

MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITÉ MOULOUD MAMMARI DE TIZI OUZOU

FACULTÉ DE GÉNIE ÉLECTRIQUE ET D'INFORMATIQUE

DÉPARTEMENT D'INFORMATIQUE

THÈSE DE DOCTORAT

SPÉCIALITÉ : INFORMATIQUE

Présentée par

Rachida AOUDJIT (épouse LAGAB)

Sujet

Répartition et Equilibrage de Charges dans les hôtes mobiles

Devant le jury d'examen composé de :

P ^r . D. HAMADOUCHE	Professeur	UMMTO	Président
P ^r . M. LALAM	Professeur	UMMTO	Rapporteur
P ^r . A. M'ZOUGH	Professeur	IRIT, Toulouse	Examineur
P ^r . M. BENMOUHAMMED	Professeur	U. Constantine	Examineur
P ^r . M. KOUDIL	Professeur	ENSI, OuedSmar	Examineur
D ^r . R. AHMED-OUAMER	M. C.	UMMTO	Examineur

Soutenu le

Remerciements

Je tiens à témoigner ma sincère gratitude à mon directeur de thèse, M^r. Lalam Mustapha, Professeur au Département d'Informatique de l'UMMTO, et ce pour de multiples raisons. Tout d'abord, je lui suis reconnaissante d'avoir dirigé mes travaux de recherche tout en me laissant libre d'explorer des pistes à ma guise. De plus, je le remercie pour ses conseils et avis tout au long de cette thèse ainsi que pour avoir été présent et motivant.

Mes sincères remerciements vont également à M^r. Djamel HAMADOUCHE , Professeur au département de Mathématique de l'UMMTO pour avoir accepté de présider le jury de soutenance.

J'exprime mes profonds remerciements pour M^r. Abdelaziz M'ZOUGHJI, Professeur à l'Institut de Recherche en Informatique de Toulouse (IRIT), pour m'avoir accueilli à plusieurs reprises dans son laboratoire, pour son hospitalité, pour sa grande contribution et ses orientations et pour avoir accepté de participer au jury d'examen.

Mes vifs remerciements vont également à M^r. Mohamed BENMOHAMED , professeur à l'Université de Constantine et à M^r. Mouloud KOUDIL , professeur à l'ENSI pour avoir accepté de participer au jury d'examen.

Je remercie également M^r. Rachid AHMED-OUAMEUR , directeur du laboratoire de recherche en informatique de Tizi-ouzou (LARI) pour avoir mis à ma disposition, les moyens du laboratoire ainsi que pour avoir accepté de participer au jury d'examen.

A ma très chère **LINA**

Résumé

L'évolution rapide des capacités de traitement des nœuds mobiles et l'amélioration de la qualité du réseau reliant ces entités ont motivé l'utilisation du réseau ad hoc comme support pour les calculs distribués. Une bonne utilisation de cette plate-forme permet ainsi d'augmenter la puissance potentielle des nœuds. La recherche d'un maximum de puissance nécessite une répartition et un équilibrage de charges efficace. Ceci est d'autant plus vrai lorsque l'on se trouve dans un contexte hétérogène. En effet, les ordinateurs mobiles possèdent des ressources intrinsèques (puissance de calcul, autonomie de la batterie) de capacités très variables, pouvant varier de ressources comparables à celles des PC, notamment dans le cas des ordinateurs portables, jusqu'à des ressources nettement inférieures, notamment dans le cas des ordinateurs de poche. Cette hétérogénéité de capacité de traitement peut engendrer des situations où des nœuds sont fortement chargés, tandis que d'autres restent faiblement chargés ou inutilisés. Une bonne utilisation de chaque nœud permet de profiter de sa puissance sans affecter ses propres travaux et d'augmenter potentiellement la puissance des nœuds mobiles.

Dans ce travail, nous proposons une nouvelle approche d'équilibrage de charges où les nœuds les moins chargés " aident " les nœuds trop chargés. La stratégie proposée est basée sur les algorithmes de clustering où la prise de décision s'effectue en fonction de la charge moyenne des clusters, ce qui permet de réduire les échanges d'information.

L'objectif du travail présenté est d'assurer la plus grande longévité possible d'un réseau ad hoc et de réduire les temps d'exécution des applications.

Chaque nœud étant en charge d'un certain nombre de tâches, réduire les temps d'exécution et la consommation énergétique d'un nœud revient donc à partager sa charge de travail avec ses voisins les moins chargés. L'idée est de faire en sorte que la charge à transférer de chaque nœud soit proportionnelle à la puissance de traitement et le niveau d'énergie restant du nœud acceptant cette charge.

Mots clés : Réseaux ad hoc, Clustering, Equilibrage de charges, Conservation d'énergie, Collaboration dans les réseaux ad hoc.

Table des matières

Résumé	i
Table des matières	ii
Liste des figures	v
Liste des tables	vi
Liste des acronymes	vii
Introduction générale	1
Objectifs de la thèse	3
Domaines d'application de l'équilibrage de charges	4
Organisation de la thèse	6
1 La collaboration dans les réseaux sans fil	7
Introduction	7
1.1 La collaboration	8
1.2 Les formes de collaboration	10
1.3 Les environnements de collaboration actuels	10
Conclusion	14
2 Conservation d'énergie dans les réseaux ad hoc	15
Introduction	15
2.1 Approches de conservation d'énergie au niveau physique	16
2.2 Approches de conservation d'énergie au niveau liaison de données	18
2.3 Approches de conservation d'énergie au niveau routage	20
Conclusion	24

3 Répartition et Equilibrage de Charges	25
Introduction	25
3.1 Equilibrage de charges dans les réseaux filaires	26
3.1.1 Classification des stratégies d'équilibrage de charges	28
3.1.1.1 Politique d'information	28
3.1.1.2 Politique de redistribution	30
3.1.2 Structure générale d'un mécanisme d'équilibrage de charges	30
3.1.2.1 Gestionnaire de l'état de charge du système	31
3.1.2.2 Élément de contrôle et de prise de décision	31
3.1.2.3 Mécanisme de transfert de la charge	31
3.2 Equilibrage de charges dans les réseaux ad hoc	32
3.2.1 Equilibrage de charges au niveau communication :	32
3.2.2 Equilibrage de charges au niveau traitement (calcul)	39
Conclusion	40
4 Le Clustering	42
Introduction	42
4.1 Définition du clustering	43
4.2 Classification des méthodes de clustering	44
4.2.0.1 Clusters à 1 saut	44
4.2.0.2 Clusters à K sauts	49
4.2.0.3 Clusters hiérarchiques	53
Conclusion	55
5 Présentation de la solution : Equilibrage de charges basé sur le clustering	56
Introduction	56
5.1 Phase de clustering	57
5.1.1 La métrique utilisée	57
5.1.2 Construction des clusters	58
5.1.3 Maintenance des clusters	60
5.2 Phase d'Equilibrage de charges	60
5.2.1 Modèle de représentation proposée	61
5.2.2 Stratégie d'équilibrage de charges	62
5.2.2.1 Principe	62
5.2.2.2 Description générique de la stratégie d'équilibrage	63

5.3 Algorithmes d'équilibrage	66
Conclusion	68
6 Simulation et évaluation	69
Introduction	69
6.1 Description du simulateur	70
6.2 Evaluation de l'algorithme de clustering	72
6.2.1 Nombre moyen de clusterheads	73
6.2.2 Impact de la mobilité sur la stabilité du réseau	74
6.3 Evaluation de l'algorithme d'équilibrage de charges	75
6.3.1 Amélioration du temps d'exécution	75
6.3.2 Pourcentage des travaux interrompus	76
6.3.3 Impact de la mobilité sur l'énergie	77
6.3.4 Facteur d'équilibre	78
6.3.5 Valeur du seuil d'équilibrage de charges	79
6.3.6 Coût d'équilibrage de charges	80
Conclusion	81
Conclusion et perspectives	82
Bibliographie	85

Table des figures

5.1	Le Modèle d'équilibrage proposé	61
6.1	Nombres de clusterheads en fonction du nombre de noeuds (petites topologies)	73
6.2	Nombres de clusterheads en fonction du nombre de noeuds (grandes topologies)	74
6.3	Durée de vie des clusterheads en fonction de la vitesse	75
6.4	Nombre moyen de clusterheads en fonction de la vitesse	75
6.5	Amélioration du temps d'exécution en fonction du nombre de noeuds . . .	76
6.6	Pourcentage des travaux interrompus	77
6.7	Impact de la mobilité sur l'énergie	78
6.8	Facteur d'équilibre après équilibrage de charge	79
6.9	Temps d'exécution pour différentes valeurs du seuil d'équilibrage	80
6.10	Coût d'équilibrage en fonction du nombre de noeuds	81

Liste des tableaux

Liste des acronymes

AIM	América On-Line Instant Messaging
API	Application Programming Interface
AM	:Active mode
ATIM	Announcement Traffic Indication Message
AODV	Ad Hoc On-Demand Distance Vector Routing
AODV-MP	(Ad Hoc On-Demand Distance Vector Routing with MultiPath
AZRP	Adaptive Zone Routing Protocol
BW	BandWidth
CSMA	Carrier Sense Multiple Access
CAO	Conception Assistée par Ordinateur
CMMBCR	Conditional Min-Max Battery Capacity Routing
COPE	Common-case Optimization with Penalty Envelope
CP	Congestion Parameter
CBRP	Cluster Based Routing Protocol
CID	Clusterhead Identifier
DARPA	Defense Advanced Research Projects Agency
DOE	Department Of Energy
DSR	Dynamic Source Routing
DDR	Distributed Dynamic Routing
DN	Designated Node
EDDSR	Energy Dependent DSR
ECGRID	Energy Conserving GRID
FTP	File Transfer Protocol
FDAR	Free-Degree Adaptive Routig
GPS	Global Positionning System
HACO	Hybrid Ant Colony Optimization
HCC	High Connectivity Clustering
IRC	Internet Relay Chat
IMPA	Instant Messaging and Presence Awareness
ISR	Interrupt Service Routine
IEEE	Institute of Electrical and Electronics Engineers

LEAR	Local Energy-Aware Routing
LPRaodv	Lifetime Prediction Routing basé sur AODV
LBAR	Load Balanced Ad hoc Routing protocol
LBW	Load Balancing Weight
LCA	Lowest-Identifier Cluster Algorithm
LCC	Least Cluster Change
MPTR	Minimum Total Transmission Power Routing
MBCR	Minimum Battery Cost Routing
MMBCR	Min Max Battery Cost Routing
MDR	Minimum Drain Rapport
MANET	Mobile Adhoc Network
MATE	MPLS Adaptative Traffic Engineering
MOBIC	Mobility Based Clustering
MCDS	Minimum Cardinality Dominating Set
MBone	Multicast Backbone
OSPFOMP	Open Shortest Path First- Optimized Multipath
ORION	Optimised Routing Independent Overlay Network
PSM	Power Saving Mode
PARAODV	Power-Aware Routing basé sur AODV
RTT	Round Trip Time
SURAN	Survivable Radio Networks
SLAR	Simple Load balancing Ad hoc Routing
TAPSM	Traffic Aware Power Saving Mode
TeXCP	Traffic engineering with an eXplicit Control Protocol
VRVS	Virtual Room Video conferencing Sytem
Vat	Visual Audio Tool
Vic	Video Conference
WLAN	Wireless Local Area Network
WMN	Wireless Mesh Networks
WCA	Weighted Clustering Algorithm
XCP	eXplicit Control Protocol
ZRP	Zone Routing Protocol

Introduction générale

De nos jours, les gens se déplacent et communiquent de plus en plus. Ils ont désormais besoin de nouvelles technologies leur permettant de façon simple et rapide de récupérer diverses informations et de communiquer avec des personnes distantes pouvant être n'importe où dans le monde. Ces dernières années ont vu le développement technologique de nombreux composants et d'appareils électroniques de toute sorte pour répondre à ces nouveaux besoins. Ces appareils communicants sont de plus en plus petits, ont des capacités de calcul de plus en plus performantes et sont de plus en plus répandus. On les rencontre partout dans notre quotidien : à la maison, au bureau, dans les voitures, etc.

L'avancement des recherches dans le domaine des réseaux mobiles et des technologies de communication a favorisé la conception de nouvelles applications (l'audio et la vidéo conférence, la télévision à la demande, les jeux de groupes interactifs, le télé-enseignement. . .etc) et la formation de groupes collaboratifs. Plusieurs de ces applications telles que les communications de groupes ou la coordination des forces civiles se déploient sur des réseaux ad hoc et requièrent des échanges efficaces d'informations afin d'accroître le savoir collectif. Les modes les plus rudimentaires de collaboration se limitent à des échanges de simples messages textuels (forums, listes de diffusion). Cependant, les utilisateurs sont de plus en plus demandeurs de solutions qui leur permettent de collaborer dans des situations de mobilité avec d'autres utilisateurs afin d'offrir à leurs applications des fonctionnalités qui leur permettent de s'exécuter plus rapidement et oeuvrer pour réduire leur consommation d'énergie. Ces nouveaux besoins engendrent de nouveaux défis à relever pour assurer des communications de groupe dans un réseau ad hoc, offrant la qualité de service pour de telles applications (temps de réponse réduit), tout en s'adaptant à la nature des réseaux

ad hoc et à leurs caractéristiques (lien sans fil, mobilité et dynamique des nœuds, énergie limitée. . .).

Cependant, ces environnements mobiles présentent une hétérogénéité importante et une grande variabilité aussi bien au niveau de la capacité de traitement que de l'énergie entre les différents nœuds. Dans de tels environnements, des déséquilibres de charges peuvent apparaître. En effet, un nœud plus puissant en terme de capacité de traitement peut devenir oisif, parce qu'il a rapidement fini son travail tandis que les autres, moins puissants, sont occupés pendant la majeure partie du temps et consomment ainsi plus d'énergie. La capacité des nœuds puissants peut être exploitée par des nœuds surchargés si une fraction de la charge de ces derniers est partagée avec eux. Le recours à une solution de répartition de charges (ou load balancing) s'avère donc crucial pour améliorer les performances d'un réseau ad hoc et satisfaire les services et les applications des utilisateurs mobiles.

Par ailleurs, malgré les progrès technologiques réalisés, l'énergie demeure encore une ressource critique et une autre problématique majeure dans un environnement ad hoc, étant donné que les hôtes mobiles sont munis de batteries autonomes dont la capacité est limitée. La consommation d'énergie dans les hôtes mobiles relève un défi pour la durée de vie du réseau, du moment que l'épuisement de l'énergie au niveau de certains nœuds peut entraîner le partitionnement du réseau et la perte de connexion en conséquence. Ainsi, pour qu'un réseau ad hoc ait une longévité maximale et afin de maintenir le réseau fonctionnel aussi longtemps que possible, les protocoles conçus pour les réseaux ad hoc doivent judicieusement utiliser les ressources énergétiques limitées.

Tous ces défis doivent être relevés en tenant compte des contraintes intrinsèques des entités technologiques et des technologies de communication.

Nous présentons dans ce travail un nouvel algorithme d'équilibrage de charges où l'idée de base est de répartir la charge entre les nœuds en évitant l'inactivité des nœuds alors que des tâches restent en attente sur d'autres nœuds. Nous nous intéressons à deux charges importantes : la *charge de calcul*, liée aux services et aux applications supportés par chaque nœud, et la *charge de communication*.

Une stratégie d'équilibrage de charges est considérée comme efficace si elle engendre un

faible surcoût de communication en raison de l'énergie et de la largeur de bande passante limitées dans les réseaux ad hoc. En regroupant des entités proches en clusters et en utilisant un algorithme d'équilibrage centralisé au sein des clusters et distribué entre les clusters, une telle approche permet à chaque nœud de stocker la totalité des informations de son cluster et seulement une partie des informations concernant les autres clusters.

Notre contribution comporte des mécanismes d'équilibrage de charges dans un réseau ad hoc dont l'idée s'inspire des algorithmes de clustering où un sous-ensemble de nœuds 'clusterheads' est élu pour maintenir un certain équilibre dans leurs clusters respectifs tout en réduisant le coût global des communications.

L'objectif de la solution proposée est de :

- o *Tirer parti de l'hétérogénéité* : les nœuds mobiles présentent des capacités disparates, une solution doit donc être construite, permettant de faire participer les nœuds "forts" (en terme d'énergie et de capacité de traitement). Nous choisissons donc de répartir les charges, pour tenir compte des capacités différentes de certains nœuds.
- o *Économiser l'énergie* : des nœuds faibles en énergie peuvent transférer leurs charges aux autres nœuds et économiser leur énergie, ce qui allonge leur durée de vie et garantit la stabilité du réseau ad hoc.
- o *Optimiser la diffusion d'information* : seuls certains nœuds 'cluster head' sont habilités à détenir l'information sur l'état de charge afin qu'elle soit exploitée par tout le réseau. Le nombre de transmissions est donc réduit, et la performance est accrue.

Objectifs de la thèse et motivation

Les futurs environnements de communication sans fil offriront beaucoup de possibilités pour de nouveaux services et de nouvelles manières pour la collaboration et l'échange d'information. Les utilisateurs ne seront plus satisfaits simplement d'être reliés et joignables à toute heure mais ils exigent des services qui les aident dans leur vie quotidienne. Notre motivation pour ce travail peut être exprimée comme suit : " Là où il y a un groupe de personnes, elles partageront très probablement des intérêts communs ". Ces intérêts peuvent être professionnels (exemple. les membres d'un projet de recherche) ou privés. Ces

groupes de personnes forment des communautés qui partagent certaines ressources. Par ailleurs, plusieurs architectures qui offrent une interaction et une collaboration dans un réseau ad hoc ont été développées [1, 2, 3]. En revanche, les systèmes existants ne se sont pas consacrés à la gestion et au partage de la puissance de calcul des nœuds mobiles en se souciant de la disponibilité de l'énergie qui est une ressource critique.

Au regard de ces limitations et des opportunités offertes par les réseaux sans fil, en particulier par le mode ad hoc de ces réseaux, il est devenu nécessaire de développer de nouveaux mécanismes pour assurer la gestion des ressources (l'énergie et la puissance de calcul). Dans ce cas, l'équilibrage de charges devient une solution clé pour exploiter au mieux la capacité des nœuds les plus puissants dans les réseaux ad hoc.

Domaines d'application de l'équilibrage de charges

Les premières applications de l'équilibrage de charges concernaient les systèmes parallèles ou la programmation parallèle [4, 5]. Cependant, avec l'avancement des recherches dans le domaine des réseaux et l'émergence des technologies sans fil d'autres applications sont apparues [6]. On distingue :

o *Le jeu vidéo* : le marché du jeu vidéo a connu ces dernières années une expansion des plus importantes sur le marché mondial. Le jeu vidéo voit autant de nouvelles opportunités s'ouvrir à lui, via le jeu dit "On Line". Aujourd'hui, la quasi-totalité des jeux vidéo multi joueurs intègre la possibilité de jouer sur Internet. L'association entre jeux et réseaux est pour ainsi dire solidement ancrée dans les usages. Cette introduction du réseau dans le monde des jeux vidéo a radicalement changé la donne de cette industrie. Il a permis l'émergence d'une nouvelle génération de jeux où les interactions entre joueurs constituent l'intérêt principal du jeu. On peut raisonnablement s'attendre à ce que la portabilité des jeux vidéo sur les réseaux sans fil constitue une évolution naturelle de ce domaine. L'affranchissement des liaisons câblées et la mobilité offerte aux joueurs sont autant d'arguments qui vont dans le sens d'une interaction plus aisée et plus riche entre joueurs. Le défi principal étant l'optimisation des ressources (puissance et consommation d'énergie) par une stratégie d'équilibrage de charges.

o Le travail scientifique collaboratif : les équipes de collaboration scientifique géographiquement dispersées constituent une réalité depuis plusieurs années. Ces scientifiques doivent pouvoir participer aux activités de collaboration en tous endroits (à la maison, à l'aéroport ...etc). Les activités de calcul, les opérations d'analyse de données et de débogage forment le cœur de leurs activités de collaboration. Des outils qui supportent la connectivité permanente jours et nuits et les besoins fondamentaux d'un groupe de collaborateurs ont été développés. Ceci est important mais il y a un besoin plus fondamental : employer la puissance de calcul combinée avec quelques dispositifs portables peut conduire d'une façon significative à une meilleure productivité du groupe.

o La robotique : Les réseaux de robots sont un domaine en plein essor. Les recherches sur ce sujet proviennent de la prise de conscience qu'il est possible de construire des dispositifs intégrant à la fois de la mobilité et des unités de calcul autonomes. Il est intéressant de remarquer que les réseaux de robots ont de multiples applications : la surveillance et l'exploration de l'environnement ou bien la réalisation de tâches collaboratives. Dans ce domaine de la robotique, des systèmes de contrôle distribué d'un nombre de robots mobiles ont été développés pour coordonner leurs tâches. La performance de ces systèmes peut être améliorée en équilibrant la charge de traitement des différents robots.

o L'environnement : La surveillance de l'environnement est une problématique de plus en plus importante dans le cadre du développement durable et de la prévention des risques. On voit se mettre en place de plus en plus de systèmes d'information environnementaux centralisant les données spatio-temporelles pertinentes sur l'état des milieux et sur les actions de l'homme sur ceux-ci. Les technologies actuelles permettent de multiplier les sources d'acquisition de données spatio-temporelles qui peuvent être fixes (des capteurs d'humidité dans un champ par exemple) ou mobiles (comme des microcapteurs de pollution de l'air embarqués dans un véhicule). Ces sources peuvent stocker leurs données pour un transfert différé ou transmettre directement leurs informations via des réseaux sans fil. Elles peuvent être également pourvues de capacité de calcul afin de conditionner leurs données pour optimiser les temps de transferts, gérer au mieux la consommation énergétique, assurer leur rôle de relais au sein du réseau. Les capteurs deviennent ainsi de véritables Objets Intelligents

Communicants (OIC). Ces OIC peuvent contribuer collectivement à l’accomplissement de tâches de plus haut niveau. Il est donc nécessaire d’avoir un outil permettant de partager la puissance de calcul et l’énergie afin de garantir la fiabilité du système et d’améliorer la performance de ces outils.

Organisation de la thèse

Cette thèse est organisée en six chapitres. Dans les chapitres 1 et 2, nous analysons, sous forme d’un état de l’art, plusieurs problématiques inhérentes aux réseaux sans fil. Nous nous intéressons plus particulièrement et respectivement au travail collaboratif et à la gestion d’énergie dans les réseaux ad hoc.

Le chapitre 3 introduit les problématiques liées à l’équilibrage de charge. Nous y présentons les principales catégories de protocoles proposées dans la littérature pour les réseaux filaires. Les méthodes d’équilibrage de charge dans les réseaux sans fil sont aussi analysées en détail. Notre intérêt se porte tout particulièrement sur le paradigme d’équilibrage au niveau traitement.

Dans le chapitre 4, nous définissons précisément ce qu’est pour nous une structure d’auto-organisation. Nous en donnons les principes fondamentaux, les propriétés que doivent suivre de telles structures. Le clustering est la structure d’auto-organisation servant de base pour notre travail, nous présentons un panorama des différentes solutions proposées dans les réseaux ad hoc.

Le chapitre 5 introduit notre solution d’équilibrage de charges, qui prend en considération des spécificités et des caractéristiques des nœuds mobiles. Ce dernier se déroule en deux étapes qui se complètent : construction d’une structure hiérarchique du réseau mobile puis équilibrage de charges.

Le chapitre 6 présente l’évaluation de notre solution. Plusieurs séries de résultats obtenues par simulation sont présentées, soulignant l’importance de l’équilibrage de charges dans les réseaux ad hoc et les améliorations apportées par notre contribution.

Chapitre 1

La collaboration dans les réseaux sans fil

Introduction

Depuis que la téléphonie mobile s'est imposée dans notre vie quotidienne et que les réseaux sans fil ont connu un grand succès, de nouveaux horizons vers des applications utilisant ces technologies sans fil sont apparus [7, 8]. Bien qu'initialement prévus pour des services sans aucune garantie, les réseaux mobiles actuels ont tendance à acheminer des flux de toutes sortes d'applications : FTP, mail, téléphonie, visiophonie, web, etc. Ainsi, poussée par plusieurs facteurs économiques, la mobilité est devenue de plus en plus un élément fédérateur des réseaux de communication [9].

A l'origine, ce sont les militaires qui se sont intéressés les premiers aux réseaux de communication sans fil fonctionnant de proche en proche tout en restant fonctionnels en cas de mobilité ou de perte d'éléments de routage. Tout a commencé au début des années 70, lorsque les techniques de commutation de paquets ont poussé l'Agence des Projets de Recherche du Département de la Défense américaine DARPA (Defense Advanced Research Projects Agency) à développer une nouvelle génération de réseaux appelée PRNet (Packet Radio Network) [10]. A l'époque, cette nouvelle génération de réseaux disposait déjà d'une architecture distribuée, permettant un partage dynamique du canal de diffusion par une combinaison des protocoles CSMA et Aloha.

