

Université de Montréal

**Optimisation et Balancement de la Consommation
d'énergie dans les réseaux ad hoc mobiles et de
capteurs**

Par

Farid Chender

Département d'informatique et de recherche opérationnelle
Faculté des arts et des sciences

Mémoire présenté à la Faculté des études supérieures
en vue de l'obtention du grade de Maître ès sciences (M.Sc)
en informatique

Décembre 2006

Copyright © Farid Chender, 2006



AVIS

L'auteur a autorisé l'Université de Montréal à reproduire et diffuser, en totalité ou en partie, par quelque moyen que ce soit et sur quelque support que ce soit, et exclusivement à des fins non lucratives d'enseignement et de recherche, des copies de ce mémoire ou de cette thèse.

L'auteur et les coauteurs le cas échéant conservent la propriété du droit d'auteur et des droits moraux qui protègent ce document. Ni la thèse ou le mémoire, ni des extraits substantiels de ce document, ne doivent être imprimés ou autrement reproduits sans l'autorisation de l'auteur.

Afin de se conformer à la Loi canadienne sur la protection des renseignements personnels, quelques formulaires secondaires, coordonnées ou signatures intégrées au texte ont pu être enlevés de ce document. Bien que cela ait pu affecter la pagination, il n'y a aucun contenu manquant.

NOTICE

The author of this thesis or dissertation has granted a nonexclusive license allowing Université de Montréal to reproduce and publish the document, in part or in whole, and in any format, solely for noncommercial educational and research purposes.

The author and co-authors if applicable retain copyright ownership and moral rights in this document. Neither the whole thesis or dissertation, nor substantial extracts from it, may be printed or otherwise reproduced without the author's permission.

In compliance with the Canadian Privacy Act some supporting forms, contact information or signatures may have been removed from the document. While this may affect the document page count, it does not represent any loss of content from the document.

Université de Montréal
Faculté des études supérieures

Ce mémoire intitulé :

Optimisation et Balancement de la Consommation d'Énergie dans les Réseaux Ad hoc
mobiles et de capteurs

Présenté par :
Farid Chender

a été évalué par un jury composé des personnes suivantes :

El moustapha Aboulhamid
Président-rapporteur

Abdelhakim Hafid
Directeur de recherche

Houari Sahraoui
Membre du jury

Mémoire accepté le 2 avril 2007

Résumé

L'inondation classique est très utilisée dans les réseaux ad hoc mobiles sans fil et de capteurs pour déterminer des chemins et acheminer les données. Cette inondation génère des surcharges dans le réseau et engendrent un gaspillage d'énergie au niveau des nœuds qui ne sont pas requis dans la construction de nouveaux chemins. Plusieurs techniques existent pour réduire les effets négatifs de l'inondation; les plus notables sont basées sur le concept de groupage. Le groupage est le regroupement de plusieurs nœuds pour former une nouvelle structure appelée cluster. Le représentant de chaque groupe est dynamiquement choisi comme chef de groupe; les nœuds qui appartiennent à deux groupes sont appelés nœuds de liaison. Seuls les chefs de groupe et nœuds de liaison sont impliqués dans l'inondation. Il existe deux types de groupage : (1) groupage actif (AC) : il nécessite un protocole de signalisation et introduit une surcharge pour la formation des groupes et la sélection des chefs de groupes et nœuds de liaison; et (2) groupage passive (PC) : il n'utilise aucun protocole de signalisation et n'introduit qu'une surcharge minimale.

Dans la première partie de ce projet, nous proposons un protocole de groupage passif géographique (GPC), qui utilise PC avec les informations de localisation (coordonnées géographiques) de tous les nœuds ou seulement une partie des nœuds du réseau. GPC permet d'inonder seulement une partie du réseau et ainsi réduit la surcharge et évite un gaspillage d'énergie inutile.

Dans la deuxième partie, nous avons introduit une heuristique appelée N-Interval-based Load Balancing Heuristic (N-ILBH) pour balancer la consommation d'énergie des nœuds dans le réseau qui utilisent PC. Elle permet une distribution uniforme de la consommation d'énergie sur la majorité des nœuds, un taux de livraison de paquet meilleur que PC, et une durée de vie plus élevée du réseau.

Les simulations, pour évaluer GPC et N-ILBH, ont été réalisées avec le simulateur GloMoSim (Global mobile simulation).

Mots-clés: Chef de groupe, nœud de liaison, groupage, inondation, balancement de la charge.

Abstract

The traditional flooding technique is widely used in sensors and mobile ad hoc networks to find a feasible path to a given destination. This technique generates considerable overhead and thus causes high energy consumption by the network nodes. To reduce overhead clustering techniques have been proposed. The basic idea of clustering is to select a set of cluster heads among the nodes in the network and cluster the rest of the nodes with these heads. Gateways are chosen to connect clusters. Then, only cluster heads and gateways perform flooding. Most of clustering protocols make use of active signaling, i.e., exchanging extra-packets, for the formation of the cluster structure. There is one clustering protocol, called Passive Clustering (PC), which does not require explicit signaling and efficiently controls the number of flooding packets on the fly, without requiring any prior structure.

In the first part of this project, we propose a protocol, called Geography-based Passive Clustering (GPC), which uses (a) passive clustering to limit the flooding to cluster heads and gateways and (b) geographic coordinates, whenever available, of the network nodes to limit the number of cluster heads and gateways involved in the flooding. Simulations show that GPC outperforms existing protocols; it generates far less overhead and thus reduces considerably energy consumption of the network nodes.

In the second part, we propose an N-Interval-based Load Balancing Heuristic (N-ILBH) for PC which forces cluster heads and gateway nodes to give up their duty to ordinary nodes which have a larger amount of residual energy. The idea is to balance the energy consumption of the network nodes and thus prevent cluster heads and gateways from consuming all their energy while ordinary nodes have plenty of residual energy; this may cause network disconnection. Simulations show that N-ILHB allows balancing energy consumption of the network nodes and thus increasing the lifetime of the network.

Keywords: Cluster head, gateway, ordinary node, cluster, clustering, flooding, load balancing.

Table des matières

RÉSUMÉ -----	III
ABSTRACT -----	IV
TABLE DES MATIÈRES -----	V
LISTE DES TABLEAUX -----	VII
LISTE DES FIGURES -----	VIII
LISTE DES SIGLES ET DES ABRÉVIATIONS -----	X
REMERCIEMENTS -----	XI
DÉDICACE -----	XII
CHAPITRE 1: INTRODUCTION -----	- 1 -
1.1 PROBLÉMATIQUE ET MOTIVATION -----	- 1 -
1.2 CONTRIBUTIONS-----	- 3 -
1.3 PLAN DU MÉMOIRE-----	- 4 -
CHAPITRE 2 INONDATION DANS LES RÉSEAUX AD HOC MOBILES ET DE CAPTEURS -----	- 6 -
2.1 INTRODUCTION -----	- 6 -
2.2 MODÉLISATIONS DES RÉSEAUX AD HOC-----	- 7 -
2.3. INONDATION DANS LES RÉSEAUX AD HOC -----	- 8 -
2.4 INONDATION DANS LES PROTOCOLES DE ROUTAGE PROACTIFS -----	- 10 -
2.4.1 Inondation dans OLSR-----	- 10 -
2.5 INONDATION DANS LES PROTOCOLES RÉACTIFS-----	- 12 -
2.5.1 Inondation dans AODV -----	- 13 -
2.6 LES PROTOCOLES DE ROUTAGE GÉOGRAPHIQUES -----	- 15 -
2.6.1 Le protocole GPSR-----	- 16 -
2.7 LES RÉSEAUX DE CAPTEURS -----	- 18 -
2.7.1 Consommation d'énergie-----	- 19 -
2.7.2 Architecture -----	- 19 -
2.7.3 Le capteur-----	- 20 -
2.7.4 Énergie d'un nœud capteur-----	- 21 -
2.8 ROUTAGE DANS LES RÉSEAUX DE CAPTEURS-----	- 21 -
2.8.1 Address-centric Protocol (AC) -----	- 22 -
2.8.2 Data-centric Protocol (DC) -----	- 22 -
2.8.3 Protocoles basés localisation-----	- 23 -
2.9 CONCLUSION-----	- 24 -

CHAPITRE 3 ALGORITHMES DE GROUPEMENT (CLUSTERING) -- 25 -

3.1 INTRODUCTION -----	25 -
3.2 ALGORITHMES DE GROUPEMENT -----	25 -
3.2.1 Algorithmes de groupement actif-----	26 -
3.2.2 Groupement passif-----	30 -
3.3 CONCLUSION-----	37 -

CHAPITRE 4 GROUPEMENT PASSIF GÉOGRAPHIQUE ----- 38 -

4.1 INTRODUCTION -----	38 -
4.2 GPC-----	39 -
4.2.1 Approche basée sur les cercles -----	39 -
4.2.2 Approche basée sur les rectangles-----	44 -
4.2.3 Format de la trame dans GPC-----	45 -
4.3 SIMULATEUR GLOMOSIM-----	45 -
4.3.1 Fonctionnement de GloMoSim-----	47 -
4.4 SIMULATIONS-----	48 -
4.5 RÉSULTATS DES SIMULATIONS -----	49 -
4.6 CONCLUSION-----	61 -

CHAPITRE 5 BALANCEMENTS DE LA CHARGE DANS UN RÉSEAU MOBILE SANS FIL----- 62 -

5.1 INTRODUCTION -----	62 -
5.2 1-ILBH -----	62 -
5.3 SIMULATIONS-----	66 -
5.4 DURÉE DE VIE DU RÉSEAU -----	71 -
5.5 N-ILHB -----	73 -
5.6 SIMULATIONS ET RÉSULTATS-----	76 -
5.7 CONCLUSION-----	81 -

CHAPITRE 6 CONCLUSION----- 83 -

6.1 PERSPECTIVES -----	84 -
------------------------	------

BIBLIOGRAPHIE ----- 86 -

Liste des tableaux

Tableau 3.1. Pseudo code de PC -----	34 -
Tableau 4.1. Pseudo code de GPC : Première variante -----	41 -
Tableau 4.2. Pseudo code de GPC: Deuxième variante -----	43 -
Tableau 4.3. Pseudo code de la variation de la zone géographique -----	44 -
Tableau 5.1. Pseudo code du balancement-----	65 -
Tableau 5.2. Pseudo code de N-ILHB-----	76 -
Tableau 5.3. Récapitulatif des résultats de comparaison-----	81 -

Liste des figures

Figure 2.1. Modélisation d'un réseau ad hoc	7
Figure 2.2. Inondation à l'aveugle ou classique	8
Figure 2.3. Protocoles de routage dans les réseaux mobiles ad hoc	9
Figure 2.4. Inondation classique	11
Figure 2.5. Inondation avec MPR	11
Figure 2.6.1. Recherche de chemin dans AODV	13
Figure 2.6.2. Réponse dans AODV	14
Figure 2.7. GPSR en Mode Greedy [40]	16
Figure 2.8. Maximum local dans GPSR	17
Figure 2.9. GPSR en mode périmètre	17
Figure 2.10. Réseau de capteurs [45]	19
Figure 2.11. Modèle en couche de l'architecture de réseau de capteurs [45]	20
Figure 2.12. Routage dans les réseaux de capteurs	22
Figure 3.1. Nœud 1 approche du nœud 3 et cause un changement de groupage	27
Figure 3.2. Format de la trame dans PC	31
Figure 3.3. Chefs de groupe, nœuds de liaison, et nœuds ordinaires	33
Figure 3.4. Nœuds de liaison sans heuristique	36
Figure 3.5. Nœuds de liaison avec heuristique	36
Figure 4.1. Approche basée sur le cercle	40
Figure 4.2. Approche basée sur le rectangle	45
Figure 4.3. Trame dans GPC	45
Figure 4.4. Couches de simulation de GloMoSim	47
Figure 4.5. Fonctionnement de GloMoSim	48
Figure 4.6.1. Variation du nombre de chefs de groupe Vs. Nœuds	50
Figure 4.6.2. Variation du nombre de nœuds de liaison Vs. Nœuds	50
Figure 4.7. Énergie consommée des chefs de groupe et des nœuds de liaisons Vs. Nœuds	50
Figure 4.8. Consommation d'énergie Vs. Sessions	51
Figure 4.9.1. GPC : Énergie Vs. Pourcentage de nœuds	51
Figure 4.9.2. PC : Énergie Vs. Pourcentage de nœuds	52
Figure 4.10. Énergie totale Vs. Nœuds	52
Figure 4.11. Chefs de groupe et nœuds de liaison Vs. Sessions	53
Figure 4.12. Nombre requêtes de route Vs. Nœuds	53
Figure 4.13. Chef de groupe et nœuds de liaison Vs. Vitesse	54
Figure 4.14. Ratio de livraison Vs. Nœuds	55
Figure 4.15. Collisions pour PC et GPC Vs. Nœuds	55
Figure 4.16. Énergie Vs. Nœuds connus	56
Figure 4.17. Débit Vs. Charge	57
Figure 4.18. RREQ Vs. Mobilité	57
Figure 4.19. Taux Vs Mobilité	58
Figure 4.20. Nœuds Vs. Temps	58
Figure 4.21. Paquets reçus et RREQ Vs. Temps	59
Figure 4.22. Énergie totale Vs. Nœuds	59
Figure 4.23. RREQ Vs. Nœuds	60
Figure 4.24. Taux de livraison Vs. Nœuds	60

Figure 4.25. Énergie Vs. Sessions -----	60 -
Figure 5.1. Distribution de la consommation de l'énergie pour PC (10 sessions) ----	67 -
Figure 5.2. 1-ILHB: $\alpha = 0.1$, $\beta = 0.9$. -----	67 -
Figure 5.3. 1-ILHB : $\alpha = 0.2$, $\beta = 0.8$ -----	68 -
Figure 5.4. 1-ILHB : $\alpha = 0.3$, $\beta = 0.5$ -----	68 -
Figure 5.5. 1-ILHB : $\alpha = 0.4$, $\beta = 0.7$ -----	69 -
Figure 5.6. 1-ILHB: $\alpha = 0.5$, $\beta = 0.6$ -----	69 -
Figure 5.7. 1-ILHB : $\alpha = 0.6$, $\beta = 0.7$ -----	69 -
Figure 5.8. Énergie totale consommée pour les différentes valeurs de α et β -----	70 -
Figure 5.9. Taux de livraison Vs. α et β -----	70 -
Figure 5.10. Nombre de paquets RREQ Vs. α et β -----	71 -
Figure 5.11. Perte de connectivité/couverture dans le réseau -----	72 -
Figure 5.12. Nombre de nœuds morts Vs. Temps -----	72 -
Figure 5.13. Taux de livraison Vs. Temps -----	73 -
Figure 5.14.1. 1-ILHB Figure 5.14.2. 2-ILHB -----	77 -
Figure 5.14.3. 3-ILHB Figure 5.14.4. 4-ILHB -----	77 -
Figure 5.15. Mort de nœuds Vs. temps-----	78 -
Figure 5.16. Taux de livraison Vs. temps -----	79 -
Figure 5.17. Nombre moyen de paquets reçus par destination Vs. Temps -----	80 -
Figure 5.18. Paquets de contrôle Vs. Heuristique-----	80 -

Liste des sigles et des abréviations

AODV	Ad hoc On-Demand Distance Vector Routing
CBR	Constant Bit Rate
CSL	cluster Sub Layer
DARPA	Defence Advanced Research Projects Agency
DREAM	Distance Routing Effect Algorithm for Mobility
DSDV	Destination-Sequenced Distance-Vector Routing Protocol
DSR	Dynamic Source Routing
GloMoSim	Global Mobile Simulation
GPSR	Greedy Perimeter Stateless Routing
HDC	Highest Degree Clustering
J	Joule
LAR	Location Aided Routing
LCA	linked Cluster Algorithm
LID	Lowest Identifier
LLC	Logical Link Control
LT	Location Table
MAC	Medium Access Control
MFR	Most Forward within R
MPR	Multipoint Relaying
MWIS	Maximal Weighted Independent Set
NFP	Nearest Forward progress
OLSR	Optimized Link State routing
OSI	Open System Interconnection
PC	Passive Clustering
RREP	Response Request
RREQ	de route
TBRPF	Topology Diffuse Based on Reverse-Path Forwarding
TTL	Time To Live
WCA	Weighted clustering Algorithm
WSN	Wireless Sensor Network

Remerciements

Je tiens à remercier chaleureusement toute personne qui a contribué de près ou de loin à ce travail.

A cet effet, j'adresse mes sincères remerciements au professeur Abdelhakim Hafid, mon directeur de recherche à l'Université de Montréal, pour ses conseils et ses directives enrichissants.

Dédicace

À ma famille, mes amis et tous ceux qui m'ont soutenu tout au long de ce travail.

Chapitre 1: Introduction

Dans ce chapitre, nous présentons la problématique d'intérêt pour notre travail. Ensuite, nous décrivons les contributions de ce travail. Finalement, nous présentons le plan de ce mémoire.

L'inondation est le processus de livraison de paquets à tous les nœuds connectés au réseau. Cette technique est utilisée dans les réseaux ad hoc pour la découverte de chemins et pour collecter des informations sur la topologie du réseau mais génère des pertes inutiles d'énergie des nœuds (tous les nœuds sont impliqués dans la recherche de nouveaux chemins).

Les réseaux ad hoc sont des réseaux sans fil capables de s'organiser sans infrastructure définie préalablement. Les réseaux ad hoc, dans leur configuration mobile, sont connus sous le nom de MANet (pour Mobile Ad hoc NETWORKS). Chaque entité (noeud) communique directement avec sa voisine. Pour communiquer avec d'autres entités, il lui est nécessaire de faire passer ses données par d'autres qui se chargeront de les acheminer. Pour cela, il est d'abord primordial que les entités se situent les unes par rapport aux autres, et soient capables de construire des routes entre elles : c'est le rôle du protocole de routage.

Le fonctionnement d'un réseau ad hoc le différencie notablement d'un réseau via un modem Wi-Fi : là où une ou plusieurs stations de base sont nécessaires à la plupart des communications entre les différents nœuds du réseau, les réseaux ad hoc s'organisent d'eux-mêmes et chaque entité peut jouer différents rôles.

L'utilisation la plus simple et la plus courante des réseaux ad hoc est faite par les réseaux sans fil Wi-Fi en permettant une mise en place rapide d'une connexion réseau entre deux ordinateurs.

1.1 Problématique et motivation

Plusieurs papiers [1, 2, 3] parlent des limitations de l'inondation à l'aveugle ou classique utilisée dans les réseaux ad hoc mobiles sans fil pour la découverte de chemins, l'acheminement des données ou la collection des informations sur la topologie du réseau. Dans cette inondation, tous les nœuds du réseau reçoivent et envoient les

paquets de données, sans restriction, même s'ils ne sont pas concernés par ces données dans la recherche d'un chemin; ainsi, beaucoup de nœuds perdent leur énergie inutilement. La ressource énergie est cruciale dans les réseaux ad hoc, surtout dans les réseaux de capteurs qui ont une petite autonomie d'énergie (en général, les nœuds sont munis de batteries avec une petite autonomie).

Pour minimiser les effets négatifs de cette inondation classique, beaucoup de travaux ont été réalisés dans ce sens. Les auteurs dans [1, 2, 3, 4] ont proposé des solutions pour une inondation efficace qui limite le nombre de nœuds inondés entre les sources et les destinations. Les travaux les plus importants sur l'inondation se sont concentrés sur le développement de mécanismes qui sélectionnent un sous ensemble de nœuds pour diffuser les données. Une classe des solutions les plus importantes se basent sur les techniques de groupage.

Le groupage [1, 7, 10] est le regroupement de plusieurs nœuds pour former une nouvelle structure appelée cluster. Le représentant de chaque groupe est dynamiquement choisi comme chef de groupe en utilisant certains critères ou poids calculés en fonction de plusieurs paramètres (p.e. vitesse, rayon de couverture, énergie du nœud, état des voisins...) afin de différencier les nœuds et leurs tâches respectives. Les nœuds qui sont à un saut du chef de groupe lui sont associés et sont sous son contrôle. Un nœud qui appartient à deux ou plusieurs groupes (clusters) en même temps est appelé nœud de liaison (gateway) et sert de liaison entre plusieurs chefs de groupes (clusterhead). Les autres nœuds sont les nœuds ordinaires qui écoutent, reçoivent les paquets de données mais ne diffusent jamais les paquets à leurs voisins. Seuls les chefs de groupes et les nœuds de liaison peuvent diffuser des données.

Avec la technique du groupage, les nœuds qui expédient les paquets sont les chefs de groupe et les nœuds de liaison seulement. Le groupage a fait l'objet, dans les réseaux ad hoc mobiles, de plusieurs études pour le routage hiérarchique [7], contrôle de puissance [8], diffusion fiable [6] et diffusion efficace [1]. Cependant, l'architecture de groupage a été rarement utilisée pour une inondation efficace et ceci pour plusieurs raisons. La première est que la majorité des techniques de groupage se basent sur la connaissance complète des voisins, ce qui génère des surcharges dues aux échanges des messages 'hello' périodiques pour la découverte de voisins. La deuxième raison est que la sélection des chefs de groupes nécessite, en général, un protocole de

signalisation entre les nœuds du réseau ce qui empire le problème de surcharge. En plus, le groupage nécessite un coût élevé de maintenance en cas de mobilité élevée (p.e. nouveaux groupes formés et nouveaux chefs sélectionnés). Ces limitations [9, 10] ont motivé des chercheurs à définir un nouveau protocole de formation de groupe appelé groupage passif (PC).

Le groupage passif (PC) [10, 11] diffère des autres techniques de groupage, appelés techniques de groupage actifs (AC), du fait qu'il se base sur l'information collectée localement au niveau du nœud et ne nécessite aucune connaissance préalable des voisins, pour un nœud donné, et ne fait appel à aucun protocole de signalisation pour la sélection des chefs de groupes. En effet, PC ne nécessite pas des messages de contrôle périodique 'hello'; il exploite les paquets de données échangés pour la formation des groupes et la sélection des chefs de groupes et de nœuds de liaison. Même si PC permet une amélioration considérable par rapport aux autres approches, nous avons identifiés deux limitations/améliorations importantes en relation avec PC :

(a) PC ne tient pas en compte les coordonnées géographiques des nœuds du réseau durant l'inondation. Nous croyons que l'intégration des coordonnées géographiques, quand elles existent, des nœuds du réseau, pourra réduire le nombre de nœuds (principalement chefs de groupes et nœuds de liaison) impliqués dans la diffusion et ainsi la consommation globale de l'énergie du réseau; et

(b) L'exécution de PC crée une situation où les nœuds ordinaires ont un niveau d'énergie résiduel beaucoup plus important que le niveau d'énergie des chefs de groupes et de nœuds de liaison puisqu'ils n'expédient jamais les paquets reçus. Cette situation pourrait causer le partitionnement du réseau (chefs de groupes meurent, i.e., énergie résiduelle est égale à zéro, bien avant les nœuds ordinaires) et ainsi réduire la vie du réseau. Nous croyons que des techniques/heuristiques de balancement de charge dans le réseau permettront une distribution uniforme de la consommation d'énergie par les nœuds et ainsi une augmentation de la vie du réseau.

1.2 Contributions

Notre travail comporte deux contributions clés : (1) un protocole de groupage passif qui intègre les coordonnées géographiques des nœuds, appelé GPC (Geographic Passive

Clustering); et (2) un ensemble d'heuristiques de balancement de charge dans le réseau, appelée N-Interval-based Load Balancing Heuristic (N-ILBH)

GPC améliore la performance de PC en utilisant les coordonnées géographiques des nœuds du réseau. Plus précisément, GPC limite le nombre de nœuds qui participent à l'inondation en impliquant seulement les nœuds qui se trouvent dans la direction de la destination; ainsi la consommation globale dans le réseau d'énergie est considérablement réduite. GPC peut fonctionner même si nous ne connaissons que les coordonnées géographiques de certains nœuds du réseau; dans le cas où aucun nœud ne connaît ses coordonnées géographiques, GPC est équivalent à PC.

N-ILBH permet de balancer la consommation d'énergie entre tous les nœuds du réseau et ainsi augmenter la vie d'un réseau. En effet, tous les nœuds consommeront leur batterie à peu près en même temps (i.e., sur un petit intervalle de temps). L'idée de base de N-LLBH est de forcer un chef de groupe (ou un nœud de liaison) à céder son rôle et devenir un nœud ordinaire quand son niveau d'énergie résiduel atteint des seuils adéquatement choisis.

Les deux contributions ont été évaluées via des simulations en utilisant le simulateur GloMosim [21]; les résultats obtenus montrent et quantifient les gains apportés par nos contributions en terme d'énergie économisée dans le réseau et en terme de répartition de la consommation d'énergie sur plusieurs nœuds. Ces gains permettent d'augmenter la vie des réseaux ad hoc et en particulier les réseaux de capteurs.

1.3 Plan du mémoire

Après cette brève description de la problématique et des contributions de ce travail, le reste de ce mémoire est divisé en 5 chapitres.

Le chapitre 2 présente une revue des différents protocoles de routages utilisés dans les réseaux ad hoc mobiles et réseaux de capteurs.

Le chapitre 3 introduit la notion de groupage; ensuite, il présente une revue des protocoles et algorithmes de groupage. En particulier, la description de PC sera détaillée. Le chapitre 4 présente les détails du protocole de groupage passif géographique, appelé GPC, et son évaluation.

Le chapitre 5 définit notre heuristique N- Interval-based Load Balancing Heuristic (N-ILBH) qui permet de balancer la charge entre les différents nœuds du réseau et ainsi augmenter sa durée de vie.

Enfin, le chapitre 6 clôture ce mémoire par une conclusion et une discussion, en présentant les perspectives et les directions futures de notre travail.

Chapitre 2 Inondation dans les réseaux ad hoc mobiles et de capteurs

2.1 Introduction

Un réseau ad hoc peut être défini comme une collection d'entités mobiles interconnectées par une technologie sans fil formant un réseau temporaire sans l'aide de toute administration ou de tout support fixe. Aucune supposition ou limitation n'est faite sur la taille du réseau, cela veut dire qu'il est possible que le réseau ait une taille très grande.

Dans un réseau ad hoc, les hôtes mobiles doivent former, d'une manière ad hoc, une sorte d'architecture globale qui peut être utilisée comme infrastructure du système. Du fait que le rayon de propagation des transmissions des noeuds soit limité, et afin que le réseau ad hoc reste connecté, il se peut qu'un noeud mobile se trouve dans l'obligation de demander de l'aide à un autre noeud pour pouvoir communiquer avec son correspondant. Il se peut donc que le noeud destination soit hors de la portée de communication du noeud source, ce qui nécessite l'emploi d'un routage interne par des noeuds intermédiaires afin de faire acheminer les paquets de messages à la bonne destination.

La stratégie de routage doit prendre en considération les changements de la topologie ainsi que les autres caractéristiques du réseau ad hoc (bande passante, nombre de liens, ressources du réseau, énergie, etc.). La méthode adoptée dans le routage, doit offrir le meilleur acheminement des données en respect des différentes métriques de coûts utilisées.

Pour la recherche des chemins, la plus part des protocoles ad hoc pratiquent l'inondation à l'aveugle pour construire les chemins entre les sources et les destinations ou pour avoir des informations sur la topologie du réseau afin de construire les tables de routage. Cette inondation, utilisée par tous les protocoles proactifs et réactifs, utilise les paquets de contrôle pour la découverte de chemins qui traversent tous les noeuds du réseau pour acheminer les données ou pour avoir des informations sur le voisinage immédiat des noeuds. Cette inondation contribue à une consommation d'énergie et à la surcharge du réseau. Plusieurs solutions existent pour réduire cette inondation dans les réseaux ad hoc; nous présentons les détails de ces solutions dans les sections suivantes et le chapitre 3.

2.2 Modélisations des réseaux ad hoc

Un réseau ad hoc peut être modélisé par un graphe $G_t = (V_t, E_t)$. Où : V_t représente l'ensemble des nœuds du réseau et E_t modélise l'ensemble des connections qui existent entre ces nœuds. Si $e = (a, b) \in E_t$, alors les nœuds a et b sont en mesure de communiquer directement à l'instant t . La topologie du réseau peut changer à tout moment, elle est donc dynamique et imprévisible ce qui fait que la déconnexion des unités est très fréquente. La figure 2.1 représente un réseau ad hoc de 10 unités mobiles sous forme d'un graphe.

