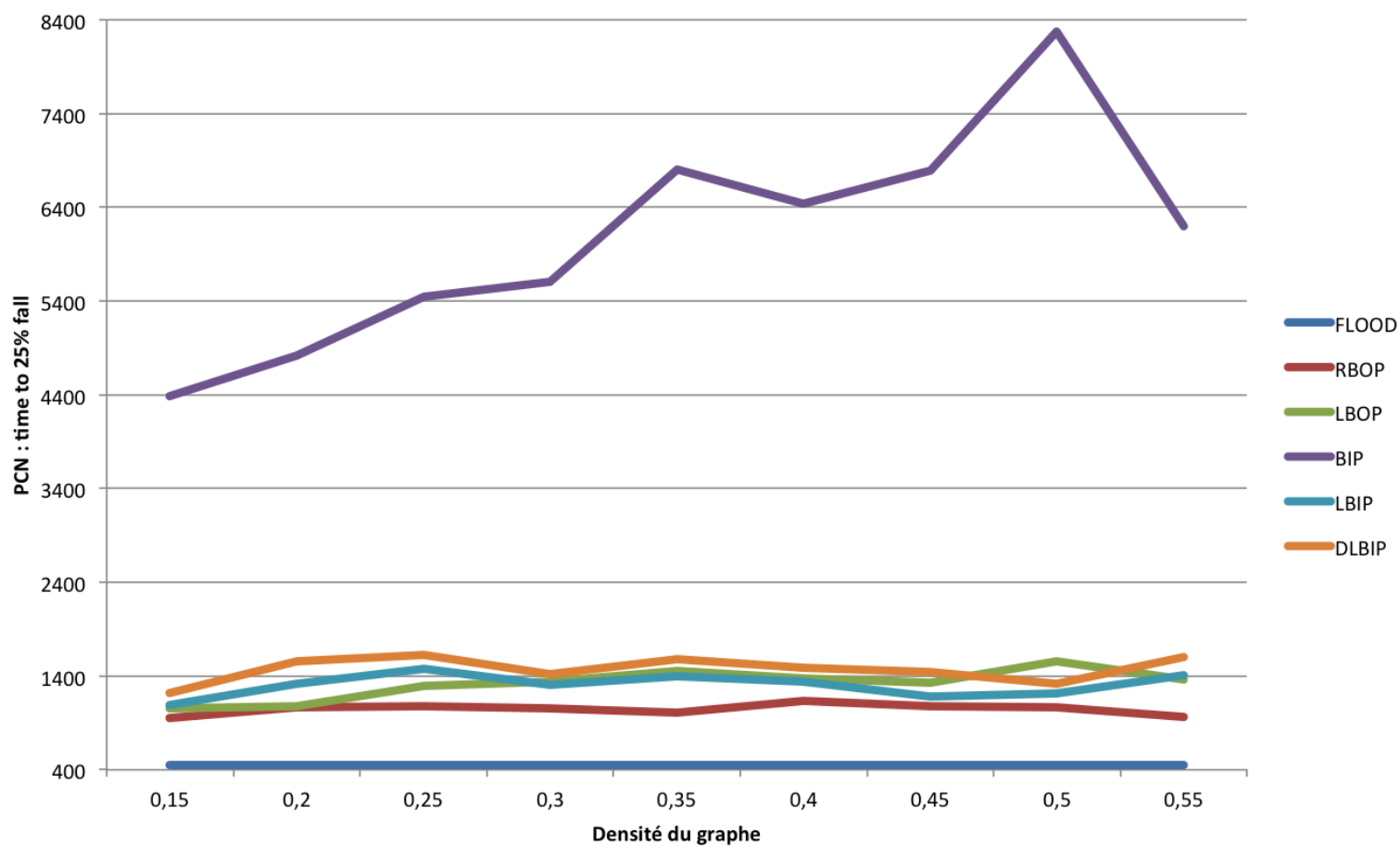
4.3.1 Modèle énergétique simple

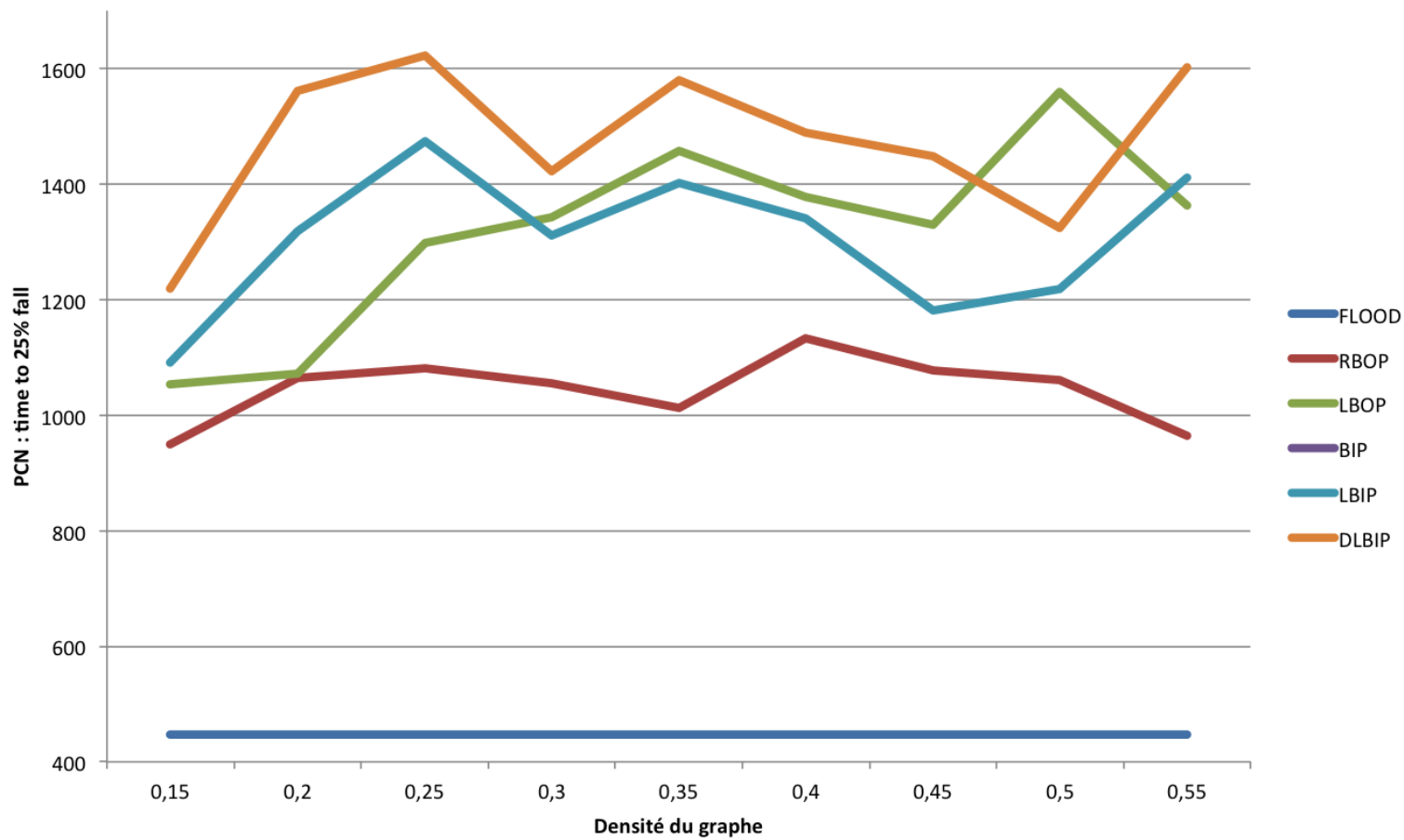
Le modèle énergétique le plus simple (mais aussi le plus éloigné de la réalité) utilise les valeurs suivantes pour les constantes énergétiques : $\alpha = 2$ et $c = 0$.