Cependant, en milieu des années 80, le DARPA a développé une nouvelle génération

de réseaux appelée SURAN (Survivable Radio Networks) [11], dont l'objectif était d'éviter certaines lacunes du projet PRNet et de permettre, entre autre, le passage vers des réseaux comportant jusqu'à une dizaine de milliers de noeuds mobiles supportant des protocoles évolués, avec des mécanismes radio simples, où la consommation d'énergie et le coût restent faibles. En revanche, ce projet est resté exclusivement militaire, jusqu'à son passage vers la fin des années 90 pour être utilisé dans des applications civiles, où des recherches ont permis d'étudier la possibilité de passer vers des réseaux mobiles totalement dynamiques et spontanés. Ceci a donné naissance aux réseaux mobiles ad hoc. Ces recherches sur les réseaux ad hoc dans le domaine civil ont en fait, pris leur essor avec l'arrivée des premières technologies radio, principalement la norme IEEE 802.11 et ses différentes dérivées. Cette norme a été standardisée en 1999 par l'IEEE (Institute of Electrical and Electronics Engineers), dans le but d'assurer la communication entre ordinateurs utilisant le médium radio. Cependant l'utilisation de ces nouveaux environnements introduit de nouvelles problématiques et crée de nouveaux besoins.

Malgré l'émergence de nombreuses applications dans les réseaux mobiles ad hoc telles que la visioconférence, les jeux distribués . . .etc, des problèmes ouverts suscitent beaucoup d'intérêts de la part des chercheurs. Les études courantes consistent principalement à proposer plusieurs solutions dans le cadre du travail collaboratif. Nous nous intéressons dans ce chapitre, en particulier, aux solutions permettant le travail collaboratif dans un environnement ad hoc, où les participants peuvent joindre et quitter dynamiquement le réseau . Nous allons dans un premier temps présenter les caractéristiques et les contraintes liées aux environnements collaboratifs dans les réseaux ad hoc, nous passerons ensuite en revue les principales solutions proposées dans la littérature.

1.1 La collaboration

A l'origine, des applications exploitant les réseaux ad hoc ont été envisagées principalement pour des situations de crise (par exemple, dans les champs de bataille ou pour des opérations de secours). Cependant, les technologies sans fil se sont sensiblement améliorées

ces dernières années et des dispositifs peu coûteux ont envahi le marché et le déploiement des réseaux ad hoc pour des applications (commerciales, scientifiques. . .) est devenu réalité. La combinaison des modèles de communication décentralisés et des réseaux mobiles sans fil offre un potentiel important pour la collaboration entre utilisateurs mobiles en tout lieu et à tout instant et crée des opportunités pour de nouvelles formes de collaboration mobile impliquant l'interaction entre des personnes géographiquement éloignées et organisées d'une manière spontanée. En effet, de plus en plus d'utilisateurs sont demandeurs de solutions qui leur permettent de collaborer dans des situations de mobilité avec d'autres utilisateurs [12, 13].

Grâce aux outils de travail collaboratif, la solution de communication apporte une organisation des groupes de travail itinérants. Une solution efficace permet d'accomplir des collaborations utilisant n'importe quel média, de n'importe où et à tout moment. De ce fait, de nombreux travaux ont porté sur la conception et la réalisation d'environnements de collaboration adaptés aux environnements mobiles pour tenter d'offrir aux utilisateurs et à leurs applications des fonctionnalités qui leur permettent d'évoluer dans ces environnements.

Par ailleurs, l'entreprise d'aujourd'hui souhaite accroître son dynamisme économique en favorisant la mise en relation du commerce inter-employés ou avec ses partenaires externes à l'entreprise. Cette poussée économique demande que chacun soit capable de communiquer en temps réel, simplement, réactivement ou proactivement en vue de participer à des activités quelque soit sa situation géographique. Recherchant à prospérer dans le monde d'aujourd'hui, beaucoup d'entreprises ont commencé l'acquisition de solutions de communication avec l'objectif de procurer des outils de travail sur sites distribués aussi efficacement que face à face [6].

Des efforts de recherche ont constaté que l'utilisation de ces outils contribue à développer le sens de communauté de collègue souvent absent dans un grand lieu de travail [14]. Ces outils sont également vus comme un moyen allégeant une partie de l'isolement dû au travail à distance ou au déplacement constant.

1.2 Les formes de collaboration

Les formes de collaboration sont très variées mais elles impliquent toujours un groupe de personnes qui peuvent partager un intérêt commun et peuvent travailler ensemble. Pour le système, collaborer signifie partager des ressources, être au courant des participants qui collaborent et de l'évolution du travail de collaboration [15, 16]. Selon le type d'application, la collaboration peut être synchrone ou asynchrone.

Suivant la collaboration synchrone, les utilisateurs participent au travail de collaboration et partagent leurs ressources en même temps. Ainsi, les résultats de chaque utilisateur sont propagés instantanément vers les autres. Ce mode de collaboration est dédié aux applications qui nécessitent de voir rapidement les mises à jour des ressources partagées (par exemple : jeux multi-joueurs, édition collaborative, Conception Assistée par Ordinateur (CAO), etc.).

Suivant la collaboration asynchrone, les utilisateurs participent à un travail en commun avec d'autres utilisateurs de manière indépendante et à des moments différents. Il n'y a pas de notion de partage de ressources au même moment. Par conséquent, dans ce type de collaboration, la propagation rapide des résultats n'est pas nécessaire. Ce mode de collaboration est dédié aux applications où la propagation des mises à jour n'a pas une forte incidence sur leur bon fonctionnement (par exemple : forum, listes de discussions, etc.).

1.3 Les environnements de collaboration actuels

La maturité des outils de travail collaboratif a été lente. Si la toute première référence que l'on peut trouver sur le sujet de la collaboration par ordinateurs remonte aux années 40, l'apparition du premier logiciel commercialisable de travail collaboratif n'a eu lieu qu'à la fin des années 80 [17]. Durant ces dernières années, le domaine n'a cessé d'évoluer et plusieurs systèmes collaboratifs ont été proposés [18, 19, 3, 1]. L'idée originelle est partie de l'amélioration des outils traditionnels de conférence, voix et vidéo. Nous allons exposer maintenant les différents environnements inhérents au travail collaboratif.

La messagerie est aujourd'hui une application collaborative incontournable. Il s'agit de la faculté de lancer une discussion, en mode synchrone, avec un ou plusieurs utilisateurs en créant une fenêtre d'échanges de messages depuis la machine de l'initiateur. Parmi les systèmes de messagerie développés dans le cadre de la collaboration, Internet Relay Chat (IRC) [20]. D'autres systèmes tels que le programme AIM (América On-Line Instant Messaging) et le système de messagerie ICQ [18] sont principalement prévus pour la conversation pair à pair. Les versions ultérieures d'AIM incluent la messagerie entre groupes, la transmission de fichiers et les possibilités de transmission de la voix.

Les outils de vidéoconférence sont essentiellement dédiés aux besoins de communication permettant de véhiculer des messages (voix, image, texte) entre différents individus. Ils constituent un socle important du travail collaboratif assisté par ordinateur. Par exemple, les outils de vidéoconférence de Mbone (vic (Video Conference), vat (Visual Audio Tool))[18] fournissent une vidéoconférence multicast, multi-manière qui permet à tous les utilisateurs d'être vus et entendus comme participants à la vidéoconférence. Le système Access Grid [21] a approfondi cette idée en fournissant des possibilités d'interaction de groupes à groupes qui permettent à beaucoup de groupes de participer simultanément. Le système de vidéoconférence à salle virtuelle (VRVS : Virtual Room Videoconferencing Sytem) permet de planifier, gérer et animer des réunions virtuelles (réunions de travail, formations, séminaires...) entre des utilisateurs répartis sur différents sites géographiques. Il a été élargi pour inclure l'interaction entre les sessions de vidéoconférence entre Mbone et Access Grid.

Plus récemment, de nouveaux outils de collaboration sont venus s'ajouter [22, 19, 18]. La demande de collaboration de fichiers s'est parfois traduite par un affichage vidéo des documents à partager impliquant qu'une seule personne pousse l'information lors de formations ou de conférences magistrales. Les technologies pair à pair (Peer to Peer) pour la gestion de fichiers distribués traitent le stockage distribué ou le partage de fichiers. OceanStore [22] et Chord [19] fournissent un stockage distribué fortement disponible par l'utilisation des tables de hachage distribuées. Gnutella, Kazaa, Limewire [18] tiennent compte du partage de fichiers entre un grand nombre de participants dynamiques. Groove [23] facilite le travail

de groupe en soutenant la transmission instantanée de messages, le partage de dossiers et le forum de discussions.

Hayes et Wilson [3] ont développé une application de partage de fichiers musicaux entre des téléphones mobiles compatibles Bluetooth. A cet effet, ils ont adapté le protocole Gnutella [24] pour l'utilisation Bluetooth.

Goel et al. [25] décrivent un protocole de transmission des données pair à pair dans un réseau mobile Ad-hoc. Leur objectif est de partager des fichiers de données entre les utilisateurs des appareils portables. Leur solution permet l'utilisation d'une technique dite "tornado de codes" pour réduire la charge réseau. En utilisant ces codes, un noeud mobile est en mesure de télécharger des fichiers codés en segments des différents utilisateurs à différentes périodes et lieux et est capable de reconstruire le fichier originel.

Klemm et al. [1, 26] proposent une approche de partage de fichiers dans un réseau mobile ad hoc appelée " Routing Optimised Independent Overlay Network " (ORION). ORION crée un réseau superposé au dessus d'un MANET qui prend en charge tous les types de messages requis pour le partage de fichiers, i.e, des requêtes, des réponses et des transmissions de fichiers.

L'article de Kortuem [2] expose des expériences dans le développement de Proem, une plateforme pour développer des applications de collaboration entre des utilisateurs mobiles. Le système Proem est basé sur une architecture de services qui permet la découverte des équipements et leurs capacités. Il prévoit la notion d'identités d'équipements et d'utilisateurs. De cette façon, il est possible d'établir des sessions de collaboration avec des utilisateurs particuliers et empêcher les utilisateurs non autorisés d'y adhérer. Il fournit une solution complète pour le développement et le déploiement d'applications collaboratives dans les réseaux ad hoc. C'est une collection d'outils et d'API (Application Programming Interface).

iClouds [3] est une architecture qui offre une interaction spontanée des utilisateurs mobiles, une collaboration et une transparence des échanges de données en réseau mobile ad-hoc. Les auteurs présentent plusieurs services qu'ils ont construits au dessus de l'architecture iClouds de base. Ils ont construit un service de base d'échange d'information, un

gicteur d'information, notice virtuelle (notice board virtuel) et un service de publicité.

D'autres applications ont été citées dans la littérature. Par exemple, Kortuem discute un système ad hoc d'échange de services [27]. L'idée est que deux individus utilisent leurs dispositifs mobiles pour négocier et échanger des services réels : acheter un livre ou des timbres à la poste, remettre un livre à la bibliothèque locale...etc. L'idée est qu'une personne rend un service pour d'autres parce qu'elle sait qu'un jour elles le feront pour elle.

Dans leur article, les auteurs [28] ont proposé une extension du paradigme de messages instantanés "Instant Messaging and Presence Awareness" . Leur plateforme permet de découvrir de nouveaux dispositifs et services, se rend compte de leur présence et leur état et capable d'interagir avec eux. Une architecture d'agents est proposée pour faciliter l'ajout de nouveaux dispositifs. Il suffit de programmer l'interface de ce dispositif et définir un document XML pour spécifier les services qu'il fournit. Aucun changement n'est exigé à l'application pour interagir avec l'environnement

Dans [29], les auteurs ont récapitulé leurs projets de recherche (Placememo, Hocman, et SoundPryer) dans le domaine de la collaboration mobile, développés sur divers types de supports.

Le Placememo est un système qui facilite le travail des inspecteurs de routes. Il permet de rapporter les défauts de route et de les communiquer aux autres et se servir comme memento (aide mémoire) . Il permet à l'utilisateur de créer des notes sous forme d'enregistrements de voix associés à des positions géographiques tout en conduisant. Il permet aussi à l'utilisateur de partager les notes rassemblées avec les autres tout en étant sur la route.

Le projet Hocman a commencé par une étude sur des motocyclistes. Ce projet reproduit leurs pratiques en matière d'interaction et d'outils qu'ils emploient pour coordonner leurs activités. Hocman est conçu comme une application pair à pair pour les ordinateurs de mains (handheld computers) pour le partage de documents HTML, des clips audio et des images sur un réseau ad hoc.

SoundPryer est une application conçue pour le partage des expériences musicales entre les personnes dans des véhicules dans un entourage immédiat. Ceci est accompli en passant

des fichiers MP3 entre les noeuds dans un réseau ad-hoc.

L'article [30] étudie les mécanismes de collaboration de groupe dans un réseau sans fil ad hoc. L'objectif général de cette recherche est de développer un modèle d'interaction pour la collaboration sur les réseaux ad hoc sans fil et permettre la collaboration à court terme ou à long terme et partager des ressources sans intervention manuelle.

Le département américain de l'Énergie a sponsorisé plusieurs projets de recherche et de développement d'outils de collaboration comme le développement d'un bloc note (notebook) électronique, un cadre d'interopérabilité collaborative, le système CORE 2000, le serveur d'autorisation d'Akenti et les composants Access Grid.

Parallèlement à l'essor de ces technologies, DOE a également sponsorisé le développement de plusieurs pilotes de collaboration dans des disciplines scientifiques spécifiques comme le service virtuel RMN, Matériaux Microcharacterization Collaboratory, le diesel Combustion Collaboratory, et le SpectroMicroscopy Collaboratory. Ces projets ont fourni un important impact dans le domaine de la collaboration scientifique [18].

Conclusion

Les utilisateurs mobiles partageant des intérêts, des tâches et des objectifs communs cherchent la possibilité de collaborer à tout moment et n'importe où avec leurs voisins. Ces nouveaux services de collaboration peuvent aider les utilisateurs dans les activités de leur vie quotidienne. Bien que les systèmes de collaboration développés offrent des avantages indéniables mais ils restent focalisés sur le partage des documents et des données et aucun système existant n'a traité le partage de la puissance de calcul des nœuds mobiles en se souciant de la disponibilité de l'énergie qui est une ressource critique. En effet, les applications de calcul distribué, le développement partagé de programmes, les opérations d'analyse des données, la simulation distribuée interactive sont au cœur des activités de collaboration des futurs environnements mobiles.

Chapitre 2

Conservation d'énergie dans les réseaux ad hoc

Introduction

La gestion d'énergie dans les réseaux ad hoc est cruciale puisque les noeuds ont de sévères contraintes d'énergie. En fait, la capacité énergétique limitée des noeuds impose le mode de communication à l'intérieur de ces réseaux sans fil. Par conséquent, les protocoles conçus doivent judicieusement utiliser les ressources énergétiques finies. Concevoir des protocoles prenant en considération la consommation d'énergie devient donc un défi majeur pour la mise en place d'un réseau viable. À cet égard, utiliser des techniques efficaces optimisant la consommation d'énergie est indispensable pour réaliser des économies d'énergie significatives et prolonger la durée de vie des noeuds et, par conséquent, celle du réseau. Idéalement, un protocole de communication pour les réseaux ad hoc doit faire fonctionner les noeuds uniquement en cas de besoin. Autrement, tous les noeuds doivent être en mode veille " sleep ". De plus, quand une communication est établie, aucune donnée redondante ne devrait être transférée. En outre, le trafic devrait être distribué équitablement parmi les noeuds pour éviter la congestion et la mort prématuré du réseau. Malheureusement, une telle configuration optimale n'est pas faisable puisqu'elle exige une connaissance préliminaire du modèle de génération de trafic et la topologie de réseau. Par conséquent, la recherche s'est orientée vers la notion d'efficacité de la consommation d'énergie (par opposition à la notion d'optimalité). L'efficacité peut être définie comme la maximisation de la

durée de vie du réseau aussi longtemps que possible. Des efforts de recherche approfondis ont été menés pour la conception de protocoles efficaces de gestion d'énergie.

Plusieurs travaux ont démontré que l'activité réseau est très coûteuse en énergie. L'émission ainsi que la réception consomment une énergie importante. Cependant, les activités liées à la communication ne sont pas les seules qui consomment de l'énergie.

L'énergie consommée par un noeud ad hoc peut être classée en deux catégories : l'énergie non liée à la communication et l'énergie liée à la communication. L'énergie liée à la communication peut, à son tour être classée en énergie de traitement des messages, énergie de transmission, énergie de réception et énergie relative au mode veille.

L'énergie consommée par un noeud est également fonction de ses activités réseau au niveau des différentes couches. Au niveau de la couche physique et liaison de données, les noeuds consomment de l'énergie essentiellement pour assurer leur connectivité au réseau. Pour ce fait, les noeuds restent à l'écoute du canal et s'échangent des messages de contrôle. Au niveau du routage, la consommation de l'énergie est due à la participation des noeuds à l'opération de routage et au traitement du trafic de contrôle défini par les protocoles responsables. Nous présentons dans ce qui suit les principales approches existantes pour la conservation d'énergie et ce dans les trois niveaux cités.

2.1 Approches de conservation d'énergie au niveau physique

La puissance de transmission a un impact direct sur la durée de vie des batteries et de la capacité du réseau en terme de trafic effectif transporté. En effet, [31] montre que plus la portée des noeuds est grande, plus la puissance nécessaire à la transmission est grande. De plus, la portée d'un noeud influe directement sur la zone d'interférence. Augmenter la portée, implique aussi augmenter la probabilité d'interférence, le taux de collisions et de perte et diminuer la capacité des noeuds à transmettre. Le contrôle de puissance ou de topologie consiste à adapter les portées et les puissances de transmission des noeuds afin d'assurer une consommation minimale d'énergie tout en gardant la connectivité du réseau.

Il s'agit de trouver une portée de transmission optimale pour les noeuds du réseau, pouvant être commune ou pas, permettant d'optimiser l'énergie consommée lors des communications [32, 33]. Le contrôle de topologie vise à réduire la portée des noeuds, si possible, d'où à réduire les interférences et les collisions permettant ainsi une meilleure conservation d'énergie [34].

Plusieurs propositions existent pour assurer le contrôle de topologie et de puissance. Le mécanisme proposé dans [35] permet d'ajuster la puissance d'un noeud jusqu'à ce qu'il ait un nombre de voisins limité. Ceci n'assure pas dans tous les cas la connectivité du réseau. Souvent, les noeuds peuvent se retrouver dans des îlots séparés avec quelques voisins directs.

Le protocole COMPOW [36] a pour objectif d'ajuster la puissance des noeuds selon une valeur commune. Ce niveau de puissance est le niveau minimal permettant d'assurer la connectivité du réseau. Ce protocole met en évidence l'importance des liens bidirectionnels puisqu'une destination directe ne peut répondre à une source que si sa puissance de transmission est au moins égale à celle de la source. De ce fait, assurer une puissance commune permet d'assurer des liens bidirectionnels. Ce protocole vise aussi à augmenter la capacité de transmission du réseau avec le plus petit niveau d'énergie ou de portée tout en gardant la connectivité du réseau.

Assurer une portée minimale des noeuds tout en gardant la connectivité du réseau pose le problème de recherche de la meilleure couverture du réseau et du contrôle de partitionnement. L'article [37] propose de calculer le digramme de Voronoï sur l'ensemble des noeuds du réseau, dont la topologie et la localisation des noeuds est connue à un instant donné, puis d'en déduire la triangulation de Delaunay qui permet de relier les noeuds ayant des cellules voisines. Le diagramme de Delaunay assure une connectivité totale des noeuds du réseau selon des liens courts assurant une portée minimale.

2.2 Approches de conservation d'énergie au niveau liaison de données

La plupart des solutions proposées à ce niveau définissent un mode veille durant lequel un noeud limite son activité réseau afin de conserver son énergie. Un des mécanismes de référence pour la conservation d'énergie dans les réseaux sans fil est PSM (Power Saving Mode) proposé par la norme IEEE 802.11 [38]. Du point de vue consommation en énergie, un noeud peut avoir trois états. Il peut être en réception, dissipant une certaine énergie, en émission, dissipant une autre énergie ou en mode veille (écoute passive du canal) et dans ce cas il dissipe aussi de l'énergie. Le standard 802.11 définit le PS (Power Save) pour conserver l'énergie des noeuds dans un réseau sans fil avec ou sans infrastructure. Une station utilisant PSM peut voir un des deux états suivants à un instant donné : - Awake : la station utilise toute sa puissance pour envoyer et recevoir des paquets à tout moment. - Doze : la station est incapable de transmettre ou de recevoir, elle utilise le minimum de son énergie. Si elle a des messages à envoyer, elle les sauvegarde localement.

Dans 802.11, on définit deux modes de gestion d'énergie : - Active mode (AM) : la station peut recevoir des paquets à n'importe quel instant. Elle est dans l'état Awake.

- Power Save (PS) : la station peut être dans l'état Doze mais doit se réveiller et revenir à l'état Awake pour recevoir certains messages de contrôle, de synchronisation et recevoir les paquets sauvegardés. Tous les noeuds du réseau doivent être réveillés en même temps pour échanger les messages qu'ils auront sauvegardés durant la période doze. De ce fait, 802.11 définit une fonction de synchronisation qui se base sur l'échange d'un paquet appelé Balise (Beacon) au début de chaque période BeaconInterval. Cette période commence par une sous période appelée ATIMWindow durant laquelle uniquement les messages de contrôle et des annonces (messages ATIM : Announcement Traffic Indication Message) sur des messages sauvegardés doivent être échangés entre les noeuds. Au terme du ATIMWindow, si un noeud n'a annoncé aucun paquet à envoyer et n'a reçu aucune annonce alors il peut entrer en état Doze pour le reste de BeaconInterval. Dans le cas contraire, il doit rester Awake pour réaliser les transferts nécessaires. La synchronisation des noeuds du réseau au niveau

du mécanisme PSM fait que tous les noeuds utilisant le mode PS doivent, si leurs activités le permettent, entrer en mode Doze durant la même période puis se réveiller aux mêmes instants. Cette caractéristique fait que durant une communication entre une source et une destination utilisant une route à plusieurs sauts, plusieurs noeuds en mode Doze peuvent se trouver sur la route. Un paquet traversant le réseau peut rencontrer tout un flût de noeuds en mode Doze empêchant le paquet d'être routé à destination. Ceci représente une faille au niveau de PSM qui peut induire un risque de partitionnement du réseau au moment du routage.

Par ailleurs, un noeud utilisant le mode PSM restera actif durant tout le reste de la période BeaconInterval s'il a reçu des annonces de messages durant la période ATIM ou s'il a des messages à envoyer. Cependant, le noeud ne sera pas réellement en communication durant toute la période puisqu'il n'aura pas accès au canal à tout instant et peut terminer la réception ou la transmission de ses données avant la fin de la période. Une amélioration de PSM consiste à limiter la durée d'activité d'un noeud durant le reste d'un BeaconInterval afin de limiter la consommation d'énergie inutile et ceci en lui allouant des slots prédéfinis pour l'échange de ses données. En dehors de ses slots, le noeud peut entrer en mode doze. Slotted PSM [39] consiste donc à diviser la période allant de la fin de la période ATIM jusqu'au reste de la période BeaconInterval en un certain nombre de slots de temps, chacun sera alloué pour un noeud donné pour effectuer sa communication. Chaque noeud restera actif uniquement durant les slots qui lui seront alloués et pourra ainsi être en mode doze plus longtemps. La réservation des slots pour l'échange de données s'effectue dynamiquement durant la période ATIMWindow. La source inclut les informations de réservation de slots dans les messages ATIM. Chaque noeud maintient une table de réservation de slots qui sera actualisée à chaque réception de trames ATIM ou ACK-ATIM. Un noeud source choisit les slots qu'il va inclure dans la trame ATIM en se basant sur ses informations locales à partir de la table des états des slots et de la quantité du trafic qui sera échangée. A la réception du message ATIM, le noeud destination réplique par un paquet ATIMRE (ATIM Response) dans lequel il inclut la liste de ses slots libres. La source rectifie alors sa réservation selon le nouvel état des slots et envoie un nouveau paquet ATIM.

Une amélioration récente de PSM, adoptant le même principe que Slotted PSM, est le mécanisme TA-PSM (Trafic Aware Power Saving Mode) [40]. Cette nouvelle approche est partie de la constatation du fait que deux noeuds en mode PS restent en état Awake pour le reste de Beacon Interval afin d'échanger des paquets de données, mais peuvent terminer leur communication avant la fin de cette période. TA-PSM consiste à réduire la consommation d'énergie des noeuds du réseau activant PSM en les rendant plus sensibles à la charge du trafic. Chaque source doit à cet effet, indiquer à sa destination l'éventuelle fin de leur communication et ceci en activant un champ particulier dans le dernier paquet qu'elle lui adresse. Ce mécanisme permettra ainsi à une source et une destination d'entrer immédiatement en mode Doze dès qu'elles ne sont plus impliquées dans aucun trafic, sans attendre la fin d'un Beacon Interval et ceci jusqu'à la fin de cette période. Ceci est réalisé par l'ajout d'une indication dans le dernier paquet transmis de la source à la destination indiquant ainsi la fin du transfert de données entre eux.