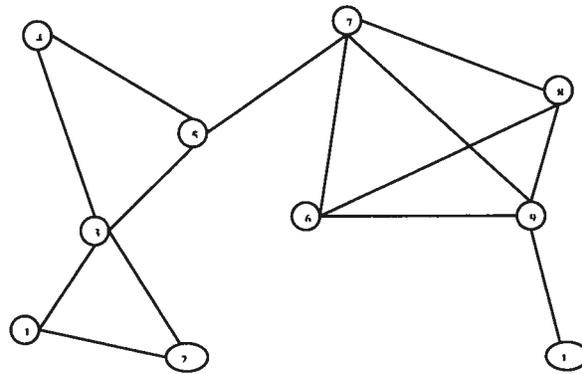


Figure 2.1. Modélisation d'un réseau ad hoc

Les réseaux mobiles ad hoc ont les caractéristiques suivantes :

- Une topologie dynamique : les unités mobiles du réseau se déplacent d'une façon libre et arbitraire. Par conséquent, la topologie du réseau peut changer, imprévisiblement, d'une manière rapide et aléatoire. Les liens de la topologie peuvent être unidirectionnels ou bidirectionnels.
- Une sécurité physique limitée : les réseaux mobiles ad hoc sont plus touchés par le paramètre de sécurité que les réseaux filaires classiques. Cela se justifie par les contraintes et limitations physiques qui font que le contrôle des données transférées doit être minimisé.
- L'absence d'infrastructure: les réseaux ad hoc se distinguent des autres réseaux mobiles par la propriété d'absence d'infrastructures préexistantes et de tout genre d'administration centralisée. Les nœuds mobiles sont responsables d'établir et de maintenir la connectivité du réseau d'une manière continue.

- Une bande passante limitée : une des caractéristiques primordiales des réseaux basés sur la communication sans fil est l'utilisation d'un médium de communication partagé. Ce partage fait que la bande passante réservée à un hôte est réduite.
- Des contraintes d'énergie : les hôtes mobiles sont alimentés par des sources d'énergie autonomes comme les batteries. Le paramètre énergie doit être pris en considération dans tout contrôle fait par le système.

2.3. Inondation dans les réseaux ad hoc

L'inondation est le processus de livraison de paquets à tous les nœuds connectés au réseau. Cette technique est utilisée dans les réseaux ad hoc pour la découverte de chemins et pour collecter des informations sur la topologie du réseau. Quand un nœud source S lance une nouvelle découverte de chemin vers le nœud destination D, il diffuse un message (requête) de découverte de route à tous ses voisins. Après la réception de ce message, un nœud X compare la destination recherchée avec ses caractéristiques; si le message lui est destiné, il envoie une réponse; sinon, il la diffuse à tous ses voisins immédiats. Pour éviter les répétitions de requêtes de route, un nœud diffuse une requête une seule fois; si la même requête est reçue une deuxième fois ou plus, il la supprime en utilisant le numéro de séquence de la requête.

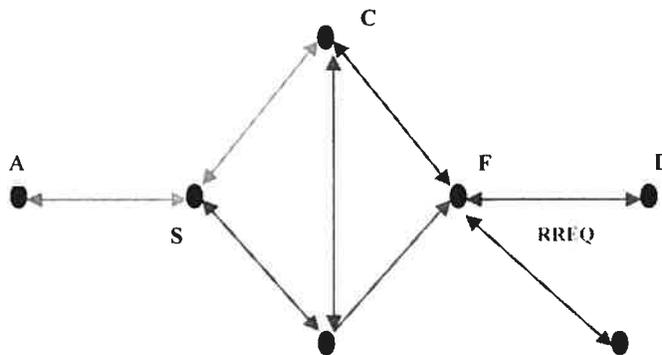


Figure 2.2. Inondation à l'aveugle ou classique

Dans la figure 2.2, le nœud S cherche le chemin menant à D. Quand les nœuds C et B reçoivent la requête de route, ils l'envoient à leurs voisins. Quand F reçoit la requête de B, il l'envoie à ses voisins. Si le nœud F reçoit la même requête du nœud C, il la

supprime tout simplement. Le chemin est gardé dans la requête de route qui est propagée à travers les nœuds du réseau. À la réception du paquet requête de route, la destination répond par le message de réponse pour la source S à travers le même chemin parcouru par la requête de route (le chemin inverse).

Il arrive que dans certaines situations, la source S ne puisse pas trouver la destination D, dans ce cas ci, il doit ré-initier une autre découverte de chemin. Un temporisateur est utilisé; à chaque fois qu'une découverte de chemin est lancée, le temporisateur est initialisé et si pendant ce laps de temps le message de réponse n'est pas reçue, une nouvelle découverte de chemin est lancée avec l'utilisation de différents numéros de séquence.

Si un nœud intermédiaire, le long du chemin, trouve que son nœud voisin (ne peut pas établir de chemin) est en panne, il envoie un message d'erreur. À la réception de ce message d'erreur par la source S, elle relance une nouvelle découverte de chemin à D. Avec l'inondation classique ou à l'aveugle, le message de découverte de route est propagé à tous les nœuds du réseau joignables de la source S; ceci contribue à la saturation du réseau et à l'effondrement du débit.

Plusieurs protocoles de routage ont été conçus pour les réseaux ad hoc; en général, ils utilisent l'inondation pour découvrir un ou plusieurs chemins entre une source et une destination. Ces protocoles se classifient selon la manière de création et de maintenance de routes lors de l'acheminement de données. Ils sont classifiés selon trois grandes catégories de base (figure 2.3). Dans les sections suivantes, nous allons définir les différentes catégories et présenter les détails (en particulier l'inondation) d'un certain nombre de protocoles (les plus populaires).

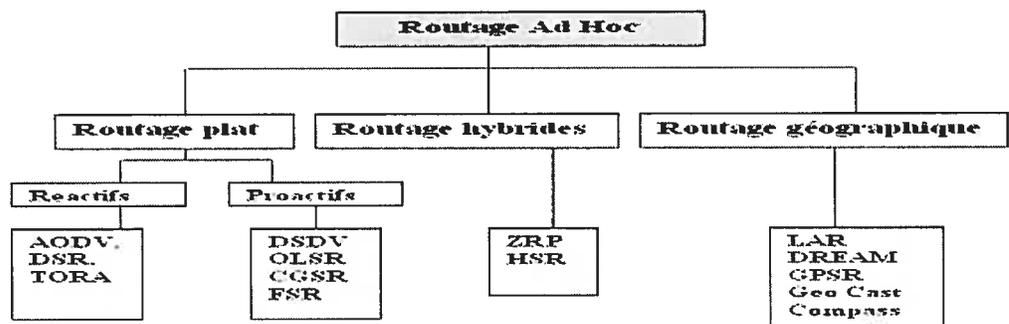


Figure 2.3. Protocoles de routage dans les réseaux mobiles ad hoc

2.4 Inondation dans les protocoles de routage proactifs

Les protocoles de routage proactifs dans les réseaux mobiles ad hoc sont basés sur la même philosophie que les protocoles de routage utilisés dans les réseaux filaires conventionnels. Les méthodes utilisées sont : la méthode état de lien [58] et la méthode vecteur de distance [59]. Les deux méthodes exigent une mise à jour périodique des données de routage qui doivent être diffusées par les différents nœuds de routage du réseau. L'algorithme "vecteur de distance" de base a été adopté pour le routage dans les réseaux ad hoc sans fil, et cela en traitant chaque hôte mobile comme un nœud de routage [60]. Le processus de calcul se répète, s'il y a un changement de la distance minimale séparant deux nœuds, et cela jusqu'à ce que le réseau atteigne un état stable. Cette technique est basée sur l'algorithme distribué de Bellman-Ford (DBF) [61]. Le principe du DBF est utilisé par une grande partie des protocoles de routage des réseaux filaires. Un problème majeur de performance de DBF est qu'il prend beaucoup de temps pour mettre à jour les tables de routage, après une partition du réseau, des défaillances de nœuds, ou lorsqu'il y a un grand nombre de nœuds dans le réseau.

Les protocoles de routage proactifs essaient de maintenir les meilleurs chemins vers toutes les destinations possibles au niveau de chaque nœud du réseau. Les routes sont sauvegardées même si elles ne sont pas utilisées. La sauvegarde permanente des chemins de routage est assurée par un échange continu des messages de mise à jour des chemins, ce qui induit un contrôle excessif, surtout dans le cas des réseaux de grande taille, et une surcharge considérable.

Dans la section suivante, nous présentons le processus d'inondation implémenté par le protocole de routage proactif OLSR [25, 56].

2.4.1 Inondation dans OLSR

OLSR utilise la technique de relais multipoints (MPRs) [17]. Les MPRs sont des nœuds choisis qui expédient des messages de diffusion pendant le processus d'inondation. Cette technique réduit sensiblement la surcharge due aux messages par rapport à un mécanisme classique d'inondation, où chaque nœud retransmet chaque message quand il reçoit la première copie du message. La figure 2.4 montre un réseau où tous les nœuds pratiquent

l'inondation classique alors que sur la figure 2.5, seulement les nœuds choisis comme relais diffusent les paquets à leurs voisins. Il est facile de voir la différence entre le volume de paquets générés dans les deux cas.

L'utilisation des MPRs permet de réduire le nombre de nœuds qui diffusent, le nombre de message de contrôle inondés dans le réseau (l'information d'état de lien est produite seulement par les MPRs).

Le message HELLO est utilisé dans OLSR pour transmettre plusieurs informations, il est envoyé par les nœuds du réseau. Il sert d'abord à découvrir l'ensemble du réseau. Il transmet ensuite l'état et le type de lien entre l'expéditeur et chaque nœud voisin. Enfin, il spécifie le MPR (Multi-Point Relays) choisi par l'expéditeur.

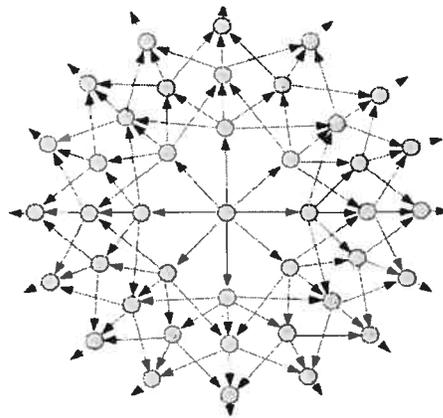


Figure 2.4. Inondation classique

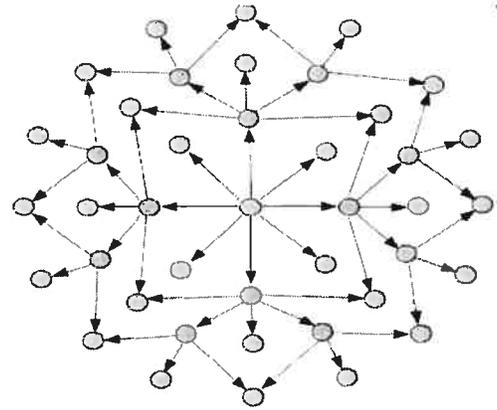


Figure 2.5. Inondation avec MPR

Les messages HELLO ne sont destinés qu'aux nœuds voisins (à un saut) de l'expéditeur; ils ne doivent jamais être routés par un MPR. Le message TC (Topology Control) permet au MPR de transmettre la liste de ses voisins qui l'ont choisi comme MPR; il sert à établir les tables de routage. Aussi, pour qu'il soit diffusé sur tout le réseau, la valeur du TTL dans l'entête du message est initialisée à 255. La figure ci-dessus illustre le format du message Hello qui contient les champs suivants :

0										1										2										3
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0
Reserved										Htime										Willigness										
Link Code					Reserved					Link Message Size																				
Neighbor Interface Address																														
Neighbor Interface Address																														
..																														
Link Code					Reserved					Link Message Size																				
Neighbor Interface Address																														
Neighbor Interface Address																														

Reserved : Ce champ doit contenir « 0000000000000000 »

Htime : Intervalle d'émission des messages HELLO.

Willigness : permet de forcer le passage d'un nœud en MPR.

Link Code : Code identifiant le type de lien entre l'expéditeur et les interfaces listées.

2.5 Inondation dans les protocoles réactifs

Les protocoles de routage réactifs représentent les protocoles les plus récents proposés dans le but d'assurer le service du routage dans les réseaux sans fil. Ils créent et maintiennent les routes selon les besoins. Lorsque le réseau a besoin d'une route, une procédure de découverte globale des routes est lancée.

Plusieurs approches peuvent être appliquées dans la découverte des routes. La majorité des algorithmes utilisés sont basés sur le mécanisme d'apprentissage en arrière. Le nœud source S, qui est à la recherche d'un chemin vers la destination D, diffuse une requête de route RREQ dans le réseau. Lors de la réception de la requête, les nœuds intermédiaires essaient de faire apprendre le chemin au nœud source S, et de sauvegarder la route dans la table envoyée. Une fois que la destination est trouvée, elle peut envoyer une réponse de route RREP en utilisant le chemin tracé par la requête RREQ; un chemin est alors établi entre le nœud source S et le nœud destination D. Le travail peut être réduit, dans le cas où un nœud intermédiaire possède déjà un chemin vers la destination D. Une fois que le chemin est établi, il doit être sauvegardé et mis à jour au niveau de la source, tant qu'il est en cours d'utilisation. Une autre technique pour tracer les chemins demandés est appelée « routage source » comme dans le protocole DSR [55].

Dans la section suivante, nous présentons le processus d'inondation implémenté par le protocole de routage réactif AODV [41].

2.5.1 Inondation dans AODV

Le protocole AODV [41] est réactif, il représente une amélioration de l'algorithme DSDV [57]. Ce protocole réduit le nombre de diffusions de messages et cela en créant les routes seulement à la demande, contrairement à DSDV qui maintient la totalité des routes grâce à des tables de routage.

Avec AODV, un nœud envoie une requête de route dans le cas où il aurait besoin de connaître une route vers une certaine destination et qu'une telle route n'est pas disponible. Cela peut arriver dans deux cas: (a) La destination n'est pas connue au préalable; ou (b) Le chemin existant vers la destination a expiré et/ou est devenu défaillant /invalidé.

Le nœud source utilise une requête de route RREQ dans le but de créer un chemin vers une certaine destination; il reçoit une réponse RREP envoyée par la destination ou, en cas d'erreur de chemin, il reçoit une réponse ERROR envoyée par un nœud, qui a reçu RREQ, mais qui ne trouve pas de voisins. Figure 2.6.1 montre le nœud source (1) inonder le réseau avec les paquets de contrôle RREQ pour trouver la destination (8).

Trois types de requêtes sont utilisés, RREQ pour la découverte de chemin, RREP pour acheminer la réponse et ERROR pour les erreurs de chemin. Figure 2.6.2 montre la destination qui envoie la réponse dans le paquet RREP mais en suivant un seul chemin.

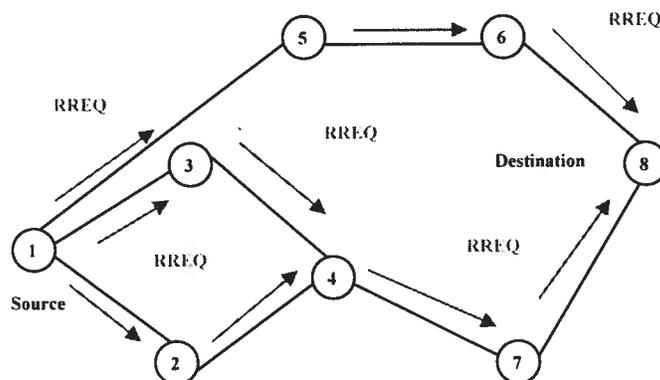


Figure 2.6.1. Recherche de chemin dans AODV

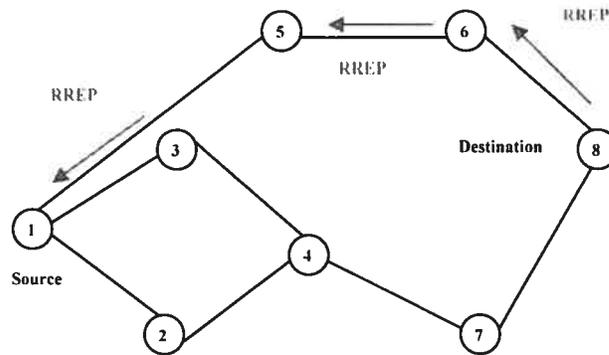


Figure 2.6.2. Réponse dans AODV

AODV utilise les principes des numéros de séquence afin de maintenir la consistance des informations de routage. À cause de la mobilité des nœuds dans les réseaux ad hoc mobiles, les routes changent fréquemment; ce qui fait que les routes maintenues par certains nœuds deviennent invalides. Les numéros de séquence permettent d'utiliser les routes les plus récentes. Ce protocole est basé sur l'utilisation des deux mécanismes "Découverte de route" et "Maintenance de route". Une entrée de la table de routage contient essentiellement : l'adresse de la destination, le nœud suivant, la distance en nombre de nœuds, le numéro de séquence destination, et le temps d'expiration de l'entrée de la table.

Le champ numéro de séquence destination du paquet RREQ (Figure 2.7.1) contient la dernière valeur connue du numéro de séquence associé au nœud destination. Cette valeur est recopiée de la table de routage. Si le numéro de séquence n'est pas connu, la valeur nulle sera prise par défaut. Le numéro de séquence source du paquet RREQ contient la valeur du numéro de séquence du nœud source.

Afin de maintenir des routes consistantes, une transmission périodique du message "HELLO" est effectuée. Si trois messages "HELLO" ne sont pas reçus consécutivement à partir d'un nœud voisin, le lien en question est considéré défaillant. Le protocole AODV ne présente pas de boucle de routage; en outre il évite le problème "counting to infinity" de Bellman-Ford, ce qui offre une convergence rapide quand la topologie du réseau ad hoc change.

Quand un lien devient défaillant, le nœud précédant diffuse un paquet UNSOLICITED RREP, avec une valeur de numéro de séquence égale à l'ancienne valeur du paquet RREP incrémentée de 1, et une valeur infinie de la distance. Le paquet UNSOLICITED RREP est diffusé aux voisins actifs, jusqu'à ce qu'il arrive à la source. Une fois le paquet reçu, la source peut initier le processus de la découverte des routes.

AODV maintient les adresses des voisins à travers lesquels les paquets destinés à un certain nœud arrivent. Un voisin est considéré actif s'il délivre au moins un paquet de données sans dépasser une certaine période. Une entrée de la table du routage est active, si elle est utilisée par un voisin actif. Le chemin reliant la source et la destination en passant par les entrées actives des tables de routage est dit un chemin actif. Dans le cas de défaillances de liens, toutes les entrées des tables de routage participants dans le chemin actif et qui sont concernées par la défaillance sont supprimées. Cela est accompli par la diffusion d'un message d'erreur entre les nœuds actifs.

En résumé, nous pouvons conclure que le protocole AODV utilise l'inondation classique pour découvrir des chemins à la demande.

2.6 Les Protocoles de routage géographiques

Dans cette Section, nous présentons une brève description de quelques protocoles de routage géographiques comme le GPSR [53], LAR [13], GeoCast [52] et DREAM [27]. La caractéristique principale de ces protocoles est qu'ils limitent l'inondation à une zone bien déterminée entre une source et une destination. Ainsi, les nœuds qui sont à l'extérieure de cette zone ne participent pas à l'inondation; ceci permet une grande réduction de surcharge et de consommation d'énergie. Cependant, pour utiliser ces protocoles, les nœuds doivent connaître leurs positions géographiques. Les protocoles géographiques sont classés en deux groupes : routage géographique hiérarchique et le routage géographique simple.

Pour connaître la position courante d'un nœud, nous avons besoin d'un service de localisation comme GPS. La position du nœud destination D est dans l'entête du paquet envoyé. Le nœud d'expédition S met à jour cette information chaque fois qu'il a la position précise du nœud destination D avant l'expédition. La position des nœuds

voisins est connue en faisant des inondations d'un saut, ce qui génère des surcharges et une perte d'énergie dans le réseau. Nous présentons dans la section suivante un des protocoles géographiques les plus connus, le GPSR.

2.6.1 Le protocole GPSR

Le GPSR est un protocole géographique qui utilise les informations de localisation [14] des nœuds qu'il envoie dans les paquets. Chaque nœud dans GPSR [53] maintient une table des nœuds voisins incluant leurs adresses et positions. Le paquet source contient la position géographique de la destination cherchée D, il ne change pas ce champ pendant que ce paquet parcourt le réseau

GPSR fonctionne en deux modes différents : Greedy [40] et périmètre.

Mode greedy : Le paquet choisit un nœud le plus proche de la destination (Figure 2.7); chaque nœud connaît la position de tous ses voisins en consultant sa table de routage. Cette stratégie est connue sous l'appellation Most Forward Within R (MFR), R représente le rayon de couverture du nœud. C'est une bonne stratégie dans le cas où l'expéditeur S du paquet ne peut pas adapter la force du signal de transmission à la distance entre la source et la destination D. Une autre stratégie, appelée Nearest Forward Progress (NFP), est utilisée dans le cas où la force du signal peut être adaptée; le paquet est envoyé au nœud le plus proche de l'expéditeur S et le plus proche de la destination D. Le niveau de réussite est plus élevé pour NFP que pour MFR.

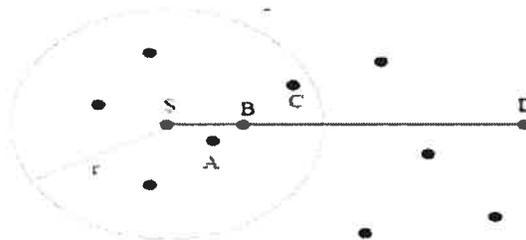


Figure 2.7. GPSR en Mode Greedy [40]

Mode périmètre : Quand le paquet envoyé en mode greedy arrive à une situation de blocage, le cas du maximum local, il bascule dans le mode périmètre. Un nœud est un

maximum local (Figure 2.8) s'il ne possède pas de voisin immédiat plus proche de la destination D.

Dans ce mode, le paquet est envoyé en utilisant un graphe planaire. Il envoie le paquet par l'une des faces du graphe planaire la plus proche de la droite SD dans le sens des aiguilles d'une montre. Un graphe planaire a deux faces : interne et externe. Quand le paquet arrive sur une face coupée par la droite SD, il bascule sur une autre face (voir Figure.2.9). Toutes les faces traversées sont coupées par le segment SD entre le nœud source S et la destination D.

Quand le paquet rentre dans le mode périmètre, GPSR enregistre dans le paquet la position L_p où le mode Greedy a échoué. A chaque fois que GPSR envoie le paquet sur une nouvelle face, il enregistre le point L_f en commun avec l'ancienne et la nouvelle face. Enfin, GPSR enregistre l'arête e_0 , la première arête que le paquet coupe dans la nouvelle face dans le paquet.

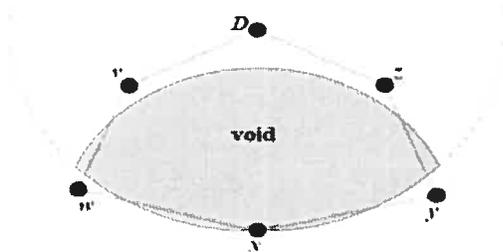


Figure 2.8. Maximum local dans GPSR

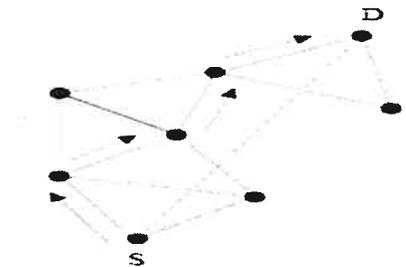


Figure 2.9. GPSR en mode périmètre

Après la réception du paquet en mode périmètre, le protocole GPSR compare la position L_p dans le paquet avec la position du nœud d'expédition; il retourne au mode greedy si la distance du nœud d'expédition à la destination D est plus petite que L_p à D. Pendant le mode périmètre, si le paquet trouve un nœud plus proche de la destination D, alors il revient en mode Greedy.

Quand le paquet est en mode périmètre, il est envoyé dans la première arête dans le sens des aiguilles d'une montre, ce qui détermine la première face par laquelle est envoyé le paquet. Après, GPSR expédie le paquet par d'autres faces en utilisant la règle Right Hand Rule (règle de la main droite). GPSR note la répétition d'expédition sur l'arête e_0 sauvegardée dans le paquet et supprime le paquet si la destination n'est pas

joignable. Les paquets qui ne trouvent pas leurs destinations peuvent se concentrer sur les faces des graphes du réseau.

2.7 Les réseaux de capteurs

Les récentes avancées dans les domaines des technologies sans-fil et électroniques ont permis le développement à faible coût de minuscules capteurs [45,47] consommant peu d'énergie. Ces capteurs ont 3 fonctions : (1) Capter des données (de type son, vibration, lumière, température,...); (2) traiter les données captées; et (3) transmettre les données à travers un réseau de capteurs.

Un réseau de capteurs (WSN) [48] est composé d'un nombre très important de nœuds qui sont posés dans des endroits précis ou dispersés aléatoirement (souvent déployés par voie aérienne). Ce dispersement aléatoire des capteurs nécessite des algorithmes d'auto organisation pour que le réseau puisse opérer.

Les réseaux de capteurs sans fil sont considérés comme un type spécial de réseaux ad hoc et utilisent tous les deux les communications multi sauts. Les capteurs collectent des données acheminées à un nœud central appelé le puit (figure 2.10) alors que dans les réseaux ad hoc, les nœuds sont autonomes et traitent l'information localement avec la possibilité de communication entre chaque paire de nœuds. Une autre différence importante entre les WSN et les MANET (Mobile Ad hoc Network) est que les capteurs ne sont pas généralement mobiles ce qui implique des stratégies de routage différentes. Les capteurs disposent d'une petite mémoire et d'une petite unité de calcul (CPU) ce qui représente un handicap supplémentaire pour le stockage des données et leur traitement. Dans les dispositifs ad hoc sans fil, les batteries peuvent être remplacées dans plusieurs cas; ainsi l'énergie n'est pas vraiment une contrainte cruciale alors qu'elle est fondamentale dans les réseaux de capteurs. L'impossibilité d'identifier les capteurs par un système d'adressage global, dû à leur forte densité, rend difficile la demande de données par le puit à un ensemble spécifique de sources.

Un réseau de capteurs ne peut pas survivre en cas de perte importante de ses nœuds (p.e. des nœuds qui ont épuisé leurs batteries); ceci peut donner lieu à un réseau fragmenté ou il n'y a aucune garantie de trouver un chemin entre deux nœuds du réseau. Les réseaux de capteurs peuvent être programmés pour supporter un large éventail

d'applications, telles que le contrôle d'intrusions, le calcul de températures pour la prévention des incendies de forêt, le calcul de changements climatiques, la surveillance des déplacements d'animaux [46, 63], surveillance de malades dans les hôpitaux ou à domicile,...

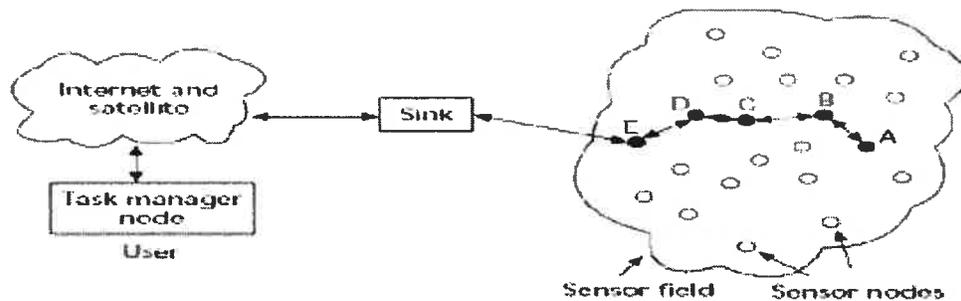


Figure 2.10. Réseau de capteurs [45]

2.7.1 Consommation d'énergie

Un capteur est limité en énergie; généralement une batterie est utilisée. Dans la plupart des cas, le remplacement de la batterie est impossible (exemple de capteurs déployés dans un champs de bataille par voie aérienne sans possibilité de les récupérer) et sa durée de vie dépend de la durée de vie de la batterie. Dans un réseau de capteurs, chaque nœud collecte des données et transmet des valeurs. Le dysfonctionnement de quelques nœuds nécessite un changement de la topologie du réseau et un re-routage des paquets. Toutes ces opérations consomment de l'énergie; c'est pour cette raison que les recherches actuelles se concentrent sur les moyens de réduire cette consommation.

2.7.2 Architecture

La figure 2.11, montre l'architecture d'un nœud (du réseau de capteurs); elle combine puissance et routage, intègre un protocole réseau, utilise les technologies sans-fil, et favorise la coopération des nœuds de capteur. L'architecture comprend 5 couches qui ont les mêmes fonctions que celles du modèle OSI [44] ainsi que 3 couches pour la gestion de la puissance, la gestion de la mobilité ainsi que la gestion des tâches. Chaque couche du modèle communique avec une couche adjacente. Chaque couche utilise ainsi les services des couches inférieures et en fournit à celles de niveaux supérieurs.

