ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
UNIVERSITÉ DU QUÉBEC

THÈSE PRÉSENTÉE À
L'ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

COMME EXIGENCE PARTIELLE
À L'OBTENTION DU
DOCTORAT EN GÉNIE
PH.D.

PAR
KAIS MNIF

CONSTRUCTION ET MAINTENANCE D'UNE DORSALE VIRTUELLE DANS LES RÉSEAUX *AD HOC* MOBILES

MONTRÉAL, 18 DECEMBRE 2006

CETTE THÈSE A ÉTÉ ÉVALUÉE
PAR UN JURY COMPOSÉ DE :

M. Michel Kadoch, directeur de thèse
Département de génie électrique à l'École de technologie supérieure

M. Robert Hausler, président du jury
Département de génie de la construction à l'École de technologie supérieure

M. Zbigniew Dziong, membre du jury
Département de génie électrique à l'École de technologie supérieure

M. Jean-Charles Grégoire, membre du jury
INRS-Énergie, Matériaux et Télécommunications

ELLE A FAIT L'OBJET D'UNE SOUTENANCE DEVANT JURY ET PUBLIC
LE 8 DÉCEMBRE 2006
À L'ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

CONSTRUCTION ET MAINTENANCE D'UNE DORSALE VIRTUELLE DANS LES RÉSEAUX *AD HOC* MOBILES

Kaïs Mnif

SOMMAIRE

Un réseau *ad hoc* mobile est un réseau complètement distribué ne nécessitant pas d'infrastructure fixe. Les terminaux sont libres de se déplacer n'importe quand et dans n'importe quelle direction. L'absence d'une infrastructure nécessite la collaboration de tous les terminaux pour acheminer le trafic d'une source vers une destination. De nombreux protocoles de routage ont été proposés pour assurer le relayage multi saut, utilisant différentes approches (réactives, proactives et hybrides). Toutefois, les performances de ces protocoles se dégradent en présence de la mobilité. Dans cette thèse, nous proposons d'améliorer la performance des protocoles de routage dans les réseaux *ad hoc* en construisant une dorsale virtuelle.

Une dorsale virtuelle est un sous-ensemble de nœuds sélectionnés de façon à ce que chaque nœud dans le réseau possède au moins un voisin dans la dorsale. L'ensemble des nœuds qui forment la dorsale doit être toujours maintenu connecté même quand les nœuds changent de position. Plus la taille de la dorsale est minimale, plus la maintenance est efficace. Pour construire la dorsale, nous avons proposé un nouvel algorithme basé sur l'approximation de l'ensemble de domination connexe de taille minimale (MCDS). Un réseau *ad hoc* est généralement modélisé par un graphe à disque unitaire UDG (*Unit Disc Graph*).Trouver l'ensemble MCDS dans un graphe UDG est un problème *NP-Complet*. Dans le but de réduire cette complexité, nous avons décomposé le problème en deux étapes : la première étape consiste à déterminer l'ensemble de domination connexe (MDS) au moyen d'une formulation en programmation linéaire et la deuxième étape consiste à trouver l'arbre de recouvrement de l'ensemble MDS et en déduire l'ensemble MCDS. Les résultats de simulations montrent bien que la solution donnée par notre algorithme est très proche de celle fournie par l'analyse théorique. De plus, la taille de la dorsale est nettement inférieure comparée à d'autres algorithmes proposés dans la littérature quand la taille du réseau augmente.

Nous avons également proposé une procédure de maintenance distribuée. Cette procédure est basée sur un échange simple des messages de contrôle *hello* modifiés, ces messages étant utilisés pour la découverte au voisinage. Un nœud qui change de position va alors appliquer cette procédure pour se connecter à la dorsale. Une maintenance locale de la dorsale sera effectuée dans la zone où le nœud va se retrouver dans sa

nouvelle position. Les résultats de simulation ont démontré l'efficacité et la fiabilité de cette approche. En effet, plus de 90 % des nœuds restent connectés pour une mobilité élevée (vitesse moyenne égale à 30 *m/s*). De plus, cette procédure est peu sensible au facteur de mise à l'échelle (*Scalability*). La nature distribuée de la procédure de maintenance s'adapte bien à la dynamique de la structure du réseau engendrée par le mouvement des nœuds.

Dans le but de vérifier l'amélioration apportée par la présence d'une dorsale pour les protocoles de routage dans les réseaux *ad hoc* mobiles, nous avons comparé les performances de certains protocoles de routage, en fonction de la mobilité, en présence de la dorsale avec leurs performances dans leurs versions standards. Les résultats de simulations ont démontré qu'une amélioration de leurs performances peut atteindre 20 % pour certains protocoles même pour une mobilité élevée.

En conclusion, ce travail de recherche présente de nouvelles solutions pour différents problèmes reliés au routage dans les réseaux *ad hoc* mobiles.

CONSTRUCTION AND MAINTENANCE OF VIRTUAL BACKBONE IN MOBILE AD HOC NETWORKS

Kaïs Mnif

ABSTRACT

Mobile *ad hoc* networks are totally distributed which do not need a fixed infrastructure and terminals are free to move anywhere at anytime. The absence of an infrastructure requires collaboration of all the terminals to forward traffic from a source to a destination. Many routing protocols have been proposed to ensure the multi hop relaying, with either reactive, proactive or hybrid approach. However, the performance of these protocols degrades when mobility is present. In this thesis, a virtual backbone-based routing protocol is proposed and evaluated for mobile *ad hoc* networks.

A virtual backbone is a subset of selected nodes with each node having at least one neighbour in the backbone. Nodes forming the backbone should be stay connected even when the nodes move. Smaller the size of the backbone, the better the maintenance will be. A new algorithm to build the backbone based on the computation of the minimum connected dominating set (MCDS) is proposed. An *ad hoc* network is modelled by a graph has disc unit UDG (Unit Graph Disc).Finding MCDS in a graph UDG is a *NP-Complet* problem. In order to reduce the complexity, we divide the problem in two steps: the first step computes the minimum dominating set (MDS) using a linear programming formulation and the second step determines the spanning tree of the MDS set, then we deduce the MCDS. Simulation results show that the solution given by our algorithm is very close solution to the one given by analysis. Moreover, the size of the bckbone is significantly lower compared with other proposed algorithms.

A distributed procedure for maintaining the connectivity of the backbone is also proposed. It is based on a simple exchange of the modified control message *hello*. A node which changes its position then will apply this procedure to be connected to the dorsal. A local maintenance of the backbone will be realized, only in the zone where the node is moved. Simulation results show the effectiveness and the reliability of this approach. Indeed, more than 90% of the nodes remain connected for a high mobility (average speed 30 *m/s*). Moreover, this procedure is not very sensitive to the scalability. The distributed nature makes maintenance procedure adapt well to the dynamic of the network structure caused by nodes movement.

In order to verify the improvement achieved by the presence of a backbone for routing protocols in mobile *ad hoc* networks. We make comparison of three existing protocols with and without a virtual backbone. Simulation results show that an improvement of 20% can be achieved for some protocols with high mobility.

In summary, this dissertation presents new results in many current problems regarding routing in wireless *ad hoc* networks.

REMERCIEMENTS

Tout d'abord, je voudrais exprimer ma gratitude à mon directeur de thèse le professeur Michel Kadoch, pour son suivi, son soutien et ses encouragements, ainsi que pour les conseils qu'il m'a prodigués tout au long de la préparation de cette thèse.

Ensuite, je tiens à témoigner ma gratitude au professeur Robert Hausler pour avoir accepté la présidence du jury. Je voudrais également remercier le professeur Zbigniew Dziong pour avoir juger cette thèse. Je tiens aussi à remercier le professeur Jean-Charles Grégoire d'avoir bien voulu être membre externe sur ce jury.

Je dois également un grand merci aux membres du LAGRIT (Laboratoire de Recherche en Gestion des Réseaux Informatiques et de Télécommunications) de l'École de technologie supérieure, avec qui j'ai eu le plaisir de travailler et d'avoir des discussions. J'adresse aussi mes remerciements aux amis que j'ai connus à Montréal.

Une dette spéciale de gratitude va à ma femme qui, pour sa patience, ses encouragements et son soutien, m'a permis de mener ce travail à terme.

Je remercie enfin mes parents et toute ma famille pour leur encouragement et leur soutien tout au long de mes années d'études et je leur dédie cette thèse. Elle est

dédiée également avec gratitude à tous mes professeurs de l'École Nationale d'Ingénieurs de Sfax (ENIS) en Tunisie, de l'Institut national de recherche scientifique (INRS) à Montréal et de l'École de technologie supérieure (ÉTS) à Montréal. Chacun d'eux a contribué à ce travail avec des perspectives importantes.

Merci à tous.

TABLE DES MATIÈRES

LISTE DES TABLEAUX

Page

LISTE DES FIGURES

LISTE DES ABRÉVIATIONS

ABR	Associativity-Based Routing protocol
ACK	Acknowledgment
AODV	Ad hoc On Demand Distance Vector protocol
AP	Access Point
APS	Ad hoc Positionning System
BS	Base Station
CA	Carrier Avoidance
CBR	Constant Bit Rate
CEDAR	Core Extraction Distributed Ad hoc Routing protocol
CSMA	Carrier Sense Multiple Access
CTS	Clear To Send
DARPA	Defense Advanced Research Projects Agency
DCF	Distributed Coordination Function
DS	Direct-Sequence
DSDV	Destination Sequenced Distance Vector protocol
DSR	Dynamic Source Routing protocol
DSSS	Direct Sequence Spread Spectrum
ETSI	European Telecommunications Standards Institute
GPS	Global Positioning System
HiperLAN	High Performance European Radio LAN
IEEE	Institute of Electrical and Electronical Engineer
IETF	Internet Engineering Task Force
IP	Internet Protocol

LAN	Local Area Network
LAR	Location Aided Routing
MAC	Medium Access Control
MANET	Mobile Ad hoc NETwork
MCDS	Minimum Connected Dominating Set
MPR	Multi-Point Relay
OLSR	Optimized Link State Routing
OFDM	Orthogonal Frequency Division Multiplexing
OSI	Open System Interconnection
PDA	Personal Digital Assistant
PRNet	Packet Radio Network
QoS	Quality of Service
RFC	Request For Comments
RREP	Route Reply
RREQ	Route Request
RERR	Route Error
RTS	Request To Send
RWP	Random Waypoint Model
ST	Spanning Tree
SN	Sequence Number
TCP	Transport Control Protocol
TORA	Temporary ORdered Algorithm
TTL	Time To Live
UDP	User Datagram Protocol
UDG	Unit Disc Graph

WRP	Wireless Routing Protocol
ZRP	Zone Routing Protocol

INTRODUCTION

Un réseau *ad hoc* sans fil est un ensemble de terminaux autonomes qui peuvent être mobiles et qui sont capables de communiquer directement entre eux sans aucune infrastructure existante. Une infrastructure peut être un point d'accès dans les réseaux locaux sans fil WLAN ou une station de base dans les réseaux cellulaires. La mobilité des terminaux résulte en une topologie du réseau changeante dans le temps. D'autre part, la portée de transmission radio étant limitée, le relayage est rendu nécessaire. Tous les terminaux doivent collaborer pour acheminer du trafic d'une source vers une destination. Les chemins utilisés et les nœuds intermédiaires à traverser sont déterminés par un protocole de routage dédié.

En raison de l'absence d'une infrastructure, les fonctionnalités réseau sont donc intégralement prises en charge par les terminaux eux-mêmes. Un terminal peut être un ordinateur de bureau, un ordinateur portatif, un PDA (*Personal Digital Assistant*), un téléphone portable ou tout autre équipement à communication sans fil. De tels types de terminaux engendrent immédiatement des contraintes en termes d'énergie, d'hétérogénéité, de capacité de traitement, etc.

Un réseau *ad hoc* est alors un réseau capable de s'auto-initialiser et s'auto-organiser à l'issue de mouvements des mobiles sans aucune infrastructure définie au préalable ou intervention humaine. Ces réseaux présentent un intérêt important et deviennent de plus en plus présents dans notre vie quotidienne [Bangnan *et al.* (2003)]. Avec l'avancement des recherches dans le domaine des réseaux et l'émergence des technologies sans fil (ex. *Bluetooth* et IEEE802.11). Plusieurs domaines d'application peuvent utiliser les réseaux *ad hoc* tels que par exemple lors de :

- un secours suite à une catastrophe naturelle où toutes les infrastructures de communication ont été détruites,

- une recherche de survivants au cours d'une avalanche,
- un chantier de construction,
- un travail coopératif ou une conférence,
- un partage d'applications et communications des équipements mobiles à domicile,
- une manifestation estivale,
- un combat militaire.

Problématique de recherche

Les réseaux *ad hoc* ont été conçu dans l'objectif de mettre à la disposition de l'usager plus de simplicité et de flexibilité. En effet, celui-ci peut communiquer à l'aide d'un terminal mobile à un autre terminal sans la présence d'un point d'accès intermédiaire. La portée d'un terminal étant limitée, et une communication entre une source et une destination doit passer par des nœuds intermédiaires. Pour réaliser cette tâche un protocole de routage doit alors déterminer l'ensemble de ces nœuds intermédiaires à traverser. Différentes approches ont été proposées pour concevoir les protocoles de routage dans les réseaux *ad hoc*, tels que les protocoles réactifs (DSR, AODV, TORA, etc.), les protocoles proactifs (DSDV, OLSR etc.) et les protocoles hybrides (ZRP, CEDAR). Ces protocoles assurent la détermination du meilleur chemin et l'acheminement des données entre la source et la destination en utilisant un routage multi saut. Le choix entre un protocole est un autre reste une tâche complexe, plusieurs facteurs entre en jeux comme le type d'application, le type de réseau (dense, clairsemé), etc. Toutefois, aucun de ces protocoles n'a prouvé son efficacité quand la mobilité est prise en considération.

Dans le cadre de nos travaux de recherche, nous nous sommes intéressés particulièrement aux problèmes reliés aux protocoles de routage et à la mobilité. L'absence d'une infrastructure fixe rendra la tâche de gérer les communications entre les terminaux plus difficile. En effet, les terminaux eux-mêmes doivent effectuer des

fonctionnalités telles que la découverte au voisinage, la recherche d'une route et l'acheminement du trafic. L'échange d'information entre les terminaux se fait par l'intermédiaire d'une interface radio. Si cet échange d'information n'est pas optimisé alors un problème connu sous le nom d'orage de diffusion (*Broadcast Storm Problem*) peut avoir lieu [Tseng *et al.* (2002)]. Par sa nature, la diffusion est une technique de transmission utilisée dans les systèmes sans fil.

Dans l'objectif de limiter la quantité d'informations échangées entre les terminaux, nous proposons de construire une structure partielle appelée dorsale virtuelle (*Virtual Backbone*). Cette dorsale sera utilisée conjointement avec les protocoles de routages actuels afin de les aider à améliorer leurs performances. Ainsi, nous considérons qu'une connaissance de la structure permet de simplifier la fonctionnalité du routage. Une telle structure sera chargée de :

- localiser les mobiles : chaque nœud appartenant à cette dorsale connaît à priori les nœuds à servir,
- maintenir l'information globale dans le réseau,
- échanger les paquets de contrôle,
- etc.

Une dorsale virtuelle est un ensemble de nœuds choisis de façon à ce qu'ils soient connectés pour être toujours en communication et elle doit couvrir tout l'ensemble des nœuds dans le réseau. La construction d'une dorsale requiert la connaissance de la position de chaque nœud. Dans ce travail de recherche, nous ne traitons pas les problèmes reliés à la position et nous supposons qu'une connaissance à priori de la position de chaque nœud peut être disponible au moment de l'établissement du réseau.

L'énergie consommée par l'interface radio du terminal comprend deux composantes : l'énergie consommée pour la réception et le traitement des messages reçus ainsi que l'énergie pour transmettre les messages à envoyer. Étant donné que l'énergie dissipée

pour la transmission est beaucoup plus importante que celle de la réception et du traitement, ce n'est alors uniquement que la puissance de transmission qui contribue à la consommation de l'énergie totale dans le terminal [Rappaport (1996)]. Les problèmes reliés à la consommation d'énergie ne sont pas traités dans notre recherche. Cependant, pour limiter la consommation d'énergie dans le terminal, il est alors recommandé de minimiser le nombre de messages transmis par l'interface radio. Minimiser le nombre de messages de contrôle revient à optimiser le nombre de nœuds qui vont échanger ces messages, et par la suite la dorsale doit avoir une taille minimale. La minimisation des messages de contrôle diminue la consommation de la bande passante dans le réseau.

Un réseau *ad hoc* est généralement modélisé par un graphe à disque unitaire UDG (*Unit Disc Graph*). Seuls les voisins se trouvant dans la zone de couverture de l'émetteur sont reliés par un lien direct avec ce dernier. Une dorsale est un ensemble de nœuds sélectionnés de façon à ce que chaque nœud dans le réseau possède au moins un voisin dans cet ensemble. Cet ensemble doit être connexe. L'ensemble de domination connexe de taille minimale MCDS (*Minimum Connected Dominating Set*) est un bon candidat pour construire cette dorsale. Dans la théorie des graphes, la détermination de l'ensemble MCDS dans un graphe UDG est un problème *NP-Complet* [Das et Bharghavan (1997)].

La nature dynamique d'un réseau *ad hoc*, engendrée par le mouvement des nœuds et l'instabilité des liens radio entre les nœuds nécessite une maintenance de la dorsale. La structure de la dorsale doit être stable tout au long du fonctionnement du réseau. Chaque nœud dans le réseau doit avoir au moins un voisin direct qui appartient à cette dorsale.

Méthodologie de recherche

Présentement, il existe plus de cinquantaine de propositions de protocoles de routage pour les réseaux *ad hoc*[1]. Certains protocoles ont été standardisé par l'IETF (OLSR, TBRPF et AODV)[2] d'autres sont en encore sous forme de *draft*. Nous avons étudié quelques-uns de ces protocoles et présenté leurs points positifs et négatifs. Un critère que nous considérons important dans les réseaux *ad hoc* consiste à étudier le comportement d'un protocole face à la mobilité. Nous avons choisi des protocoles qui utilisent différentes techniques de routage (réactive, proactive et hybride) et nous avons défini un certain nombre de métriques afin d'établir une comparaison rigoureuse.

L'étude de certains protocoles de routage développés pour les réseaux *ad hoc* nous a permis de conclure que la mobilité reste un handicap pour ces protocoles. Aucun de ces protocoles n'a prouvé son efficacité quand la mobilité augmente. Dans le but d'améliorer la performance des protocoles de routage face à la mobilité, nous proposons de construire une dorsale virtuelle. Une dorsale virtuelle est une structure partielle déterminée de façon à ce qu'elle couvre l'ensemble de tous les nœuds dans le réseau. L'ensemble des nœuds qui forme cette dorsale doivent être connexes.

De nature dynamique, la construction d'une dorsale n'est pas une tâche simple. La dorsale doit s'adapter au changement de la topologie causé par le déplacement des nœuds mobiles. De plus, l'absence d'une entité centralisée dans un réseau *ad hoc* fait en sorte que toute approche centralisée n'est pas pratique quand la taille du réseau augmente. Nous proposons alors une solution distribuée. Cette solution utilise un premier algorithme centralisé basé sur la détermination de l'ensemble MCDS dans la phase d'établissement du réseau. Cet algorithme garanti une taille minimale de la

---

[1] http://en.wikipedia.org/wiki/Ad_hoc_protocol_list

[2] OLSR (RFC-3626) TBRPF (RFC-3684) et AODV (RFC-3561)

dorsale. Un deuxième algorithme pour la maintenance de la dorsale sera alors appliqué une fois que la mobilité est introduite dans le réseau. La construction d'une dorsale est un phénomène rare alors que la maintenance se produit fréquemment : un nœud qui change de position doit alors appliquer la procédure de maintenance pour se connecter à la dorsale. Cette approche garanti le passage à l'échelle (*scalability*).

Nous allons proposer un nouveau algorithme appelé (MDS-based) pour déterminer l'ensemble des nœuds qui vont former la dorsale dans un graphe. Il est composé de deux étapes. La première étape consiste à déterminer l'ensemble de domination de taille minimale MDS (*Minimum Dominating Set*). La deuxième étape consiste à appliquer un algorithme pour déterminer l'arbre de recouvrement d'un graphe réduit : seul l'ensemble des nœuds MDS forme les sommets de ce graphe. A partir de cet arbre de recouvrement, nous extrairons la dorsale. Pour déterminer l'ensemble MDS, une formulation basée sur la programmation linéaire est développée pour garantir une taille minimale de cet ensemble. Trouver un arbre de recouvrement dans un graphe a fait l'objet d'une littérature abondante et il existe plusieurs algorithmes efficaces en termes de temps de calcul et de complexité tels que les algorithmes de Prim et Kruskal [Ahuja *et al.* (1993)]

Afin de vérifier et valider notre algorithme qui donne une approximation de l'ensemble MCDS, une analyse théorique basée sur un modèle probabiliste est développé. Le but est de vérifier que la taille de la dorsale est minimale comparée aux algorithmes existants. De plus, une comparaison avec deux autres approches basées sur deux techniques différentes (coloration dans un graphe, processus de marquage, etc.) élaborées par [Guha et Khuller (1998)] et [Wu et Li (2001)]. Ces deux approches donnent une meilleure approximation de l'ensemble MCDS et sont les plus utilisées comme référence dans la littérature. En plus de ces deux algorithmes, nous allons utiliser l'algorithme proposé par [Butenko *et al.* (2003)] avec ses deux versions (centralisée et distribuée) pour comparer la solution finale de notre algorithme.

Comme nous l'avons mentionné ci-haut, la construction d'une dorsale permet de structurer le réseau afin d'améliorer la performance des protocoles de routage, notamment en minimisant le nombre de messages de contrôle échangés dans le réseau et minimiser ainsi la bande passante et l'énergie consommées. La plupart des protocoles de routage dans les réseaux *ad hoc* utilisent l'inondation conventionnelle pour diffuser les messages de contrôle. C'est une technique simple qui ne demande pas d'algorithme complexe pour être implémentée. Un nœud qui reçoit un message pour la première fois va le retransmettre à ses voisins. Cette technique consomme beaucoup de bande passante (ressource précieuse dans les réseaux *ad hoc*) du fait qu'il y a beaucoup de trafic redondant. Aussi, elle risque de dégrader les performances du réseau. Les concepteurs du protocole OLSR ont développé une technique qui a pour objectif de réduire le trafic redondant. Cette technique consiste à déterminer un ensemble de relais multipoint [Jacquet *et al.* (2003)]. Seuls les relais auront à retransmettre les messages de contrôle. Nous allons comparer la performance de trois techniques de diffusion à savoir l'inondation conventionnelle, la technique de relais multipoint et la technique basée sur la dorsale. Différents paramètres ont été définis pour effectuer cette comparaison.

La mobilité des nœuds dans un réseau *ad hoc* entraînera une dynamique dans la topologie globale. Cette dynamique nécessitera une maintenance de la dorsale pour garder une structure stable tout au long de la durée de vie du réseau. Deux approches sont possibles pour effectuer cette maintenance. La première approche consiste à faire une reconstruction de la dorsale de façon périodique. La deuxième approche consiste à effectuer une maintenance locale. Dans notre recherche, nous adoptons la deuxième approche. En effet, la première approche est centralisée et elle n'est pas du tout pratique surtout dans le contexte des réseaux *ad hoc* étant donné qu'elle nécessite une connaissance globale de la topologie. De plus, le choix de la fréquence de mise à jour doit être justifié. Alors que la deuxième approche est distribuée. En d'autres termes, à chaque fois qu'un nœud change de position, il appliquera une procédure de maintenance

pour arranger la structure de la dorsale localement (dans la zone où le nœud va se retrouver et dans la zone où il était).

Dans le but de valider et de vérifier l'utilité d'une dorsale, nous appliquons certains protocoles de routage conçus pour les réseaux *ad hoc* en présence d'une dorsale. Nous comparons la performance de ces protocoles dans leur version originale et en présence de la dorsale virtuelle. Nous examinons leur comportement face à la mobilité.

Organisation de la dissertation

Le présent document est composé de six chapitres. Le chapitre suivant présente une introduction générale sur les réseaux sans fil. On distingue deux types de réseaux : les réseaux avec infrastructure et qui ne nécessitent pas de routage (les réseaux cellulaires et les réseaux locaux sans fil) et les réseaux sans infrastructure et qui nécessitent le routage, les réseaux *ad hoc*.

Le chapitre 2 sera consacré à une revue de la littérature où nous allons présenter et discuter les propositions qui ont étudié le problème de contrôle de topologie dans les réseaux *ad hoc*.

Dans le chapitre 3, nous allons présenter notre approche pour construire la dorsale virtuelle basée sur la détermination de l'ensemble MCDS dans un graphe UDG. Dans le but d'évaluer notre approche, une analyse théorique basée sur une approche probabiliste est développée, ainsi qu'une comparaison avec d'autres approches qui déterminent l'ensemble MCDS.

Nous allons effectuer dans le chapitre 4 une étude comparative des différentes techniques de diffusion dans les réseaux *ad hoc*. Pour comparer la performance de ces techniques, nous avons défini certains paramètres. Les techniques que nous avons

considérées sont : l'inondation pure, les relais multipoints utilisés par le protocole OLSR et la dorsale construite à partir de l'ensemble MCDS.

Nous allons proposer dans le chapitre 5 une procédure de maintenance distribuée pour garder la structure de la dorsale connexe et stable tout au long du fonctionnement du réseau.

Le chapitre 6 est consacré à la validation de l'approche proposée. Pour ce faire nous étudions certains protocoles de routage développés pour les réseaux *ad hoc* et nous comparons leur performance sans et avec la dorsale.

La conclusion générale dresse un bilan des travaux entrepris dans cette thèse, nos contributions, et enfin nos suggestions sur les perspectives futures.

CHAPITRE 1

LES RÉSEAUX SANS FIL

1.1 Introduction

Ces dernières années, les téléphonies sans fil et mobile ont connu un essor fulgurant. Cet essor est dû principalement à la commercialisation et l'émergence des appareils de communication (téléphone portable, ordinateurs portable, assistants personnel portable,.etc.) et la convergence des réseaux fixes et mobiles. Le domaine des réseaux mobiles ne se limite pas à la communication téléphonique et aux réseaux cellulaires. Un utilisateur ne veut plus se limiter à un lieu ou un temps pour consulter son courrier électronique ou naviguer sur Internet. Actuellement, les applications mobiles les plus utilisées sont la connexion réseau et les services de données associés. Une telle application est généralement réalisée à l'aide d'un réseau sans fil basé sur une infrastructure fixe : le terminal sans fil se connecte à un point d'accès du réseau via une communication sans fil, puis le reste de la connexion s'effectue sur l'infrastructure filaire réseau. Un utilisateur doit se trouver dans la zone de communication (zone de couverture) du point d'accès pour qu'il puisse se connecter au réseau. Sur une géographie étendue, la mise en place d'une infrastructure fixe conséquente est nécessaire avec l'établissement d'un nombre important de points d'accès. Ce déploiement est coûteux en termes de réalisation et matériel, de configuration et de temps [Lassous (2004)].

Une solution pour remédier à ces limitations consiste à utiliser les réseaux sans infrastructure fixe. Dans ce genre de réseau la portée de transmission des terminaux sans fil détermine ceux qui peuvent communiquer directement les uns avec les autres. La communication ne doit pas se limiter entre les terminaux à portée directe mais elle doit

s'étendre entre les terminaux à portée non-directe, ce qui nécessite que les terminaux soient capables de relayer les paquets. Ce réseau est capable de s'auto-initialiser et s'auto-configurer quand la topologie du réseau change suite au mouvement des mobiles sans aucune intervention d'un opérateur. Un tel réseau est appelé un réseau *ad hoc*. Il est possible d'obtenir les services Internet dans un réseau *ad hoc* en lui ajoutant des points d'accès connectés à l'Internet. Les réseaux *ad hoc* présentent de nombreuses avantages : ils permettent une mobilité sans contrainte, ils ne nécessitent aucune installation et sont faciles à déployer, ils sont flexibles et faciles à utiliser. Les applications des réseaux *ad hoc* sont nombreuses dans les domaines militaire et civil.

Vers la fin des années 1980, la recherche pour les applications militaires a été intensivement déployée à travers la planète. Le besoin de créer des standards ouverts dans le domaine des communications entre ordinateurs s'est fait sentir et un groupe de travail au sein de l'IETF[3], nommé MANET[4], a été formé pour établir des standards sur les protocoles et les spécifications fonctionnelles des réseaux *ad hoc*. L'objectif principal de ce groupe de travail consistait à fournir une standardisation pour la fonctionnalité de routage afin qu'elle supporte une infrastructure de réseau mobile capable de s'auto-organiser.

1.2 Les réseaux *ad hoc*

Dans un réseau *ad hoc*, une communication entre source et destination est généralement acheminée par des nœuds (terminaux) intermédiaires. La source et la destination ne sont pas nécessairement accessibles directement, les nœuds intermédiaires devant relayer le trafic de la source vers la destination. Les terminaux doivent s'auto-organiser et collaborer pour acheminer le trafic d'un nœud à un autre. Toutes les fonctions doivent

---

[3] IETF : *Internet Engineering Task Force*, organisme de standardisation des protocoles d'Internet

[4] MANET : *Mobile Ad hoc Network*

être déployées automatiquement sans paramétrages éventuels de l'utilisateur. À cet environnement multi saut à nœuds hétérogènes s'ajoute la mobilité potentielle de chaque nœud : chaque terminal est libre de se déplacer dans n'importe quelle direction et avec n'importe quelle vitesse, obligeant ainsi une adaptation dynamique du réseau.

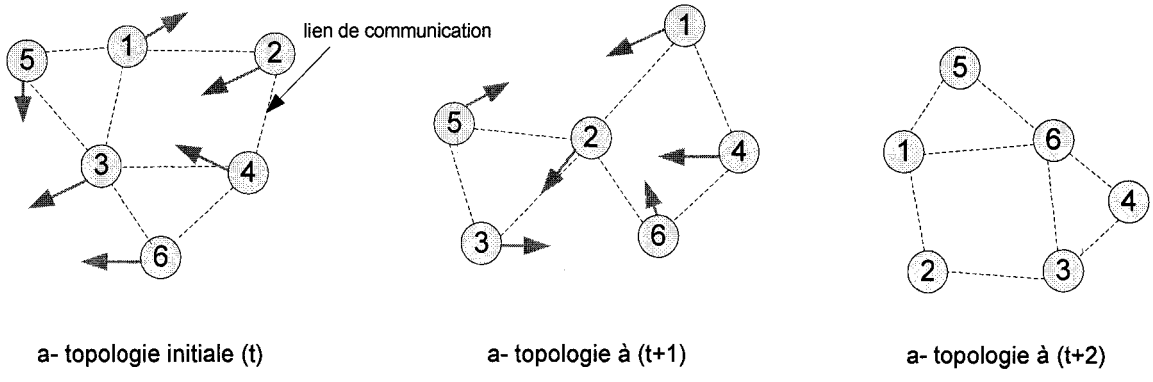


Figure 1 La topologie change dans un réseau *ad hoc*

La figure 1 illustre l'exemple d'une topologie dynamique pour un réseau sans fil *ad hoc*. Les liens radio entre les nœuds sont schématisés par des traits discontinus. Le sens de déplacement de chaque nœud est représenté par un vecteur. Au cours du temps, le déplacement des nœuds entraîne non seulement le changement de voisinage pour chaque nœud mais aussi la topologie globale du réseau.

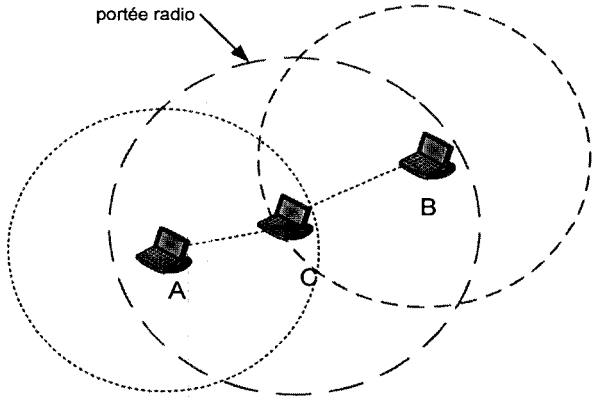


Figure 2 Principe de relayage dans un réseau *ad hoc* sans fil

Un réseau *ad hoc* utilise un médium radio avec des communications partagées. La portée des transmissions radio étant limitée, le relayage est alors obligatoire. Comme illustré dans la figure 2, le nœud *A* doit passer par *C* pour communiquer avec *B*. Dû aux évanouissements locaux du signal, le médium physique change et par conséquent la topologie change rapidement. Cet effet s'ajoute à la mobilité des nœuds.

Dans un réseau *ad hoc*, les nœuds sont équipés d'émetteurs et de récepteurs sans fil utilisant des antennes qui peuvent être omnidirectionnelles (*broadcast*), fortement directionnelles (point-à-point), probablement orientables, ou une combinaison de tout cela. Il y a une connectivité sans fil qui existe entre les nœuds sous forme de graphe multi saut aléatoire[5] qui dépend de la position des nœuds, de la configuration de leur émetteur-récepteur, des niveaux de puissance de transmission et de l'interférence entre les canaux.

1.2.1 Caractéristiques des réseaux *ad hoc*

Comme nous venons de le voir, les réseaux *ad hoc* possèdent des caractéristiques très spécifiques. Ces caractéristiques sont différentes de celles des réseaux filaires et des réseaux cellulaires avec station de base. Les principales caractéristiques sont les suivantes :

- **Absence d'infrastructure :** à l'opposé des réseaux cellulaires où une station de base est chargée de la gestion et de l'acheminement des appels, les terminaux dans les réseaux *ad hoc* sont eux-mêmes chargés par ces fonctionnalités. Pour que le réseau fonctionne, il faut que chaque nœud soit volontaire pour relayer le trafic des autres participants.

---

[5] RFC 2501 : www.ietf.org/rfc/rfc2501.txt

- **Communication multi saut :** une source et sa destination peuvent ne pas se retrouver à l'intérieur de la même portée radio. Dans ce cas, la communication source-destination nécessite le support des autres nœuds intermédiaires pour pouvoir véhiculer les paquets de données.
- **Liaisons à débit variable et à bande passante limitée :** les liaisons sans fil auront toujours une capacité inférieure à leurs homologues câblés. De plus, le débit réel des communications sans fil, après avoir déduit les effets des accès multiples, du *fading*, du bruit, des interférences, etc., est souvent inférieur aux taux de transfert maximum de la radio. Un système de communication radio se caractérise par un taux de perte assez élevé[6]. Ainsi, il est nécessaire que le trafic de gestion du réseau occupe le minimum de la bande passante possible afin de préserver le reste pour le trafic de données.
- **Topologie dynamique :** les nœuds sont libres de se déplacer arbitrairement, de telle sorte que la topologie du réseau, typiquement multi saut, peut changer aléatoirement et rapidement, et peut être constituée à la fois de liaisons unidirectionnelles et bidirectionnelles. Une infrastructure temporaire peut se former à n'importe quel moment et sous n'importe quelle forme. Un lien actif entre deux nœuds peut devenir brusquement non valide dû aux phénomènes d'évanouissement multi-chemin (*multipath fading*), à l'accès multiple, au bruit et à l'interférence d'autres transmissions. Un tel lien de communication est considéré non fiable, et on peut avoir recours à la retransmission pour avoir des services fiables.

---

[6] Dans un environnement sans fil on parle d'un taux d'erreur de l'ordre de 10^{-3} alors qu'il est normalement de l'ordre de 10^{-9}

- **Utilisation limitée d'énergie :** une partie des nœuds d'un réseau *ad hoc*, ou encore l'ensemble des nœuds, peut dépendre des batteries ou d'une autre source d'énergie limitée qui peut s'épuiser rapidement. Pour ces nœuds, le plus important est sans doute de mettre en place des mécanismes capables de limiter la consommation de l'énergie.
- **La qualité de service :** la qualité de service (QoS) définit un ensemble de règles permettant, entre autres, le transport de trafic isochrone ou temps réel tels que la téléphonie ou la vidéo. Les modèles proposés dans les réseaux filaires ne sont pas viables dans un environnement comme celui des réseaux *ad hoc*.
- **Sécurité physique limitée :** les réseaux sans fil mobiles sont généralement plus sensibles aux menaces physiques que ceux qui sont câblés et fixes. La possibilité accrue d'attaques par écoute du canal est réelle. Les techniques existantes pour la sécurité des liaisons sont souvent appliquées au sein des réseaux sans fil pour réduire les risques d'attaques. Il est donc indispensable de définir pour les réseaux *ad hoc* une politique qui assure une sécurité stricte basée sur des mécanismes tels que l'authentification, le contrôle d'intégrité, le chiffrement, etc.

1.2.2 Principe d'auto configuration

Afin d'assurer une auto configurationn, chaque nœud doit être capable de collecter et de maintenir suffisamment d'information sur la topologie du réseau [Murthy et Manoj (2004)]. Ces informations sont alors stockées dans deux types de tables :

- **Table des voisins :** chaque nœud maintient une liste de ses voisins à un saut obtenu grâce à l'envoi périodique d'un paquet de contrôle appelé *hello* par chaque nœud amorçant son existence. Seuls les voisins avec lesquels le nœud a un lien radio (les nœuds se trouvant dans la portée de transmission) seront retenus dans la table des voisins; la qualité d'un lien sera calculée en fonction du nombre de paquets reçus d'un voisin sur un intervalle de temps donné; cette métrique permet de garder tous les voisins pour lesquels le lien radio est stable.

- **Table de routage :** Cette table indique, pour chaque nœud, une destination possible dans le réseau et l'identité du nœud vers lequel les paquets doivent être relayés ainsi que le nombre de sauts restant avant d'atteindre la destination; une telle table sera construite à l'aide des paquets *hello* : au lieu d'indiquer sa seule identité dans ces paquets, comme indiqué précédemment, chaque nœud ajoute à son identité la liste de ses voisins à un saut; les nœuds recevant de tels paquets vont donc connaître leurs voisins à un saut et leurs voisins à deux sauts; en répétant ce principe, chaque nœud va connaître sa distance en nombre de sauts et l'identité du nœud voisin à qui il doit transmettre les paquets. Le but de la table de routage est de choisir la meilleure route jusqu'à la destination.

1.2.3 Accès au médium radio

Dans un réseau sans fil, deux mobiles peuvent communiquer en émettant des ondes radio. Ils se partagent un médium unique mais ne peuvent pas émettre en même temps. En effet, deux émissions simultanées sur le même canal entraînent une perte d'information à l'intersection des deux zones de communication (on parle alors de collision).

Pour pouvoir construire les tables de routage et réaliser le routage, il est indispensable de s'assurer que chaque nœud dans le réseau est capable d'accéder au canal radio afin de pouvoir émettre ses paquets. En effet, le canal est partagé entre les différentes entités et par conséquent un protocole d'accès au médium radio est nécessaire. Le protocole utilisé dans les réseaux radio par paquets (PRNet)[7] est basé sur la technique CSMA (*Carrier Sense Multiple Access*). Cette technique assure qu'un nœud ne pourra commencer une émission si un autre nœud, dans son voisinage, est en train d'émettre sur le canal. Pour ce faire, chaque nœud va écouter le médium avant son émission potentielle. Si ce

[7] Packet Radio Network

médium est occupé par une transmission en cours, alors la radio n'envoie pas son paquet et attend que le médium devienne libre. Cependant, avec cette technique, on ne peut pas empêcher deux transmissions de commencer en même temps. Afin de réduire la probabilité de transmission concurrente, un temps d'attente aléatoire est ajouté à la technique CSMA. Une fois que le médium est libre, le nœud qui cherche à émettre devrait attendre pendant un temps choisi aléatoirement. Si au bout de ce temps, le canal est toujours libre, alors il peut émettre sinon il recommence le processus (attendre que le médium soit libre puis choisir un temps d'attente aléatoire). Toutefois, des transmissions concurrentes peuvent avoir lieu dans le cas de nœuds cachés. Dans ce genre de situation, deux nœuds indépendants (non à la portée de communication) cherchent à communiquer avec le même destinataire. Par exemple, la figure 3 nous montre que les nœuds *A* et *B* veulent communiquer avec le même nœud *C*. Comme *A* et *B* ne sont pas à portée de communication, ils ne détectent pas leur activité réciproque sur le canal radio, et par conséquent, ils considèrent que le médium est libre et ils peuvent émettre leurs paquets. Ainsi, il y aura une collision au niveau du récepteur qui ne comprendra aucune des deux communications.