Time To First Fall

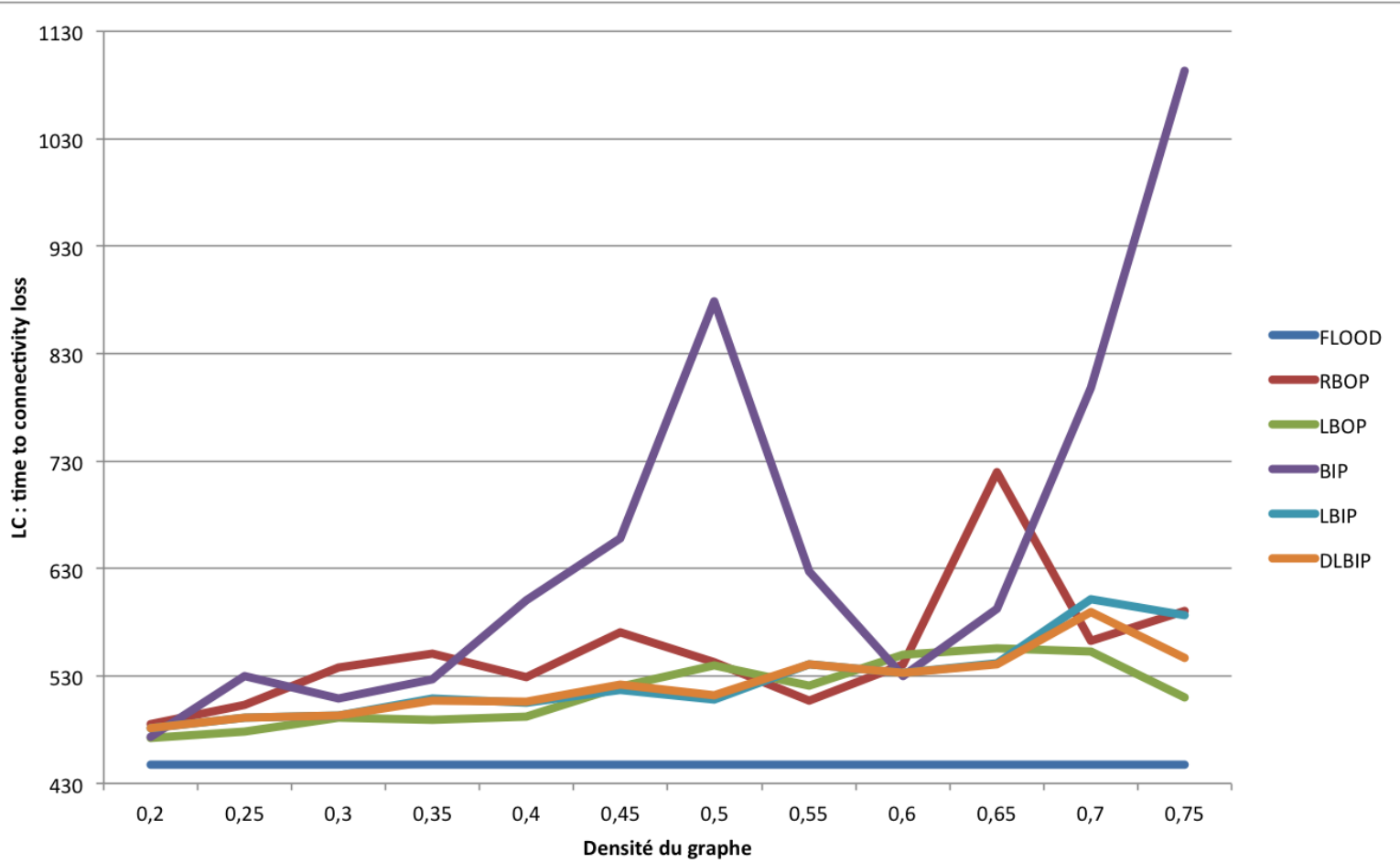


PerCent Node





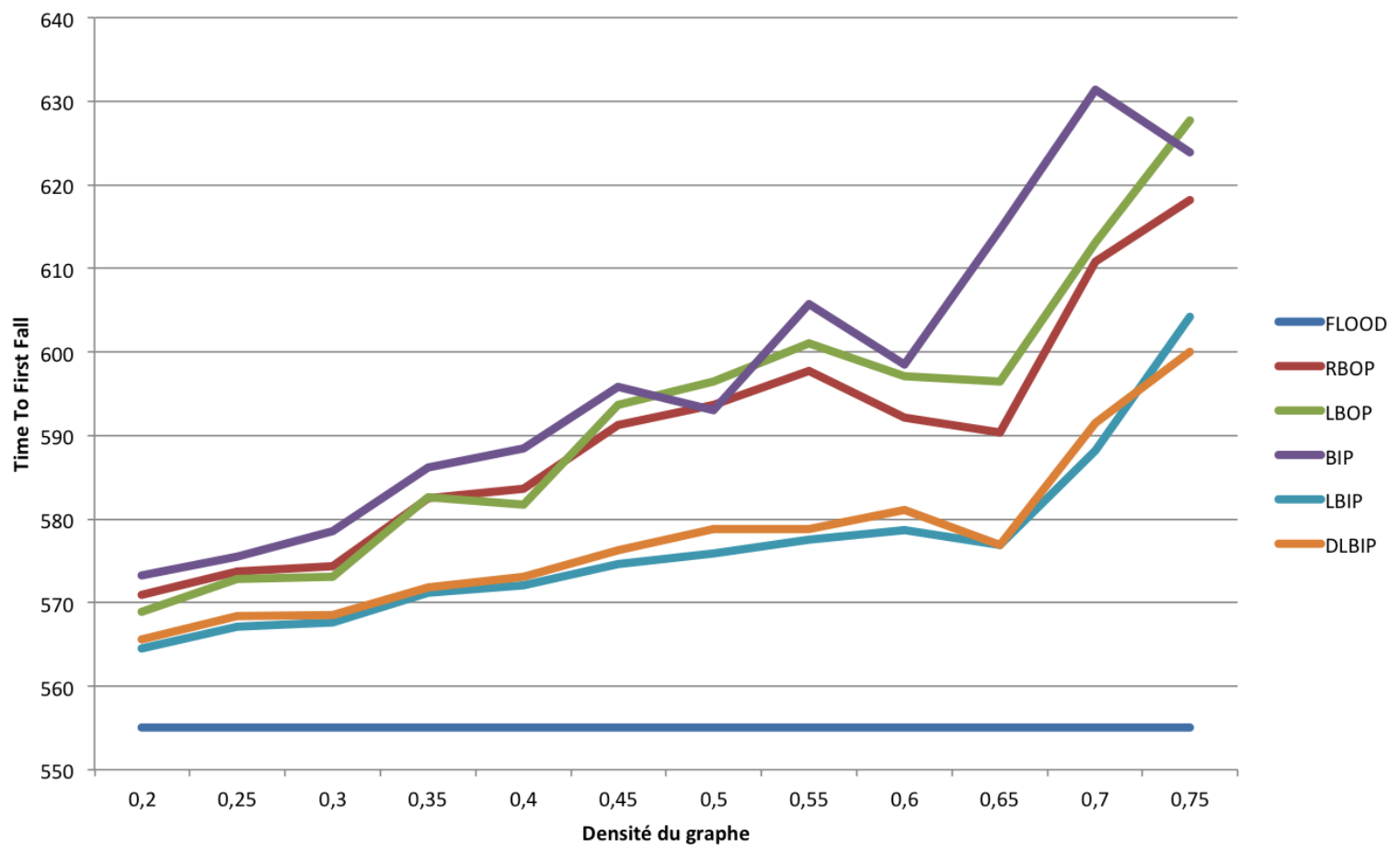