Couche Application
Couche Transport
Couche Réseau
Couche Données
Couche Physique

Figure 2.11. Modèle en couche de l'architecture de réseau de capteurs [45]

2.7.3 Le capteur

Un nœud capteur est un dispositif qui transforme l'état d'une grandeur physique observée en une grandeur utilisable (une tension électrique, une hauteur de mercure, une intensité, la déviation d'une aiguille...). Il se distingue de l'instrument de mesure par le fait qu'il ne s'agit que d'une simple interface entre un processus physique et une information manipulable. Les capteurs sont les éléments de base des systèmes d'acquisition de données.

Il existe une multitude de modèles de capteurs en relation avec l'application à laquelle il est destiné. Parmi les modèles les plus utilisés, on trouve les MICA [49], les MICA2 [50] et les MICA2DOT [51]. On caractérise un capteur selon plusieurs critères dont les plus courants sont : la grandeur physique observée, étendue de mesure, sensibilité, précision, linéarité, bande passante, plage de température de fonctionnement, dérivée thermique, résolution, hystérésis.

Les dernières années ont vu paraître le concept de capteurs intelligents [63]. En plus de leur faculté de mesurer une grandeur physique, ils possèdent d'autres fonctionnalités comme des fonctions configurables de traitement du signal, d'autotest et d'autocontrôle, d'étalonnage automatique, et de sortie sur des bus de terrain

Les capteurs peuvent aussi faire l'objet d'une classification analogique et numérique. Le signal d'un capteur numérique est sous la forme d'un train d'impulsions, avec un nombre précis d'impulsions ou avec une fréquence précise, ou d'un code numérique binaire. Les capteurs numériques les plus fréquents sont les capteurs incrémentaux, les codeurs absolus, les lecteurs de codes-à-barres, et les lecteurs de pistes magnétiques.

2.7.4 Énergie d'un nœud capteur

La consommation d'énergie se produit en trois étapes : l'écoute, la communication, et le traitement des données. La modification des algorithmes de routages peut avoir des conséquences directes sur la consommation d'énergie. Habituellement, la communication des données consomme plus d'énergie que l'écoute ou le traitement de ces données.

Un modèle simple de consommation d'énergie est proposé [33]. Ce modèle peut être utilisé pour simuler les nœuds capteurs quand ils transmettent et reçoivent les données. Pour émettre un paquet de K bits à une distance d , l'énergie est donnée par:

$$E_{Tx}(k, d) = kE_{elec} + \epsilon_{amp} kd^2$$

$\epsilon_{amp} (J/b/m^2)$: représente l'énergie dépensée par amplificateur de puissance au niveau de l'émetteur en joule par bit sur la surface couverte ($J/b/m^2$).

$E_{elec} (J/b)$: représente l'énergie dissipée pour transmettre ou recevoir 1 bit de donnée en joule par bit (J/b).

| Pour recevoir un paquet de K bits, l'énergie consommée est donnée par:

$$E_{Rx}(k) = kE_{elec}$$

2.8 Routage dans les réseaux de capteurs

Nous avons cité plus haut certaines différences entre les réseaux ad hoc sans fil et les réseaux de capteurs, elles font en sorte que les nombreux protocoles de routage pour les réseaux ad hoc sans fil proposés ces dernières années ne sont pas appropriés pour les réseaux de capteurs.

En particulier, l'impossibilité d'identifier les capteurs par un système d'adressage global, dû à leur forte densité, rend difficile la demande de données par le puit à un ensemble spécifique de sources. Si l'inondation est utilisée, chaque nœud du réseau transmet l'information à tous ses voisins. Cela entraîne une redondance de l'information. En effet, les contraintes majeures de l'inondation proviennent des problèmes d'implosion et de recouvrement. Le problème d'implosion est causé par la réception de

copies du même message. Le problème de recouvrement est aussi dû au fait que l'information perçue à un sommet dans le réseau peut être collectée par plus d'un capteur à la fois. Nous présentons dans les sections suivantes quelques protocoles de routage qui ont été développés pour les réseaux de capteurs.

2.8.1 Address-centric Protocol (AC)

Le puit diffuse dans tout le réseau (inondation) la requête de demande de données. Chaque source envoie indépendamment des données le long du chemin le plus court au puit basé sur l'itinéraire que les requêtes ont pris (cheminement bout à bout). Dans l'approche adresse centrale, chaque source envoie son information séparément au puit. Figure 2.12.a montre la source 1 acheminant les données marquées "1" par le noeud A, et la source 2 acheminant les données marquées "2" par les nœuds C et B. les protocoles proactifs, réactifs et hybrides sont tous à adresse centrale.

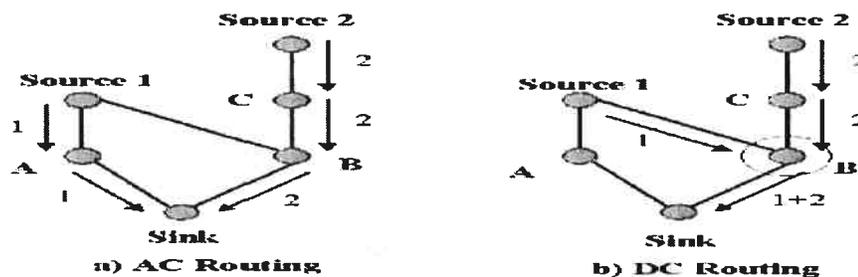


Figure 2.12. Routage dans les réseaux de capteurs

2.8.2 Data-centric Protocol (DC)

A la différence de AC, les sources envoient des données au puit mais les nœuds intermédiaires acheminent les données suivant leur contenu et exécutent une certaine forme d'agrégation sur les données envoyées par les sources multiples. Dans l'approche données centrale, les données des deux sources sont agrégées au niveau du nœud B, et les données combinées (marquées "1+2") sont envoyées de B au puit (figure 2.12.b). Ce protocole est similaire à un autre utilisé dans les réseaux de capteurs, appelé la diffusion directe [43]. D'après les travaux qui ont été faits [44], l'approche donnée centrale donne de meilleurs résultats que l'approche adresse centrale. Dans ce qui suit, nous présentons le protocole SPIN [52] qui est un mécanisme de routage à données centrales.

SPIN (*Sensor Protocols for Information via Negotiation*) vise à diffuser l'information parmi tous les capteurs en utilisant des descripteurs d'information pour la négociation avant la transmission des données. Ces descripteurs d'information s'appellent les méta-données et sont employées pour éliminer la transmission des données redondantes dans le réseau. Dans le SPIN, chaque nœud capteur a également son gestionnaire de ressource propre qui maintient la quantité d'énergie que le nœud possède. Avant la transmission ou le traitement des données, les nœuds consultent leur gestionnaire de ressource pour vérifier s'ils ont assez d'énergie ou pas. Ceci permet aux nœuds d'éviter des activités quand leurs ressources sont basses augmentant la durée de vie du nœud dans le réseau.

2.8.3 Protocoles basés localisation

La position de chaque nœud dans un réseau de capteurs est une information requise par de nombreux algorithmes de routage. Elle peut être obtenue par un signal GPS [20] ou protocoles de localisation [74, 75, 76]. Les réseaux de capteurs ne sont pas basés sur un mode adressage à base d'adresses IP comme dans les réseaux de communication traditionnels; la position des nœuds peut donc être nécessaire pour un routage efficace des données collectées. Il serait aisé de réduire la transmission d'informations si l'on connaissait exactement la position de chaque capteur, simplement en ne diffusant la requête qu'aux capteurs capables de nous renseigner et non à l'ensemble d'entre eux. Dans ce qui suit nous présentons un protocole de routage, appelé Geographic Adaptive Fidelity (GAF), dans les réseaux de capteurs basé sur la localisation des capteurs.

Utilisé généralement dans les réseaux ad hoc, GAF est implémenté aussi dans les réseaux de capteurs du fait qu'il minimise la consommation d'énergie. Le principe de ce protocole est le suivant : la capacité d'énergie de chaque nœud du réseau est préservée en arrêtant (faire dormir) chacun des capteurs n'intervenant pas dans la route utilisée à un certain moment pour transmettre les données vers le puit [52].

Chacun des capteurs s'affecte automatiquement à un point géographique du réseau (on parle alors de grille virtuelle) et lorsque plusieurs d'entre eux sont affectés à la même zone géographique (cela signifie qu'ils sont équivalents en terme de coût de

routage), un ou plusieurs de ces capteurs sont arrêtés. Il y a donc une forte économie d'énergie, ce qui augmente sensiblement la durée de vie du réseau. Les nœuds dans l'état «dormant » se relaient à tour de rôle afin d'équilibrer la charge. Trois états peuvent être définis: découverte (lorsqu'un capteur est à la recherche de sa position), actif et dormant. Chaque capteur détermine approximativement sa durée de sommeil afin de pouvoir permuter efficacement avec un autre capteur dans sa zone.

Il est à noter que GAF peut également se classer dans la catégorie des protocoles hiérarchiques ou de groupage (voir chapitre 3) dans la mesure où le chef de groupe est localisé géographiquement.

2.9 Conclusion

Dans ce chapitre nous avons présenté le concept de d'inondation classique dans les réseaux ad hoc mobile et réseaux de capteurs. Ces réseaux connaissent aujourd'hui plusieurs applications telles que les applications militaires, les applications de secours et de façon générale, toutes les applications caractérisées par une absence d'infrastructure préexistante. Nous avons vu que les protocoles de routage utilisent la technique de l'inondation classique afin d'avoir les informations indispensables à la construction des chemins, pour acheminer les données ou pour connaître la topologie du réseau. Cette inondation cause des surcharges et une grande perte d'énergie. Nous avons abordé brièvement quelques concepts relatifs aux nœuds capteurs et l'énergie consommée par ces nœuds.

Nous avons aussi présenté quelques notions et modèles de protocoles nécessaires à la compréhension du routage dans les environnements ad hoc mobiles et de capteurs. Dans le chapitre suivant, nous allons donner une présentation des différentes solutions qui existent et qui essaient de résoudre le problème de l'inondation classique dans les réseaux ad hoc mobiles et réseau de capteurs. Une de ces solutions qui minimise l'inondation de tout le réseau est la technique de groupage (clustering).

Chapitre 3 Algorithmes de groupage (clustering)

3.1 Introduction

Les algorithmes de groupage sont basés sur une partition du réseau en sous-structures logiques appelées groupes (clusters) qui confinent et marquent les interactions locales des nœuds entre eux au reste du réseau. Pendant le processus de création de ces sous-structures logiques, un seul nœud, appelé chef de groupe (cluster head), peut être sélectionné dans chacun des groupes pour exercer le contrôle et la coordination dans cette structure. La communication entre les différents groupes est réalisée par des nœuds appelés nœuds de liaison (gateway).

Dans les algorithmes de groupage, seuls les chefs de groupes et les nœuds de liaison diffusent les paquets pour réaliser des communications entre des nœuds appartenant à différents groupes. Ainsi, un algorithme de groupage réduit considérablement les effets négatifs de l'inondation/diffusion ; un sous ensemble de nœuds de réseaux participe à la diffusion.

Les algorithmes de groupage diffèrent principalement dans la façon de construire les groupes et de choisir les chefs de groupes et nœuds de liaison. Dans cette section, nous détaillons les approches représentatives de groupage.

3.2 Algorithmes de groupage

Dans les réseaux ad hoc mobiles, les informations du voisinage (vitesse, état, nombre de nœuds, distance) ne sont pas faciles à collecter; en réalité, la raison principale vient du manque de fiabilité et de la limite des capacités de liens et de la mobilité des nœuds. Les informations de voisinage sont des informations capitales pour le groupage; en effet, en se basant sur ces informations qu'un algorithme de groupage est capable de construire les groupes et choisir les chefs de groupes et nœuds de liaison. Pour assurer la collecte des informations des nœuds voisins, les anciennes solutions de groupage diffusent continuellement les informations des listes de voisins. Pour avoir une convergence pendant la phase initiale de groupage et lors de la découverte des voisins, il est

nécessaire de ne pas avoir de mobilité afin d'éviter des ruptures de chemin plus fréquentes, ce qui est une contrainte majeure dans les réseaux ad hoc mobiles.

Il existe deux types d'algorithmes de groupage : (1) groupage actif; et (2) groupage passif. Dans les sections suivantes, nous définissons les deux types de groupage et présentons des algorithmes de groupage pour chaque type de groupage. Avant, nous définissons trois types de nœuds qui interviennent dans tous les algorithmes de groupage.

Chef de groupe (Clusterhead): c'est un nœud qui s'occupe de la coordination des transmissions à l'intérieur de son groupe et avec d'autres groupes. Il est différent de la station de base (SB) utilisée dans la téléphonie cellulaire parce qu'il n'a pas de matériel spécial et est sélectionné dynamiquement parmi un ensemble de nœuds. Plusieurs heuristiques sont utilisées pour élire un chef de groupe dans chaque groupe suivant un poids calculé qui le distinguera des autres nœuds du réseau. Ce nœud diffuse tous les paquets qu'il reçoit.

Nœud de liaison (Gateway): c'est un nœud qui permet la communication entre les chefs de groupes. Plusieurs heuristiques existent pour élire et minimiser le nombre de nœuds de liaisons.

Nœud ordinaire : c'est un nœud qui n'expédie jamais les paquets qu'il reçoit. Il peut être source de données et ainsi diffuser les paquets à envoyer; il est 'tout le temps' en écoute pour lire/capturer les paquets qui lui sont destinés.

3.2.1 Algorithmes de groupage actif

Un algorithme de groupage actif a besoin des informations de voisinage pour construire les groupes et choisir les chefs de groupes et nœuds de liaison; ainsi, un protocole de signalisation est requis pour réaliser cette activité. Ceci peut donner lieu à une surcharge (overhead) considérable de réseau, surtout, en cas de grande mobilité.

3.2.1.1 Lowest-ID (LID)

C'est un protocole de formation de groupe basé sur l'identificateur (Id) du nœud, il choisit au début le nœud qui a le plus petit identificateur de paquet (LID) [84]. Il partitionne le réseau en groupes qui ne se recouvrent pas et qui ne dépassent pas deux sauts. Cet algorithme nécessite que la topologie ne change pas pendant l'exécution de l'algorithme, il suppose aussi :

- Qu'un nœud a un Id unique et que chaque nœud connaît l'Id de son voisin.
- Un message envoyé par un nœud est reçu correctement en un temps fini par ses voisins à un saut.
- Chaque nœud peut déterminer son groupe et seulement un seul.
- Dans un groupe, deux nœuds sont à deux sauts l'un de l'autre.

Avec l'heuristique LID, quand un nœud chef de groupe écoute un nœud avec un identificateur plus petit, il abdique son rôle de chef de groupe à ce nœud là. Figure 3.1 montre avec les cercles en pointillés le résultat de l'algorithme quand le nœud 1 est en dehors du rayon de couverture du nœud 3. Si le nœud 1 s'approche du nœud 3, ce dernier abandonne son rôle de chef de groupe et déclenche une réaction qui change tous les chefs de groupe. Ce phénomène est plus accentué en présence d'une grande mobilité de nœud, on aura des ruptures de chemins plus souvent, ce qui impliquera de reconstruire les groupes plus fréquemment. Dans le cas du groupage actif, c'est une grande contrainte vu que la phase d'initialisation est coûteuse en paquets de contrôle RREQ et en énergie.

Dans le cas du groupage avec LID, le nœud ayant le plus petit Id devient un chef de groupe mais il n'existe aucune garantie que ce chef de groupe ait le plus petit identificateur, il peut être aussi un nœud de liaison.

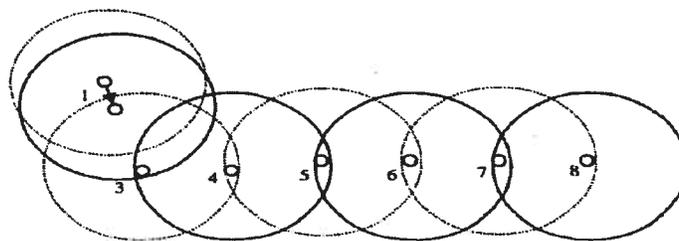


Figure 3.1. Nœud 1 s'approche du nœud 3 et cause un changement de groupage

3.2.1.2 Highest Degree (HD)

Dans cet algorithme le nœud qui peut se connecter à un nombre de nœuds voisins le plus élevé devient chef de groupe. Pour former un groupe avec cet algorithme, chaque nœud calcule son degré de connectivité, celui qui possède le degré le plus grand se déclare alors chef de groupe. Les expériences [28] ont montré que HD a un taux de stabilité plus bas que LID pour le changement de chefs de groupe et cela quelque soit le degré de mobilité des nœuds.

3.2.1.3 Algorithme k-LID and k-CONID

Cet algorithme fait appel à deux techniques séparées, LID et le degré maximal de connectivité au voisinage d'un nœud pour l'élection des chefs de groupe.

L'algorithme distribué LID est généralisé pour la formation de groupe sur k sauts (K-LID). Un nœud quelconque doit connaître ses nœuds voisins à k sauts; s'il dispose de l'identificateur le plus petit, il prend la décision de former le groupe en étant lui même le chef de ce groupe. Le degré d'un nœud est défini comme étant la connectivité pour 1 groupe.

Dans l'algorithme k-CONID [64], est associé à chaque nœud une paire (d, ID), connectivité et identificateur du nœud respectivement, appelée priorité du chef de groupe. Cette priorité est élevée pour un nœud s'il a la plus grande connectivité ou le plus petit identificateur (en cas d'égalité de connectivité). Les nœuds qui appartiennent à plusieurs groupes s'appellent les nœuds frontières.

Cette technique de groupage [29] définit quatre algorithmes de groupage k-LID et k-CONID avec k=1 et k=2 combinés. Pour la sélection de chef de groupe, deux critères sont utilisés: (1) la connectivité et (2) l'identificateur du nœud le plus petit. Le but de cette technique de groupage est de minimiser le nombre de groupes.

3.2.1.4 Weighted Clustering Algorithm (WCA)

L'algorithme WCA [30] combine les avantages des autres algorithmes de groupage; il prend comme paramètres d'entrées le degré d'un nœud, sa puissance de transmission, sa mobilité et la capacité de sa batterie. L'algorithme tente de calculer un nombre minimal de chefs de groupe pour maximiser l'attribution des ressources. Avec les

paramètres cités, on calcule un poids W pour le nœud v qui servira à l'élection des chefs de groupe: $W(v) = w1 * Degree(v) + w2 * Distance(v) + w3 * Mobility(v) + w4 * Power(v)$.

Plusieurs heuristiques pour l'élection de chef de groupe, déjà utilisées, sont des cas particuliers de WCA avec des choix différents des paramètres $w1$, $w2$, $w3$ et $w4$. Les résultats des expérimentations montrent que WCA est plus performant dans la formation de chefs de groupe que les algorithmes HD et LID [30].

3.2.1.5 Maximal Weighted Independent Set (MWIS)

L'algorithme MWIS [62] ramène le problème du groupage à trouver un poids maximal pour un nœud parmi un ensemble de nœuds indépendants. À chaque nœud est assigné un poids basé sur sa convenance à être un chef de groupe; ce poids est calculé en tenant compte de la vitesse du nœud. Des nœuds plus lents auront un plus grand poids que ceux qui sont plus rapides.

MWIS a beaucoup d'avantages : il exige seulement la connaissance de la topologie locale de chaque nœud, il est rapide et facile à mettre en application, et il s'avère que sa complexité de temps est liée à un paramètre dépendant de la topologie du réseau.

L'inconvénient de cet algorithme est qu'il se base sur l'hypothèse qu'un message envoyé par un nœud est reçu correctement en un temps fini par ses voisins immédiats. Un nœud attendra toutes les réponses de ses voisins pour prendre une décision d'être chef de groupe ou non.

3.2.1.6 Low Energy Adaptive Clustering Hierarchy (LEACH)

LEACH [73] est conçu spécifiquement pour des réseaux de capteurs où un utilisateur, par exemple, veut surveiller à distance l'environnement. Dans une telle situation, les données des différents capteurs doivent être envoyées à une station de base centrale, souvent située loin du réseau de capteurs, par laquelle l'utilisateur peut accéder aux données.

LEACH inclut la formation distribuée de groupe, le traitement local pour réduire la communication globale, et la rotation des chefs de groupe. Ensemble, ces dispositifs permettent à LEACH en utilisant l'inondation des nœuds de réaliser les propriétés désirées et permet un rendement optimum qui prolonge la durée de vie du réseau. L'exécution de LEACH est divisée en tours pour permettre la rotation des chefs de

groupe, chaque tour comporte deux phase : (1) phase installation; (2) phase équilibre. Pendant la phase installation, les chefs de groupe sont déterminés et les groupes organisés. A la phase équilibre, les données sont transférées à la station de base.

Les chefs de groupe diffusent leurs états aux capteurs du réseau, collectent les données des capteurs, agrègent ces données et les transferts à la station de base. Chaque capteur détermine le groupe auquel il veut appartenir en choisissant le chef de groupe qui a besoin d'un niveau d'énergie minimal pour la communication.

3.2.1.7 Hybrid Energy-Efficient Distributed clustering (HEED)

HEED [19] se base sur LEACH mais incorpore en plus les limites du rayon de couverture des capteurs et l'information de coût. Dans HEED, la probabilité pour qu'un capteur devienne un chef de groupe dépend de son énergie résiduelle. Les capteurs qui ne sont couverts par un chef de groupe doublent leur probabilité de devenir eux même des chefs de groupe. Ce procédé est répété jusqu'à ce que tous les capteurs soient couverts par au moins un chef de groupe. LEACH et HEED exigent un regroupage après une période de temps appelée « Round » ce qui cause une consommation d'énergie supplémentaire.

HEED a quatre objectifs principaux: (i) prolonger la durée vie du réseau en distribuant la consommation d'énergie, (ii) finaliser le processus de groupage en un nombre constant d'étapes, (iii) réduire les perte dues aux paquets de contrôle, et (iv) produire une bonne distribution des chefs de groupe et des groupes compacts. HEED ne fait aucune supposition au sujet de la distribution, capacités des nœuds ou de la densité des nœuds.

3.2.2 Groupage passif

En opposition aux approches de groupage actif, le groupage passif est une approche de formation de groupes qui n'utilise pas de protocole de signalisation et ne nécessite aucune surcharge due aux échanges des informations de voisinage. En plus, le groupage passif construit la structure de groupes avec les chefs de groupes et nœuds de liaison *sur demande*; ce qui permet de minimiser les coûts de maintenance d'une structure de groupe.

C'est clair, que le groupage passif est beaucoup plus performant que le groupage actif. En effet, le groupage passif a été défini pour surmonter/éliminer les limitations du groupage actif en termes de surcharge et de consommation d'énergie (plus de trafic implique plus de consommation d'énergie). Dans la section suivante, nous présentons en détail le fonctionnement de groupage passif ; notre travail de mémoire consiste à améliorer/étendre le groupage passif.

3.2.2.1 Algorithme du groupage passif (PC)

Kwon et al. étaient les premiers à proposer un protocole de groupage passif (PC) [11,28]. C'est un protocole de formation de groupes qui n'utilise pas un protocole spécifique pour le contrôle de paquets. Il utilise une procédure de groupage simple, facile à implémenter et complètement distribuée.

PC utilise la trame MAC (couche 2 du modèle TCP/IP) pour coder l'état d'un nœud du réseau; plus particulièrement, PC introduit 2 bits pour coder 4 états: ordinary, cluster_head, gateway, clutster_head_ready. Chaque trame de données contient 2 bits supplémentaires (Figure 3.2) qui représentent l'état du nœud qui vient de diffuser la trame.



Figure 3.2. Format de la trame dans PC

Initialement, tous les nœuds sont à l'état initial ; un nœud qui rejoint le réseau, commence aussi avec l'état initial. Un nœud change son état seulement lorsqu'il reçoit un paquet des ses voisins.

Quand un nœud reçoit un paquet, il vérifie l'état de l'émetteur (en lisant les 2 bits d'état). Si l'émetteur n'est pas un chef de groupe (dans le paquet l'état≠clutster_head) et n'est pas un nœud de liaison (état≠gateway), il change son état à cluster_head_ready; sinon, il change son état à ordinary. Un nœud dans l'état clutster_head_ready, devient chef de groupe s'il réussit à transmettre un paquet avant qu'il ne reçoive un paquet d'un chef de groupe (un paquet avec état=clutster_head); cette heuristique est appelée 'First Declaration Wins' (FDW).

Un chef de groupe, qui ne reçoit pas de paquet d'un nœud de liaison (le chef de groupe ne reçoit pas de paquet avec l'état=gateway) pendant une période de donnée prédéfinie (cette situation est appelée problème d'isolation), change son état à ordinary.

Un nœud qui reçoit un paquet de chef de groupe (dans le paquet reçu l'état=cluster_head), le nœud change son état à ordinary et ajoute les informations du chef de groupe (adresse MAC, temps de réception) dans sa liste des chefs de groupe; en effet, chaque nœud, qui n'est pas chef de groupe, maintient une liste de chefs de groupes voisins (identificateur, temps de réception de paquet); si le nœud ne reçoit pas de paquet pendant une période de temps prédéfinie d'un nœud de la liste, la rentrée correspondante est supprimée (p.e., une situation où un chef de groupe épuise sa batterie). Cette période de temps doit être choisie minutieusement; il doit tenir en compte la mobilité des nœuds.

Un nœud qui écoute plus d'un chef de groupe à la fois devient un nœud de liaison. Ainsi la distance entre deux chefs de groupe est au moins 2 sauts (Figure 3.3). En général, dans les réseaux ad hoc mobiles sans fil et capteurs, le nombre de nœuds de liaison est significatif et peut dépasser le nombre des nœuds ordinaires; donc il faut réduire le nombre de ces nœuds de liaisons pour avoir une inondation efficace. Pour avoir un ensemble minimal de nœuds de liaison, chaque chef de groupe doit maintenir la liste des chefs de groupe de tous les nœuds de liaison dans son groupe; en se basant sur cette liste, il peut choisir un nombre réduit de nœuds de liaisons qui lui permettent de communiquer avec les chefs de groupe dans la liste. Ce nombre doit être choisi d'une façon très minutieuse : un grand nombre permettra une plus grande connectivité (plusieurs chemins entre chefs de groupe) mais augmentera le nombre de collisions (surcharge) dans le réseau. Ainsi, il faut chercher un équilibre entre la surcharge et la connectivité (plus de détails dans la section prochaine).

Avec PC, chaque fois qu'un paquet arrive sur un nœud quelconque, le nœud n'a besoin que de l'état qui est collé sur le paquet reçu pour faire les traitements relatifs à la sélection de nœuds de liaison, chef de groupe ou nœud ordinaire. PC utilise la couche CSL (Clustering Sub Layer), qui se situe au dessous de la couche MAC, coller/décoller les 2 bits d'état du paquet échangé. Avec ce mécanisme collage/décollage de l'état, PC peut être utilisé sans aucune modification du protocole MAC ou du protocole réseau

(IP). Seuls les chefs de groupe et les nœuds de liaison diffusent les paquets de données et de contrôle; les nœuds ordinaires reçoivent les paquets mais ne participent pas aux diffusions de ces paquets, ce qui limite les collisions et réduit le volume de paquets échangés. En plus, il n'y a aucun trafic supplémentaire (à l'exception de 2 bits par paquet de données) qui est généré pour échanger les informations de voisinage comme c'est le cas des approches de groupage actif). Par conséquent, il y a une diminution considérable de la consommation d'énergie par les nœuds ordinaires ce qui permet d'augmenter la durée de vie du réseau.

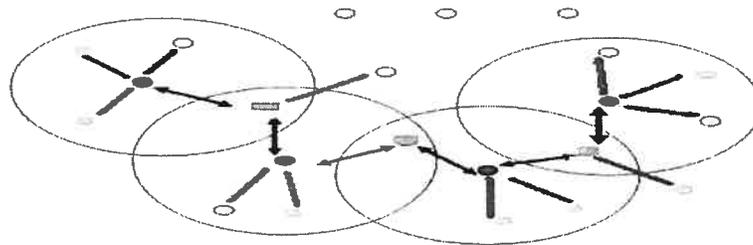


Figure 3.3. Chefs de groupe, nœuds de liaison, et nœuds ordinaires

Figure 3.3 montre les flèches en noir qui symbolisent un échange de paquets bidirectionnels qui se fait seulement entre les chefs de groupes et nœuds de liaisons. Les traits en rouge symbolisent un échange unidirectionnel de paquets vers les nœuds ordinaires puisqu'ils ne diffusent jamais les paquets reçus.