S-MAC [41] est un mécanisme permettant aux noeuds d'entrer en mode veille pour des périodes assez longues. Dans S-MAC, un noeud entre en mode veille quand un voisin est en cours de transmission.

T-MAC [42] étend S-MAC en ajustant la longueur de la période de réveil des noeuds selon les communications environnantes. Ceci permet de réduire l'énergie consommée suite à l'écoute passive du canal.

D'autres propositions [43] se basent sur une architecture à deux canaux radios assurant une conservation de l'énergie à travers la mise en veille d'un premier canal et l'utilisation du second à une puissance minimale pour réveiller un voisin spécifique ou pour écouter périodiquement le canal.

2.3 Approches de conservation d'énergie au niveau routage

Pour minimiser la consommation de l'énergie au niveau routage, il faut veiller à choisir le meilleur chemin en se basant sur les contraintes d'énergie comme facteur de stabilité

du chemin. Ceci se traduit par le fait de choisir le chemin garantissant la consommation d'énergie la plus faible ou encore de choisir le chemin passant par les noeuds intermédiaires ayant les batteries les plus chargées. Il faut aussi garantir une diminution de l'énergie consommée par la procédure de découverte des routes et ce, en adoptant une technique efficace.

Une autre technique pour diminuer la consommation de l'énergie consiste à optimiser le nombre de messages de contrôle engendrés par le protocole utilisé au niveau du routage.

Divers protocoles ont été implémentés pour assurer la conservation de l'énergie au niveau routage :

Minimum Total Transmission Power Routing (MPTR) [44] : permet de choisir la route minimisant la consommation totale d'énergie en considérant que le meilleur chemin est celui assurant la consommation minimale des capacités des batteries. Il effectue le choix de la route en calculant la puissance de transmission des noeuds. La puissance de transmission exigée pour une bonne réception doit être proportionnelle à une certaine puissance de la distance entre l'émetteur et le récepteur. En fait, des routes avec beaucoup de sauts ayant de courtes portées de transmission sont préférées à celles avec peu de sauts et ayant de longues portées de transmission. En conséquence, plus de noeuds peuvent être impliqués dans la transmission des paquets induisant une augmentation de délai du bout en bout d'une part et à plus de perte d'énergie le long du réseau d'autre part. Comme un plus grand nombre de noeuds sont impliqués dans le routage, tous les voisins des noeuds intermédiaires seront également affectés. L'inconvénient est que cette approche ne considère pas l'énergie restante des noeuds, ainsi elle ne prolonge pas la vie de chaque noeud.

Minimum Battery Cost Routing (MBCR)[45] : considère que la capacité restante des batteries reflète mieux la durée de vie d'un noeud et choisit la route qui maximise la capacité restante de la batterie. Il utilise la somme de l'inverse de la capacité de batterie pour tous les noeuds intermédiaires comme métrique pour le choix de la route. Cependant, cette approche exige que l'addition devrait être minimale, ce qui permet à certains noeuds d'être sur-utilisés parce qu'une route contenant des noeuds avec peu de capacité restante de batterie peut être choisie. L'avantage de ce protocole est qu'il protège les noeuds

contre l'épuisement et qu'il garantit le choix du meilleur chemin dans le cas où les noeuds présentent le même coût. Son inconvénient par contre est qu'un chemin contenant un noeud avec une capacité faible peut comme même être choisi.

Min Max Battery Cost Routing (MMBCR)[45] : choisit la route qui maximise la capacité restante du goulet d'étranglement de la route (le noeud ayant le minimum de capacité restante). Il considère la capacité restante des noeuds comme métrique de choix des routes afin de prolonger la vie de chaque noeud, tels que des noeuds avec des capacités restantes plus petites sont évités et ceux d'une plus grande capacité de batterie sont favorisés en choisissant une route. Ce protocole a pour avantage d'équilibrer l'utilisation des batteries des noeuds. Par contre, il ne garantit pas que le chemin choisi soit le chemin ayant la plus faible consommation d'énergie.

Conditional Min-Max Battery Capacity Routing (CMMBCR)[45] : propose de borner la capacité restante minimale d'un ensemble de chemins puis d'appliquer MTPR. Ceci assurera le choix d'une route vérifiant que la capacité minimale restante est au-dessus d'un certain seuil et que cette route minimise la consommation d'énergie.

Dans [46], trois protocoles de routage orientés puissance et basés sur le protocole de DSR ont été développés : ce sont respectivement les protocoles MDR , LEAR, et EDDSR.

Le mécanisme (MDR : Minimum Drain Rapport) comme son nom l'indique, propose l'écoulement de drainage minimum comme nouvelle métrique à employer conjointement avec la capacité résiduelle de batterie en prévoyant la durée de vie des noeuds selon les conditions courantes du trafic.

Local Energy-Aware Routing (LEAR) a été proposé comme un mécanisme de choix de routes basé sur la puissance avec le but d'équilibrer également la consommation totale d'énergie de tous les noeuds du réseau. Il distribue la décision de la coopération des noeuds en transmettant les paquets entre tous les noeuds dans le réseau.

Enfin, le **mécanisme Energy Dependent DSR (EDDSR)** vient comme une optimisation basée puissance appliquée au procédé de découverte de routes de DSR. Il essaye d'éviter l'utilisation des noeuds à faibles puissances de batteries, en utilisant l'information liée à l'énergie résiduelle dans le procédé de découverte de routes. Considérant que chaque

noeud détermine sa bonne volonté de participer au routage des paquets en se basant sur son niveau d'énergie courante. Néanmoins, considérer ce dernier comme facteur unique peut ne pas être suffisant dans tous les cas, car elle n'assure aucune disponibilité de lien parmi les noeuds d'une plus grande énergie. Les itinéraires choisis dans ce cas peuvent alors être exposés à plus d'échecs de liens qui consomment plus d'énergie à leur tour pendant le processus de maintenance de routes.

Trois extensions d'AODV ont été aussi présentées dans [47], considérant la consommation d'énergie pendant le procédé de découverte de routes.

Dans la première extension appelée Local Energy-Aware Routing (Lear-aodv), chaque noeud mobile se base sur des informations locales sur son niveau restant de batterie pour décider s'il va participer au processus de choix d'une route ou pas. Ceci permet à des noeuds faibles en énergie de conserver leurs batteries en refusant d'acheminer des paquets de données des autres noeuds, qui peuvent soulever dans une certaine mesure la notion de l'égoïsme des noeuds.

La deuxième extension dans ce travail est Power-Aware Routing basé sur AODV (PARAODV). Son objectif principal est de prolonger la durée de vie utile d'un réseau ad-hoc. PARAODV essaie de trouver une route R au temps t de découverte de routes tout en minimisant une fonction de coût.

La troisième extension proposée est Lifetime Prediction Routing basé sur AODV (Lpr-aodv). Cette extension favorise une route ayant une durée de vie maximum, en d'autres termes la route qui ne contient pas des noeuds avec une faible durée de vie prévue. Cette approche prévoit la durée de vie de chaque noeud en estimant son activité passée en transmettant des paquets. Chaque noeud emploie son historique récent comme indicateur de son trafic.

L'étude menée dans [48] propose un nouveau protocole de routage basé sur l'énergie, qui essaie de prolonger la durée de vie des ressources énergétiques. Le protocole proposé, appelé Energy Conserving GRID (ECGRID), exploite le concept d'un protocole de routage appelé GRID [Lia01] en considérant les contraintes d'énergie.

Conclusion

Malgré les progrès technologiques perçus, l'énergie demeure encore une ressource critique, relevant encore des défis pour le déploiement à grande envergure des réseaux ad hoc. Ce critère ayant un impact direct sur la durée de vie du noeud et du réseau par conséquence. L'importance accordée à ce paramètre a motivé le domaine de recherche. Dans cet axe, plusieurs approches ont été proposées afin de réduire les sources de gaspillage d'énergie. Le contrôle de puissance de transmission et le routage orienté conservation d'énergie essaient de réduire la consommation d'énergie en choisissant les noeuds les plus appropriés pour les opérations de communication ou la puissance adéquate à utiliser.

Toutefois, les sources de gaspillage d'énergie persistent. En effet, les noeuds doivent être toujours actifs même s'ils ne sont pas impliqués dans des opérations d'échange de données. Ainsi, de tels noeuds vont perdre de leur énergie à leur insu, à travers l'écoute passive du canal et de l'interception de paquets destinés à des voisins. Le mode conservation d'énergie introduit dans le mécanisme PSM défini dans le standard 802.11, était le fondement de plusieurs travaux [40, 49]. Il permet aux noeuds mobiles d'entrer en mode veille pendant certaines périodes afin de réduire leur consommation d'énergie et de prolonger la durée de vie du noeud et du réseau par conséquence. Par ailleurs, l'activation du mode conservation d'énergie peut pénaliser certains critères de performance importants pour certaines applications, à l'instar du délai d'acheminement.

A la lumière de ces remarques, nous avons proposé une approche permettant d'assurer la plus grande longévité possible d'un réseau ad hoc en équilibrant les consommations énergétiques entre les noeuds voisins. Cette approche a été aussi exploitée conjointement à des techniques de clusterisation afin de conserver de l'énergie tout en préservant la capacité du système, qui fera l'objet des chapitres suivants.

Chapitre 3

Répartition et Equilibrage de Charges

Introduction

Partant du principe l'union fait la force, l'homme a toujours cherché à faire travailler plus de monde sur un même problème pour surmonter les contraintes temporelles et les problèmes techniques rencontrés. A l'invention de l'ordinateur séquentiel, l'homme a su s'inspirer de ses ancêtres en cherchant à multiplier les ordinateurs travaillant ensemble, en parallèle. L'objectif est resté inchangé depuis l'enfance de l'humanité : maîtriser plus de puissance pour traiter les problèmes plus vite, en traiter de plus gros et surtout en les traiter plus rapidement. Cependant, faire travailler plus de monde sur le même problème entraîne entre autre les questions suivantes : qui fait quoi ? Quelles sont les informations qui doivent circuler entre les travailleurs ? Tout le monde a-t-il du travail ? Cette dernière interrogation relève de l'équilibrage de charges.

La problématique de l'équilibrage de charges consiste à mettre à profit, de la meilleure façon, les possibilités d'utilisation des ressources. Elle vise à garantir qu'aucune machine n'est sous chargée ou surchargée et à instaurer une charge uniforme sur toutes les machines [50, 51]. Comment répartir le travail, quel coût de temps et de communication s'autorise-t-on pour prendre une décision d'équilibrage, quand doit-on équilibrer ? voilà quelques unes des questions qui forment le cœur de la problématique d'équilibrage de charges.

Dans ce qui suit nous étudions dans un premier temps la problématique d'équilibrage

de charges dans les réseaux filaires. Nous allons ensuite présenter les principaux travaux de recherche traitant l'équilibrage de charges dans le contexte particulier des réseaux ad hoc et décrivons quelques protocoles spécialement conçus pour un tel environnement.

3.1 Equilibrage de charges dans les réseaux filaires

En centrant notre attention sur l'étude des mécanismes d'équilibrage de charges dans les réseaux filaires, nous avons rencontré une volumineuse littérature disponible dans ce domaine.

Le problème de la répartition de charges a été étudié depuis plus d'une trentaine d'années dans les domaines des systèmes répartis et parallèles [4, 5, 52]. Profitant de cette expérience nous faisons un petit parcours.

Dans leur article [53], Casavant et Kuhl présentent le problème général de la répartition de charges de façon très simple, d'un côté se trouvent les ressources physiques du système (mémoire, périphériques, réseaux, processeurs), d'un autre les besoins de l'application (les consommateurs), et au milieu un dernier composant qui est la politique d'allocation qui décide où seront allouées les tâches pour accéder aux ressources.

Pour Eager [52] l'amélioration des performances du système peut être recherchée par deux moyens, le partage de charges (load sharing) et l'équilibrage de charges (load balancing). Dans le partage de charges, on cherche à maintenir la plupart du temps tous les processeurs occupés. Ceci est réalisé en transférant une partie de la charge de travail de la machine surchargée vers d'autres machines. Le placement ou la migration de processus n'est donc envisagé que lorsque la charge de travail local dépasse un certain seuil. Par contre dans l'équilibrage de charges l'allocation des processus est envisagée chaque fois que les conditions globales du système changent, c'est à dire à chaque création ou terminaison de processus.

Dans cette thèse nous utilisons comme synonymes les mots répartition, partage ou équilibrage de charges où l'idée de base est de répartir la charge entre les noeuds en évitant qu'un noeud ne soit inactif alors que des tâches restent en attente sur d'autres noeuds.

Une approche pour l'équilibrage de charges dynamique est d'utiliser le passé pour prédire l'avenir, c'est-à-dire d'utiliser la vitesse de calcul observée pour chaque processeur de décider de la redistribution du travail. De nombreux auteurs se sont intéressés à cette approche, notamment dans [54] où l'équilibrage de charges implique d'assigner à chaque processeur un travail proportionnel à ses capacités, en minimisant le temps d'exécution du programme. Bien que l'équilibrage de charges statique puisse résoudre beaucoup de problèmes (par exemple ceux provoqués par l'hétérogénéité des processeurs) pour la plupart des applications régulières, la charge passagère due aux utilisateurs multiples sur un réseau de stations de travail nécessite une approche dynamique afin d'équilibrer la charge. Les auteurs examinent les différents comportements des stratégies d'équilibrage de charges : global contre local et centralisé contre distribué. Ils montrent ainsi que chaque stratégie est la meilleure pour différentes applications données en fonction des variations des paramètres du programme et du système.

Une vision très particulière de l'équilibrage de charges dynamique est basée sur la redistribution de données parmi les processeurs participants pendant l'exécution d'un programme [55]. Cette redistribution est faite en transférant des données des processeurs les plus chargés vers les processeurs les moins chargés. Cette phase d'équilibrage de charges peut être centralisée par un seul processeur ou être distribuée sur l'ensemble des processeurs.

Un autre exemple des nombreux modèles existants pour les stratégies d'équilibrage de charges dynamiques dans [50] où H. Willebeeck-Lemair et P. Reeves présentent cinq approches de manière à illustrer la différence entre connaissance (exactitude de chaque décision d'équilibrage) et surcoût (quantité de processus et communication supplémentaires induite par la phase d'équilibrage).

Plusieurs auteurs [54, 56] se sont intéressés à l'équilibrage de charges dynamique pour des applications parallèles. Par exemple, J. Weissman examine dans [56] le problème de l'adaptation d'applications parallèles de données dans un environnement dynamique de stations de travail. Il développe un cadre de travail analytique pour comparer un large éventail de stratégies : équilibrage de charges dynamique, déplacement de données, ajout

ou suppression de processeurs.

D'autres auteurs, comme Lan et Taylor [57] se sont intéressés eux aussi à l'équilibrage de charges dynamique mais cette fois sans l'hypothèse d'applications parallèles comme précédemment. Ils proposent un modèle d'équilibrage de charges dynamique pour systèmes distribués. Leur modèle prend en compte l'hétérogénéité des processeurs ainsi que l'hétérogénéité et la charge dynamique du réseau. L'hétérogénéité des processeurs est traitée en leur attribuant un poids relatif à leur vitesse de calcul. La charge de travail est donc distribuée en fonction de ces poids.

Un autre modèle d'équilibrage de charges dans les systèmes distribués a été présenté dans [58]. Dans ce travail, deux types de noeuds sont distingués : les noeuds primaires et les noeuds de soutien. Les primaires sont les noeuds principaux et les noeuds de soutien sont utilisés pour traiter les surcharges. Quand un noeud primaire est surchargé, il recherchera un autre noeud primaire. Si un tel noeud est trouvé, il maintiendra l'équilibrage de charges avec le noeud primaire qu'il a sollicité. S'il n'est pas trouvé, il sollicitera un noeud de soutien qu'il interrompra. Le noeud de soutien exécutera l'ISR (Interrupt Service Routine). En ISR, chaque noeud de soutien maintient une file des processus auxquels sont assignés des priorités. Les processus des noeuds primaires leur sont aussi attribués des priorités. Quand un processus migre d'un noeud primaire à un noeud de soutien, il sera inséré suivant sa priorité dans la file d'attente des processus du noeud de soutien.

3.1.1 Classification des stratégies d'équilibrage de charges

De nombreuses classifications ont été proposées pour regrouper les méthodes de répartition de charges [53, 59]. Celles-ci sont traditionnellement mises en place au sein de deux politiques :

3.1.1.1 Politique d'information

Dans la politique d'information sont regroupés :

- le choix du déclenchement de l'évaluation de la charge,
- la stratégie d'évaluation elle-même,

- le choix du déclenchement de la répartition.

Le premier critère caractérisant une politique d'information concerne le déclenchement de l'évaluation. Deux classes principales peuvent être identifiées :

- Déclenchement à la demande : une requête d'information est émise par un ou plusieurs processeurs. Le demandeur est soit un processeur surchargé voulant céder une partie de sa charge, soit un processeur inutilisé ou sous chargé voulant augmenter sa charge. Des exemples sont proposés par Kumar et al. [51].

- Déclenchement périodique : un relevé d'information est effectué suivant une fréquence qui peut être fixe ou modifiée dynamiquement en fonction de l'évolution du système. L'ensemble des processeurs participe à cette demande d'information [60].

Le deuxième critère définissant une politique d'information est le type d'évaluation réalisée. L'information la plus couramment utilisée est la charge moyenne par processeur. Il peut s'agir d'une moyenne sur l'ensemble des charges ou seulement d'une information recueillie auprès d'un voisinage. D'autres informations sont nécessaires pour avoir une vue, même partielle, de la répartition de la charge sur l'ensemble des processeurs. Des vecteurs de charges peuvent ainsi transiter entre les processeurs. Dans une approche centralisée, un serveur accumule les informations qu'il pourra ensuite fournir à la demande. Dans une approche distribuée, à la fin de l'évaluation, chaque processeur détient des informations sur l'état du système.

Enfin, la politique d'information inclut le choix du moment du déclenchement de la redistribution. Ce choix est important car un déclenchement trop tardif provoque une augmentation de l'inactivité des processeurs ou d'éventuels problèmes de saturation, alors qu'une redistribution trop précoce entraîne des surcoûts inutiles en communication et en traitement. En général, ce choix est ajusté par le réglage de seuils sur les déséquilibres constatés. Comme précédemment, le déclenchement est soit décentralisé si chaque processeur décide, à partir des informations recueillies, de la nécessité d'une redistribution, soit centralisé si un unique processeur prend l'initiative du déclenchement.

3.1.1.2 Politique de redistribution

Dans la politique de redistribution, sont regroupés :

-le choix des processeurs, des processus et des données impliqués dans la redistribution,
- le choix des schémas de communication pour la redistribution proprement dite. La sélection des processeurs impliqués dans la redistribution est, en général, réalisée par comparaison de la charge locale de chaque processeur avec un ou plusieurs seuils. Elle peut aussi se faire après consultation de l'état de processeurs sur un voisinage [61] ou sur l'ensemble des machines [60]. La désignation, au niveau de chaque processeur, des éléments de charge à distribuer est réalisée localement et dépend du type de charge traité. Pour une charge représentée par des tâches, des objets ou des données indépendants ou faiblement couplés, la redistribution pourra être des plus exotiques et permettre toutes sortes de choix : sélection aveugle, aléatoire, aux enchères, par diffusion [62, 63], etc. Si la redistribution doit respecter des contraintes (par exemple, de localité, de voisinage), la politique appliquée sera étroitement liée aux types d'éléments de charge considérés. Il existe néanmoins une caractéristique de la redistribution qui est indépendante du type de charge considéré : la politique peut être centralisée si les décisions sont prises globalement, distribuée si les choix sont locaux, ou hiérarchique si des groupes de processeurs sont formés pour établir une politique à leurs niveaux [51].

3.1.2 Structure générale d'un mécanisme d'équilibrage de charges

Un mécanisme d'équilibrage de charges peut être décrit par trois éléments essentiels qui englobe tous les choix énoncés dans les deux politiques précédentes.

- 1- un gestionnaire de l'état de charge du système (composant d'information) ;
- 2- un élément de contrôle et prise de décision (composant de contrôle).
- 3- un mécanisme de transfert de la charge.

3.1.2.1 Gestionnaire de l'état de charge du système

Le gestionnaire de l'état de charge (composant d'information) a comme objectif de récupérer les informations nécessaires sur l'état de chaque machine pour établir la charge globale du système. Ces informations nous permettent de déterminer les meilleures machines réceptrices (les moins chargées) ou de déterminer les machines les plus chargées ou potentielles émettrices de charges. Deux sous fonctionnalités forment ce gestionnaire :

- l'estimation de la charge de chaque machine du système ;
- l'estimation de l'état global du système.

3.1.2.2 Élément de contrôle et de prise de décision

Ce contrôleur (composant de contrôle) aura diverses sous fonctionnalités. À ce niveau, les aspects auxquels il faut s'intéresser sont les principales activités que chaque machine doit réaliser : l'activation, la sélection et le transfert

- **Technique d'activation** : la technique d'activation détermine à quel moment la distribution (ou redistribution) des tâches doit être effectuée, soit à partir des informations fournies par le gestionnaire du système. Un élément spécifique peut être responsable de l'activation (rôle du contrôleur ou des contrôleurs) ou on laisse chaque machine autonome.
- **Technique de sélection** : la technique de sélection détermine les différentes machines déséquilibrées qui joueront le rôle d'émetteur ou de récepteur. Une machine est considéré comme déséquilibrée si elle se trouve dans l'un des états suivants : elle est considérée comme chargée ou futur émettrice soit comme faiblement chargée ou inactive et est alors considérée comme récepteur.
- **Technique de détermination locale de la charge à transférer** : Il faut faire le choix de la tâche à transférer.

3.1.2.3 Mécanisme de transfert de la charge

Nous définissons le mécanisme de transfert comme l'action de déplacer une tâche de la machine source vers la machine réceptrice. Il ne s'agit pas de faire un choix mais d'effectuer la migration qui a été décidée par l'élément de contrôle.

3.2 Equilibrage de charges dans les réseaux ad hoc

Les réseaux ad hoc héritent des mêmes propriétés et problèmes liés aux réseaux filaires. D'autres caractéristiques spécifiques aux réseaux ad hoc conduisent à ajouter une complexité et des contraintes supplémentaires qui doivent être prises en compte lors de la conception des algorithmes d'équilibrage de charges dans les réseaux ad hoc, à savoir l'énergie, la mobilité...etc.

Une étude bibliographique relative aux travaux de recherche sur l'équilibrage de charges [64, 65, 66, 67, 68, 69, 70] permet de distinguer deux niveaux sur lesquels il serait possible de définir une stratégie d'équilibrage de charges : niveau communication, niveau calcul.

3.2.1 Equilibrage de charges au niveau communication :

Dans le cadre des réseaux de communication, l'un des problèmes majeurs de la qualité de service est la congestion. En effet, le phénomène de congestion est dû entre autre, au fait que de nombreux paquets émis empruntent les mêmes chemins. Par conséquent, l'augmentation de la bande passante offerte ne peut résoudre le problème qu'à court terme. Afin de résoudre ce problème, plusieurs solutions ont été proposées, telles que les algorithmes de routage multi chemins, qui permettent de distribuer le trafic entre une source et une destination sur plusieurs chemins différents, tout en tenant compte du niveau de saturation des noeuds intermédiaires composant les différents chemins. Dans le monde des réseaux sans fil, le problème devient plus compliqué à cause de la limitation de la bande passante sur les liens radios, ainsi que de la vulnérabilité des communications radios. De plus, dans un réseau ad hoc, les problèmes de saturation du réseau ou de congestion sont plus fréquents et la quantité d'énergie des noeuds mobiles est limitée, ce qui rend l'équilibrage de charges très important pour ce type de réseaux.

L'équilibrage de charges au niveau communication est devenu un mécanisme efficace pour répondre aux exigences de qualité de service des applications. Il consiste à choisir le meilleur chemin en évitant les chemins encombrés. Son but est de répartir la charge excessive d'un nœud sur ses voisins afin d'améliorer les performances, d'exploiter efficacement

les ressources réseaux (buffer, canal radio) et de réduire le taux de perte des paquets et les délais d'acheminement.

Plusieurs métriques sont utilisées dans le choix du meilleur chemin. Les solutions proposées dans la littérature utilisent généralement une formulation adaptée à la résolution d'un problème d'optimisation. Selon la variabilité du trafic et les objectifs d'optimisation (cas moyen, pire des cas), la répartition de charges peut se faire à partir de connaissances sur le trafic à priori. Les mesures utilisées peuvent être réalisées en temps réel ou à l'avance (on line/off line). Lorsque les mesures sont obtenues on line, la répartition est réactive, alors que des mesures off line sont par nature associées aux modèles proactifs permettant un choix des routes par anticipation.