Lose Connectivity



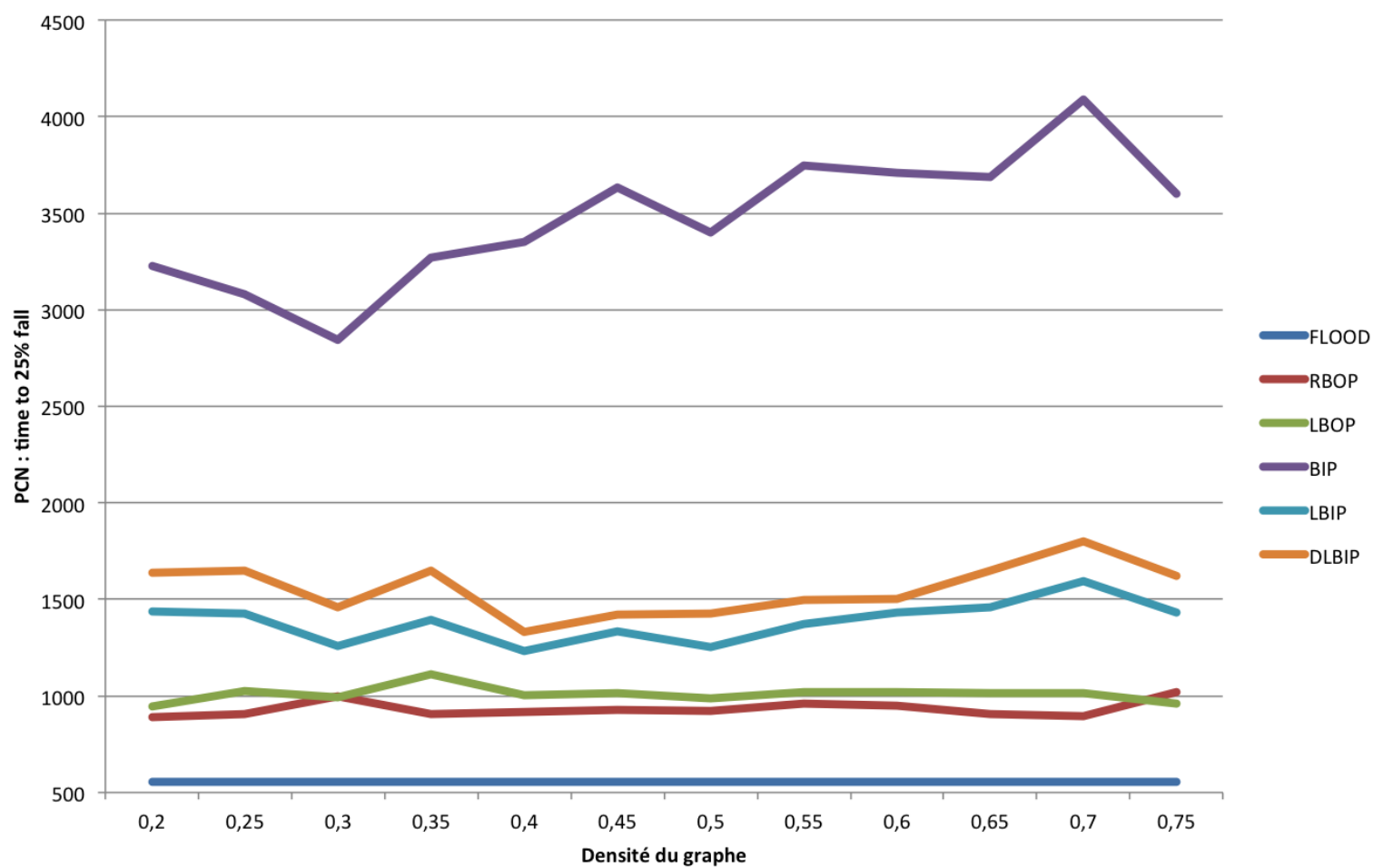
4.3.2 Modèle énergétique plus réaliste

Ce modèle énergétique un peu plus réaliste utilise les valeurs suivantes pour les constantes énergétiques : $\alpha = 4$ et $c = 10^6$.