Le tableau 3.1 montre le pseudo code du protocole de groupage passif. Chaque nœud du réseau exécute ce code après initialisation.

```
Input
  Paquet; /* paquet envoyé par la source; il contient l'état du nœud émetteur*/
  NCH; /* nombre de chefs de groupe */
  Timeout; /* temps d'attente du nœud pour revenir à Initial */
  Node; /* nœud courant, il a un enregistrement qui contient un attribut : état courant
        du nœud */

Begin

  Node.etat = Initial;
  /* initialiser le noeud au début */

  While (true)
    If Node reçoit (paquet)
```

```

If ( Paquet.etat <> Clusterhead)
    Node.etat = ClusterHead_Ready;
Else
    If (Node.send= true) /* noeud envoie paquet */
        Node.etat = ClusterHead;
        Enjoyer(Paquet);
    Endif
Else
    If ( Paquet.etat = clusterhead)
        MAJ (NCH)
        /* metre à jour le nombre de chefs de groupe */
    Endif
    If (NCH>= 2)
        Node.etat = Noeud de liaison;
        Envoyer (Paquet);
    Else
        If ( Paquet.etat = ClusterHead) && (Node.etat=ClusterHead)
            Node.etat = ordinary_node;
            /* Bloquer la transmission des paquets */
        Endif
        Else
            If (paquet.etat = ClusteHead) &&( node.etat = Initial)
                Node.etat = initial;
            Endif
        Endif
    Endif
    If (Node.reçoit=false) && (Node.send=false) && (Temps > TimeOut)
        Node.etat =Initial;
    Endif
Endwhile
End

```

Tableau 3.1. Pseudo code de PC

3.2.2.2 Heuristiques de sélection de nœuds de liaison

Généralement dans les réseaux de capteurs très denses composés de milliers de nœuds, le nombre de nœuds de liaison est très élevé, ce qui pousse les chercheurs dans ce domaine à introduire des méthodes ou des techniques qui puissent réduire ce nombre

élevé de nœuds de liaisons sans altérer la connectivité du réseau. Certains se basent sur des heuristique mathématiques consistant à calculer le nombre de chefs de groupe et de nœuds de liaisons au voisinage d'un nœud et faire des comparaisons selon des tests spécifiques suivant les applications utilisées [29, 28, 30, 62].

Les auteurs de PC [10, 11, 28] ont proposé une heuristique simple qui n'affecte pas le fonctionnement de base PC; cette heuristique n'exige pas l'utilisation de protocole de signalisation pour échanger des informations de contrôle pour le choix des nœuds de liaison. Lorsqu'un nœud ordinaire (état=ordinary), reçoit un paquet d'un chef de groupe (le paquet reçu contient l'état=cluster_head) ou d'un nœud de liaison (le paquet reçu contient l'état=gateway), il devient un nœud de liaison si $\alpha * NC + \beta > NG$ où NC est le nombre de chefs de groupe, NG est le nombre de nœuds de liaison, α est le degré de redondance des nœuds de liaisons désiré, et β est un facteur compris entre 0 et 1 à choisir selon la densité du réseau. Rappelons qu'un nœud qui n'est pas chef de groupe, maintient une liste de chefs de groupes et nœuds de liaison voisins; chaque fois, il reçoit un paquet, il décolle l'état correspondant, et ajoute une entrée, s'il n'existe pas, pour le nœud émetteur (qui a envoyé le paquet) s'il est chef de groupe ou nœud de liaison. Ainsi NC et NG peuvent être facilement calculés en utilisant la liste.

En manipulant α et β , l'heuristique peut contrôler le nombre de nœuds de liaison dans le réseau. Un grand nombre de nœuds de liaison induit une grande surcharge du réseau mais une bonne connectivité/fiabilité alors qu'un petit nombre de nœuds de liaison induit une connectivité/fiabilité limité mais une réduction considérable de la surcharge du réseau. Les valeurs de α et β doivent être choisies en fonction de la qualité du canal, le niveau de bruit et la nature du trafic utilisé. Par exemple, dans les réseaux très denses, il y a beaucoup de collisions dues aux paquets transmis; par conséquent, de grandes valeurs pour α et β mènent à plus de nœuds de liaisons et ainsi à une meilleure performance du réseau par une distribution du trafic à travers plusieurs nœuds de liaisons [10].

Avec cette heuristique, PC réalise un équilibrage entre les chefs de groupe et les nœuds de liaisons; il sert à contrôler le nombre de nœuds de liaisons dans un espace donné sans altérer la nature de PC; plus précisément, il permet de limiter le nombre de

nœuds de liaisons qui diffusent les paquets de données entre les différents chefs de groupe.

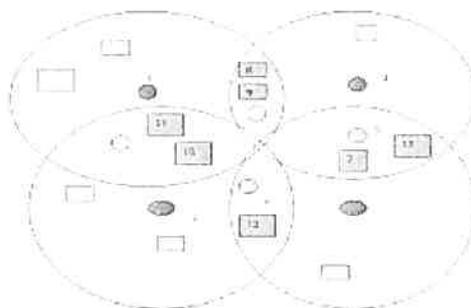


Figure 3.4. Nœuds de liaison sans heuristique

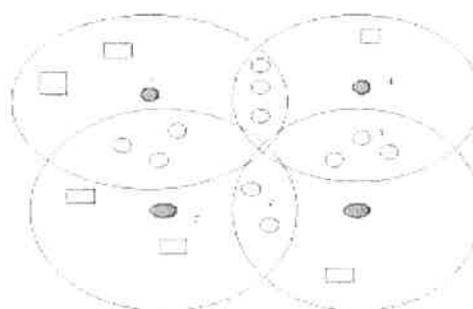


Figure 3.5. Nœuds de liaison avec heuristique

La figure 3.4 montre que les nœuds 1, 2, 3 et 4 sont des chefs de groupe et les nœuds de 7 à 13 sont des nœuds de liaison (communiquent avec deux chefs de groupes). Les nœuds non marqués sont des nœuds ordinaires qui écoutent et reçoivent les paquets de données mais ne diffusent jamais ces paquets. Nous sommes en présence d'une heuristique qui minimise le nombre de nœuds de liaisons entre chaque paire de chefs de groupe à 1 seulement. Sans cette heuristique, on aurait pu avoir les nœuds 7, 8, 9, 10, 11, 12 et 13 comme des nœuds de liaisons (Figure 3.5), et en conséquence il existerait des collisions et des interférences entre ces nœuds. Il faut noter qu'avoir un seul nœud de liaison entre deux chefs de groupes n'est pas un bon choix [16]; ça pourrait, par exemple, causer une déconnection entre ces deux groupes en cas de panne de ce nœud de liaison.

3.3 Conclusion

Les algorithmes de groupage consistent, généralement, à structurer les nœuds du réseau sous forme de groupes et à utiliser des algorithmes/heuristiques pour élire un nœud, appelé chef de groupe, dans chaque groupe pour représenter les nœuds de ce groupe. Seuls les chefs de groupes et nœuds de liaison (permettant la communication entre 2 chefs de groupes ou plus) diffusent le trafic limitant ainsi les l'inondation à ces nœuds à l'inverse de l'inondation classique où tous les nœuds du réseau diffusent le trafic.

Il y a deux types de groupage : groupage actif et groupage passive. Les algorithmes de groupage actif (p.e. AC, PC, WCA) utilisent généralement un protocole de signalisation pour échanger des informations de voisinage qui sont ensuite utilisées pour former les groupes et élire les chefs de groupe; ce processus introduit une surcharge de réseau, surtout dans le cas de grande mobilité de nœuds, et peut être relativement long. À l'inverse, le groupage passive (PC [10, 11]) ne nécessite aucun protocole de signalisation; en plus, il fonctionne sur demande. Il utilise une règle (First Declaration Wins) d'élection de chefs de groupe très souple, distribuée et rapide qui se base sur des informations d'états du voisinage immédiat du nœud. Les informations de voisinage sont échangées en utilisant les paquets de données; PC propose l'ajout de 2 bits, représentant l'état du nœud émetteur, au paquet de données (2 bits ajoutées à la trame MAC).

Nous pouvons conclure que PC est plus performant que n'importe quel algorithme de groupage actif. Il introduit une surcharge beaucoup moins importante et ainsi réduit la consommation d'énergie par les nœuds du réseau. Dans le chapitre suivant, nous proposons un nouveau protocole, appelé GPC (Geography-based Passive Clustering), qui intègre les informations de localisation géographiques des nœuds avec PC pour limiter la zone inondée et ainsi réduire le nombre de chefs de groupe et nœuds de liaison impliqués dans la diffusion des données. L'objectif est de réduire encore plus la consommation d'énergie par les nœuds du réseau. Si aucun nœud du réseau ne connaît sa localisation géographique, GPC devient équivalent à PC.

Chapitre 4 Groupage passif géographique

4.1 Introduction

Plusieurs applications, utilisant les réseaux mobiles sans fil (p.e. réseaux de capteurs), exigent la connaissance des coordonnées géographiques des nœuds du réseau.

Nous pouvons citer quelques exemples de leurs utilisations :

- (a) localiser l'origine d'un feu de forêt ou d'une inondation ;
- (b) vérifier la température d'un endroit géographique spécifique ;
- (c) déterminer la localisation d'un objet mobile (p.e. char ennemi dans un conflit);
- (d) localiser l'endroit précis d'une secousse sismique ; etc.

Ces applications fonctionnent correctement quand les nœuds du réseau connaissent leurs coordonnées géographiques (p.e. en utilisant GPS [20]). Autrement, les données produites par les nœuds du réseau sont inutiles; par exemple, la génération d'un signal d'incendie sans information sur l'endroit exact du feu n'aidera pas à éteindre ce feu. Pour ces applications, l'inondation des paquets de données dans le réseau peut être limitée à un espace géographique [14] bien déterminé; par exemple, si la destination est localisée au nord de la source, c'est inutile de diffuser les paquets au sud de la source. En limitant l'inondation, nous pouvons réduire considérablement la surcharge ainsi que la consommation d'énergie dans le réseau.

Dans les réseaux mobiles sans fil, plusieurs protocoles de routage géographique ont été proposés [13, 18] pour limiter l'inondation des paquets en utilisant les coordonnées géographiques des nœuds de réseau. L'idée fondamentale derrière ces protocoles est de limiter l'inondation à un espace géographique spécifique (rectangle ou cercle) entre la source et la destination.

Dans la section suivante, nous présentons les détails du protocole de Groupage passif géographique, appelé GPC. L'idée de base de GPC est de limiter l'inondation à une zone limitée du réseau entre la source et la destination; l'inondation est basée sur PC et la détermination de la zone géographique est similaire aux techniques utilisées par les protocoles de routage géographique. Plus spécifiquement, GPC réduit, par rapport à PC, le nombre de chefs de groupes et de nœuds de liaison impliqués dans

l'inondation. Ainsi, la consommation globale dans le réseau d'énergie est considérablement réduite avec GPC. GPC peut fonctionner même si on connaît les coordonnées géographiques de seulement certains nœuds du réseau; dans le cas où aucun nœud ne connaît ses coordonnées géographiques, GPC est équivalent à PC. Dans le cas où la destination n'est pas atteinte pour une raison quelconque (par exemple, une topologie particulière où la source ne peut pas communiquer avec la destination en utilisant seulement les nœuds dans l'espace géographique déterminé par GPC), GPC, itérativement, augmente l'espace géographique jusqu'à ce que la destination soit atteinte ou tout le réseau est inondé comme avec PC. Ce cas de figure est très peu probable surtout dans un réseau dense; GPC et PC produisent les meilleures performances dans le cas de réseaux denses.

4.2 GPC

Avec GPC, à chaque fois qu'un nœud du réseau reçoit un paquet, il doit vérifier s'il est dans la zone géographique inondable préalablement déterminée. Si la réponse est oui, le nœud exécute la procédure de groupage passif; sinon, il est mis à l'état ordinaire. L'objectif de ce test dans GPC est de réduire l'inondation dans le réseau en évitant que les nœuds qui sont à l'extérieur de la zone inondable ne participent à l'inondation; ainsi le nombre de chefs de groupe et nœuds de liaison peut être considérablement réduit. Nous obtenons ainsi une réduction de la surcharge dans le réseau ainsi qu'une diminution de la consommation de l'énergie totale du réseau.

Il existe deux approches pour déterminer la zone géographique inondable: (1) approche basée sur les cercles; et (b) approche basée sur les rectangles.

4.2.1 Approche basée sur les cercles

Dans cette approche, la zone géographique inondable représente un cercle qui a son centre égal à la destination. La valeur du rayon du cercle peut (1) être fixé pour la durée de découverte de chemin entre la source et la destination; ou (2) varier durant la découverte du chemin. Si nous utilisons la première variante, la valeur du rayon du cercle est initialement fixée à la distance entre la source et la destination; sinon, la

valeur du rayon va prendre une valeur égale à la distance entre le nœud courant (nœud qui reçoit le paquet originalement généré par la source) et la destination.

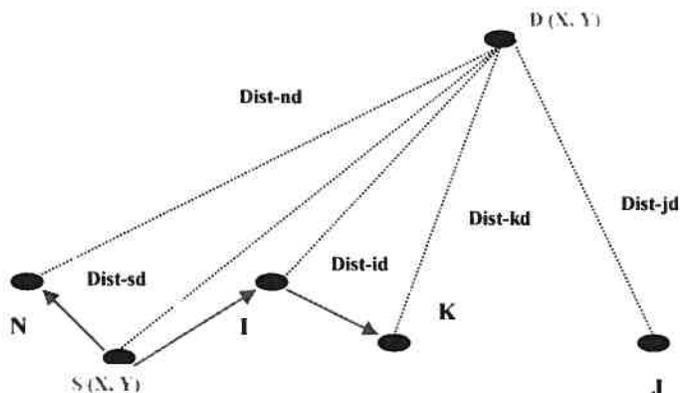


Figure 4.1. Approche basée sur le cercle

Pour une meilleure compréhension de la première variante considérons l'exemple de la figure 4.1; S représente la source et D la destination. Le nœud I reçoit le paquet, initialement généré par S, et calcule sa distance, Dist-id, relative à D pour vérifier s'il est à l'intérieur du cercle centré sur D et qui a un rayon égale à Dist-sd (distance entre S et D). Cette vérification consiste à compare Dist-id à Dist-sd; si Dist-id est inférieure à Dist-sd, I est à l'intérieur du cercle; sinon, il est à l'extérieur du cercle. Dans cet exemple, I (aussi K et N) détermine qu'il est à l'intérieur du cercle; donc, il exécute PC et éventuellement diffuse le paquet si son état change à chef de groupe ou nœud de liaison. Le nœud J fait la même vérification et détermine qu'il est à l'extérieur du cercle; son état est mis à ordinaire et ainsi il ne diffuse pas le paquet reçu.

Pour pouvoir réaliser cette vérification, un nœud a besoin des coordonnées de la destination et le rayon du cercle qui représente la zone géographique inondable. Pour ce faire, nous exigeons que chaque paquet diffusé contienne ces 2 attributs; initialement, ces attributs sont insérés par la source. Tableau 4.1 montre le pseudo code de ce test de la zone géographique au niveau d'un nœud du réseau.

<i>Input</i>	
<i>Dist-sd;</i>	<i>/* représente le rayon de la zone à inonder */</i>
<i>(Xl, Yl);</i>	<i>/* coordonnées du noeud courant */</i>

```

(Xd, Yd;)      /* coordonnées de la destination D */
Paquet;       /* paquet envoyé */
Nœud;         /* nœud courant ; il a un enregistrement qui contient deux
                attributs : (1) coordonnées: coordonnées du nœud ; et (2) état:
                état courant du nœud */

Begin
  Dist-sd2 := (Xd- Xs)2 + (Yd- Ys)2;
  /* calcul du rayon de la zone géographique inondable */
  While (true)
    If (Nœud reçoit paquet)
      If (Noeud.coordonnées= true)
        Dist-ld2 = (Xd- Xl)2 + (Yd- Yl)2;
        If (Dist-ld < Dist-sd)
          /* si le noeud appartient à la zone géographique */
          Exécute PC ;
        Else
          Noeud.état =ordinaire;
          /* le noeud est en dehors de la zone à inonder */
        Endif
      Else
        Exécute PC;
      Endif
    Endif
  Endwhile
End

```

Tableau 4.1. Pseudo code de GPC : Première variante

La deuxième variante de l'approche basée sur le cercle, permet de réduire (par rapport à la première variante) le nombre de nœuds, à l'intérieur de la zone inondable, qui sont impliqués dans la diffusion des paquets. Pour ce faire, au lieu que chaque paquet diffusé contient Dist-sd et les coordonnées de D, il contiendra Dist-cd (distance entre le nœud courant, qui diffuse le paquet, et la destination) et les coordonnées de D. Ainsi, un nœud qui reçoit un paquet compare sa distance à la destination avec la distance du nœud voisin (qui a diffusé le paquet) et la destination. Cette variante est plus optimale puisque le nombre de nœuds qui peuvent diffuser diminue; cependant,

dans un réseau qui n'est pas dense, la probabilité de ne pas atteindre la destination augmente.

Dans la figure 4.1, le nœud source S diffuse un paquet (incluant Dist-sd et les coordonnées de D) à ses voisins immédiats, I et N. Quand N reçoit le paquet, il calcule sa distance à la destination D, Dist-nd, et détermine qu'elle inférieure à la distance Dist-sd; dans cet exemple, N diffuse le paquet (incluant Dist-nd et les coordonnées de D) à ses voisins. Le nœud k (voisin de N) calcule sa distance à D et détermine qu'elle supérieure à Dist-nd; son état change à ordinaire et ainsi il ne diffuse pas. Tableau 4.2 montre le pseudo code de ce test de la zone géographique au niveau d'un nœud du réseau.

```

Input
Dist-sd;      /* represente le rayon de la zone à inonder */
(XI,YI) ;     /* coordonnées du nœud courant */
(Xd,Yd) ;     /* coordonnées de la destinations D */
Paquet ;      /* paquet envoyer par la source S */
Nœud ;        /* nœud courant ; il a un enregistrement qui contient deux
                attributs : (1) coordonnées: coordonnées du nœud ; et (2) état:
                état courant du nœud */

begin
D:= (Xd- Xs)2 + (Yd- Ys)2;
/* calcul du rayon de la zone géographique */
While (true)
  If (Noeud Receive (paquet)
    If (Noeud.coordonnée= true)
      Dist-ld2 = (Xd- XI)2 + (Yd- YI)2;
      If (Dist-ld < D)
        /* Si le noeud est dans la zone géographique */
        D:= Dist-ld;
        /* Rétrécir la zone géographique en remplaçant le rayon SD */
        Execute PC ;
        /* si le noeud est dans la zone géographique, executer PC */
      Else
        Noeud.etat =Ordinary_Node;
        /* si le noeud est en dehors de la zone, le
        bloquer */

```

```

                                Endif
                                Else Exécute PC;
                                Endif
                                Endif
                                Endwhile
                                End

```

Tableau 4.2. Pseudo code de GPC: Deuxième variante

Dans le cas où la destination n'est pas atteinte pour une raison quelconque (p.e. dans un réseau qui n'est pas dense), GPC, itérativement, augmente le rayon du cercle centré sur la destination D jusqu'à ce que la destination soit atteinte ou tout le réseau est inondé comme avec PC; un fonctionnement particulier de GPC (que nous avons implémenté) est de couvrir tout le réseau dans le cas où la première itération (cercle avec rayon égale à Dist-sd) ne réussit pas. Nous utilisons un temporisateur pour déterminer l'instant où nous devons augmenter la zone géographique inondable. Le tableau 4.3 montre le pseudo code de GPC en tenant compte des itérations.

```

Input
    comp      /* valeur du temporisateur */
    Paquet    /* paquet envoyé par la source S */
    (Xs,Ys)   /* coordonnées du nœud source S */
    (Xd,Yd)   /* coordonnées de la destination D */
    Nœud      /* Nœud courant */
    N         /* le nombre maximum d'itérations */
    Incrément /* valeur utiliser pour augmenter le rayon de la zone
              géographique*/

Variables
    I=0 ; /* entier */
    R : réel;

Begin
    R2 := (Xd- Xs)2 + (Yd- Ys)2;
    Nœud envoie paquet incluant R et (Xd, Yd);
    While (nœud ne reçoit pas la réponse correspondante)
        If (comp expire)
            R=R+incrément;
            Envoyer paquet; /* un autre essai pour trouver un chemin entre la source
                             et la destination*/
            I=I+1;
            If (I>=N)
                Exit; /*il y aucun chemin trouvé entre la source et la destination*/
            Endif
        Endif
    Endif
Endif

```

<p style="text-align: center;"><i>Endwhile</i> <i>End</i></p>

Tableau 4.3. Pseudo code de la variation de la zone géographique

4.2.2 Approche basée sur les rectangles

Sur la figure.4.2, nous avons le rectangle A, B, C et S qui forme une zone appelée zone de requête (request Zone), elle représente la zone à inonder par les paquets de découverte de routes. Une deuxième zone appelée zone prévue (expected zone) est formée en ayant les coordonnées du nœud destination et de la vitesse moyenne du nœud D; cette zone est l'endroit où on prévoit trouver la destination D. Les paquets diffusés contiennent les coordonnées des nœuds A (X_s, Y_s), B (X_d+R, Y_d+R), C (X_d+R, Y_s) et S (X_s, Y_s). Chaque nœud qui appartient à cette zone diffuse les paquets qu'il reçoit; par contre un nœud qui n'est pas dans ce rectangle ne diffuse pas.

Le rayon R, du cercle centré à D, est calculé en connaissant les coordonnées (X_d, Y_d) de la destination D à l'instant $T1$ avec la vitesse moyenne V de déplacement du nœud. En ayant la vitesse et l'intervalle de temps [$T1, T2$], nous pourrions calculer la distance parcourue par le nœud D pendant l'instant $T = T2 - T1$ et ainsi prévoir son emplacement.

D'après des simulations qui ont été reportées dans [13], l'approche basée sur le cercle est plus efficace et génère moins de collisions dans le réseau que l'approche basée sur le rectangle. L'un des inconvénients des rectangles est le fait qu'il faut coller dans chaque paquet quatre champs (coordonnées de A, B, C, et D); ceci augmente la taille des paquets et donc les collisions. Un autre inconvénient de l'approche basée sur le rectangle réside dans la difficulté à collecter les coordonnées des rectangles vu la mobilité des nœuds. Généralement, nous pouvons faire certaines approximations pour obtenir ces coordonnées; cependant, ces approximations augmentent les erreurs de calcul surtout dans le cas d'une forte mobilité.

Pour toutes ces raisons, nous avons optés pour l'approche basée sur le cercle pour réaliser GPC.

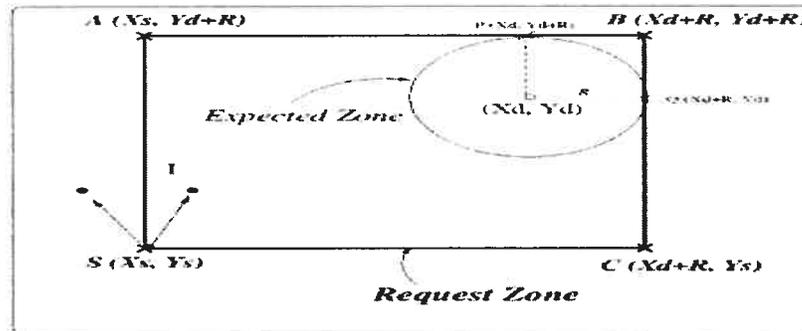


Figure 4.2. Approche basée sur le rectangle

4.2.3 Format de la trame dans GPC

Dans PC, les trames comportent une modification du format au niveau de la couche MAC [42] (couche 2). Le champ état (2 bits) qui représente les trois états possibles d'un nœud a été introduit. Pour réaliser GPC, nous proposons d'ajouter deux nouveaux champs : rayon de la zone géographique inondable (rayon du cercle centré sur la destination D) et les coordonnées (X, Y) de la destination. Avec ces ajouts, chaque nœud peut calculer sa distance à la destination, sans avoir des tables de routage comme c'est le cas des protocoles proactifs, et vérifier s'il est à l'intérieur de la zone inondable ou non.

Trame	État	Rayon	(X, Y)
-------	------	-------	--------

Figure 4.3. Trame dans GPC

La figure 4.3 montre le format de la trame utilisée par GPC. Le champ état peut être nœud ordinaire, nœud de liaison ou chef de groupe. Le troisième champ, Rayon, contient le rayon de la zone géographique inondable (initialement, le rayon est égale à la distance entre la source et la destination). Le dernier champ, contient les coordonnées du nœud destination D.

4.3 Simulateur GloMoSim

Les simulations ont été réalisées avec le simulateur GloMoSim (global Mobile simulator) [21, 22] qui est un environnement de simulation, qui permet le passage à

l'échelle (scalable), pour les réseaux filaires et sans fil très larges. GloMoSim utilise les simulations parallèles avec les modèles mathématiques à événements discrets.

Un simulateur à événements discrets est une technique de modélisation permettant de construire une abstraction (modèle) de la réalité et de faire évoluer cette abstraction en fonction du temps. Au regard de sa puissance d'évaluation des performances d'un système, cette méthode a été utilisée dans plusieurs domaines allant des systèmes simples, tels que la gestion des stocks, les problèmes d'ordonnements et l'étude du trafic routier, aux domaines plus complexes, tels que l'étude de la désintégration radioactive.

Techniquement, la simulation à événements discrets consiste à décrire le fonctionnement d'un système pas à pas dans le temps (c'est à dire décrire les changements dans le temps des variables d'entrée et de sortie nécessaires à la description du comportement du système). A la manière dont le temps est géré par une simulation à événements discrets, nous distinguons deux types de simulations:

- *Simulation dirigée par événements* : le temps de la simulation est géré à partir d'un échéancier qu'on peut représenter conceptuellement comme une liste linéaire des dates d'événements d'un système. L'incrémentación du temps de simulation se fait d'une date à l'autre. La tête de la liste représente l'événement courant et la fin de la liste représente l'événement le plus éloigné.
- *Simulation dirigée par une horloge* : le système dispose d'une horloge centrale qui progresse d'une unité de temps à chaque fois. À chaque incrémentación de l'horloge, le système explore la liste des événements pour voir si l'un d'eux apparaît.

GloMoSim est reconnu dans les milieux de recherche; il est en grande partie dédié aux réseaux mobiles ad hoc. Il est gratuit pour les institutions académiques et open source; nous pouvons modifier son code pour l'adapter à nos propres besoins. Il existe sur le marché une version commerciale de ce simulateur appelée QualNet.

Application	Telnet, FTP, CBR, HTTP
Transport	TCP/UDP
Réseau	AODV, Bellman, DSR, DSDV, OSPF
MAC	CSMA, FAMA, IEEE 802.11
RADIO	Avec et sans capture
Propagation	Random Waypoint

Figure 4.4. Couches de simulation de GloMoSim

L'architecture en couches (figure 4.4) fait de GloMoSim un simulateur évolutif dans le sens où chaque utilisateur peut intégrer ses propres événements au système de base [22]. Au niveau de la couche application, nous pouvons utiliser 4 modèles de trafic : telnet, ftp, http, ou CBR. Dans nos simulations, nous avons utilisé le modèle CBR. Pour la couche transport, GloMoSim offre les protocoles UDP et TCP. Nous avons au niveau de la couche réseau plusieurs protocoles à notre disposition: AODV, BELLMAN, DSR, DSDV et OSPF. Au niveau de la couche MAC, 3 protocoles sont supportés : CSMA, FAMA et IEEE 802.11. Dans nos simulations, nous avons utilisé le protocole IEEE802.11.

GloMoSim supporte aussi la mobilité, ce qui permet de définir différentes stratégies de mobilité. Pour supporter le parallélisme, GloMoSim se base sur le langage PARSEC [46] (parallel simulation Environment for Complex Systems) qui est lui-même basé sur le langage C. Le Parsec C offre un environnement de programmation semblable à celui de C.

4.3.1 Fonctionnement de GloMoSim

La figure 4.5 montre que GloMoSim utilise deux fichiers (Trafic.conf, Config.in) pour entrer les données de simulation; après traitement, il génère un fichier (GLOMO.STAT) qui contient toutes les statistiques relatives à un nœud.