Dans [71], Sridharan et al proposent une approche ne nécessitant pas de modifications spécifiques sur le mécanisme de routage traditionnel. En effet, la solution proposée agit en amont et permet de contrôler le choix d'un meilleur chemin. Les auteurs reformulent le problème du routage optimal sur les meilleurs chemins donnés dans [72] pour un routage par destination. Le dual de ce problème permet de déterminer la valuation à attribuer à chaque lien si la matrice de trafic est connue. Une bonne référence sur la façon de construire une telle matrice de trafic peut être trouvée dans [73].

Wang et al. [72] proposent une méthode permettant de mettre en oeuvre de l'ingénierie de trafic. Il s'agit de manipuler la valuation des liens du réseau pour adapter le routage aux demandes. Ainsi, il est possible de résoudre un problème d'optimisation minimisant l'utilisation du lien le plus chargé en utilisant seulement les meilleurs chemins de coûts égaux.

Les travaux d'ingénierie de trafic par équilibrage de charges peuvent être classifiés en trois grandes catégories :

- Prédire les demandes moyennes : l'ingénierie de trafic est mise en place pour optimiser les demandes représentatives sur un ensemble de matrices de trafic collectées dans le temps.
- Prévenir le cas pire : l'ingénierie de trafic prend en compte le pire des cas sur l'ensemble des matrices de trafic.
- Réagir en fonction des demandes instantanées : l'ingénierie de trafic est réalisée à partir

de mesures temps réel obtenues via des sondes évaluant la qualité des routes.

Dans le cas où le réseau dispose d'outils de métrologie permettant de maintenir un historique des matrices de trafic évoluant dans le temps, il est possible d'estimer la charge à venir sur l'intervalle de temps suivant. Les travaux de Zhanga et al. [74, 75] sont un exemple d'approche utilisant l'ingénierie de trafic . Lorsque la charge du réseau est relativement stable, en conditions normales, cette catégorie d'algorithmes permet d'obtenir des résultats efficaces. Cependant, lorsque les changements de charge sont brusques et importants, la seconde catégorie, le routage " cas pire : oblivious ", permet d'anticiper les variations de charge ne correspondant pas au trafic moyen mesuré sur les périodes antérieures. Ces cas sont relativement fréquents, en particulier lors de l'occurrence d'une panne car la charge induite par la redirection de trafic est imprévisible. Les propositions avancées par Applegate et Cohen [76, 77] vont précisément dans ce sens. En revanche, cette approche n'est pas optimale en condition normale lorsque la charge correspond au cas moyen.

D'autres protocoles tel que MATE (MPLS Adaptative Traffic Engineering : MultiProtocol Label Switching Adaptative Traffic Engineering) et TeXCP (Traffic engineering with an eXplicit Control Protocol) [78, 79] utilisent des sondes permettant d'estimer en temps réel l'utilisation d'un noeud. A la différence de MATE, TeXCP est un protocole distribué ne nécessitant pas la présence d'un gestionnaire pour permettre à chaque source de connaître l'état du réseau. Les sources sont informées par les noeuds intermédiaires de l'utilisation des liens par un feedback similaire à XCP [80]. Par ailleurs, TeXCP privilégie l'utilisation des chemins les plus courts car la métrique utilisée a tendance à allonger les délais d'acheminement contrairement à MATE qui a pour objectif de minimiser les délais.

Une autre proposition, COPE (Common-case Optimization with Penalty Envelope) , [81]), affirme que les protocoles réactifs tels que TeXCP peuvent, durant de larges périodes transitoires, être pénalisés par des changements de charge brusques et importants. Les auteurs utilisent à la fois la méthodologie classique par prédiction via une séquence de matrices de trafic obtenues sur des intervalles de temps consécutifs et proposent une nouvelle approche basée sur le concept d'enveloppe couvrante (penalty envelope). Le principe est de restreindre l'espace des solutions tel que les proportions soient optimales à l'intérieur de

cette enveloppe. En pratique, il s'agit d'une contrainte supplémentaire permettant d'inclure dans l'analyse les pires cas possibles. Les dimensions de l'enveloppe sont déterminées en fonction de toutes les matrices de trafic possibles, dont celles obtenues pour la prédiction. Le but est de minimiser la fonction objectif sur les matrices de prédiction, en vérifiant que la contrainte liée à l'enveloppe couvrante soit satisfaite quel que soit le trafic.

Dans le cas de partage de charges avec un routage saut par saut, la répartition des flux est effectuée à chaque saut. Chaque routeur est potentiellement capable, si plusieurs prochains sauts sont activés vers une destination, de distribuer le trafic. Pour une destination donnée, la commutation d'un paquet sur un noeud actif dépend de la proportion qui lui a été attribuée. Les paquets sont acheminés selon leurs destinations et les proportions établies sur chaque routeur. Le choix de l'aiguillage est calculé à chaque saut.

Villamizar propose dans [82] une technique intitulée OSPF-OMP (Open Shortest Path First- Optimized Multipath). Ses travaux utilisent une mesure de bande passante 'Charge équivalente : Equivalent load', adaptée aux liens chargés subissant des pertes. L'auteur propose d'intégrer dans ses mesures de débit la probabilité de perte P sur chaque lien. Pour évaluer la bande passante réelle, Villamizar utilise une approximation de l'estimation par dérivation de Mathis et al. (MSM97). Le débit BW d'un flux est fonction des caractéristiques de ce flux : le Round Trip Time (RTT), la taille du segment maximal (MSS) et une constante C caractérisant le mécanisme d'acquittement utilisé. Ainsi, Villamizar définit une métrique permettant de déterminer le lien le plus chargé.

Dans [64], les auteurs proposent un nouveau protocole de routage avec équilibrage de charges (LBAR : Load Balanced Ad hoc Routing protocol). LBAR définit une nouvelle métrique pour le routage connue sous le nom " degré d'activité nodale " pour représenter la charge sur un noeud mobile. Dans LBAR, l'information de routage pour toutes les routes de la source à la destination est expédiée à travers des messages " setup messages". Ces messages incluent l'information sur l'activité nodale de tous les noeuds sur le chemin traversé. Après le regroupement de l'information sur tous les chemins possibles, la destination fait alors un choix du chemin en se basant sur la valeur de meilleur coût et envoie une réponse au noeud source.

Le protocole de routage de zone (ZRP : Zone Routing Protocol) est une approche hybride (proactive/reactive) pour permettre une certaine extension du réseau ad-hoc. Cette approche a le potentiel d'être plus efficace que les approches de routage traditionnelles dans la génération du trafic de contrôle. Malheureusement, sans mécanisme d'équilibrage de charges, ZRP ne peut pas garantir une qualité de service pour des applications sensibles aux délais dans le cas de la congestion du réseau. Dans [83], un nouveau protocole ARLB (Application Routing Load Balancing) basé sur ZRP et combinant le routage avec qualité de service et équilibrage de charges est présenté. Dans le protocole ARLB, la meilleure route est le chemin ayant une charge de trafic minimale et un minimum d'interférence des nœuds voisins.

Afin d'analyser l'effet de la distribution du trafic sur les chemins dans le protocole AZRP (Adaptive Zone Routing Protocol), dans [84] le problème d'équilibrage de charge est considéré comme un problème d'optimisation en utilisant un paramètre de congestion (CP : Congestion Parameter). Le paramètre CP est calculé en fonction du nombre de paquets transitant entre toute paire (source et destination) et le nombre de liens. Les résultats de simulation montrent que le routage avec équilibrage est plus efficace en termes d'utilisation de réseau qu'avec AZRP.

Dans [85], les auteurs proposent une technique de routage efficace et pratique pour les réseaux ad hoc, basée sur une nouvelle métrique de routage appelé degré-libre (FD Free-Degree). Dans la technique de routage proposée FDAR (Free-Degree Adaptive Routing), les paquets de données sont envoyés en évitant les chemins encombrés, afin de réaliser un court délai de bout en bout et un équilibrage modéré de charges du réseau global. Le protocole proposé est un protocole réactif orienté trafic dont la métrique de routage adoptée est la notion de degré libre (FD) des routes. La valeur FD des nœuds est définie comme le rapport entre son débit de transmission et le carré de son débit de réception. Chaque nœud met à jour régulièrement ses débits de transmission et de réception toutes T unités de temps. La valeur FD d'une route entre une source et une destination est définie comme la somme des valeurs FD des nœuds intermédiaires composant cette route. Après avoir calculé la valeur FD pour chaque route reliant une source et une destination, la meilleure route est

définie comme la route ayant la valeur FD maximale.

L'équilibrage de charges est devenu un mécanisme efficace pour répondre aux exigences en temps réel des applications multimédias (comme la vidéoconférence). Mais beaucoup de protocoles de routage actuels utilisent un routage au mieux (best-effort) et à unique chemin et donc ne sont pas adaptés pour l'équilibrage de charges. L'article [86] présente un protocole de routage à multiples chemins et à noeud-disjoint nommé (EMR : Enhanced Multi-path Routing). Pour équilibrer la charge entre plusieurs chemins, une équation de poids (LBW : Load Balancing Weight) est utilisée. Les résultats de simulation indiquent que le mécanisme d'équilibrage de charges proposé diminue les délais de bout en bout et réduit le taux de perte des paquets.

Etant donné que le routage est exécuté par des noeuds avec des ressources limitées, la charge devrait être efficacement distribuée. Autrement, les noeuds surchargés peuvent composer un goulot d'étranglement qui diminue les performances du réseau en causant une congestion et de plus grands délais d'acheminement. Les auteurs de [87] proposent un mécanisme d'équilibrage de charges qui pousse le trafic plus loin du centre du réseau car les noeuds au centre du réseau sont beaucoup chargés que les autres. Fondamentalement, ils ont présenté une nouvelle métrique de routage (proactif et réactif) qui tient compte du degré des noeuds.

D'après leur étude, il semble que la charge est maximale au centre du réseau, diminue à mesure que nous allons plus loin du centre et est minimale à la périphérie du réseau. Par conséquent, il suffit d'éviter la circulation du trafic au centre du réseau pour que la charge réseau devienne équilibrée. L'idée de leur travail est la suivante : au lieu de choisir les plus courts, ils ont suggéré de choisir des chemins relativement courts mais formés par des noeuds qui sont les plus loin possible du centre du réseau. Ainsi, la taille de la table de routage d'un noeud caractérise sa centralité. La métrique de routage considérée est la moyenne des tailles des tables de routage des noeuds constituant une route. La meilleure route est celle qui minimise cette valeur.

Dans le cas de la congestion du réseau, la plupart des protocoles de routage existant dans les réseaux ad hoc tel que AODV et DSR n'essayeront pas de découvrir un nouvel

itinéraire s'il n'y a aucun changement de topologie du réseau. Par conséquent, avec une faible mobilité, le trafic peut devenir concentré sur quelques noeuds spécifiques. Puisque les noeuds mobiles ont de faibles capacités de calcul et de batterie, la concentration de trafic sur un noeud spécifique devient un phénomène très gênant. Par conséquent, afin de remédier à cette situation plusieurs extensions des protocoles existants ont été proposées. Une modification du protocole de routage DSR a été introduite dans [65]. Leur approche repose sur l'utilisation de multiples chemins généralement appelé routage multi chemins afin d'équilibrer le trafic. Leur algorithme repose sur le calcul des plus courts chemins basés sur une métrique de coût adaptative en attribuant des poids aux liens. Chaque court chemin calculé devient une route pour le protocole de routage DSR. Le flux de données est uniformément répartie sur les chemins déjà calculés.

Dans [88], les auteurs ont proposé un nouveau protocole de routage WLBR (weight-based load balanced routing). La stratégie de routage proposée tente de distribuer le trafic sur des routes constitués de noeuds ayant des durées de vie relativement plus longues et moins de trafic à acheminer. Chaque route est pondérée par un poids "W" qui est fonction de trois paramètres : l'énergie de la route (la somme des énergies de tous les noeuds composant la route), la longueur des files d'attente des noeuds constituant la route et le nombre de sauts. Une route ayant un plus grand poids est choisie comme une meilleure route entre une source et une destination.

Dans [66], une nouvelle approche de routage appelée SLAR (Simple Load balancing Ad hoc Routing) est proposée. Elle consiste à résoudre le problème de concentration de trafic en permettant à chaque nœud de contrôler sa propre charge et renonçant à son rôle d'expéditeur de paquets dans le cas d'une charge élevée. En effet, si la quantité de trafic d'un noeud atteint un certain seuil, ce dernier change son état et envoie un message "GIVE-UP" à tous les noeuds sources passant par lui afin de les avertir pour chercher d'autres routes. Chaque noeud recevant le message "GIVE-UP" initie la procédure de découverte de nouvelles routes afin de contourner le noeud surchargé.

Une extension du protocole AODV a été présentée dans [67]. Les auteurs ont analysé l'effet de la distribution du trafic entre chemins multiples en réalisant un algorithme

d'équilibrage de charges basé sur AODV-MP (Ad Hoc On-Demand Distance Vector Routing with MultiPath), qui distribue les paquets sur divers chemins en fonction de l'état de charge du réseau. AODV-MP limite le nombre de chemins multiples utilisés à trois. Selon le nombre de chemins trouvés (un, deux ou trois), le nœud source distribue les paquets sur ces différents chemins en fonction du nombre de routes valides sur ces derniers. Leurs résultats montrent que le protocole d'équilibrage de charges permet de diminuer la congestion, réduire le délai de bout en bout et d'augmenter le taux de livraison de paquets, ainsi la performance du protocole de routage multi chemin peut être amélioré.

Dans [89], une approche pratique d'équilibrage de charges de trafic dans les réseaux ad hoc maillés (wireless mesh networks (WMN)) est présentée. Dans cette approche, deux mécanismes sont distingués : le mécanisme de détection de la charge de trafic (traffic load detection) pour détecter les nœuds lourdement chargés et le mécanisme de distribution de charge pour orienter une partie du trafic des nœuds surchargés vers les nœuds faiblement chargés.

Une étude comparative d'un ensemble de protocoles de routage avec équilibrage de charges est donnée dans [88].

3.2.2 Equilibrage de charges au niveau traitement (calcul)

A l'heure actuelle, les réseaux ad hoc sont exploités dans des contextes d'application divers et des environnements variés. Dans ces différents contextes d'application, un utilisateur a besoin d'accéder aux ressources présentes dans le réseau ad hoc. Ces ressources peuvent correspondre à des documents mis à disposition, par exemple lors d'une réunion de travail. Elles peuvent aussi désigner des ressources matérielles. Enfin, elles font référence à des services effectuant des traitements et des calculs. Dans ces différents contextes d'application, un mobile doit pouvoir accéder de façon transparente aux ressources de calcul présentes dans le réseau ad hoc. Les stratégies développées dans ce sens tentent de distribuer les tâches de calcul d'une manière à équilibrer la charge afin de réduire les temps d'exécution et améliorer l'utilisation des nœuds mobiles. Peu de travaux traitant le problème de ce point de vue peuvent être cités dans la littérature.

L'équilibrage de charge dans [68] est traité comme une extension de la procédure d'élection de clusterheads. Il permet à tous les noeuds d'avoir une égalité des chances à être un clusterhead et ainsi prolonger sa durée de vie. Quand un réseau ad hoc est découpé en clusters, un des noeuds est choisi comme clusterhead dans chaque cluster. Un clusterhead a plus de responsabilités qu'un noeud ordinaire, ceci peut avoir comme conséquence un plus grand épuisement de sa batterie par rapport aux autres. Les auteurs proposent une heuristique d'équilibrage de charge pour prolonger la durée de vie de clusterhead avant de se retirer et permettre à un autre noeud de devenir clusterhead. Dans ce travail, un processus de délégation du rôle de clusterhead a été mis en place pour permettre à un noeud, estimant qu'il ne pourrait plus assurer ce rôle, de passer la main à un autre noeud. Un mécanisme a été défini pour garantir l'élection d'un successeur, choisi parmi les voisins. Ainsi, l'équilibrage de charges assure l'équité et la stabilité du réseau ad hoc.

L'article [90] propose une approche hACO (hybride Ant Colony Optimization) qui choisit un ensemble de clusterheads de telle manière à permette aux noeuds restants d'être distribués entre ces clusterheads tout en s'assurant que la charge maximale qu'un clusterhead peut prendre en charge est maintenue. Des fourmis artificielles sont employées pour choisir les clusterheads. Après le choix d'un clusterhead, une heuristique est employée pour déterminer les membres du cluster.

Dans [69], une tentative d'équilibrage de charges pour trouver le meilleur noeud pour partager la charge est présentée. Le papier fournit des résultats intéressants mais un certain nombre d'aspects ne sont pas détaillés tels que le surcoût engendré par l'équilibrage et l'énergie des noeuds.

Conclusion

Nous avons exposé dans ce chapitre les différents travaux traitant la répartition et l'équilibrage de charges dans les réseaux. L'équilibrage de charges dédié aux environnements mobiles est confronté aux éléments qui caractérisent les infrastructures de ces environnements, à savoir les terminaux mobiles et les réseaux mobiles sans fil. Chacun de

ces éléments apporte des contraintes que les systèmes doivent prendre en compte dans les fonctionnalités qu'ils offrent aux applications. Ces contraintes sont essentiellement liées à la faible autonomie des entités mobiles et à la connectivité intermittente dans les réseaux mobiles sans fil.

Bien que la plupart des méthodes existantes considèrent la charge au sens général (c'est-à-dire la charge de communication due à la fonctionnalité du routage), aucune méthode ne mentionne l'équilibrage de plusieurs types de charges (charge de communication et charge de traitement liée aux services et aux applications supportées par chaque nœud) en même temps.

Dans notre travail (chapitre 5) nous nous intéressons au problème d'équilibrage de charges à partir d'un point de vue différent en se basant particulièrement sur une représentation hiérarchique du réseau mobile ad hoc (regrouper les nœuds mobiles en clusters, gérés par des cluster-heads), en tenant compte de l'hétérogénéité des nœuds mobiles. Avant de présenter notre proposition d'équilibrage de charges dans les réseaux ad hoc, qui se base sur l'organisation des réseaux ad hoc en clusters, nous allons présenter dans le chapitre suivant les principales méthodes de clustering présentées dans la littérature.

Chapitre 4

Le Clustering

Introduction

Un MANET (Mobile Ad hoc Network) est classiquement considéré à plat : chaque terminal intervient de façon égalitaire dans le réseau ; la charge de communication devant donc être répartie de façon uniforme dans le réseau pour ne désavantager aucun noeud. Ainsi, le véritable défi inhérent aux MANET résidait dans la proposition de protocoles distribués permettant de réaliser les fonctions réseaux classiques (telles que le routage, la découverte de services, la réservation des ressources) directement sur un grand nombre de noeuds non hiérarchisés. Proposer des protocoles fonctionnant dans des réseaux déstructurés constituait une des questions clé. Plusieurs auteurs [91, 92, 93] pensent au contraire qu'un réseau ad hoc doit fondamentalement être organisé afin de rendre les protocoles plus efficaces. Organiser un réseau représente l'introduction d'un comportement global visant à la création d'une vue logique structurée, organisée, basée sur un ensemble de terminaux qui, matériellement, étaient au départ désorganisés. Deux grandes catégories de structures virtuelles sont définies dans la littérature : les dorsales permettent de collecter le trafic de contrôle et forment une dorsale centralisatrice dans le réseau. Les clusters, eux, introduisent une hiérarchie dans le réseau, découpant le réseau en zones homogènes.

Etant donné que l'auto organisation ne constitue aucunement une fin en soi dans notre travail, nous discutons dans ce chapitre uniquement le clustering. Pour nous, cette structure virtuelle (clustering) ne sert que d'intermédiaire pour le réel problème traité dans cette

thèse.

4.1 Définition du clustering

Le clustering consiste en un découpage virtuel du réseau en groupes de noeuds proches géographiquement. Ces groupes sont appelés clusters. Ils sont généralement identifiés par un noeud particulier, un chef de groupe aussi nommé cluster-head. Dans la plupart des algorithmes de clustering, les clusters sont construits à partir d'une métrique particulière qui permet d'assigner un chef à chaque cluster ; le cluster étant alors constitué du cluster-head et de tous les noeuds qui lui sont rattachés. L'idée initiale du routage hiérarchique est de permettre à chaque entité de stocker la totalité des informations de son cluster et seulement une partie des informations concernant les autres clusters. Cela minimise la taille des tables de routage et la quantité de trafic généré. Outre le fait de rendre le routage plus efficace, le clustering présente également d'autres avantages. Il peut faciliter le partage des ressources et/ou la synchronisation au sein d'un cluster et permettre une réutilisation spatiale des fréquences radio pour minimiser les interférences [94]. Plus important encore, l'organisation d'un réseau apporte aussi plus de stabilité [93].

De nombreuses solutions de clustering ont été proposées [95, 92, 96, 97, 98, 99]. La majorité d'entre elles proposent l'utilisation d'une métrique qui permet aux noeuds de choisir un chef. Cette métrique peut être par exemple l'identifiant ou le degré des noeuds, une valeur de mobilité des noeuds ou encore une somme pondérée de tous ces éléments. D'autres solutions cherchent dans un premier temps à déterminer un ensemble dominant connecté sur lequel les clusters sont bâtis. Une grande partie des solutions de clustering construisent des clusters à 1 saut (dits 1-clusters), c.à.d des clusters où chaque noeud est à un saut de son chef de cluster. Les protocoles donnant naissance à des k-clusters (clusters où chaque noeud est à au plus k sauts de son cluster-head) sont plus récents. Dans ce chapitre, nous dressons un état de l'art qui permet de passer en revue les principaux types de solutions proposées dans la littérature pour organiser un réseau ad hoc en clusters.

4.2 Classification des méthodes de clustering

4.2.0.1 Clusters à 1 saut

De nombreux algorithmes de clustering produisent des clusters à 1 saut. L'un des algorithmes les plus anciens est l'algorithme du "plus petit ID" ou LCA (Lowest-Identifier Cluster Algorithm), proposé initialement par Ephremides, Wieselthier et Baker dans [95]. Chaque noeud se désigne ou non cluster-head en se basant sur son identifiant et celui de ses voisins. Un noeud peut avoir trois statuts différents : cluster-head, passerelle ou noeud ordinaire. À l'origine, tous ont un statut de noeud ordinaire. Si un noeud μ a le plus petit identifiant parmi les noeuds de son voisinage, il se déclare cluster-head. Sinon, il attend que tous ses voisins ayant un identifiant plus petit que le sien aient déclaré leur statut. Si au moins l'un d'eux s'est déclaré chef, μ déclare à son voisinage son statut de noeud ordinaire, μ appartient alors à chacun des clusters ayant comme chef un noeud voisin de μ . Si tous les voisins de μ ayant un identifiant plus petit que celui de μ se sont déclarés noeuds ordinaires (car ils se sont attachés à un autre de lien voisin de plus petit ID), μ se déclare cluster-head. Une fois que chaque noeud ait déclaré son statut de noeud ordinaire ou de cluster-head, si un noeud est voisin de plusieurs cluster-heads, il se déclare passerelle. Le protocole LCA est notamment utilisé par le protocole de routage CBRP (Cluster Based Routing Protocol) [92], pour la formation des clusters.

Par la suite, avec le protocole HCC (High Connectivity Clustering), Gerla et Tsai [96] ont cherché à apporter plus de stabilité à la structure de clusters formés par le LCA, en utilisant le degré des noeuds plutôt que leur identifiant. Le noeud ayant le plus fort degré dans son voisinage se déclare cluster-head. Si deux voisins ont le même degré, c'est celui de plus petit identifiant qui prend sa décision le premier. L'idée est que des noeuds de fort degré sont de bons candidats pour être cluster-heads car ils couvrent un grand nombre de noeuds et le nombre de clusters résultant en sera réduit. Par ailleurs, l'identifiant d'un noeud étant unique, un noeud de faible ID aura tendance à rester cluster-head longtemps, malgré la mobilité des noeuds. Néanmoins, si ce noeud est très mobile, il détruira constamment la structure. Ainsi, ces protocoles construisent des clusters à 1 saut qui se recouvrent (les

passerelles appartiennent à plusieurs clusters). Cette structure a été proposée pour acheminer les messages de contrôle et de routage où seuls les cluster-heads et les passerelles agissent. Leur maintenance s'avère coûteuse car le mouvement d'un noeud peut engendrer des réactions en chaîne et nécessiter une reconstruction totale de la structure. C'est pourquoi les auteurs de [97] ont proposé "Least Cluster Change" (LCC). LCC ajoute une étape de maintenance des clusters formés avec le LCA ou le HCC. Les clusters ne sont reconstruits que si deux cluster-heads se retrouvent voisins (le noeud de plus faible degré et/ou de plus fort ID suivant le cas abandonne le rôle de cluster-head) ou si un noeud ordinaire n'a plus aucun cluster-head dans son voisinage (il relance le processus de clustering). De cette façon, LCC améliore la stabilité de la structure. Cependant, les réactions en chaîne de reconstruction ne sont que limitées et ne sont pas complètement supprimées du fait qu'un seul noeud peut relancer la procédure de clustering s'il n'a plus aucun cluster-head dans son voisinage.