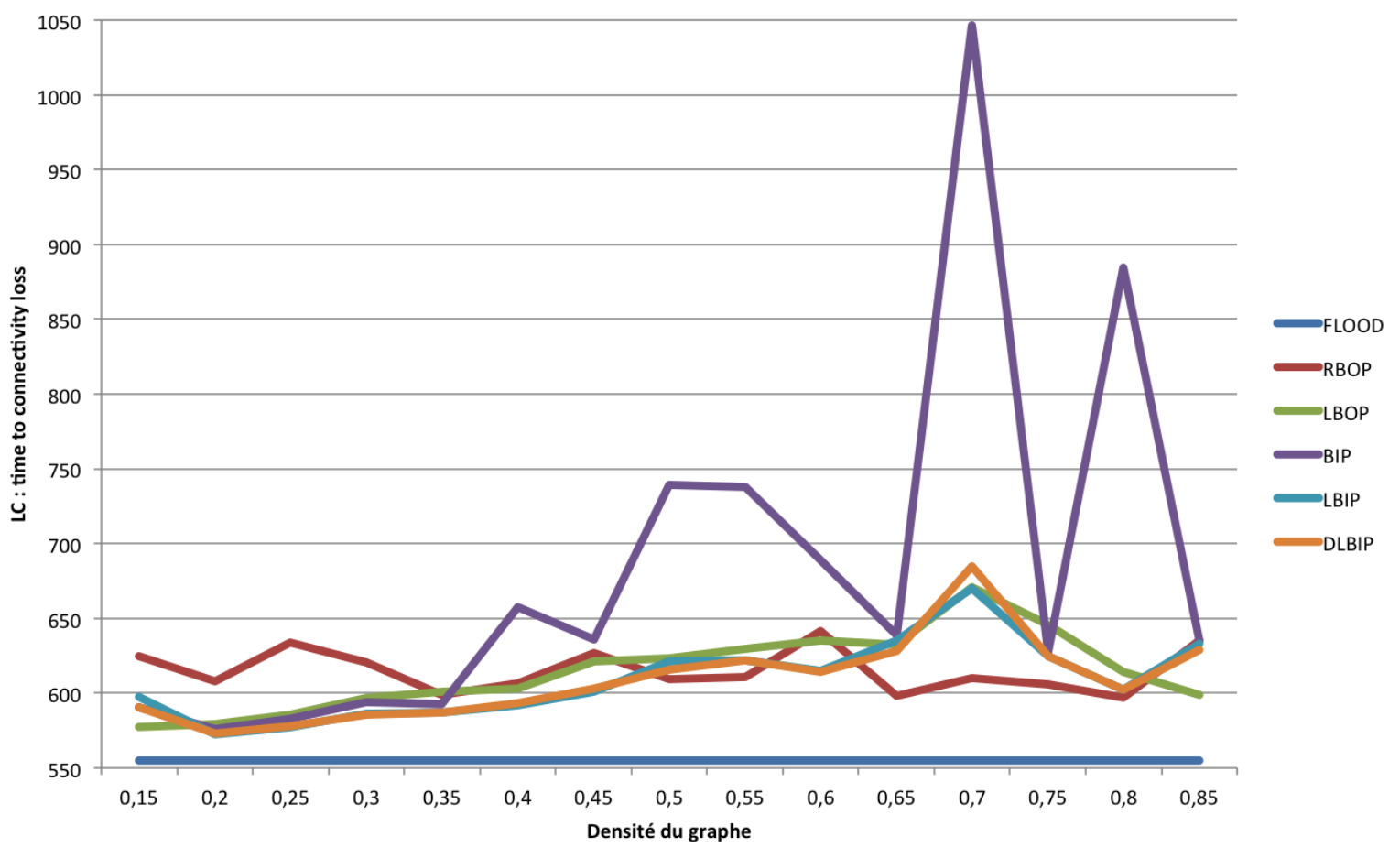
Time To First Fall



PerCent Node



Lose Connectivity



Chapitre 5

Conclusion

5.1 Bilan de notre travail de groupe

La diffusion est primordiale dans les réseaux de capteurs sans fils. Dans ce TER, nous avons présenté le problème, puis établi un cadre formel nous permettant de présenter et de simuler quelques algorithmes. Notre travail de groupe a été planifié par le diagramme de GANTT (5.1) que nous avons élaboré.

Nous nous sommes tout d'abord entendus pour que chacun lise deux articles différents afin d'approfondir notre vision d'ensemble du sujet. Puis nous avons constitué 2 sous-groupes : le premier qui commençait la programmation des principaux algorithmes de routage et le second qui approfondissait les recherches. Quelques semaines plus tard, étant satisfaits de l'avancement de la programmation, nous avons décidé de diminuer le nombre de programmeurs et réorienté les ressources sur la rédaction du rapport.

Pour le bon déroulement de notre travail, nous avons utilisé *GIT* et *github* comme moyens de travail collaboratif. Nous avons également organisé des réunions régulières (toutes les une ou deux semaines). Entre ces réunions, nous avons communiqué par mails.

[illegible]

5.2 Bilan du TER

5.2.1 Problèmes rencontrés

Durant notre travail, nous avons rencontré quelques problèmes dus à l'ambiguïté du thème, et ceci particulièrement dans la désignation des choix des algorithmes qui convergent le plus vers notre sujet. Ceci s'ajoute à la difficulté de la prise en main de GIT et de \LaTeX puisque la moitié du groupe n'y était pas familiarisé.

5.2.2 Contributions

Dans le cadre de ce TER, nous avons contribué à établir une classification des algorithmes existants. Nous avons également présentés les différents modèles théoriques utilisés et mis en évidence le manque de cohérence des différentes publications. Enfin, nous avons développé différents modules pour WSNET et comparé les performances des principaux algorithmes de routage dans les réseaux de capteurs.