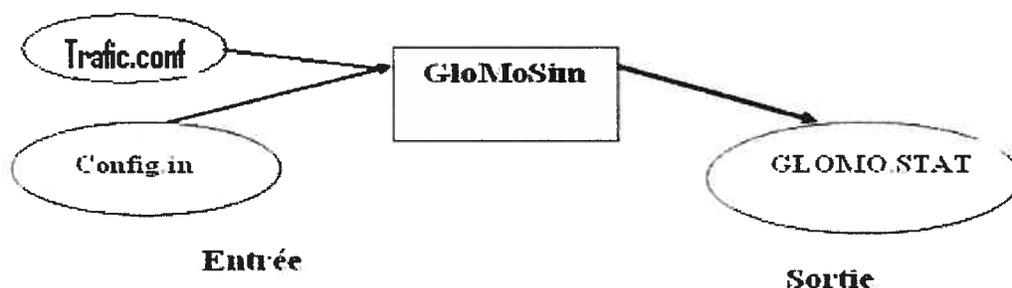


Figure 4.5. Fonctionnement de GloMoSim

Le fichier Config.in contient les différents paramètres de configuration pour une simulation donnée. Nous trouvons par exemple :

- SIMULATION-TIME : Temps de simulation
- PARTITION -NUM : nombre de partitions de simulation
- TERRAIN-RANGE-X : longueur du terrain de simulation
- TERRAIN-RANGE-Y : largeur du terrain de simulation
- NUMBER-OF-NODES : nombre de nœuds
- NON-MOBILITY : pas de mobilité
- NODE-PLACEMENT : emplacement des nœuds sur le terrain,
- MOBILITY-WP-MIN-SPEED : vitesse maximale des nœuds
- POWER-RANGE : rayon de couverture d'un nœud.

Dans le fichier trafic.conf, nous spécifions le nombre des sources et des destinations avec leurs identificateurs, le modèle de trafic (CBR, TELNET, http, FTP), le nombre de paquet que chaque source va envoyer et la vitesse de transmission (intervalle entre deux paquets successifs).

Pour lancer une simulation avec ce simulateur, il suffit de taper à la ligne de commande :% **glomosim config.in**.

4.4 Simulations

L'objectif des simulations est d'évaluer les performances de GPC et le comparer à PC. Plus particulièrement, nous voulons montrer que GPC permet de réduire

considérablement, par rapport à PC, la surcharge et la consommation d'énergie dans le réseau.

Nous exécutons les simulations dans un terrain de 600*600 m² avec 100 nœuds disposés aléatoirement. Le rayon de couverture de chaque nœud est de 100m et une vitesse maximale de 4m/s. Le Protocole MAC utilisé est le 802.11 alors que le protocole de transport est UDP. Le temps de simulation est fixé à 30 minutes et la bande passante du canal sans fil est fixée à 2Mb/s. Le modèle de trafic utilisé dans le fichier trafic est constant bit rate (CBR). Chaque nœud envoie 2 paquets par second pour un total de 100 paquets de 512; la taille d'un paquet est égale à 512 bits. Pour la carte radio, la configuration que nous avons utilisé dans Glomosim est : transmission (Tx)=18 mW, (Rx) réception=18mW et écoute (Idle)=5,5 mW.

Le protocole AODV est utilisé avec PC et GPC, nous avons 10 paires (source, destination) avec les paramètres α et β (voir heuristique de choix de nœuds de liaison dans PC) qui sont mis à 1.

4.5 Résultats des simulations

Les figures 4.6.1 et 4.6.2 montrent le nombre de chefs de groupe (CH) et le nombre de nœuds de liaison (GW) dans le réseau en fonction du nombre de nœuds. Le nombre de CH et GW augmente proportionnellement avec le nombre de nœud du réseau pour PC et GPC. Puisque nous utilisons un réseau dense (là où PC est recommandé), GPC exécute une seule itération de la zone géographique inondable pour trouver les chemins.

Pour 20 nœuds dans le réseau, PC et GPC sont proches (10 et 12) mais en augmentant le nombre de nœuds, l'écart est de plus en plus marqué. Pour 400 nœuds, on a 166 chefs de groupe (154 nœuds de liaisons) pour GPC et 295 chefs de groupes (340 nœuds de liaison) pour PC. Nous avons 36% de nœuds en plus qui diffusent les paquets de données dans le réseau.

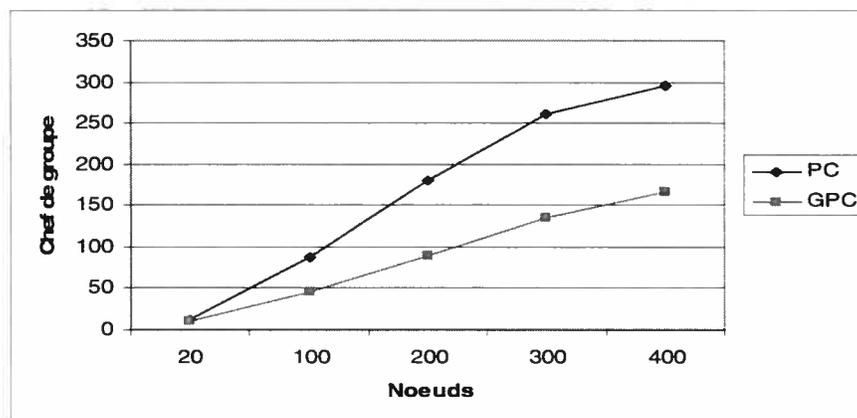


Figure 4.6.1. Variation du nombre de chefs de groupe Vs. Nœuds

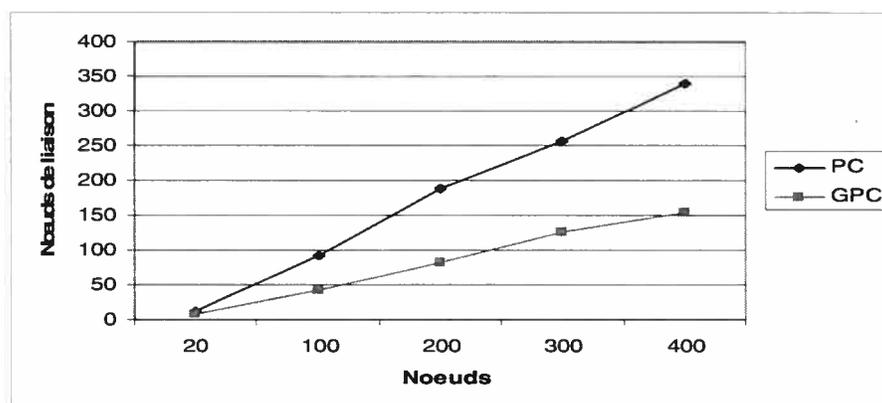


Figure 4.6.2. Variation du nombre de nœuds de liaison Vs. Nœuds

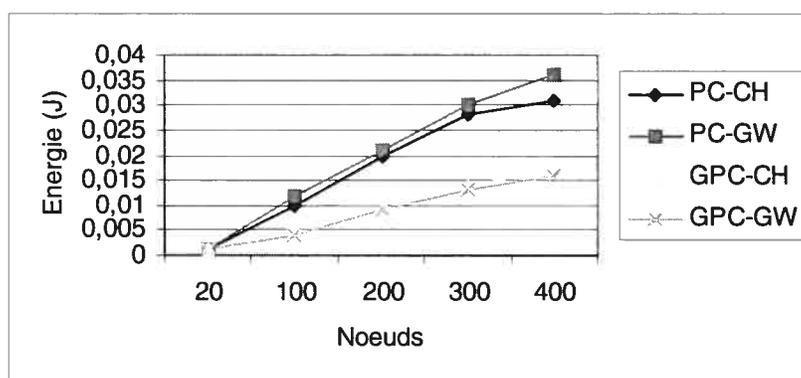


Figure 4.7. Énergie consommée des chefs de groupe et des nœuds de liaisons Vs. Nœuds

Dans la figure 4.7, nous avons l'énergie totale consommée par les nœuds chefs de groupe et nœuds de liaison en fonction du nombre de nœuds dans le réseau qui varie

de 20 à 400. Pour 20 nœuds, nous avons un niveau d'énergie qui est inférieur pour GPC que celui de PC seul.

Nous constatons que la différence de consommation d'énergie entre PC et GPC est proportionnelle avec l'augmentation du nombre de session.

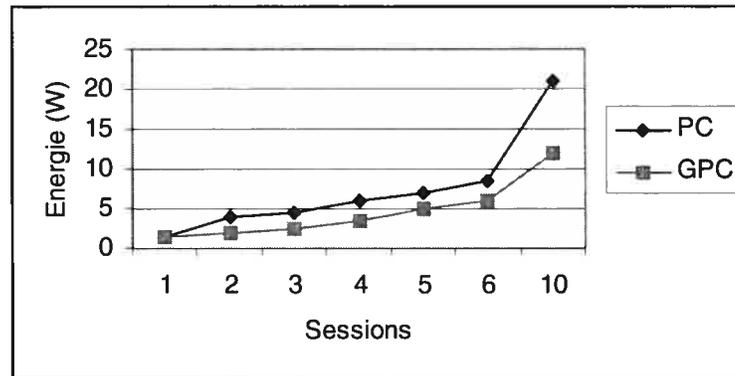


Figure 4.8. Consommation d'énergie Vs. Sessions

Dans la figure 4.8, nous avons l'énergie totale consommée en variant le nombre de sessions. Nous constatons que là aussi une plus grande consommation d'énergie est utilisée pour PC que dans le cas de GPC, cette consommation d'énergie s'accroît en augmentant le nombre de sessions dans le réseau. Pour 10 sessions, on a 2 fois plus d'énergie consommée pour PC (22 watt) que pour GPC (12 watt).

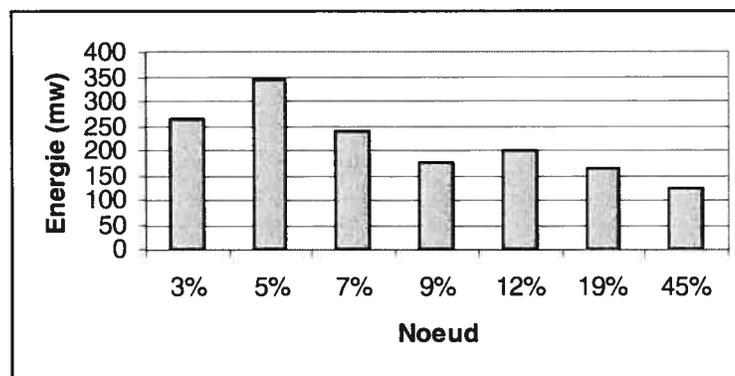


Figure 4.9.1. GPC : Énergie Vs. Pourcentage de nœuds

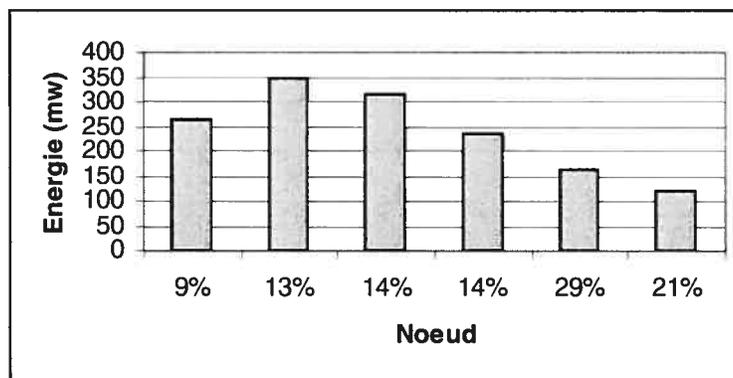


Figure 4.9.2. PC : Énergie Vs. Pourcentage de nœuds

Les figures 4.9.1-4.9.2 illustrent la distribution de l'énergie en fonction des nœuds de PC et GPC. On voit que pour PC il y a 27% de nœuds du réseau qui dépassent les 310 mW alors que seulement 5% des nœuds pour GPC dépassent les 310mW. Nous avons aussi 47% des nœuds pour GPC qui ne dépassent pas les 100 mW (nœuds qui écoutent seulement) alors que c'est 21% des nœuds pour PC. Ceci explique le gain d'énergie de GPC par rapport à PC. Nous avons 50% des nœuds pour PC qui passent le seuil des 200 mW alors que c'est 15% pour pc géographique.

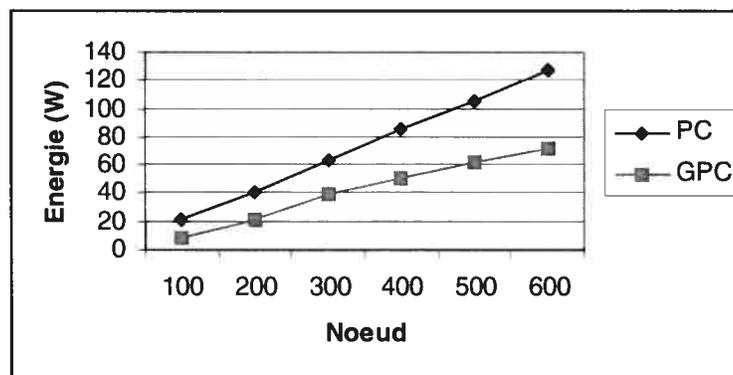


Figure 4.10. Énergie totale Vs. Nœuds

Sur la figure 4.10, la courbe de l'énergie consommée montre un écart d'énergie, entre PC et GPC, qui s'accroît avec le nombre de nœuds qui augmente. Dans le cas de 600 nœuds dans le réseau, PC consomme 127 w et GPC consomme 73 w; ceci présente un gain d'énergie de 54 w pour GPC. Nous voyons bien que le gain en énergie est considérable surtout pour les réseaux de capteurs et cela à cause de leur densité élevée.

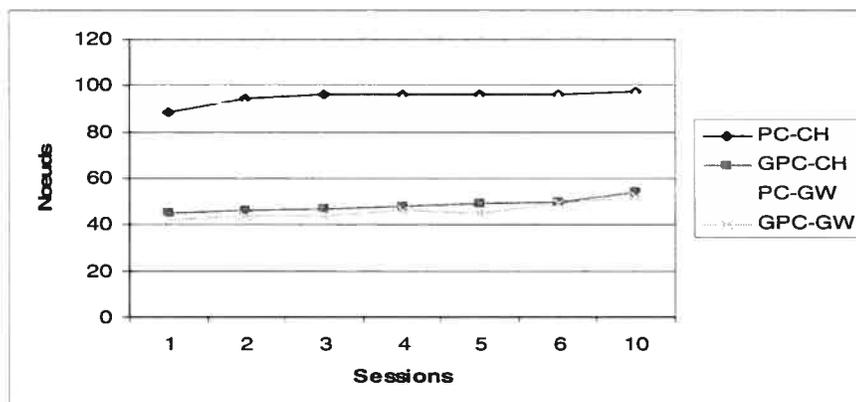


Figure 4.11. Chefs de groupe et nœuds de liaison Vs. Sessions

La figure 4.11 montre le nombre de chefs de groupe et de nœud de liaisons générés dans le réseau en variant le nombre de sessions de 1 à 10. GPC au niveau des chefs de groupe et nœud de liaisons ne dépasse pas les 52% des nœuds du réseau alors que pour PC, ce nombre a tendance à augmenter jusqu'à impliquer 99% des nœuds du réseau pour les chefs de groupe et 95% pour les nœuds de liaisons.

La figure 4.11 montre aussi, pour PC, que le nombre de nœuds de liaisons est presque le même pour une session ou 10 sessions; donc indépendamment du volume de trafic, PC inonde tout le réseau. Le nombre de nœuds de liaison augmente légèrement avec l'augmentation du nombre de sessions. Un nombre de chefs de groupe et nœuds de liaisons élevé se traduit par une grande consommation et un volume de paquets de contrôle important.

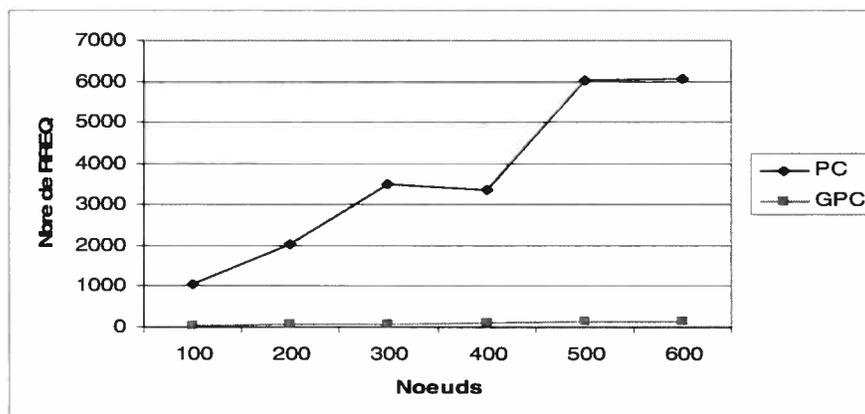


Figure 4.12. Nombre requêtes de route Vs. Noeuds

Pour la formation des groupes, on se base sur l'état des paquets. Dans PC, tous les nœuds envoient à leurs voisins un paquet de contrôle RREQ pour trouver un chemin. Pour apporter une autre preuve sur la limitation de l'inondation entre la source et la destination dans notre solution, GPC, et que les autres nœuds ne diffusent jamais les paquets de données, on calcul le nombre de paquet RREQ générés par tous les nœuds du réseau.

Le résultat paraît presque évident sur l'apport de GPC voir figure 4.12. Dans GPC, nous avons un nombre qui ne dépasse pas les 146 RREQ pour 600 nœuds alors qu'il dépasse les 6071 RREQ pour PC. C'est en limitant la zone parcourue par les paquets de contrôle RREQ qu'on réduit réellement le nombre de chefs de groupe et les nœuds de liaison dans le réseau.

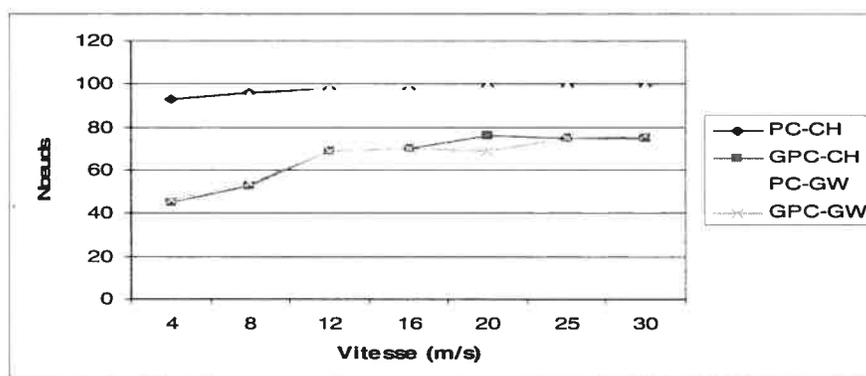


Figure 4.13. Chef de groupe et nœuds de liaison Vs. Vitesse

La figure 4.13 illustre la relation entre les chefs de groupe et nœuds de liaison en fonction de la mobilité des nœuds. Pour PC, dès qu'on arrive à une vitesse de 12 m/s, la totalité des chefs de groupe ou nœuds de liaison. Mais à partir de 16 m/s, on a 99% des nœuds qui deviennent chefs de groupe ou nœuds de liaison, en conséquence, tous les nœuds diffusent les paquets de données. Dans GPC, avec 4 m/s nous avons moins de 46% des nœuds qui sont chefs de groupe ou nœuds de liaison, leur nombre monte jusqu'à 65% en ayant une vitesse de 12 m/s. À partir de 25 m/s, les nœuds du réseau se stabilisent autour de 72% pour GPC alors que c'est 99% des nœuds pour PC.

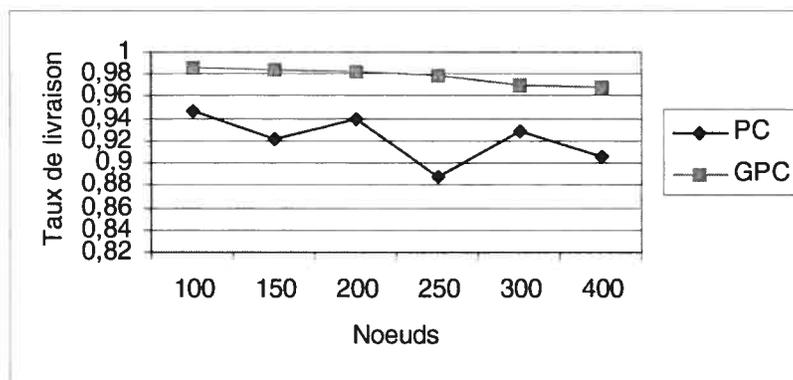


Figure 4.14. Ratio de livraison Vs .Noeuds

Pour le ratio de livraison, nous voyons sur la figure 4.14 que pour PC, ce ratio est instable. Il est de 95% pour 100 nœuds avec une vitesse de 4 m/s et descend à 89% pour 250 nœuds. Globalement, le ratio diminue avec l'augmentation du nombre de nœuds, ceci est dû essentiellement aux collisions dans le réseau qui devient de plus en plus dense et donc, plus d'interférences. Pour ce qui est de GPC, nous obtenons un ratio qui est plus élevé et ne descend pas en dessous de 97%, mais que là aussi la tendance est légèrement à la baisse avec la densification du réseau.

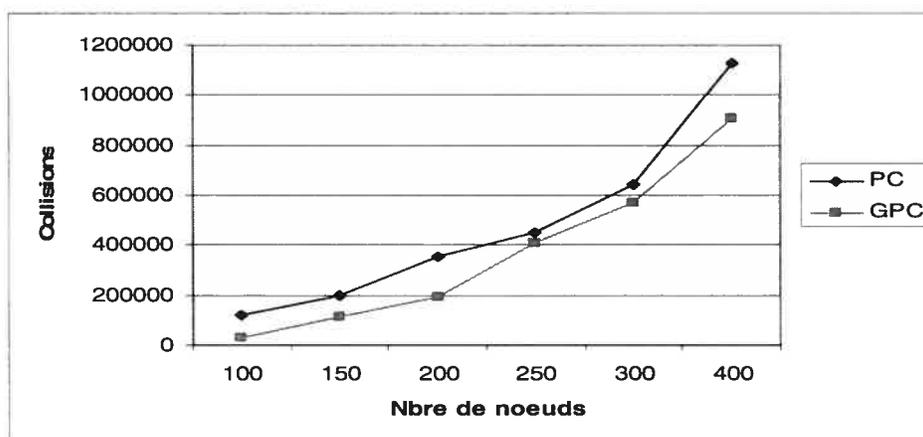


Figure 4.15. Collisions pour PC et GPC Vs. Nœuds

La figure 4.15 illustre le nombre de collisions pour les deux approches. Pour 100 nœuds, PC génère 3.5 fois plus de collisions que GPC alors que pour 400 nœuds on a 1.2 fois plus de collisions.

Dans la figure 4.16, nous faisons varier le nombre de nœuds qui connaissent leurs coordonnées géographiques et nous calculons à chaque simulation le nombre de chefs de

groupe et de nœuds de liaison obtenus ainsi que leur énergie. Nous avons dans cette simulation 600 nœuds dans le réseau avec une mobilité de 4 m/s.

Dans le cas où nous avons 0% de nœuds qui connaissent leurs coordonnées, cela veut dire que nous ne pourrions pas faire des calculs de zone géographique au niveau des nœuds qui reçoivent les trames, donc PC sera exécuté normalement. Ainsi, nous retomberons dans un cas où tout le réseau est inondé et tous les nœuds diffusent leurs paquets sans aucune restriction.

Dans le cas où 100% des nœuds connaissent leur coordonnées géographiques, nous sommes en présence du cas de GPC, tous les nœuds connaissent leurs coordonnées et peuvent faire des tests d'appartenance à la zone géographique et diffusent seulement s'ils sont à l'intérieur de cette zone géographique.

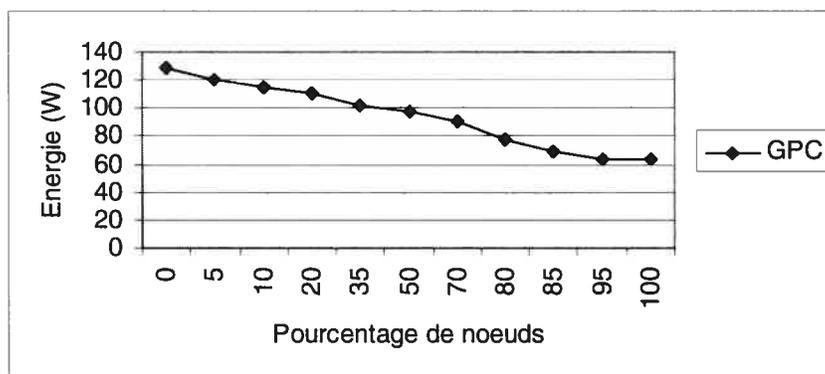


Figure 4.16. Énergie Vs. Nœuds connus

Les résultats obtenus sont illustrés dans la figure 4.16. Avec 0% de nœuds connus, nous avons une énergie totale consommée de 127 w et pour le cas 100% des nœuds connus, nous avons une énergie totale consommée de 62 w (un gain en énergie de 51%). Pour les cas intermédiaires, comme par exemple 50% de nœuds sont connus dans le réseau, nous constatons une consommation totale d'énergie égale à 100 w.

La courbe de l'énergie reflète bien celle des nœuds de liaison et des chefs de groupe, l'énergie totale consommée dans le réseau décroît graduellement en augmentant le nombre de nœuds dont les coordonnées sont connues. Notre approche peut être donc appliquée même si nous ne disposons pas des coordonnées géographiques de tous les nœuds du réseau, on voit bien que nous avons un gain de 27 w (correspond à un gain de

21%) avec seulement 50% des nœuds connus. Cette figure illustre un des grands avantages de notre solution GPC, même si nous avons seulement quelques nœuds dont les coordonnées sont connues, nous pouvons appliquer GPC et avoir un gain d'énergie substantiel.

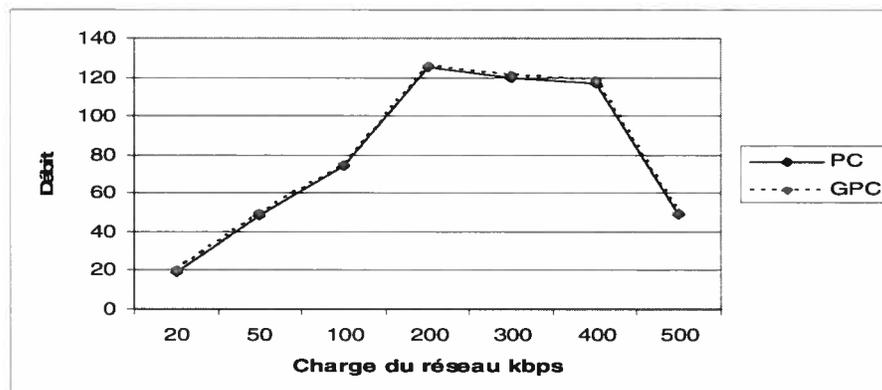


Figure 4.17. Débit Vs. Charge

Pour la figure 4.17, le débit est calculé pour 10 sessions qui envoient 1000 paquets pour un temps de simulation de 30 min. Les autres paramètres de simulation incluant le rayon de couverture des nœuds (100m), disposition des nœuds, terrain (600*600) et mobilité (4 m/s) ne changent pas. Nous constatons que le débit de PC et GPC reste inchangé ou constant. Ceci peut être considéré comme un point positif pour dire que notre solution, GPC, réduit le nombre de nœuds de liaison et chefs de groupe, par conséquent l'énergie, sans diminuer ou altérer les avantages et performances de PC.

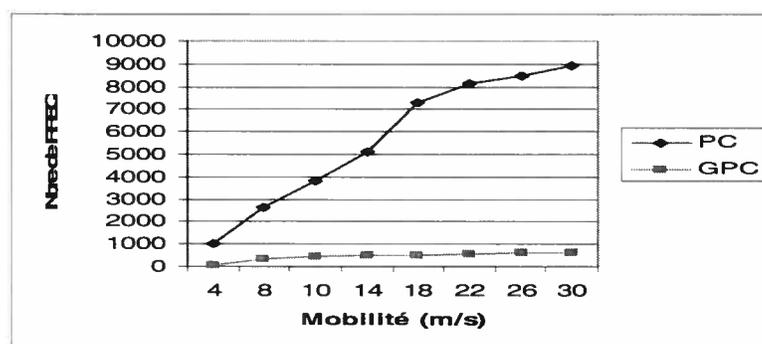


Figure 4.18. RREQ Vs. Mobilité

La figure 4.18 illustre le changement de la mobilité des nœuds (100 nœuds) dans le réseau et l'impact sur le volume de paquet de contrôle RREQ. On fait varier cette vitesse

de 4 m/s à 30 m/s. Nous constatons que pour PC, ce nombre est vraiment proportionnel à la vitesse des nœuds, c'est presque linéaire.