Le protocole MOBIC (Mobility Based Clustering)[98], autre protocole de clustering à 1 saut, applique le même algorithme que LCA et HCC mais utilise une métrique basée sur la mobilité plutôt que le degré ou l'identifiant des noeuds. Cette métrique cherche à caractériser la mobilité relative d'un noeud. L'idée est qu'un noeud peu mobile est un bon candidat pour être cluster-head car stable. Pour calculer sa mobilité relative, un noeud mesure le niveau de signal qui l'unit à chacun de ses voisins. La mobilité d'un noeud u est calculée à partir des rapports entre ce niveau de signal et celui mesuré à l'étape précédente pour chaque voisin de u , l'atténuation du signal étant dépendante de la distance séparant les noeuds. Le noeud dont la mobilité est la plus faible dans son voisinage devient clusterhead. Les auteurs de MOBIC utilisent l'algorithme LCC pour la maintenance de leur structure en ajoutant une règle supplémentaire : si deux cluster-heads u et v arrivent dans le voisinage l'un de l'autre, le cluster-head u de plus fort identifiant n'abandonne son rôle de cluster-head que si v fait toujours partie de ses voisins après une certaine période de temps. Cela permet de ne pas reconstruire la structure si deux cluster-heads ne se retrouvent voisins que pour une courte période. La mobilité des noeuds n'est plus reconsidérée par la suite à moins d'avoir à reconstruire toute la structure. Cependant, les inconvénients du LCC ne

sont pas éliminés. Bien que la prise en compte de la mobilité des noeuds semble intéressante pour déterminer les cluster-heads, cette méthode est un peu complexe et nécessite que les noeuds soient en mesure d'estimer les puissances de signal. De plus, elle ne considère pas certains phénomènes physiques qui provoquent des atténuations hétérogènes du signal.

Plutôt que d'utiliser l'identifiant ou le degré des noeuds, d'autres protocoles de clustering utilisent une somme pondérée de plusieurs métriques. Cette catégorie d'algorithmes vise à élire le cluster-head le plus adapté à une topologie pour une utilisation donnée. Par exemple, dans un réseau de capteurs où l'énergie est un facteur important, le paramètre d'énergie résiduelle peut obtenir un poids plus élevé dans la somme pondérée de la métrique résultante. WCA (Weighted Clustering Algorithm)[99] est un protocole utilisant une somme pondérée à quatre critères : la différence de degré D , la somme des distances avec les voisins P , la vitesse relative moyenne M et le temps de service en tant que cluster-head. Pour un noeud u , la différence de degré D est la différence entre le degré de u et une constante C représentant le nombre de noeuds qu'un cluster-head peut servir. Cependant, les auteurs n'explicitent pas le moyen de déterminer C . La mobilité relative M est obtenue comme dans MOBIC. Les distances P , entre u et ses voisins sont calculées à l'aide d'un GPS. L'élection se fait en se basant, là encore, sur l'algorithme de LCA, le noeud dont la somme pondérée de ces critères est la plus petite devient cluster-head. Les clusters sont ensuite maintenus sans plus reconsidérer la métrique pondérée. Le processus de clustering est relancé quand un noeud arrive dans une zone qui n'est couverte par aucun cluster-head, ceci pouvant entraîner des réactions de reconstruction en chaîne comme dans les algorithmes précédents.

Ainsi, plusieurs méthodes de clustering à 1 saut se basent sur l'algorithme du LCA et changent juste le critère de décision. C'est pourquoi Basagni, dans [100] reprend l'algorithme de LCA en donnant comme critère un poids générique que chacun définit comme il le souhaite. Il en étudie alors théoriquement les différentes propriétés.

Toutes les méthodes de clustering mentionnées jusqu'à maintenant produisent des clusters recouvrants, c'est à dire une structure dans laquelle un noeud peut appartenir à plusieurs clusters. Leur inconvénient majeur est que le mouvement d'un noeud peut provoquer

la reconstruction d'un cluster, qui, par réaction en chaîne, provoque la reconstruction de la structure entière. Afin d'éviter cela, d'autres protocoles de clustering ont été proposés, produisant des clusters non recouvrants où un noeud appartient à exactement un cluster. Dans [101], les auteurs proposent un protocole qui impose trois sauts entre deux cluster-heads. Le noeud ayant le plus fort degré dans son voisinage se déclare clusterhead. Ses voisins s'attachent à lui et se déclarent "noeuds membres". Les noeuds voisins de ces noeuds membres et non voisins d'un cluster-head se déclarent "non spécifié" et ne peuvent plus être clusterhead. Lorsque deux cluster-heads se retrouvent dans le voisinage l'un de l'autre, celui de plus grand identifiant abandonne son rôle de clusterhead et devient un noeud membre. Ses voisins deviennent soit membres (s'ils sont voisins du clusterhead) soit non spécifiés. Les réactions en chaîne de reconstruction sont ainsi évitées.

Dans "Adaptive Clustering" [94], les auteurs n'utilisent le statut de cluster-head que pour la formation des clusters. Une fois les clusters formés, la notion de cluster-head disparaît, chaque noeud du cluster tenant alors le même rôle. La motivation des auteurs est que les cluster-heads peuvent devenir des goulets d'étranglement par la suite, sources de perte de trafic et de saturation de bande passante. De plus, les cluster-heads seraient appelés à dépenser leur énergie plus vite que les autres noeuds. Pour construire de tels clusters, chaque noeud maintient un ensemble F qui initialement contient les identifiants de tous ses 1-voisins. Un noeud n'est autorisé à diffuser son statut (clusterhead, membre, non spécifié) que s'il possède un identifiant plus petit que les noeuds de F . Il ne se déclare cluster-head que s'il a un identifiant plus petit que tous les noeuds de son ensemble F . Sur réception du statut d'un noeud u , les voisins de u suppriment u de leur ensemble F . Si u a annoncé qu'il était cluster-head, ses voisins s'attachent à lui s'ils n'étaient encore membres d'aucun cluster ou si le cluster-head auquel ils s'étaient attachés avait un identifiant plus grand que u . Le processus s'arrête lorsque l'ensemble F de chaque noeud est vide. Comme le rôle de cluster-head disparaît une fois les clusters formés, la maintenance de la structure est un peu différente que dans les cas précédents. Chaque noeud doit connaître son voisinage à deux sauts. De cette façon, il sait si les membres de son cluster restent à deux sauts de lui. Si deux noeuds du même cluster se retrouvent éloignés de plus de deux sauts, seul

celui encore voisin du noeud de plus fort degré dans le cluster reste dans le cluster. L'autre doit se rattacher à un autre cluster. Bien que n'utilisant pas la notion de cluster-head, la maintenance de cet algorithme maintient le noeud de plus fort degré au centre du cluster, ce qui peut revenir au même que de l'élire comme cluster-head.

Le protocole de maintenance de "Adaptive Clustering" a ensuite été repris par les auteurs de [102] qui se proposent de l'appliquer au LCA. Tous les algorithmes décrits jusqu'à présent peuvent être qualifiés de "clustering actif", c'est à dire, des messages de contrôle sont envoyés dans le but de construire et de maintenir les clusters.

À l'opposé, les auteurs de [103] proposent un protocole de "clustering passif", c'est à dire, qu'ils n'utilisent aucun message dédié à la construction des clusters. Les clusters ne sont créés que lorsque c'est nécessaire, c'est à dire, lorsqu'un noeud a une information à diffuser. Le protocole de clustering passif utilise alors ces messages d'information pour construire les clusters, en ajoutant des champs aux paquets d'information. Un noeud a quatre statuts possibles cluster-head, passerelle, noeud ordinaire et non défini. Par défaut, le statut des noeuds est non défini. Seul un noeud ayant un statut non défini peut devenir cluster-head. Si un tel noeud a un message à envoyer, il se déclare cluster-head et diffuse son statut en l'ajoutant à l'information qu'il devait envoyer. Les noeuds voisins d'un cluster-head deviennent des noeuds ordinaires, les noeuds voisins de plusieurs cluster-heads deviennent des passerelles. Les noeuds ordinaires ne relaient pas les messages de diffusion. Aucun message n'étant dédié à la maintenance de la structure, les passerelles et les noeuds ordinaires activent des compteurs lorsqu'ils reçoivent des nouvelles de leur(s) cluster-head(s). S'ils restent sans nouvelle d'eux le temps que leur compteur expire, les noeuds ordinaires reprennent un statut non défini et les passerelles prennent le statut de noeud ordinaire ou non défini suivant le nombre de cluster-heads qu'elles entendent encore. Comme nous venons de le voir, il existe de nombreux protocoles de clustering à 1 saut. Les solutions les plus anciennes proposaient des clusters recouvrants. Ce type de clusters permet principalement de bâtir un ensemble dominant connecté sur le réseau (constitué des cluster-heads et des passerelles) pour pouvoir diffuser une information (principalement pour le routage) sur le réseau sans solliciter tous les noeuds. Puis d'autres études ont donné

des clusters non-recouvrants, plus robustes face à la mobilité des noeuds. Ce type de clusters permet également d'autres applications comme la réutilisation spatiale de fréquences ou de codes (les noeuds de deux clusters non voisins peuvent utiliser la même fréquence). Puis, des propositions plus récentes sont apparues permettant la construction de clusters à k sauts, encore plus robustes et permettant de nouvelles applications comme l'application de zones de services ou de protocole de routage hiérarchique.

4.2.0.2 Clusters à K sauts

La méthode la plus répandue pour la construction de clusters à k sauts est une extension des algorithmes de clustering à 1 saut. Par exemple, les auteurs de [104] généralisent l'algorithme de Lin et Gerla [94]. Leur algorithme suppose que chaque noeud connaît ses voisins situés jusqu'à k sauts de lui. Le noeud ayant le plus petit identifiant parmi les noeuds à au plus k sauts de lui, diffuse son statut de cluster-head à ses k -voisins. Lorsque tous les noeuds de son k -voisinage ayant un plus petit identifiant que lui ont diffusé leur décision d'être chef de cluster ou de s'attacher à un autre chef, le noeud u peut prendre sa propre décision de s'attacher au noeud de son k -voisinage de plus petit identifiant s'étant déclaré chef de cluster s'il existe, ou de créer son propre cluster sinon. De la même façon que pour les clusters à 1 saut, ce même algorithme est utilisé en utilisant différentes métriques. Dans le même papier [104], les auteurs proposent également d'utiliser le k -degré des noeuds (nombre de voisins à au plus k sauts) pour déterminer le cluster-head : le noeud de plus fort k -degré et de plus petit identifiant en cas d'égalité est promu chef de cluster. Les clusters résultants sont des k -clusters (chaque noeud est à au plus k sauts de son chef) recouvrants (un noeud peut appartenir à plusieurs clusters). Deux chefs sont éloignés d'au moins $k + 1$ sauts. Cependant, nous retrouvons les mêmes inconvénients que pour les algorithmes de clusters à 1 saut, à savoir qu'un petit changement de noeuds peut engendrer une reconstruction complète de la structure.

Les auteurs de [105] introduisent une métrique qu'ils appellent "associativité" qui se veut représenter la stabilité relative des noeuds dans leur voisinage. Pour chaque noeud, l'associativité comptabilise le temps que chacun des noeuds de son voisinage reste effec-

tivement dans son voisinage et en fait la somme sur chaque voisin. A chaque période de temps, un noeud u considère quels sont ses voisins actuels déjà présents lors de la période précédente et ajoute +1 à la valeur associée à chacun d'eux. Si un voisin a disparu, la valeur qui lui était associée passe à 0, si un autre apparaît, il prend la valeur 1. À chaque période de temps, l'associativité de u est la somme des valeurs associées à chacun de ses voisins. Cette valeur prend donc en compte la stabilité de u (si u est relativement stable dans son voisinage, il aura une forte associativité) et le degré des noeuds, cette valeur n'étant pas normalisée.

L'algorithme de formation des clusters est le suivant : Un noeud considère les noeuds de son k -voisinage ayant un degré supérieur à une valeur "seuil" et élit parmi eux celui ayant la plus forte associativité. Le plus fort degré et le plus faible identifiant sont ensuite utilisés pour élire un clusterhead en cas d'égalité des valeurs d'associativité. Les clusters résultants sont également des k -clusters recouvrants mais qui visent à être plus stables dans le temps et dans l'espace que ceux se basant sur le simple degré ou identifiant.

Dans [106], Lin et Chu proposent une approche basée sur aucune métrique particulière. Lorsqu'un noeud u arrive dans le réseau, il est en phase "d'initialisation". Il demande alors à ses voisins s'ils sont comme lui en phase d'initialisation ou s'ils ont un clusterhead et dans ce cas, à quelle distance ce cluster-head se situe-t-il. Si tous les voisins de u sont en phase d'initialisation, u s'élit chef de cluster et diffuse cette information. Tous les voisins de u qui n'ont aucun autre chef plus proche que u s'attache au cluster de u . Sinon, u s'attache au cluster de son voisin dont le chef est le plus proche et situé à, au plus, k sauts de lui. Si tous les cluster-heads des clusters de ses voisins sont à plus de k sauts de u , u se déclare chef de cluster et rallie à son cluster tous ses voisins à moins de k sauts dont le chef est plus éloigné que u . Si deux cluster-heads se retrouvent à moins de D sauts l'un de l'autre, $D \leq k$, le chef de cluster de plus faible identifiant doit céder son rôle et tous les membres de son cluster doivent se trouver un autre chef. Cette méthode de clustering est intéressante dans la mesure où elle produit des k -clusters non recouvrants où les chefs sont éloignés de au moins D sauts. Cela assure une certaine stabilité à la structure. Cependant, l'abandon du rôle de cluster-head par un noeud peut engendrer de fortes réactions en chaîne.

Une approche plus originale est celle proposée par Fernandess et Malkhi dans [107]. Leur algorithme se décompose en deux phases. La première étape consiste à trouver un arbre couvrant du réseau basé sur un ensemble dominant connecté de cardinalité minimale MCDS (Minimum Cardinality Dominating Set). Les auteurs proposent d'utiliser l'algorithme de [108] pour construire le MCDS mais précisent que n'importe quelle méthode peut être utilisée. La seconde phase de l'algorithme consiste en une partition de l'arbre couvrant en $2k$ -sous-arbres, un $2k$ -sous-arbre étant un arbre de diamètre au plus $2k$ sauts. Chaque sous-arbre consiste en un k -cluster. Cependant, une telle approche a une complexité temporelle et une complexité en messages en $O(n)$ (n étant le nombre de noeuds dans le réseau) et est par conséquent difficilement extensible. De plus, les auteurs n'abordent pas la maintenance d'une telle construction, qui ne semble pas triviale.

Les auteurs de Max-Min D-cluster [109] utilisent l'identifiant des noeuds pour construire des D-clusters non recouvrants où D est un paramètre de l'heuristique. Cependant, leur algorithme est un peu plus complexe que ceux vus jusqu'à maintenant. Il se décompose en trois phases. Lors de la première phase, chaque noeud collecte l'identifiant de ses voisins jusqu'à d sauts et en garde le plus grand qu'il diffuse de nouveau à D sauts lors de la seconde phase. Chaque noeud garde alors le plus petit des identifiants qu'il reçoit lors de cette deuxième phase (le plus petit parmi les plus grands). La troisième étape consiste au choix du cluster-head basé sur les identifiants collectés lors des deux phases précédentes. Si un noeud u a vu passer son propre identifiant lors de la deuxième phase, il devient chef de cluster. Sinon, si u a vu passer un identifiant durant chacune des phases 1 et 2, il élit le noeud portant cet identifiant comme chef. Sinon, u élit comme chef le noeud de plus grand identifiant dans son d voisinage. La structure résultante s'avère robuste, cependant la latence induite par l'algorithme est non négligeable.

Dans [110], les mêmes auteurs introduisent une notion d'identifiant virtuel. Le but est d'apporter une certaine équité entre les noeuds et d'éviter qu'un même noeud soit trop longtemps cluster-head et épuise ainsi ses ressources, tout en assurant qu'il le reste suffisamment longtemps pour apporter une stabilité à la structure. Les noeuds prennent le rôle de cluster-head tour à tour. Initialement, l'identifiant virtuel d'un noeud est égal à son

propre identifiant. À chaque période de temps, chaque noeud non cluster-head incrémente de 1 son identifiant virtuel jusqu'à atteindre un maximum MAXVID. Le noeud ayant l'identifiant virtuel le plus fort parmi ses k -voisins devient le cluster-head. En cas de conflits, c'est le noeud qui a le moins opéré en tant que chef qui devient cluster-head (et de plus fort identifiant initial). Un noeud qui devient cluster-head ajoute à son ancienne valeur d'identifiant virtuelle MAXVID de façon à assurer qu'il conserve le plus fort identifiant virtuel et reste cluster-head. Un noeud reste cluster-head pendant une période de temps $\Delta(t)$ au bout de laquelle il passe son identifiant virtuel à 0 et abandonne son rôle de chef. Lorsque deux chefs entrent dans le voisinage l'un de l'autre, celui de plus faible identifiant virtuel abandonne son rôle. Dans le même papier, les auteurs proposent également une construction où l'identifiant virtuel de base serait le degré des noeuds. Cet algorithme permet donc la formation de k -clusters en assurant une certaine stabilité de la structure. Néanmoins, elle nécessite une synchronisation des noeuds afin que chacun se base sur la même période de temps pour incrémenter son identifiant virtuel et surtout pour comptabiliser la période durant laquelle il est cluster-head. Or, une synchronisation dans de tels réseaux est non triviale et nécessite beaucoup de messages.

Les auteurs de [Krishna] proposent un autre type d'algorithme, formant cette fois des clusters sans chef de cluster. Pour cela, chaque noeud nécessite également la connaissance de son k -voisinage. Un cluster est formé par un ensemble de noeuds tel qu'il existe entre deux noeuds de cet ensemble un chemin d'au plus k -sauts. Si $k = 1$, chaque cluster est une clique. Un noeud appartenant à plusieurs clusters est dit noeud frontière. Les clusters sont donc recouvrants. Malheureusement, cet algorithme implique beaucoup de messages de contrôle, de maintenance et de données à gérer par les noeuds.

Les auteurs de DDR (Distributed Dynamic Routing) [111] proposent également une structure sans cluster-head. Contrairement à la plupart des algorithmes de formation de k -clusters, les noeuds ne nécessitent que de la connaissance de leur 1-voisinage. La formation des clusters se base sur la construction d'un arbre. Chaque noeud choisit comme père son voisin de plus faible identifiant. Il existe alors exactement une arête sortante par noeud. Cela conduit à la formation d'un arbre. Tous les noeuds du même arbre appartiennent au même

cluster. Le diamètre de tels clusters n'est pas fixé a priori et s'adapte automatiquement à la topologie sous-jacente. Cet algorithme a été ensuite repris par Baccelli [Baccelli] en y ajoutant la notion de cluster-head et en contrôlant la taille des clusters. Pour cela, un noeud a le droit de se choisir comme père s'il a le plus fort identifiant dans son 1-voisinage. Il existe alors des noeuds sans arête sortante qui deviennent des cluster-heads. Ces clusterheads ont alors la possibilité de borner la hauteur des arbres à d sauts en diffusant l'information le long des branches de l'arbre. Si la branche est trop longue, le noeud se trouvant à $d + 1$ sauts de son cluster-head doit s'attacher à un autre père (et donc casser la branche).

D'autres algorithmes comme ceux proposés dans [112, 113] ne proposent qu'une solution de maintenance. Par exemple, les auteurs de [113] proposent de maintenir un certain nombre de noeuds dans un cluster, qui dépendrait du nombre d'entités que le clusterhead est en mesure de gérer. L'idée est de maintenir en permanence le nombre de noeuds entre deux seuils. Si un cluster est trop petit, le chef de cluster doit élire parmi ses clusters voisins celui le plus adapté pour une fusion, c'est-à-dire celui dont le nombre de noeuds permet la fusion des deux clusters. Si aucun ne correspond, le chef de cluster doit déterminer un cluster qui peut lui céder des entités pour un meilleur équilibre du nombre de noeuds. Si les clusters sont trop gros, le chef doit élire parmi ses membres un autre cluster-head et scinder son cluster en deux. Il reste cluster-head d'un cluster résultant tandis que le noeud qu'il a élu devient chef du second cluster. Cette méthode est cependant très coûteuse en calculs, latence et messages et supporte mal le passage à l'échelle du réseau.

4.2.0.3 Clusters hiérarchiques

Il existe également des propositions de structures hiérarchiques à plusieurs niveaux, c'est-à-dire où les clusters sont ensuite regroupés en d'autres clusters de niveaux supérieurs et ainsi de suite. Bien que la majorité des algorithmes vus jusqu'à maintenant peut être appliquée récursivement sur les clusters pour former des clusters de niveau supérieur, ils n'ont pas été écrits dans ce but contrairement aux exemples que nous énonçons ici.

Dans [114], Banerjee et Khuller se basent sur un arbre couvrant, construit grâce à un parcours en largeur, pour la construction de k -clusters. Les clusters sont formés par

branche, en fusionnant récursivement deux sous-arbres de l'arbre couvrant jusqu'à obtenir une taille correcte. Le processus est alors réitéré jusqu'à obtenir un certain nombre de niveaux.

Dans [115], les auteurs cherchent à combiner les partitions physiques et logiques des noeuds ainsi que leur mobilité. Pour cela, ils utilisent un GPS. Les auteurs supposent que les noeuds répondent à un modèle de mobilité de groupe. L'algorithme consiste ensuite à regrouper en un même cluster les noeuds proches géographiquement et qui se déplacent à une vitesse semblable dans une même direction. Le processus est ensuite réitéré jusqu'à obtenir le nombre de niveaux voulus.

La structure de cellules hiérarchiques de SAFARI [116] est basée sur une auto-sélection des noeuds en tant que cluster-heads. Le nombre de niveaux hiérarchiques s'établit automatiquement en fonction de la topologie sous-jacente des noeuds. Les clusters de niveau i sont groupés en clusters de niveau $(i + 1)$ et ainsi de suite, les simples noeuds étant considérés comme des cellules de niveau 0. Chaque cluster-head de niveau i se choisit un cluster-head de niveau $(i + 1)$. Tous les cluster-heads de niveau i ayant choisi le même cluster-head de niveau $(i + 1)$ appartiennent au même cluster de niveau $(i + 1)$. Un cluster-head u de niveau i décide de monter ou descendre son niveau en fonction du nombre de cluster-heads de niveau $(i + 1)$ et $(i - 1)$ qui existent à une certaine distance. S'il n'existe aucun cluster-head de niveau supérieur à une distance plus petite que D_i (D_i constante dépendant du niveau i du cluster-head) de u , u décide d'augmenter son niveau. Si deux cluster-heads de même niveau sont à moins de $h \times D_i$ ($0 < h < 1$) sauts, le cluster-head de plus grand identifiant descend son niveau. Un cluster-head de niveau i est également cluster-head de tout niveau j tel $0 < j < i$. Cet algorithme construit des k -clusters hiérarchiques, où k dépend du niveau du noeud i : $k = D_i$. D_1 doit être fixé. À partir de là, D_i dépendant de D_{i-1} , le rayon des clusters de chaque niveau est fixé.

Conclusion

Une auto-organisation consiste en la création d'un comportement émergent, structurant le réseau, et permettant par ce biais d'en faciliter l'exploitation. Nous avons présenté un recueil des approches de clusterisation existantes, avec une tentative de classification de ces dernières en se basant sur la notion de distance. Dans le chapitre suivant nous allons présenter la structure de clustering que nous avons proposée et exploitée pour notre problématique d'équilibrage de charge.

Chapitre 5

Présentation de la solution : Équilibrage de charges basé sur le clustering

Introduction

Ce chapitre présente la méthodologie que nous avons développée et adoptée. Nous présentons les mécanismes de notre solution qui vise à équilibrer les charges et la consommation énergétique dans un réseau ad hoc. Nous nous intéressons à deux charges importantes : la charge de communication et la charge de calcul.

La première partie de ce chapitre est consacrée à la présentation de la structure d'auto-organisation choisie. Nous allons dans un premier temps donner un aperçu général de notre solution de clustering. Ensuite, nous présentons dans la section 5.1.2 l'algorithme de construction, puis la section 5.1.3 détaillera la méthode de maintenance des clusters formés. Tous les processus chargés de construire et de maintenir cette hiérarchie partagent des informations communes afin de réduire le trafic de contrôle induit.

La deuxième partie analyse plus en détail l'algorithme d'équilibrage de charges proposé en précisant notamment ses propriétés.

5.1 Phase de clustering

Obtenir de meilleures performances avec les réseaux ad hoc résulte d'une auto-organisation efficace de ces réseaux. Ainsi, la proposition d'une technique d'auto-organisation s'avère nécessaire pour que ces réseaux puissent accomplir leurs missions sans que leurs performances soient dégradées. En plus, pour que cette technique d'auto-organisation soit réaliste, elle doit tenir compte des spécificités des réseaux ad hoc et de leurs contraintes imposées. L'architecture basée sur le clustering est considérée comme une approche prometteuse, elle consiste à partitionner virtuellement le réseau en groupes appelés clusters.