Bibliographie

- [1] Manish Agarwal, Lixin Gao, Joon Ho Cho, and Jie Wu. Energy Efficient Broadcast in Wireless Ad hoc Networks with Hitch-hiking. *MONET*, 10(6) :897–910, 2005.
- [2] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. Wireless sensor networks : a survey. *Computer Networks*, 4(38) :393–422, 2002.
- [3] J. Cartigny, F. Ingelrest, D. Simplot-Ryl, and I. Stojmenovic. Localized LMST and RNG based minimum-energy broadcast protocols in ad hoc networks. *Ad Hoc Networks*, 3(1) :1–16, 2005.
- [4] J. Cartigny, D. Simplot, and I. Stojmenovic. Localized minimum-energy broadcasting in ad-hoc networks. In *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies*, volume 3, pages 2210–2217. IEEE, 2003.
- [5] J. Champ, A.E. Baert, and V. Boudet. Dynamic localized broadcast incremental power protocol and lifetime in wireless ad hoc and sensor networks. *Wireless and Mobile Networking*, pages 286–296, 2009.
- [6] J. Champ, C. Saad, and A.E. Baert. Lifetime in wireless sensor networks. In *Complex, Intelligent and Software Intensive Systems, 2009. CISIS'09. International Conference on*, pages 293–298. IEEE, 2009.
- [7] Jae-Hwan Chang and Leandros Tassiulas. Energy conserving routing in wireless ad-hoc networks. *INFOCOM*, pages 22–31, 2000.
- [8] J. Deng, Y. S. Han, W. B. Heinzelman, and P. K. Varshney. Scheduling sleeping nodes in high density cluster-based sensor networks. *ACM/Kluwer Mobile Networks and Applications (MONET), Special Issue on “Energy Constraints and Lifetime Performance in Wireless Sensor Networks”*, 2005.
- [9] Isabel Dietrich and Falko Dressler. On the lifetime of wireless sensor networks. *TOSN*, 5(1), 2009.
- [10] Q. Dong. Maximizing system lifetime in wireless sensor networks. *IPSN '05 : Proceedings of the 4th international symposium on Information processing in sensor networks*, page 3, 2005.
- [11] Abdelrahman Elleithy and Gonhsin Liu. A simulation model for the lifetime of wireless sensor networks. *CoRR*, abs/1201.2237, 2012.
- [12] S. Giordano, I. Stojmenovic, and L. Blazeovic. Position based routing algorithms for ad hoc networks : a taxonomy. *Ad Hoc Wireless Networking*, pages 103–136, 2003.
- [13] F. Ingelrest, D. Simplot-Ryl, and I. Stojmenovic. Target transmission radius over LMST for energy-efficient broadcast protocol in ad hoc networks. In *Communications, 2004 IEEE International Conference on*, volume 7, pages 4044–4049. IEEE, 2004.
- [14] F. Ingelrest, D. Simplot-Ryl, and I. Stojmenović. Energy-efficient broadcasting in wireless mobile ad hoc networks. *Resource Management in Wireless Networking*, pages 543–582, 2005.
- [15] François Ingelrest and David Simplot-Ryl. Localized broadcast incremental power protocol for wireless ad hoc networks. *Wirel. Netw.*, 14 :309–319, June 2008.
- [16] O. Kasten. Energy consumption, 2001.
- [17] W. Liang. Constructing minimum-energy broadcast trees in wireless ad hoc networks. In *Proceedings of the 3rd ACM international symposium on Mobile ad hoc networking & computing*, pages 112–122. ACM, 2002.
- [18] Francisco Javier Ovalle-Martínez, Amiya Nayak, Ivan Stojmenovic, Jean Carle, and David Simplot-Ryl. Area-based beaconless reliable broadcasting in sensor networks. *IJSNet*, 1(1/2) :20–33, 2006.
- [19] R.C. Prim. Shortest connection networks and some generalizations. *Bell system technical journal*, 36(6) :1389–1401, 1957.

- [20] R. C. Shah and J. M. Rabaey. Energy aware routing for low energy ad hoc sensor networks. *Energy aware routing for low energy ad hoc sensor networks*, 1 :350–355, 2002.
- [21] I. Stojmenovic and J. Wu. Broadcasting and activity scheduling in ad hoc networks. *Mobile Ad Hoc Networking*, pages 205–229, 2004.
- [22] Jeffrey E. Wieselthier, Gam D. Nguyen, and Anthony Ephremides. On the Construction of Energy-Efficient Broadcast and Multicast Trees in Wireless Networks. In *INFOCOM*, pages 585–594, 2000.
- [23] J. Wu and W. Lou. Forward-node-set-based broadcast in clustered mobile ad hoc networks. *Wireless Communications and Mobile Computing*, 3(2) :155–173, 2003.