Dans PC géographique, nous avons un nombre qui ne dépasse pas les 631 paquets requêtes RREQ alors ce nombre dépasse les 8927 requêtes RREQ pour PC à 30m/s. Pour GPC, nous constatons une légère augmentation du nombre de requêtes RREQ mais ce dernier reste très inférieur par rapport au nombre de RREQ générées pour PC. Le gain est vraiment considérable pour GPC.

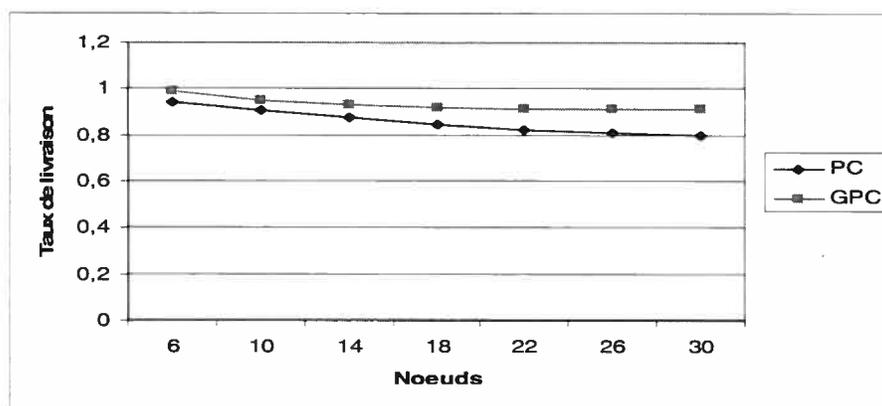


Figure 4.19. Taux Vs Mobilité

La figure 4.19 montre le taux de livraison pour les deux approches PC et GPC avec 100 noeuds. Nous voyons bien que le ratio descend pour les deux cas avec la mobilité qui augmente, il atteint à 80% pour PC et 90% pour GPC et cela à une vitesse de 30m/s. Ces analyses nous permettent d'affirmer ici que les performances de GPC sont nettement meilleures que PC seul.

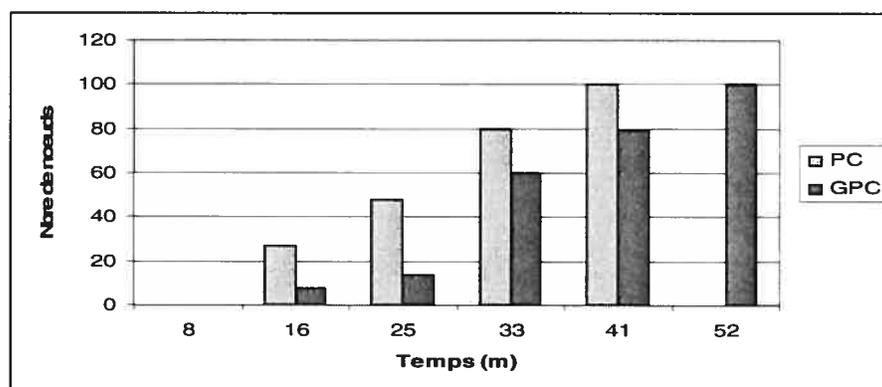


Figure 4.20. Nœuds Vs. Temps

La figure 4.20 illustre le nombre de nœuds qui meurent après un certain seuil fixé arbitrairement à 500 mW. Après un temps de simulation de 8 min, aucun nœud n'a consommé toute son énergie pour PC et GPC; cependant, à l'instant 16 min nous avons 27 nœuds morts pour PC et 8 pour GPC. Tous les nœuds meurent à l'instant 41 min alors que c'est 52 min pour GPC. La durée de vie des nœuds pour GPC est plus élevée que celle de PC (un gain de 26 % de temps en plus).

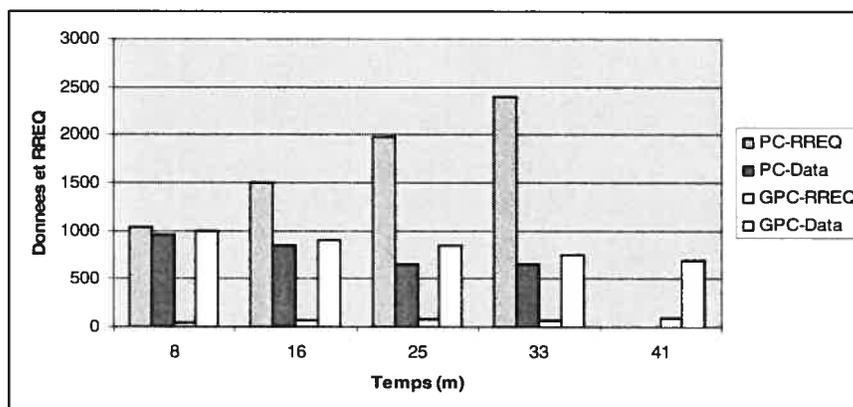


Figure 4.21. Paquets reçus et RREQ Vs. Temps

La figure 4.21 montre le nombre de paquets reçus et le nombre de paquets RREQ générés. Le nombre de paquets RREQ augmente plus rapidement pour PC que pour PC géo, nous avons 1035 (41 pour GPC) paquets à 8 min et 2400 (69 pour GPC) paquets RREQ à l'instant 33 min. Nous remarquons une diminution des paquets reçus par les destinations. Pour PC, nous avons 955 paquets reçus (992 pour GPC) à 8 min et 650 paquets reçus à (750 pour GPC) à l'instant 33 min.

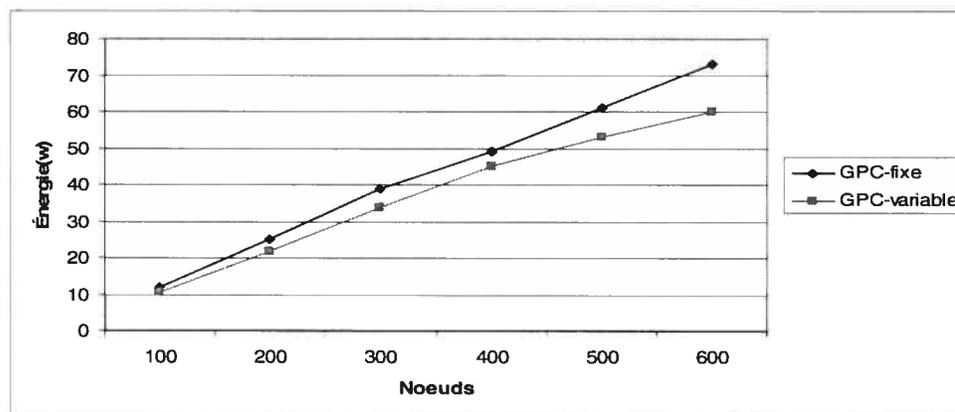


Figure 4.22. Énergie totale Vs. Nœuds

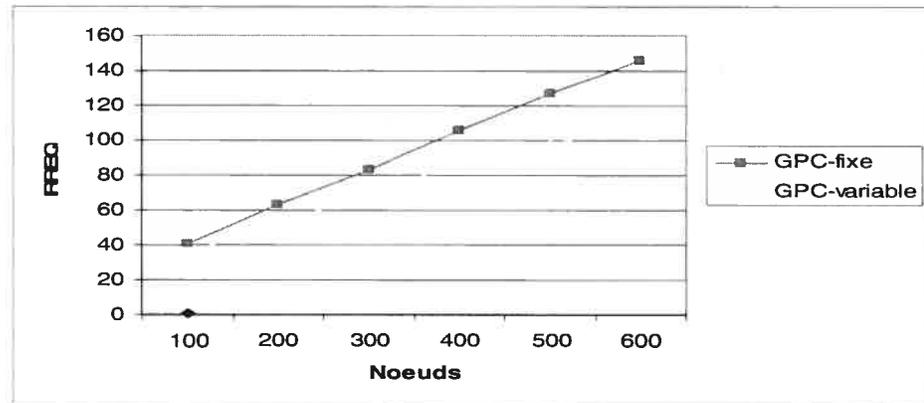


Figure 4.23. RREQ Vs. Nœuds

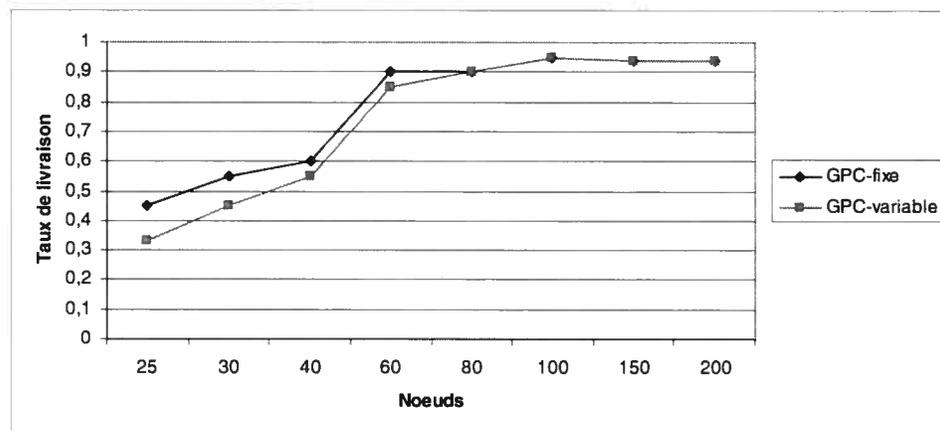


Figure 4.24. Taux de livraison Vs. Nœuds

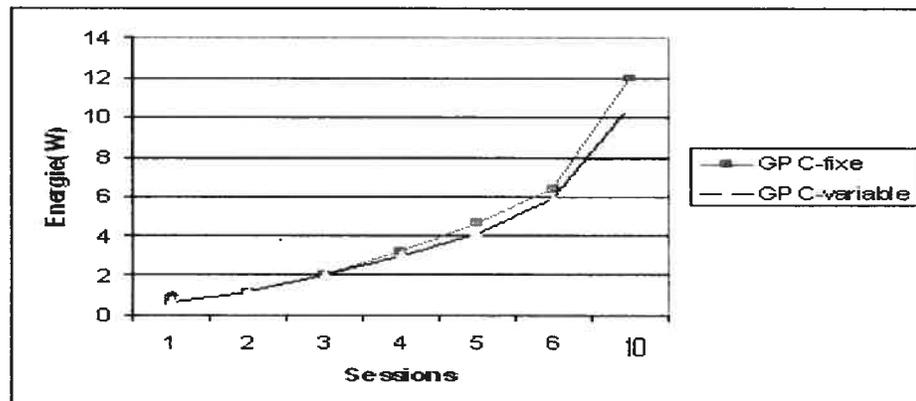


Figure 4.25. Énergie Vs. Sessions

Les figures 4.22-4.23-4.24-4.25 illustrent les différences entre les deux variantes de l'approche basée sur le cercle (voir section 4.2): Utilisation d'un rayon fixe (GPC-fixe) et l'utilisation d'un rayon variable (GPC-variable). L'envoi de la distance du

nœud courant à la destination (GPC-variable) apporte une légère amélioration par rapport GPC-fixe.

4.6 Conclusion

Les simulations ont montré que GPC permet de réduire considérablement la quantité d'énergie consommée dans le réseau avec un ratio de livraison légèrement supérieur à PC et cela en limitant l'inondation des paquets de contrôle à travers tout le réseau. Nous avons montré, à travers nos simulations, que la durée de vie du réseau est plus élevée avec GPC ce qui permet d'augmenter la durée du réseau et un fonctionnement normal. Ceci étant dit, les nœuds consomment l'énergie d'une façon non uniforme dans le cas de PC et GPC (Figures 4.9.1 et 4.9.2). En effet, des nœuds peuvent mourir (épuiser l'énergie de la batterie) bien avant d'autres nœuds et peuvent ainsi créer un réseau fragmenté/partitionné. Dans ce cas, il n'y a aucune garantie que deux nœuds puissent communiquer (il n'existe pas de chemin entre ces deux nœuds) même si plusieurs nœuds du réseau ont encore de l'énergie.

Nous savons que les nœuds doivent disposer de dispositif de localisation (relativement pas cher) pour connaître leur coordonnées, cela représente un coût mais qui reste négligeable par rapport au gain d'énergie dans GPC.

Dans le chapitre suivant, nous définissons un ensemble d'heuristiques qui permettent de balancer la consommation de d'énergie des nœuds et ainsi augmenter la durée de vie du réseau; tous les nœuds meurent sur un petit intervalle de temps.

Chapitre 5 Balancements de la charge dans un réseau mobile sans fil

5.1 Introduction

Nous avons montré dans le chapitre précédent que la consommation de l'énergie au sein des nœuds exécutant PC ou GPC n'est pas répartie uniformément sur tous les nœuds. En effet, la consommation d'énergie d'un nœud dépend de son état; les chefs de groupe et les nœuds de liaison consomment beaucoup plus d'énergie (que les nœuds ordinaires) car ils sont responsables de la diffusion des paquets. Cette situation peut avoir des conséquences néfastes sur la durée de vie du réseau et sur le ratio de livraison de paquets qui baisse avec la rupture de certaines sessions.

La rupture d'une session est causée par la mort d'un certain nombre de nœuds (énergie résiduel est égale à zéro), généralement des chefs de groupe et des nœuds de liaison, de telle façon qu'on ne peut pas trouver un chemin entre la source et la destination de cette session. Dans ce cas, nous avons un réseau partitionné où nous ne pouvons pas assurer l'existence de chemins entre deux nœuds donnés; cette situation peut arriver même si les nœuds morts constituent une petite fraction de l'ensemble des nœuds du réseau.

Pour remédier à ce problème, nous proposons une heuristique, appelée 1 Interval-based Load Balancing Heuristic (1-ILBH), qui permet de distribuer le trafic sur un maximum de nœuds du réseau de sorte à avoir un niveau de consommation d'énergie relativement égale pour tous les nœuds et ainsi augmenter la durée de vie du réseau. Nous proposons aussi une généralisation de cette heuristique qu'on appelle N Interval-based Load Balancing Heuristic (N-ILBH).

5.2 1-ILBH

Le balancement de la charge peut réduire les goulots d'étranglement, maintenir le réseau connecté et ainsi prolonger la durée de vie du réseau. Dans les réseaux mobiles sans fil, en particulier les réseaux de capteurs très denses et déployés aléatoirement,

l'énergie consommée est inégale entre les nœuds. Des travaux ([71, 72]) ont été réalisés pour le balancement de la charge dans les réseaux de capteurs; les travaux présentés dans [69, 70] se sont inspirés des modèles appliqués à un essaim de robots coopérant entre eux, alors que les auteurs dans [71, 72] proposent un balancement de la charge avec la réalisation dynamique d'un arbre d'équilibrage de la charge. Cependant, à notre connaissance il n'y pas de travaux qui ont traité le problème de balancement de charge dans le contexte de groupage passive.

Dans cette section, nous présentons 1-ILBH qui est une heuristique permettant de balancer la charge dans un réseau qui utilise le protocole de groupage passive pour la diffusion des paquets de données.. L'idée de base de 1-ILBH est d'éviter le problème de ruptures de session, nous devons éviter que les chefs de groupes et les nœuds de liaisons consomment toute leur énergie (batterie) et meurent alors que d'autres nœuds dans le réseau ont un bon taux d'énergie résiduel. Pour ce faire, 1-ILBH force un chef de groupe ou un nœud de liaison à changer son état à *ordinaire* quand son niveau d'énergie consommé atteint un seuil donné. Après ce changement, 1-ILBH interdit à ce nœud de changer d'état pendant un bout de temps; en effet durant cette période le nœud n'exécute pas PC. Cette interdiction est nécessaire, sinon le nœud peut immédiatement, après son changement d'état à *ordinaire*, changer son état à *chef de groupe* ou *nœud de liaison* en exécutant PC; dans ce cas l'opération de 1-ILBH deviendrait inutile. Après ce bout de temps, méticuleusement déterminé par 1-ILBH à l'état *ordinaire*, le nœud exécute PC et ainsi peut redevenir chef de groupe ou nœud de liaison. 1-ILBH définit deux seuils d'énergie pour déterminer la période où un nœud doit rester à l'état ordinaire (après être forcé à changer son état de chef de groupe ou de nœud de liaison); plus spécifiquement, 1-ILBH utilise deux constantes arbitraires α et β pour limiter la consommation d'énergie d'un nœud tel que $\alpha * \text{énergie_batterie}$ (énergie_batterie est l'énergie maximale de la batterie) constitue le premier seuil où un nœud est forcé de changer son état à ordinaire et $\beta * \text{énergie_batterie}$ le deuxième seuil où le nœud recommence à exécuter PC. Avec un seul seuil α , le nœud restera tout le temps bloqué et cela à cause du changement des paramètres radio.

Pour réduire la consommation d'énergie d'un nœud forcé, par 1-ILBH, à changer d'état à ordinaire, nous devrions reconfigurer sa carte radio. En effet, un nœud (p.e.

capteur) possède une carte radio qui peut être configurée selon les besoins en fonction de trois paramètres : temps d'écoute (idle), temps de réception (Rx) et temps d'envoi (Tx). L'énergie totale consommée par le nœud est la somme des énergies consommées durant la transmission et la réception des paquets et aussi durant le temps d'écoute (idle) : Énergie-idle+ Énergie-Tx + Énergie-Rx. Dans la configuration par défaut de notre carte radio, l'énergie consommée par l'écoute et la réception des paquets représente 56% de l'énergie totale du nœud. Un nœud à l'état ordinaire écoute et reçoit les paquets mais ne transmet/diffuse pas de paquets. 1-ILBH réduit la consommation du nœud forcé à l'état ordinaire en réduisant les valeurs des paramètres de temps d'écoute (idle) et temps de réception (Rx). En effet, 1-ILBH en plus de forcer un nœud qui atteint le premier seuil d'énergie à changer son état à ordinaire, il le force aussi à dormir. Cependant, il ne faut pas réduire trop les valeurs de temps d'écoute et temps de réception; sinon, une situation de blocage peut survenir (tous les nœuds dorment). Les simulations vont permettre de choisir les valeurs appropriées.

En résumé, 1-ILHB force un chef de groupe (ou un nœud de liaison), quand son énergie consommée atteint α *énergie_batterie, à changer d'état à ordinaire et réduit son temps d'écoute et de réception (en configurant sa carte radio). Après ce changement d'état, le chemin traversant ce nœud pourrait devenir invalide; par exemple, dans le cas où le nœud était chef de groupe, en changeant d'état à ordinaire, il arrête de diffuser les paquets de données vers la destination. Donc dans ce cas, un nouveau chemin doit être déterminé en utilisant PC. Après que le nœud ait atteint le deuxième seuil β *énergie_batterie, 1-ILHB réveille le nœud (en changeant les valeurs des paramètres de temps d'écoute et de réception aux valeurs par défaut au niveau du simulateur GloMosim) qui peut changer d'état en exécutant PC. Il est évident que 1-ILHB va générer plus de paquets de contrôle car à chaque fois un chef de groupe (ou nœud de liaison) est forcé de changer son état à ordinaire, il y a reconstruction des chemins. Cependant, même avec cette surcharge qui est minimale (voir section 5.3), 1-ILHB permet d'augmenter la vie du réseau (voir simulations dans la section suivante).

Le tableau 5.1 contient le pseudo code de l'heuristique 1-ILHB. Chaque nœud, *Nœud*, du réseau exécute ce pseudo code d'une façon continue.

```

Input
  Paquet;
  /*paquet reçu par le nœud*/
  batterie=500;
  /* énergie maximale de la batterie d'un nœud réseau */
   $\alpha$  et  $\beta$ ;
  /* constantes entre 0 et 1 */
  K=10;
  /* Facteur de réduction des temps d'écoute et temps de réception */

  Nœud;
  /* un enregistrement qui contient deux attributs : (1) niveauEnergie: valeur
    courante de l'énergie consommée par le nœud ; et (2) état: état courant du
    nœud */
  temps_écoute_par défaut;
  temps_réception_par défaut;
Variables
  Temps_écoute;
  Temps_réception;
Begin
  Noeud.état=initial;
  While (true)
    If (Nœud reçoit paquet)
      If ( (Noeud.niveauEnergie>  $\alpha$ *batterie) && (Noeud.niveauEnergie< $\beta$ 
        *batterie ))
        /* changer l'état et reconfigurer la carte radio */
        Noeud.état= ordinaire;
        Temps_écoute= temps_écoute_par défaut /k;
        Temps_réception= temps_réception_par défaut /k;
      Else
        /* Reconfigurer la carte radio */
        Temps_écoute=temps_écoute_par défaut;
        Temps_réception= temps_réception_par défaut;
        exécuter PC;
      Endif
    Endif
  Endwhile
End

```

Tableau 5.1. Pseudo code du balancement

Les valeurs de α et β doivent être méticuleusement choisis et cela en simulant plusieurs situations possible (Dans notre cas, des dizaines de simulations ont été nécessaires pour trouver la bonne combinaison); sinon, nous pouvons créer une situation où nous ne pourrions pas trouver de chemin entre une source et une destination même si tous les nœuds du réseau ont une énergie résiduelle différente de zéro. En effet, les nœuds forcés à l'état ordinaire ne peuvent pas participer à la création de chemins jusqu'à ce que leur énergie consommée atteigne le deuxième seuil d'énergie β *énergie_batterie.

Nous avons exécutés des simulations avec plusieurs valeurs de α et β pour déterminer les meilleures (voir Section 5.3).

5.3 Simulations

Les métriques de nos simulations sont les mêmes que dans le chapitre précédent, mais dans certaines courbes, nous avons augmenté le trafic pour que les nœuds mettent à terre leurs batteries (énergie consommée est égale à 500 mW).

un terrain de 600*600 m² avec 100 nœuds disposés aléatoirement. Le rayon de couverture de chaque nœud est de 100m et une vitesse maximale de 4m/s. Le Protocole MAC utilisé est le 802.11 alors que le protocole de transport est UDP. Le temps de simulation est fixé à 30 minutes et la bande passante du canal sans fil est fixée à 2Mb/s. Le modèle de trafic utilisé dans le fichier trafic est constant bit rate (CBR). Chaque nœud envoie 2 paquets par second pour un total de 100 paquets de 512; la taille d'un paquet est égale à 512 bits. Pour la carte radio, la configuration que nous avons utilisé dans Glomosim est : transmission (Tx)=18 mW, (Rx) réception=18mW et écoute (Idle)=5,5 mW.

Les premières séries de simulations ont été effectuée pour choisir les valeurs de α et β qui permettent une meilleur distribution de la consommation d'énergie sur les nœuds du réseau. Cependant, avant d'exécuter ces simulations, le facteur de réduction du temps d'écoute et du temps de réception doit être déterminé. Après plusieurs séries de simulations, nous avons déterminé que la valeur 10 est adéquate. Une grande valeur du facteur est souhaitable car elle permet une réduction considérable de la consommation d'énergie des nœuds forcés, par 1-ILHB, à changer leurs états à ordinaire. En effet, les simulations ont montré qu'avec des valeurs grandes du facteur de réduction, le nœud reste trop longtemps silencieux et consomme très peu d'énergie; donc, il serait difficile pour le nœud d'atteindre le deuxième seuil d'énergie β *énergie_batterie et, ainsi, de revenir à un fonctionnement normal (exécution de PC). Cette situation causerait un blocage du réseau : plusieurs nœuds, dans un état *ordinaire* forcé, ne peuvent pas être utilisés pour la transmission de données (ne peuvent pas changer leurs états à chef de groupe ou nœud de liaison).

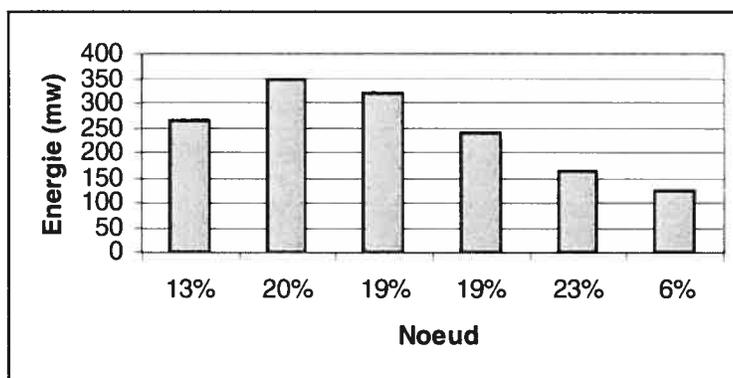


Figure 5.1. Distribution de la consommation de l'énergie pour PC (10 sessions)

La figure 5.1 montre la consommation d'énergie des nœuds du réseau qui exécutent PC. La consommation varie entre 350mW et 124mW; ceci représente un écart de 226mW. Dans ce qui suit nous présentons les résultats de consommation d'énergie des nœuds dans un réseau qui implémente 1-ILHB pour des valeurs différentes de α et β .

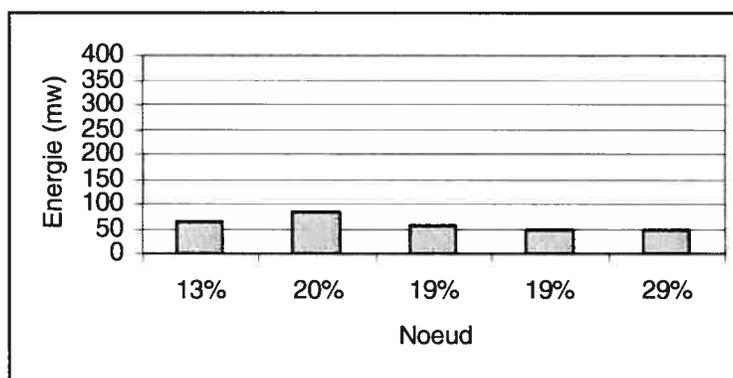


Figure 5.2. 1-ILHB: $\alpha=0.1, \beta=0.9$.

La figure 5.2 montre la consommation de l'énergie des nœuds pour $\alpha=0.1$ et $\beta=0.9$. Nous avons une consommation d'énergie qui ne dépasse pas les 85mW et un écart de 35 mW, mais le pourcentage de paquets délivrés est vraiment bas, 20% (Figure.5.10). Dans ce cas précis, nous sommes en présence d'une situation où tous les nœuds du réseau ont une consommation d'énergie entre $\alpha \cdot \text{énergie_batterie}$ et $\beta \cdot \text{énergie_batterie}$; ceci représente une situation de blocage où aucun nœud ne peut prendre le rôle de chef de groupe ou nœud de liaison ce qui rend impossible la diffusion des données (toutes les sessions sont rompues).

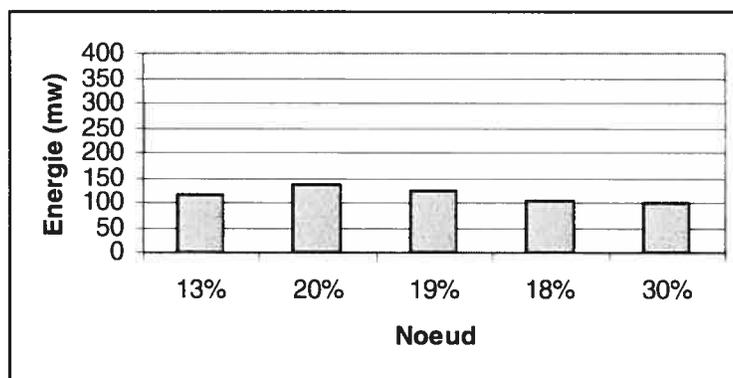


Figure 5.3. 1-IHLB : $\alpha=0.2, \beta=0.8$

Pour les valeurs de $\alpha=0.2, \beta=0.8$ (moyenne =100mW et écart = 24mW, paquets reçus = 40%), $\alpha=0.3, \beta=0.5$ (moyenne =150mW, écart = 35mW et paquet reçus = 55%), $\alpha=0.4, \beta=0.7$ (moyenne =200mW, écart = 31mW et paquet reçus = 84%), nous avons une situation de blocage où toutes sessions sont rompues mais à des instants différents (Figures 5.4-5.5-5.6).

Le pourcentage de paquets reçus pour les trois valeurs de $\alpha =0.1, 0.2, 0.3$ est inférieur à 55% (Figure 5.10); ceci est dû au fait que nous avons des ruptures de sessions. En effet, à partir d'un certains temps t , plusieurs nœuds sont dans un état ordinaire forcé et ne peuvent pas contribuer à la diffusion de données jusqu'à ce qu'ils atteignent le deuxième seuil ($\beta \cdot \text{énergie_batterie}$). Puisque l'écart entre α et β est important, le temps pendant lequel un nœud reste silencieux est considérable.