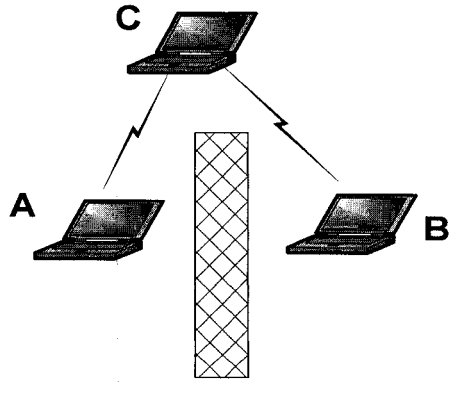


Figure 3 La collision se produit en *C* si *A* et *B* transmettent en même temps

La technique CSMA combinée à un temps d'attente aléatoire ne règle pas tous les problèmes. En effet, elle permet un accès simple au médium radio : aucune information ne peut être collectée ni mise à jour, et elle ne permet que la détection d'un signal sur le canal, le mécanisme étant complètement distribué (on n'a pas besoin d'une connaissance globale de tout le réseau). L'environnement voisin à lui seul suffit pour prendre la décision d'accès ou non au canal.

1.3 Classification des réseaux *ad hoc*

Les réseaux *ad hoc* peuvent être classés suivant deux catégories : les réseaux *ad hoc* statiques et les réseaux *ad hoc* dynamiques. Bien que la mobilité d'un terminal soit une des caractéristiques des réseaux *ad hoc* sans fil, il existe des situations où l'environnement du réseau est considéré comme statique. Dans un réseau statique, tous les terminaux ne se déplacent pas ou ils se déplacent rarement par rapport à leurs positions physiques. Alors que dans un réseau mobile la totalité ou la plupart des nœuds se déplacent d'une position à une autre de telle sorte que la topologie change. Un exemple de réseau *ad hoc* statique est le réseau de capteurs où les capteurs sont déployés sur des objets fixes. Un réseau *ad hoc* mobile peut être formé par des dispositifs (PDA, *laptop*, téléphone, etc.) qui sont utilisés par l'être humain et par conséquent ils ne sont pas fixes.

1.3.1 Les réseaux de capteurs sans fil

Les réseaux de capteurs sans fil (*Wireless Sensor Networks*) sont généralement formés pour surveiller un environnement éloigné ou inhospitalier comme par exemple : contrôle environnemental, mesure ou surveillance en climatologie, contrôle de processus, défense, etc. Un réseau de capteurs est caractérisé par :

- un emplacement non-déterministe des nœuds capteurs,
- un grand nombre de nœuds,
- une énergie limitée pour chaque nœud.

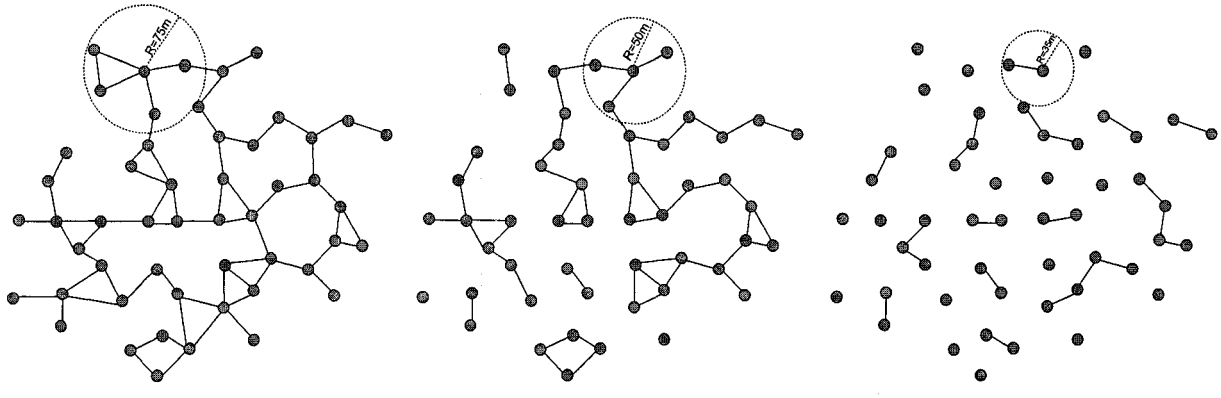
Une fois que les nœuds capteurs sont déployés, ils doivent réaliser la fonction de détection ainsi que la communication en utilisant leur propre énergie qui est généralement une batterie de faible autonomie. Ceci engendre un réseau avec une durée de vie limitée.

Maximiser la durée de vie d'un réseau est l'un des principaux objectifs des réseaux capteurs sans fil. Pour atteindre cet objectif, l'énergie provenant de la batterie doit être suffisante pour réaliser les tâches de détection, de routage et de communication. Une étude réalisée par Benini *et al.* [Benini *et al.* (2000)] a montré que les batteries ont approximativement deux fois plus d'autonomie si elles sont déchargées pendant une courte durée avec une période assez longue de repos. Les résultats de cette étude ont révélé que la durée de vie d'un réseau peut croître si on arrive à trouver de multiples ensembles exclusifs de nœuds actifs où chaque ensemble contient des nœuds capteurs qui couvrent la zone de surveillance. Slijepcevic et Potkonjak ont proposé [Slijepcevic et Potkonjak (2001)] une heuristique pour déterminer des ensembles mutuellement exclusifs des nœuds capteurs où chaque ensemble couvre une zone entière de surveillance. Ce problème est connu sous le nom *Set K-Cover Problem.* Leurs résultats démontrent qu'on peut économiser l'énergie d'une manière significative en maximisant le nombre des ensembles disjoints.

Dans [Zhao *et al.* (2002)], les auteurs ont développé une technique permettant de balayer l'énergie résiduelle dans un réseau capteur qui détermine approximativement la distribution de l'énergie restante dans le réseau. Cette technique utilise des algorithmes de localisation qui sont par la suite agrégés pour construire un système de balayage composite. Cette technique peut être utilisée pour informer les utilisateurs sur les régions ayant une faible énergie, ceux-ci pouvant décider d'ajouter le nouveau capteur dans cette région ou de prévoir de nouvelles stratégies pour aviser le niveau de puissance résiduelle. D'autre part, les auteurs ont montré aussi qu'une grande partie de l'énergie

dissipée pour la communication, utilisée par l'interface radio, est beaucoup plus importante que celle utilisée pour le traitement.

Dans un réseau capteur les nœuds possèdent un rayon de couverture limitée (R) et par conséquent le routage se fait par multi saut et le relayage se fait au moyen des nœuds intermédiaires. Par la suite, les informations recueillies par chaque nœud doivent être envoyées à la station de base et amalgamées pour obtenir des informations globales et significatives. Cependant, la minimisation de l'énergie pour le routage constitue un autre problème dans les réseaux de capteurs sans fil.



a- R=75m b- R=50m c- R=35m

Figure 4 Un réseau avec 50 nœuds

La figure 4 illustre la connectivité d'un réseau comprenant 50 nœuds pour différents rayons de couverture R est égale à respectivement 75m, 50m et 35m. Dans cet exemple, les nœuds sont statiques et possèdent le même rayon de transmission. Nous remarquons que si le rayon de couverture passe de 75m à 35m, nous obtenons un réseau non-connecté.

1.4 Les réseaux sans fil sans routage

1.4.1 Les réseaux cellulaires

Les réseaux de téléphonie cellulaire ont été conçus pour établir des communications téléphonie sans fil. Toutefois, ils offrent également des services de transmission de données très intéressants pour l'échange de message courts SMS (*Short Message Service*) ou de données en mode circuit (9,6 ou 14,4 kbps pour le GSM). De plus, L'évolution de la norme GSM a permis de définir un nouveau réseau : le GPRS (*General Packet Radio Service*) qui fonctionne en mode paquet jusqu'à un débit de 171,2 kbps. Enfin, la standardisation d'une nouvelle modulation (8PSK) a donné naissance à EDGE (*Enhanced Data for GSM Evolution*) permettant d'atteindre plus de 300 kbps [Vivier E. (2004)].

La norme GSM (*Global System for Mobile communication*) est largement utilisée en Europe et en Asie et elle ne constitue pas la seule norme pour les réseaux cellulaires. D'autres normes de téléphonie numérique existent comme l'IS-95 (*Interim Standard*) en Amérique du nord et le PDC (*Personal Digital Cellular*) au Japon.

La topologie utilisée est la topologie avec stations de base BS (*Base Station*). Une station de base est un ensemble d'émetteurs récepteurs munis d'une ou plusieurs antennes. Chaque station de base couvre une cellule indépendante. Un réseau cellulaire est formé d'une multitude de stations de base. Les BTS sont gérées par un contrôleur de stations de base BSC (*Base Station Controller*) qui assure également la concentration du trafic. Chaque BSC est connecté à un TCU (*TransCoder Unit*) qui rend compatible le réseau GSM avec les réseaux numériques fixes publiques par une adaptation du débit des circuits de parole. Les cellules adjacentes doivent utiliser des fréquences différentes afin d'éviter les interférences entres les communications. Les terminaux mobiles ne peuvent pas communiquer entre eux directement, ils doivent passer par la station de base même s'ils sont proches physiquement.

1.4.1.1 Les réseaux *ad hoc* versus les réseaux cellulaires

La figure 5–a illustre la topologie d'un réseau cellulaire ayant une infrastructure dépendante, alors que la figure 5–b montre un réseau *ad hoc* sans fil basé sur le multi saut. L'établissement d'un chemin entre la source *S* et la destination *D* doit passer par les stations de base dans un réseau cellulaire. Alors que dans un réseau *ad hoc*, le chemin entre *S* et *D* s'établit par l'intermédiaire de nœuds mobiles.

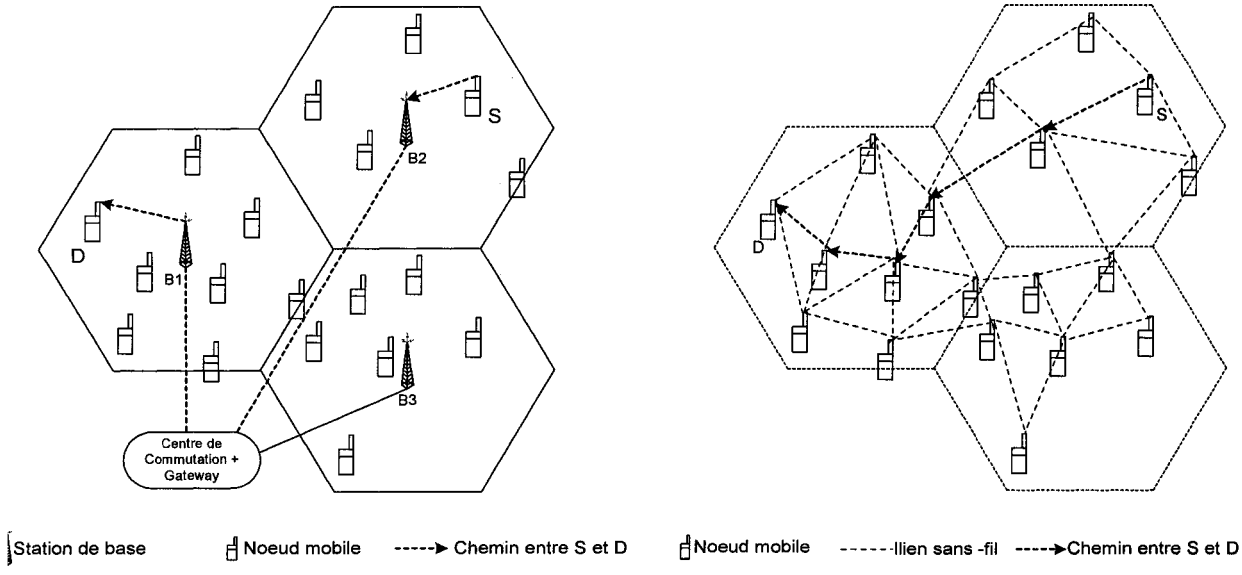


Figure 5 a– réseau cellulaire b– réseau *ad hoc* sans fil

Les différences majeures entre les réseaux cellulaires et *ad hoc* sont résumées dans le Tableau I. L'existence d'une station de base simplifie le routage et la gestion de ressources dans un réseau cellulaire. Dans un réseau cellulaire, la décision de routage est réalisée de manière centralisée. Alors que dans un réseau *ad hoc* sans fil, le routage et la gestion de ressources sont réalisés de manière distribuée dans laquelle tous les nœuds se coordonnent pour établir une communication. Ceci demande à ce que chaque nœud soit plus intelligent alors qu'il doit fonctionner comme un hôte pour transmettre et recevoir les données et comme un routeur pour acheminer les paquets entre les nœuds. Dans un

réseau *ad hoc* les nœuds sont plus complexes que ceux des réseaux cellulaires [Murthy et Manoj (2004)].

Tableau I

Différence entre les réseaux cellulaires et les réseaux *ad hoc*

	Réseaux cellulaires	Réseaux *ad hoc*
Caractéristiques physiques	Basé sur une infrastructure fixe	Sans infrastructure
	Domaine d'application inclut civil et commercial	Urgence, opération de sauvetage, champs de bataille, travail de collaboration
	Coût élevé et temps de déploiement	Rapide et coût d'installation faible
	Coût élevé pour la maintenance (*backup*, *staffing*, etc.)	Fonctions d'auto-organisation et maintenance sont incluses dans le réseau
Gestion et routage	Lien à simple saut	Lien à multiple saut
	Routage centralisé	Routage distribué
	Commutation de circuit	Commutation de paquets
	Largeur de bande garantie (désignée pour le trafic de voix)	Canal radio partagé (plus convenable pour le trafic de données Best Effort)
	Facile pour employer la réservation de la bande passante	La réservation de la bande passante demande des protocoles de contrôle d'accès assez complexes
Déploiement	Hôte mobile ne sont pas trop complexe	Hôte mobile demande plus d'intelligence (doit être capable d'acheminer les paquets)
	Principaux objectifs du routage et l'admission d'appel sont de maximiser le taux d'acceptation d'appel et de minimiser le taux d'appel perdu	Principal objectif est de trouver les chemins avec un minimum d'*overhead* et une reconfiguration rapide en cas de chemins brisés.
	Largement déployé et actuellement dans la 3ème génération	Plusieurs problèmes restent à investiguer pour un déploiement commercial bien qu'ils soient très utilisés en militaire.

1.4.2 Les technologies sans fil

La fin des années 90 a été marquée par la popularité des réseaux locaux informatiques. D'une part, l'utilisation courante du réseau local chez les particuliers, due en grande partie à Internet et, d'autre part, l'arrivée en masse des ordinateurs et autres matériels mobiles. Pour cela il faut trouver une technologie permettant de simplifier le câblage du réseau chez un particulier et de préserver la mobilité des produits portables. Un seul principe permet de concilier les deux; le sans fil. La solution peut paraître simple, autant elle est complexe à mettre en place pour plusieurs raisons. En effet, il faut tout d'abord développer une technologie proposant le plus d'avantages possibles comme une longue portée, le franchissement d'obstacles (murs) et surtout un prix bas.

Deux solutions sont proposées. La première consiste à utiliser l'infrarouge. Malgré son coût faible, l'infrarouge présente deux inconvénients majeurs : un temps de réaction très lent et l'obligation de maintenir l'émetteur dans la zone de réception sans obstacle en chemin. La deuxième est la transmission par radio, qui ne possède pas les inconvénients rencontrés avec la transmission infrarouge et dont les coûts de production ne cessent de baisser. Deux normes utilisant la technologie de transmission par radio sont déjà sur le marché, soit le *Bluetooth* et l'IEEE 802.11.

Ces standards n'ont pas été conçus dans l'objectif d'une utilisation complètement *ad hoc*. Ces technologies ne permettent que la communication entre terminaux à portée radio directe. Afin de pouvoir communiquer entre n'importe quelle paire de mobiles dans le réseau, il est indispensable de mettre en place un protocole de routage dédié. Un protocole de routage assure le relayage multi saut.

1.4.2.1 *Bluetooth*

Bluetooth est le premier standard pour une communication point-à-point à simple saut capable d'échanger les données et la voie. *Bluetooth* a pour objectif de faire disparaître les câbles entre les divers équipements numériques (périphériques d'ordinateurs tels que clavier, imprimante, modem, ou encore appareil photo numérique, PDA, walkman, etc.). Les équipements *Bluetooth* ont donc des portées et des débits assez limités, ainsi qu'une consommation électrique en rapport.

Un réseau *Bluetooth* forme des *piconets* composés par un groupe de nœuds dans une petite région géographique où chaque nœud est capable de rejoindre à un saut n'importe quel autre nœud dans son groupe. Le groupe de travail *Bluetooth* SIG (*Bluetooth Special Interest Group*) en a élaboré les spécifications. Ce réseau permet des communications ayant une portée d'une dizaine de mètres. Le *Bluetooth* opère dans la bande de fréquences des 2.4 GHz.

Les réseaux *Bluetooth* sont construits de manière centralisée. Un maître élu peut prendre en charge jusqu'à huit esclaves et forme ainsi un piconet. Dans un piconet, c'est le maître qui contrôle toutes les transmissions. Les esclaves ne peuvent émettre des paquets que s'ils ont été invités par le maître. L'accès au médium est un protocole de type jeton sans collision.

Le débit maximum dans un piconet est de 1 Mbits/s et une communication bidirectionnelle entre deux terminaux est d'au plus 433kbits/s. Un tel piconet peut constituer un réseau *ad hoc* à deux sauts dans lequel un terminal est à un saut de tous les autres. Pour construire un réseau multi saut, il est possible d'interconnecter plusieurs piconets, donnant lieu à un *scatternet*. La figure 6 donne un exemple de *scatternet*. Un esclave alterne entre les piconets 1 et 2 afin d'en assurer la liaison et le mobile esclave du piconet 2 est aussi maître dans le piconet 3 (le processus d'association est réversible,

il est possible de quitter temporairement un piconet puis de le rejoindre à nouveau et de revenir à la situation initiale).

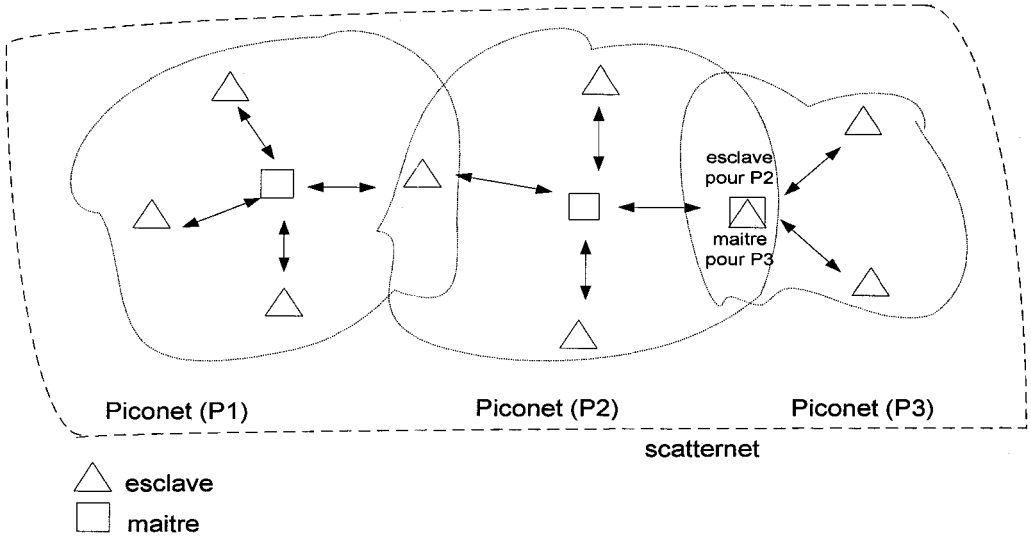


Figure 6 Un *scatternet* piconet nuage [Pujolle (2003)]

Bluetooth présente bel et bien une forme de réseau *ad hoc* multi saut. Mais du fait de l'utilisation ciblée pour un usage personnel, ces réseaux restent assez statiques et leur étendue reste limitée.

1.4.2.2 IEEE 802.11

Dans cette section, nous présentons les caractéristiques du standard IEEE 802.11 utilisées dans les réseaux *ad hoc* et ses performances dans un tel contexte. IEEE 802.11 a été développé en 1997 pour définir une couche physique et une couche accès au médium pour les réseaux locaux sans fil. En 1999, une extension de 802.11 a donné lieu aux standards 802.11a et 802.11b. Le 802.11a utilise la technique de modulation OFDM. Il fonctionne dans la bande de fréquence de 5 GHz et offre un débit de 6 Mbits/s. Alors que le 802.11b utilise la technique de modulation DSSS, il fonctionne dans la bande de

fréquence de 2,4 GHz et offre un débit jusqu'à 11 Mbits/s. Par la suite, un autre standard 802.11g[8] a été créé pour atteindre un débit de 54 Mbits/s. Un nouveau standard 802.11n est en cours de standardisation pour atteindre un débit de l'ordre de quelques centaines de Mbits/s [Stallings (2005)].

Deux modes d'opération ont été décrits dans ce standard : le premier est le mode infrastructure et le deuxième est le mode *ad hoc*. Le mode infrastructure requiert une station de base alors que le mode *ad hoc* ne requiert pas de station de base. Par conséquent, deux mobiles à portée de transmission l'un de l'autre pourront communiquer sous ce dernier mode sans passer par l'intermédiaire d'une station de base. Le mode infrastructure utilise la fonction d'accès PCF (*Point Coordination Function*) qui sert pour supporter les trafics synchrones, c'est-à-dire les trafics temps réel. Le mode *ad hoc* utilise le service DCF (*Distributed Coordination Function*). Avec DCF, si un mobile détecte une activité sur le canal, le mobile va tirer une valeur aléatoire (*backoff*) dans une fenêtre de contention CW (*Contention Window*) puis il va décrémenter cette valeur, tant que le canal est libre. Une fois que le *backoff* atteint la valeur zéro, le mobile envoie son paquet. Durant la décrémentation du *backoff*, si le mobile détecte encore de l'activité sur le canal, il va arrêter cette décrémentation et continuer à utiliser ce *backoff* lors de la tentative d'émission suivante jusqu'à ce que le canal redevienne libre [Pujolle (2003)].

La spécification adoptée IEEE 802.11e[9] propose le support de la QoS dans les réseaux sans fil avec une nouvelle fonction de contrôle EDCA (*Enhanced Distributed Channel Access*), considérée comme la nouvelle version de la fonction DCF, et une fonction de coordination hybride HCF (*Hybrid Coordination Function*). EDCA décrit quatre catégories de trafics TC (*Traffic Categories*). Les priorités sont contrôlées par les

---

[8] Approuvé le 12 juin 2003

[9] http://en.wikipedia.org/wiki/IEEE_802.11e

stations en modifiant le schéma d'accès de base (DCF). HCF est plus flexible que la fonction PCF. Il est utilisé par les points d'accès pendant la période d'accès contrôlée CAP (*Control Access Period*), qui peut commencer à n'importe quel moment durant la "*superframe*". Autrement dit, il permet d'accéder au médium pour faire passer un trafic ayant des contraintes de QoS.

Pour pallier au problème des nœuds cachés, décrit dans la section 1.2.3, le standard IEEE 802.11 donne la possibilité d'ajouter des paquets de contrôle avant d'échanger un paquet de données. Avant de transmettre son paquet, un émetteur envoie un paquet de contrôle appelé RTS (*Request To Send*). Ce paquet a pour rôle de réserver le lien radio en empêchant les voisins de transmettre suite à la réception de ce paquet. Un récepteur qui reçoit le RTS répond par un paquet de contrôle CTS (*Clear To Send*) s'il est en mesure d'accepter la communication. Le paquet CTS a le même effet que le paquet RTS, c'est-à-dire qu'il va réserver le lien médium et empêcher tous les nœuds voisins du récepteur d'émettre. À la réception du paquet CTS, l'émetteur sait que la communication va être préservée et qu'il peut transmettre ses paquets de données. En utilisant ce mécanisme, deux nœuds cachés désirant transmettre à la même destination seront alors au courant de leurs transmissions concurrentes. Dans le cas des nœuds cachés, s'il y a collision, elle se fera au niveau des paquets de contrôle et non pas au niveau des paquets de données; moins de bande passante sera consommée étant donné que les paquets de contrôle ont une taille beaucoup plus petite que les paquets de données.

Ce mécanisme est très efficace dans le cas des réseaux sans fil avec station de base, mais pas autant dans le cas des réseaux *ad hoc*. En effet, des collisions peuvent se produire même au niveau des paquets de données. Dans l'exemple de la figure 7, considérons le scénario suivant : *A* veut communiquer avec *B* et *C* avec *D*; *A* envoie un RTS; *B* reçoit ce RTS et pendant que *B* envoie un CTS, *C* envoie un RTS; comme *B* et *C* sont en train d'émettre, ils ne se rendent pas compte de leur émission respective; *D* reçoit le RTS de C et envoie un CTS; pendant ce temps *A* a déjà reçu le CTS de *B* et suppose que les liens

sont réservés, il va donc envoyer des paquets de données. En même temps, *C* reçoit le CTS de *D* et pense que les liens sont réservés; il envoie alors les données à *D*. Or ses données vont rentrer en collisions avec les données provenant de *A* au niveau du nœud *B* et ainsi *B* ne sera pas en mesure de comprendre la communication de *A*.

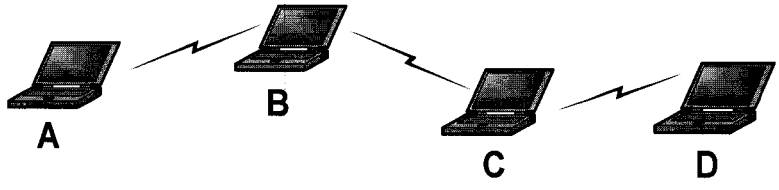


Figure 7 Méthode d'accès DCF

Un autre problème se pose très souvent dans le cadre des réseaux *ad hoc*. C'est le problème des stations exposées. Ce problème, à l'opposé des stations cachées qui se traduit en termes de sur-utilisation du médium, correspond à une sous-utilisation du médium radio. Considérant le même réseau de la figure 7, si *B* veut communiquer avec *A* et *C* veut communiquer avec *D*, *B* va accéder en premier au canal radio. *C*, situé dans la portée de transmission de *B*, va détecter de l'activité sur le canal et va donc s'empêcher d'émettre d'après le fonctionnement de DCF. Or, *C* aurait pu communiquer en même temps que *B*, dans le cas où les nœuds destinataires sont suffisamment éloignés des émetteurs de transmissions concurrentes, *A* de *C* et *D* de *B*.

Le standard 802.11b utilise différents débits de transmission. Les paquets en mode diffusion, comme les paquets de contrôle (RTS, CTS, etc.), sont envoyés à un ou deux Mbits/s alors que les paquets *unicast* peuvent être envoyés à 11 Mbits/s. Toutefois, si le débit augmente la portée diminue; un paquet envoyé à deux Mbits/s va être reçu par des nœuds plus éloignés que le même paquet envoyé à 11Mbits/s. Cependant, les routes construites avec les paquets diffusés ne sont pas directement exploitables à des débits

plus élevés. Ce problème est connu sous le nom de problème de la zone grise [Lundgren *et al.* (2002)].

Le problème de la zone grise n'est pas l'unique problème issu de l'utilisation de débits différents dans 802.11. Un autre problème a été mentionné dans [Heuse *et al.* (2003)]. Si plusieurs flux sont en contention pour l'accès au médium, la technique DCF cherche à donner à chaque paquet les mêmes chances pour y accéder. Ceci n'est possible que si tous les mobiles se trouvent dans la même zone de communication. Or, dans un réseau *ad hoc*, les nœuds peuvent être étendus géographiquement et certains mobiles sont indépendants et non à la portée de communication. Ces mobiles indépendants peuvent réduire fortement l'égalité des chances dans l'accès au canal ou l'égalité de transmissions réussies. Par exemple dans la figure 7, les nœuds *A* et *C* sont indépendants et n'ont pas conscience de leur transmission réciproque. Le nœud *B* est alors sujet à des collisions, et il n'est pas en mesure de comprendre la communication provenant de *A*, par conséquent le débit entre *A* et *B* est beaucoup plus faible qu'entre *C* et *D*.

1.5 Les protocoles de routage

Dans un réseau *ad hoc*, un protocole de routage a pour rôle d'établir et de maintenir une route d'une source vers une destination. Les caractéristiques des réseaux *ad hoc* rendent l'utilisation des protocoles filaires classiques non-utilisable. En effet, ces protocoles ont été conçus pour des réseaux ayant des topologies statiques et les nœuds possèdent suffisamment de ressources ce qui n'est pas le cas pour les réseaux *ad hoc*.

Les protocoles de routage dans les réseaux *ad hoc* sans fil sont classés en deux catégories : proactif (*table-driven*) et réactif (à la demande). Dans le routage proactif, chaque nœud doit maintenir une table de routage pour toutes les destinations et cette table doit être mise à jour de façon périodique. Dans le routage réactif, les nœuds n'ont pas de table de routage, c'est uniquement quand la source désire communiquer avec la destination qu'une route est déterminée. Ceci permet de réduire la taille des tables de

routage maintenues par les nœuds. Il existe une troisième classe : hybride. Cette classe est la combinaison des deux autres classes.

Tableau II

Protocoles de routage pour les réseaux *ad hoc*

Catégorie	Exemple de protocole de routage	Type
Proactif	OLSR, DSDV	plat
Réactif	AODV, ABR, DSR, TORA	plat
Hybride	ZRP	hiérarchique

Pour des raisons de mise à l'échelle, les protocoles de routage réactifs gagnent plus de terrain et attirent plus d'attention que les protocoles proactifs. Généralement, les protocoles réactifs utilisent l'inondation (*flooding*) pour trouver une route de la source vers la destination et l'inondation peut engendrer le phénomène d'orage de diffusion (*Broadcast Storm Problem*) tel que décrit dans [Tseng *et al.* (2002)].

L'économie d'énergie de la batterie est un autre problème pour les réseaux *ad hoc* sans fil. Plusieurs travaux de recherche ont été menés pour minimiser l'énergie totale consommée [Wan *et al.* (2001); Wieselther *et al.* (2000)]. Si l'énergie diminue dans un nœud critique le fonctionnement de tout le réseau sera alors perturbé. Pour augmenter la durée de vie du réseau, il est nécessaire d'étudier aussi bien l'énergie totale consommée que la consommation de l'énergie [Stallings (2005)].

D'autres protocoles de routage on été proposés et leur application dépend de l'utilisation d'un système de positionnement tel le système GPS (*Global Positionning System*). Pour pouvoir utiliser ces protocoles, chaque terminal mobile doit être équipé d'un tel système

ce qui n'est pas toujours le cas. La plupart des terminaux actuels ne sont pas équipés de GPS et l'intégration de ce système engendre un coût additionnel. Aussi, ce système n'est efficace que pour les applications en plein air (pour des applications *indoor* le signal GPS est faible et n'est pas efficace). Le tableau III donne des exemples de protocole de routage basés sur l'utilisation du GPS.

Tableau III

Protocole de routage pour les réseaux dynamiques

Catégorie	Exemple de protocole de routage	Type
Proactif	GeoCast, DREAM, GPSR	géographique
Réactif	LAR	géographique

1.6 Conclusion

Dans ce chapitre, nous avons présenté un aperçu général sur les réseaux *ad hoc* mobiles, leurs caractéristiques, leurs applications, leurs types etc. Nous avons aussi présenté les technologies sans fil qui peuvent être utilisées comme infrastructure pour assurer la communication entre les mobiles dans le réseau. En effet, les différentes technologies proposées offrent différentes portées de communication et seuls les terminaux à portée de communication les unes des autres vont pouvoir échanger directement les messages. L'absence d'infrastructure et la nature dynamique de la topologie dans le réseau nécessitent l'existence d'un protocole de routage qui rendra la communication possible entre tous les nœuds du réseau. Un protocole de routage dans un réseau *ad hoc* doit réagir rapidement pour permettre la continuation de la communication.

Dans le chapitre suivant, nous allons effectuer une revue plus détaillée sur les protocoles de routage proposés dans littérature. Par la suite, nous allons discuter les approches

proposées pour la configuration et l'adressage dans un réseau *ad hoc*. Finalement, nous allons discuter les différentes approches pour effectuer le contrôle de la topologie dans les réseaux *ad hoc*.

CHAPITRE 2

REVUE DE LA LITTÉRATURE

2.1 Introduction

Ce chapitre est composé de trois parties : dans la première partie, nous allons présenter certains protocoles de routage développés dans le cadre du groupe de travail MANET de l'IETF. Ces protocoles sont définis au niveau IP et sont donc indépendants des couches physiques et MAC. Le routage IP permet en particulier une interconnectivité aisée avec toutes sortes d'autres réseaux ou matériel. Les protocoles présentés sont parmi les plus représentatifs des diverses techniques utilisées pour le routage dans les réseaux *ad hoc*.

Un protocole de routage a pour but d'établir et de maintenir des routes entre sources et destinations, pour que les messages soient correctement délivrés. Les caractéristiques des réseaux *ad hoc* rendent l'utilisation des protocoles filaires classiques inadéquate. Les protocoles de l'Internet ont été conçus pour des réseaux ayant des topologies statiques, et sont exécutés par des nœuds dotés de suffisamment de ressources. Les protocoles de routage dans les réseaux *ad hoc* opèrent dans des réseaux dont les changements de topologie sont fréquents, et sont exécutés sur des équipements ayant des contraintes de ressources (de batterie, de mémoire, de CPU, etc.).

La deuxième partie discute le problème de configuration et adressage dans les réseaux *ad hoc*. La nature des réseaux *ad hoc* nécessitent de nouvelles approches pour effectuer la configuration et l'adressage. Nous allons présenter les principales approches étudiées dans la littérature.

Dans la troisième partie de ce chapitre, nous nous intéressons à l'étude du contrôle de topologie. Depuis quelques années, plusieurs travaux de recherche ont été consacrés à l'étude du contrôle de la topologie dans les réseaux *ad hoc*. Ces travaux visent à

maintenir une topologie adéquate tout au long du fonctionnement du réseau. Le but est d'atteindre un ensemble d'objectifs comme :

- effectuer la mise à jour des informations de routage de manière plus efficace,
- échanger les paquets de contrôle plus rapidement,
- utiliser de manière optimale la bande passante dans le réseau tout en éliminant le trafic redondant,
- réduire la consommation d'énergie.

2.2 Protocoles de routage dans les réseaux *ad hoc*

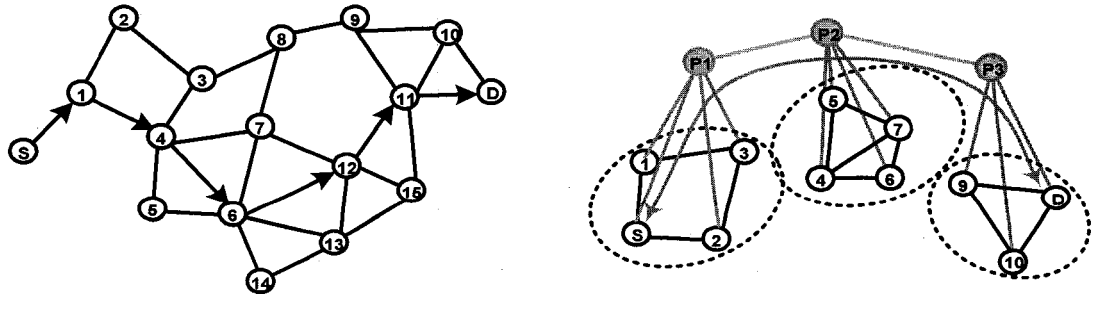
L'étude et la mise en œuvre de protocoles de routage dans les réseaux *ad hoc* doivent tenir compte d'un certain nombre de points :

- minimiser la charge du réseau : l'optimisation des ressources du réseau renferme deux autres sous-problèmes comme par exemple les boucles de routage et la concentration du trafic autour de certains nœuds ou liens.
- assurer un routage efficace : la stratégie de routage doit créer des chemins optimaux et pouvoir prendre en compte différentes métriques de coûts (bande passante, nombre de liens, ressources du réseau, etc.). Si la construction des chemins optimaux est un difficile problème, la maintenance de tels chemins peut devenir encore plus complexe et la stratégie de routage doit assurer une maintenance efficace des routes avec le moindre coût possible.
- minimiser le latence : certaines applications peuvent être sensibles au délai. Il est important alors de minimiser le délai de bout en bout.

2.2.1 Routage plat ou hiérarchique

Les protocoles de routage pour les réseaux *ad hoc* peuvent être classés suivant différents critères. Le premier concerne le type de voisin qu'ils possèdent et les rôles qu'ils accordent aux différents mobiles.

- **Les protocoles de routage plat** considèrent que tous les nœuds sont égaux (voir figure 8-a). La décision d'un nœud d'acheminer du trafic pour un autre dépendra de sa position et pourra être remise en cause dans le temps.
- **Les protocoles de routage hiérarchique** fonctionnent en confiant aux mobiles des rôles qui varient de l'un à l'autre. Certains nœuds sont alors élus pour assumer des fonctions spécifiques qui conduisent à une vision en plusieurs niveaux de la topologie du réseau. Par exemple, un nœud servira de passerelle pour un certain nombre de nœuds qui lui seront attachés. Le routage sera alors simplifié, puisqu'il se fera de passerelle à passerelle, jusqu'à celle directement attachée à la destination, figure 8-b. Dans ce type de protocole, les passerelles supportent la majeure partie de la charge de routage. Dans les réseaux où certains nœuds s'avèrent très sédentaires et disposent suffisamment d'énergie (par exemple, un réseau d'ordinateurs portables où certains sont reliés au secteur, des stations de base disposées là pour garantir la connectivité, etc.).



a- routage plat b- routage hiérarchique

Figure 8 Routage plat *vs* routage hiérarchique

2.2.2 État de lien ou vecteur de distance

Une autre classification, inspirée du monde filaire, est possible pour les protocoles de routage *ad hoc* :

- **Les protocoles à état de lien** cherchent à maintenir dans chaque nœud une image plus ou moins complète du réseau où figurent les nœuds et les liens les reliant. À partir de cette image, il est possible de construire les tables de routage. Un des avantages de ce type de protocole est leur capacité à pouvoir trouver des routes alternatives lorsqu'un lien n'est plus accessible. Il est même possible d'utiliser simultanément plusieurs routes pour la même destination, augmentant ainsi la répartition de la charge et la tolérance aux pannes dans le réseau. Cependant, dans le cas d'un réseau étendu, la quantité d'informations à stocker et à diffuser devient considérable.
- **Les protocoles à vecteur de distance**, plutôt que de maintenir une image complète du réseau, ne conservent que la liste des nœuds du réseau et l'identité du voisin par lequel passer pour atteindre la destination par le chemin le plus court. À chaque destination possible est donc associée le prochain nœud et une distance. Si un voisin envoie un paquet de contrôle dans lequel il indique être plus près d'une destination que le prochain nœud que l'on utilisait jusqu'alors il le remplace dans la table de routage. Un des inconvénients de cette technique est qu'il est difficile de conserver plusieurs routes alternatives (on ne dispose que de l'information sur le prochain nœud et on ne sait pas si la suite de la nouvelle route est indépendante de celle qui a été brisée).