Afin de permettre une extensibilité totale et ne pas avoir à reconstruire les clusters après chaque mouvement individuel d'un noeud, nous avons cherché à construire des clusters qui n'ont aucun paramètre fixé à l'avance. De plus, le clustering doit d'être distribué tout en minimisant le nombre d'informations à échanger. Les clusters formés doivent être stables (les clusterheads doivent conserver ce statut suffisamment longtemps pour limiter le trafic de contrôle nécessaire à la reconstruction des clusters) tout en s'adaptant aux changements de la topologie sous-jacente.

Dans cette partie, nous présentons tout d'abord la métrique utilisée pour la formation des clusters, puis nous exposons les deux phases principales de la technique de clustering : la formation des clusters et la maintenance de ces derniers. La phase de formation des clusters est déclenchée à l'initialisation du réseau.

5.1.1 La métrique utilisée

Etant donné l'intérêt porté au concept de clustering et ses apports indéniables pour améliorer les performances d'un réseau ad hoc, le choix du mécanisme de clustering s'avère important. De ce fait, un algorithme de clustering doit en premier lieu être capable de sélectionner les noeuds appropriés pour assurer les fonctionnalités du clusterhead.

Comme mentionné dans le chapitre 4, différentes métriques ont été utilisées pour le choix des clusterheads. L'identifiant des noeuds étant immuable, il permet de conserver les clusterheads (chefs de cluster) très longtemps. Cependant de tels clusters sont indépendants

de la topologie sous-jacente et ne sont pas toujours adaptés. Le degré des noeuds s'avère l'une des métriques les plus adaptées, l'idée étant qu'un clusterhead de fort degré permet de couvrir un grand nombre de noeuds, ce qui permet d'en minimiser le nombre. Cependant, un mouvement individuel d'un noeud dans des clusters basés sur cette métrique peut conduire à une réorganisation complète du réseau, alors que la structure globale du réseau reste inchangée. Basés sur cette constatation, nous avons introduit une nouvelle métrique, que nous avons appelée Poids P

$$P = \frac{\text{Energie}}{\text{Mobilité}} \quad (5.1)$$

Un clusterhead est élu pour sa capacité énergétique relativement élevée et sa faible mobilité. En effet, des candidats faiblement mobiles, avec une autonomie en énergie élevée sont aptes à assurer des tâches additionnelles. Un tel choix pourrait refléter l'hétérogénéité naturelle des réseaux ad hoc et offrir une infrastructure virtuelle exploitant au mieux les ressources disponibles dans un tel environnement hétérogène, sans pénaliser les noeuds défavorisés. L'idée est que l'énergie est une ressource critique dans les réseaux ad hoc. Un clusterhead consomme plus d'énergie qu'un noeud ordinaire puisqu'il a d'autres fonctionnalités : coordination entre ses membres, maintenance du cluster et équilibrage de charges. La mobilité est également un facteur important pour l'élection d'un clusterhead. En effet, afin d'éviter les changements fréquents de clusterheads, il est important de choisir celui ayant une faible mobilité. Si le clusterhead se déplace rapidement, les noeuds peuvent être détachés de lui induisant des réaffiliations qui causent des échanges significatifs de messages de mise à jour pouvant dégrader les performances du réseau ou affecter les performances d'autres algorithmes reposant sur cette structure de clustering.

5.1.2 Construction des clusters

La construction des clusters s'appuie sur la connaissance du voisinage. Chaque noeud surveille son voisinage en envoyant périodiquement un paquet hello. Puisque le médium

radio est de type diffusion (broadcast), tous ses voisins le recevront. Périodiquement, chaque noeud calcule son poids et le diffuse localement à son 1-voisinage (voisins immédiats). Chacun est alors en mesure de comparer son propre poids à celui de ses voisins. À partir de là, un noeud décide soit de s'élire comme clusterhead (s'il possède le plus fort poids), soit de choisir comme père (clusterhead) son voisin de plus fort poids. En cas d'égalité, afin de privilégier la stabilité de la structure, le noeud choisi sera celui déjà élu au tour précédent s'il est en course, sinon celui de plus forte énergie. De cette façon, deux voisins ne peuvent pas être tous deux clusterheads.

Les caractéristiques suivantes sont considérées dans la procédure de formation des clusters :

- Chaque noeud maintient son statut (clusterhead ou noeud ordinaire). Un noeud non clusterhead connaît le cluster auquel il appartient et le clusterhead correspondant.
- La mobilité des noeuds cause des changements de la topologie du réseau. Si à un moment donné, un noeud est détaché de son cluster courant et attaché à l'autre, les clusterheads correspondants mettront à jour les tables de leurs membres.
- Un clusterhead garde l'information de ses membres (identifiant ID, statut, charge, énergie, puissance de calcul). Il peut détecter si un autre clusterhead est rentré dans son cluster. Dans ces conditions, l'un d'entre eux est contraint de renoncer à son rôle de clusterhead. Dans notre cas, c'est celui ayant moins d'énergie.
- En raison des fonctionnalités additionnelles pour lesquelles le clusterhead est prévu, son énergie est susceptible d'être épuisée. Un seuil minimum d'énergie est défini pour chaque clusterhead. S'il est atteint, la procédure d'élection de clusterheads est invoquée.
- Dans le cas où un cluster-head migre vers d'autres clusters, le processus de réélection du nouveau clusterhead concerne seulement les clusters ayant perdu leurs clusterheads et le futur clusterhead est choisi parmi les membres du cluster. Nous avons adopté cette solution pour ne pas affaiblir les performances de cette technique et éviter les réactions en chaîne qui peuvent se produire pendant le déclenchement du processus de clustering.

Algorithme 1 : Formation des clusters

Pour tout noeud $u \in V$ (V est l'ensemble des noeuds)

- Scrutation du voisinage

- Calcul de la valeur du Poids P
- Diffusion locale de la valeur de son poids P à ses 1-voisins.
- Après réception des poids de tous ses voisins le noeud u peut soit s'élire lui même cluster-head ou choisir son clusterhead

Fin

5.1.3 Maintenance des clusters

En raison de la nature dynamique du système considéré, les noeuds aussi bien que les clusterheads tendent à se déplacer dans différentes directions. Ainsi, il est primordial de maintenir continuellement la structure virtuelle. La construction est exécutée à l'initialisation du réseau tandis que la maintenance doit être continuellement active, tout en générant un trafic de contrôle réduit. La mise à jour (la maintenance) peut avoir comme conséquence la formation de nouveaux clusters ou le changement de point d'attachement des noeuds d'un clusterhead à un autre dans le même ensemble de clusters existant. Dans ce cas on parle de réaffiliation. Tous les noeuds surveillent continuellement leur puissance de signal reçu du clusterhead. Quand la distance entre un noeud et son clusterhead augmente, le signal reçu diminue. Dans ce cas, le mobile doit se détacher du clusterhead courant et s'attacher à un autre clusterhead et met à jour l'identificateur de son clusterhead (CID : Clusterhead Identifier) et les deux clusterheads correspondants mettent à jour leurs listes de membres. Si le noeud entre dans une région et ne trouve aucun autre cluster pour s'y attacher, alors la procédure de formation des clusters est invoquée.

5.2 Phase d'Equilibrage de charges

Une bonne utilisation d'un réseau ad hoc permet d'augmenter potentiellement la puissance et la durée de vie des noeuds. Ainsi, chaque noeud permet de profiter de sa puissance et de son énergie sans affecter ses propres travaux.

L'objectif recherché dans la répartition des charges étant d'accélérer la vitesse d'exécution et minimiser la consommation énergétique, nous avons constamment à vérifier deux exi-

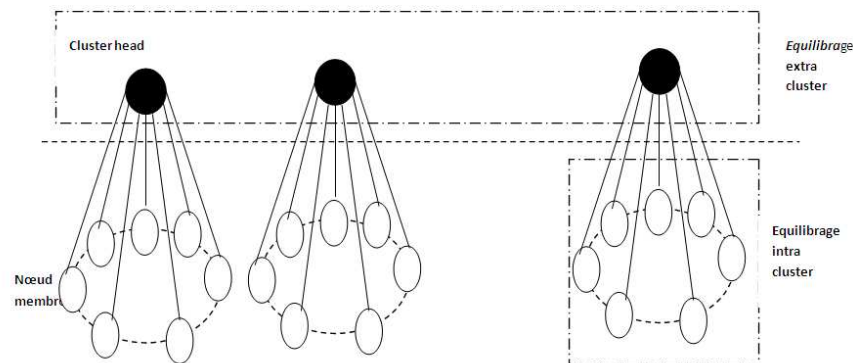


FIG. 5.1 – Le Modèle d'équilibrage proposé

gences souvent contradictoires : minimiser la communication et équilibrer au mieux la charge de travail des noeuds.

Dans le domaine de l'équilibrage de charges, il existe deux grandes familles d'algorithmes : centralisés et décentralisés. Pour un algorithme centralisé, les informations sur la charge des différentes machines seront envoyées sur une machine centrale qui va se charger de la répartition et du déplacement des tâches. Pour un algorithme décentralisé, la décision se fait en local sur chacune des machines ; celles-ci ne connaissent que les caractéristiques de leurs voisins directs. Les algorithmes utilisés sont itératifs et l'équilibrage de charges s'effectue de proche en proche dans le temps. Il s'agit d'un équilibrage de charges dynamique dans lequel le déplacement de charge se fait au cours de l'évolution des travaux sur les noeuds.

Dans ce travail, nous avons opté pour un algorithme d'équilibrage de charges dynamique et centralisé (au niveau cluster) et décentralisé (entre les clusters).

5.2.1 Modèle de représentation proposée

Pour représenter un réseau ad hoc nous proposons de le modéliser sous forme d'un ensemble de clusters. Tel qu'illustré par la figure 5.1, cette structure hiérarchique peut être représentée par la configuration C/M où C représente le cluster et M le nombre total de membres d'un cluster.

1) **Le niveau cluster** correspond à un clusterhead ayant pour fonction de :

- Gérer l'information de charge relative aux nœuds membres du cluster.
- Maintenir l'état de charge du cluster.
- Maintenir l'état d'énergie du cluster.
- Décider de déclencher un équilibrage local, que nous appellerons équilibrage intra cluster.
- Informer les membres, pour mettre en oeuvre l'équilibrage décidé par le clusterhead.
- Décider d'un équilibrage global au niveau des autres clusters, que nous appellons équilibrage extra-cluster.

2) **Le niveau nœud du cluster**, où chaque nœud correspond à un membre ayant pour fonction de :

- Maintenir à jour l'information de l'état de sa charge.
- Maintenir à jour l'information de l'état de son énergie.
- Envoyer périodiquement cette information à son clusterhead.
- Exécuter les opérations d'équilibrage décidées par son clusterhead.

5.2.2 Stratégie d'équilibrage de charges

5.2.2.1 Principe

La structure hiérarchique du modèle proposé nous permet de développer une stratégie hiérarchique à deux niveaux d'équilibrage : Intra-cluster, Extra-cluster.

1) **Équilibrage Intra-cluster** : Dans ce premier niveau, chaque clusterhead décide de déclencher une opération d'équilibrage à la demande d'un membre du cluster et en fonction de la charge courante du cluster qu'il gère. Cette charge est estimée à partir des différentes informations de charge envoyées périodiquement par les membres qui composent le cluster. Le clusterhead tente, en priorité, d'équilibrer la charge du cluster localement en l'envoyant vers un nœud choisi parmi les nœuds qui composent le cluster. Cette approche de localité a pour objectif de réduire les coûts de communication, en évitant les communications extra cluster.

2) **Équilibrage Extra-cluster** : Dans ce deuxième niveau, l'équilibrage se fait à l'échelle des extra clusters. Il intervient dans le cas où certains clusterheads n'ont pas

réussi à équilibrer localement leurs charges. Il y aura ainsi transfert de tâches entre clusters surchargés et clusters sous-chargés. Dans un souci de réduire au maximum les coûts de communication, les clusters récepteurs seront sélectionnés en fonction du voisinage.

5.2.2.2 Description générique de la stratégie d'équilibrage

A n'importe quel niveau du modèle défini ci-dessus, nous proposons une stratégie d'équilibrage composée de trois étapes.

a- Estimation de la charge

Cette étape définit les mécanismes de mesure et de communication de charges. Connaissant le nombre de tâches (nombres d'instructions), sa puissance de calcul ainsi que son niveau d'énergie, chaque noeud estime ses capacités en effectuant les actions suivantes :

- Estimation de sa charge courante.
- Estimation de son énergie courante.
- Envoi de l'information de charge et d'énergie à son clusterhead associé.
- Comparaison des valeurs aux deux seuils : la charge maximum (maxcharge) qu'un noeud peut effectuer et l'énergie minimum, une fois atteinte, le noeud sait qu'il va mourir bientôt, il décide alors de transférer sa charge à un autre noeud.
- Envoi d'un message de décharge au clusterhead en cas de dépassement d'un ou des deux seuils prédéfinis.

Remarque : Afin de prendre en considération l'hétérogénéité des noeuds, nous proposons comme indice de charge, le temps d'exécution d'un noeud qui est le rapport entre la charge (LOAD exprimée en nombre d'unités de calcul) et la puissance (SPD exprimée en nombre d'unités de calcul exécutées par unité de temps) de ce noeud.

b- Prise de décision

Durant cette étape, le clusterhead décide de l'opportunité de déclencher un équilibrage de charges local (intra cluster). Pour cela, il exécute les actions suivantes :

- Définition de l'état de charge d'un cluster : Sachant que l'écart type mesure la variation moyenne entre le temps d'exécution des membres et celui de leur cluster associé, nous pouvons dire qu'un cluster est en état d'équilibre lorsque cet écart est relativement faible.

Cela signifie que le temps d'exécution de chaque membre converge vers le temps d'exécution de son cluster.

- Partitionnement du cluster : Pour déterminer si un élément d'un cluster est dans un état approprié pour participer à un transfert de tâches comme receveur, nous définissons deux classes d'éléments : les éléments surchargés (sources) et les éléments sous-chargés (receveurs). Cette classification dépend de la charge et du niveau d'énergie de chaque élément.

c- Transfert de tâches

Pour réaliser une opération d'équilibrage de charges, nous proposons l'heuristique suivante :

a) **Calculer la demande**, i.e., la charge à transférer par chacun des noeuds sources.

La demande d'un noeud source (N_s) correspond à la charge qu'il souhaite faire transférer.

Deux cas peuvent être distingués pour le calcul de la demande :

Cas de surcharge : si la charge actuelle du noeud est inférieure à seuilcharge, la charge demandée correspond à la charge X à transférer pour que son temps d'exécution T_{Execs} converge vers le temps d'exécution $T_{ExecCluster}$ du cluster auquel il appartient.

Il s'agit de faire converger le temps d'exécution du noeud source vers le temps d'exécution moyen du cluster.

$$T_{Execs} = \frac{Loads - X}{SPDs} \simeq \frac{Loadc}{SPDc} \quad (5.2)$$

donc :

$$X \simeq Loads - \frac{LoadcSPDs}{SPDc} \quad (5.3)$$

où

T_{Execs} : Temps d'Exécution du noeud source

$Loads$: Charge du noeud source

X : Charge à transférer par le noeud source (la demande)

$Loadc$: Charge moyenne du cluster

SPDs : Puissance du noeud source

SPDc : Puissance moyenne du cluster

Cas de manque d'énergie : si l'énergie actuelle du nœud est inférieure à seuil'énergie, la charge demandée correspond à la charge à transférer pour que le nœud n'épuise pas complètement son énergie et survive encore.

$$X = Loadactuelle - \frac{Energieactuelle}{Q}. \quad (5.4)$$

où

Q : quantité d'énergie nécessaire pour exécuter une instruction.

Loadactuelle : Charge actuelle du noeud source

Energieactuelle : Energie actuelle du noeud source

b) Calculer la disponibilité en terme de capacité de calcul et d'énergie qui correspond à la charge offerte par chacun des noeuds receveurs. Par un raisonnement analogue, L'offre d'un noeud receveur correspond à la charge Y qu'il accepte de recevoir pour que son temps d'exécution TExecr converge vers le temps d'exécution TExecCluster du cluster auquel il appartient en ayant une énergie suffisante pour traiter la charge locale plus la charge supplémentaire.

$$Y = \frac{Loadc.SPDr}{SPDc} - Loadr \quad (5.5)$$

où

Loadr : Charge du noeud receveur

Y : Charge à recevoir (acceptée) par le noeud receveur (l'offre)

Loadc : Charge moyenne du cluster

SPDr : Puissance du noeud receveur

SPDc : Puissance moyenne du cluster

c) Si l'offre n'est pas en mesure de satisfaire suffisamment la demande, il n'est pas recommandé d'entamer un équilibrage local. Un équilibrage extra-cluster peut être décidé et un transfert de charge est effectué en tenant compte des coûts de communication.

Remarque : Pour notre travail le mécanisme de transfert des charges (migration) sera considéré comme un aspect technique supporté par le noyau d'exécution.

5.3 Algorithmes d'équilibrage

1 - Algorithme d'équilibrage Intra-cluster : Cet algorithme qui utilise une approche source-initiative constitue le noyau de notre stratégie. L'approche de localité adoptée fait que c'est le niveau d'équilibrage qui sera le plus fréquemment sollicité. Il est déclenché lorsqu'un noeud constate qu'il est surchargé ($\text{charge} > \text{seuilcharge}$ ou $\text{energie} < \text{seuilenergie}$). Il envoie un message de déchargement à son clusterhead. Sur la base des informations de charge et d'énergie que le clusterhead reçoit périodiquement à partir de chaque membre de son cluster, il analyse cette demande. En fonction du résultat de cette analyse, soit il décide de déclencher un équilibrage local, soit il décide de chercher un clusterhead voisin et de l'informer sur la charge à transférer.

2 - Algorithme d'équilibrage Extra-cluster : Cet algorithme est exécuté uniquement lorsque certains clusterheads n'ont pas réussi à équilibrer localement leur charge pour cause d'offre insuffisante. Dans ce cas, le clusterhead tente d'équilibrer la charge à travers les clusterheads voisins. L'équilibrage extra-clusters devra tenir compte des coûts de communication. Durant le transfert de tâches, nous choisirons comme cluster récepteur, celui qui nécessite le plus petit coût de transfert. Ainsi, le critère de sélection sera pondéré par le coût de communication (coût de transfert de tâches) pour assurer qu'une tâche ne peut être transférée que lorsque son temps de réponse estimé dans le cluster récepteur, auquel nous rajoutons le temps de transfert à partir du cluster source, est meilleur que son temps de réponse dans le cluster source.

Dans ce qui suit, nous allons décrire l'algorithme générique associé à notre stratégie.

Algorithme d'Equilibrage de charges (au niveau d'un Cluster C)

Estimation de la charge du cluster

1. Collecte périodique de l'information de charge

Pour chaque membre M_i du Cluster C_j faire

Chapitre 5. Présentation de la solution : Equilibrage de charges basé sur le clustering67

Envoi de sa charge actuelle $LOAD_i$ à son clusterhead associé.

Envoi de son niveau d'énergie $ENRG_i$ à son clusterhead associé.

Vérifie : si $(LOAD_i > \text{seuilcharge})$ ou $(ENRG_i < \text{seuilenergie})$ alors

Envoi d'une requête de décharge au clusterhead.

Fin Pour

2. Le clusterhead du Cluster C_j effectue les opérations suivantes :

a- Calculer la charge courante $LOAD_c$ et le temps d'exécution $T_{ExecCluster}$ de C_j ;

b- Calculer l'écart type sur les temps d'exécution des éléments de C_j ;

Prise de décision

3. Partitionnement des nœuds de C_j en surchargés (NS : Nœuds Sources) et sous-chargés (NR : Nœuds Receveurs).

- Trier l'ensemble NR par ordre croissant de leurs charges (en tenant compte de la puissance) ;

Transfert de tâches

4. Calcul de la demande X

5. Transfert de tâches :

Tant que ($NR \neq \emptyset$) faire

- Choisir le premier (le moins chargé) élément de NR

- Calcul de l'offre Y

- Si Offre > Demande

-Vérifier si le nœud receveur a une énergie suffisante

- Si les deux conditions sont vérifiées alors

(i) transfert de charge

(ii) Mettre à jour les charges des nœuds source et receveur.

(iii) Mettre à jour les ensembles NS, NR.

(iiii) Trier NR par ordre croissant.

Sinon aller au nœud suivant

Fin Tant que

Si aucun nœud receveur n'est trouvé alors Equilibrage Extra-Cluster

Transfert de tâches extra cluster

La procédure de transfert de charges extra cluster reste la même que le transfert intra cluster sauf qu'il faut tenir compte des coûts de transfert

Conclusion

Dans ce chapitre nous avons présenté notre solution d'équilibrage de charges basée sur un algorithme de clustering. Nous avons visé avec la création d'une topologie virtuelle stable la minimisation des surcoûts engendrés par les communications et les transferts de charges.

Pour le clustering, nous avons impliqué des métriques de stabilité pour élire les clusterheads : la mobilité et l'énergie afin d'éviter la réélection fréquente et éviter la restructuration globale de tout le réseau.

Cette solution devra améliorer les performances d'un réseau ad hoc. L'évaluation de la solution a été réalisée par simulation. Le chapitre suivant présente les détails de l'évaluation.

Chapitre 6

Simulation et évaluation

Introduction

Il existe différentes techniques d'évaluation de performances. Parmi elles, on peut citer la modélisation analytique, la simulation, et les mesures obtenues à partir d'un système réel.

La modélisation analytique consiste à représenter les conditions réelles de façon formelle à l'aide d'outils mathématiques.

La simulation consiste à modéliser la globalité du système étudié et à le simuler numériquement à l'aide d'événements provenant de mesures sur un système réel ou de modèles probabilistes. L'intérêt de la simulation est de pouvoir travailler sur des systèmes non disponibles. Par exemple, lors de l'étape de conception, il est beaucoup moins coûteux de réaliser une simulation préalable des alternatives envisagées. De plus, la simulation est un moyen très souple d'étudier un problème. Cette technique permet des réexecutions de programmes avec changement de paramètres et une prise de trace d'exécution sans perturbation. Enfin elle permet, puisque le temps est modélisé, d'arrêter la simulation pour revenir à un état antérieur. On peut enfin évaluer le système à partir de mesures réelles. Nous présenterons dans ce chapitre l'architecture de notre simulateur, le modèle de mobilité utilisé ainsi que les résultats commentés des simulations effectuées.

6.1 Description du simulateur

Afin de tester le comportement du modèle et d'évaluer les performances de la stratégie proposée, nous avons utilisé un simulateur à événements discrets que nous avons développé et implémenté en Java modélisant un réseau ad hoc utilisant notre solution. Utiliser ce simulateur plutôt qu'un simulateur réseau qui prend en compte les caractéristiques des protocoles de niveaux inférieurs nous permet de focaliser notre étude sur le comportement de notre stratégie d'équilibrage de charges uniquement, sans tenir compte des aléas des protocoles utilisés au niveau des autres couches.

Le principe du simulateur est le suivant :

- Au lancement, le simulateur crée un ensemble variable (paramétrable) de nœuds mobiles et les distribue sur une surface géographique définie (paramétrable).
- Durant toutes les expérimentations conduites, on assume que la portée de transmission maximale est la même pour tous les nœuds du réseau, cette portée étant égale à 250m.
- Pour chaque nœud, le simulateur lui affecte un identifiant, une énergie initiale, une charge, une puissance, une vitesse de mobilité ainsi que d'autres informations concernant son statut.
- Nous avons considéré une distribution d'un nombre variable de tâches d'une manière aléatoire aux différents nœuds.
- Pour la consommation énergétique, nous nous sommes appuyés sur des travaux qui ont le mérite de proposer des modèles de consommation d'énergie validés par des expérimentations réelles. Margi et al. [117] ont mesuré la consommation énergétique des traitements CPU de différentes tâches sur un ordinateur portable sans fil de type DELL Latitude C600. Nous avons utilisé leurs résultats dans nos simulations.
- Nous soulignons que toutes les tâches ne sont pas "exportables" c-à-d transférées entre les nœuds. Le cas des tâches de routage, qui dépendent de la topologie du réseau à un instant donné, est le bon exemple.
- Les nœuds se déplacent sur la surface de simulation selon le modèle Random Way Point dans lequel un nœud choisit aléatoirement une vitesse et une destination. Il se rend à cette

destination avec la vitesse choisie. Une fois à cette destination, il choisit un temps de pause aléatoire qui varie de 0 à Tempsmax. Puis il choisit une nouvelle destination et vitesse. Un tel modèle présente des avantages certains en terme de simplicité d'implémentation.