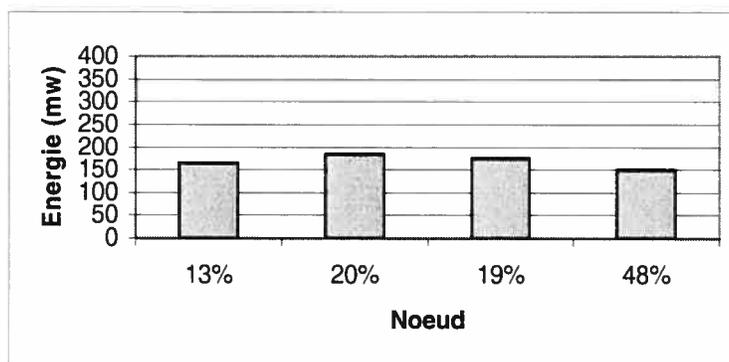


Figure 5.4. 1-ILHB : $\alpha=0.3, \beta=0.5$

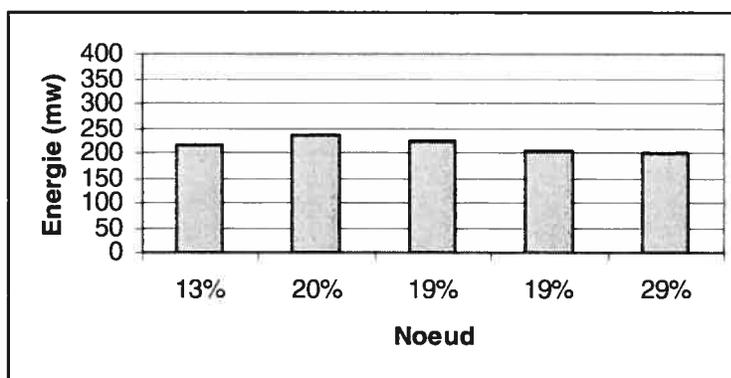


Figure 5.5. 1-ILHB : $\alpha = 0.4, \beta = 0.7$

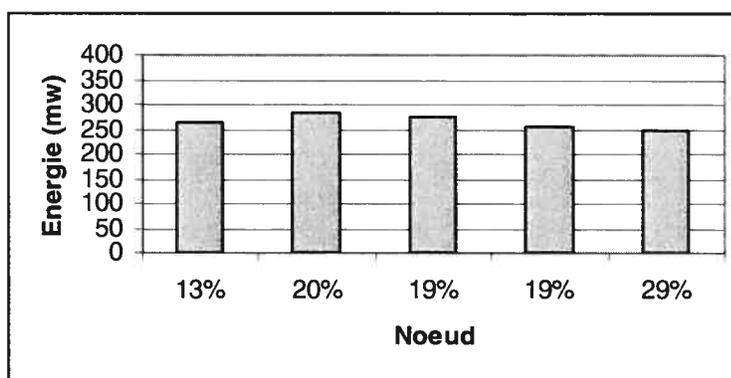


Figure 5.6. 1-ILHB: $\alpha = 0.5, \beta = 0.6$

La figure 5.6 montre une bonne distribution de la consommation d'énergie pour les valeurs $\alpha = 0.5$ et $\beta = 0.6$. En effet, la consommation de tous les nœuds est proche de 250 mW. En plus, nous avons un taux de livraison de paquet presque similaire à celui de PC sans 1-ILHB (Figure 5.10).

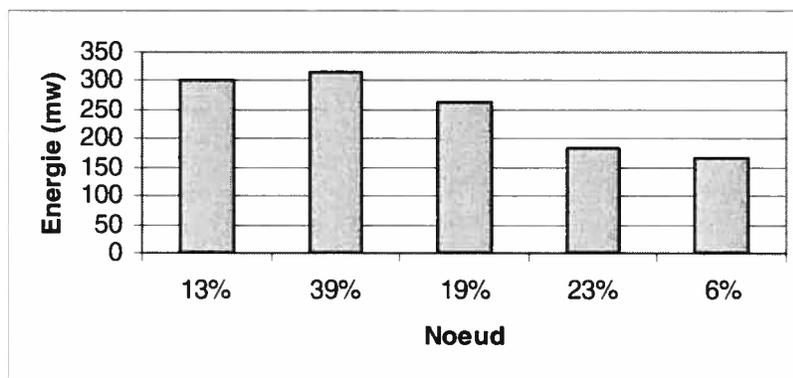


Figure 5.7. 1-ILHB : $\alpha = 0.6, \beta = 0.7$

Pour le cas $\alpha=0.6, \beta=0.7$, nous obtenons un taux de livraison similaire à celui de PC sans 1-ILHB (figure 5.10); cependant, la distribution de la consommation d'énergie n'est pas uniforme.

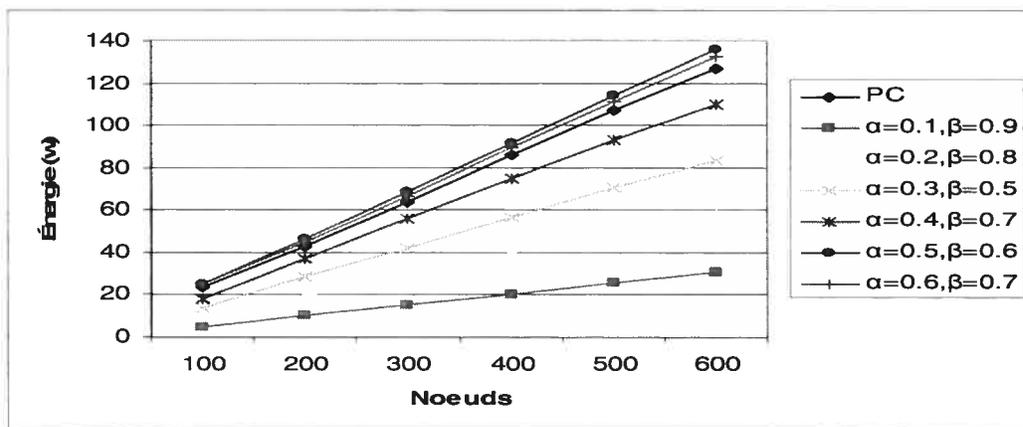


Figure 5.8. Énergie totale consommée pour les différentes valeurs de α et β

La figure 5.8 montre l'énergie totale consommée du réseau (somme des énergies consommées par tous les nœuds du réseau) pour les différentes valeurs de balancement α et β . Nous voyons bien que pour les cas où il y a eu ruptures de sessions, nous avons une énergie totale consommée inférieure à celle de PC alors que pour les valeurs de $\alpha=0.5$ et 0.6 , nous avons une énergie consommée légèrement supérieure. Cet écart d'énergie est dû aux paquets de contrôle émis pendant la période de balancement où de nouvelles découvertes de chemins, avec des nœuds qui ont une énergie plus petite, sont effectuées.

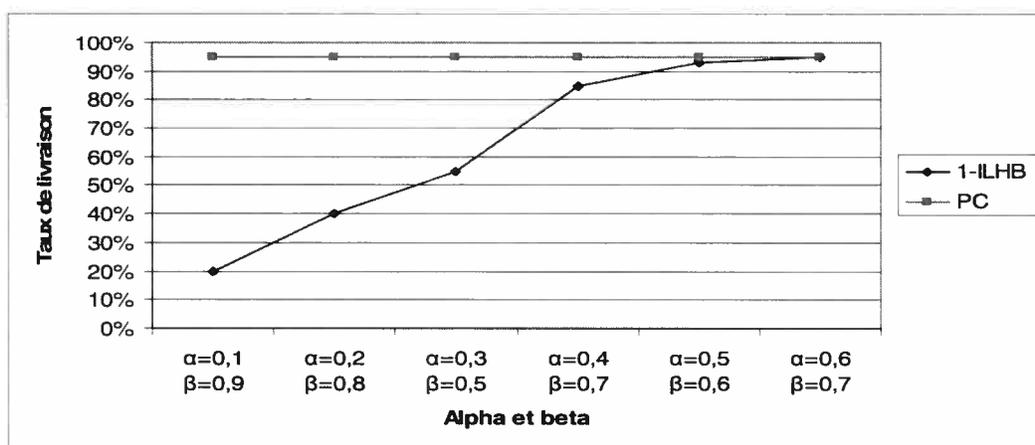


Figure 5.9. Taux de livraison Vs. α et β

La figure 5.9 montre le pourcentage de paquets de données reçues par les sources en fonction de α et β . Il est évident que pour α inférieur à 0.4 , nous avons un taux inférieur

à 85% et qui descend jusqu'à 20% pour $\alpha=0.1$. Ce taux de paquet devient acceptable pour $\alpha=0.5$ ($\beta=0.6$) et 0.6 ($\beta=0.7$) qui atteint 93%.

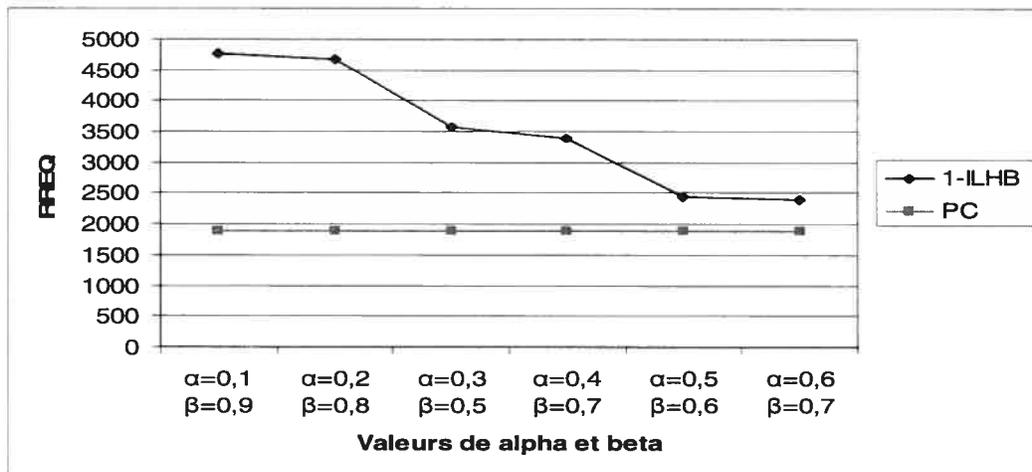


Figure 5.10. Nombre de paquets RREQ Vs. α et β

La figure 5.10 montre le nombre de paquets de contrôle pour les différentes valeurs de α et β . Nous voyons bien qu'en présence de ruptures de sessions, le volume de paquets de contrôle est très élevé. Ce nombre reste légèrement élevé pour $\alpha=0.5$ ($\beta=0.6$) et $\alpha=0.6$ ($\beta=0.7$) à cause de nouvelles découvertes de chemins quand des chefs de groupes ou des nœuds de liaison sont forcés à l'état ordinaire par 1-ILHB.

D'après les différentes simulations, le meilleur balancement de consommation d'énergie est réalisé pour $\alpha=0.5$ et $\beta=0.6$. Tous les nœuds ont une énergie qui avoisine les 250mW et la différence d'énergie est de 34mW entre la consommation maximale et minimale. Le pourcentage de paquets reçus est similaire au cas de PC sans 1-ILHB; il y a une légère augmentation de consommation d'énergie totale du réseau à cause d'une augmentation des messages de contrôle.

5.4 Durée de vie du réseau

La durée de vie d'un réseau de capteurs sans fil (ou un réseau ad hoc) est limitée à cause de l'utilisation de batteries. Un réseau est considéré mort dès qu'un nœud du réseau meurt, ou une fraction des nœuds du réseau meurt, ou la couverture/connectivité du réseau est perdue [5]. Pour réduire la consommation d'énergie et améliorer ainsi la durée de vie du réseau de capteurs sans fil, plusieurs méthodes ont été développées ces dernières années [66, 67,68]. Ces méthodes utilisent, en général, une ou plusieurs des

actions suivantes : (a) Former des groupes avec des chefs de groupes et des nœuds de liaison; (b) Développer des techniques qui permettent de gérer l'état d'un nœud entre un état actif et un état passif (sommeil); (c) Choisir les routes qui consomment la moins d'énergie (p.e. chemins les plus courts); (d) Utiliser sélectivement les nœuds sans fil en se basant sur leur niveau d'énergie; et (e) Réduire la quantité de données (p.e. agrégation/compression de données)

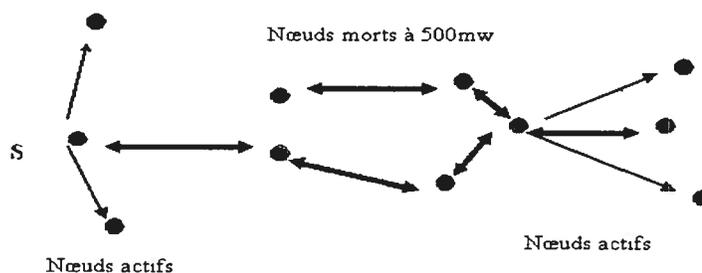


Figure 5.11. Perte de connectivité/couverture dans le réseau

Pour mieux expliquer le problème qu'engendre la mort de certains nœuds du réseau, nous considérons l'exemple montré dans la figure 5.11. Les nœuds en rouge sont morts puisqu'ils ont consommé toute l'énergie de leurs batteries. Les nœuds noirs ont encore de l'énergie; cependant, un nœud noir du côté gauche de la figure ne peut pas communiquer avec un nœud noir du côté droit de la figure. Dans ce cas, nous dirons que le réseau est fractionné, partitionné, ou déconnecté.

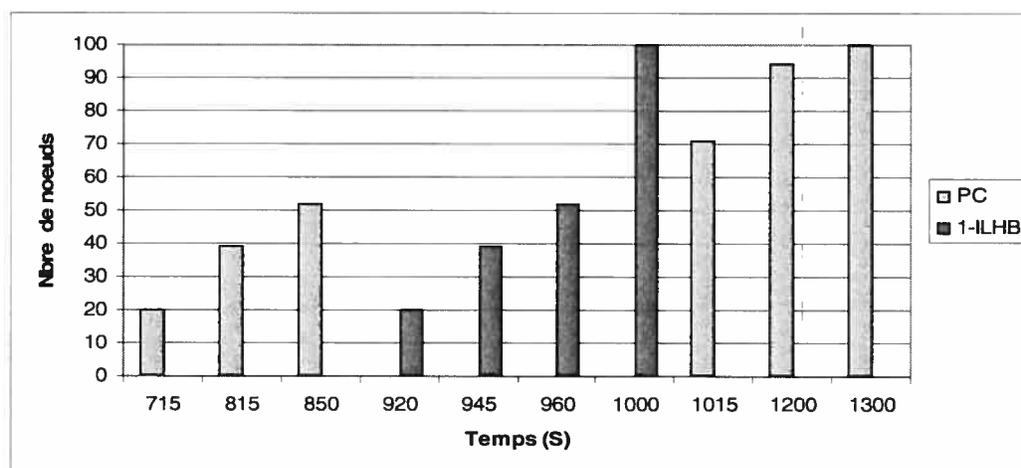


Figure 5.12. Nombre de nœuds morts Vs. Temps

La figure 5.12 illustre le nombre de nœuds morts dans le réseau pour PC avec et sans balancement de la charge en fonction du temps. À l'instant $t= 715$ s, nous enregistrons 20 nœuds morts pour PC et ce nombre passe à 40 à $t= 815$ s et ensuite à 50 à $t= 850$ s. Nous remarquons qu'au même instant, nous n'enregistrons aucun nœud qui meure pour PC avec balancement (1-ILHB); les premiers 20 nœuds qui meurent surviennent à $t= 920$ s (205 s après les premiers nœuds morts de PC sans 1-ILHB) et 40 nœuds à $t= 945$ s. Avec 1-ILHB, les nœuds du réseau meurent tous entre l'instant 920s et 1000s alors que dans le cas sans balancement de charge les nœuds meurent entre 715s et 1300s. Ces résultats montrent qu'avec 1-ILHB le réseau fonctionne plus longtemps.

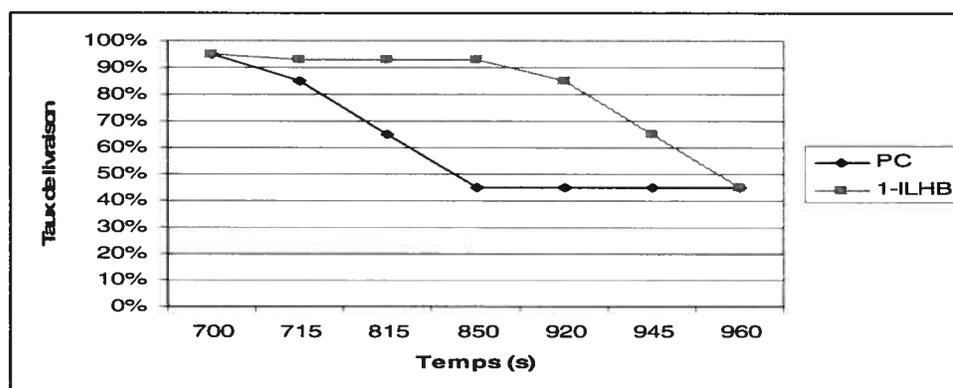


Figure 5.13. Taux de livraison Vs. Temps

La figure 5.13 illustre le taux de livraison pour PC et 1-ILHB. Nous remarquons qu'à l'instant 880s, dans le cas de PC sans balancement 50% des nœuds sont morts; ceci cause la chute du taux de livraison jusqu'à 46% alors qu'il reste à 93% pour PC balancé. La chute du taux de livraison est causée par la perte de connectivité du réseau; ainsi des sessions ne peuvent pas être rétablies. Le taux de livraison commence à baisser pour PC balancé à l'instant 920s, mais reste loin au dessus de PC sans balancement.

D'après ces résultats obtenus, nous pouvons dire que 1-ILHB permet un fonctionnement normal du réseau pour une période de temps plus longue indépendamment de la définition de la mort du réseau utilisé.

5.5 N-ILHB

Dans la section précédente, nous avons vu que le balancement apporte une bonne solution au problème de consommation d'énergie des nœuds chefs de groupe et nœuds de liaison. 1-ILHB permet une distribution plus homogène de la consommation de

l'énergie sur la majorité des nœuds qui composent le réseau et ainsi évite que des nœuds tombent en panne rapidement.

Nous allons introduire dans cette section plusieurs intervalles de balancement pour voir quelles seront les conséquences sur la durée de vie du réseau et sur le taux de livraison. En effet, cette approche représente une généralisation de 1-ILHB qui considère un seul intervalle d'où le nom N-ILHB; dans ce qui suit, nous limiterons N à 4. Le tableau 5.2 montre le pseudo code de N-ILHB. Depuis son initialisation, un nœud exécute le programme, qui implémente le pseudo code, jusqu'à sa mort.

Dans N-ILHB, un nœud fonctionne normalement jusqu'à ce qu'il atteigne un seuil de blocage α_i ($i=1, 4$) où il reste inactif (il est forcé). Le nœud ne pourra plus participer à la construction de chemins jusqu'à ce qu'il arrive au seuil de déblocage β_i ($i=1, 4$). Dans ce cas de $N=4$, un nœud passe par quatre blocages successifs, donc des ruptures de chemins, ce qui explique le niveau élevé de paquets RREQ. Nous pouvons dire que le paramètre N doit être grand pour des nœuds qui fonctionnent pour une période de temps élevée.

```

Input
    Paquet;          /*paquet reçu par le nœud*/
    batterie =500;   /*énergie maximale de la batterie d'un nœud réseau */
    ( $\alpha_1, \beta_1, \alpha_2, \beta_2, \alpha_3, \beta_3, \alpha_4, \beta_4$ );
                    /* constantes entre 0 et 1 */
    k =10 ;         /*Facteur de réduction des temps d'écoute et temps de réception */
    N = 4 ;         /* Nombre de balancement */
    Nœud;
    /* un enregistrement qui contient deux attributs : (1) niveauEnergie: valeur courante de l'énergie
    consommée par le nœud ; et (2) état: état courant du nœud */
    temps_écoute_par défaut;
    temps_réception_par défaut;
Variables
    Temps_écoute;
    Temps_réception;

Begin
    Introduire (N);
    Introduire ( $\alpha_1, \alpha_2, \alpha_3, \alpha_4$ );
    Introduire ( $\beta_1, \beta_2, \beta_3, \beta_4$ );
    Nœud. état= Initial;
    While (true)
        If (Nœud reçoit paquet)
            If (N=1)
                If ( (Noeud.niveauEnergie>  $\alpha_1$ *batterie) && (Noeud.niveauEnergie< $\beta_1$ 
                *batterie ))
                    /* Reconfigurer la carte radio et changer état */
                    Nœud.état= ordinaire;

```



```

                                default /k;
                                Else
                                /* reconfigurer la carte radio */
                                Temps_écoute=temps_écoute_par défaut
                                Temps_réception=temps_réception_par
                                défaut;
                                Execute PC;
                                Endif
                                Endif
                                Endif
                                Endif
                                Endif
                                Endwhile
                                End

```

Tableau 5.2. Pseudo code de N-ILHB

5.6 Simulations et résultats

Dans cette section, nous présentons les résultats des simulations en utilisant un seul intervalle, 2 intervalles, 3 intervalles, et 4 intervalles. Les intervalles sont choisis de telle sorte que l'écart entre les valeurs α et β soit égale à 0.1. Le choix de la valeur de l'écart est le résultat des simulations présentées dans la section 3; en effet, nous avons remarqué que si l'écart est plus grand que 0.1, une situation de blocage pouvait survenir.

Les paramètres et l'environnement de simulation sont les mêmes que ceux des simulations présentées dans la section 3. Les valeurs de α et β des 4 intervalles, sous considération, ont été choisies après avoir effectué plusieurs simulations préliminaires. Les intervalles pour les différentes heuristiques sont :

- 1-ILHB: $\alpha = 0.5, \beta = 0.6$.
- 2-ILHB: $\alpha_1 = 0.26, \beta_1 = 0.36$; $\alpha_2 = 0.54, \beta_2 = 0.64$.
- 3-ILHB: $\alpha_1 = 0.2, \beta_1 = 0.3$; $\alpha_2 = 0.4, \beta_2 = 0.5$; $\alpha_3 = 0.6, \beta_3 = 0.7$
- 4-ILHB: $\alpha_1 = 0.17, \beta_1 = 0.27$; $\alpha_2 = 0.34, \beta_2 = 0.44$; $\alpha_3 = 0.5, \beta_3 = 0.6$; $\alpha_4 = 0.66, \beta_4 = 0.76$

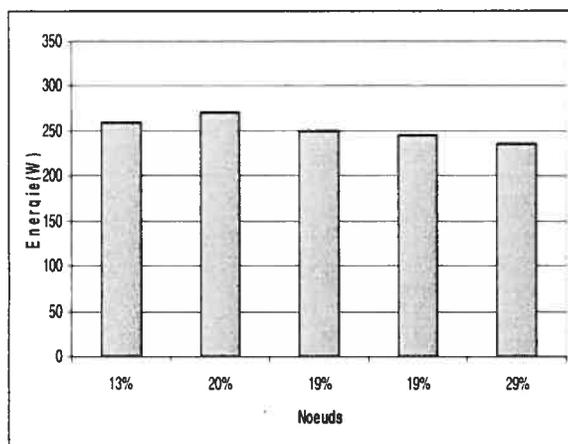


Figure 5.14.1. 1-ILHB

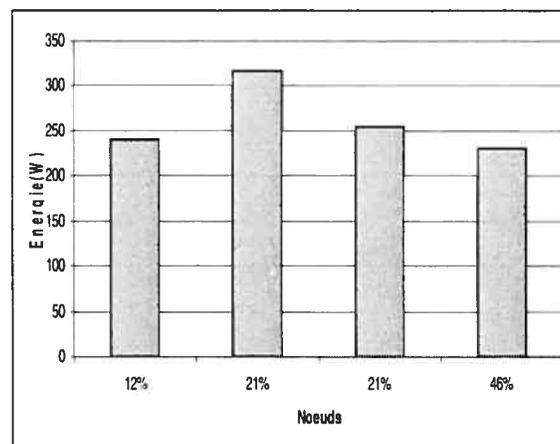


Figure 5.14.2. 2-ILHB

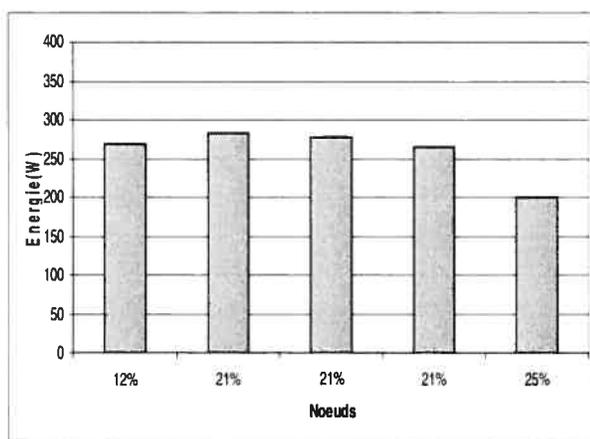


Figure 5.14.3. 3-ILHB

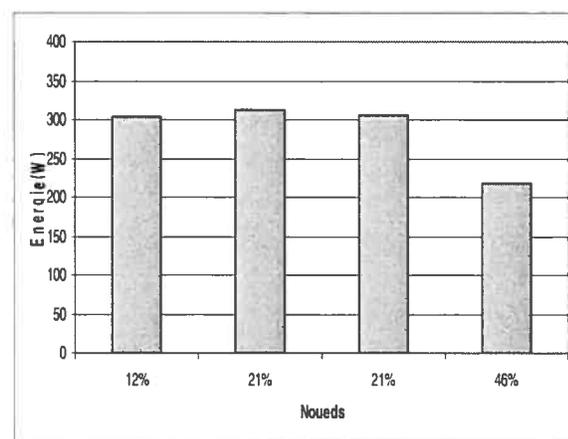


Figure 5.14.4. 4-ILHB

Les quatre figures ci-dessus illustrent la distribution de la consommation de l'énergie des nœuds pour un total de 100 nœuds dans le réseau pour les 4 heuristiques. La figure 5.14.1 montre une distribution où nous avons 48% des nœuds qui ont une consommation d'énergie comprise entre 250 mW et 260 mW et 32% des nœuds ont une consommation inférieure à 250 mW. Il y a seulement 20% des nœuds qui ont une consommation supérieure à 260 mW. La figure 5.14.2 montre que 21% des nœuds ont une consommation qui dépasse 300 mW alors que 46% des nœuds ont une consommation inférieure à 240 mW. Figure 5.14.3 montre que 75% des nœuds ont une consommation d'énergie comprise entre 200 mW et 260 mW alors que 25% des nœuds ont une consommation qui ne dépasse pas 200 mW. La figure 5.14.4 montre que 54% des nœuds ont une consommation d'énergie comprise entre 290 mW et 310 mW alors que 46% des nœuds ont une consommation qui ne dépasse pas 220 mW. En se basant sur ces résultats,

nous pouvons déjà conclure que 1-ILHB et 3-ILHB ont des performances meilleures que 2-ILHB et 4-ILHB.

Pour produire des résultats qui montrent la mort des nœuds avec le temps (et ainsi la durée de vie de réseau), nous augmentons la quantité de trafic dans le réseau, pour une longue période de temps, pour épuiser les batteries des nœuds (capacité maximale est égale à 500 mW). Plus précisément, nous augmentons le nombre de paquets envoyé par source à 700 paquets, le nombre de sessions à 10, et le temps de simulation à 60 minutes.

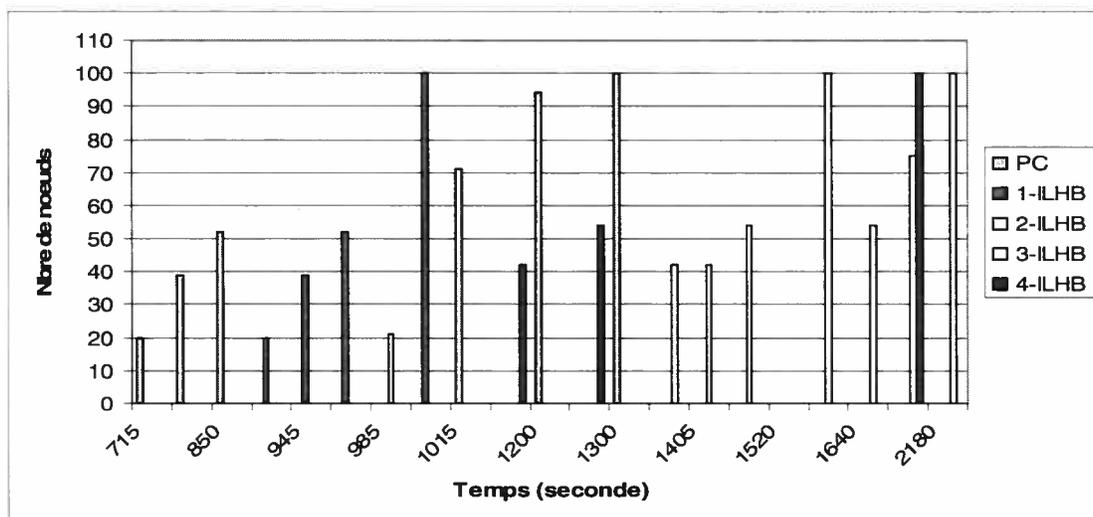


Figure 5.15. Mort de nœuds Vs. temps

La figure 5.15 montre que pour PC sans balancement de la charge nous enregistrons les premiers nœuds qui meurent, 20 nœuds exactement, à l'instant 715 secondes et 40 nœuds à l'instant 815 secondes. Figure 5.17 montre qu'à $t = 715$ s, le taux de livraison est de 85% seulement et devient 65% à $t = 815$ s. Le taux de livraison s'effondre jusqu'à 45% à $t = 850$ secondes. Nous avons un arrêt total du réseau à l'instant $t = 1300$ s.