2.2.3 Taxonomie des protocoles de routage *ad hoc*

La figure 9 présente une taxonomie des protocoles de routage pour les réseaux *ad hoc*. Ces protocoles se différencient d'abord par le niveau d'implication des nœuds dans le routage. Ils sont uniformes si tous les nœuds du réseau jouent le même rôle pour la

fonction de routage. À l'inverse, ils peuvent être non-uniformes si une structure hiérarchique est donnée au réseau et que seuls certains nœuds assurent le routage. Ainsi, dans les protocoles à sélection de voisins, chaque nœud sous-traite la fonction de routage à un sous-ensemble de ses voisins directs. Pour les protocoles à partitionnement, le réseau est découpé en zones dans lesquelles le routage est assuré par un unique nœud maître.

Les protocoles de routage uniforme peuvent également être regroupés selon les données qu'ils utilisent pour effectuer leur tâche. Dans les protocoles orientés topologie et à état de lien, chaque nœud utilise comme données l'état de ses connexions avec ses nœuds voisins; cette information est ensuite transmise aux autres nœuds pour leur donner une connaissance plus précise de la topologie du réseau.

Les protocoles orientés destinations (*vector protocols*) maintiennent pour chaque nœud destination une information sur le nombre de nœuds qui les en séparent (la distance) et éventuellement sur la première direction à emprunter pour y arriver (le vecteur).

Avec un protocole proactif, les routes sont disponibles immédiatement. Cependant, le trafic induit par les messages de contrôle et de mise à jour des tables de routage peut être important et partiellement inutile. De plus, la taille des tables de routage croît linéairement en fonction du nombre de nœuds. À l'opposé, dans le cas d'un protocole réactif, aucun message de contrôle ne charge le réseau pour des routes inutilisées. Mais, pour ces dernières, la mise en place d'une route par inondation peut être coûteuse et provoquer des délais importants avant l'ouverture de la route.

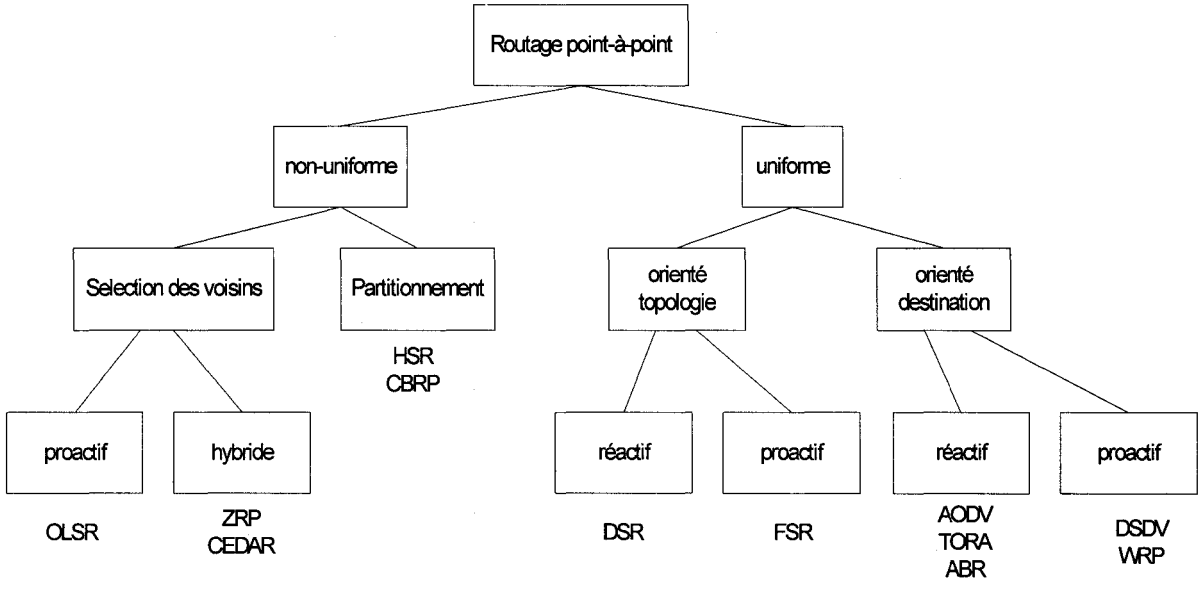


Figure 9 Taxonomie des protocoles de routage pour les réseaux *ad hoc*

2.3 Description des protocoles de routage

Dans cette section nous décrirons différents protocoles de routage dans les réseaux *ad hoc*. Nous choisissons les plus significatifs parmi les trois catégories suivantes : les protocoles proactifs, les protocoles réactifs et les protocoles hybrides.

2.3.1 Les protocoles proactifs

Le principe de base de ces protocoles est de maintenir à jour les tables de routage, de manière à ce que lorsqu'un mobile désire transmettre un paquet à un autre mobile, une route soit immédiatement déterminée. Ces tables de routage sont obtenues grâce à des messages envoyés périodiquement ou déclenchés à la suite d'événements importants pour le réseau.

Dans le contexte des réseaux *ad hoc* les nœuds peuvent apparaître ou disparaître de manière aléatoire et par la suite la topologie change. Cela signifie qu'il faut un échange

continuel d'information pour que chaque nœud ait une mise à jour de l'état du réseau. Les tables sont donc maintenues au moyen des paquets de contrôle, et il est possible d'y trouver directement et à tout moment une route vers chaque destination en fonction de différents critères. On peut par exemple privilégier les routes comportant moins de sauts, celles qui garantissent une bande passante minimale, ou encore celles où le délai est le plus faible. L'avantage dans ces protocoles est d'avoir les routes immédiatement disponibles quand les applications en ont besoin, mais cela se fait au coût d'échanges continus de messages (consommation de bande passante) qui ne sont certainement pas tous nécessaires (seules certaines routes seront utilisées par les applications en général). Comme exemple de protocoles réactifs, on trouve OLSR, DSDV, WRP, etc.

2.3.1.1 DSDV (*Destination-Sequenced Distance Vector Protocol*)

Ce protocole est basé sur l'idée classique de l'algorithme distribué de Bellman-Ford en rajoutant quelques améliorations [Perkins et Bhagwat (1994)]. Chaque station mobile maintient une table de routage qui contient :

- toutes les destinations possibles,
- le nombre de nœuds (ou de sauts) nécessaires pour atteindre la destination,
- le numéro de séquence, *SN* (*sequence number*) qui correspond à un noeud destination.

Le SN est utilisé pour faire la distinction entre les anciennes et les nouvelles routes, ce qui évite la formation de boucles de routage.

La mise à jour dépend donc de deux paramètres : le temps, c'est-à-dire la période de transmission, et les événements, par exemple : l'apparition d'un nœud, la détection d'un nouveau voisin etc. La mise à jour doit permettre à une unité mobile de pouvoir localiser, dans la plupart des cas, une autre unité de réseau.

Un paquet de mise à jour contient :

- le nouveau numéro de séquence incrémenté du nœud émetteur ;

et pour chaque nouvelle route :

- l'adresse de la destination,
- le nombre de nœuds (ou de sauts) séparant le nœud de la destination,
- le numéro de séquence (des données reçues de la destination) tel qu'il a été estampillé par la destination.

Ce protocole DSDV offre la disponibilité des chemins à toutes les destinations à n'importe quel moment ce qui engendre un délai minimal pendant le processus d'établissement du chemin. D'autre part, le mécanisme de mise à jour avec les *flags* de *SN* fait en sorte que le réseau filaire actuel est adaptable aux réseaux *ad hoc*. Le DSDV élimine les deux problèmes de boucle de routage (*routing loop*), et celui du *counting to infinity*. Par contre, DSDV souffre d'un *overhead* de contrôle excessif qui est proportionnel au nombre de nœuds dans le réseau. Par conséquent, il ne garantit pas la mise en échelle dans le réseau *ad hoc*. De plus, pour obtenir une information particulière pour un nœud de destination, la source doit attendre le message de mise à jour envoyé par le nœud de destination. Ce délai peut engendrer des informations inexactes dans les nœuds.

2.3.1.2 OLSR (*Optimized Link State Routing Protocol*)

OLSR est un protocole proactif, non uniforme et basé sur la sélection de voisins [Jacquet *et al.* (2003)]. C'est une optimisation de l'algorithme classique appelé «à états des liens». Le protocole repose sur le concept clef de relais multipoint MPR (MultiPoint Relay). Les MPR d'un nœud correspondent à l'ensemble des voisins qui permettent d'atteindre tous les nœuds situés à deux sauts. La diffusion des différents messages de contrôle se

fait uniquement vers les MPR (voir la figure 10). Cette technique permet de réduire les transmissions inutiles. D'autre part, OLSR distingue les liens unidirectionnels des liens bidirectionnels, seuls utilisés pour le routage.

Chaque nœud maintient de l'information sur les nœuds qui l'ont élu en tant que MPR. Ceci est fait grâce à des messages de présence (les messages *hello*) envoyés par chaque nœud à ses voisins. Ces messages contiennent :

- la liste des nœuds que l'émetteur a choisis comme MPR,
- la liste des nœuds avec lesquels l'émetteur possède des liens bidirectionnels,
- la liste des nœuds que l'émetteur peut entendre (reliés par des liens unidirectionnels).

La diffusion de ces messages permet aux nœuds du réseau de maintenir, dans leur table des voisins, les visions à deux sauts et de calculer l'ensemble de leurs MPR. Cet ensemble est recalculé dès qu'un changement est détecté dans le voisinage à deux sauts.

La diffusion sur la totalité du réseau (via les MPR) de messages de contrôle de topologie (*Topology Control*, TC) donne l'information topologique nécessaire au routage. Ces messages contiennent, pour chaque MPR, la liste des nœuds qui l'ont choisi. Grâce à ces messages, les nœuds peuvent maintenir une table de topologie (*Topology Table*), indiquant le dernier saut pour chaque destination.

Un algorithme de plus court chemin, appliqué à la table des voisins et à la table de topologie, permet de construire la table de routage de chaque nœud. Cette table mémorise, pour tous les nœuds du réseau, le nombre de sauts et le premier saut pour l'atteindre. Elle doit être recalculée dès que l'une des deux tables sources est modifiée. OLSR fournit des routes optimales en nombre de sauts. Il convient pour des grands réseaux grâce à son mécanisme de MPR, mais est moins efficace pour de petits réseaux.

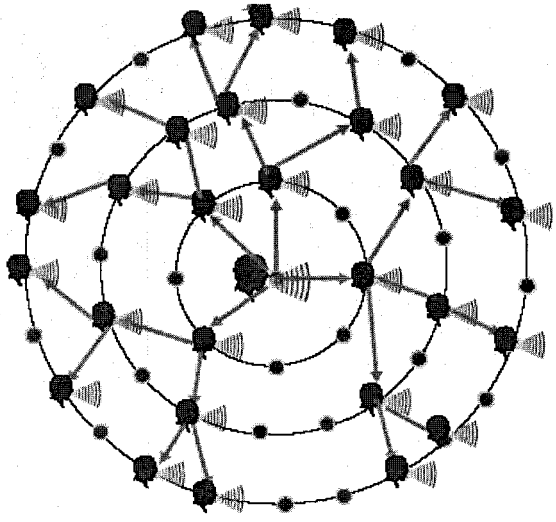


Figure 10 Inondation par les relais multipoint dans OLSR[10]

2.3.1.3 TBRPF (*Topology Dissemination Based on Reverse-Path Forwarding*)

Dans ce protocole, chaque nœud maintient en permanence un arbre dont il est la racine et qui fournit les chemins les plus courts pour tous les autres nœuds du réseau [Ogier *et al.* (2004)]. TBRPF est constitué de deux parties complémentaires : la découverte des voisins et le routage.

La découverte des voisins est assurée par un mécanisme de paquets *hello* diffusés régulièrement au voisinage direct. Ces paquets *hello* contiennent la liste des voisins du nœud, et permettent ainsi de connaître rapidement la topologie complète du réseau à deux sauts. Il faut noter que TBRPF utilise une technique de *hello* différentielle ou seuls les changements de topologie sont notifiés (diminuant ainsi la taille moyenne des paquets et autorisant leur envoi à une plus grande fréquence).

---

[10] www.ares.insa-lyon.fr/tarot/download/PhilippeJacquet_03_01.ppt

Le routage est basé sur un échange des arbres de routage entre nœuds voisins, conduisant progressivement à la diffusion de l'information dans l'ensemble du réseau. Là encore seules des parties d'arbres sont échangées. Normalement, un nœud ne diffuse qu'un sous-arbre à deux niveaux dont il est la racine. Au premier niveau apparaissent les liens vers tous les voisins directs du nœud, et au deuxième niveau un unique lien vers chaque voisin à deux sauts (on peut noter ici une certaine similitude avec le mécanisme des relais multipoints d'OLSR). En conjonction avec ce système de base, TBRPF peut également ajouter des informations sur d'autres liens à l'arbre diffusé, avant de réagir plus vite en cas de changement de la topologie. À noter enfin que dans un souci d'économie de bande passante, les sous-arbres et les paquets *hello* sont regroupés autant que possible dans un même paquet (on parle d'agrégation ou *piggybacking* puisque l'on profite des paquets *hello* pour envoyer en même temps les sous-arbres).

2.3.1.4 WRP (*Wireless Routing Protocol*)

Le WRP est un protocole basé sur l'utilisation des algorithmes de recherche des chemins, PFA (*Path-Finding Algorithm*) [Murthy et Garcia-Luna-Aceves (1996)]. Ces algorithmes utilisent des données sur la longueur et le nœud prédécesseur du chemin le plus court, correspondant à chaque destination; et cela pour éviter le problème du (*counting to infinity*). Le problème avec les algorithmes PFA est la présence des boucles de routage temporaires dans le chemin spécifié par le prédécesseur, avant qu'ils ne convergent. Pour résoudre ce problème, WRP utilise un algorithme de recherche de chemins qui permet de réduire les situations des boucles temporaires qui limitent les mises à jour, uniquement aux changements significatifs des entrées de la table de routage.

Dans le protocole WRP, chaque nœud du réseau maintient quatre tables :

- table de distance (la distance entre le nœud et les différents correspondants),

- table de prochain saut (le voisin à joindre pour contacter un mobile dans le réseau),
- table de coût (la latence entre le nœud et les différents destinataires),
- table de retransmission de messages, MRL (*Message Retransmission List*) (avec les informations de mise à jour).

Chaque mobile émet régulièrement un message indiquant les changements dans sa table de routage; il diffuse également des demandes de confirmation de présence à ses voisins pour les informer des changements topologiques tout en vérifiant la validité de son voisinage.

Le protocole WRP possède les mêmes avantages que le DSDV. En plus de ces avantages, il possède une mise à jour rapide des tables ce qui minimise la complexité de maintenance des tables multiples. Cependant, il a des inconvénients tels que par exemple : la nécessité d'une grande mémoire et un long temps de traitement. Pour une mobilité importante, l'*overhead* de contrôle intervient dans la mise à jour des tables ce qui n'est pas souhaitable pour les réseaux de taille importante et les réseaux ayant un grand dynamisme.

2.3.2 Les protocoles réactifs

Le principe d'un protocole réactif consiste à ne rien faire tant qu'une application ne demande pas explicitement d'envoyer un paquet vers un nœud distant. Cela permet d'économiser de la bande passante et de l'énergie. En revanche, il peut s'écouler un temps plus ou moins long avant que la route ne soit construite et que le mobile ne puisse envoyer ses paquets de données. Lorsqu'un paquet doit être transmis, le protocole de routage va déterminer une route jusqu'à la destination. Une fois la route déterminée, elle sera inscrite dans la table de routage et pourra être utilisée. D'une façon générale, cette recherche se fait par inondation (un paquet de recherche de routes est transmis de proche

en proche dans tout ou partie du réseau). L'avantage de cette méthode est qu'elle ne génère du trafic de contrôle que lorsqu'il est nécessaire. Les principaux inconvénients sont que l'inondation est un mécanisme coûteux qui va faire intervenir tous les nœuds du réseau en très peu de temps et qu'il va y avoir un délai d'établissement des routes. Comme exemple de protocoles réactifs, on trouve AODV, DSR, TORA, LAR, etc.

2.3.2.1 DSR (*Dynamic Source Routing*)

Dans ce protocole, le routage est basé sur le routage source [Johnson et Maltz (1996)]; la source des données détermine la séquence complète des nœuds à travers lesquels, les paquets de données seront acheminés. Le processus de découverte de routes se fait comme suit; la source diffuse un paquet *RREQ* (*Route REQuest*). Si l'opération de découverte est réussie, la source reçoit un paquet réponse de route, RREP (*Route REPly*), qui contient la séquence de nœuds à travers lesquels la destination peut être atteinte. Le paquet requête de route contient donc un champ enregistrement de route, dans lequel sera accumulée la séquence des nœuds visités durant la propagation de la requête dans le réseau (figure 44).

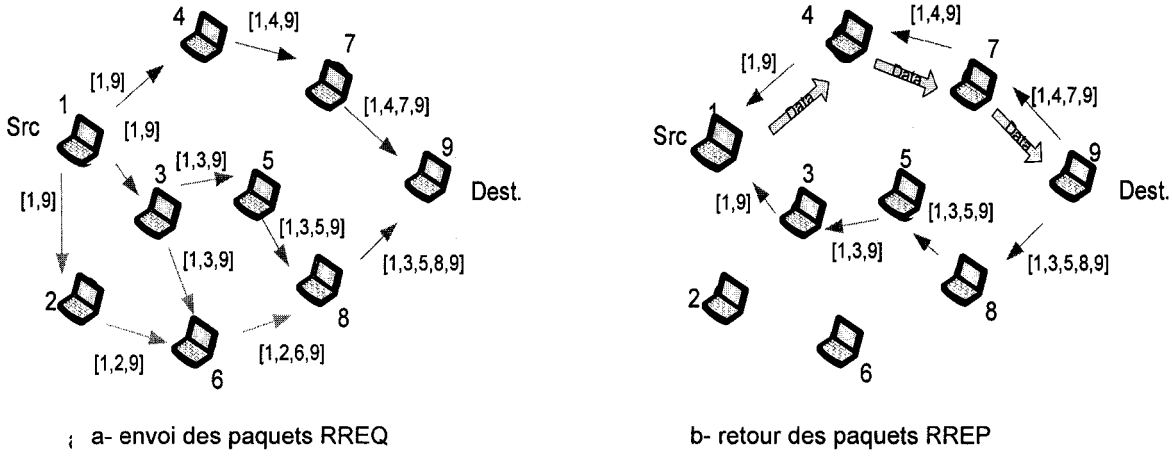


Figure 11 Le principe de découverte d'une route dans DSR

Afin d'assurer la validité des chemins utilisés, le DSR exécute une procédure de maintenance de routes :

- quand un nœud détecte un problème fatal de transmission, à l'aide de sa couche de liaison, un message RERR (*Route ERRor*) est envoyé à l'émetteur original du paquet,
- le message d'erreur contient l'adresse du nœud qui a détecté l'erreur et celle du nœud qui le suit dans le chemin,
- lors de la réception du message RERR par l'hôte source, le nœud concerné par l'erreur est supprimé du chemin sauvegardé, et tous les chemins qui contiennent ce nœud sont tronqués à ce point-là. Par la suite, une nouvelle opération de découverte de routes vers la destination est initiée par l'émetteur.

Le protocole DSR présente les avantages suivants :

- L'utilisation de la technique «routage source», fait en sorte que les nœuds de transit n'aient pas besoin de maintenir les informations de mise à jour pour

envoyer les paquets de données, puisque ces derniers contiennent toutes les décisions de routage,

- Il autorise à la source de conserver dans sa table de routage plusieurs chemins valides pour la même destination, ce qui peut être utile dans le cas de ruptures de liens,
- L'absence totale de boucle de routage, car le chemin source–destination fait partie des paquets de données envoyés.

Cependant, le temps d'établissement d'une route est beaucoup plus important que pour les protocoles proactifs. De plus, l'*overhead* pour le routage est important parce que le routage est initié par la source. Cet *overhead* est directement proportionnel à la longueur du chemin et par la suite il y a un surcoût dans la signalisation. DSR performe mieux dans un environnement statique ou à faible mobilité, cette performance se dégradant quand la mobilité augmente.

2.3.2.2 AODV (*Ad hoc On-demand Distance Vector Protocol*)

Le protocole AODV représente essentiellement une amélioration de l'algorithme DSDV que nous avons décrit dans la section 2.3.1.1 dans un contexte réactif [Perkins *et al.* (2003)]. Contrairement à DSDV qui maintient toutes les routes possibles, le protocole AODV réduit le nombre des messages diffusés en créant les routes en cas de besoin. Il utilise le principe des numéros de séquences afin de maintenir la consistance des informations de routage. Les numéros de séquence permettent d'utiliser les routes les plus récentes (*fresh routes*).

Le protocole AODV repose sur trois principales composantes :

- l'initiation et la propagation des messages RREQ,
- l'initiation et la propagation des messages RREP
- la maintenance des tables « vecteurs à distance ».

AODV utilise le message RREQ dans le but de trouver un chemin vers une destination. La route peut ne pas exister si la destination n'est pas connue au préalable, ou si le chemin existant vers la destination a expiré ou est devenu défaillant. Cependant, AODV maintient les routes d'une façon distribuée en conservant une table de routage au niveau de chaque nœud de transit appartenant à la route cherchée. Afin de maintenir des routes cohérentes, une transmission périodique du message *hello* est effectuée. Si au bout d'un certain temps aucun message *hello* n'est reçu à partir d'un nœud voisin, le lien en question est considéré défaillant.

Un nœud diffuse une requête de route dans le cas où il aurait besoin de connaître une route vers une certaine destination et qu'une telle route n'est pas disponible dans le cas où : (i) la destination n'est pas connue au préalable, (ii) la durée de vie du chemin existant vers la destination a été expirée ou le chemin est devenu défaillant.

Les avantages de ce protocole sont :

- il demande moins de bande passante,
- il performe mieux pour les réseaux de grande taille,
- il offre une convergence rapide quand la topologie du réseau change, car il évite la boucle de routage et il évite le problème de «*counting to infinity*» de Bellman-Ford.

Les inconvénients du protocole AODV sont :

- une seule requête *RREQ* peut générer plusieurs paquets *RREP* qui peuvent engendrer un nombre important de message de contrôle,
- les nœuds intermédiaires peuvent mener à des chemins incohérents; c'est le cas où le *SN* de la source n'a pas été mis à jour et que les nœuds intermédiaires possèdent des *SN* plus grands mais plus petits que le *SN* de la dernière destination.

2.3.2.3 TORA (*Temporary ORdered Algorithm*)

Ce protocole avait pour objectif de minimiser l'effet des changements de topologie qui sont fréquents dans les réseaux *ad hoc* [Park et Corson (2001)]. Afin de s'adapter à la mobilité, le protocole stocke plusieurs chemins vers une même destination, ce qui fait que beaucoup de changements de topologie n'auront pas d'effet sur le routage des données, à moins que tous les chemins qui mènent vers la destination soient perdus (rompus). Il est caractérisé par le fait que les messages de contrôle sont limités à l'ensemble des nœuds proches du lieu de l'occurrence du changement de la topologie.

Dans ce protocole, l'utilisation des meilleurs chemins a une importance secondaire. Les longs chemins peuvent être utilisés afin d'éviter le contrôle induit par le processus de découverte de nouveaux chemins. Il consiste à établir un graphe acyclique orienté (*Directed Acyclic Graph* DGA) dont la racine est la destination (c'est-à-dire que le nœud destination est le seul sans arc sortant). Ainsi, depuis chaque station, on peut retrouver la destination en suivant l'orientation du graphe. Un DGA est orienté destination s'il y a toujours un chemin possible vers une destination spécifiée. Il devient non orienté destination si un lien (ou plus) devient défaillant. Dans ce cas, le protocole utilise la technique d'inversement de liens. Ceci assure la transformation du graphe précédent en un graphe orienté destination dans un temps fini.

Afin de maintenir le DAG orienté destination, l'algorithme TORA utilise la notion de taille de nœud. Chaque nœud possède une taille qu'il échange avec l'ensemble de ses voisins directs. Un lien est toujours orienté vers le nœud qui a la plus grande taille vers le nœud qui a la plus petite taille. Dans ce protocole, on trouve quatre fonctions de base : création de routes, maintenance de routes, élimination de routes et optimisation de routes. La figure 12-a montre la création du DAG dans le protocole TORA : le nœud source diffuse le paquet *Query* (QRY) spécifiant l'identifiant de la destination, ID-destination, qui détermine le nœud pour lequel l'algorithme est exécuté. Un nœud qui a

une taille différente de *Null* répond par un paquet *Update* (UPD) qui contient sa taille, figure 12-b. La mise à jour des liens se fait de façon réactive et pour cette raison dans certaines classifications le protocole TORA fait partie des protocoles hybrides.

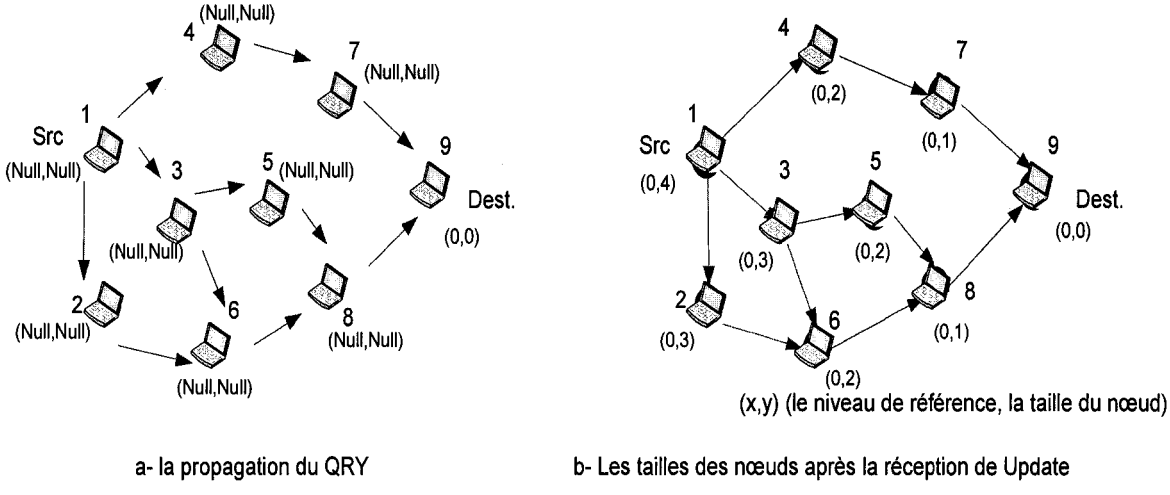


Figure 12 La création des routes dans le protocole TORA

Quand un nœud détecte une défaillance, c'est-à-dire l'invalidité d'un lien (sachant qu'il ne possède pas de suivants valides vers la destination), il lance un nouveau niveau de référence. L'objectif de ce nouveau niveau de référence consiste à indiquer à la source l'invalidité des chemins rompus. La figure 13 donne un exemple de ce processus. La fonction de suppression dans TORA est réalisée en diffusant un paquet CLR (*Clear*) dans le réseau afin de supprimer toutes les routes invalides qui sont sauvegardées par les nœuds du réseau.

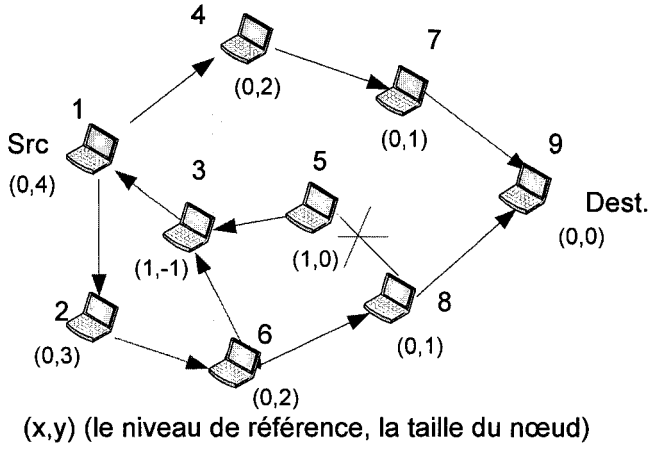


Figure 13 La réaction du protocole TORA à la défaillance du lien (5,8)

En limitant les paquets de contrôle pour la reconfiguration des chemins à petit rayon, TORA a pour avantage de générer moins d'*overhead.* Cependant, la détection de partition simultanée et la détection des routes subséquentes peuvent engendrer des oscillations temporaires. Avec TORA, la réparation des routes se fait localement, ce qui résulte en des chemins non-optimaux.

2.3.2.4 LAR (*Location-Aided Routing*)

Le protocole LAR appelé « Routage aidé par la localisation » est un protocole de routage réactif basé sur l'utilisation de localisation [Ko et Vaidya (1998)]. Ce protocole procède d'une manière très similaire au protocole DSR. La principale différence entre les deux protocoles réside dans le fait que le LAR utilise les informations des localisations, fournies par le système de positionnement global (*Global Positionning System*, GPS), dans le but de limiter l'inondation des paquets de requête de route. Afin d'assurer cela, deux approches peuvent être utilisées.

Dans la première approche, le nœud source définit une région circulaire dans laquelle la destination peut être localisée. La position et la taille de la région sont estimées en se basant sur :

- la position de la destination telle qu'elle est connue par la source,
- l'instant qui correspond à cette position,
- la vitesse moyenne du mouvement de la destination.

LAR permet un routage géographique classique et des fonctions intéressantes en optimisant le nombre de messages émis lors des routes non valides. Lorsqu'une route n'est plus valide, il est possible, avec ce protocole, de faire une découverte locale des routes quand l'information de localisation est disponible. Dans la zone de rupture, LAR route les paquets vers les nœuds du réseau les plus proches de la destination, tout en étant dans la zone de couverture du nœud précédent (figure 14). Une fois la nouvelle route établie, on utilise à nouveau le protocole de routage comme DSR.

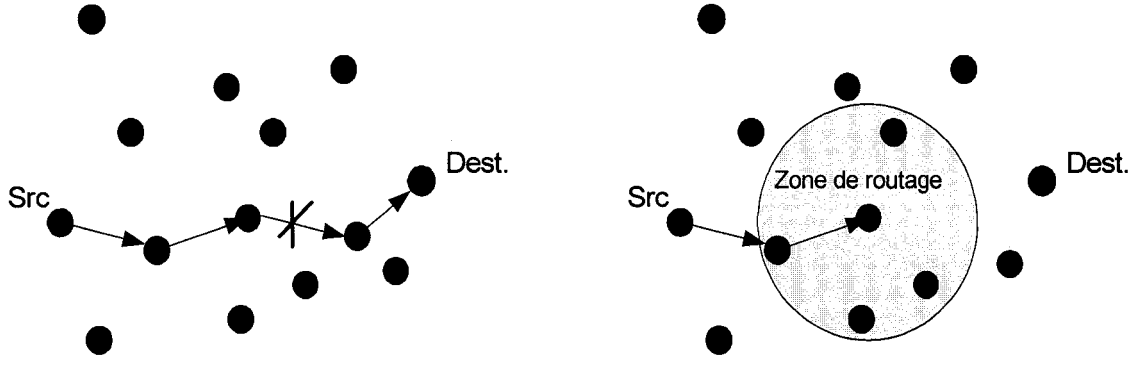


Figure 14 Exemple d'application du protocole LAR

Le protocole LAR a pour avantage de réduire l'*overhead* de contrôle en limitant la zone de recherche d'un chemin. L'utilisation efficace de l'information de position géographique réduit l'*overhead* de contrôle et augmente l'utilisation de la bande passante. Cependant, l'application de ce protocole dépend énormément de la disponibilité du GPS ou d'un autre mécanisme qui donne l'information sur la position. Il ne peut pas être applicable si de telles informations n'existent pas.

2.3.3 Les protocoles hybrides

Les deux approches décrites précédemment ne sont pas les seules alternatives pour les protocoles de routage dans les réseaux *ad hoc*. Une autre approche appelée hybride permet de combiner les approches proactives et réactives afin de bénéficier de la combinaison de leurs avantages.

Les protocoles hybrides combinent les deux approches précédentes, réactives et proactives. Le principe est de connaître le voisinage de manière proactive jusqu'à une certaine distance (trois ou quatre sauts) et si jamais une application cherche à envoyer un paquet à un nœud qui n'est pas dans cette zone, une recherche réactive sera effectuée à l'extérieur de cette zone. Avec ce système, on dispose immédiatement des routes dans le voisinage proche, et lorsque la recherche doit être étendue plus loin, elle sera optimisée. Un nœud qui reçoit un paquet de recherches de routes réactives va tout de suite savoir si la destination est dans son propre voisinage. Si c'est le cas, il va pouvoir répondre, sinon il va propager de manière optimisée la demande dans sa zone proactive. Selon le type de trafic et des routes demandées, ce type de protocole hybride peut cependant combiner les inconvénients des deux méthodes : échanger des paquets de contrôle réguliers et l'inondation de l'ensemble du réseau pour trouver une route vers un nœud éloigné. Comme exemple de protocoles réactifs, on trouve ZRP, CEDAR, etc.

2.3.3.1 ZRP (*Zone Routing Protocol*)

ZRP utilise un protocole réactif au niveau local, c'est-à-dire avec des voisins situés à une distance inférieure à k sauts et un protocole proactif pour le routage entre groupes, appelés *routing zone* [Haas (1997)]. Pour l'opération de recherche d'une route, chaque nœud sont comment rejoindre les autres nœuds dans son groupe. Si le mobile a besoin de contacter un nœud à l'extérieur de cette zone, il utilise le protocole IERP (*Interzone Routing Protocol*) qui envoie une demande de route aux voisins situés à la périphérie du groupe. Ces derniers vérifient alors si le destinataire est dans leur groupe et lui

transmettent le message si celui-ci est présent. Dans le cas contraire, l'algorithme est réitéré : chacun de ces nœuds procède à une demande de route vers les nœuds en périphérie de son groupe. Une fois le chemin trouvé, le nœud destinataire renvoie un message point-à-point vers la source.

En combinant les avantages des mécanismes de routage proactifs et réactifs, le protocole ZRP réduit l'*overhead* de contrôle comparé au mécanisme de diffusion de *RouteRequest* utilisé dans les approches à la demande et à la diffusion périodique de l'information de routage dans les approches proactives. Cependant, et en l'absence d'une demande de contrôle, ZRP produit un *overhead* de contrôle supérieur à CEDAR. Ceci peut avoir lieu du fait que les nœuds peuvent se chevaucher dans des zones de routage. La demande de contrôle doit s'assurer que les messages *RouteRequests* redondants ou dupliqués ne sont pas envoyés. Aussi, la décision pour déterminer le rayon de la zone a un impact significatif sur la performance du protocole.

2.3.3.2 CEDAR (*Core Extraction Distributed Ad hoc Routing*)

CEDAR est un protocole de routage réactif avec qualité de service basé sur une élection dynamique d'un cœur de réseau stable [Sinha *et al.* (1999)]. Des informations sur les liens stables disposant d'une grande bande passante sont propagées entre les nœuds du cœur.

Ce protocole est basé sur trois composantes essentielles :

- **Extraction d'un cœur du réseau :** un ensemble de nœuds est choisi pour calculer les routes et maintenir l'état des liens du réseau. L'avantage d'une telle approche est qu'avec un ensemble réduit de nœuds les échanges d'information d'état et de route seront minimisés, évitant ainsi plus de messages circulant dans le réseau. En outre, lors d'un changement de route, seuls les nœuds du cœur serviront au calcul.

- **Propagation d'état de lien :** le routage avec qualité de service est réalisé grâce à la propagation des informations sur les liens stables avec une grande bande passante.
- **Calcul de route :** celui-ci est basé sur la découverte et l'établissement d'un plus court chemin vers la destination satisfaisant la bande passante demandée. Des routes de «secours» sont utilisées lors de la reconstruction de la route principale, quand cette dernière est perdue. La reconstruction peut être locale (à l'endroit de la cassure), ou à l'initiative de la source.

Un des avantages de ce protocole est qu'il assure le routage avec QoS au moyen des nœuds du noyau. La diffusion du noyau fournit un mécanisme fiable pour établir un chemin qui satisfait la QoS. Cependant, la détermination d'un chemin est réalisée par les nœuds du noyau seulement, le mouvement de ces nœuds affecte la performance du protocole. De plus, la mise à jour des nœuds du noyau peut causer un *overhead* de contrôle assez important.

2.4 Configuration et adressage

Dans les réseaux fixes, la configuration se fait manuellement ou en utilisant un serveur DHCP[11] (*Dynamic Host Configuration Protocol*) qui alloue des adresses aux machines qui joignent le réseau. Dans les réseaux ad hoc, et en absence d'une entité fixe, il est clair que pour assigner une adresse à un mobile qui souhaite joindre le réseau, les autres mobiles doivent connaître cette assignation et l'approuver afin d'éviter la duplication d'adresse. Dans la suite, nous allons explorer les principales propositions présentées dans la littérature et qui s'intéresse à la configuration et à l'adressage dans les réseaux *ad hoc*.

---

[11] DHCP : IETF RFC 2131

2.4.1 DAD

La méthode de détection d'adresse duelle DAD (*Dual Address Detection*) a été proposée par le groupe de travail *Zeroconf* de l'IETF [Perkins *et al.* (2001)]. Avec cette méthode, un mobile qui souhaite rejoindre un réseau *ad hoc* sélectionne une adresse dans une gamme prédéfinie puis il diffuse un message RREQ (*Route Request*) pour cette adresse dans tous le réseau. Si aucun message RREP (*Route Response*) n'est reçu au cours d'un certain temps, il essaye encore jusqu'au temps de RREQ RETRIES pour tenir compte du fait que le message reçu a été perdu le long de la route. Dans ce cas, un problème peut avoir lieu. En effet, lorsqu'un nœud attend une réponse, il doit d'abord avoir une adresse. Il est alors préférable durant la phase d'attribution d'adresse permanente d'affecter une adresse obtenue aléatoirement à partir d'un intervalle d'adresse réservé. Quand une adresse de cet intervalle n'est pas encore choisie comme adresse permanente, et que la plage des adresses temporaires est grande, la probabilité que deux mobiles aient la même adresse durant la phase d'attribution d'adresse permanentes est assez petite.

2.4.2 ANANAS

Dans l'architecture ANANAS (*A New Ad hoc Network Architectural Scheme*), les auteurs proposent de séparer les niveaux 2, 2.5 (niveau *ad hoc*) et le niveau 3 (IP) [Chelius et Fleury (2002)]. Cette approche consiste alors à utiliser deux systèmes d'adressage indépendant : un système au niveau *ad hoc* et l'autre au niveau IP. Ceci nécessite à l'introduction d'une interface virtuelle. Elle est utilisée pour effectuer le lien entre les couches d'adressage *ad hoc* et IP. Avec cette architecture, l'adressage IP n'a pas besoin de configuration spécifique étant donné que le niveau ad hoc est transparent pour le niveau IP. Le niveau IP voit un bus Ethernet auquel tous les nœuds du réseau sont attachés. Puisque le niveau *ad hoc* est vu comme une couche Ethernet simple, alors il supporte pleinement les services IP. Plusieurs réseaux *ad hoc* peuvent être alors considérés par une même couche IP et ils seront vus comme des liens Ethernet

différents. Ainsi, le routage entre ces différents réseaux sera purement IP. La mobilité entre les réseaux *ad hoc* est donc réalisée par le protocole Mobile IP.