Durant la simulation, des événements sont générés et traités par le simulateur en fonction de leurs temps d'arrivée (déplacement, Consommation de Charge, Communication de Charge, le rattachement à un cluster, le détachement à un cluster...etc). Indépendamment des autres événements, des demandes d'équilibrage de charges peuvent être émise par ces derniers. Le simulateur crée des évènements de décharge et les insère dans le gestionnaire d'évènements. Ce processus est répété jusqu'à la fin de la simulation.

Notre simulateur est architecturé autour d'un gestionnaire d'évènements qui contient la liste de tous les événements qui sont classés selon l'ordre chronologique de leur arrivée.

Une session de simulation consiste à retirer un événement du gestionnaire, traiter l'événement et enregistrer dans le gestionnaire les évènements engendrés.

Le simulateur est composé des classes suivantes :

La classe Nœud : Cette classe modélise un nœud mobile. Elle contient les informations telles que l'Id du mobile (Id nœud), sa charge actuelle (Charge), son énergie (EnergieIni), sa puissance de calcul (Puissance), son statut (Statut) et l'identificateur de son cluster head (IdHead) s'il est un nœud ordinaire.

La classe Voisins : Elle regroupe tous les voisins d'un nœud. (Idvoisin, Chargevoisin, Energievoisin, Mobilityvoisin, puissancevoisin).

La classe Evenement : Elle contient les informations concernant le temps d'occurrence de l'événement (TempOccure) et son type (type).

La classe Scheduler : c'est le gestionnaire d'évènements. Elle contient les méthodes nécessaires à la création de nouveaux événements (AjouterEvenement) et le retrait des événements (RetirerEvenement).

Comme nous l'avons vu précédemment (chapitre 5), le régulateur de charge est composé de deux principaux modules :

Le collecteur de charge : correspond à l'élément de gestion de l'état de charge du réseau.

Le gestionnaire de charge : correspond à l'élément de contrôle.

Les deux modules se trouvent sur chaque nœud .

La classe CollecteCharge : Cette classe encapsule une partie de l'algorithme d'équilibrage de charges. Elle Implémente la collecte d'information de charge du réseau et l'évaluation locale de la charge.

Au niveau de chaque clusterhead, un tableau (global) est maintenu avec les valeurs envoyées par chaque nœud.

Les méthodes de cette classe :

SendCharge (charge, nœud) :Envoi de l'information sur la charge à un nœud spécifique.

DiffusionCharge (chargemoy, membre) : diffusion de la valeur de la charge moyenne du cluster vers les membres.

MiseAJourInfo () : actualise l'information de charge.

RecvCharge () : réception de l'information de la charge.

Evaluation () :évaluation de l'état de charge du cluster à partir des informations collectées. Cette méthode estime la charge des nœuds et ordonne les nœuds par leur niveau de charges .

La classe décision : Cette classe implémente le module de décision. Elle interagit avec la classe CollecteCharge. Les méthodes de cette classe :

LocalDécision () : décide du nœud auquel la charge doit être transférée.

Global Décision () : méthode qui décide du cluster auquel la charge doit être transférée.

La classe principale : Elle permet de créer l'ensemble des nœuds mobiles. Elle contient les méthodes nécessaires à la recherche des voisins de chacun des nœuds (RechercheVoisins), la formation des clusters (FormationCluster).

6.2 Évaluation de l'algorithme de clustering

L'algorithme de clustering proposé a été conçu dans l'objectif d'offrir une structure virtuelle pouvant servir comme appui à l'algorithme d'équilibrage de charges. Pour mettre en valeur les apports de ce mécanisme, on va se focaliser sur les critères suivants : le nombre moyen de clusterheads et l'impact de la mobilité sur les performances du réseau.

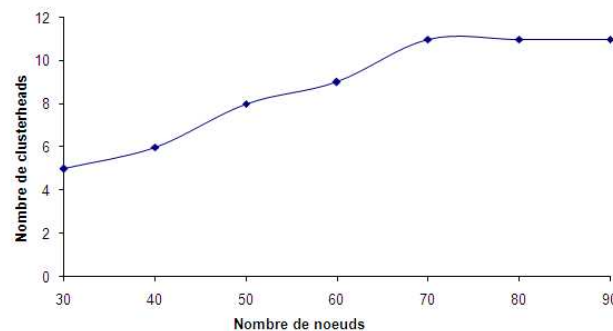


FIG. 6.1 – Nombres de clusterheads en fonction du nombre de noeuds (petites topologies)

Les résultats présentés tout au long de cette partie d'évaluation correspondent à la moyenne des résultats obtenus par simulation sur plusieurs topologies choisies de manière aléatoire.

6.2.1 Nombre moyen de clusterheads

Le critère le plus répandu pour évaluer l'efficacité d'un algorithme de clustering est le nombre de clusterheads qui servira comme infrastructure virtuelle. Ce critère reflète le nombre de noeuds qui vont assurer des tâches additionnelles au profit des autres noeuds du réseau. Dans cette suite d'expérimentations, on a calculé le nombre moyen de clusterheads générés, tout en variant le nombre de noeuds et la taille de l'aire de simulation. La vitesse de déplacement des noeuds est générée d'une manière aléatoire.

Dans une première étape, on choisit des petites topologies de manière aléatoire, confinées dans une aire de simulation $500 * 500\text{m}$. Les résultats exposés dans la figure 6.1 montrent que le nombre de clusterheads comptabilisé pour notre algorithme nous permet d'avoir un nombre moyen de coordinateurs et approximativement stable à partir de 70 noeuds. En effet, le recours au choix de clusterheads en considérant les critères d'énergie et la mobilité permet de favoriser certains noeuds plus aptes à assurer les rôles de clusterhead.

Dans la deuxième étape, on considère de grandes topologies, de taille variable. Ces différentes topologies sont réparties dans une aire de simulation de $1000 * 1000\text{m}$. Les résultats de la figure 6.2 consolident l'efficacité de notre mécanisme. Ce dernier maintient

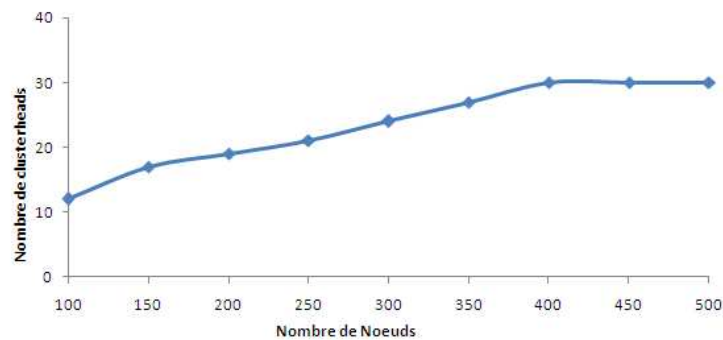


FIG. 6.2 – Nombres de clusterheads en fonction du nombre de nœuds (grandes topologies)

la taille de l'ensemble dominant aussi réduite que possible. Ainsi, on montre que la densité n'a pas beaucoup d'impact sur les performances de notre mécanisme de clustering.

6.2.2 Impact de la mobilité sur la stabilité du réseau

Un autre aspect auquel il faut s'intéresser est la robustesse de la structure de clustering générée face aux changements topologiques dus à la libre mobilité des différents nœuds. Dans ce contexte, on a évalué l'impact de la vitesse de déplacement sur les performances du réseau. Lors des simulations réalisées, le choix a été porté sur différentes topologies de 100 nœuds en mouvement dans une aire de simulation de $500 * 500$ m. Nous avons fait varier la vitesse des nœuds entre (2 et 10 m/s) et nous nous sommes intéressés à la durée de vie moyenne des clusterheads et le nombre moyen de clusters générés.

La durée de vie représentée dans la figure 6.3 diminue en fonction de la vitesse des nœuds. En effet, la mobilité des nœuds induit plus d'instabilité dans le réseau, toutefois, on remarque que cette diminution n'est pas importante et tend à se stabiliser à partir de ($v= 7$ m/s).

La figure 6.4 présente le nombre de clusters formés en fonction de la vitesse de déplacement des nœuds. Nous constatons que le nombre moyen de clusters formés est approximativement stable malgré le déplacement des nœuds, ce qui implique que la métrique utilisée pour élire les cluster-heads permet aux nœuds de se réorganiser et de joindre des

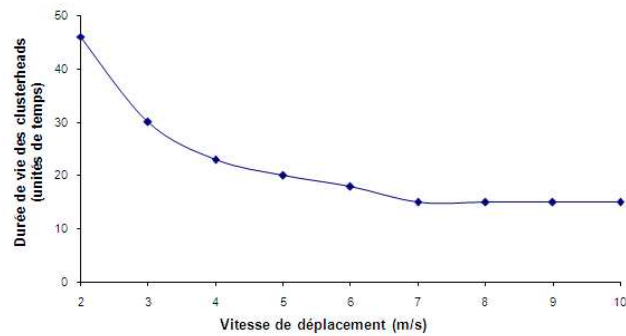


FIG. 6.3 – Durée de vie des clusterheads en fonction de la vitesse

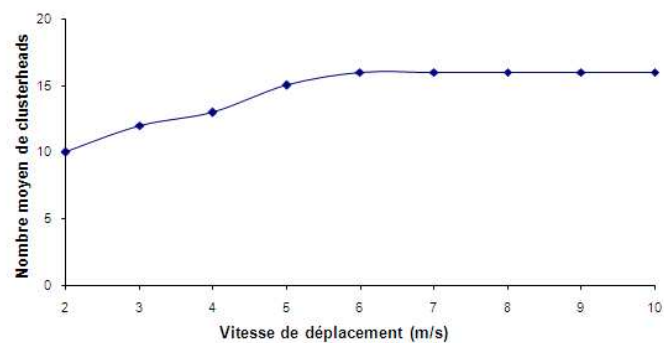


FIG. 6.4 – Nombre moyen de clusterheads en fonction de la vitesse

clusters déjà existants.

6.3 Évaluation de l'algorithme d'équilibrage de charges

Afin de mesurer la performance de notre algorithme, nous avons identifié les métriques suivantes : l'amélioration du temps d'exécution, l'énergie, le facteur d'équilibre, le seuil d'équilibrage et le surcoût d'équilibrage.

6.3.1 Amélioration du temps d'exécution

La première simulation a pour but de voir si l'algorithme d'équilibrage de charges donne une bonne amélioration du temps d'exécution des tâches. Pour cela nous avons fait

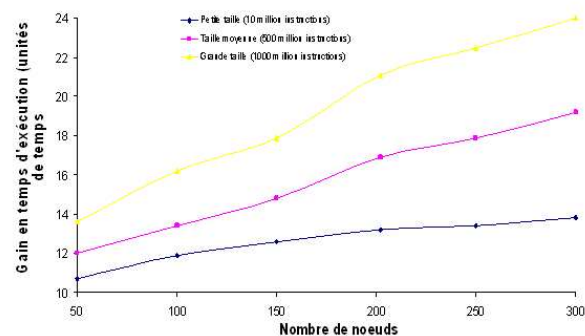


FIG. 6.5 – Amélioration du temps d'exécution en fonction du nombre de nœuds

varier le nombre de nœuds pour différentes tailles des travaux (100 millions instructions, 500 millions instructions, 1000 millions instructions). L'amélioration du temps d'exécution d'une tâche est la différence entre son temps d'exécution avec équilibrage de charges et son temps d'exécution sans équilibrage.

Nous remarquons dans la figure 6.5 que le temps d'exécution donne de meilleurs résultats pour un grand nombre de nœuds. Une amélioration nette est constatée surtout pour de grandes tailles de travaux. Ceci peut être expliqué par le fait que plus le nombre de nœuds présents dans le réseau est grand, plus la distribution des travaux à d'autres nœuds est grande. Ce qui améliore leurs temps d'exécution. Par rapport aux travaux de petite taille, quelque soit le nombre de nœuds, l'amélioration du temps d'exécution n'est pas vraiment remarquable, ceci est dû au fait qu'il n'y a pas un déséquilibre de charges dans le réseau et en conséquence il n'y a pas un grand besoin d'équilibrage.

6.3.2 Pourcentage des travaux interrompus

Le but de cette simulation est de voir si notre solution apporte une amélioration de l'exécution des charges liées aux activités des nœuds mobiles. Pour cela nous avons fixé la vitesse des nœuds à 6 m/s et nous avons fait varier les tailles des travaux et cela pour un nombre total de nœuds égal à 100.

Le pourcentage de travaux interrompus est calculé selon la formule suivante :

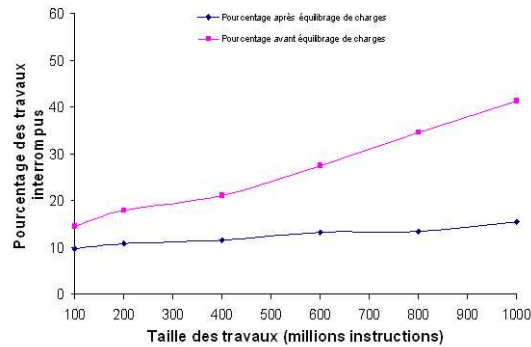


FIG. 6.6 – Pourcentage des travaux interrompus

$$\text{Pourcentage} = \frac{\text{nombre de noeuds morts ayant charge actuelle} > 0}{\text{nombre total de noeuds}} \quad (6.1)$$

La figure 6.6 montre une amélioration significative du pourcentage des travaux interrompus après l'équilibrage de charges. Ceci est la conséquence directe d'une plus grande probabilité de la présence de nœuds moins chargés avec des provisions énergétiques suffisantes. En effet, sans équilibrage les nœuds ayant une faible énergie sont incapables d'accomplir toutes leurs charges. Avec équilibrage de charges et avec la sollicitation des autres nœuds plus puissants, ces nœuds arrivent à achever leurs charges, ce qui permet de minimiser le pourcentage des travaux interrompus grâce à une bonne gestion de l'énergie des nœuds.

6.3.3 Impact de la mobilité sur l'énergie

L'objectif de la simulation suivante est d'étudier l'effet de la mobilité sur l'énergie des nœuds, sans et avec clustering.

D'après la figure 6.7, on peut dire que sans clustering, la consommation d'énergie varie proportionnellement avec la vitesse de déplacement des nœuds. Plus les nœuds se déplacent, la structure du réseau tend à changer, causant plus de messages de mises à jour de l'ensemble des informations sur les nœuds. Ainsi la recherche d'un nœud moins chargé pour recevoir une charge supplémentaire engendre des surcoûts dûs au temps de communication

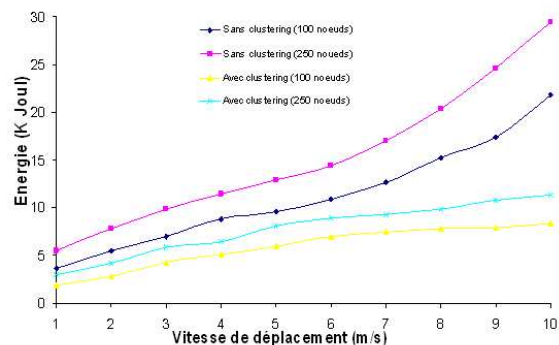


FIG. 6.7 – Impact de la mobilité sur l'énergie

pour avoir l'information sur l'état de charges des noeuds du réseaux , ce qui induit une plus grande consommation d'énergie. Comme cette approche (sans clustering) est complètement décentralisé, elle est donc sensible aux changements de voisinage qui a comme conséquence immédiate une augmentation de la consommation énergétique.

Cette situation n'est pas remarquée en distribuant les charges entre les différents noeuds groupés en clusters. En effet, pour de petites valeurs de vitesse, l'énergie consommée augmente légèrement, arrivée à une certaine vitesse (6 m/s), malgré qu'il y a changement de la structure des noeuds, avec un certain équilibre de charges au niveau local (niveau cluster) , la consommation d'énergie se stabilise. Ceci permet de prolonger la vie des noeuds ayant une faible énergie.

6.3.4 Facteur d'équilibre

Il est difficile de maintenir un système parfaitement équilibré. Afin de mesurer quantitativement le degré d'équilibre de charges entres les membres des clusters, nous introduisons un facteur d'équilibre F défini comme suit :

$$F = \frac{n}{\sum_i (load_i - \mu)^2} \quad (6.2)$$

n : nombre de noeuds d'un cluster (sa cardinalité) $load_i$: charge du noeud i

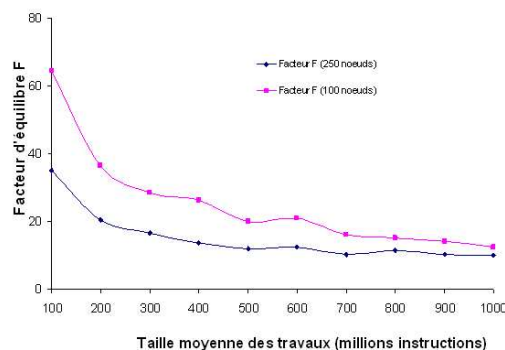


FIG. 6.8 – Facteur d'équilibre après équilibrage de charge

$$\mu = \frac{\sum_i load_i}{n} \quad (6.3)$$

μ : charge moyenne d'un cluster

Plus la valeur de F est grande, meilleure est la répartition des charges.

Nous remarquons d'après la figure 6.8 que le facteur F diminue légèrement avec l'augmentation de la taille des travaux, mais il se stabilise après. Ce qui explique que notre algorithme d'équilibrage de charges garantit une uniformité de distribution des travaux.

6.3.5 Valeur du seuil d'équilibrage de charges

Le seuil est une valeur qui est employée pour indiquer si un nœud est fortement ou légèrement chargé. Il est important de déterminer un seuil approprié pour un bon équilibrage de charges. Si le seuil est fixé à une valeur trop petite, des invocations excessives de l'algorithme d'équilibrage de charges se produisent, provoquant ainsi la dégradation des performances (car cela entraînerait des surcoûts de communication). Toutefois, si le seuil est trop élevé, le mécanisme d'équilibrage de charges ne sera pas très efficace car des déséquilibres seront détectés trop tard.

Le but de cette simulation est d'ajuster la valeur du seuil de charges qui influe directement sur l'algorithme d'équilibrage de charges.

Pour cela nous avons fixé le nombre de nœuds à 100 et la taille des travaux à 500 mil-

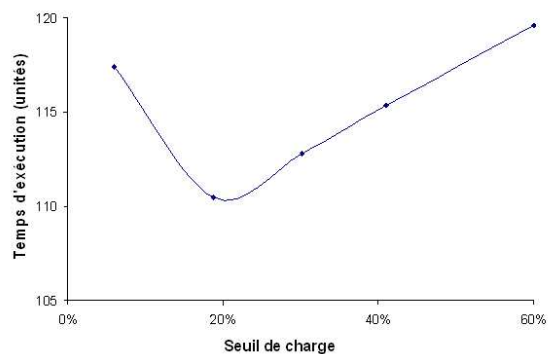


FIG. 6.9 – Temps d'exécution pour différentes valeurs du seuil d'équilibrage

lions instructions.

La figure 6.9 montre que le temps d'exécution est réduit avec l'augmentation de la valeur du seuil. Cependant, le temps d'exécution a augmenté après 20. Par conséquent, avoir un algorithme d'équilibrage de charges restrictif est inefficace. D'autre part, si la valeur du seuil est élevée, l'équilibrage de charges devient inefficace car les nœuds surchargés ne seront détectés qu'après qu'il soit trop tard. Cette simulation conduit à une constatation que ce mécanisme d'équilibrage de charges fonctionne mieux lorsque la valeur du seuil est de 20.

6.3.6 Coût d'équilibrage de charges

La simulation suivante a pour but d'étudier le surcoût engendré par l'algorithme d'équilibrage de charges au niveau de chaque nœud. Le surcoût des communications est le nombre total de messages relatifs à l'équilibrage de charges envoyés par chaque nœud. Il donne une mesure des surcoûts créés pour équilibrer les charges, qui sont également une indication du surcoût de notre algorithme en termes d'énergie.

Il est clair que le réseau devient plus dense avec l'augmentation du nombre de nœuds. Toutefois, la simulation de la figure 6.10 montre que l'algorithme d'équilibrage de charges ne coûte pas beaucoup de messages sur chaque nœud même lorsque le réseau devient plus dense. Cela est dû au fait que l'approche d'équilibrage de charges utilisée est centralisée

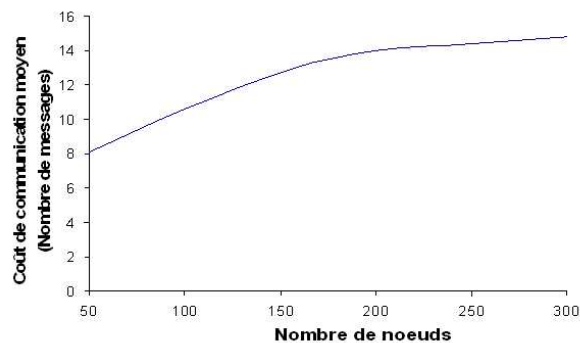


FIG. 6.10 – Coût d'équilibrage en fonction du nombre de nœuds

au niveau de chaque cluster. En effet, les échanges se font entre le clusterhead et ses membres. Les résultats des simulations montrent que les performances de notre algorithme sont stables lorsque le nombre de nœuds change.

Conclusion

Les nœuds mobiles présentent des capacités disparates. La solution que nous avons présentée permet l'amélioration du temps d'exécution des travaux en permettant de faire participer les nœuds forts (en termes d'énergie et de capacité de traitement). Cette solution permet également de mieux gérer l'énergie des nœuds, ce qui allonge leur durée de vie et garantit la stabilité du réseau ad hoc. La structure virtuelle d'auto-organisation présentée a permis de structurer le réseau en offrant une vue plus stable de la topologie. Cette structure est d'une grande utilité pour l'équilibrage de charges, comme nous l'avons vu dans ce chapitre.

Conclusion et perspectives

L'équilibrage de charge est l'une des clés de voûte pour améliorer la performance des réseaux Ad hoc. L'un de ses principaux objectifs est de garantir la stabilité du réseau et de satisfaire les besoins des utilisateurs en termes de temps d'exécution de leurs applications.

Dans cette thèse, nous avons proposé une solution d'équilibrage de charges où les nœuds les moins chargés aident les nœuds trop chargés. Cette solution se décompose en deux étapes : la première étape permet de structurer le réseau en clusters. Dans la deuxième étape, nous avons présenté notre algorithme d'équilibrage de charges.

L'équilibrage de charges se fait au niveau de chaque cluster et entre les différents clusters s'il y a nécessité. L'avantage de l'architecture proposée est que la prise de décision s'effectue en fonction de la charge locale du cluster, ce qui permet de réduire les échanges d'information. Comparativement aux travaux existants sur l'équilibrage de charges, notre stratégie se caractérise par les éléments suivants :

- C'est une stratégie d'équilibrage au niveau des tâches de calcul du nœud ;
- C'est une stratégie qui favorise un transfert local de tâches (intra-cluster) pour éviter un surcoût de communication ;
- C'est une stratégie distribuée, dans la mesure où plusieurs opérations d'équilibrage peuvent se faire en parallèle (équilibrage au niveau de chaque cluster) ;
- La décision d'effectuer une opération d'équilibrage de charges, pour un cluster, se fait uniquement sur la base des informations de charge locales à ce cluster.

Les résultats obtenus sont présentés dans le dernier chapitre. Ils soulignent l'efficacité de notre stratégie pour assurer une amélioration nette en termes de temps d'exécution et d'énergie. Nous avons analysé cette amélioration selon plusieurs perspectives. Nous avons

effectué des expérimentations en modifiant la taille du problème, c'est à dire en augmentant la charge et le nombre de noeuds.

Les résultats de ces expérimentations montrent que l'apport de l'équilibrage de charges est d'autant plus important que la taille du problème est grande. En variant le nombre de noeuds et leurs vitesses de déplacement, nous avons constaté que la consommation d'énergie se stabilise, ce qui permet de prolonger la vie des noeuds ayant une faible énergie.

De nombreux aspects, et plus particulièrement pour l'équilibrage de charges, restent à approfondir.

- La prise de décision dépend des seuils de charge et d'énergie et la réactivité du système dépend de ces seuils. La définition des seuils nécessite la quantification de la charge engendrée par l'exécution des noeuds mobiles. Il est certain que ce point mérite d'être approfondi (seuils dynamiques). Une question à résoudre est de considérer si les seuils sont fixes ou s'ils s'adaptent au fur et à mesure que la charge du réseau évolue. L'implémentation de seuils variables n'est pas facile à résoudre car elle pose plusieurs questions : quand démarrer l'activation de la modification, qui devrait faire la modification, comment diffuser la nouvelle valeur ou quelle valeur le nouveau seuil doit-il prendre ? etc.

- Un autre point important est que toutes les analyses réalisées sur l'algorithme proposé ont été conduites en supposant que tous les noeuds du réseau sont coopératifs. Considérer un réseau totalement coopératif permet de n'étudier que les caractéristiques propres à l'algorithme considéré, ce qui était le but recherché dans cette thèse. Cependant, un noeud mobile peut présenter dans certaines situations un comportement malicieux. Il serait intéressant dans l'avenir d'étudier l'impact des noeuds malicieux sur les performances de l'algorithme d'équilibrage de charges.