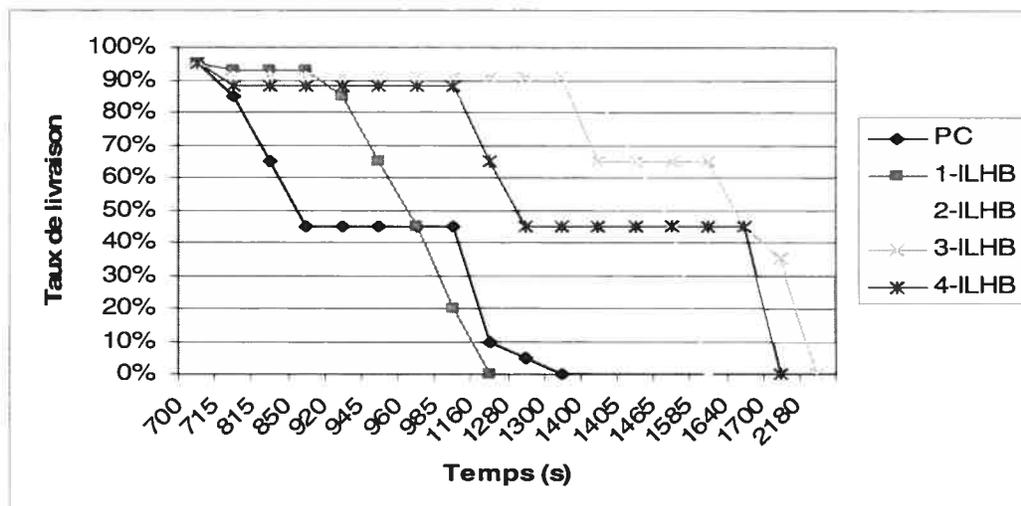


Figure 5.16. Taux de livraison Vs. temps

La figure 5.15 montre qu'avec 1-ILHB, les 20 premiers nœuds meurent à $t = 920$ s après quoi le taux de livraison devient 85% et s'effondre ensuite à 45% à $t = 985$ s (Figure 5.16). Nous avons un arrêt total du réseau à l'instant $t = 1000$ s. Avec 2-ILHB, les premiers nœuds meurent à $t = 1160$ s après quoi le taux de livraison devient 85% et change à 65% à l'instant 14605 s. Nous avons un arrêt total du réseau à l'instant $t = 1640$ s. Avec 3-ILHB, nous obtenons les premiers nœuds morts à l'instant $t = 1400$ s et le taux de livraison est 85 % et descend à 65 % à l'instant 1405 s. Nous avons un arrêt total du réseau à l'instant $t = 2280$ s. Finalement avec 4-ILHB, nous avons une baisse du taux de livraison jusqu'à 70% à $t = 1280$ s et une mort totale du réseau à $t = 2180$ s avec une stabilité du taux de livraison à 45% entre l'instant 1300 et l'instant 1700 s.

Les résultats de simulations montre que N-ILHB (N=1, 4) permet de prolonger la durée de vie du réseau et de garder un taux de livraison plus élevé pour une longue période par rapport à PC sans balancement. Nous pouvons aussi conclure que 3-ILHB offre les meilleures performances en termes de durée de vie de réseau et du taux de livraison.

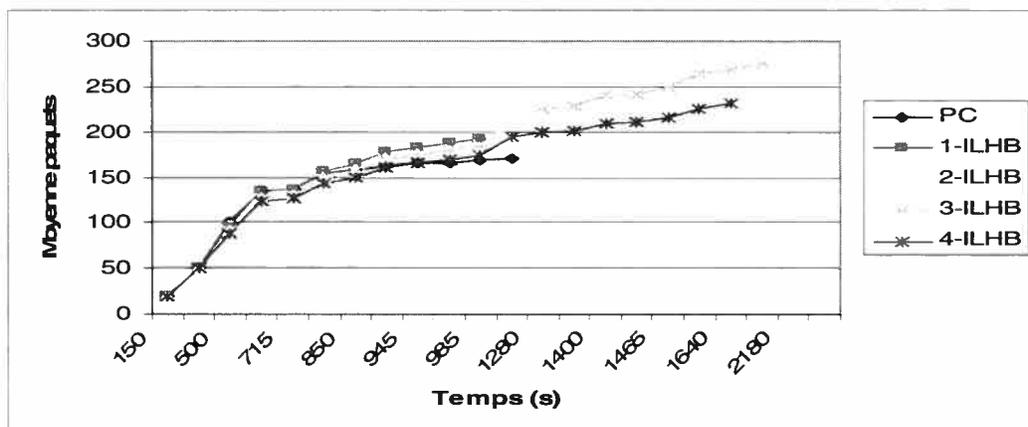


Figure 5.17. Nombre moyen de paquets reçus par destination Vs. Temps

La figure.5.17 montre la moyenne de paquets reçus (cumulés) pour toute la durée des simulations pour chaque nœud destination. Nous constatons que PC sans balancement de la charge est celui qui délivre le moins de paquets de données, 170 seulement par destination. Nous avons 191 paquets/destination reçus avec 1-ILHB, 247 paquets/destination reçus avec 2-ILHB, 275 paquets reçus/ destination avec 3-ILHB, et 232 paquets reçus/destination avec 4-ILHB. La encore, 3-ILHB fournit les meilleurs résultats.

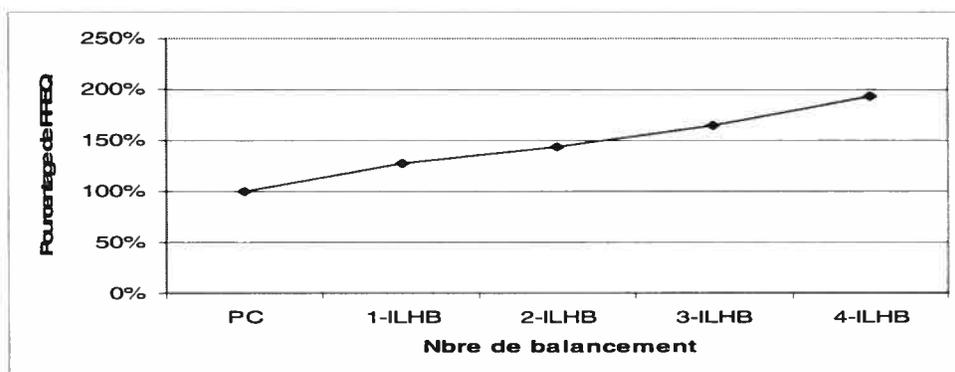


Figure 5.18. Paquets de contrôle Vs. Heuristique

La figure 5.18 illustre le pourcentage de paquets de contrôle pour PC avec et sans balancement de la charge. Nous constatons une augmentation presque linéaire du pourcentage de paquets de contrôle RREQ en fonction de N. Pour PC sans balancement de la charge, nous avons 1900 paquets de contrôle RREQ. Nous constatons une augmentation de 28% (2450 paquets RREQ) de plus avec 1-ILHB et de 94% (3700 paquets RREQ) de plus avec 4-ILHB. Ceci s'explique par le fait qu'à chaque seuil de

balancement (α de chaque intervalle considéré), de nouvelles reconstructions de chemins sont exécutées, ce qui génère des paquets RREQ.

Le tableau 5.3 récapitule les différences notables entre PC sans balancement et les 4 heuristiques. La deuxième colonne du tableau présente la consommation d'énergie totale pour 100 nœuds. La troisième colonne présente le pourcentage de paquets livré en plus par rapport à PC du début de la simulation jusqu'à la mort de tous les nœuds (Figure 5.18). La quatrième colonne montre le nombre de paquets de contrôle généré durant les simulations (Figure 5.18). En dernier, la cinquième colonne représente l'instant quand le(s) premier nœud meurt (Figure 5.16).

	Énergie consommée dans le réseau	Pourcentage de Paquets reçus en plus par rapport à PC	Nombre de paquets RREQ	Temps de la mort des premiers nœuds
PC sans balancement	24,4 w	-----	1900	715 s
1-ILHB sans reconfigurer La carte radio	26,9 w	+12%	2450	785 s
1-ILHB	25,5 w	+12%	2450	920 s
2-ILHB	26,8 w	+45%	2900	985 s
3-ILHB	27,9 w	+61%	3300	1400 s
4-ILHB	29,1 w	+36%	3700	1160 s

Tableau 5.3. Récapitulatif des résultats de comparaison

5.7 Conclusion

Nous avons proposé un ensemble d'heuristiques pour balancer la consommation d'énergie des nœuds dans un réseau qui utilise PC comme protocole de diffusion. L'idée de base derrière notre proposition est de forcer un chef de groupe ou nœud de liaison à changer son état à ordinaire quand sa consommation d'énergie atteint un certain seuil. Le nœud reste dans cet état, avec une diminution de son temps d'écoute et de réception (pour minimiser la consommation d'énergie due à l'écoute et la réception), jusqu'à ce

que sa consommation atteigne un deuxième seuil; à ce moment, le nœud pourrait participer à la diffusion en devenant chef de groupe ou nœud de liaison.

Les résultats de simulations montrent que les heuristiques proposés, surtout 3-ILHB, permettent d'augmenter la durée de vie du réseau et ainsi réaliser un taux de livraison supérieure à PC sans balancement de charge.

Le choix du paramètres N dépendra du temps de fonctionnement des nœuds, pour une durée courte choisir un N petit et pour de longue période de fonctionnement choisir un N grand.

Chapitre 6 Conclusion

L'objectif ultime de notre recherche est de réduire la consommation d'énergie dans les réseaux ad hoc mobile sans fil et de capteurs en utilisant le protocole de groupage passif et à réaliser un balancement de la charge pour augmenter la durée de vie du réseau. Une diminution globale de la consommation d'énergie ne garantit pas une augmentation de la durée de vie du réseau puisque certains nœuds meurent rapidement et génèrent des déconnexions ou des ruptures de chemin.

Le protocole de groupage passif cherche un chemin en pratiquant l'inondation de la source à la destination sans aucune information préalable sur l'état du réseau et sans avoir à utiliser de table de routage. Il apporte une amélioration considérable par rapport aux autres protocoles de groupage conventionnels, dits actifs, car il n'exige aucun protocole de signalisation pour former les groupes et choisir les chefs de groupe et nœuds de liaison. Donc il ne crée aucune surcharge (à l'exception de 2 bits par trame de données) pour opérer contrairement aux protocoles de groupage actifs qui génèrent une surcharge spécialement dans le cas d'une haute mobilité des nœuds.

Notre travail consiste, en premier lieu, à utiliser le protocole de groupage passif entre des sources et des destinations dont les coordonnées géographiques sont connues à l'avance, ou en partie, et de calculer les distances entre les nœuds intermédiaires et la destination à joindre. L'idée de base de ce protocole, appelé GPC, est de limiter l'inondation à une zone limitée du réseau entre la source et la destination (p.e. cercle de diamètre égal à la distance entre la source et la destination).

En deuxième lieu, nous proposons un ensemble d'heuristiques qui permettent d'équilibrer la consommation d'énergie des nœuds du réseau. L'idée de base derrière notre proposition est de forcer un chef de groupe ou un nœud de liaison à changer son état à ordinaire quand sa consommation d'énergie atteint un certain seuil. Le nœud reste dans cet état, avec une diminution de son temps d'écoute et de réception, jusqu'à

ce que sa consommation atteigne un deuxième seuil; à ce moment, le nœud peut participer à la diffusion en devenant chef de groupe ou nœud de liaison.

Les résultats des simulations que nous avons effectuées montrent que l'utilisation des coordonnées géographiques des nœuds en conjonction avec le protocole de groupage passif permet de réduire le nombre de chefs de groupes et nœuds de liaison impliqués dans la diffusion des données. Ceci permet de réduire considérablement la surcharge de réseau due à une diffusion des données à l'extérieur de la zone d'intérêt (entre source et destination) et ainsi la consommation d'énergie.

Nous avons pu montrer, via les simulations, que l'intégration des coordonnées géographiques avec PC n'altère en rien les performances de PC, comme par exemple le débit qui reste stable, mais au contraire, il apporte une grande efficacité dans la découverte des chemins. Le nombre de nœuds traversés diminue en utilisant la notion de distance la plus proche, et le taux de livraison est plus élevé et plus stable avec la mobilité des nœuds.

Une caractéristique intéressante de GPC est qu'il fonctionne même si on ne peut pas avoir les coordonnées géographiques de tous les nœuds du réseau. Les gains en termes d'énergie de GPC sont intéressants à partir de 40% des nœuds du réseau qui connaissent leurs coordonnées.

Les résultats de simulations montrent que les heuristiques proposés pour le balancement de la charge, surtout 3-ILHB, permettent d'augmenter la durée de vie du réseau et ainsi réaliser un taux de livraison supérieure à PC sans balancement de charge.

6.1 Perspectives

Comme travaux futurs, nous pensons que nous pouvons développer une nouvelle heuristique autre que FDW pour l'élection des chefs de groupe dans PC qui présenterait plus d'avantages et cela en utilisant des métriques comme l'énergie des nœuds.

Nous pourrions penser à développer une nouvelle heuristique (autre que ' $\alpha * NC + \beta > NG$ ' qui est utilisée actuellement) pour la sélection d'un nombre optimal de nœuds de liaison afin de réduire les collisions et optimiser la consommation d'énergie sans pour autant altérer la fiabilité.

Le défi majeur est de ne pas affecter les caractéristiques désirées de PC en développant de nouvelles heuristiques; plus particulièrement, il ne faut pas introduire de protocole de signalisation pour la sélection de chefs de groupes ou nœuds de liaison.

Bibliographie

- [1] S-Y. Ni, Y.C. Tseng, Y-S. Chen, J-P. Sheu, "The diffuse storm problem in a mobile ad hoc network," *Mobicom*, 1999.
- [2] Y.C. Tseng, S-Y. Ni, E-Y. Shih, "Adaptive Approaches to Relieving Diffuse Storms in a Wireless Multihop," *Mobile Ad Hoc Network, Infocom*, 2001.
- [3] A. Qayyum, L. Viennot, A. Laouiti, "Multipoint relaying: An efficient technique for flooding in mobile wireless networks," *INRIA report*, March 2000.
- [4] H. Lim, C. Kim, "Flooding in wireless ad hoc networks," *IEEE computer Communications*, 2000.
- [5] M. Bhardwaj, et. al, "Upper Bounds on the Lifetime of Sensor Networks," *In Proceeding of ICC*, 2001.
- [6] E. Pagani, G.P. Rossi, "Reliable diffuse in mobile multihop packet networks," *Mobicom*, 1997.
- [7] P. Krishna, N.H. Vaidya, M. Chatterjee, D.K. Pradhan, , "A clusterbased approach for routing in dynamic networks," *Computer Communication Review*, vol.27, (no.2), ACM, April 1997.
- [8] M. Cardei, D.Z. Du, "Improving Wireless Sensor Network Lifetime through Power Aware Organization," *ACM Wireless Networks*, Vol.11, 2005
- [9] C.R. Lin, M. Gerla, "Adaptive clustering for Mobile Wireless Networks," *IEEE Journal on Selected Areas in Communications*, Sep. 1997.
- [10] T.J. Kwon, M. Gerla. "Efficient flooding with Passive clustering in ad hoc networks," *ACM Sigcomm Computer Communication Review*, 2002.
- [11] M. Gerla, T.J. Kwon, G. Pei, "On Demand Routing in Large Ad Hoc Wireless Networks with Passive clustering," *Proceedings of IEEE WCNC 2000*, Chicago, IL, Sep. 2000.
- [12] D. L. Gu, G. Pei, M. Gerla, X. Hong, "Integrated Hierarchical Routing for Heterogeneous Multi-hop Networks," *In Proceeding of IEEE Milcom 2000*, CA, Oct. 2000.
- [13] Y-B. Ko, N.H. Vaidya, "Location-aided routing (LAR) in mobile ad hoc networks," in: *Proc. of Mobicom*, 1998.

- [14] Y.-B. Ko, N.H. Vaidya, "Using location information to improve routing in ad hoc networks," *Technical report 97-013, Texas A&M University*, 1997.
- [15] F. Nocetti, J.S. Gonzalez, "Connectivity Based k -Hop Clustering in Wireless Networks," *Telecommunication Systems* 22:1-4, 205-220, 2003.
- [16] Y.-B. Ko, N.H. Vaidya, "Location-based multicast in mobile ad hoc networks," *Technical report 98-018, Texas A&M University*, 1998.
- [17] A. Qayyum, A. Laouiti, L. Viennot, "Multipoint relaying technique for flooding diffuse messages in mobile wireless networks," *HICSS, Hawaii*, Jan 2002.
- [18] J.C. Navas and T. Imielinski, "Geocast - geographic addressing and routing," in: *Proc. of ACM/IEEE Mobicom*, 1997.
- [19] O. Younis, S. Fahmy, "Distributed clustering in ad-hoc sensor networks: A hybrid energy efficient approach," *Proceedings of IEEE Infocom*, Hong Kong, March, 2004.
- [20] G. Dommety, R. Jain, "Potential networking applications of global positioning systems (GPS)," *Technical report TR-24, The Ohio State University*, 1996.
- [21] Glomosim: Global mobile information Systems Simulation Library.
- [22] M. Gerla, L. Bajai, M. Takai, R. Ahuja, GloMoSim: "A Scalable Network Simulation Environment ". *Technical Report 990027, University of California*, 13, 1999.
- <http://pcl.cs.ucla.edu/projects/glomosim/>
- [23] R. Want et al, "The Active Badge Location System," *ACM Trans. Information Systems*, pp. 91-102. 4. A. Harter and A. Hopper, "A Distributed Location System for the Active Office," *IEEE Network*, Jan./Feb. 1994, pp. 62-70, Jan 1992.
- [24] R.G Ogier et al, "Topologie dissemination based on reverse path forwarding," *Network working group*, February 2004.
- [25] T. Clausen, P. Jacquet, "Optimized Link State Routing Protocol," RFC 3626, October 2003.
- [26] P. H. Hsiao, A. Hwang, H. T. Kung, D. Vlah, "Load-Balancing Routing for Wireless Access Networks," *IEEE Infocom*, April 2001.
- [27] S. Basagni, I. Chlamtac, B. A. Woodward, "A Distance Routing Effect Algorithm for Mobility," *Erik Jonsson School of Engineering and Computer Science*, 1998.

- [28] K. Xu, M. Gerla, "A heterogeneous routing protocol based on a new stable clustering scheme," *Computer Science Department, UCLA University of California, Los Angeles CA 90095*, MILCOM 2002. Proceedings, Oct 2002.
- [29] F. G. Nocetti, J. S. Gonzalez, I. Stojmenovic, "Connectivity based k-hop clustering in wireless networks," *Telecommunication Systems*, 1-4, 205- 220.22, 2003.
- [30] M. Chatterjee, S. K. Das, D. Turgut, "WCA: A weighted clustering algorithm for mobile ad hoc networks," *Cluster Computing* 5 - 193-204, April 2002,
- [31] C.E. Perkins, E.M. Royer, "Ad hoc on demand distance vector routing," (*Internet-draft*), in: *Mobile Ad-hoc Network (MANET) Working Group, IETF* 1998.
- [32] S. Ramanathan, M. Steenstrup, "A survey of routing techniques for mobile communication networks," *Mobile Networks and Applications*, 1986.
- [33] A. Wang, W. R. Heinzelman, A. P. Chandrakasan, "Energy-Scalable Protocols for Battery-Operated Microsensor Networks," *IEEE Workshop on Signal Processing Systems*, pp. 483–492, 1999.
- [34] M. Stemm R. H. Katz, "Measuring and Reducing Energy Consumption of Network Interfaces in Hand-held Devices," *IEICE Transactions on Communications*, vol. E80-B, no. 8, 1997.
- [35] V. Raghunathan, C. Schurgers, S. Park, M. B. Srivastava, "Energy-aware Wireless Microsensor Networks," *IEEE Signal Processing Magazine*, Mar. 2002.
- [36] Y. Sankarasubramaniam, I. F. Akyildiz, S. W. McLaughlin, "Energy Efficiency based Packet Size Optimization in Wireless Sensor Networks," *Proc. of the First IEEE International Workshop on Sensor Network Protocols and Applications*, Anchorage, Alaska, 2003.
- [37] W. Ye, J. Heidemann, D. Estrin, "An Energy-Efficient MAC Protocol for Wireless Sensor Networks," *Proc. IEEE Infocom*, 2002.
- [38] C. M.Teng, K. C Chen, "A Unified Algorithm for Wireless MAC protocols," *Vehicular Technology Conference. 2002. VTC Spring* 2002.
- [39] I. Chlamtac, M. Conti, J. Liu, "Mobile Ad Hoc Networking: imperatives and challenges," *Ad-hoc networks*, 1:13--64, 2003.
- [40] M. Raya, J. P. Hubaux, I. Aad, "DOMINO: A System to Detect Greedy Behavior in IEEE 802.11 Hotspots," *ACM MobiSys 2004, Boston, USA*, June 2004.

- [41] C. Perkins, E. Royer, "Ad Hoc On Demand Distance Vector Routing," *2nd International Workshop on Mobile Computing Systems and Applications*, Feb. 1999.
- [42] IEEE Std. 802-11. "IEEE Standard for Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specification," June 1997.
- [43] C. Intanagonwiwat, R. Govindan, D. Estrin, "Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks," *ACM/IEEE International Conference on Mobile Computing and Networks (Mobicom 2000)*, Boston, Massachusetts, August 2000.
- [44] B. Krishnamachari, D.Estrin, S.Wicke, "Modelling Data-Centric Routing in Wireless ensor Networks," IEEE Infocom 2002.
- [45] Y. Sankarasubramaniam, E. Cayirci, "A Survey on Sensor Networks," IEEE Communications Magazine, Août 2002.
- [46] Parsec user Manuel.
<http://pcl.cs.ucla.edu/projects/parsec/>
- [47] T. Haenselmann, "Lecture on Sensor Networks," University of Mannheim, Germany, 2005.
- [48] C. Yee, Chong, Srikanta, P.Kumar, "Sensor Networks: Evolution, Opportunities, and Challenges," 2003.
- [49] Fiche technique de la puce MICA, communiqué de l'entreprise Crossbow.
- [50] Fiche technique de la puce MICA2, communiqué de l'entreprise Crossbow.
- [51] Fiche technique de la puce MICA2DOT, communiqué de l'entreprise Crossbow.
<http://www.xbow.com/>
- [52] J. Kulik , W. Heinzelman , H. Balakrishnan, "Negotiation-based protocols for disseminating information in wireless sensor networks," *Wireless Networks*, March 2002.
- [53] B. KARP, "Greedy perimeter state routing," Invited Seminar at USC /information Science institute, July 1998.
- [54] X. Hong, K. Xu, M. Gerla. "Scalable Routing Protocols for Mobile Ad Hoc Networks," *Computer Science Department, UCLA University of California, Los Angeles CA 90095, IEEE Network*, July-Aug. 2002 .

- [55] D. Johnson, D. Maltz, "Dynamic Source Routing in Ad Hoc Wireless Networks," *Mobile Computing*, volume 5, pages 153-181, Kluwer Academic Publishers, 1996.
- [56] P. Jacquet, P. Mühlethaler, A. Qayyum, A. Laouiti, T. Clausen, L. Viennot, "Optimized Link State Routing Protocol," IEEE INMIC Pakistan, Dec 2001.
- [57] C.E. Perkins, P. Bhagwat, "Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers," *Comp. Comm. Rev.*, pp.234-244, Oct. 1994.
- [58] International Standards Organisation. "Intermediate system to intermediate system intra-domain routing exchange protocol for use in conjunction with the protocol for providing the connectionless- mode network service," (ISO 8473). ISO DP 10589, February 1990.
- [59] Xerox Corporation. "Internet transport protocols". *Xerox System Integration Standard*, " 028112, December 1981.
- [60] K. Al Agha, L. Viennot "The DARPA packet radio network protocols," *Proceedings of the IEEE*, 75(1): pp 21-32, January 1987.
- www.darpa.mil
- [61] D. Bertsekas, R. Gallager, "Data Networks," Prentice Hall Inc, pp 297-333, J. Broch, D. Johnson and D. Maltz, 1992
- [62] S. Basagni, " Finding a maximal weighted independent set in wireless networks," *Telecommunication Systems* 18, 1-3, 155-168, 2001
- [63] C. Y.Chong, S. P. Kumar, "Sensor Networks: Evolution, Opportunities, and Challenges," *Proceedings of the IEEE*, Vol. 91, No. 8, August 2003.
- [64] O. Younis, S. Fahmy, "Distributed clustering in ad-hoc sensor networks: A hybrid, energy efficient approach," *Proceedings of IEEE Infocom*, (Hong Kong, March, 2004), an extended version appeared in *IEEE Transactions on Mobile Computing*, 3, 4, Oct-Dec, 2004.
- [65] V. Kawadia, P. R. Kumar, "Power control and clustering in ad hoc networks," *Proceedings of IEEE Infocom*, April, 2003.
- [66] D. M. Blough, P. Santi, " Investigating upper bounds on network lifetime extension for cellbased energy conservation techniques in stationary ad hoc networks," *Proceedings of ACM Mobicom*, 2002.

- [67] Y. Xu, J. Heidemann, D. Estrin, "Geography-informed energy conservation for ad hoc routing," *Proceedings of ACM Mobicom*, Rome, Italy 70–84, July, 2001.
- [68] F. Ye, G. Zhong, S. Lu, L. Zhang, "PEAS: A robust energy conserving protocol for longlived sensor networks," *Proceedings of IEEE International Conference on Distributed Computing Systems (ICDCS)*, 2003.
- [69] W. Agassounon, A. Martinoli, "Efficiency and Robustness of Threshold-Based Distributed Allocation Algorithms in Multi-Agent Systems," *Proc. of the First Int. Joint Conf. on Autonomous Agents and Multi-Agent Systems AAMAS-02 (Bologna, 2002)*, ACM Press, pp. 1090–1097, 2000.
- [70] W. Agassounon, A. Martinoli, K. Easton, "Macroscopic Modeling of aggregation Experiments Using Embodied Agents in Teams of Constant and Timevarying Sizes," *Autonomous Robots, Special Issue on Swarm Robotics 17*, 163–192, 2004.
- [71] E. Bonabeau, M. Dorigo, G. Theraulaz, "Swarm Intelligence From Natural to Artificial Systems," *Oxford University Press*, New York, NY, 1999.
- [72] T. Yan , Y. Bi , L. Sun , H. Zhu ,L. Xicheng, Z. Wei, "Probability based dynamic load-balancing tree algorithm for wireless sensor networks," *Institute of Software, Chinese Academy of Sciences, Beijing 100080, CHINE*, 2005.
- [73] W. Heinzelman, A. Chandrakasan, H. Balakrishnan, "Energy-Efficient communication Protocol for Wireless Microsensor Networks," *Proc. of IEEE Proc. Of the Hawaii International Conf. on System Sciences*, pp. 1-10, Jan 2000.
- [74] N. Bulusu, J. Heidemann, D. Estrin, "GPS-less low cost outdoor localization for very small devices ," *IEEE Personal Communications*, vol. 7, no 5 p. 28–34, 2000.
- [75] S. Capkun, M. Hamdi, J-P. Hubaux, "GPS-free positioning in mobile ad hoc networks," *Proceedings of the 34th HICSS*, janvier 2001.
- [76] L. Doherty et al, "Convex Position Estimation in Wireless Sensor Networks," *Proc. Infocom 2001*, IEEE CS Press, Los Alamitos, Calif, 2001.