2.4.3 DDHCP

Le protocole DDHCP (*Distributed Dynamic Host Configuration Protocol*) est une solution plus complète qui prend en considération les pertes de messages pendant la phase d'attribution des adresses dues à la nature des liens physiques entre les mobiles [Nesgari et Prakash (2002)]. En effet, les auteurs proposent un mécanisme pour renvoyer des adresses libérées à un ensemble d'adresses disponible pour la réutilisation. En d'autres termes, un nœud mobile souhaitant joindre le réseau essaye d'entrer en contact avec un nœud déjà dans le réseau et lui demande d'effectuer l'attribution d'adresse en son nom. Le nœud se joignant est connu comme le demandeur et le nœud effectuant l'attribution d'adresse est connu comme initiateur. Étant donné que l'initiateur appartient au réseau, il peut être atteint par les autres nœuds déjà dans le réseau et peut recevoir ainsi toutes les réponses envoyées. L'initiateur est donc une sorte de procuration pour le demandeur jusqu'à ce qu'il lui assigne une adresse IP permanente et devienne lui-même un nœud du réseau *ad hoc*.

2.4.4 ACDAD

Le mécanisme de configuration et de détection d'adresse dupliquée ACDAD (*an Advanced Configuration and Duplicate Address Detection*) a été proposé pour configurer et optimiser la détection de conflit pour le protocole OLSR [Adjih *et al.* (2005a)]. Ce mécanisme est basé sur l'algorithme DAD, un nouveau message de contrôle MAD (*Mulitple Address Declaration*) est défini et il est utilisé par l'algorithme DAD. Ce mécanisme comprend trois étapes principales :

- **assignation d'adresse** : un nœud sélectionne une adresse IP pour rejoindre un réseau *ad hoc*,

- **détection d'adresse dupliquée** : chaque nœud vérifie qu'il n'y a pas un autre nœud qui utilise la même adresse IP,
- **résolution de conflit** : quand un nœud détecte qu'il existe un autre nœud utilise la même adresse, il va sélectionner une autre adresse.

L'assignation d'adresse se fait de manière simple, par le nœud lui-même sans utiliser un message spéciale. Le nœud choisit une adresse aléatoire. La détection de duplication est basée sur le message de contrôle MAD. Ce message est envoyé par le nœud et inclut un identificateur (ID) et l'adresse choisie. Il sera diffusé de façon périodique dans tous le réseau par les MPR, en assumant que chaque nœud possède un ID unique. Un conflit survient entre deux nœuds quand un nœud reçoit un message MAD incluant sa propre adresse avec un ID différent du sien. Ce nœud déduit que le message MAD n'est pas son propre message et il a été envoyé par un autre nœud qui utilise la même adresse. La règle dans ACDAD pour résoudre un conflit d'adresse est que le nœud ayant un ID plus petit change son adresse. Puisque le nœud a déjà reçu les messages MAD de tous les nœuds du réseau, il va choisir alors une nouvelle adresse qui n'est pas assignée.

2.5 Contrôle de la topologie dans les réseaux *ad hoc*

Il existe principalement trois différentes approches pour le contrôle de la topologie dans les réseaux *ad hoc* : contrôle basé sur la puissance de transmission, contrôle basé sur la formation des clusters et contrôle basé sur la formation d'une dorsale. Dans cette section nous détaillerons chacune de ces approches tout en présentant les principaux algorithmes développés pour chaque approche.

2.5.1 Contrôle basé sur la puissance de transmission

Dans ces approches, on assigne à chaque nœud sa puissance de transmission comme paramètre afin que la topologie du réseau ait certaines propriétés de connectivité et que

la consommation des nœuds soit optimisée. Il existe deux types de contrôle de la topologie :

- les algorithmes centralisés où on suppose que toutes les positions sont connues par une entité centralisée qui détermine leur puissance de transmission. Toutefois, ces algorithmes ont un problème du passage à l'échelle quand le nombre de nœuds augmente,
- les algorithmes distribués quant à eux sont *scalables* et adaptés à la mobilité des nœuds, vu que seules des informations locales suffisent pour calculer la puissance appropriée.

Hou et Li ont étudié le rapport entre la portée de transmission et le débit dans un réseau *ad hoc* [Hou et Li (1986)]. Un modèle analytique a été proposé pour permettre à chaque nœud d'ajuster sa puissance de transmission, de réduire l'interférence et d'ainsi maximiser le débit. Dans [Hu (1993)], l'auteur a développé un algorithme distribué pour que chaque nœud ajuste sa puissance de transmission et construise une topologie fiable avec un haut débit. La minimisation de l'énergie n'a pas été étudiée dans ces deux travaux.

Ramanathan *et al.* ont proposé deux algorithmes centralisés dont la topologie induite avec un critère d'optimisation MINMAX pour réduire au minimum la puissance maximale utilisée par n'importe quel nœud [Ramanathan et Rosales-Hain (2000)]. En plus, deux heuristiques distribuées ont été proposées à savoir LINT (*Local Information No Topology*) et LILT (*Local Information Link-state Topology*) pour ajuster de manière adaptative la puissance de transmission des nœuds afin de garantir une topologie connectée quand celle-ci change. Mais, ni LINT ni LILT ne garantissent la connectivité du réseau. Dans [Lloyd *et al.* (2002)], une généralisation a été faite dans le but de

démontrer que : si la propriété désirée de la topologie est « monotone »[12], et si l'objectif d'optimisation est MINMAX, alors la puissance peut être calculée dans un temps polynomial.

Une méthode distribuée *cone-based* a été développée dans [Wattenhofer *et al.* (2001)]. Chaque noeud augmente graduellement sa puissance de transmission jusqu'à ce qu'il trouve un noeud voisin dans chaque direction, formant ainsi un cône. Par conséquent, la connectivité globale est garantie avec la puissance minimum pour chaque noeud. Huang *et al.* ont étendu ce travail dans le cas où des antennes directionnelles sont utilisées [Huang *et al.* (2002)]. Dans [Marsan *et al.* (2002)], les auteurs ont présenté une méthode pour optimiser la topologie d'un réseau *Bluetooth*, qui vise à réduire la charge de trafic maximum des noeuds, réduisant de ce fait au minimum la puissance maximale des noeuds.

Dans [Singh *et al.* (1998)], les auteurs ont étudié plusieurs métriques pour un routage efficace, comme la minimisation d'énergie consommée par paquet, la réduction de la variation du niveau de consommation d'énergie des nœuds, la réduction du coût par paquet, etc. Wieselthier *et al.* ont analysé le problème d'ajustement de la puissance de transmission de chaque nœud de façon à ce que le coût exprimé en termes d'énergie de l'arbre de *broadcast/multicast* soit minimal [Wieselthier *et al.* (2000)].

2.5.2 Contrôle basé sur la formation des *clusters*

Baker et Ephremides étaient les premiers à introduire la notion de *clustering* dans les réseaux *ad hoc* en proposant une architecture appelée LCA (*Linked Cluster Architecture*) [Baker et Ephremides (1981)]. Ils ont démontré sa capacité à s'adapter au changement de la topologie. Dans [Krishna *et al.* (1997)], les auteurs proposent la

---

[12] Ici, une propriété est monotone si elle est toujours la même lorsqu'un nœud augmente sa puissance de transmission.

formation de *clusters* recouvrants. L'algorithme est exécuté par chaque nouveau nœud après la réception d'une liste des *clusters* actuels. Cet algorithme permet de choisir tous les *clusters* possibles auxquels être connecté, et dont la cardinalité est maximale. Ensuite, un algorithme de suppression est appliqué afin d'éliminer les *clusters* inutiles. Enfin, la liste des *clusters* sera mise à jour et diffusée.

Dans [Gerla *et al.* (2000)], la clustérisation et la diffusion ont été combinés pour diminuer l'*overhead* de la communication. Il n'existe pas de paquet de contrôle explicite pour la construction et la maintenance du *cluster*. Chaque hôte décide de son statut en écoutant le trafic existant. On ajoute à tout paquet qui traverse la couche MAC deux bits qui indiquent le statut de l'expéditeur. Un hôte ordinaire devient alors un *clusterhead* s'il n'existe aucun voisin *clusterhead* actif au moment où il envoie un paquet. Chaque hôte maintient une liste incluant une entrée pour chaque voisin *clusterhead*. Une hôte qui reçoit un paquet de ce dernier décide de son propre statut (*gateway* ou ordinaire) en observant sa liste de voisins *clusterheads*. Les auteurs mentionnent que la clustérisation passive est meilleure par rapport à la clustérisation conventionnelle au niveau de la stabilité et de la robustesse dans un environnement *ad hoc*.

Kawadia *et al.* ont proposé une méthode de clustérisation pour un routage dans les réseaux non homogènes [Kawadia *et al.* (2003)], où les nœuds sont distribués dans des clusters. Le but était de choisir un niveau de puissance de transmission adéquate, de telle façon que les niveaux bas seront utilisés pour des communications à l'intérieur du *cluster* et les niveaux élevées pour celles inter-*clusters*.

2.5.2.1 Algorithme de clustérisation

Un algorithme de clustérisation est basé sur les étapes suivantes :

- **Élection des *clusterheads* :** le réseau est ainsi divisé en plusieurs clusters, la phase d'élection ou de *cluster setup phase* utilise des heuristiques comme le plus grand/plus petit ID dans le voisinage, le degré de connectivité, la zone géographique, la puissance de transmission ou la vitesse de déplacement (exemple : utiliser les nœuds les moins mobiles), ou bien en utilisant un poids pour chaque nœud qui représente une combinaison des derniers attributs.
- **Communication entre les *clusterheads* :** dans un *cluster*, chaque paire de nœuds est à deux sauts de distance entre eux. De plus, comme les *clusterheads* ne sont pas directement reliés, des nœuds passerelles sont aussi élus et utilisés pour les communications entre *clusterheads*.
- **Maintenance des *clusterheads* :** dans le but de s'adapter aux changements de la topologie fréquents dans le réseau, une mise à jour des *clusterheads* élus est dynamiquement réalisée.

2.5.2.2 Clustérisation hiérarchique

La clustérisation hiérarchique adopte le même principe en assignant dynamiquement des *Clusters* ID avec différents niveaux. Un *cluster* peut dynamiquement se fusionner ou s'éclater en fonction du nombre de nœuds dans un *cluster*. Cependant, étant donné la difficulté de prévoir le temps nécessaire pour propager les messages de contrôle de clustérisation à travers les nœuds ainsi que le délai important de convergence de l'algorithme de clustérisation, cette approche dégrade rapidement les performances dans le réseau.

2.5.3 Contrôle basé sur la construction d'une dorsale

Une troisième approche pour effectuer le contrôle de la topologie est de sélectionner un sous-ensemble de nœuds qui forment le réseau de façon à ce que ce sous-ensemble forme une dorsale pour tous les nœuds. Nous avons intérêt à ce que cet ensemble soit de taille minimale étant donné qu'il y aura un *overhead* supplémentaire qui est proportionnel à la taille de la dorsale. De plus, le rôle de cette dorsale nécessite que les nœuds qui forment cette dorsale soient connectés. De ce fait, la détermination de l'ensemble de domination de taille minimale (*Minimum Connected Dominating Set*, MCDS) dans un graphe présente un bon candidat pour construire la dorsale. Cependant la détermination de l'ensemble MCDS dans un graphe est un problème *NP-complet* [Das et Bharghavan (1997)]. Mais il existe plusieurs propositions qui ont pour objet de trouver une approximation de cet ensemble. A titre d'exemple, nous citons les travaux de [Das et Bharghavan (1997)] [Guha et Khuller (1998)] [Sivakumar *et al.* (1998)] [Sinha *et al.* (1999)] [Sinha *et al.* (2001)] [Wu et Li (1999)] [Wan *et al.* (2002)] [Butenko *et al.* (2003)].

Guha *et al.* ont proposé deux algorithmes pour calculer l'ensemble de domination connexe (CDS) [Guha et Khuller (1998)]. Dans le premier algorithme, tous les nœuds sont initialement colorés en blanc. Par la suite, le nœud ayant un degré maximum est coloré en noir et tous ses voisins sont colorés en gris. À chaque itération, un nœud gris ou une paire de nœuds (un nœud gris et un de ses voisins blancs) qui a un nombre maximum de nœuds voisins blancs (chaque nœud blanc est considéré au moins une fois) sont colorés en noir et tous les voisins blancs sont colorés en gris. Le taux de performance de cet algorithme est de $2.\ln(\Delta_{max})+2$, avec Δ_{max} qui représente le degré maximum. Le deuxième algorithme est basé sur l'idée de former l'ensemble de domination connexe. Spécialement à chaque étape, un nœud est coloré en noir s'il connecte un nombre maximum de composants (au moins deux composants). Un composant est ou bien un nœud blanc ou un graphe induit par un ensemble de nœuds noirs. Quand un nœud est coloré en noir, tous ses voisins sont colorés en gris. Cette

étape prend fin quand le graphe ne contient aucun nœud blanc. Dans la seconde étape, deux nœuds gris seront choisis pour connecter deux composants. Cet algorithme a un taux de performance de $\ln(\Delta)+3$, avec Δ qui représente le degré moyen. Dans les deux algorithmes, tous les noeuds colorés en noir forment l'ensemble de domination connexe (CDS).

Dans [Sivakumar *et al.* (1998)], on a présenté deux implémentations distribuées pour ces deux algorithmes proposés par [Guha et Khuller (1998)]. Ces implémentations souffrent du nombre très élevé de message et du temps de complexité. [Sinha *et al.* (1999)] et [Sinha *et al.* (2001)] déterminent un cœur (*core*) qui forme l'ensemble de domination. Chaque nœud choisit son propre dominant ayant le plus haut degré et qui a un nombre maximum de dominés en considérant les voisins à un saut. Il est possible que le nœud choisisse lui-même son dominé. Chaque hôte appartenant au cœur envoie des paquets, en *unicast*, à tous ses voisins à trois sauts. Dans [Sinha *et al.* (2001)], les résultats de simulation montrent que les protocoles de routage DSR et AODV performent mieux quand la structure de dorsale virtuelle qui est employée.

Dans [Wu et Li (1999)], on a présenté un algorithme simple pour calculer l'ensemble CDS. Une version plus étendue de cet algorithme est présentée dans [Wu et Li (2001)]. Chaque hôte a besoin de connaître l'information de ses voisins à deux sauts. Une hôte se déclare comme dominant s'il existe deux voisins qui n'ont pas de chemin direct entre eux. Un dominant u change son statut à dominé si un de ses voisins ayant un *id* plus grand est aussi un dominant et couvre tous les dominés de u, ou si deux de ses voisins connexes v et w possédant des *ids* supérieurs à u et qui sont aussi dominants peuvent couvrir tous les dominés de u.

Dans [Alzoubi *et al.* (2002)], les auteurs ont proposé un algorithme ayant une efficacité de 8-*approximatif* demandant la connaissance d'information des voisins à un saut. Dans cet algorithme, un arbre de recouvrement minimum est arbitrairement construit dans une première étape. Chaque nœud connaît son propre niveau ainsi que les niveaux de ses

voisins à un saut. La racine déclare le statut de ses dominants en diffusant un message DOMINATOR. Chaque hôte recevant le message DOMINATOR déclare son statut comme dominé. Quand un hôte détecte que tous ses voisins de niveau inférieur et que tous les voisins du même niveau ayant un ID inférieur ont le statut de dominé, il se déclare comme un dominant. Un dominé change son statut à dominant s'il détecte qu'un de ses fils est un dominant. Cette règle d'assignation de statut assure que chaque arbre de recouvrement peut générer un ensemble de domination connexe de taille, dans la plupart du temps, 8.OPT, avec OPT étant la taille de l'ensemble de domination connexe optimale.

Butenko *et al.* ont proposé un algorithme (B-CDS) pour déterminer l'ensemble MCDS en deux versions : une version centralisée et une version distribuée [Butenko *et al.* (2003)]. Dans la version centralisée, un nœud est sélectionné à chaque itération, en utilisant une méthode gloutonne, pour être ajouté à la solution finale ou être éliminé de l'ensemble courant tout en vérifiant que la solution finale soit toujours connectée. Dans la version distribuée, on commence par élire des chefs de groupes (*leader*). Par la suite, un leader va exécuter une procédure *self-removal* qui va vérifier si en éliminant un nœud u le sous-graphe induit reste toujours connecté. Si c'est le cas une procédure *remove-vertex* est exécutée, sinon le nœud u doit être présent dans la solution finale et une procédure *fix-vertex* est alors exécutée.

Dans la suite de cette section, nous détaillerons les algorithmes WCDS, CDS-based et B-CDS de [Wu et Li (2001)], [Guha et Khuller (1998)] et [Butenko *et al.* (2003)] respectivement. Nous utiliserons ces trois algorithmes pour comparer et évaluer notre algorithme proposé dans le chapitre 3 pour déterminer l'ensemble MCDS.

Nous détaillerons également un algorithme qui permet de déterminer l'ensemble des relais multipoint MPR (MultiPoint Relay), nous allons utiliser cet algorithme pour comparer différentes techniques de diffusion (chapitre 4).

2.5.3.1 Description de l'algorithme WCDS

L'algorithme WCDS proposé par Wu et Li est basé sur le principe de marquage. Cet algorithme a été proposé initialement dans [Wu et Li (1999)] pour le cas de graphes non-directs en utilisant la notion de l'ensemble dominant uniquement. Cet algorithme a été étendu pour le cas de graphes directs dans [Wu et Li (2001)] en introduisant la notion de l'ensemble absorbant. Chaque nœud est marqué (devient un *gateway*) s'il possède deux voisins non-connectés. Il a été démontré que la collection de nœuds aboutit à l'objectif global (un ensemble de nœuds connectés CDS de taille minimale). Basé sur le principe de marquage, les nœuds *u* et *w* dans la figure 15 sont marqués et forment un ensemble dominant dans le réseau. L'ensemble CDS dérivé de cet ensemble est déduit en appliquant deux règles d'élagage : dans la règle n°1, un nœud peut devenir non marqué si l'ensemble de ses voisins est couvert par un autre nœud marqué; celui-ci, si tous ses voisins sont connectés ensemble via un autre *gateway*, peut abandonner ses responsabilités de *gateway*. Dans la figure 16, un seul des nœuds *u* ou *w* peut enlever le marquage (mais pas les deux). D'après la règle n°2, un nœud marqué peut devenir non marqué si ses voisins sont couverts par deux autres nœuds marqués qui sont directement connectés.

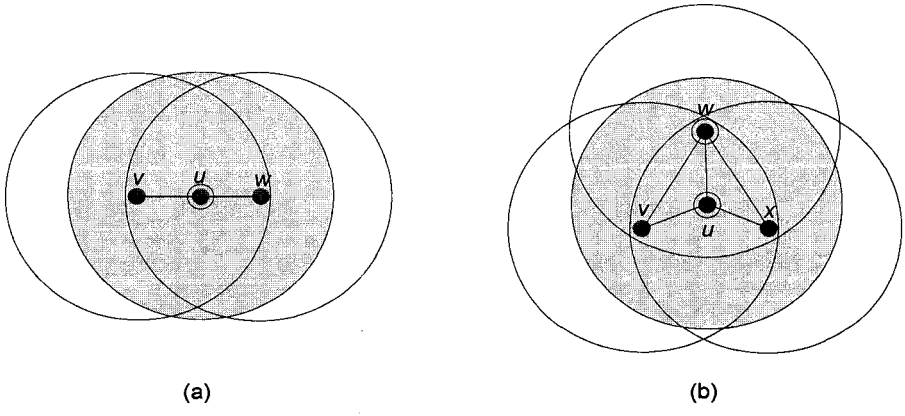


Figure 15 Principe de marquage dans WCDS

Plus formellement, cet algorithme peut se résumer de la façon suivante :

Soit $V' \subset V$ l'ensemble des dominants de G si chaque nœud $v \in V \backslash V'$ est dominé par au moins un nœud $u \in V'$. Aussi, l'ensemble $V \backslash V'$ est appelé l'ensemble des absorbants si pour chaque nœud $u \in V \backslash V'$, il existe un nœud $v \in V'$ qui est un absorbant de u.

L'ensemble des nœuds voisins dominant $N_d(u)$ pour le nœud u est défini par $\{w : (w,u) \in E\}$. L'ensemble des nœuds voisins absorbant $N_a(u)$ pour le nœud u est défini par $\{v : (u,v) \in E\}$. $N(u) = N_d(u) \cup N_a(u)$ représente l'ensemble de voisins pour le nœud u.

À chaque nœud $v \in V$, on assigne un identificateur $id(v)$. Tous les nœuds sont initialement marqués en F (non-marqué).

Règle n°1

> Considérons deux nœuds u et v de G' (sous-graphe induit de V'). Si $N_d(u) \setminus \{v\} \subseteq N_d(v)$ et $N_a(u) \setminus \{v\} \subseteq N_a(v)$ dans G et $id(u) < id(v)$, alors changer le marquage de u à F.

Règle n°2

> Supposons deux nœuds v et w sont bidirectionnellement connectés dans G'. Si $N_d(u) \setminus \{v, w\} \subseteq N_d(v) \cup N_d(w)$ et $N_a(u) \setminus \{v, w\} \subseteq N_a(v) \cup N_a(w)$ dans G et $id(u) = min\{id(u), id(v), id(w)\}$, alors changer le marquage de u à F.

Toutefois, l'utilisation de ces deux règles n'aboutit pas à une solution minimale, et en voici deux contre-exemples :

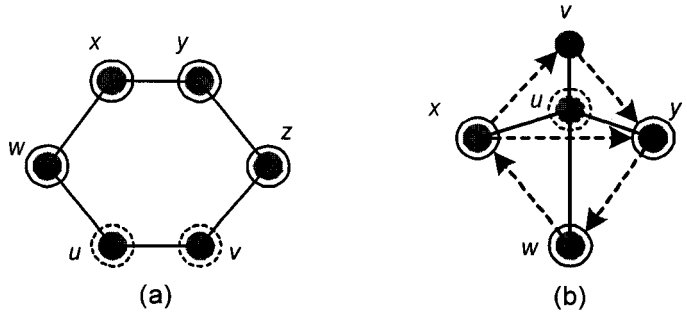


Figure 16 Limitations des règles n°1 et n°2

Dans la figure 16-a, les nœuds *u* et *v* peuvent être éliminés; ils sont couverts par les nœuds *w* et *z*. Dans la figure 16-b, le nœud *u* peut être éliminé car il est couvert par les nœuds *w*, *x* et *y*. Notons que *x* et *y* ne sont pas directement bidirectionnellement connectés; ils peuvent se rejoindre via le nœud *w*. Cependant, aucun de ces nœuds ne peut être éliminé en appliquant les règles n°1 et n°2, car ils ne peuvent pas être couverts par un ou deux nœuds directement bi-directionnellement connectés.

2.5.3.2 Description de l'algorithme CDS-based

L'algorithme CDS-based a été proposé pour la première fois par Guha et Kuller dans [Guha et Khuller (1998)]. Il a été amélioré par Chen et Liestman dans [Chen et Liestman (2003)]. Étant donné un graphe $G = (V, E)$, on associe une couleur (blanche, grise ou noire) pour chaque nœud. Tous les nœuds sont initialement de couleur blanche et changent de couleur au fur et à mesure que l'algorithme progresse. L'algorithme est essentiellement un processus itératif pour le choix des nœuds noirs parmi les nœuds blancs et gris. Quand un nœud est noir, tous ses voisins blancs vont changer de couleurs et devenir gris. L'ensemble des nœuds noirs forme l'ensemble CDS. Le terme morceau (*piece*) est utilisé pour définir une sous structure du graphe. Un morceau blanc est tout simplement un nœud blanc. Un morceau noir contient un maximum de nœuds noirs qui forment un graphe faiblement connecté et un ensemble de nœuds gris voisins à au moins un nœud noir. La figure 17 illustre un morceau blanc et un morceau noir. Les morceaux

sont définis par un trait discontinu. Les nœuds 4, 5, 6 et 7 sont des morceaux blancs. Les autres nœuds font partie de deux morceaux noirs, un contient le nœud 1 et l'autre contient les deux nœuds noirs 2 et 3.

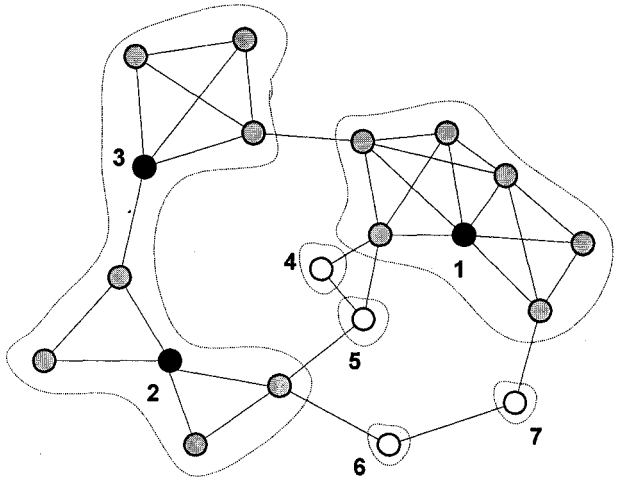


Figure 17 Illustration de morceaux blancs et noirs

À chaque itération, l'algorithme choisit un nœud blanc ou gris pour devenir noir. Le nœud est choisi de manière gloutonne (*greedy*) afin de réduire le nombre de structures autant que possible jusqu'à ce qu'il ne reste une seule structure. L'exemple illustré à la figure 17 montre l'exécution de l'algorithme à sa troisième étape. À l'étape suivante, si le nœud 5 est choisi pour devenir noir, le nœud 4 sera alors fusionné dans la structure, ce qui réduit le nombre de structure à 3. Choisir un nœud (autre que le nœud 5) pour devenir noir, ceci entraînera au plus une fusion de trois structures. C'est pourquoi le nœud 5 sera choisi dans la prochaine étape.

La taille de l'ensemble CDS-based construit en appliquant l'algorithme décrit dans cette section est de l'ordre de $(\ln \Delta_{max}+1).|OPT|$, Δ_{max} est le degré maximum dans le graphe et *OPT* représente la taille minimale de l'ensemble MCDS.

2.5.3.3 Description de l'algorithme B-CDS

Comme mentionné ci-haut, les auteurs dans [Butenko *et al.* (2003)] ont proposé deux versions pour cet algorithme : une version centralisée et une version distribuée.

a- Version centralisée

Dans l'étape d'initialisation, l'ensemble de tous les nœuds du graphe est choisi comme l'ensemble CDS. Deux types de nœuds ont été définis : (i) un nœud fixe est un nœud qui ne peut pas être éliminé de l'ensemble CDS, son élimination engendrait une solution non réalisable ; (ii) un nœud non-fixe qui peut être éliminé, son élimination n'engendrait pas un sous-graphe induit non-connexe. A chaque itération de l'algorithme un nœud est choisi et il sera considéré soit fixe, et il fera partir de la solution finale, soit il sera éliminé de la solution réalisable. Plus formellement l'algorithme s'écrit :

```
/* D est l'ensemble CDS, F est l'ensemble des nœuds fixes*/
D ←V F←∅                                   /* initialisation */
tant que D\F ≠∅ fait
        u ← argmin {δ(v)|v∈D\F}            /* δ(v) représente le nombre des nœuds
                                              Si G[D\{u}] n'est pas connexe alors
                F←F∪{u}                       adjacents pour le nœud v */
        Sinon
                D←D\{u}
                pour tout s∈ D∩N(u) fait    /* N(v) représente l'ensemble des nœuds
                        δ(s)←δ(s) – 1              adjacents pour le nœud v δ(v) = |N(v)|*/
                et pour tout
                if N(u) ∩ F = ∅ alors
                        w←argmax {δ(v)|v∈N(u)}
                        F←F∪{w}
                end
        end
end
retourner D
```

Le temps d'exécution de la version centralisée de cet algorithme est de l'ordre de $O(nm)$ [Butenko *et al.* (2003)].

b- Version distribuée

Dans la version distribuée, la première étape consiste à élire des chefs de groupes (*leaders*). Pour cela, les auteurs utilisent l'algorithme proposé dans [Malpani*et al.* (2000)]. Cet algorithme d'élection s'exécute dans un temps de l'ordre de $O(n \log n)$. Chaque *leaders* élu va alors exécuter une procédure appelée *self-removal*. Cette procédure consiste à vérifier si l'élimination d'un nœud u mènera à un sous graphe induit qui n'est pas connexe. Dans ce cas, une procédure *fix-vertex* est alors exécutée et ce nœud doit être dans la solution finale. Dans le cas contraire, la procédure *Remove-vertex* est exécutée.

Dans cet algorithme, on dit qu'un lien (u, v) est actif pour le nœud v si le nœud u n'a pas été éliminé de l'ensemble CDS auparavant. De plus, les messages sont envoyés uniquement en utilisant des liens actifs, puisque les autres liens ne mènent pas à des nœuds qui font partie de l'ensemble CDS.

Dans la procédure *fix-vertex* différentes étapes sont exécutées : la première étape consiste à diffuser un message NEWDOM, par le nœud u, pour informer l'ensemble des nœuds qu'il est devenu dominant. La deuxième étape est de vérifier si ce nœud u pourra élimer d'autres nœuds. Ceci est réalisé en se basant sur le degré de chaque nœud voisin : le voisin qui possède le degré le plus petit sera considéré en premier et il reçoit un message TRY-DISCONNECT.

La procédure *Remove-vertex* est exécutée uniquement si le nœud v peut être éliminé. Il commence alors par envoyer un message DISCONNECTED à tous les voisins. Par la suite, il sélectionne le nœud qui va être un dominant pour v. S'il existe un dominant au voisinage, alors il sera utilisé. Sinon, un nœud ayant le plus grand nombre de nœuds

adjacents $N(v)$ sera choisi comme nouveau dominant. Finalement, un message SET-DISCONNECT est envoyé au nœud choisi.

Le temps d'exécution de la version distribuée de cet algorithme est l'ordre de $O(nlog^3n)$ [Butenko *et al.* (2003)].

2.5.3.4 Description de l'algorithme MPR

Les relais multipoint MPR (*Multipoint Relay*) sont une technique pour déterminer l'ensemble de domination connexe [Qayyum *et al.* (2002)] [Lim et Kim (2000)] [Calisnescu *et al.* (2001)]. Dans cette technique, on sélectionne une source. Partant de cette source, chaque hôte qui a besoin de diffuser un paquet, choisit un ensemble de relais multipoint de taille minimale parmi ses propres voisins à un saut de façon à ce que cet ensemble de relais couvre tous les hôtes qui sont loin de deux sauts. Chaque hôte diffuse son propre ensemble relais multipoint à tous ses voisins d'un saut. Quand un hôte v reçoit un paquet de u, et v est dans l'ensemble MPR de u, v détermine son propre MPR et retransmet le paquet. Dans un MRS, seuls les voisins de u sont responsables de retransmettre le paquet. L'union de tous les MPRs forme l'ensemble transitaire « *forwarding* ». Qayyum *et al.* ont démontré que le problème de trouver l'ensemble de relais multipoint de taille minimale est un problème *NP-Complet* [Qayyum *et al.* (2002)]. Dans [Jacquet et Claussen (2003)], le protocole OLSR utilise l'ensemble de relais multipoint et optimise ainsi la diffusion des paquets d'information dans le réseau. De plus, [Calisnescu *et al.* (2001)] ont présenté un autre algorithme d'approximation avec une performance fixe de rapport 6. Tous ces algorithmes ont besoin de l'information de voisinage à deux sauts. Une mise à jour périodique de l'information au voisinage consomme de la bande passante qui augmente la communication *overhead.* Plus formellement, l'algorithme de sélection des relais multipoint peut se résumer comme suit [Adjih *et al.* (2005)] :

Soit $N(i)$ l'ensemble des voisins directes de i, $N_2(i)$ l'ensemble des voisins à deux sauts et MPR(i) l'ensemble des relais multipoint de i :

- commencer par un ensemble de relais multipoint vide $MPR(i) = \phi$,
- choisir les nœuds de l'ensemble des voisins $N(i)$ qui sont les seuls ayant un lien avec un voisin du second niveau. Ajouter ces nœuds sélectionnés de $N(i)$ à l'ensemble $MPR(i)$ et éliminer tous les nœuds du second niveau couverts par ces derniers de l'ensemble $N_2(i)$,
- tant que $N_2(i) \neq \phi$, refaire,
    - calculer le degré de chaque nœud dans $N(i)$. Le degré pour un nœud est le nombre de voisins du second niveau couverts par celui-ci présent dans $N_2(i)$,
    - ajouter le nœud de $N(i)$, ayant le degré maximal à l'ensemble des relais multipoint $MPR(i)$, et enlever tous les nœuds du second niveau couverts par celui de $N_2(i)$.

2.6 Conclusion

Bien que d'autres protocoles de routage aient été proposés pour les réseaux *ad hoc*, un chapitre ne suffirait pas pour les présenter. Les principales approches ont été décrites et il est important de retenir qu'aucun des protocoles proposés n'a tous les avantages et la performance recherchée.

Les techniques de configuration et d'adressage (comme DHCP) utilisées dans les réseaux filaires ne sont plus applicables dans le contexte des réseaux *ad hoc.* D'autres solutions, qui tiennent compte des caractéristiques des réseaux *ad hoc*, ont été proposées

dans le but d'assurer la configuration et l'adressage. Dans ce chapitre, nous avons présenté les principales propositions.

Dans ce chapitre, nous avons présenté également les techniques proposées pour effectuer le contrôle de la topologie dans les réseaux *ad hoc*. Ce contrôle de la topologie est primordial étant donné l'absence d'une infrastructure (fixe ou mobile) dans les réseaux *ad hoc*. Ce contrôle permet en quelque sorte de fournir une information utile pour les protocoles de routage surtout quand la mobilité est introduite.

Dans notre recherche, nous nous intéressons particulièrement au contrôle basé sur la construction d'une dorsale et par la suite à la détermination de l'ensemble MCDS pour les raisons suivantes :

- le contrôle basé sur la puissance de transmission ne garantit pas la connectivité du réseau. De plus, nous considérons les réseaux *ad hoc* homogènes, les terminaux possédant le même type de source d'énergie et d'autonomie,
- le contrôle basé sur la formation des *clusters* nécessite une justification adéquate pour choisir les *clusterheads*. De plus, deux nœuds adjacents mais n'appartenant pas au même *cluster* doivent acheminer leurs trafics par une route qui passe par les *clusterhead*s au lieu de l'acheminer directement. Un autre problème survient lorsqu'un *clusterhead* change de position et n'est pas dans le centre du cluster. Dans ce cas les membres du groupe vont s'occuper de l'élection d'un nouveau *clusterhead* ce qui affecte la performance du protocole de routage.

Nous allons proposer un algorithme qui fournit une taille minimale de la dorsale dans un graphe UDG. Dans la littérature, les algorithmes WCDS et CDS-based sont les plus utilisés comme références pour évaluer d'autres algorithmes [Wu et Li (2001)] et [Guha et Khuller (1998)]. Ces deux algorithmes performent mieux que les autres au niveau du temps de calcul, de la complexité et de l'approximation de la solution finale. Dans le chapitre 4, nous allons proposer un nouveau algorithme pour déterminer une

approximation de l'ensemble MCDS et construire ainsi la dorsale. Nous utiliserons ces deux algorithmes pour évaluer et pour comparer la solution finale de notre algorithme.

CHAPITRE 3

CONSTRUCTION D'UNE DORSALE VIRTUELLE

3.1 Introduction

Dans les réseaux sans fil *ad hoc*, les terminaux communiquent entre eux sans l'intermédiaire d'une entité physique. La communication a tendance à être inefficace en termes de traitement et d'utilisation des ressources dans le réseau. L'étude d'une structure partielle ou dorsale virtuelle attire l'attention de plusieurs chercheurs dans le but de réduire *l'overhead* de la communication. Mais la structure d'une dorsale virtuelle est très vulnérable à cause de plusieurs facteurs comme la mobilité des nœuds, l'instabilité des liens, etc. Ainsi, un mécanisme efficace capable de construire et de maintenir la structure de la dorsale est indispensable tout en considérant la stabilité et la couverture des nœuds.

L'absence d'une infrastructure physique augmente non seulement le coût de communication mais fait aussi face à un problème sévère connu sous le nom d'orage d'inondation (*Broadcast Storm Problem*) [Tseng *et al.* (2002)]. Ce problème est engendré par le processus de diffusion des messages de contrôle dans les protocoles de routage qui utilisent la technique d'inondation. Tseng *et al.* ont démontré qu'une rediffusion après la première inondation ne peut uniquement atteindre que 41 % d'espace à couvrir, et ce pourcentage décroît de 5 % à chaque itération. Ils ont démontré également que deux récepteurs d'un même message diffusé se contentent de rediffuser le message avec une probabilité de 0,59 ce qui résulte en une très faible probabilité de l'existence des hôtes sans contention. En conclusion, l'inondation engendre des problèmes tels que : la redondance, la contention et la collision des messages à la réception.

Dans ce chapitre, nous allons détailler notre algorithme proposé pour la construction de la dorsale au moment de l'établissement. Ensuite, une analyse probabiliste sera présentée pour évaluer notre algorithme. Enfin, nous comparerons notre algorithme avec d'autres approches proposées dans la littérature pour construire une dorsale dans un réseau.

3.2 Description détaillée du problème

La minimisation du nombre de messages échangés est requise dans un réseau *ad hoc* où la bande passante est une ressource précieuse, et un réseau trop chargé peut provoquer des pertes de paquets considérables. Minimiser le nombre de messages revient à minimiser le nombre de nœuds qui vont être désignés pour retransmettre les messages de contrôle. L'ensemble de ces nœuds sera alors appelé une dorsale virtuelle. La construction et la maintenance de cette dorsale imposent l'ajout d'un *overhead* de contrôle qui est proportionnel à la taille de la dorsale. Par conséquent, la taille de l'ensemble des nœuds qui forment la dorsale doit être la plus petite possible. De plus, le rôle de cette dorsale nécessite que les nœuds qui forment cette dorsale soient connectés. De ce fait, la détermination de l'ensemble de domination de taille minimale (*Minimum Connected Dominating Set*, MCDS) dans un graphe présente une solution adéquate pour construire la dorsale.

Dans la théorie des graphes, l'ensemble de MCDS dans un graphe est l'ensemble des nœuds tels que chaque nœud est soit élément de l'ensemble MCDS, soit il possède un voisin dans cet ensemble. Rappelons dans ce contexte qu'un voisin est le nœud se trouvant dans la zone de couverture du transmetteur.

Dans ce chapitre, nous supposons que tous les nœuds ont le même rayon de couverture et par la suite nous pouvons modéliser le réseau en utilisant un graphe de disque unitaire, *Unit-Disk Graph* UDG. Toutefois, la détermination de l'ensemble MCDS dans un graphe UDG est un problème *NP-Complet* [Das et Bharghavan (1997)].