- La solution décrite ici est générique et peut s'appliquer à tout type de réseau sans fil ad hoc. Cependant, il nous semble que pour concevoir un réseau efficace, il faut également considérer l'application du réseau. Ainsi, nous aimerions par la suite considérer plusieurs applications plus ciblées et un type de réseau plus précis comme par exemple les réseaux de capteurs. En effet, les réseaux de capteurs offrent un certain nombre de défis et de verrous scientifiques.

- Dans le cadre de cette thèse, nous avons étudié des stratégies à l'initiative des noeuds surchargés (source initiative). Avec notre approche nous avons obtenu des résultats satisfaisants. Nous aimerions élargir notre approche pour étudier d'autres stratégies. Il serait ainsi pertinent de procéder à l'étude des stratégies à l'initiative du receveur.

- Enfin, les études que nous avons menées, ont été effectuées en se basant sur une approche de clustering. Il serait intéressant de réaliser ces études dans des conditions variables de choix des clusterheads en utilisant différents algorithmes de clustering présentés dans la littérature de manière à comparer leurs performances.

Bibliographie

- [1] Alexander Klemm, Christoph Lindemann, and Oliver P. Waldhorst. A special-purpose peer-to-peer file sharing system for mobile ad hoc networks. In *IEEE Semiannual Vehicular Technology Conference (VTC2003-Fall)*, 2003.
- [2] Gerd Kortuem, Steve Fickas, and Zary Segall. Architectural issues in supporting ad-hoc collaboration with wearable computers. In *The 22nd International Conference on Software Engineering (ICSE2000)*, editor, *Workshop on Software Engineering For Wearable and Pervasive Computing (SEWPC)*, Limerick, Ireland, 4-11 June 2000.
- [3] Andreas Heinemann, Jussi Kangasharju, Fernando Lyardet, and Max Mühlhäuse. iclouds peer-to-peer information sharing in mobile environments. *Lecture Notes in Computer Science*, 2790 :1038–1045, 2004.
- [4] Y. T. Wang and R. J. T. Morris. Load sharing in distributed systems. *ACM Journal*, 17 :123–130, 1970.
- [5] A. Thomasian. A performance study of dynamic load balancing in distributed systems. In *7th international conference on Distributed Systems*, pages 178–184, 1987.
- [6] Gilles Carpeuteer, Thomas Coustenoble, and Benjamin Crombe. *Solutions mobiles*. Dunod, Paris, 2003.
- [7] W. R. Young. Advanced mobile phone service : Introduction, background, and objectives. Technical report, Bell Syst, January 1979.
- [8] J. Jubin and J. D. Tornow. The darpa packet radio network protocols. In *Proceedings of the IEEE*, volume volume 75, January 1987.

-
- [9] Khaldou Al Agha, Guy Pujolle, and Guillaume Vivier. *Réseaux de Mobiles et réseaux sans fil*. Edition Eyrolles, 2001.
- [10] S. Jiang, N. Vaidya, and Wei Zhao. Preventing traffic analysis in packet radio networks. Number 2, pages 153–158, Anaheim, CA, USA, Mai 2001.
- [11] J. A. Freebersyser and B. Leiner. *A DoD Perspective on Mobile Ad hoc Networks*, pages 29–51. Ad hoc Networking, Addison-Wesley Longman Publishing Co, Boston, MA, USA, 2001.
- [12] V. Bellotti and S. Bly. Walking away from the desktop computer : Distributed collaboration and mobility in a product design team. 1996.
- [13] M. Perry, K. O’Hara, A. Sellen, B. Brown, and R. Harper. Dealing with mobility understanding access anytime, anywhere. *Journal of ACM Transactions on Computer Human Interaction (TOCHI)*, 8(4), 2001.
- [14] B. Nardi, S. Whittaker, and E. Bradner. Interaction and outeraction : Instant messaging in action. Philadelphia, PA, Dec 2000.
- [15] D. Margulius. Collaborative challenges. *Journal of InfoWord*, July 2002.
- [16] T. Saridakis. Nomadic collaboration management. 2002.
- [17] Dennis A.R and Venkatesh V. Adoption of collaboration technologies : integrating technology acceptance model and collaboration technologies research. Technical report, Working Papers on Information Systems / Indiana University, 2003.
- [18] D. Agarwal and K. Berket. Supporting dynamic ad hoc collaboration capabilities. La Jolla, California, 24-28 March 2003.
- [19] I. Stoica Morris and H. Balakrishnan. Building peer-to-peer systems with chord, a distributed lookup service. Schloss Elmau, Germany, May 2001.
- [20] <http://www.irc.org/>.
- [21] <http://www.accessgrid.org>.

- [22] J. Kubiawicz, D. Bindel, Y. Chen, S. Czerwinski, P. Eaton, D. Geels, R. Gummadi, R. Rhea, H. Weatherspoon, W. Weimer, C. Wells, and B. Zhao. Oceanstore : An architecture for global-scale persistent storage. November 2000.
- [23] <http://www.groove.net/>.
- [24] Justin Frankel and Tom Pepper. Gnutella protocol specs version 0.4. *Available from : http://www9.limewire.com/developer/gnutella_protocol_0.4.pdf*, année 2000.
- [25] Siddhartha K. Goel, Manish Singh, and Dongyan Xu. Efficient peer-to-peer data dissemination in mobile ad-hoc networks. In IEEE Computer Society, editor, *International Conference on Parallel Processing Workshops*, pages 152–158, 2002.
- [26] Alexander Klemm, Christoph Lindemann, and Oliver P. Waldhorst. Peer-to-peer computing in mobile ad hoc networks. In *Performance Tools and Applications to Networked Systems : Revised Tutorial Lectures in Computer Science*, volume 2965 :187–208, Springer 2004.
- [27] G. Kortuem, Z. Segall, and T.G.C. Thompson. Close encounters : supporting mobile collaboration through interchange of user profiles. In *First International Conference on Handheld and Ubiquitous Computing (HUC)*, Karlsruhe, Germany, September 1999.
- [28] J Favela, C Navarro, and M Rodriguez. Extending instant messaging to support spontaneous interactions in ad-hoc networks. In *The ACM 2002 Conference on Computer Supported Cooperative Work (CSCW 2002)*, New Orleans, LA, USA, 2002.
- [29] M Esbjörnsson and Östergren. Issues of spontaneous collaboration and mobility. In *Spontaneity'02, workshop proceedings on Supporting Spontaneous Interaction in Ubiquitous Computing Settings*, at UBICOMP'02 2002.
- [30] Arman Danesh and Kori Inkpen. Collaborating on ad hoc wireless networks. www2.parc.com/csl/projects/ubicomp-workshop/danesh.pdf.
- [31] Gupta P and Kumar P. R. The capacity of wireless networks. *Actes des transactions IEEE sur l'informatique théorique*, pages 388–404, 2000.

- [32] S. Narayanaswamy, V. Kawadia, R. S. Sreenivas, and R. Kumar. Power control in ad hoc networks : theory, architecture, algorithm and implementation of the compow protocol. In *European Wireless Conference*, 2002.
- [33] Yuen W. H. On energy efficiency and network connectivity of mobile ad hoc networks. *Actes de IEEE ICDCS 2003, Providence, Rhode Island, Mai 2003*.
- [34] Burkhart M, von Rickenbach P, Wattenhofer R, and Zollinger A. Does topology control reduce interference? *Actes du 5me symposium international ACM sur Les Réseaux mobiles ad hoc*, pages 9–19, Tokyo, Japan, 2004.
- [35] Ram Ramanathan and Regina Rosales-Hain. Topology control of multihop wireless networks using transmit power adjustment. In *Proceedings of INFOCOM*, pages 404–413, 2000.
- [36] S. Narayanaswamy, V. Kawadia, R. S. Sreenivas, and R. Kumar. Power control in ad hoc networks : theory, architecture, algorithm and implementation of the compow protocol. *European Wireless Conference*, 2002.
- [37] Meguerdichian S, Koushanfar F, Potkonjak M, and Srivastava M. B. Coverage problems in wireless ad hoc sensor networks. *Actes de la seconde Conférence internationale ACM sur les réseaux sans fil et de sensors et leurs applications, San Diego, USA*, pages 115–121, 2003.
- [38] Wireless lan medium, access control (mac) and physical, layer (phy) specifications. ANSI/IEEE Standard 802.11, 1999.
- [39] Changsu Suh, Young-Bae Ko, and Jai-Hoon Kim. Enhanced power saving for ieee 802.11 wlan with dynamic slot allocation. *LNCS*, Vol 3794 :466–477, 2005.
- [40] A. Belghith and W. Akkari. Traffic aware power conservation mechanism for ad hoc networks. *International Journal of Computing and Information Sciences (IJCIS), Canada*, 2006.
- [41] Ye W, Heidemann J, and Estrin D. An energy-efficient mac protocol for wireless sensor networks. *Actes de la conférence IEEE INFOCOM 2002*, Juin 2002.

-
- [42] Van Dam T and Langendoen K. An adaptive energy-efficient mac protocol for wireless sensor networks. *Actes de la conférence ACM SenSys 2003*, November 2003.
- [43] Rabaey J M, Ammer M. J, da Silva Jr J L, Patel D, and Roundy S. Picoradio supports for ad hoc ultra-low power wireless networking. *Actes de la Conférence IEEE Computer*, Juillet 2000.
- [44] Singh S, Woo M, and Raghavendra C.S. Power aware routing in mobile ad hoc networks. In *proceeding of mobicom'98*, Dallas, Texas, octobre 1998.
- [45] C-K. Toh. Maximum battery life routing to support ubiquitous mobile computing in wireless ad hoc networks. *IEEE Communications Magazine*, Juin 2001.
- [46] J-E Garcia. A novel dsr-based energy-efficient routing algorithm for mobile ad-hoc networks. *IEEE VTC-fall 03*, October 2003.
- [47] S-M. Senouci and G. Pujolle. Energy efficient routing in wireless ad hoc networks. In *Proc. ICC'04*, June 2004.
- [48] C-M Chao, J-P Sheu, and C-T. Hu. Energy conserving grid routing protocol in mobile ad hoc networks. *IEEE ICPP'03*, 2003.
- [49] A. Belghith and W. Akkari. Performance evaluation of power save protocols using carrier sensing in multihop ad hoc networks. In *The Seventh ACS/IEEE International Conference on Computer Systems and applications, AICCSA'09*, Rabat, Morocco, May 10-13 2009.
- [50] H. Willebeek-LeMair and P. Reeves. Strategies for dynamic load balancing on highly parallel computers. *IEEE Transactions on Parallel and Distributed Systems*, 4(9) :13, September 1993.
- [51] V.Kumar, A.Y.Grama, and V.N Rao. Scalable load balancing techniques for parallel computer. *Journal of Parallel and Distributed Computing*, 22 :60–79, 1994.
- [52] D.L. Eager, E.D. Lazwska, and J. Zahorjan. A comparison of receiver-initiated and sender-initiated adaptative load sharing. *Performance Evaluation*, pages 53–68, 1986.

- [53] T. Casavant and J.G. Kuhl. A taxonomy of scheduling in general-purpose distributed computing systems. *IEEE Transactions on Software Engineering*, 14(2) :141–154, February 1988.
- [54] M. Cierniak, M. Zaki, and W. Li. Customized dynamic load balancing for a network of workstations. *Journal of Parallel and Distributed Computing*, 43 :156–162, 1997.
- [55] M. Hamdi and C. Lee. Dynamic load balancing of data parallel applications on a distributed network. In ACM Press, editor, *Dans 9th International Conference on Supercomputing ICS'95*, pages 170–179, 1995.
- [56] D. Culler, R. Karp, D. Patterson, A. Sahay, K. Schauer, E. Santos, R. Subramonian, and T. v. Eicken. Logp : Towards a realistic model of parallel computation. In *Proceedings of the Fourth ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming*, ACM Press, pages 12–13, 1993.
- [57] Z. Lan, V. Taylor, and G. Bryan. Dynamic load balancing of samr applications on distributed systems. In *Proceedings of the ACM/IEEE Symposium on Supercomputing (SC'01)*, pages 14–19, IEEE Computer Society Press, 2001.
- [58] Parveen Jain and Daya Gupta. An algorithm for dynamic load balancing in distributed systems with multiple supporting nodes by exploiting the interrupt service. *International Journal of Recent Trends in Engineering*, 1(1), May 2009.
- [59] NG.Shivarti, P.Krueger, and M.Singhal. Load distributing for locally distributed systems. *IEEE comput*, pages 33–44, December 1992.
- [60] S.Zhou. A trace-driven simulation study of dynamic load balancing. *IEEE Transaction on Software Engineering*, 14(9) :1327–1340, December 1988.
- [61] R.Luling, B.Moniem, and F.Ramme. Load balancing in large networks : A comparative study. *3 rd IEEE Symposium on Parallel and Distributed Processing, Dallas*, 1991.
- [62] H.Kuchen and A.Wagener. Comparison of dynamic load balancing strategies. *Journal of Parallel and Distributed Computing*, pages 303–314, 1991.

- [63] R. Subramanian and I.D.Scherson. An analysis of diffusive load balancing. *In SPA A'94*, 1994.
- [64] Hossam Hassanein and Audrey Zhou. Routing with load balancing in wireless ad hoc networks. *In Proceedings of the 4th ACM international workshop on Modeling, analysis and simulation of wireless and mobile systems*, pages 89 – 96, 2001.
- [65] Andre.Schumacher, Harri.Haanpaa, Elisa.Schaeffer, and Pekka Orponen. Load balancing by distributed optimisation in ad hoc networks. *Lecture Notes in computer Science, Springer Berlin*, pages 873–884, 2006.
- [66] Sanghyun Ahn, Yujin Lim, and Jongwon Choe. A load-balancing approach in ad-hoc networks. *Springer Berlin / Heidelberg*, 2662/2003 :pages 672–681, octobre 2003.
- [67] Lu Xuechao, Xu Chunxiu, Wu Muqing, Zhen Yan, and Wu Dapeng. Design and realization of a novel multi-path load-balancing routing protocol in ad hoc network. *In International Conference on Information Engineering*, volume 1, pages 247 – 250, 10-11 July 2009.
- [68] D.A. ALAN and P RAVI. Load-balancing clusters in wireless ad hoc networks. *In Proceedings of the 3rd IEEE Symposium on Application-specific Systems and Software Engineering Technology (ASSET'00)*, pages 25–32, 2000.
- [69] TUGRUT D.K., DAS S.K., and ELMASRI R. Balancing loads in ad hoc mobile networks. *In 10th International Conference on Telecommunications. ICT 2003*, pages 490–495, 2003.
- [70] Rachida Aoudjit, Mustapha Lalam, Abdelaziz M'zoughi, Malika Belkadi, and Mehammed Daoui. Load balancing : An approach based on clustering in ad hoc networks. *Journal of Computing and Information Technology - CIT*, 17(2) :177–184, 2009.
- [71] A. Sridharan, R. Guérin, and C. Diot. Achieving near-optimal traffic engineering solutions for current ospf/is-is networks. *IEEE/ACM Trans. Netw.*, 13(2) :234–247, 2005.

- [72] Y. Wang, Z. Wang, and L. Zhang. Internet traffic engineering without full mesh overlaying. *In INFOCOM*, 2001.
- [73] A. Feldmann, A. Greenberg, C. Lund, N. Reingold, J. Rexfor, and F. True. Deriving traffic demands for operational ip networks : Methodology and experience. *IEEE/ACM Transaction on Networking*, 30(4) :257 – 270, October 2000.
- [74] C. Zhanga, J. Kurosea, D. Towsley, Z. Ge, and Y. Liu. Optimal routing with multiple traffic matrices tradeoff between average and worst case performance. *In ICNP*, pages 215–224, Washington, DC, USA, 2005. IEEE Computer Society.
- [75] C. Zhang, Y. Liu, W. Gong, J. F. Kurose, R. Moll, and D. F. Towsley. On optimal routing with multiple traffic matrices. *In INFOCOM*, pages 607–618, 2005.
- [76] D. Applegate, L. Breslau, and E. Cohen. Coping with network failures : routing strategies for optimal demand oblivious restoration. *In SIGMETRICS*, pages 270–281, New York, NY, USA, 2004. ACM.
- [77] D. Applegate and E. Cohen. Making intra-domain routing robust to changing and uncertain traffic demands : understanding fundamental tradeoffs. *In SIGCOMM*, pages 313–324, New York, NY, USA, 2003. ACM.
- [78] A. Elwalid, C. Jin, S. H. Low, and I. Widjaja. Mate : Mpls adaptive tra ?c engineering. *In INFOCOM*, pages 1300–1309, 2001.
- [79] Tran Cong Hung, Nguyen Hoang Thanh, Nguyen Duc Thang, Hae Won Jung, Tae Il Kim, Sung Hei Kim, and Woo Jin Yang. Advanced routing algorithms and load balancing on mpls. *In The 9th International Conference on Advanced Communication Technology*, volume 3, pages 1886 – 1891, 2007.
- [80] D. Katabi, M. Handley, , and C. Rohrs. Internet congestion control for future high bandwidth-delay product environments. *SIGCOMM*, 2002.
- [81] H.Wang, H. Xie, L. Qiu, Y. R. Yang, Y. Zhang, and A. Greenberg. Cope : traffic engineering in dynamic networks. *In SIGCOMM*, pages 99–110, New York, NY, USA, 2006. ACM.

- [82] C. Villamizar. Ospf optimized multipath (ospf-omp). draft-ietf-ospf-omp-02.txt. Draft, IETF, February 1999.
- [83] SANON. C, KOMWUT. W, and SUWAN. R. Load balancing for zone routing protocol to support qos in ad hoc network. In *the 2002 International Technical Conference on Circuits/Systems, Computers and Communications (ITC-CSCC 2002)*, pages 1685–1688, Phuket, Thailand, 16-19 July 2002.
- [84] Thongpook. Load balancing of adaptive zone routing in ad hoc networks. *TENCON 2004. 2004 IEEE*, Vol.2 :pages 672 – 675, 21-24 Nov 2004.
- [85] XiaoRan Wang, Shigeaki Tagashira, and Satoshi Fujita. Fdar : A load-balanced routing scheme for mobile ad-hoc networks. In *Ad Hoc, Mobile and Wireless Networks, 6th international Conference, ADHOC- NOW 2007*, pages 186–197, Morelia, Mexico, September 2007.
- [86] ZHEN KONG and FURONG WANG. Load balancing mechanism for real-time traffic in wireless ad hoc networks. In *Wireless communications and networks. Conference*, volume 5284, pages 504–513, Wuhan , CHINE, 2004.
- [87] Oussama Souihlia, Mounir Frikha, and Mahmoud Ben Hamouda. Load-balancing in manet shortest-path routing protocols. *Elsevier Science Publishers*, 7(2) :431–442, March 2009.
- [88] Rani.A and Dave.M. Weighted load balanced routing protocol for manet. In *Proceedings of the 16th International Conference on Networks, ICON 2008*, pages 1–6, December 12-14 2008.
- [89] Algamali. M and Wang Jian xin. Aptlb : A practical approach for providing traffic load-balancing in wireless mesh networks. In *5th International Conference on Wireless Communications, Networking and Mobile Computing, 2009 (WiCom '09)*, pages 1 – 4, Sept 2009.
- [90] HO, C.K.and EWE, and HONG. A hybrid ant colony optimization approach (haco) for constructing load balanced clusters. *EvolutionaryComputation . The 2005 IEEE Congress*, Vol. 3 :pages 2010–2017, Sept 2005.

- [91] I. Cardei, S. Varadarajan, A. Pavan, L. Graba, M. Cardei, and M. Min. Resource management for ad hoc wireless networks with cluster organization. *Cluster Computing*, 7(1) :91–103, January 2004.
- [92] M. Jiang, J. Li, and Y. Tay. Cluster based routing protocol (cbrp). *DRAFT draft-ietf-manet-cbrp-spec-01.txt*, IETF, July 1999.
- [93] C. Perkins. Ad hoc networking. *Addison-Wesley*, 2001.
- [94] C. R. Lin and M. Gerla. Adaptive clustering for mobile wireless networks. *IEEE Journal of Selected Areas in Communications*, 15(7) :1265–1275, 1997.
- [95] A. Ephremides, J. Wieselthier, and D. Baker. A design concept for reliable mobile radio networks with frequency hopping signaling. In *IEEE 75*, pages 56–73, 1987.
- [96] M. Gerla and J. T.-C. Tsai. Multicluster, mobile, multimedia radio network. *ACM/JVI/Baltzer Journal of Wireless Networks*, 1(3) :255–265, July 1995.
- [97] C. Chiang, H. Wu, W. Liu, and M. Gerla. Routing in clustered multihop, mobile wireless networks with fading channel. In *ICCS/ISPAcS'96*, Singapore, November 1996.
- [98] P. Basu, N. Khan, and T. Little. A mobility based metric for clustering in mobile ad hoc networks. In *Distributed Computing Systems Workshop (DISC)*, 2001.
- [99] M. Chatterjee, S. K. Das, and D. Turgut. Wca : A weight based distributed clustering algorithm for mobile ad hoc networks. *Journal of Cluster Computing (Special Issue on Mobile Ad hoc Networks)*, 5(2) :193–204, April 2002.
- [100] S. Basagni. Distributed clustering for ad hoc networks. In *International Symposium on parallel architectures algorithms and networks (I-SPAN'99)*, pages 310–315, Fremantle, Australia, June 1999.
- [101] J. Y. Yu and P. H. Chong. 3hbc (3hop between adjacent clusterheads) : A novel non-overlapping clustering algorithm for mobile ad hoc networks. In *PacRim '03*, Victoria, Canada, August 2003.

- [102] L. Kai and L. Jiandong. Mobile cluster protocol in wireless ad hoc networks. In *International Conference on Communication Technology (ICCT'2000) Proceedings*, August 2000.
- [103] T. J. K. Kwon and M. Gerla. Efficient flooding with passive clustering (pc) - an overhead-free selective forward mechanism for ad hoc/sensor networks. In *Proceedings of IEEE*, pages 1210–1220, 2003.
- [104] G. Chen, F. Garcia, J. Solano, and I. Stojmenovic. Connectivity-based k-hop clustering in wireless networks. In *35th Annual Hawaii International Conference on System Sciences (HICSS'02)*, Hawaii, USA, January 2002.
- [105] A. Ramalingam, S. Subramani, and K. Perumalsamy. Associativity-based cluster formation and cluster management in ad hoc networks. In *9th International conference on high performance computing (H1PC'02)*, Bangalore, India, December 2002.
- [106] H.-C. Lin and Y.-H. Chu. A clustering technique for large multihop mobile wireless networks. In *In Vehicular Technology Conference (VTC '00)*, Tokyo, Japan, May 2000.
- [107] Y. Fernandess and D. Malkhi. k-clustering in wireless ad hoc networks. In *ACM international workshop on Principles of mobile computing*, Toulouse, France, 2002.
- [108] K. Alzoubi, P. Wan, and O. Frieder. New distributed algorithm for connected dominating set in mobile ad hoc networks. In *35th Annual Hawaii International Conference on System Sciences (HICSS'02)*, Hawaii, USA, January 2002.
- [109] A. Amis, R. Prakash, T. Vuong, and D. Huynh. Max-min cl-cluster formation in wireless ad hoc networks. In *INFOCOM*, Tel-Aviv, Israël, March 2000.
- [110] A. Amis and Prakash. Load-balancing clusters in wireless ad hoc networks. In *ASSET, Richardson, Texas, USA*, March 2000.
- [111] M. Chatterjee P. Krishna, N. H. Vaidya and D. K. Pradhan. A cluster based approach for routing in dynamic networks. *ACM SIGCOMM*, April 1997.

-
- [112] T-C Hou and T-J Tsai. An access-based clustering protocol for multihop wireless ad hoc networks. *IEEE Journal on Selected Areas in Communications*, 19(7) :1201–1210, July 2001.
- [113] T Ohta, S moue, and Y Kakuda. An adaptive multihop clustering scheme for highly mobile ad hoc networks. In *IEEE International Symposium on Autonomous Decentralized Systems (ISADS'03)*, pages 293–300, April 2003.
- [114] Banerjee and S. Khuller. A clustering scheme for hierarchical control in multi ho wireless networks. In *INFOCOM, Anchorage, Alaska, USA*, April 2001.
- [115] B. An and S. Papavassiliou. A mobility-based clustering approach to support mobility management and multicast routing in mobile ad-hoc wireless networks. *International Journal of Network Management*, 11(6) :387–395, November 2001.
- [116] R. Riedi, P. Druschel, Y. C. Ru, D. B. Johnson, and R. Baraniuk. Safari : A selforganizing hierarchical architecture for scalable ad hoc networking networking. Research report tro4-433, Rice University, February 2005.
- [117] Cintia B.Margi, Ktia Obraczka, and Roberto Manduchi. Characterizing system level energie consumption in mobile computing platforms. In *Wireless Networks, Communications and Mobile Computing, 2005 International Conference on*, volume 2, pages 1142– 1147, June 2005.