La connaissance de la position de chaque nœud dans le réseau nécessite l'utilisation d'un système de localisation tel que le GPS. Dans le cas où un tel système n'est pas disponible, il existe des outils de localisation utilisés dans les réseaux à capteurs (systèmes statiques) tels que Metricom Ricochet et Nokia Rooftop. Il existe aussi des algorithmes efficaces tels que VCap (*Virtual Coordinate assignement protocol*) présenté dans [Caruso *et al.* (2001)] ou APS (*Ad hoc Positionning System*) proposé par Niculescu et al. dans [Niculescu *et al.* (2003)]. D'autres travaux de recherche ont proposé des algorithmes pour déterminer la position des nœuds dans les réseaux *ad hoc* en se basant sur la puissance reçue et en supposant qu'on peut déduire la distance entre deux nœuds connaissant cette puissance [Capkun *et al.* (2001)], [Youssef *et al.* (2005)] et [Dimitri et Dudek (2006)]. Ces algorithmes utilisent généralement les méthodes de trilatération, triangulation ou multilatération pour calculer la position relative d'un nœud par rapport à un autre. Pour déterminer l'ensemble MCDS dans un graphe, nous supposons que la position des nœuds est connue au moment de l'établissement du réseau. En effet, un de ces algorithmes peut être alors utilisé pour déterminer la position de chaque nœud dans le réseau.

Le mouvement des nœuds dans les réseaux *ad hoc* rend la construction d'une dorsale plus difficile. Nous présumons qu'une connaissance appropriée de la topologie du réseau au moment de son établissement est acquise. La topologie du réseau peut être établie suite à l'information sur le nombre des terminaux et leur position. Il est alors faisable de déterminer la dorsale en considérant un système statique. Les nœuds identifiés pour être éléments de cette dorsale formeront une zone de service pour tout le réseau, et prendront en charge des fonctionnalités de localisation des mobiles, de diffusion des informations de topologie, etc. Cette structure doit s'adapter au déplacement des nœuds dans le réseau *ad hoc* de nature dynamique, une procédure sera exécutée par chaque nœud afin de maintenir la dorsale connectée. Dans le chapitre 5, nous allons détailler la procédure de maintenance qui sera exécutée par chaque nœud.

Une dorsale virtuelle permet de réduire le nombre de terminaux qui vont s'occuper de la tâche de diffusion des messages dans tout le réseau. En effet, seuls les membres qui forment cette dorsale auront à retransmettre les messages de contrôle reçus. Cette méthode réduit aussi la consommation de la bande passante dans le réseau, et par la suite l'énergie consommée par les terminaux. Nous avons intérêt à ce que cette dorsale comprenne un minimum de terminaux possible; la taille de cette dorsale doit être aussi la plus petite possible.

L'utilisation d'une dorsale virtuelle s'avère très avantageuse :

- elle permet de limiter l'inondation, puisque les messages de contrôle peuvent être acheminés de façon fiable et économique dans tous les réseaux; seuls les nœuds qui forment la dorsale auront à relayer les messages de contrôle,
- elle peut être utilisée pour collecter l'information sur la topologie pour fin de routage et fournir une route de secours en cas de besoin,
- elle joue un rôle primordial dans le cas des réseaux hybrides, des réseaux *ad hoc* connecté au réseau Internet. En effet, un terminal membre de la dorsale peut être désigné comme un point d'accès (*Access Point*, *AP*),
- elle est utile pour les applications multipoints, chaque nœud de la dorsale peut jouer le rôle d'un relais multipoint chargé de répliquer les paquets multipoint.

3.3 Hypothèses, notations et définitions

Un réseau *ad hoc* est généralement représenté par un graphe $G = (V, E)$, avec V l'ensemble des nœuds et $E \in V^2$ l'ensemble des arcs donnant les communications directes possibles : (u, v) appartient à E si et seulement si u peut envoyer directement un message à v (on dit alors que v est voisin de u). Les couples appartenant à E dépendent de la position des nœuds et de leur portée de communication. Nous prenons l'hypothèse que tous les nœuds ont la même portée R. Soit $d(u, v)$, la distance entre les nœuds u et v. L'ensemble E est défini alors comme suit :

$$E = \{(u, v) \in V^2 \mid d(u, v) \leq R\} \quad (1)$$

Ce graphe est connu sous le nom de graphe unitaire UDG normalisé par R, avec R rayon de transmission ou rayon de recouvrement. Dans ce graphe, $G = (V,E)$, nous définissons $N = |V|$ comme le nombre de nœuds dans le réseau. Le degré d'un nœud u, Δ_u, représente le nombre des nœuds voisins de u, défini par $\{v \mid (u,v) \in E\}$. Le degré moyen d'un réseau est alors égal à :

$$\Delta = \frac{\sum_{u \in V} \Delta_u}{N} \quad (2)$$

Un nœud v est appelé un voisin dominant d'un autre nœud u s'il existe un lien direct. Un sous-ensemble de nœuds est dominant si chaque nœud n'appartenant pas à ce sous-ensemble possède au moins un voisin dominant dans ce sous-ensemble. On désigne par Δ_{max} degré maximal du graphe $\Delta_{max} = max\{\Delta_u \mid u \in E\}$.

Déf. 1 : *Un graphe $G = (V,A)$ est appelé connexe si et seulement si $\forall (u, v) \in V$, $\exists$ un chemin entre u et v.*

Déf. 2 : *Un ensemble $S \subset V$ est appelé dominant si chaque nœud de G, $\notin S$, possède au moins un voisin dans S.*

$$\forall u \in V, \exists v \in S / \exists (u,v) \in E \mid (u,v) \in E \}$$

Déf. 3 : *Basé sur le concept de domination, chaque nœud non-dominant possède un voisin dominant.*

Un ensemble dominant est un sous-ensemble de nœuds du graphe tel que chaque nœud du graphe est soit dans l'ensemble dominant, soit qu'il possède un voisin dans l'ensemble dominant. Par exemple, dans la figure 18-a, les nœuds noirs forment un ensemble de

domination de taille minimale. Chaque nœud dans le réseau n'appartenant pas à cet ensemble, est connecté à au moins un nœud de cet ensemble de domination. La figure 18-b illustre l'ensemble de domination connexe pour le même exemple. L'ensemble des nœuds en gris seront alors ajoutés à l'ensemble des nœuds en noirs pour former la dorsale.

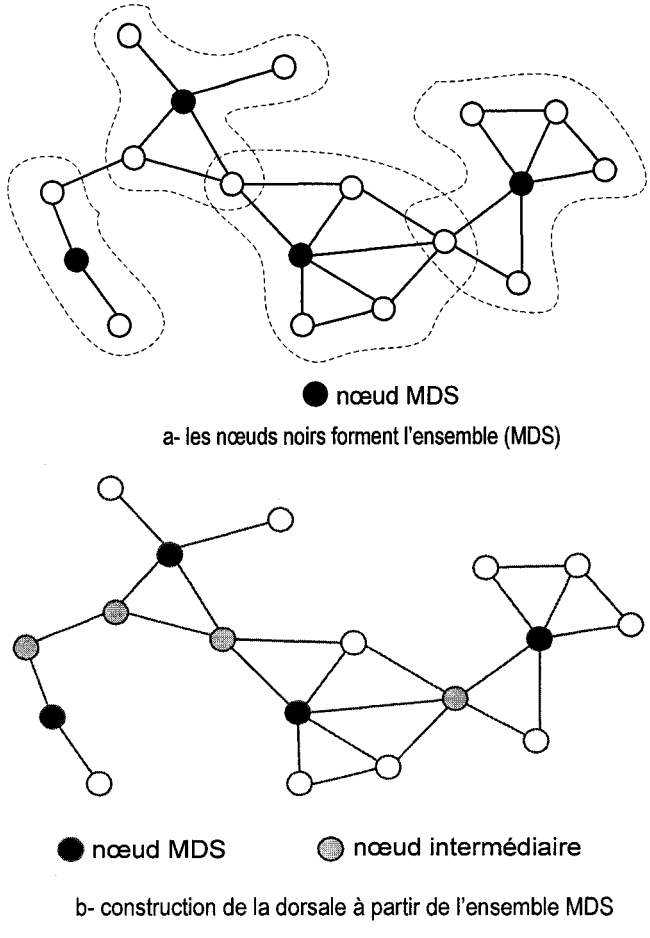


Figure 18 Construction d'une dorsale dans un graphe

Rappelons que la détermination de l'ensemble de domination connexe de taille minimale dans un graphe est un problème *NP-Complet* [Das et Bharghavan (1997)]. Il existe plusieurs heuristiques pour la détermination de l'ensemble de domination connexe.

Quelques-unes essayent de minimiser la taille de cet ensemble, d'autres non. Comme nous l'avons présenté dans le Chapitre 2, toutes ces heuristiques sont basées sur des approches combinatoires basées sur la théorie de coloration dans les graphes ou sur les processus de marquage. De plus, elles commencent par déterminer l'ensemble de domination connexe CDS puis elles essaient de le minimiser. Notre approche, différentes des autres, consiste à trouver, dans un premier temps, l'ensemble de domination de taille minimale (MDS) dans un réseau puis déterminer l'ensemble des nœuds intermédiaires pour avoir un ensemble MDS connexe. Nous allons vérifier au moyen des simulations que notre approche fourni une solution minimale de la taille de la dorsale comparée à d'autres algorithmes proposés dans la littérature.

3.4 Description de l'algorithme MDS-based

Dans cette section, nous allons présenter un nouveau algorithme (CDS-based) pour déterminer une approximation de l'ensemble MCDS dans un graphe UDG. Nous avons décomposé ce problème en deux étapes [Mnif *et al.* (2005a)]. Notre approche consiste à déterminer dans un premier temps l'ensemble de domination de taille minimale (MDS) en utilisant une formulation en programmation linéaire. Une fois cet ensemble déterminé, une deuxième étape consiste à trouver l'arbre de recouvrement d'un graphe réduit formé par l'ensemble MDS. Nous obtiendrons ainsi l'ensemble des nœuds qui vont former la dorsale. Le diagramme de la figure 19 illustre les étapes de résolution pour trouver l'approximation de l'ensemble MCDS.

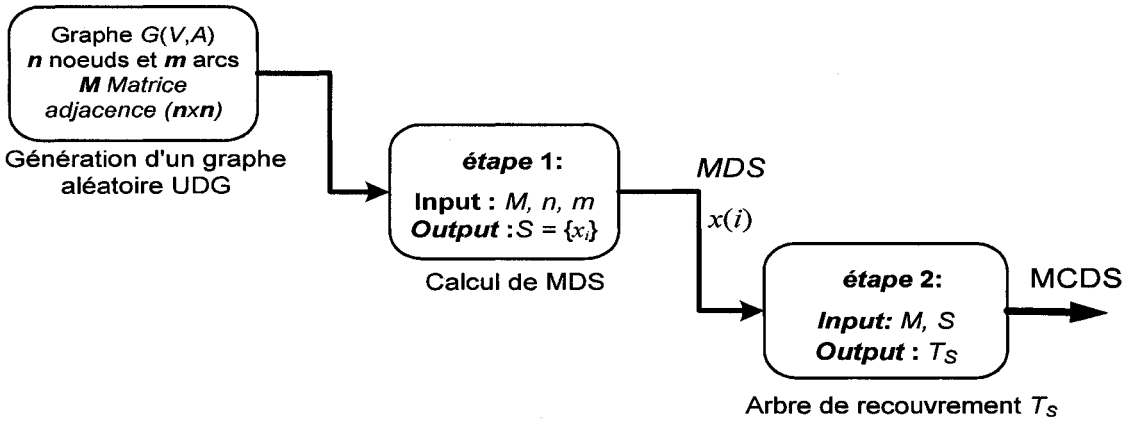


Figure 19 Étapes de résolution

Théorème 1

Soit $G = (V, E)$ comme défini dans la section 3.3, nous allons démontrer que :

> $S \subset V$ est un ensemble de domination connexe de G si et seulement si un arbre de recouvrement T de G tel que $V \backslash S$ est un sous-ensemble des nœuds terminaux de T.

Preuve

Soit $S \subset V$ un ensemble de domination connexe de G. Soit T_S un arbre de recouvrement minimum de sous graphe de G induit par les nœuds de S. Pour chaque nœud $w \in V \backslash S$, choisissons un nœud $u_w \in S \cap N(w)$, avec $N(v) = \{u \in V \mid (u,v) \in E\}$. Il est évident de voir que $T = T_S \cup \{u_w \mid w \in V \backslash S\}$ est un arbre de recouvrement de G et $V \backslash S$ est un sous ensemble des nœuds externes de T (les nœuds ayant un degré égal à 1). Inversement si T est un arbre de recouvrement de G et U est un sous ensemble des nœuds externes de T, alors pour tout nœud de U est relié à au moins un nœud de $V \backslash U$ et le sous graphe de G induit par les nœuds de $V \backslash U$ est connexe. Alors $V \backslash U$ est un ensemble de domination connexe de G.

3.4.1 Détermination de l'ensemble de domination MDS (étape 1)

La première étape consiste à minimiser l'ensemble des nœuds dominants dans un graphe donné. Pour cela, nous définissons une variable de décision x_i tel que :

$$x_i = \begin{cases} 1 & \text{si le noeud } i \text{ appartient à l'ensemble de domination} \\ 0 & \text{sinon} \end{cases}$$

La fonction objectif cherche à minimiser la taille de l'ensemble de domination S, sous la contrainte que chaque nœud dans le graphe G possède au moins un nœud voisin dans cet ensemble de nœuds dominants S.

$$\text{(P)} \qquad \min \sum_{i \in V} x_i \tag{3}$$

La contrainte de domination s'écrit :

$$X + M \times X \geq 1 \tag{4}$$

$$\text{Avec } x_i \in \{0,1\} \tag{5}$$

avec $X = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}$ représente le vecteur de décision,

et M est une matrice d'adjacence ($n \times n$) de G, $M = \begin{bmatrix} m_{11} & \cdots & m_{1n} \\ \vdots & m_{ij} & \vdots \\ m_{n1} & \cdots & m_{nn} \end{bmatrix}$, $m_{ij} = 1$ si et seulement si le nœud i est connecté au nœud j, et $m_{ij} = 0$ autrement.

Avec cette formulation, la solution en x_i est fournie dans un temps de calcul de l'ordre de $O(n)$. En se référant à [Megiddo (1987)], un problème en programmation linéaire avec d variables et n contraintes, le temps de sa résolution est de l'ordre de $O(n)$, avec n peut tendre vers l'infini.

À cette étape, nous n'avons aucune garantie sur la connectivité de la solution fournie par la résolution de ce problème. Pour cela et pour s'assurer que la solution finale soit connexe, nous allons déterminer l'ensemble des nœuds intermédiaires pour que l'ensemble MDS soit connexe.

3.4.2 Détermination de l'ensemble MDS Connexe (étape 2)

La résolution du problème (P) nous donne l'ensemble de domination de taille minimale (MDS) qui n'est pas nécessairement connexe. Afin de trouver les nœuds intermédiaires, nous allons déterminer l'arbre de recouvrement d'un graphe réduit construit à partir de l'ensemble MDS.

Cette étape suivante consiste à :

- construire un graphe réduit *G'* de façon à ce que :
  - les sommets de ce graphe sont formés par l'ensemble MDS,
  - deux nœuds sont reliés par le plus court chemin en nombre de sauts calculé à partir du graphe initial,
  - le poids sur un arc représente le nombre de sauts entre deux nœuds.
- déterminer l'arbre de recouvrement de taille minimale, *T*, dans le graphe *G'*,
- extraire les nœuds intermédiaires en remplaçant les arcs choisis par les chemins réels dans le graphe original.

Pour déterminer le plus court chemin entre deux nœuds, nous avons utilisé l'algorithme de *Dijsktra*. L'algorithme *Dijsktra* peut s'exécuter dans un temps de l'ordre de $O(m + n.\log n)$ en utilisant l'implémentation de tas de *Fibonocci* [Ahuja *et al.* (1993)], avec n et m représentent respectivement le nombre de nœuds et le nombre d'arcs dans le graphe G.

Soit le graphe réduit $G'(V',E')$ formé à partir de l'ensemble MDS. Soit n' et m' représentent respectivement le nombre de nœuds et le nombre d'arcs dans le graphe

réduit G'. La formulation pour déterminer T_S l'arbre de recouvrement minimum *MST* (*Minimum Spanning Tree*) dans un graphe s'écrit alors :

La variable binaire y_{ij} indique si l'arc (i, j) fait partie de l'arbre de recouvrement minimum, T. En d'autres termes :

$$y_{ij} = \begin{cases} 1 & \text{si les noeuds } i \text{ et } j \text{ appartiennent à l'arbre de recouvrement minimum } T_S \\ 0 & \text{sinon} \end{cases}$$

$$min \sum_{(i,j)\in V'} c_{ij}\, y_{ij} \tag{6}$$

Sous les contraintes :

$$\sum_{(i,j)\in T_S} y_{ij} = n_T - 1 \tag{7}$$

$$\sum_{(i,j)\in \pi(R)} y_{ij} \le |R| - 1 \quad \forall\, R \subseteq T_S, \tag{8}$$

$$\text{Avec } y_{ij} \in \{0,1\} \tag{9}$$

Dans cette formulation, le coefficient $c_{i,j}$ désigne le coût sur l'arc (i, j) reliant le nœud i et le nœud j. Dans notre cas, le coût sur un arc représente le nombre de sauts entre les nœuds i et j.

La contrainte (7) est une contrainte de cardinalité ; l'arbre comprend exactement n_T-1 arcs. Avec $n_T = |T_S|$ représente le nombre de liens dans le sous graphe engendré par T_S. La contrainte (8) est appelée *packing constraint* ; l'ensemble d'arcs choisis ne doit pas contenir de cycle (si la solution choisie contient un cycle et R est l'ensemble de nœuds dans ce cycle, la solution viole alors cette contrainte) [Ahuja *et al.* (1993)]. T_S est l'arbre de recouvrement minimum pour S qui contient la solution finale de l'ensemble des nœuds dominant connexe de taille minimale. $\pi(R)$ dénote l'ensemble des arcs dans le sous graphe de S induit par l'ensemble des nœuds R [i.e., $\pi(R)$ est un ensemble d'arcs de

E' avec des points finaux dans *R*]. Notons que, comme fonction du nombre de nœuds dans le réseau, ce modèle contient un nombre exponentiel de contraintes. Néanmoins, ce problème peut être résolu de façon efficace en utilisant les algorithmes de *Prim* et *Kruskal* [Cormen *et al.* (1994)]. Le temps d'exécution pour les algorithmes de *Prim* et *Kruskal* est de l'ordre de $O(m' + n'\log n')$. Dans notre cas, n' est la taille de l'ensemble MDS. Il existe aussi un algorithme distribué : l'algorithme de *Sollin* est une version hybride de les algorithmes de *Kruskal* et *Prim*. Le temps d'exécution de cet algorithme est de l'ordre de $O(m'.\log n')$. Cet algorithme performe mieux que l'algorithme de *Prim* pour les graphes clairsemés (*sparse graphs*) [Ahuja *et al.* (1993)].

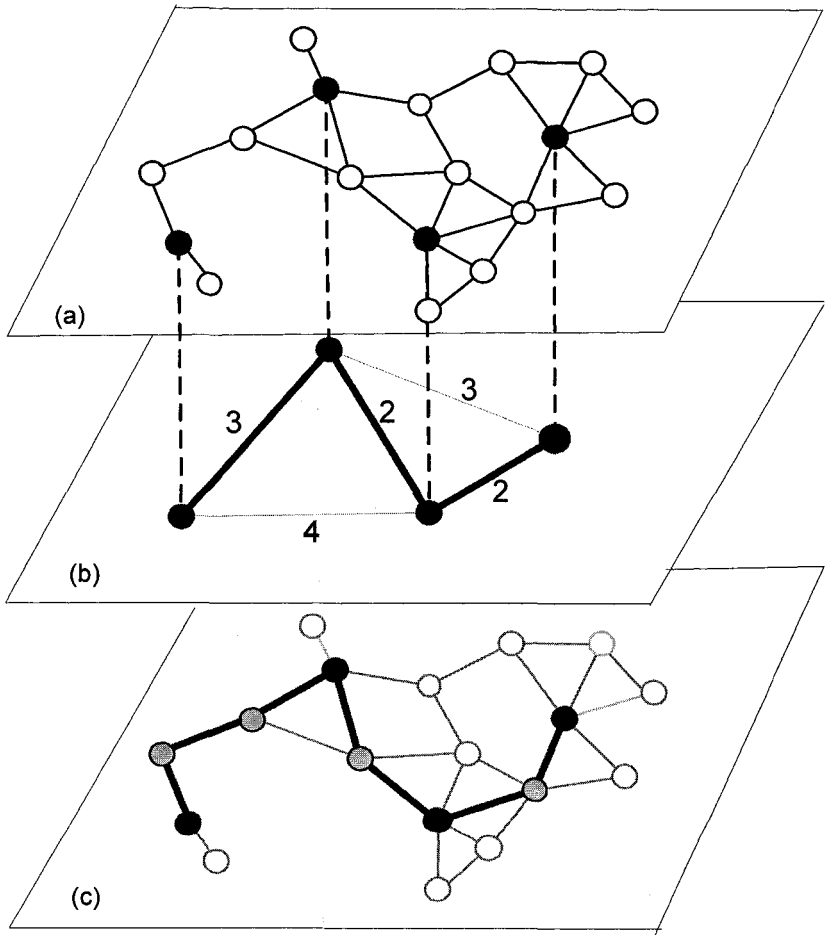


Figure 20 Détermination de la dorsale à partir de l'ensemble MDS

Reprenons l'exemple de la figure 18, la première étape de notre algorithme a permis de déterminer l'ensemble MDS (les quatre nœuds en noirs), comme illustré sur la figure 20-a. Nous construisons le graphe réduit composé de quatre nœuds et cinq arcs, le poids sur chaque arc représente le nombre de saut du plus court chemin entre deux nœuds. Nous déterminons ainsi l'arbre de recouvrement de taille minimale pour ce graphe réduit, figure 20-b. Nous extrairons les nœuds intermédiaires (les nœuds en gris) en effectuant la correspondance entre les arcs de l'arbre de recouvrement et les chemins dans le graphe original, figure 20-c. La dorsale est alors l'ensemble des nœuds noirs et gris et qui représente la dorsale pour cet exemple.

3.4.3 Analyse de performance

Dans cette section, une étude analytique sera présentée afin de vérifier que la taille de la dorsale trouvée en utilisant l'algorithme CDS-based est bien la taille minimale estimée. Pour ce faire, nous allons déterminer la probabilité p_{MCDS} qu'un nœud soit un élément de l'ensemble MCDS [Mnif *et al.* (2006)]. Connaissant la taille du réseau N, la taille estimée de l'ensemble MCDS sera alors :

$$N_{MCDS} = p_{MCDS} N \tag{10}$$

Désignons par $\rho = \frac{N}{S}$ la densité du réseau, avec N qui est le nombre total de nœuds et S la surface totale. Un nœud P ayant une portée, rayon de couverture égal à R, le nombre de nœuds N_1 qui peut être situé dans cette zone de centre P et de rayon R est :

$$N_1 = \rho S_1 \qquad \text{avec } S_1 = \pi R^2$$

$$\text{alors} \quad N_1 = \rho \pi R^2$$

Si les nœuds sont placés dans le plan suivant la distribution de Poisson[13], alors la probabilité d'avoir k nœuds dans la surface S_1 est :

$$p_1(k, N_1) = \frac{N_1^{\,k}}{k!} e^{-N_1} \text{ , avec } N_1 = \rho S_1 \qquad (11)$$

Un nœud est élément de l'ensemble MCDS si et seulement si l'une des deux conditions suivantes est vraie :

p_2 : le nœud est un élément de l'ensemble MDS (étape 1)
p_3 : le nœud est un nœud intermédiaire (étape 2)

La probabilité pour qu'un nœud soit l'élément de l'ensemble MCDS est la probabilité qu'un nœud soit un nœud de l'ensemble MDS, p_2, ou bien un nœud intermédiaire, p_3, sans être un nœud de l'ensemble MDS. Ceci est exprimé par l'équation suivante :

$$p_{MCDS} = p_2 + (1-p_2)p_3 \qquad (12)$$

Dans une surface donnée et pour k nœuds, la probabilité pour un nœud d'être choisi parmi ses k nœuds voisins est donnée par $\frac{1}{k+1}$, alors :

$$p_2 = \sum_{k=1}^{\infty} p_1(k, N_1) \frac{1}{k+1} \text{ avec } p_1(k, N_1) = \frac{N_1^{\,k}}{k!} e^{-N_1}$$

[13] la distribution de Poisson est caractérisée par le fait qu'elle est sans mémoire, purement aléatoire, elle est souvent utilisée dans ce contexte.

$$
\begin{aligned}
p_2 &= \sum_{k=1}^{\infty} \frac{N_1^{\,k}}{k!} e^{-N_1} \frac{1}{k+1} \\
&= \frac{1}{N_1} \sum_{k=1}^{\infty} \frac{N_1^{\,k+1}}{(k+1)!} e^{-N_1} \\
&= \frac{1}{N_1} \sum_{k=2}^{\infty} \frac{N_1^{\,k}}{k!} e^{-N_1} \qquad \text{or} \sum_{k=0}^{\infty} \frac{N_1^{k}}{k!} = e^{N_1} \\
&= \frac{1}{N_1} \left(e^{N_1} - 1 - N_1 \right) e^{-N_1}
\end{aligned}
$$

$$
p_2 = \frac{1 - e^{-N_1}(N_1 + 1)}{N_1} \tag{13}
$$

Dans l'expression de p_2, la somme commence à partir de $k = 1$ et non pas $k = 0$ parce que nous avons éliminé le cas où le nœud ne possède pas de voisin (aucun nœud ne se trouve dans S_1).

Pour évaluer la probabilité que le nœud soit un nœud intermédiaire p_3, considérons la figure 21 où les deux nœuds I et J sont deux nœuds éléments de l'ensemble MDS et qui ne sont pas connectés. Supposons que les nœuds I et J sont à deux sauts, il existe alors un nœud K qui va être le nœud intermédiaire (nœud déterminé par l'étape 2 de l'algorithme). K doit se situer dans la zone grise $S(x)$ définie par l'intersection des deux cercles de centres I et J. Dans le cas général, I et J sont situé à n-sauts, les nœuds intermédiaires sont déterminés en utilisant le même principe.

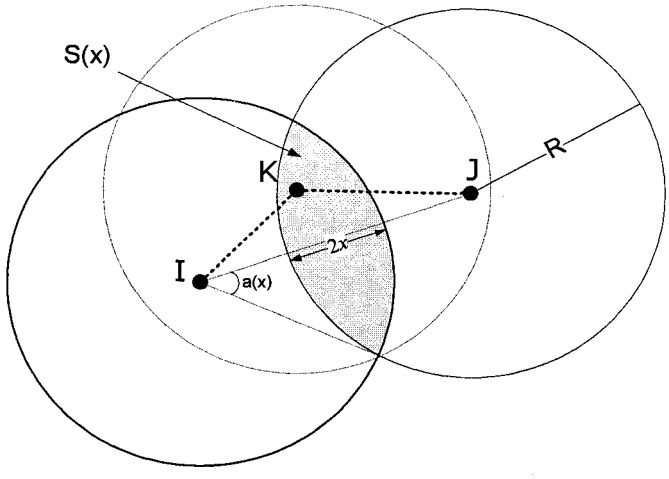


Figure 21 Le nœud *K* doit se situer dans la région *S* pour qu'il soit un nœud intermédiaire pour *I* et *J*.

Sur la figure 21, la surface $S(x)$ dépend de la distance qui sépare les nœuds *I* et *J*; $d(I, J)$. En d'autres termes, S est fonction de la distance x, avec $x = R - \frac{d(I,J)}{2}$ pour $R<d(I,J)<2R$.

$S(x)$ est donnée par [Spiegel et Liu (2000)] :

$$S(x) = 2R^2 a(x)$$

avec $a(x)$ est l'angle que fait l'axe (I,J) avec le point d'intersection des deux cercles de centres *I* et *J* (figure 21), $a(x)$ est donné par :

$$a(x) = \cos^{-1}\left(\frac{x}{R}\right) - \frac{x}{R}\sqrt{1-\left(\frac{x}{R}\right)^2}$$

Désignons par $N(x)$ le nombre de nœuds situés dans la surface $S(x)$, nous avons alors :

$$N(x) = \rho S(x)$$

Et par la suite, la probabilité que le nœud soit un nœud intermédiaire est :

$$p_3 = \sum_{k=1}^{\infty} p_1(k, N(x)) \frac{1}{k+1}$$

avec $p_1(k, N_1) = \frac{N_1^{\,k}}{k!} e^{-N_1}$

En utilisant le même raisonnement dans le calcul de p_2 et le fait que $\sum_{k=0}^{\infty} \frac{N_1^{\,k}}{k!} = e^{N_1}$, nous obtiendrons alors :

$$p_3 = \frac{1 - e^{-N(x)}}{N(x)} - e^{-N(x)} \quad \text{pour } R < d(I,J) < 2R \qquad (14)$$

Dans le cas où nous avons $d(I,J) = 2R$, le nœud K peut être alors considéré comme un nœud de l'ensemble MDS. Donc, ce cas particulier est inclus dans le calcul de p_2. Vérifions ceci mathématiquement; si $d(I,J) = 2R$ alors l'intersection des deux cercles est un point et par la suite $S(x)=0$ et $N(x)=0$ ce qui donne $p_1=0$ et $p_3=0$.

Dans le cas où nous avons $d(I,J) = R$, les nœuds I et J peuvent communiquer directement et ce cas fait partie déjà de p_2.

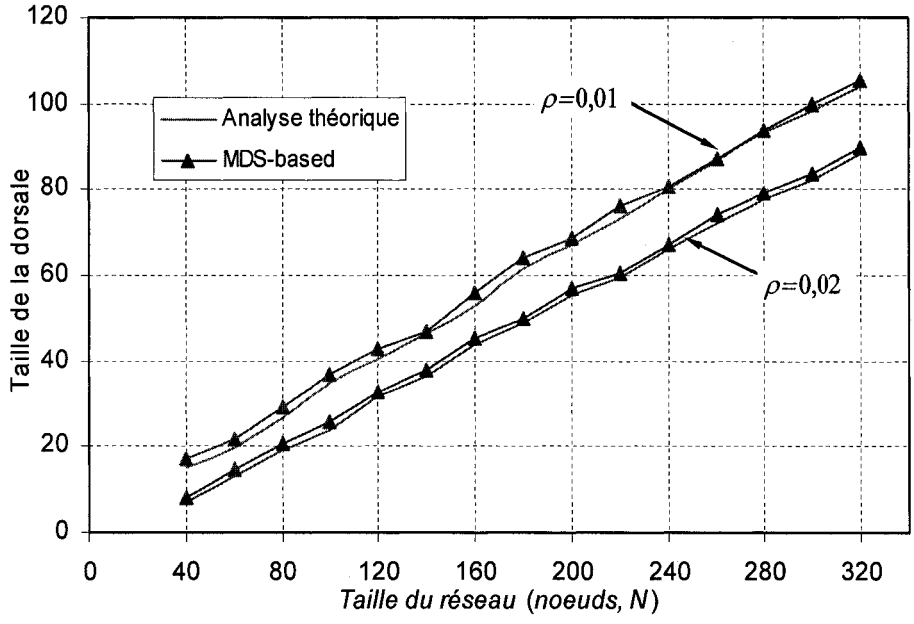


Figure 22 Taille de la dorsale en fonction de la taille du réseau

En utilisant MATLAB, nous allons comparer la taille de la dorsale, dans un graphe UDG donné, calculée par l'algorithme proposé (MDS-based) par rapport à celle donnée par l'analyse théorique (équation 8). Une première comparaison consiste à faire varier la taille du réseau. Les nœuds sont placés dans un plan 2D suivant une distribution exponentielle. La portée de chaque terminal est de $R = 20m$. Pour chaque valeur de N, nous effectuons dix simulations pour prendre une valeur moyenne. La figure 22 illustre les résultats de simulation pour $40 \leq N \leq 320$ pour deux valeurs de densité soit $\rho = 0.01$ et $\rho = 0.02$. La taille de la dorsale croit quasi-linéairement avec la taille du réseau. Les résultats de simulation sont légèrement supérieurs à ceux de l'analyse. L'écart relatif moyen entre les valeurs données par l'algorithme (MDS-based) et celles données par l'analyse est de l'ordre de 4 %.

Une deuxième simulation consiste à comparer la taille de l'ensemble MCDS en variant la portée de transmission, le rayon R, et de comparer les résultats trouvés par notre algorithme (MDS-based) et l'analyse (équation 8). Pour un nombre de nœuds fixe (N =

200) placés exponentiellement dans une surface (100x100 m^2), nous faisons varier le rayon de transmission. Pour chaque valeur de R, nous effectuons dix simulations pour prendre une valeur moyenne. La figure 23 illustre les résultats trouvés par l'algorithme MDS-based et l'analyse pour $10m \leq R \leq 100m$. La taille de l'ensemble MCDS décroît rapidement quand la portée augmente et ceci est expliqué par le fait que le nœud dominant couvre plus de voisins. Les résultats de simulations données par l'algorithme MDS-based sont très proches de ceux retrouvés par l'analyse.

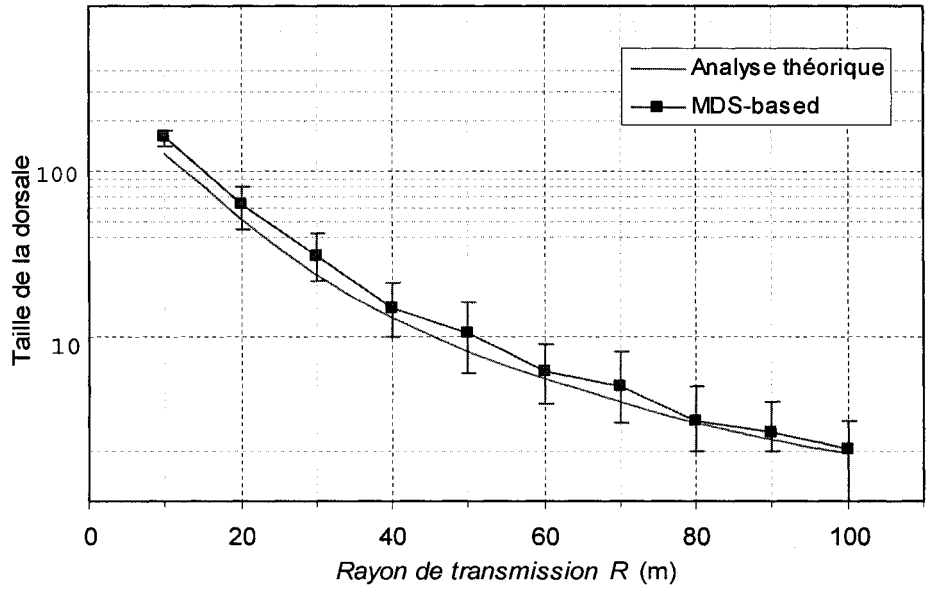


Figure 23 Taille de dorsale en fonction du rayon de transmission R

3.4.4 Comparaison avec d'autres algorithmes (WCDS, CDS-based et BCDS)

Dans cette section, nous allons comparer la solution fournie par notre algorithme (MDS-based) avec celle trouvée en utilisant les algorithmes décrits dans le chapitre 2 soit WCDS, CDS-based et BCDS dans ses deux versions. Plus la taille de la dorsale est petite, plus le résultat est meilleur. Rappelons que ces deux approches commencent par trouver un ensemble de domination connexe (CDS) dans le réseau puis elles essaient de

la minimiser. Alors que notre algorithme MDS-based détermine, dans une première étape, l'ensemble de domination de taille minimale et par la suite nous cherchons à déterminer l'ensemble MDS connexe.

Nous considérons des graphes dont la taille varie de 40 à 320 nœuds. Tous les nœuds possèdent le même rayon de transmission. Les nœuds sont générés et placés aléatoirement dans un plan carré. La taille de ce plan varie en fonction du nombre des nœuds pour avoir une densité constante (même degré moyen) pour toute les simulations. Afin de simuler la structure des réseaux *ad hoc*, deux nœuds seront connectés si la distance qui les sépare est inférieure à $R = 30\ m$. Nous effectuons un prétraitement sur le graphe généré afin de s'assurer qu'il est bien connexe.

Pour chaque valeur de N, une trentaine de simulations ont été exécutées pour prendre une valeur moyenne de la taille de la dorsale. La figure 24 illustre les caractéristiques des graphes générés dans nos simulations. Les résultats démontrent que la procédure utilisée fournit un degré moyen constant avec une variance faible (≤ 0.2). Cette procédure est adéquate et fournit des graphes peu denses et connexes. Le nombre d'arcs croît de façon proportionnelle en fonction de la taille du graphe (nombre de nœuds).

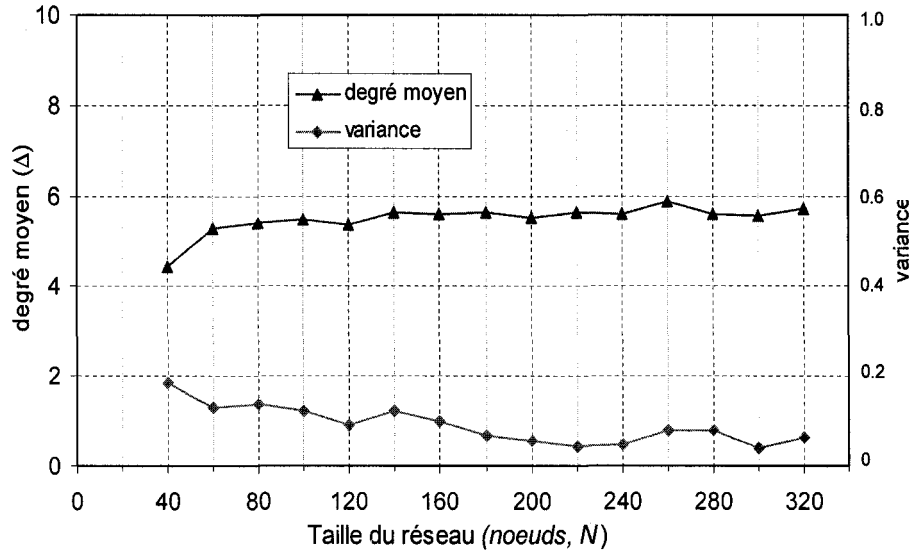


Figure 24 Degré moyen et sa variance en fonction de la taille du réseau N

La figure 25 illustre un exemple de graphe généré par la procédure décrite dans cette section pour $N = 60$. Dans cet exemple, le degré moyen pour ce graphe $\Delta = 4.8$ et le nombre d'arrêtes total $m = 143$. L'ensemble des nœuds dominants (MDS), déterminés par l'étape 1 de l'algorithme, comprend 11 nœuds {11, 12, 14, 15, 20, 21, 24, 25, 26, 27 et 31}. Les nœuds intermédiaires, déterminés par l'étape 2 de l'algorithme, comprennent 14 nœuds {3, 6, 7, 23, 29, 30, 33, 34, 40, 49, 51, 54, 58 et 60}. La taille de la dorsale est égale à 25. En utilisant les deux autres approches CDS-*based* et WCDS, on retrouve pour le même exemple une taille égale à 28 et 27 respectivement.

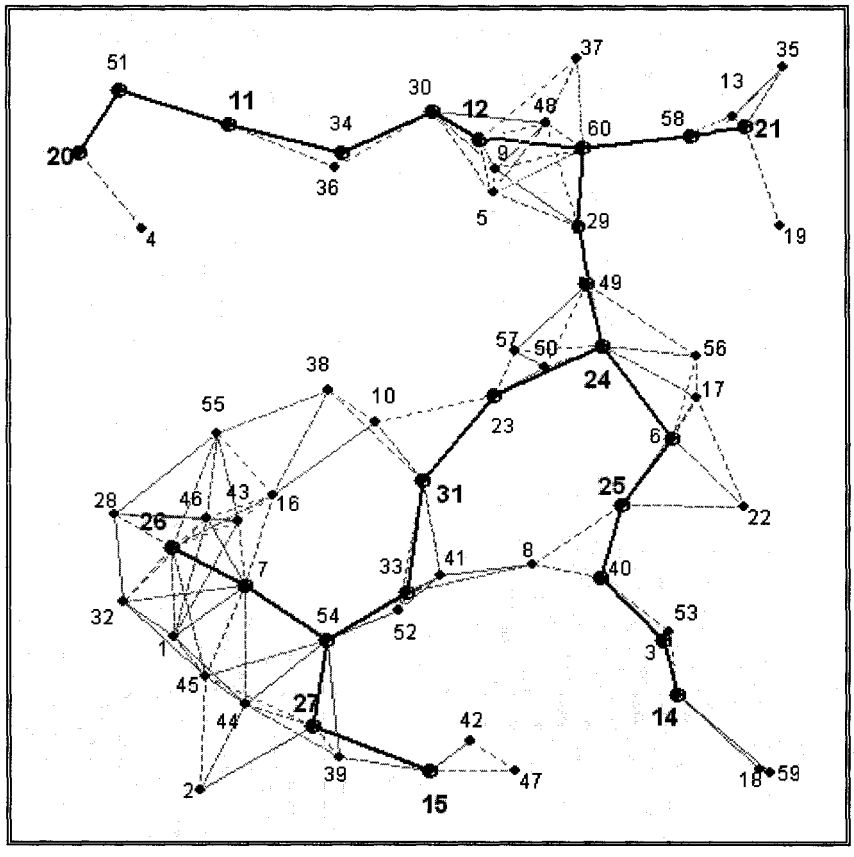


Figure 25 Graphe généré pour $N = 60$ et $\Delta = 4.8$

Pour comparer les trois approches (MDS-based, WCDS, CDS-based et B-CDS dans ces deux versions centralisée et distribuée) nous allons déterminer la taille de l'ensemble de domination connexe MCDS en fonction de la taille du graphe. Pour cela, nous prendrons des graphes dont la taille varie entre 40 et 320 nœuds ($40 \leq N \leq 320$). Pour chaque valeur de N, nous effectuerons plusieurs simulations afin d'obtenir une moyenne statistiquement confiante de la valeur moyenne de la taille de MCDS.

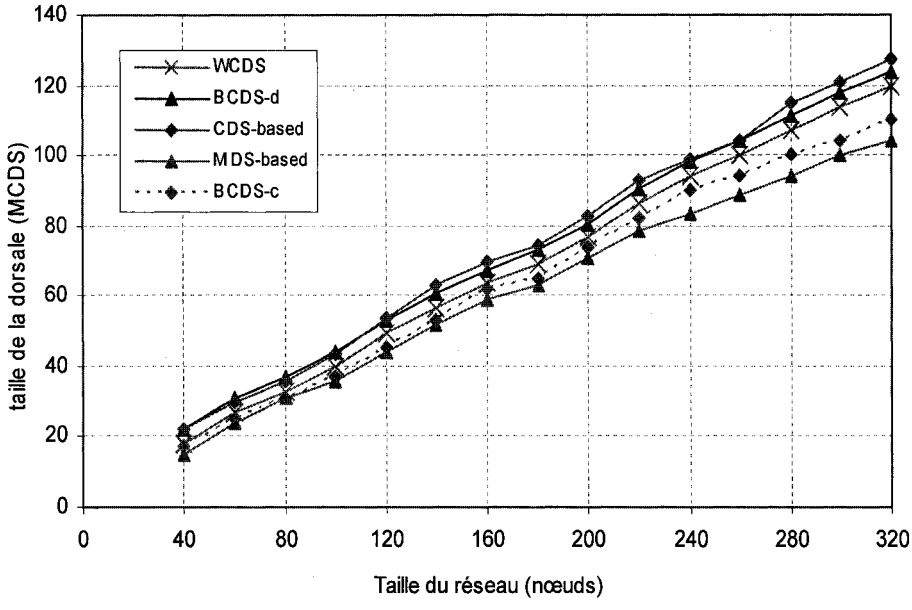


Figure 26 Taille moyenne de l'ensemble MCDS en fonction de la taille du réseau

La figure 26 illustre la taille de l'ensemble MCDS calculé en utilisant les algorithmes WCDS, CDS-based et BCDS dans ces deux versions centralisée et distribuée décris dans la section 2.5.3, ainsi que notre algorithme présenté dans ce chapitre MDS-based, en fonction de la taille du réseau. Ces résultats de simulation montrent bien que les algorithmes centralisés (MDS-based et BCDS-c) donnent une taille de l'ensemble MCDS beaucoup plus petite comparé aux algorithmes distribués (WCDS, CDS-based et BCDS-d). Ceci est expliqué par le fait que les algorithmes distribués ne possèdent pas une information globale du réseau comme dans le cas des algorithmes centralisés. Par ailleurs, l'algorithme proposé dans ce chapitre (MDS-based) fournit une solution minimale de l'ensemble MCDS comparé à l'algorithme BCDS dans sa version centralisée. L'écart est nettement remarquable quand la taille du réseau est importante. Ces résultats prouvent que notre algorithme donne une solution minimale étant donné que la formulation de la première étape donne une solution exacte de l'ensemble de domination de taille minimale. Notre algorithme, dans sa première étape, garantit une

solution minimale de l'ensemble des dominants. En effet, il minimise la taille de l'ensemble de dominants tout en sélectionnant les nœuds ayant le plus haut degré (ayant plus de voisins) tout en dans le réseau. La figure 27 illustre le degré moyen des nœuds sélectionnés par chaque algorithme pour former l'ensemble MCDS en fonction de la taille du réseau. Ces résultats montrent bien que les nœuds sélectionnés par notre algorithme (MDS-based) possèdent le plus haut degré. Pour un même nombre de nœud dans un réseau, en choisissant les nœuds ayant plus de voisins on aura un ensemble de domination connexe de taille plus faible.

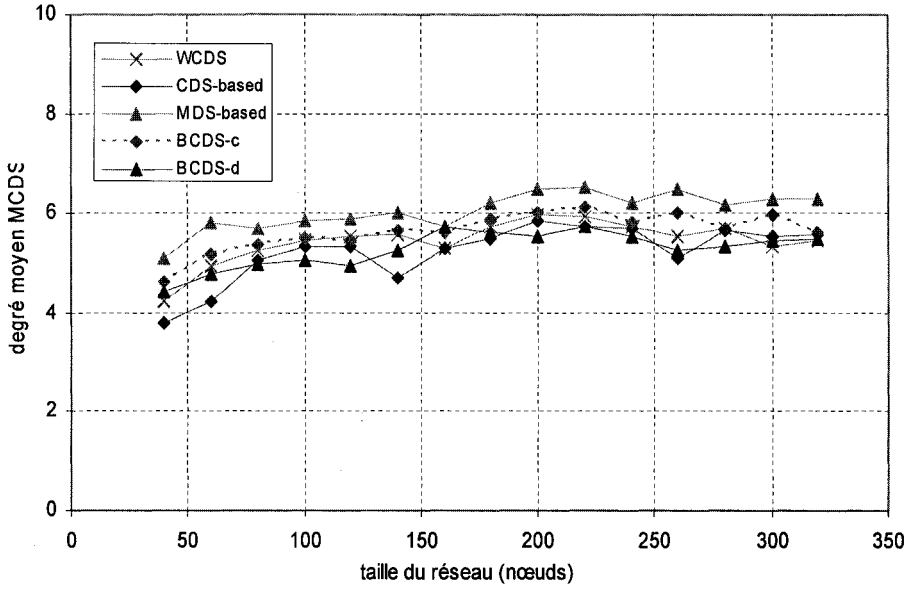


Figure 27 Degré moyen en fonction de la taille du réseau

Au niveau de la complexité, le temps d'exécution de la première étape de notre algorithme est de l'ordre de $O(n)$ et le temps d'exécution de la deuxième étape et de l'ordre de $O(m+n.log\ n)$. Alors la complexité totale de notre algorithme MDS-based est de l'ordre de $O(m+n.log\ n)$ dans le pire cas. Le tableau suivant compare les trois algorithmes en termes de temps d'exécution et de facteur d'approximation par rapport à la solution optimale. Le facteur d'approximation est défini comme étant le rapport de la taille de la dorsale construit par la taille de l'ensemble MCDS.

Tableau IV

Comparaison de la complexité des différents algorithmes

	W-CDS	CDS-based	B-CDS		MDS-based
Type	distribué	distribué	distribué	centralisé	centralisé
Temps d'exécution	$O(\Delta^3_{max})$	$O(n^2)$	$O(nlog^3n)$	$O(mn)$	$O(m+nlogn)$
Facteur d'approximation	$2+\ln\Delta$	$1+\ln\Delta_{max}$			δ

Δ_{max} : degré maximum dans le graphe G

δ : approximation de la solution pour l'arbre de recouvrement minimum ($\delta=2+\varepsilon$, avec $\varepsilon>0$) [Arora et Karakostas (2000)]

3.5 Conclusion

Dans la première partie de ce chapitre, nous nous sommes intéressés à la construction d'une dorsale virtuelle basée sur l'approximation de l'ensemble de dominations connexe de taille minimale (MCDS) dans un graphe UDG. Nous avons proposé un algorithme (MDS-based) composé de deux étapes; la première étape consiste à trouver l'ensemble des nœuds dominants (MDS) dans un graphe et la deuxième étape détermine l'arbre de recouvrement d'un graphe réduit contenant l'ensemble MDS pour, finalement, extraire l'ensemble des nœuds qui forme la dorsale.

La deuxième partie était consacrée à l'étude de performance de notre solution. En utilisant un modèle probabiliste, nous avons vérifié à l'aide des simulations que la solution fournie par notre algorithme est très proche de celle trouvée par analyse. Les simulations ont été faites en variant le nombre de nœuds dans le réseau et le rayon de transmission des nœuds.

Dans la dernière partie de ce chapitre, nous avons comparé notre algorithme avec trois autres algorithmes proposés dans la littérature. Les résultats de simulation montrent que

notre approche performe mieux que les autres algorithmes en fournissant un ensemble MCDS de taille plus petite. Comme nous l'avons mentionné auparavant, plus la taille de l'ensemble des nœuds qui forment la dorsale est minimale plus le résultat est meilleur.

Dû à l'absence d'une infrastructure fixe dans les réseaux *ad hoc*, les nœuds doivent échanger les messages de contrôle utilisés par les protocoles de routage. Le trafic de contrôle génère un trafic redondant et consomme une bonne partie de la bande passante, ressource rare dans les réseaux *ad hoc*. Une dorsale virtuelle construite à partir de l'ensemble MCDS permet de réduire l'ensemble des nœuds qui sera désigné pour retransmettre les messages de contrôle. Ainsi, le trafic de contrôle sera minimisé. Dans le chapitre suivant, nous allons évaluer notre approche comme technique de diffusion en la comparant avec d'autres techniques de diffusion utilisées dans les réseaux *ad hoc* tels que l'inondation pure, la technique de relais multipoint etc.

CHAPITRE 4

DORSALE VIRTUELLE : UNE TECHNIQUE EFFICACE DE DIFFUSION

4.1 Introduction

Les terminaux dans un réseau *ad hoc* utilisent la transmission radio comme moyen de communication entre eux. D'une façon générale, la destination ne se trouve pas à la portée de la source, on a recourt alors au relayage des paquets de données afin d'atteindre la destination. Le relayage est ainsi réalisé par des nœuds intermédiaires. Dans le but d'échanger les informations de mise à jour sur la position, le nombre de voisins etc., les paquets de contrôle générés par les protocoles de routage doivent être diffusés dans tout le réseau. Un trafic de contrôle important dans un réseau *ad hoc* entraînera une grande consommation de la bande passante dans le réseau (ressource rare dans les réseaux *ad hoc*). De plus, il engendrera une augmentation de l'énergie dissipée par le terminal ce qui limite sa durée de vie.

Pour diffuser les messages de contrôle, les protocoles réactifs tels qu'AODV et DSR utilisent l'inondation pure. Le principe est simple : chaque nœud qui reçoit un message de contrôle pour la première fois le diffuse à tous les nœuds voisins situés dans sa zone de couverture. Ce processus est répété jusqu'à atteindre tous les nœuds. Ce qui résulte en un grand nombre de messages redondants à chaque nœud. L'inondation est une technique de diffusion simple et efficace, mais elle est coûteuse. Dans un environnement comme celui des réseaux *ad hoc*, dynamique et sans infrastructure, elle dégrade la performance du réseau en terme de consommation excessive de la bande passante.

Dans la première partie de ce chapitre nous décrirons les techniques de diffusion utilisées dans les réseaux *ad hoc* sans fil. La deuxième partie sera consacrée à la comparaison des performances pour la diffusion de la technique basée sur la construction d'une dorsale virtuelle en la comparant avec la technique de diffusion pure

et la technique utilisant les relais multipoint (utilisé par le protocole OLSR). Cette comparaison sera effectuée dans le cas des réseaux larges et denses (1024 nœuds).

4.2 Techniques de diffusion dans les réseaux

La diffusion est l'opération qui consiste à envoyer un message à tous les membres du réseau. De nombreux algorithmes ont été proposés afin d'effectuer cette opération en essayant de limiter au maximum son coût. La qualité d'un algorithme de diffusion va dépendre de différents critères :

- **la fiabilité :** c'est sa capacité à joindre effectivement tous les nœuds du réseau. Une technique fiable doit permettre de joindre la totalité du réseau si ce dernier est connexe.
- **la consommation énergétique :** si on se place dans un contexte où l'énergie des stations est limitée (ce qui est le cas dans les réseaux *ad hoc* où les terminaux utilisent des batteries et la durée de vie du terminal dépend de l'autonomie de sa batterie), il est important de consommer le moins d'énergie possible. L'énergie consommée dans un terminal dépend principalement du processeur, du traitement des paquets reçus et transmis et de l'interface radio qui se charge de la transmission des paquets. En général, l'énergie consommée par le processeur est négligeable devant celle consommée par l'interface radio qui peut alors être considérée comme une fonction du nombre d'émissions de messages de diffusion.
- **le délai moyen :** ce qui représente le délai moyen entre le début de la diffusion et la dernière réception du message diffusé.

Plusieurs solutions ont été proposées dans la littérature pour gérer et optimiser la diffusion dans les réseaux *ad hoc*. La diffusion dans les réseaux *ad hoc* est très différente du cas des réseaux filaires où un routeur connaît a priori les routeurs qui lui sont reliés.

Une transmission radio est par nature une diffusion, et tous les voisins à la portée sont des récepteurs potentiels; quant aux nœuds hors portée, ils ont besoin d'un ou de plusieurs relayeurs pour recevoir cette information. Le défi est de savoir comment procéder de façon plus efficace, tout en limitant aussi bien la consommation de la bande passante radio que la consommation d'énergie.

Il existe différentes techniques de diffusion dans les réseaux sans fil :

- **inondation pure :** c'est la solution la plus classique. Chaque participant du réseau rediffuse le message reçu s'il le reçoit pour la première fois. C'est une méthode facile à implémenter et elle ne nécessite aucun trafic de contrôle supplémentaire. Son inconvénient est qu'elle est coûteuse en termes de bande passante et qu'elle provoque des flux excessifs de messages redondants inutiles qui peuvent dégrader la performance du réseau. Néanmoins, il existe des algorithmes qui tentent de rendre efficace cette méthode ; à titre d'exemple, on peut citer l'algorithme proposé par Paruchuri *et al.* dans [Paruchuri *et al.* (2002)],
- **division en *clusters*** : cette technique consiste à diviser le réseau en clusters et à élire un *clusterhead* pour chaque cluster. Le *clusterhead* doit avoir connaissance de chaque membre de son groupe. La diffusion s'effectue alors via les *clusterhead* de proche en proche [Basagni (1999)]. Cette méthode pose le problème de division du réseau en *clusters* et l'élection d'un *clusterhead.* Les membres d'un même *cluster* doivent se mettre d'accord sur la méthode d'élection. Ceci est délicat et peut mener à des situations d'instabilité à cause de la nature changeante de la topologie du réseau *ad hoc*,
- **construction d'une dorsale virtuelle** : elle repose sur la détermination de l'ensemble de domination connexe de taille minimale. Les messages de contrôle seront diffusés à l'intérieur de cet ensemble de nœuds formant la dorsale, et les messages seront relayés aux terminaux à un saut,

- **Relais multipoint :** dans cette technique les nœuds choisissent un ensemble de relayeurs par élection ce qui lui permet d'acheminer les informations vers ses voisins à deux sauts. Afin de couvrir tout le réseau, il faut appliquer le même processus de façon récursive [Qayyum *et al.* (2002)]. Cette technique a été conçue pour être utilisée conjointement avec le protocole de routage OLSR.

Dans [Stojmenovic *et al.* (2001)], les auteurs ont décrit, avec plus de détails, les techniques de diffusion décrites ci-dessus. Dans la suite, nous allons comparer les performances de la technique de dorsale virtuelle avec les techniques d'inondation pure et les relais multipoint.

4.3 Résultats de simulation

Nous comparons au moyen de la simulation les performances des trois techniques; dorsale virtuelle, relais multipoint et inondation pure dans un environnement des réseaux *ad hoc* sans fil [Mnif *et al.* (2005b)]. On introduit un nouveau paramètre de simulation α_p [Sasson Y. *et al.* (2002)]. La quantité α_p est la probabilité à la réception, en d'autres termes c'est la probabilité pour qu'un message envoyé soit reçu correctement par un nœud de réception. Lorsque $\alpha_p = 1$, tous les nœuds du réseau reçoivent le message diffusé. Alors que pour $\alpha_p = 0$, le message n'est reçu par aucun nœud.

Nous allons examiner jusqu'à quelle limite l'algorithme de dorsale virtuelle est capable d'assurer la diffusion et de garantir de bons résultats. Ces résultats seront comparés aux résultats utilisant l'inondation pure et les relais multipoint. Il est clair que l'inondation présente une meilleure fiabilité et une robustesse mais elle est gourmande en bande passante ce qui diminue la performance du réseau en terme d'utilisation des ressources disponibles. Nous allons considérer un grand réseau en termes de nombre de nœuds. Les nœuds auront un nombre important de voisins, le réseau sera donc dense et connexe.

La simulation consiste à faire varier la probabilité de succès à la réception α_p entre 0 et 1. Pour chacune des valeurs, nous appliquons la diffusion d'un message d'un nœud vers le reste du réseau. Cette étape est répétée pour tous les nœuds du graphe dans le but de déterminer une valeur moyenne du paramètre à mesurer pour chaque valeur de α_p. Nous allons examiner les paramètres suivants :

- **temps pour diffuser un message** : le temps nécessaire pour qu'un message envoyé du nœud central soit diffusé dans tout le réseau,
- **nombre moyen des messages répliqués** : le nombre moyen de fois qu'un nœud reçoit le même message,
- **nombre de retransmission** : le nombre de fois que le message est relayé et diffusé à travers le réseau,
- **nombre de nœuds qui reçoivent la diffusion** : le nombre total des nœuds qui ont reçu le message diffusé.

Pour mieux cerner le champ de notre étude, nous allons introduire quelques hypothèses qui seront appliquées au cours de la diffusion. On suppose qu'il n'y a pas de mécanisme de réémission quand il y a une erreur de réception. De plus, les liens sont parfaitement symétriques, c'est-à-dire que la matrice d'incidence du graphe composée de 0/1 est parfaitement symétrique. On suppose aussi qu'à chaque fois qu'un nœud émet un paquet, ses voisins à un saut le reçoivent avec une probabilité p_r. On suppose qu'aucun routage n'est considéré, puisqu'on s'intéresse uniquement à la diffusion. D'autre part, il n'y a pas de trafic autre que le paquet à diffuser. Et finalement, un nœud ne retransmet un paquet que s'il le reçoit pour la première fois, autrement il l'ignore.

La topologie utilisée dans notre simulation est constituée de 1024 nœuds. Les nœuds sont dispersés sur une grille de 32 nœuds par 32 nœuds. Chaque nœud peut avoir jusqu'à 48 voisins au maximum. Les voisins sont choisis aléatoirement dans un rayon de trois

sauts. La probabilité d'être un voisin diminue avec la distance. Dans la simulation, nous avons remarqué que le nombre de voisins varie entre 22 et 36.

Le but de cette simulation est d'évaluer la technique de la diffusion en utilisant la dorsale en fonction du taux de succès à la réception. Alors pour une valeur de α_p donnée, il suffit de choisir aléatoirement, pour chaque voisin, la valeur de $p_r \in [0,1]$ et la comparer avec α_p pour déterminer le voisin qui recevra le message ou non. En d'autres termes :

Si $p_r > \alpha_p$ alors la transmission est réussie, et le voisin reçoit le message diffusé,
Si $p_r \leq \alpha_p$ alors le voisin ne reçoit pas le message diffusé.

Rappelons que α_p est le paramètre de simulation avec $0 \leq \alpha_p \leq 1$. Le but étant de prendre en compte l'erreur. La figure 28 présente un exemple d'illustration. Les nœuds en noirs recevront le message envoyé par la source, située au centre, alors que les nœuds en gris ne recevront pas ce message.

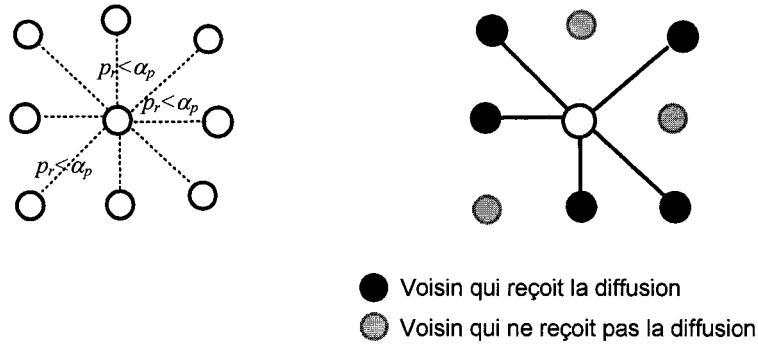


Figure 28 Illustration des voisins qui reçoivent la diffusion

Généralement, les erreurs à la réception sont dues à la collision et d'autres phénomènes. Dans cette évaluation, nous ne nous intéressons pas à la collision mais plutôt aux autres phénomènes. Alors pour ne pas avoir de collision, nous avons utilisé un système de

blocage à deux niveaux. Pour qu'un nœud puisse émettre, il ne faut pas que l'un de ses voisins à deux niveaux possède des messages à émettre ou il est en train d'émettre. Si c'est le cas, alors l'un des deux nœuds émetteurs est bloqué. Ceci a été utilisé afin d'éliminer le problème d'interférence lorsqu'un nœud reçoit deux transmissions radio en même temps, figure 29. On peut vérifier qu'un blocage à deux niveaux est suffisant pour assurer cela, car récursivement, ceci est généralisé sur tout le réseau [Jacquet *et al.* (2002)].

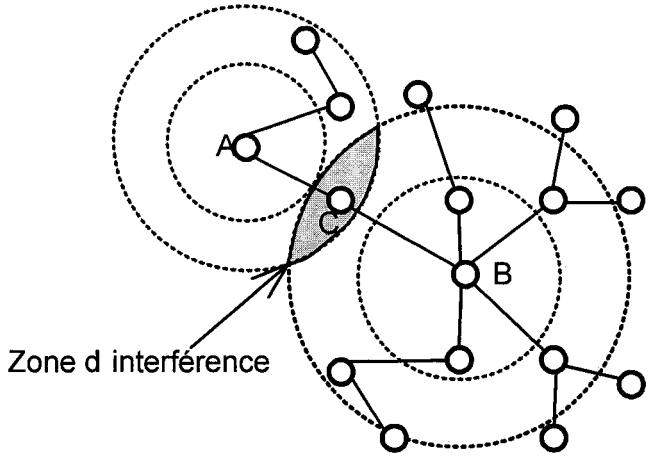


Figure 29 Interférence en C due aux émissions de A et B

La figure 30 illustre le temps nécessaire pour diffuser un message dans tout le réseau en fonction de la probabilité d'erreur à la réception. Ce temps est donné en % du fait que nous avons normalisé par la valeur maximale retrouvée en simulation. Dans le cas de l'inondation, ce temps croît rapidement pour atteindre son maximum à 100 % qui correspond à la valeur de $\alpha_p = 0.10$ de probabilité de succès à la réception, puis il décroît avec une vitesse faible pour se stabiliser à (80 %) qui correspond à la valeur de $\alpha_p = 1$ de probabilité de succès à la réception.

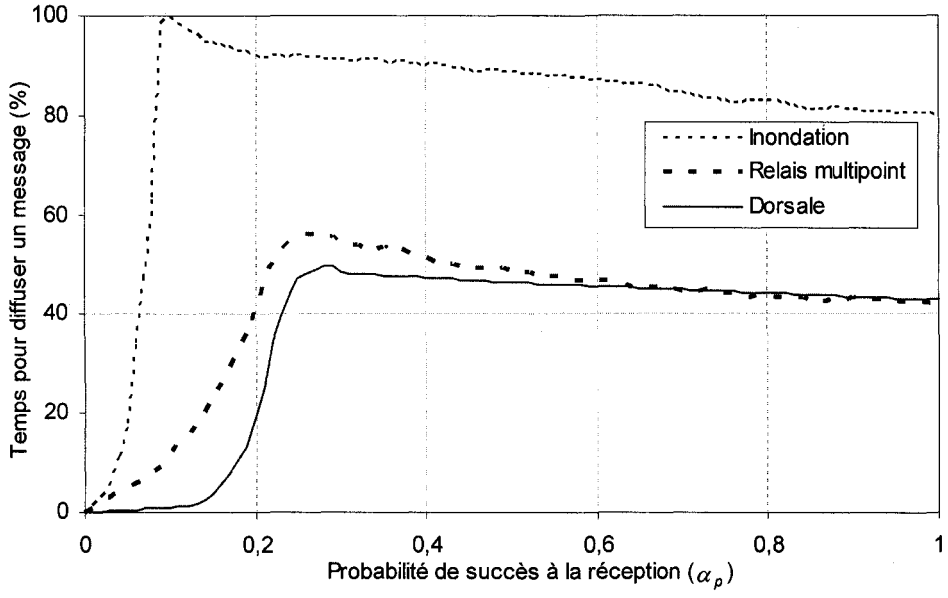


Figure 30 Temps nécessaire pour diffuser un message en fonction de la probabilité de succès à la réception

4.3.1 Temps pour diffuser un message

D'après les résultats de la figure 30, la diffusion se termine beaucoup plus rapidement dans le réseau en utilisant la technique de dorsale et la technique des relais multipoint. Ceci permet d'économiser le temps de diffusion et la bande passante. Ces deux techniques suivent la même trajectoire pour des probabilités de succès à la réception supérieures à 0,4. En utilisant la dorsale, le temps nécessaire pour diffuser un message croît pour atteindre son maximum à 48% qui correspond à la valeur de α_p = 0,30, avec une vitesse moins importante que la diffusion. Ce temps décroît légèrement pour se stabiliser à 43 % pour α_p = 1. Le grand pic dans le cas d'inondation s'explique par le fait qu'avec une faible probabilité de succès à la réception il existe encore des nœuds qui reçoivent le message pour la première fois avec un peu de retard. Ce qui n'est pas le cas lorsqu'on utilise la dorsale ou les relais multipoint.

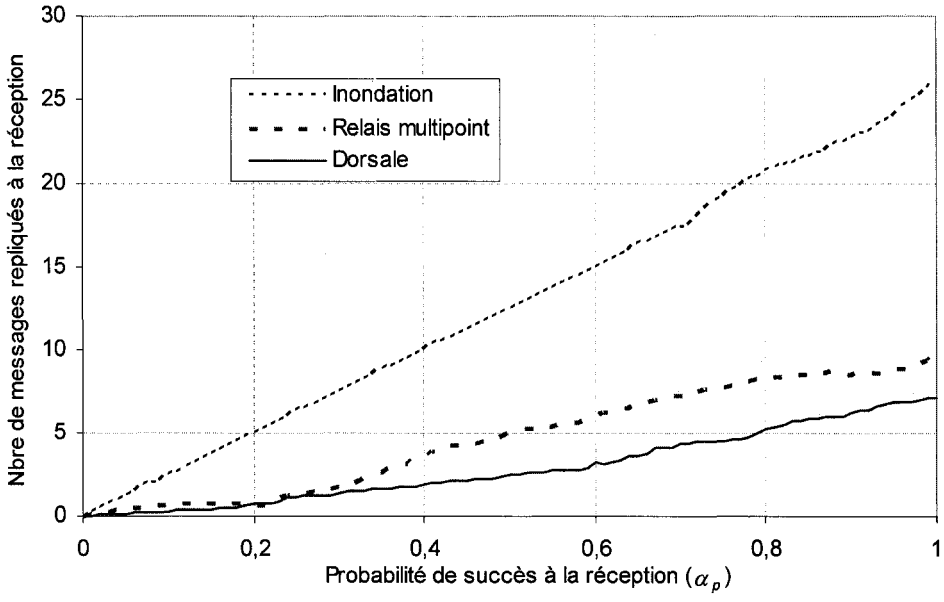


Figure 31 Nombre moyen des messages en fonction de la probabilité de succès à la réception

4.3.2 Nombre moyen des messages répliqués

La figure 31 représente le nombre moyen de fois qu'un nœud reçoit le même message. Le nombre de messages reçus par chaque nœud croît proportionnellement avec la probabilité de succès à la réception. En utilisant la dorsale, le nombre moyen de message ne dépasse pas 8; il est de l'ordre d'un tiers comparé au nombre dans le cas de l'inondation; et il est légèrement inférieur au nombre dans le cas du relais multipoint. Il est à noter que pour une probabilité de succès égale à 1, pour le cas d'inondation, le nombre de messages répliqués est égal à 26 ce qui représente le nombre de voisins moyens dans tout le réseau.

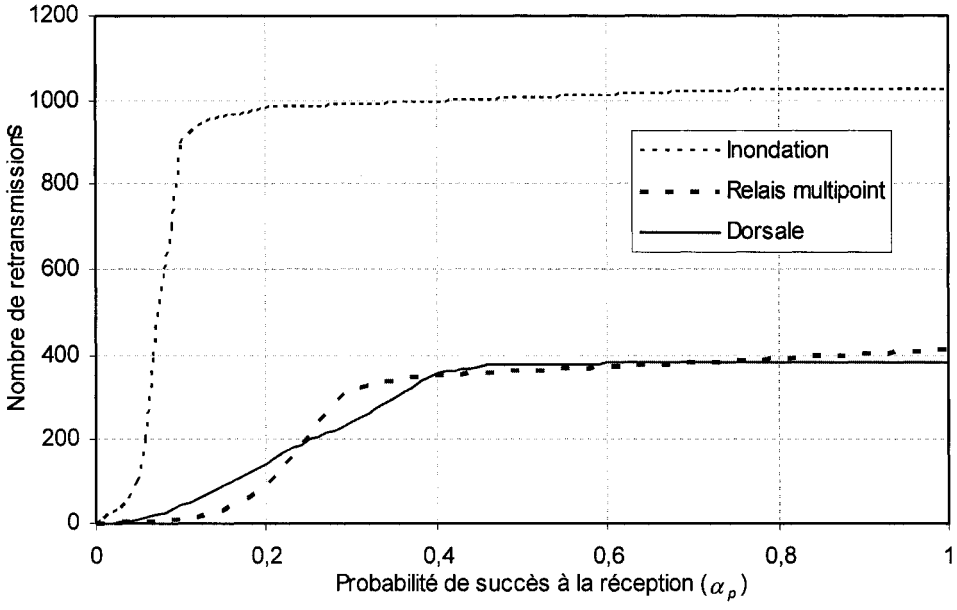


Figure 32 Nombre de retransmissions dans le réseau en fonction de la probabilité de succès à la réception

4.3.3 Nombre de retransmissions

Les courbes de la figure 32 illustrent le nombre total de retransmissions nécessaires en fonction de la probabilité de succès à la réception pour atteindre l'ensemble de tous les nœuds dans le réseau, soit 1024 nœuds. Dans le cas de l'inondation, ce nombre croît rapidement en fonction de la probabilité de succès à la réception pour atteindre le nombre de nœuds dans le réseau. Ce nombre est beaucoup moins important dans le cas de la dorsale et du relais multipoint. Pour une probabilité de succès à la réception égale à 1, le nombre de retransmissions est multiplié par trois dans le cas de l'inondation par rapport à la technique utilisant la dorsale ou les relais multipoint. Ceci est très intéressant du fait que cela nous permet d'épargner les retransmissions inutiles tout en atteignant tous les nœuds dans le réseau. Pour atteindre les 1024 nœuds dans le réseau, on a eu besoin seulement de 400 retransmissions en utilisant la dorsale.

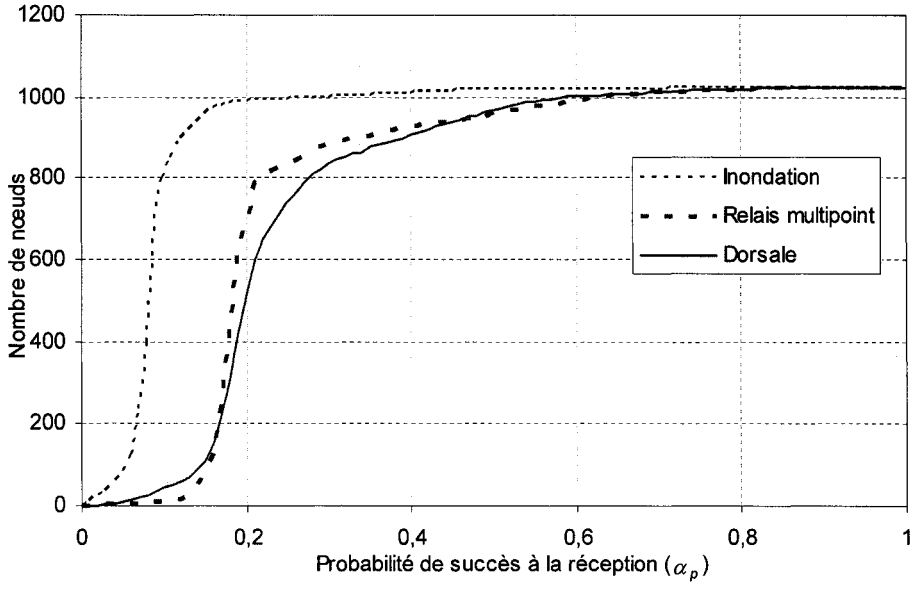


Figure 33 Nombre de nœuds qui ont reçu la diffusion en fonction de la probabilité de succès à la réception

4.3.4 Nombre de nœuds qui reçoivent la diffusion

La figure 33 illustre le nombre de nœuds qui ont reçu le message diffusé en fonction de la probabilité de succès à la réception. Ce résultat projette en quelque sorte l'efficacité de la technique de diffusion utilisée. Les techniques de dorsale virtuelle et de relais multipoint performent moins bien que la technique d'inondation quand la probabilité de succès à la réception est inférieure à 0.6. Au delà de cette valeur de probabilité, ces deux techniques peuvent atteindre plus de 95% des nœuds dans le réseau. Les trois techniques possèdent le même comportement quand la probabilité de succès à la réception est proche de 1.

4.4 Conclusion

Dans ce chapitre, nous avons évalué la robustesse de la diffusion en utilisant la dorsale virtuelle. Les résultats ont été comparés avec la méthode d'inondation pure et la méthode utilisant les relais multipoint. Les résultats montrent que la diffusion, en

utilisant la dorsale, est une bonne alternative pour économiser la bande passante dans le réseau.

L'utilisation d'une dorsale améliore le temps nécessaire pour que la diffusion d'un message atteigne au moins une fois tous les nœuds. Ce temps est très comparable à celui donné par les relais multipoint. Dans ces deux cas, la diffusion se termine plus rapidement si on la compare avec la méthode d'inondation pure. Ceci est dû au nombre réduit de nœuds responsable du relayage des messages. Les nœuds sont moins submergés par les copies multiples du même message, ce qui fait qu'on épargne la bande passante et le traitement des messages au niveau de chaque nœud. La technique de la dorsale virtuelle performe mieux que la technique de relais multipoint en termes du nombre moyen de messages répliqués et du nombre de retransmission, ceci a pour effet de diminuer le trafic redondant dans tout le réseau. La technique de dorsale virtuelle donne une performance comparable à la technique utilisant les relais multipoint en ce qui concerne les nœuds qui reçoivent le même message diffusé pour une probabilité de succès $\alpha_p \geq 0.6$. En pratique, la probabilité de succès est au delà de 80 %. Par conséquent, on peut dire qu'avec la dorsale, nous garantissons de très bons résultats dans le cas des réseaux denses.

CHAPITRE 5

MAINTENANCE DE LA DORSALE VIRTUELLE

5.1 Introduction

Dans un réseau *ad hoc*, un nœud est libre de se déplacer à n'importe quel moment vers n'importe quelle direction. Il peut également ne plus être disponible; sa batterie est épuisée. Ceci a pour effet que tous les liens joignant ce nœud et ses voisins ne seront plus disponibles. Ce qui résulte en une topologie dynamique dans le temps. Comme nous l'avons mentionné dans le Chapitre 4, une dorsale virtuelle construite à partir d'une approximation de l'ensemble MCDS permet de limiter le trafic de contrôle dans le réseau. Rappelons que la détermination de la dorsale nécessite une connaissance globale de la topologie du réseau. Pour ne pas recommencer la construction de la dorsale à chaque changement la topologie, une solution consiste à appliquer une procédure de maintenance. La procédure de maintenance sera chargée de réarranger la structure de la dorsale. Un nœud qui change de position va chercher à se connecter à la dorsale. L'auto-organisation de la structure de la dorsale est alors indispensable pour aider les protocoles de routage à mieux performer.

Dans un tel environnement, la construction de la dorsale est un phénomène rare vu que la connaissance des positions des nœuds est nécessaire, alors que la fonction maintenance est omniprésente pendant toute la durée de vie du réseau. La maintenance s'applique chaque fois qu'il y a détection de mouvement ou disparition d'un nœud dans le réseau. En d'autres termes, le nœud qui change de position ou désire rejoindre à un réseau existant va appliquer la procédure de maintenance pour pouvoir se connecter à la dorsale. Pour appliquer la procédure de maintenance, il faut y avoir une dorsale déjà en

place et des nœuds sont identifiés comme dominants, ceux qui forment la dorsale, et d'autres comme dominés, ceux qui ont au moins un voisin dans la dorsale. La procédure de maintenance ne sera pas la même pour un nœud dominant et un nœud dominé.

La durée de vie d'une dorsale est alors directement liée à la taille de la dorsale et à l'efficacité du mécanisme de maintenance, puisqu'à chaque fois qu'il y a un changement dans la topologie du réseau, il y aura une nécessité à la maintenance. Ainsi, plus la taille de la dorsale est minimale plus la procédure de maintenance est fiable, plus la durée de vie du réseau est longue.

La maintenance de la dorsale est l'ensemble de procédures permettant de maintenir cette structure, l'ensemble MCDS doit rester connecté, afin de garder ses propriétés tout au long du fonctionnement du réseau. C'est une problématique rarement considérée dans les propositions construisant la dorsale. Une approche consiste à reconstruire périodiquement la structure. Cependant, elle peut engendrer des délais induits importants par rapport à la fréquence de changement de la topologie qui peut dégrader la performance du réseau. Ainsi, nous proposons une procédure de maintenance complète qui sera appliquée de façon distribuée [Mnif et Kadoch (2006a)], c'est-à-dire chaque nœud qui change de position et se déconnecte de ses voisins la dorsale va appliquer cette procédure pour se connecter à la dorsale par l'intermédiaire du nœud le plus proche suite à un échange de message de contrôle utilisé pour la découverte de voisinage. La performance de cette procédure sera évaluée au moyen de simulations.

5.2 Préliminaires

5.2.1 Terminologie et définitions

Nous traitons le cas général où nous avons à construire des k-CDS. Le k indique qu'un dominé est situé à k-saut par rapport à son dominant, la figure 34 illustre un exemple pour k=2. Il existe des travaux de recherche qui s'intéressent à la construction d'une dorsale à k-saut comme par exemple [Rubin et Vincent (2001)] et [Srivastava et Ghosh

(2002)]. Pour cela, nous voudrons concevoir une procédure générale qui peut servir également à ce type de dorsale. Il est à noter que dans notre recherche nous ne nous sommes intéressés principalement qu'au cas k=1. Toutefois, nous allons comparer la performance d'une telle procédure de maintenance pour différentes valeurs de k.

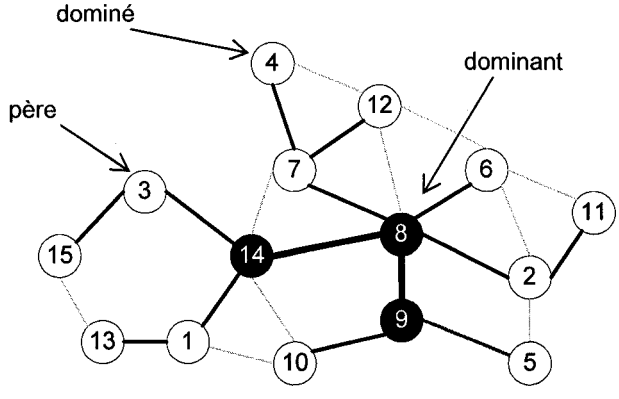


Figure 34 Exemple de 2-CDS

Dans ce qui suit, nous introduisons la terminologie suivante :

- **dominant** : un nœud membre de l'ensemble CDS,
- **dominé** : un nœud qui possède au moins un dominant voisin à k_{cds} sauts,
- **père** : un nœud intermédiaire entre un dominant et un dominé, un nœud père fait partie aussi de l'ensemble des nœuds dominés.

5.2.2 Diagramme d'état

Notre approche pour la maintenance consiste à définir un certain nombre d'états qu'un nœud pourra prendre. Un changement de position pour un nœud entraînera un changement dans les liens radio avec ses voisins. En se basant sur cette observation, nous définissons quatre états possibles pour un nœud :

- **Dominant** : membre de l'ensemble CDS, membre de la dorsale,

- **Dominé** : nœud possédant au moins un dominant à moins de k_{cds} sauts,
- **Actif** : nœud en processus d'être dominant ou dominé,
- **IDLE** : nœud attend une décision d'initialisation.

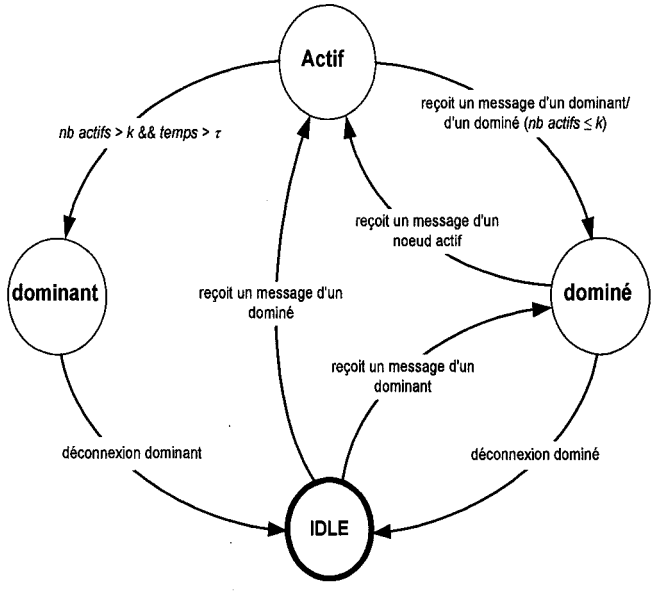


Figure 35 Diagramme d'états

Le diagramme de la figure 35 explique comment on peut passer d'un état à un autre. Le principe de fonctionnent de ce diagramme est comme suit : une fois qu'un nœud change de position, qu'il soit dominant ou dominé, cela va changer son état à IDLE, un état d'initialisation en attendant la réception d'une réponse du voisinage. Si dans le voisinage il existe ou moins un dominant alors le nœud, dans l'état IDLE, va passer à l'état dominé. S'il n'y a aucun dominant au voisinage et qu'il existe un ou plusieurs dominés alors le nœud dans l'état IDLE va alors changer son état à Actif. Un nœud dans un état Actif va changer d'état à dominant s'il est voisin direct d'un dominant et s'il existe k nœuds ou plus, dans l'état actif pour le même dominant et que le temps depuis qu'il a envoyé son premier paquet *hello* est supérieur à τ (où τ un temporisateur qui sera défini ultérieurement). Un nœud Actif peut devenir dominé s'il reçoit un message d'un dominant ou d'un dominé qui se retrouve à une distance $\leq k$ sauts. Il est à noter qu'un nœud dans un état dominé recevant un message *hello* d'un nœud actif va changer son

état à actif. Car, comme nous venons de voir, un nœud dominé peut devenir Actif et par la suite il devient dominant s'il existe plus que k nœuds Actif pour le même dominant. Dans le cas où un nouveau nœud veut rejoindre un réseau en cours, il va se mettre dans l'état IDLE et suivant son voisinage va changer son état à dominé ou dominant en suivant la même logique décrite ci-haut.

Plus formellement, la procédure de maintenance s'écrit :

```
forall x / etat(x) = IDLE do
   if N_D(x) ≠ Ø                                  /* il existe un nœud dominant au voisinage de x  */
      j = argmin {plus_proche(u)| u ∈ N_D(x)}     /* s'il existe plus qu'un nœud dominant au voisinage
                                                     de x, on choisit celui le plus proche avec N_D(x)
                                                     l'ensemble des nœuds dominants au voisinage de x,*/
      etat (x)          = dominé                  /* mettre à jour les attributs de x */
      ID_dominant(x) = j
      nb_saut (x)       = 1
   elseif N_d(x) ≠ Ø                              /* il existe un nœud dominé au voisinage de x  */
      j = argmin {nb_saut(u)| u ∈ N_d(x)}         /* s'il existe plus qu'un nœud dominé au voisinage
                                                     de x, on choisit celui qui a le nb_saut le plus faible */
                                                     avec N_D(x)l'ensemble des nœuds dominants au voisinage de x,*/
                                                  /* le nœud x ne peut pas se connecter au dominant de j, j=k*/
         etat (père(j, nb_saut =1)) = dominant    /* tous les pères du nœud j vont changer d'état à Actif
         for m=2 to nb_saut                          et le premier va changer en dominant
            etat(père(j,m)) = Actif                  N_A(u) l'ensemble de nœuds Actifs du dominant u*/
            N_A(u)= N_A(u)∪ père(j,m)             /* Un père à nb_saut d'un dominé est noté par père(dominé, nb_saut)*/
         end
         D = D ∪ j                                /*  le nœud j va être ajouté à l'ensemble des dominants D dans tout le réseau*/
   end
end
 for all u ∈ D
   for all x ∈ N_A (u)
         if (nb_saut(x) < k && N_d(x) ≠ Ø)
            etat (x)          = dominé            /* mettre à jour les attributs de x */
            ID_dominant(x) = u
         else if (nb_actifs (u) > k)              /* On effectue un test sur le nombre des noeuds actifs du
            etat (x)          = dominant             dominant u
            nb_saut (x)       = 0
            ID_dominant(x) = NULL
         end
   end
 end

return x(ID_dominant, état, nb_saut)              /* Un nœud x possède les attributs suivant
                                                     ID_dominant, état et  nb_saut avec nb_saut le
                                                     nombre de saut depuis le dominant (nb_saut≤k) */
```

5.2.3 Format du paquet *hello* modifié

Dans les protocoles de routage actuels, on utilise le paquet *hello* pour la découverte de voisinage. Nous avons donc choisi d'utiliser ces mêmes paquets classiques dans lesquels nous allons ajouter quelques champs (figure 36). Pour un dominant, il est nécessaire de mettre à jour une liste de ses dominés et des pères associés. Ceci peut se faire en obtenant ces informations des paquets *hello*.

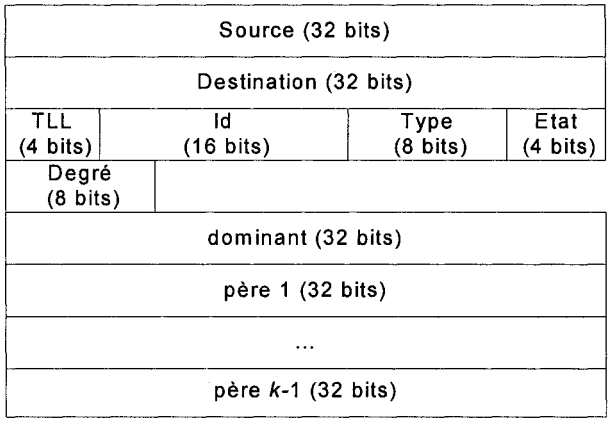


Figure 36 Format du paquet *hello* modifié

Le paquet *hello* comprend les champs classiques tels que les adresses source et destination, le TTL[14], le type de paquet[15] et l'identifiant du paquet.

Nous ajoutons alors un certain nombre d'informations qui seront utiles pour la maintenance de la dorsale, telles que :

- état du nœud : IDLE/actif/dominant/dominé,
- degré : nombre de voisins,

---

[14] *Time To Live* : nombre maximum de sauts à parcourir par un paquet. Ce nombre décrémente à chaque saut.

[15] Un paquet peut être de type *hello*, *unicast*, *data* etc.

- nœud dominant pour un nœud dominé,
- liste des pères pour un nœud dominé.

5.2.4 Temporisateur τ

Un temporisateur τ sert à ce qu'un nœud en mode actif ne se déclare pas dominant avant de vérifier le degré et l'état de tous les nouveaux nœuds actifs. Lorsqu'un nœud se déclare actif, il va diffuser le message de changement d'état à (k-1)-sauts. De même, le dernier nœud qui recevra ce message de changement d'état va le retourner à (k-1)-sauts. Le temporisateur τ doit alors être supérieur au temps aller-retour qu'un paquet va effectuer dans le trajet de (k-1)-sauts. Soit Δ_t le délai maximum qu'un paquet subit pour le traitement, Δ_t est de l'ordre de 20~50 millisecondes.

$$\tau \geq 2 \cdot (k\text{-}1)\, \Delta_t$$

$$\text{pour } \Delta_t = 50ms, \text{ alors} \qquad \tau_{\min}(ms) = 100 \cdot (k\text{-}1)$$

Dans cette expression, nous ne prenons pas en considération le temps de propagation vu qu'il est négligeable comparé au temps de traitement d'un paquet.

5.3 Procédure de maintenance

Dans cette section, nous allons détailler le principe de fonctionnement de la procédure de maintenance proposée dans ce chapitre. Pour réaliser la maintenance, différentes tâches doivent être entreprises par le nœud qui désire se connecter à la dorsale. Tout d'abord, le nœud doit effectuer une découverte au voisinage. Ceci est réalisé en envoyant des messages *hello*. Par la suite, il se connecte à la dorsale. Suivant les messages réponses reçus de son voisinage, une procédure de connexion est alors employée en utilisant le diagramme d'état (figure 35). Deux autres tâches sont

nécessaires pour réaliser la maintenance, il s'agit de maintenir des informations pour chaque nœud et de maintenir les relais vers le dominant.

5.3.1 Découverte au voisinage

Un nœud dominé ou dominant qui se déplace (ou même un nouveau nœud qui désire se joindre au réseau) va donc tenter de se connecter à la dorsale, son état est IDLE. Un nœud envoie un paquet *hello* contenant sa propre adresse et un TTL égal à k. Le paquet est alors diffusé en *broadcast*. Nous avons choisi le TTL = k pour que ce message *hello* puisse arriver à un dominant, s'il existe. Les nœuds voisins vont comprendre que ce nœud est IDLE puisque le champ dominant est nul, et vont donc l'ajouter dans leur table de voisinage. Chaque nœud va alors envoyer une réponse *hello* dans lequel les champs degré, état, dominant et la liste des pères sont bien identifiés. Pour un dominé, il doit indiquer dans son paquet de retour un TTL égal au nombre de pères dans sa liste, ceci afin d'informer son dominant.

Quoique cette méthode possède l'inconvénient de générer beaucoup de paquets (puisque chaque nœud envoie un paquet *hello* à chaque intervalle de *hello*), elle converge rapidement. Chaque nœud connaîtra son voisinage, s'il n'y a pas de perte de paquets de *hello*. Il existe une autre méthode où chaque nœud envoie la table de ses voisins à un saut. Ainsi, un nœud connaîtra de proche en proche son k-voisinage. La convergence de cette méthode est beaucoup plus lente, car la propagation se fait sur un saut malgré que la charge soit moins faible avec un seul paquet de taille plus grande.

5.3.2 Procédure de connexion

Un nœud dans l'état IDLE (et donc n'a pas de dominant) recevant une réponse d'un dominant (ce dernier peut alors couvrir le nœud) va changer son état à dominé et la procédure se termine. Si aucun dominant n'est à son voisinage et si un ou plusieurs

dominés lui répondent, il change son état à actif pour finalement devenir soit dominant soit dominé :

- un nœud actif va être dominant s'il est le père de *k* dominés y compris le dernier nœud qui cherche à se connecter. Ce nœud se trouve à un saut d'un dominant,
- un nœud actif va être dominé si un dominant peut l'accepter comme fils c'est-à-dire s'il se retrouve à une distance inférieure à *k*-sauts du dominant.

En recevant un message d'un nœud actif, un dominé d'ordre k_{CDS} va changer son état à actif parce qu'il pourrait devenir dominant si un nouveau nœud s'ajoute à la branche de *k*-dominés. Le dominé d'ordre 1 change son état à dominant pour pouvoir couvrir le nouveau nœud (dominé). Un dominé qui change son état à dominant se retrouve à un saut d'un dominant, et il sera donc connecté à la dorsale.

Reprenons le même exemple de la figure 34, où nous avons un réseau composé de 15 nœuds et la dorsale est constituée des nœuds 14, 8 et 9 (k_{CDS} =2). Le nœud 13 va se déplacer et le nœud 2, qui était dominé avant la présence du nœud 13, devient un dominant afin de pouvoir couvrir le nœud 13 dans sa nouvelle position (figure 37-b).

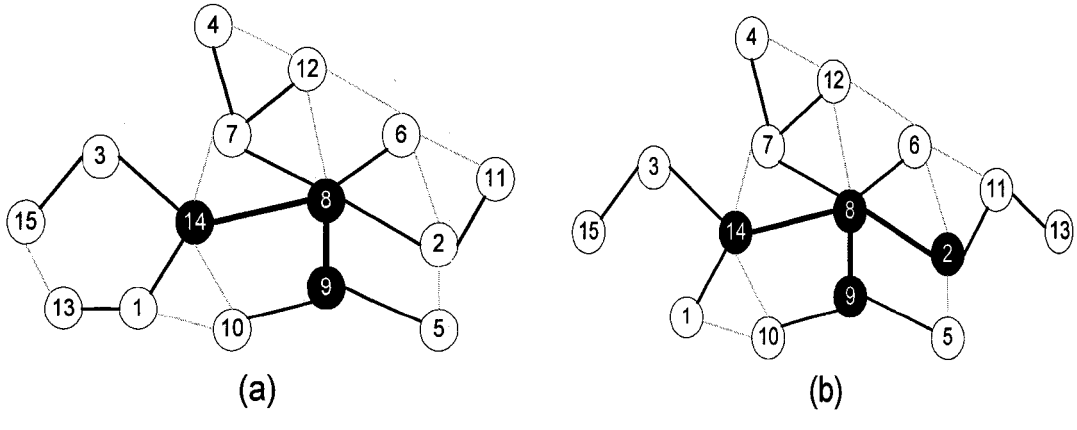


Figure 37 La structure s'auto organise suite à un déplacement d'un dominé

Le nœud 13 dans sa nouvelle position, en état IDLE, va tenter de découvrir le voisinage en envoyant des paquets *hello.* En recevant ces paquets, le nœud 11 va changer son état à actif et va faire savoir qu'il est dominé et qu'il possède un dominant à un saut. Le nœud 2 sachant qu'il y a un nœud qui tente de se connecter, va aussi changer son état à actif, en conséquence de l'action du nœud 11. Dans cet exemple, nous avons k_{CDS}=2, le nœud 2 devra donc changer son état à dominant pour avoir une dorsale connexe.

Considérons maintenant le cas où le nœud 14, dominant, va se déplacer. Les nœuds 3, 15, 13 et 1 qui étaient des dominés du nœud 14 vont changer leur état à IDLE et vont lancer le processus de découverte du voisinage. Dans un premier temps le nœud 10 va donc répondre au nœud 1 en lui indiquant qu'il peut se connecter. Une fois que le nœud 1 répond, le nœud 10 change son état à actif en même temps que le nœud 15. Étant donné que dans cet exemple k_{CDS} = 2, le nœud 10 doit changer d'état à dominant pour couvrir les nœuds 1 et 13. De la même façon, les nœuds 1 et 13 vont devenir dominants pour couvrir le nœud 3 (figure 38). Le nœud 14 qui se retrouve dans sa nouvelle position, dans un état actif, va lancer le processus de découverte de voisinage en envoyant des paquets *hello*. Le nœud 5 va répondre à ces paquets *hello* pour faire savoir qu'il est dominé et qu'il se trouve à un saut d'un dominant, le nœud 14 passe alors de l'état actif à dominé.

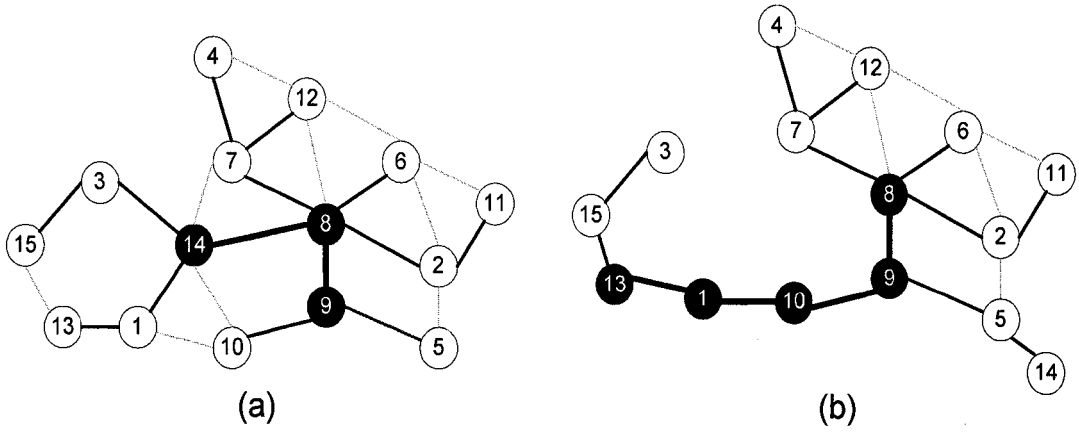


Figure 38 La structure s'auto organise suite à un déplacement d'un dominant

5.3.3 Maintien des informations

Nous avons introduit dans le paquet *hello* certaines informations propres à la dorsale telles que : état, degré et liste des pères. Nous pouvons alors utiliser ces données pour mettre à jour les informations sur la dorsale. Grâce au champ père, nous pouvons mettre à jour la liste des dominés pour chaque dominant. Ces informations peuvent servir à optimiser la structure de la dorsale après avoir effectué plusieurs maintenances successives.

5.3.4 Relais vers le dominant

Avec la structure d'une dorsale, un dominé a toujours besoin de dialoguer avec son dominant. Il doit donc posséder une route vers celui-ci. Un dominant envoie dans sa zone, de façon régulière, des paquets de contrôle ou des paquets *hello*, il suffit alors de noter l'identité du père précédent pour déterminer l'intermédiaire à contacter sur le chemin vers le dominant.

Il est important pour un nœud de pouvoir communiquer avec son dominant quand c'est nécessaire. Un nœud source qui désire envoyer des données vers un autre nœud destination va alors demander à son dominant de trouver un chemin et d'acheminer ce trafic de données. C'est le dominant qui va procéder à la recherche d'une route et à l'acheminement du trafic. Le dominant connaît a priori la destination et le dominant qui lui est associé. Une telle communication est obligatoire pour certaines procédures de maintenance. Nous stockons alors l'identification du nœud (ID) qui relaie les paquets de contrôle venant du dominant. Ce relais est donc mis à jour pour devenir dominé après la réception d'un paquet d'un dominant.

Remarque

Il est important de mentionner qu'il est possible qu'un ou plusieurs nœuds restent non connectés à la dorsale. Dans ce cas, le nœud isolé reste dans un état IDLE jusqu'à ce que la situation change et d'autres nœuds se rejoignent à lui. D'autres cas peuvent aussi se présenter. En effet, deux ou plusieurs sous-graphes peuvent se former après plusieurs maintenances. Dans cette situation, les sous-réseaux correspondants vont alors être indépendants jusqu'à ce qu'un nœud change de position et se trouve à la portée des nœuds qui n'appartiennent pas aux mêmes sous réseaux. Après l'échange périodique des paquets *hello*, le nœud va découvrir que ses voisins n'appartiennent pas à la même dorsale et par conséquent il change son état à dominant, ainsi que les nœuds dominés dans les chaînes vers les dominants, et les deux sous réseaux seront alors connectés. La figure suivante illustre ce cas particulier où nous avons k_{CDS} =2 et à un instant donné deux sous-réseaux sont formés (figure 39-a). Ces deux sous-réseaux restent déconnectés jusqu'à ce que le nœud 4, par exemple, aille changer de position pour se retrouver à la portée des nœuds 11 (du sous-réseau 1) et 3 (du sous-réseau 2). Le nœud 4 ainsi que les nœuds dans les chaînes vers dominants (nœuds 2 et 11 dans le sous-réseau 1 et nœud 3 dans le sous-réseau 2) vont changer d'état à dominant (figure 39-b).

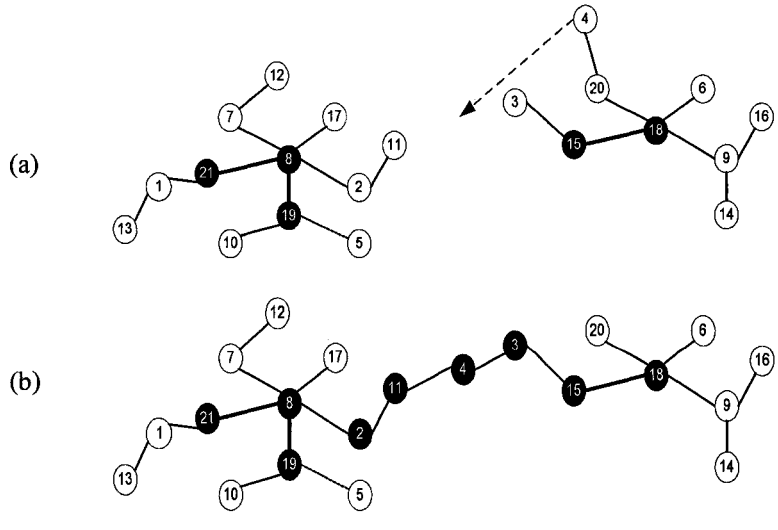


Figure 39 Cas où deux sous-réseaux se forment

Théorème 2 (maintenance)

En appliquant la procédure de maintenance sur une dorsale construite à partir de l'ensemble k_{CDS}, un dominé possède toujours un dominant, et il est d'au plus à une distance de k_{CDS} sauts de ce dominant.

Preuve : Nous supposons que la topologie est stable après quelques maintenances. Un dominé ayant un voisin dominant va choisir ce dernier comme dominant. Supposons qu'un ensemble de dominés se retrouvant à une distance de i sauts de leur dominant ait un père. Un dominé situé à i+1 sauts de son dominant peut choisir ce dernier comme dominant car il est situé à moins de k_{CDS} sauts via un autre dominé ayant choisi le même dominant, mais à i sauts, avec $i< k_{CDS}$. Ainsi, comme le père des dominés à i sauts d'un dominant existe, chaque dominé qui choisit un dominant possède un père.

Un dominé n'ayant aucun dominant possible dans sa table de voisinage (il n'existe aucun voisin qui pourra être choisi comme dominant situé d'au plus à k_{CDS}-1 sauts) il devient actif pour passer finalement à l'état dominant. D'après le diagramme d'état de la figure 35, un nœud actif va devenir dominé s'il existe un dominant valide, ou bien, il va devenir dominant et par la suite il va exécuter la procédure de maintenance d'un dominant. Ainsi, tout dominé possède au moins un dominant situé à moins de k_{CDS} sauts. ■

Théorème 3 (connectivité)

Soit un graphe connexe et une dorsale construite à partir de l'ensemble CDS. Suite à maintenance de la dorsale, l'ensemble des dominants forme un arbre et reste connecté quand les liens radio sont stables.

Preuve :

Supposons que les liens radios sont stables (topologie stable) après quelques maintenances. Chaque dominant reçoit alors les paquets *hello* maintenant ainsi la liste de ses dominés. Soit D_i l'ensemble des dominants qui forment la dorsale, l'ensemble D_i est supposé être connexe. L'indice i représente le nombre maximum de sauts entre deux dominants. Soit l'ensemble D_{i+1} tels que pour tout nœud de D_{i+1} on choisit s l'ensemble D_i, ainsi, l'ensemble D_{i+1} est connexe puisque ces nœuds reçoivent le paquet *hello* de leur père et ils sont alors à un saut de la dorsale.

Supposons que D_i ne contient pas de cycle; E_i est l'ensemble des arcs de D_i, et V_i est l'ensemble de ses nœuds. Nous avons alors $|E_i| = |V_i - 1| = |V_i| - 1$. Pour chaque nœud de l'ensemble D_{i+1} - D_i, on ajoute un nœud dans V_i et un arc dans E_i, alors :

$$| E_{i+1} | = |V_i| - 1 + \left[|V_{i+1}| - |V_i| \right] = |V_{i+1}| - 1$$

Donc D_{i+1} est connexe et ne contient pas de cycle. ■

5.4 Modélisation

Pour évaluer l'efficacité de la procédure de maintenance, *Opnet Modeler* est choisi, en utilisant à la couche 2 le protocole 802.11b qui est déjà intégré dans le simulateur. *Opnet* est un simulateur à événements discrets qui fournit un environnement de développement global permettant de modéliser et d'évaluer les performances des réseaux de communication. Il permet de spécifier le format des paquets, de définir des processus qui

seront représentés par des diagrammes d'états et enfin, de collecter et d'analyser les résultats. Un processus est représenté par un automate état/transition. Chaque état contient le code des tâches à exécuter à l'entrée et à la sortie de l'état. La transition spécifie la condition nécessaire pour passer d'un état à un autre.

La figure 40 illustre le modèle de simulation du plus haut niveau implémenté dans Opnet. Le réseau comprend 50 nœuds sont placés aléatoirement dans une zone carrée de 1,5 *km* x 1,5 *km*. Chaque terminal est représenté par un nœud, modélisé comme un émetteur/récepteur, une couche mac 802.11b (wireless_lan_mac) reliée au dispositif d'émission/réception et une couche MCDS au dessus de la couche 802.11b comme illustré dans la figure 41. Un processus indépendant est ajouté pour modéliser la mobilité d'un nœud et un processus d'interface pour établir l'interface entre la couche MAC et la couche MCDS.

Chaque processus est représenté par un diagramme d'état : lorsqu'on entre dans un état, une série d'instructions sont exécutées; d'autres instructions sont également à la sortie de l'état. Au moment où un événement se produit (interruption, arrivée d'un paquet, etc.), on sort de l'état et suite à une transition vraie pour entrer dans un autre état. Le diagramme d'état développé pour le processus MCDS est illustré à la figure 42.

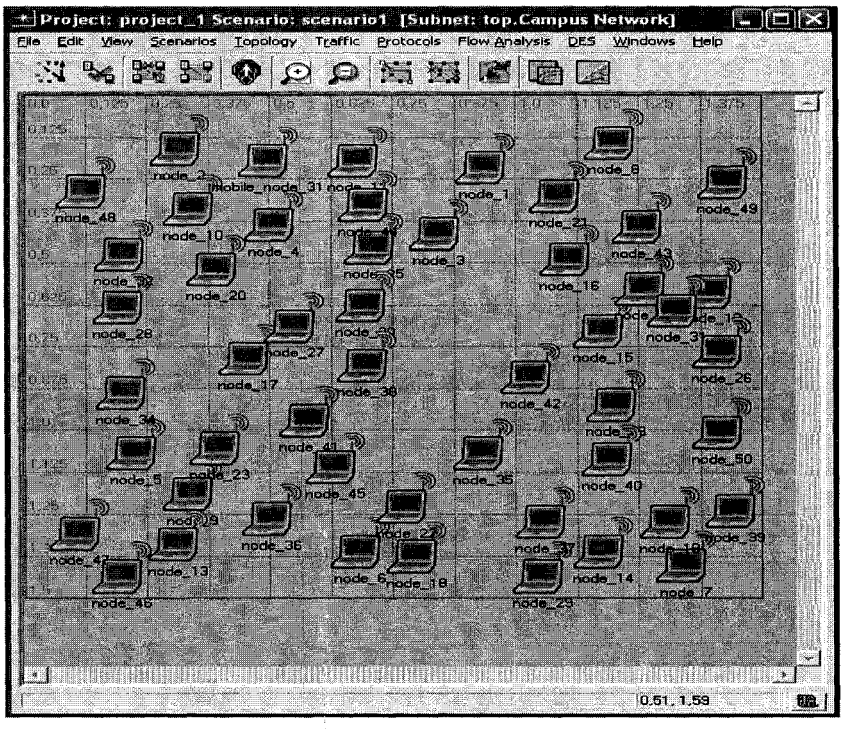


Figure 40 Vue d'ensemble du modèle de simulation (N=50)

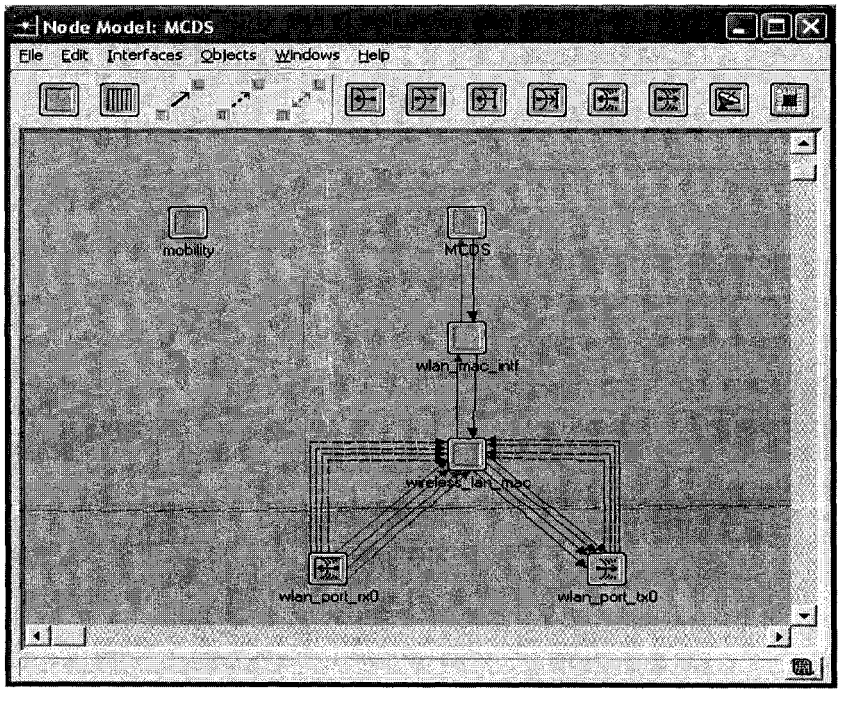


Figure 41 Modélisation d'un terminal *ad hoc*

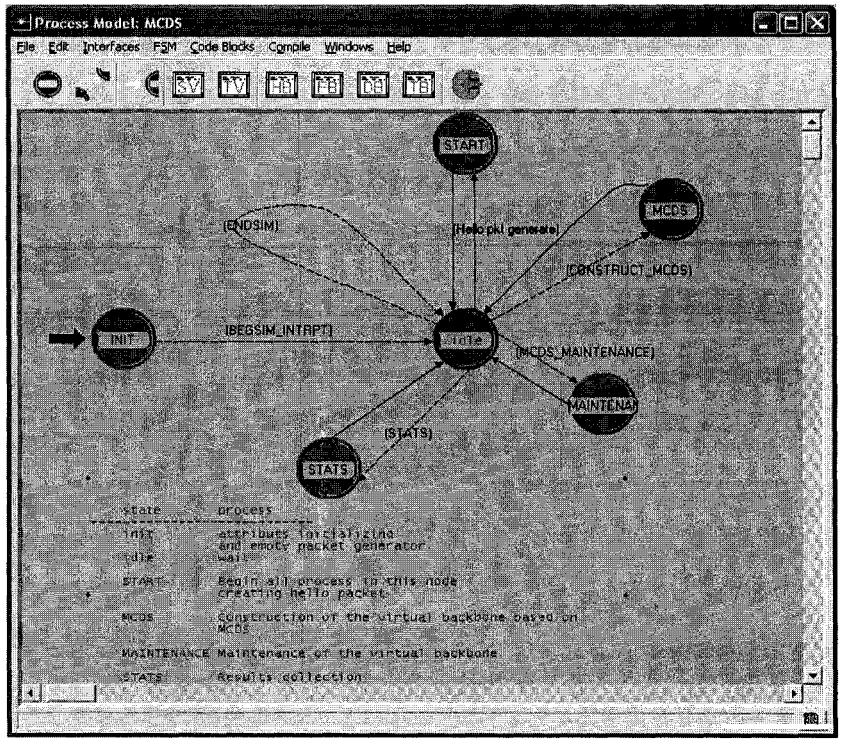


Figure 42 Diagramme d'état pour le processus MCDS

5.4.1 Modèle de mobilité

Pour le modèle de mobilité, nous avons utilisé le modèle le plus fréquent pour modéliser le mouvement des nœuds dans le réseau *ad hoc* soit le modèle *Random WayPoint Model* (RWP). Le modèle RWP a été proposé pour la première fois par Johnson et Maltz dans leur simulation de DSR [Johnson et Maltz (1996)]. Cinq paramètres décrivent l'environnement de simulation pour le modèle RWP original :

- taille et forme de la région de déploiement Q,
- distribution des nœuds dans la région Q,
- un temps de repos (*pause time*) constant pour chaque nœud, et
- vitesse minimale et vitesse maximale $0 \leq v_{min} \leq v_{max}$.

Dans le modèle RWP, chaque nœud choisit une position destination et une vitesse suivant une distribution uniforme et se dirige vers la destination. Quand le nœud arrive à

la position destination, il s'arrête pour une certaine durée de temps (temps de repos). Le temps de repos est choisi aléatoirement suivant une distribution uniforme. Différents travaux de recherche ont été publiés pour montrer les insuffisances de ce modèle [Bettstetter et al. (2003)] et [Yoon et al. (2003)]. En effet, ces travaux montrent que ce modèle ne permet pas d'atteindre un état d'équilibre (*steady state*) en termes de :

- distribution des nœuds,
- vitesse moyenne par nœud.

Bettstetter *et al.* ont étudié le problème de distribution des nœuds comme étant l'effet de bordure et ceci vient du fait qu'un nœud choisira sa prochaine position de destination uniquement dans la zone du réseau prédéfinie et ne doit pas aller en dehors de cette zone. Un nœud proche de la bordure de la zone va choisir une position de destination proche du centre de cette zone. Ils ont démontré que la distribution des positions n'est pas uniforme et que la distance entre le bord de la zone et le centre a un effet sur le résultat. Afin de résoudre ce problème, nous suivrons la recommandation des auteurs [Bettstetter et al. (2003)]. Cette recommandation consiste à introduire un paramètre de stabilité, p_s, avec $0 \leq p_s \leq 1$. Ce paramètre représente la probabilité qu'un nœud reste statique pendant toute la durée de la simulation. Bettstetter *et al.* ont montré qu'avec l'introduction de ce paramètre, le système pourra atteindre son régime permanent plus rapidement.

L'autre problème avec le modèle RWP est que la vitesse moyenne d'un nœud tend vers zéro et non pas vers une valeur moyenne entre la valeur maximale et la valeur minimale. Ceci se produit surtout quand la valeur minimale de la vitesse prend la valeur zéro et que la vitesse maximale est relativement faible. Pour remédier à ce problème,

nous prenons alors :

- $v_{min} \neq 0$,
- une vitesse qui varie uniformément entre v_{min} et v_{max} pour avoir une moyenne autour de $\frac{v_{min} + v_{max}}{2}$
- un temps de repos variable compris entre $[t_{min}, t_{max}]$, avec $t_{min} \neq 0$.

5.4.2 Évaluation de la procédure de maintenance

Comme nous l'avons mentionné dans notre méthodologie, la construction et la maintenance de la dorsale se font avec deux algorithmes différents. De ce fait, nous supposons que la phase d'établissement est déjà réalisée et à l'instant t=0 de la simulation une dorsale est déjà formée en utilisant l'algorithme proposé dans le chapitre 3. En d'autres termes, avant de lancer la simulation et les nœuds commencent à se déplacer un certain nombre de nœuds ont été sélectionnés comme nœuds dominants et les autres nœuds sont alors des dominés.

Nous avons effectué un certain nombre de simulations pour apprécier le comportement et la performance de notre solution de maintenance vis-à-vis différents paramètres tels que la variation du nombre de dominants dans le temps, la variation de pourcentage de connexions à la dorsale dans le temps, l'effet de la mobilité et l'effet du nombre total de nœuds dans le réseau.

Dans un premier temps, nous nous intéressons au comportement de cette procédure pour une durée de vie d'un réseau *ad hoc*. La taille du réseau est N = 50 nœuds. Un nœud se déplace avec une vitesse moyenne de $10m/s$. Le temps moyen de repos (t_{moy}) est égal à 50 secondes. Un nœud possède une portée radio R = $300m$, paramètre par défaut du modèle 802.11b d'*Opnet*. La durée de vie d'un réseau est choisie égale à une heure (3600s). Nous déterminons le nombre de dominants dans tout le réseau durant la période de simulation ; les valeurs sont prises toutes les 10 secondes.

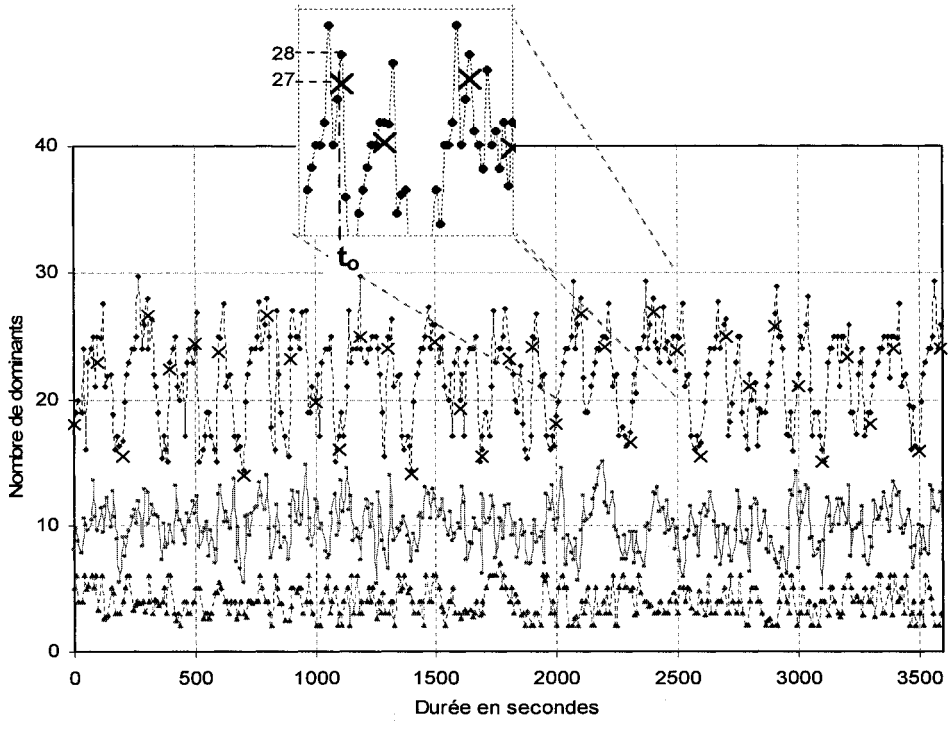


Figure 43 Variation du nombre de dominants en fonction du temps pour k_{cds} = 1, 3 et 5

La figure 43 représente les résultats de simulation pour k_{cds} = 1, 3 et 5. Ces résultats montrent que plus la valeur de k augmente plus la structure est stable dans le sens où le nombre de dominants ne change pas rapidement; absence des pics comme pour k = 1. En ce qui concerne la taille de la dorsale après l'application successive de la procédure de maintenance, nous pourrons vérifier que la solution a tendance de converger vers la solution donnée par la construction. Sur la partie zoomée de la figure 41, nous voyons clairement que la taille de la dorsale est très proche de la taille de l'ensemble MCDS. La taille de l'ensemble MCDS est représentée par des"**x**", et il est calculé en utilisant l'algorithme de construction tout en considérant qu'il s'agit d'un système statique à des instants bien déterminés. Par exemple à l'instant t_o = 2100 s, la taille de la dorsale après la maintenance est égale à 28 et la taille de l'ensemble MCDS est égale à 27. Ces résultats démontrent que la procédure de maintenance s'adapte bien avec la variation de

la topologie du réseau et a tendance à converger vers la solution donnée par l'algorithme de construction de la dorsale. L'écart entre la solution donnée après la maintenance et la solution donnée par la construction est de l'ordre de 6%. Dans le cas où cette solution s'éloigne de la solution optimale une initialisation du réseau est alors nécessaire et le coût engendré par cette action sera le temps nécessaire pour exécuter l'algorithme de construction proposé dans le chapitre 3.

Une deuxième série de simulations consiste à évaluer la robustesse de la procédure de maintenance dans un environnement dynamique. Pour cela, nous allons vérifier la connectivité de la structure formée par la dorsale dans le temps; en d'autres termes, nous allons déterminer (en %) le nombre de nœuds qui restent connectés à la structure globale formée par la dorsale en présence de la procédure de maintenance. Avec les mêmes paramètres que précédemment : $N = 50$, $v_{moy} = 10m/s$, $t_{moy} = 50s$, $R = 300m$ et une durée de simulation de 3600 secondes, soit la durée de vie du réseau. Nous illustrons uniquement le cas pour $k = 1$ étant donné que c'est le cas où la structure est la plus critique, d'après les résultats de simulations précédentes (figure 44). La structure globale du réseau est peu sensible à la mobilité; la variation du pourcentage de connexions dans le temps. En moyenne, les nœuds restent connectés 92,6 % du temps. Cette valeur est plus spécifiquement de 94,2 % et 95,3 % pour $k = 3$ et $k = 5$ respectivement. Les creux, là où le pourcentage chute au-dessous de cette moyenne, correspondent à une variation brusque et importante du nombre de dominants ce qui fait qu'un nombre important de nœuds entrent dans un processus de changement d'état.

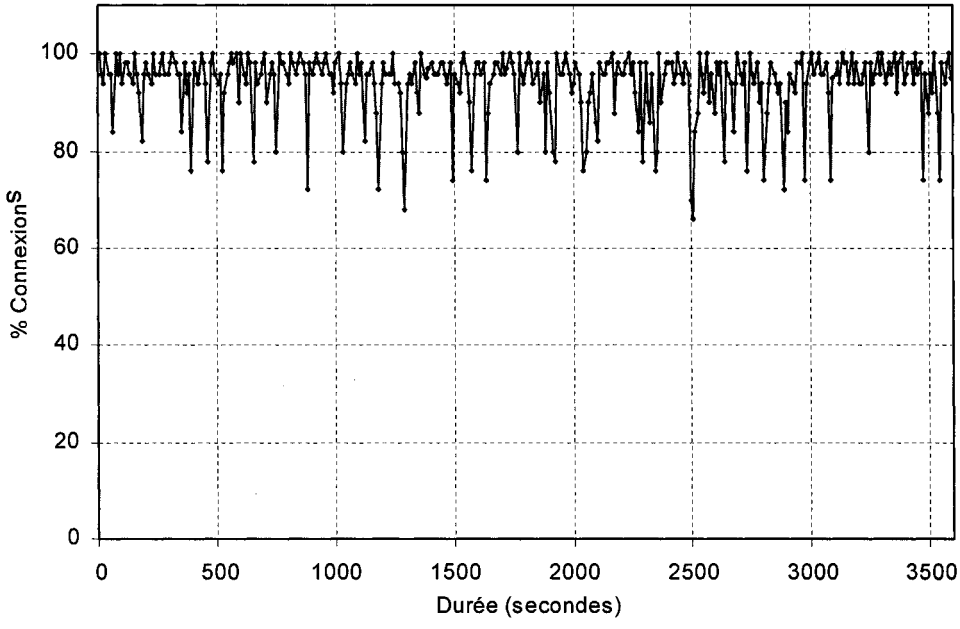


Figure 44 Variation du pourcentage de nœuds connectés en fonction du temps pour k_{cds}=1

Nous avons aussi collecté des valeurs reliées à la performance de la procédure de maintenance telles que :

- la durée moyenne qu'un nœud reste dans l'état dominant,
- la durée moyenne qu'un dominé reste avec le même dominant,
- le temps moyen nécessaire pour qu'un nœud (dans l'état IDLE) se connecte,
- le nombre moyen de dominés par dominant.

Tableau V

Résultats pour différentes valeurs de k_{mcds}

	$k_{mcds} = 1$	$k_{mcds} = 3$	$k_{mcds} = 5$
durée moyenne qu'un nœud reste dans l'état dominant (s)	110 (7.4)♦	118 (8.5)	105 (6.4)
durée moyenne qu'un dominé reste avec le même dominant (s)	64 (4.2)	86 (7.5)	94 (6.7)
temps moyen nécessaire pour qu'un nœud se connecte (*ms*)	185 (12)	254 (18)	312 (27)
nombre moyen de dominés par dominant	2.7 (0.7)	5.2 (1.3)	9.2 (1.8)

♦La valeur entre () représente l'écart type.

D'après les résultats trouvés (tableau V), nous avons pu remarquer que les durées (la durée moyenne qu'un nœud reste dans l'état dominant ainsi que la durée moyenne qu'un dominé reste avec le même dominant) possèdent le même ordre de grandeur pour différents valeurs de k_{cds}. Par contre, ce paramètre k_{cds} a un impact direct sur le temps moyen nécessaire pour qu'un nœud se connecte. Ceci est dû au fait qu'un nœud qui va tenter de se connecter et qui se retrouve à k_{cds} sauts d'un dominant doit attendre que le message *hello* envoyé atteigne le dominant s'il existe. D'autre part, plus la valeur de k_{cds} augmente, plus le nombre de dominés attachés au même dominant augmente. Ceci s'explique par le fait qu'il y a moins de dominants pour le même nombre de terminaux dans le réseau.

5.4.2.1 Effet de la mobilité

Nous voulons évaluer l'influence de la mobilité sur la performance de la procédure de maintenance. Pour ce faire, nous allons faire varier la vitesse de 1 à 30 m/s. Les autres paramètres de simulation sont les mêmes que précédemment. Pour chaque valeur de k_{cds}, nous déterminons la valeur moyenne (en %) des nœuds qui restent connectés à la dorsale ainsi que le nombre de dominants dans la dorsale. La figure 35 illustre les résultats de

simulations pour k_{cds}=1, 3 et 5. Ces résultats montrent que la proportion moyenne de nœuds connectés à un dominant, à un temps donné, diminue légèrement avec la mobilité. Mais elle reste au delà de 90 %, ce qui représente un taux de connectivité élevé même quand la vitesse augmente. La variation du pourcentage de connexion est pratiquement identique quand k_{cds} varie.

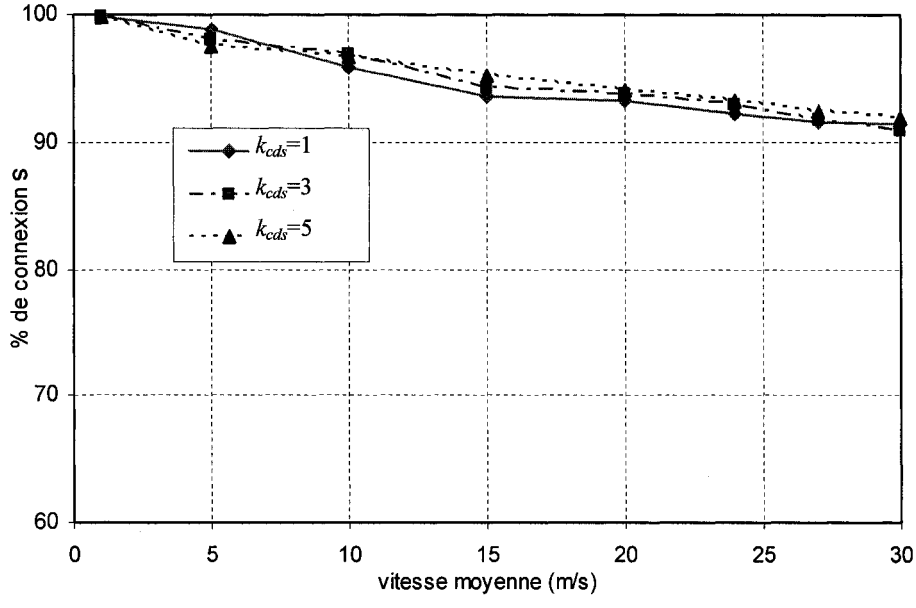


Figure 45 Pourcentage de connexions en fonction de la vitesse moyenne

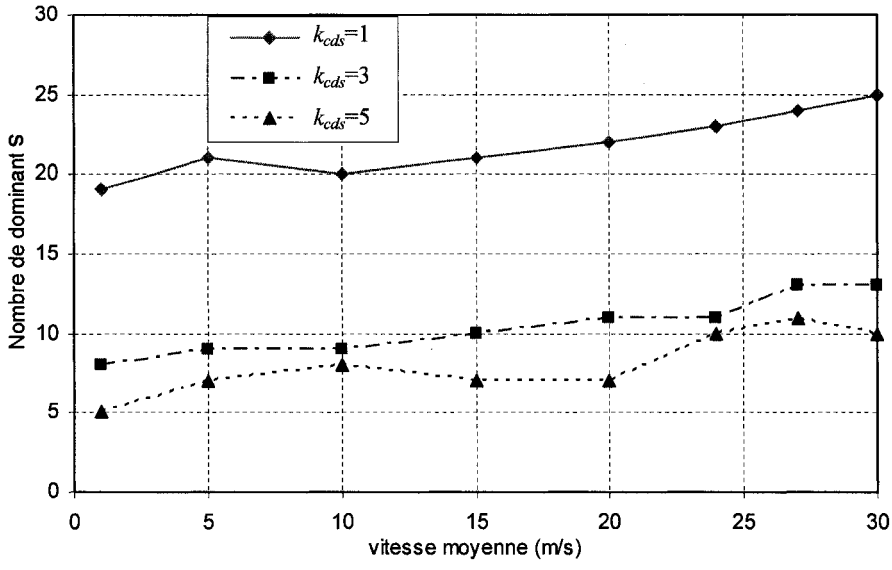


Figure 46 Variation du nombre de dominants en fonction de la vitesse moyenne

Les résultats de la figure 46 montrent la variation de la valeur moyenne du nombre de dominants quand la mobilité varie. Le nombre de dominants a tendance à croître légèrement quand la mobilité augmente. Ce nombre ne dépasse pas la moitié du nombre total des nœuds pour une vitesse moyenne de 30 *m*/s et $k_{cds} = 1$. Ces résultats témoignent bien de l'efficacité de la procédure de maintenance conçue dans ce chapitre. Elle est capable de bien s'adapter même quand la mobilité augmente.

5.4.2.2 Effet de la taille du réseau

Le problème de mise à l'échelle (*scalability*) est largement considéré dans les réseaux *ad hoc* et représente un critère primordial pour évaluer la performance des mécanismes de routage. Nous examinerons l'effet du nombre de nœuds dans le réseau sur l'efficacité de notre procédure de maintenance. Nous prenons alors des réseaux dont le nombre de nœuds varie de 20 à 80, la vitesse moyenne est de 10m/s et les autres paramètres de simulation sont les mêmes que précédemment.

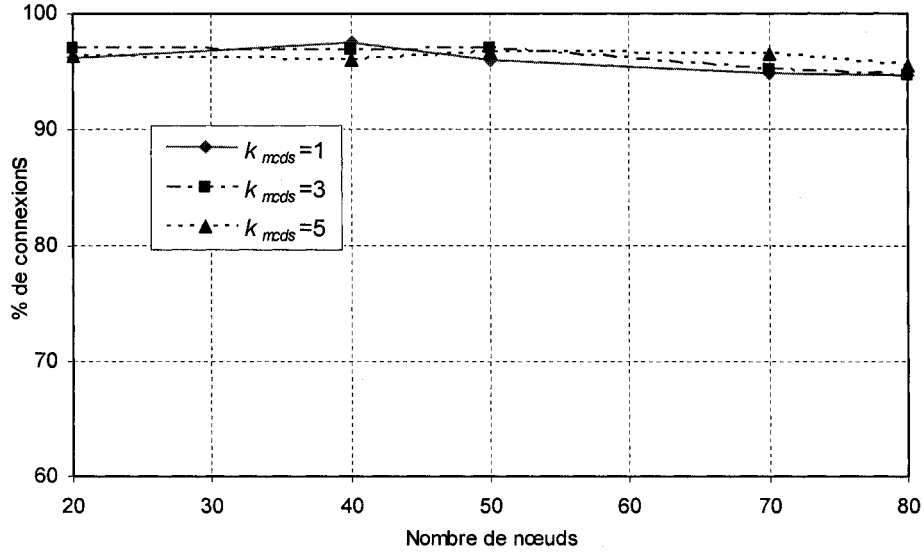


Figure 47 Le pourcentage de connexion en fonction de la taille du réseau

D'après la figure 47, nous observons que le pourcentage de nœuds connectés est peu sensible à la taille du réseau, bien que la taille de la dorsale soit proportionnelle à la taille du réseau (chapitre 4). Le nombre de nœuds dans le réseau n'a pas d'effet sur l'efficacité de notre procédure de maintenance étant donné quelle est distribuée. Notre procédure de maintenance permet le passage à l'échelle.

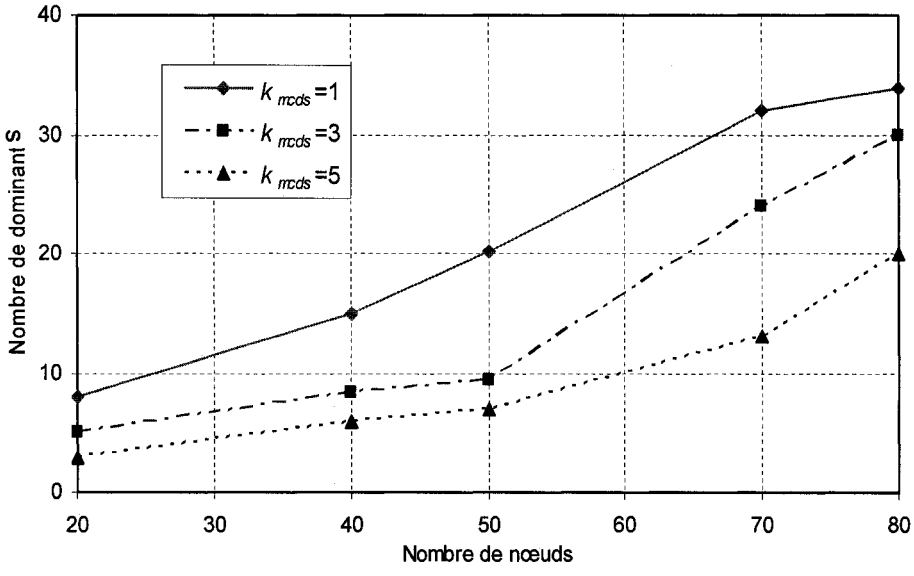


Figure 48 Nombre de dominants en fonction de la taille du réseau

D'après les courbes de la figure 48, nous remarquons que la taille de la dorsale, représentée par le nombre de dominants, croît proportionnellement avec la taille du réseau ce qui est tout à fait raisonnable. L'allure de cette croissance est presque identique à celle retrouvée dans le Chapitre 4, où nous n'avons ni mobilité ni maintenance.

5.4.2.3 Occurrence à l'état dominant

Dans cette section, nous voulons vérifier l'occurrence pour chaque nœud dans l'état dominant. Pour cela, nous allons déterminer le nombre de fois qu'un nœud passe à l'état dominant durant toute la simulation. Les paramètres de simulation sont les suivants : le réseau comprend 50 nœuds, k_{cds} =1, la vitesse moyenne pour chaque nœud est égale à $10 m/s$ et la durée de simulation est d'une heure.

La figure 49 illustre le nombre de fois qu'un nœud devient dominant au cours de la simulation. Plus de 70 % des nœuds passent à l'état dominant entre 20 et 28 fois.. Ces résultats reflètent en quelque sorte qu'il y a un équilibre de la charge dans le réseau en

termes de consommation d'énergie car les nœuds dominants auront à consommer plus d'énergie. Rappelons que nous sommes partis avec l'hypothèse que tous les nœuds sont homogènes et qu'au départ ils possèdent le même niveau d'énergie. De plus, seulement les nœuds qui forment la dorsale (dominants) sont chargés de retransmettre le trafic de contrôle. Une grande partie de la bande passante dans un réseau *ad hoc* est utilisée par le trafic de contrôle [Corson et Macker (1999)].

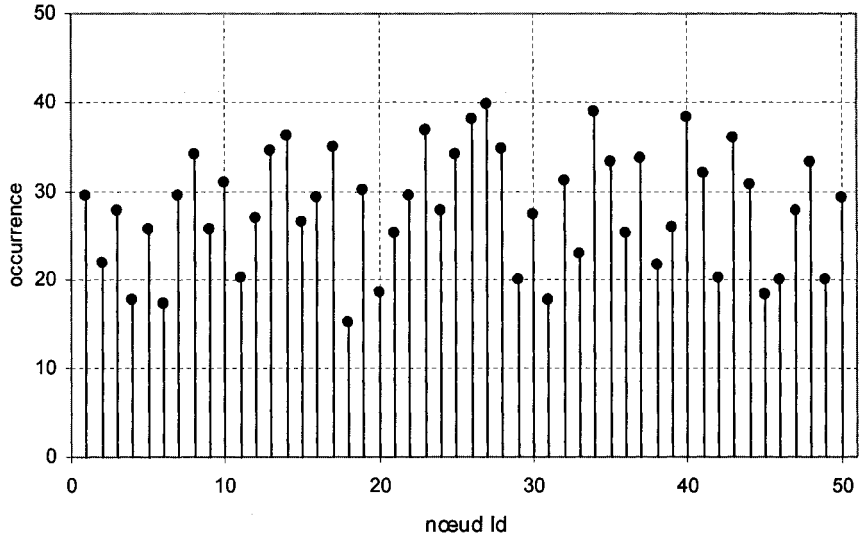


Figure 49 Nombre de fois qu'un nœud devient dominant

5.5 Comparaison avec d'autres algorithmes

Dans cette section, nous allons évaluer et comparer la performance de la procédure de maintenance proposée dans ce chapitre avec d'autres algorithmes proposés dans la littérature. Comme nous avons mentionné dans l'introduction, toutes les propositions utilisent un seul algorithme pour la construction et la maintenance. Nous allons reprendre les algorithmes décrits dans la section 2.5.3 (WCDS, CDS-based et B-CDS). Par la suite nous noterons MW, MC et MB pour désigner les algorithmes WCDS, CDS-based et B-CDS respectivement. La procédure proposée dans ce chapitre sera identifiée par M1. Pour mieux comparer notre procédure avec ces algorithmes en ce qui concerne la maintenance, nous ne considérons pas la phase d'établissement du réseau. En d'autres

termes, pour chaque simulation les statistiques ne seront prises qu'après la construction de la dorsale.

Dans le but de faire cette comparaison nous allons définir les deux critères d'évaluation suivants :

- **Taille de la dorsale** : c'est le nombre moyen des nœuds dominants. Nous examinons la variation de ce paramètre en fonction de la taille du réseau et la mobilité des nœuds. Rappelons que nous avons intérêt à garder une taille réduite: plus la taille de la dorsale est minimale meilleur est le résultat,
- **Durée de la maintenance :** c'est le temps moyen nécessaire pour effectuer la maintenance. Pour la procédure de maintenance proposée, ce paramètre est définit comme étant le délai moyen entre le moment où le nœud change de position (état IDLE) et le moment où le dernier nœud concerné par ce mouvement change son état à dominant ou dominé. Pour les autres algorithmes, c'est le temps moyen nécessaire pour qu'un nœud se connecte à la dorsale. Nous évaluons ce paramètre en fonction de la mobilité des nœuds,

5.5.1 Taille de la dorsale

La figure 50 illustre la variation de la taille de la dorsale (en %) en fonction de la taille du réseau. Dans cette simulation, la vitesse moyenne est égale à 15 *m/s*. Les résultats de simulations montrent bien que la procédure de maintenance proposée (M1) ainsi que les trois algorithmes (MC, MB et MW) possèdent un comportement similaire. En d'autres termes, la taille de la dorsale varie proportionnellement avec la taille du réseau. En effet elle est de l'ordre de 47%, 44%, 43% et 40% pour MC, MB, MW et M1 respectivement. Ces résultats montrent la nature distribuée des ces algorithmes. La maintenance se fait localement là où le changement de la topologie s'est produit.

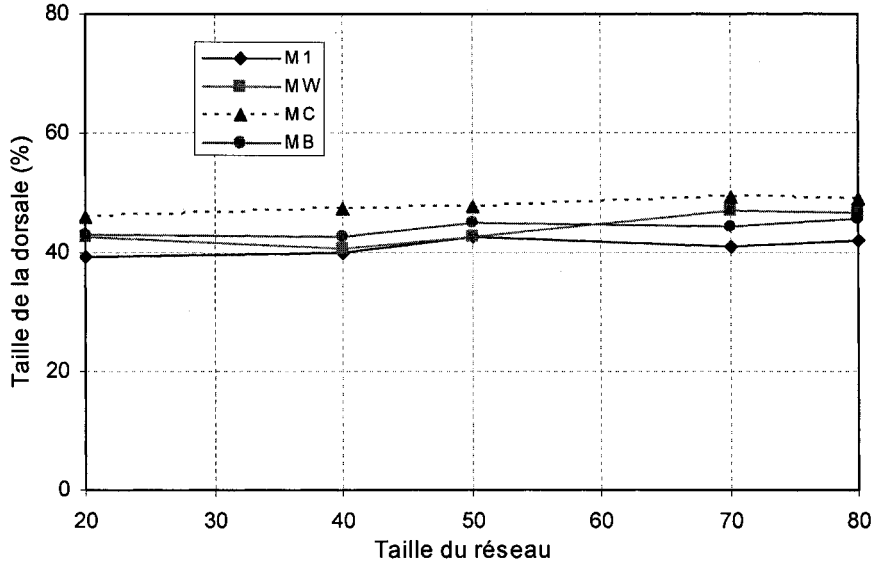


Figure 50 Taille de la dorsale en fonction de la taille du réseau

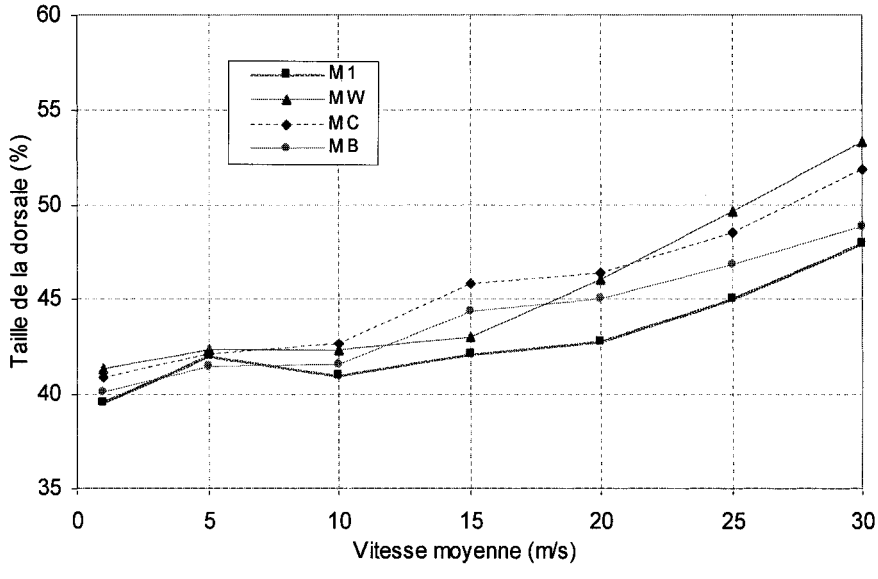


Figure 51 Taille de la dorsale en fonction de la vitesse moyenne des nœuds

La figure 51 illustre la variation de la taille de la dorsale (en %) en fonction de la vitesse moyenne des nœuds. Dans cette simulation, le nombre de nœuds dans le réseau est égal à 50 nœuds. La taille de la dorsale croît quand la mobilité des nœuds augmente. Lorsque

la dynamique augmente, le nombre de nœuds concernés par le changement de la topologie augmente également et par la suite la taille de la dorsale augmente. Pour une mobilité élevée, la procédure de maintenance M1 présente de meilleurs résultats. En effet, pour une vitesse moyenne de $30m/s$, la dorsale possède une taille nettement inférieure avec la procédure M1 comparée aux autres algorithmes.

5.5.2 Durée de la maintenance

D'après les résultats de simulation, figure 52, le temps nécessaire pour effectuer la maintenance croit proportionnellement quand la vitesse moyenne des nœuds augmente. Ces résultats confirment ceux retrouvés dans la section précédente (figure 51). Plus la vitesse des nœuds augmente, plus le nombre de nœuds qui sont concernés, ayant subis un changement d'état, augmente ce qui augmente également le temps de maintenance augmente. L'algorithme M1 réalise la maintenance plus rapidement que les algorithmes MW, MC et MB.

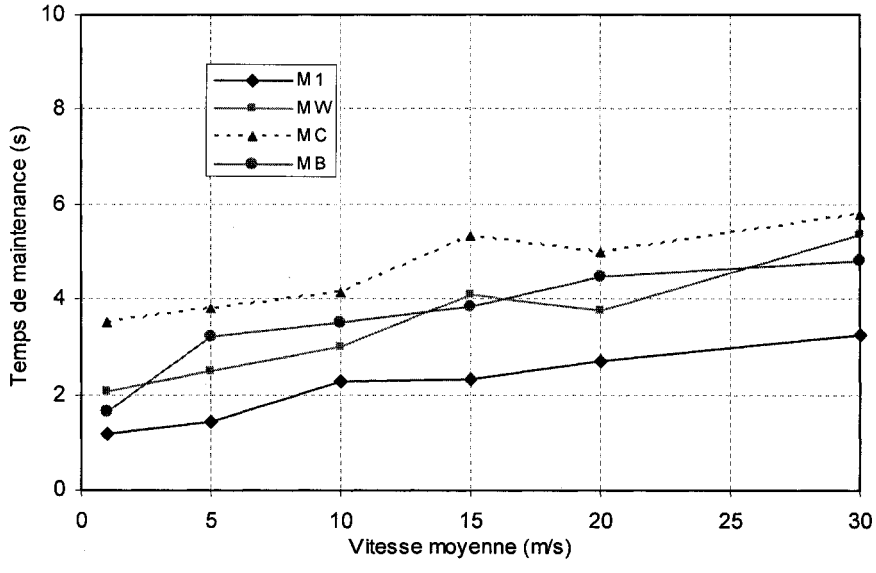


Figure 52 Temps de maintenance en fonction de la vitesse moyenne des nœuds

5.6 Conclusion

Dans ce chapitre, nous avons présenté une nouvelle procédure de maintenance pour la dorsale virtuelle dans un réseau *ad hoc* mobile. Ensuite, nous avons procédé à l'évaluation de sa performance en fonction de la mobilité et de la taille du réseau. Cette procédure est distribuée et elle ne sera appliquée que s'il y a un changement dans la topologie. En d'autres termes, quand un terminal change de position il va tenter de se connecter au nœud appartenant à la dorsale le plus proche en appliquant la procédure de maintenance. Rappelons qu'une dorsale est construite dans la phase d'établissement en utilisant l'algorithme développé dans le chapitre 4. Cet algorithme garantit une dorsale de taille petite et par la suite la maintenance sera plus efficace.

Nous avons traité le cas général où une dorsale est construite à partir de k_{cds}. Dans ce cas, les résultats de simulation nous indiquent qu'il n'y a aucun avantage à utiliser un k_{cds} élevé. En effet, nous avons constaté que, quand k augmente, le temps nécessaire pour qu'un terminal se connecte à la dorsale augmente aussi. Toutefois, les performances en termes de pourcentage des nœuds connectés en fonction de la mobilité et de la taille du réseau sont invariables avec k_{cds}.

De nature distribuée, la procédure de maintenance proposée a démontré son efficacité non seulement au niveau de la variation de pourcentage de connexion en tenant compte de la mobilité, mais également au niveau de la mise à l'échelle (la variation du nombre des nœuds dans le réseau). En effet, cette procédure s'adapte bien avec la dynamique de la topologie au cours du temps et le pourcentage de connexions décroît légèrement quand la vitesse moyenne des terminaux augmente (92 % des terminaux restent connectés pour une vitesse moyenne de $30 m/s$), alors que ce pourcentage est quasiment constant quand le nombre de nœuds varie dans le réseau. Ces résultats montrent que la procédure de maintenance proposée assure une certaine stabilité de la structure globale du réseau quand la mobilité des nœuds augmente. De plus, nous avons vérifié qu'à long terme les nœuds auront quasiment la même chance de passer en état dominant ce qui

donne un certain équilibre de la charge et de l'énergie dissipé dans le réseau étant donné qu'un nœud dominant aura à relayer plus de trafic (contrôle et donné) qu'un nœud dominé.

La dernière partie de ce chapitre a été consacrée pour comparer la procédure de maintenance proposée dans ce chapitre avec d'autres algorithmes proposés dans la littérature. Les résultats de simulation montrent bien que notre procédure de maintenance proposée a une meilleure performance par rapport aux autres algorithmes. En effet, en utilisant notre procédure, nous obtenons toujours une dorsale de taille réduite dans un environnement dynamique tout en réduisant le temps de maintenance.

CHAPITRE 6

LES PROTOCOLES DE ROUTAGE EN PRÉSENCE D'UNE DORSALE

6.1 Introduction

Les réseaux *ad hoc* que nous considérons sont multi saut. Le trafic de donnée envoyé par la source vers une destination doit être relayé par les nœuds intermédiaires. Dans un réseau *ad hoc* tous les nœuds doivent collaborer pour acheminer le trafic. Les paquets de données vont alors être transmis de proche en proche jusqu'à la destination. Même si aucune technologie sans fil n'a été conçue spécifiquement pour les réseaux *ad hoc*, dans le sens où elle est capable d'effectuer le relayage du trafic entre les nœuds, il est possible d'utiliser les technologies sans fil actuelles en leur ajoutant les protocoles adéquats pour faire fonctionner un réseau *ad hoc*. Par exemple, on peut ajouter le standard 802.11, qui ne permet que la communication entre mobiles dans la portée de communication, un protocole de routage adéquat qui rendra la communication possible entre les mobiles dans le réseau.

L'absence d'infrastructure et la dynamique des terminaux dans un réseau *ad hoc* rendent la tâche moins facile pour un protocole de routage afin de trouver un chemin entre source et destination. Les protocoles de routage élaborés et utilisés dans les réseaux filaires ne sont plus compatibles et ne peuvent pas être utilisés par la suite dans les réseaux *ad hoc*. De nombreux protocoles de routage ont donc été proposés pour réaliser les communications multi saut et la rendre plus efficace (moins de retransmissions, chemins plus courts, etc.).

Dans le chapitre 2, nous avons présenté, quelques protocoles de routage développés dans le cadre du groupe de travail MANET de l'IETF. Ces protocoles sont définis au niveau IP et sont donc indépendants des couches physiques et MAC. Le routage IP permet en particulier une interconnectivité aisée avec toutes sortes d'autres réseaux ou de matériel. Les protocoles présentés sont parmi les représentatifs des diverses techniques utilisées pour le routage *ad hoc*. La première partie de ce chapitre est consacrée à la comparaison de quatre protocoles de routage; un protocole proactif (OLSR) et trois protocoles réactifs (AODV, DSR et TORA) sans l'utilisation d'une dorsale. Nous voudrons étudier leurs comportements face à la mobilité. Pour réaliser cette comparaison, nous avons défini cinq métriques : le taux de paquets délivrés, le nombre moyen de sauts, le nombre de paquets transmis par le nombre de paquets de données délivrés, le nombre d'octets utilisés pour le trafic de contrôle et le délai de bout en bout. Dans la deuxième partie, nous présenterons les changements apportés aux protocoles AODV, DSR et TORA pour qu'ils puissent être opérationnels avec une dorsale virtuelle. Nous évaluerons ainsi l'amélioration de leurs performances en présence de la dorsale.

6.2 Protocoles de routage sans dorsale

Dans cette section, nous allons évaluer l'effet de la mobilité sur certains protocoles et comparer leur performance. Nous nous sommes limités dans cette comparaison aux protocoles implémentés dans OPNET (DSR, AODV, TORA et OLSR). Pour le modèle de mobilité, nous avons utilisé le modèle *Random Waypoint Model* (RWP) modifié. Comme nous l'avons mentionné dans le chapitre précédent, le modèle original présente des faiblesses. Nous avons alors tenu compte des recommandations données par [Bettstetter et al. (2003)].

Nous considérons un réseau d'une taille de 50 nœuds, uniformément placés dans une zone carrée (1,5 *km* x 1,5 *km*). Tous les nœuds ont une portée radio de 300*m*. Une quarantaine de sessions (sources, destinations) sont considérées pour effectuer une moyenne empirique. Le trafic est différent pour chaque paire (Source, Destination). Un

taux de transmission de paquets constant (CBR) est choisi avec les paramètres suivants : chaque source transmet les paquets de données à un taux variant de 1 à 8 paquets/*s*. La taille des paquets de données est fixe et égale à 512 octets. Le choix des paquets de petite taille est une façon de pénaliser le protocole [Johansson *et al.* (1999)]. Le temps d'arrivée des paquets est une variable aléatoire suivant une distribution exponentielle en moyenne de 300 *ms*. Les résultats de simulation sont obtenus en utilisant une couche MAC et physique IEEE 802.11b, qui est le standard implémenté dans le simulateur OPNET.

Nous avons également défini un certain nombre de métriques utilisées dans le but de comparer le comportement de chacun des protocoles. Quelques-unes de ces métriques ont été proposées par le groupe de travail MANET de l'IETF dans l'objectif d'évaluer les protocoles de routage :

- **Taux de paquets délivrés** : il est défini comme étant le rapport (en pourcentage) du nombre total des paquets de données délivrés aux destinations par le nombre total de paquets générés par les sources. En d'autres termes, vis-à-vis de la stabilité des chemins, cette métrique reflète en quelque sorte l'efficacité du protocole de routage.
- **Nombre de sauts** : c'est le nombre moyen de sauts que les paquets de données effectueront pour atteindre la destination. Un nombre réduit de saut indique l'efficacité de la procédure de sélection de route. Pour deux protocoles ayant un taux de livraison comparable, ils seront alors comparés par le nombre moyen de sauts. Cependant, si les protocoles ont des taux de livraison différents (spécifiquement dans les réseaux avec taux de mobilité élevé où les liens changent fréquemment), le nombre de sauts est relié au taux de livraison des paquets, c'est-à-dire, plus le taux de livraison augmente plus le nombre de sauts augmente. Puisque, seuls les paquets de données qui arrivent à destination sont considérés, un nombre de saut minimum implique que tous les paquets de données délivrés sont destinés aux nœuds à proximité et les paquets envoyés aux

nœuds éloignés seront abandonnés. Ainsi, la mesure du nombre de sauts fournit de l'information sur la capacité de survie du protocole.

- **Nombre de paquet de données transmis par rapport au nombre de paquets de données délivrés** : le nombre de paquets transmis, du point de vue réseau, prend en considération chaque retransmission de paquet de données pour chaque nœud. Il inclut la transmission des paquets qui ont été éventuellement abandonnés et retransmis par les nœuds intermédiaires. Plus ce rapport augmente plus le nombre de paquets de données redondants dans le réseau augmente. Alors, il est préférable d'avoir un rapport qui est de l'ordre du nombre de sauts.
- **Nombre d'octets utilisés pour le trafic de contrôle** : ce nombre inclus l'*overhead* dans les paquets de données (la séquence de route pour DSR par exemple) ainsi que tous les paquets de contrôle. Le nombre total d'octets envoyés durant la simulation sera comptabilisé. Seul l'*overhead* de la couche réseau (niveau IP) est pris en considération. Ce nombre peut être considéré comme étant l'efficacité du protocole de routage en terme d'utilisation de la bande passante [Corson et Macker (1999)].
- **Délai de bout en bout** : c'est le délai moyen, mesuré en milliseconde (*ms*), entre la génération d'un paquet de données par la source et la réception par la destination. Il inclut le temps d'acquisition d'une route, le temps de traitement, le délai dans la file d'attente à chaque nœud intermédiaire entre source et destination, et le délai de propagation. Le délai de bout en bout est comptabilisé uniquement pour les paquets qui se rendent à la destination.

6.2.1 Taux de paquets délivrés

La figure 53 illustre le taux de paquets délivrés pour les quatre protocoles. Tous ces protocoles performent mieux quand la mobilité est faible. À une vitesse moyenne nulle le taux exède 95 %. Quand la mobilité croît, le taux de livraison des paquets décroît pour tous les protocoles. La mobilité des nœuds engendre directement un changement dans la

topologie du réseau et par la suite une recherche d'une nouvelle route est nécessaire. Le protocole performe alors suivant le mécanisme de maintenance de route utilisé.

En général, les protocoles réactifs (à la demande) DSR et TORA, à l'exception d'AODV, possèdent un taux de livraison de paquets assez important. Ils performent mieux comparés au protocole proactif (OLSR). TORA performe mieux que DSR. Pour DSR et dans la situation où la mobilité est élevée, les nœuds intermédiaires n'ont pas de mécanisme de récupération pour déterminer des routes alternatives. Quand une route se brise, un message d'erreur est envoyé jusqu'à la source, la source recommence alors une nouvelle recherche pour une autre route, des paquets sont alors perdus. Ainsi, nous trouverons que le délai pour découvrir une route joue un rôle important dans la dégradation des performances quand la mobilité est importante.

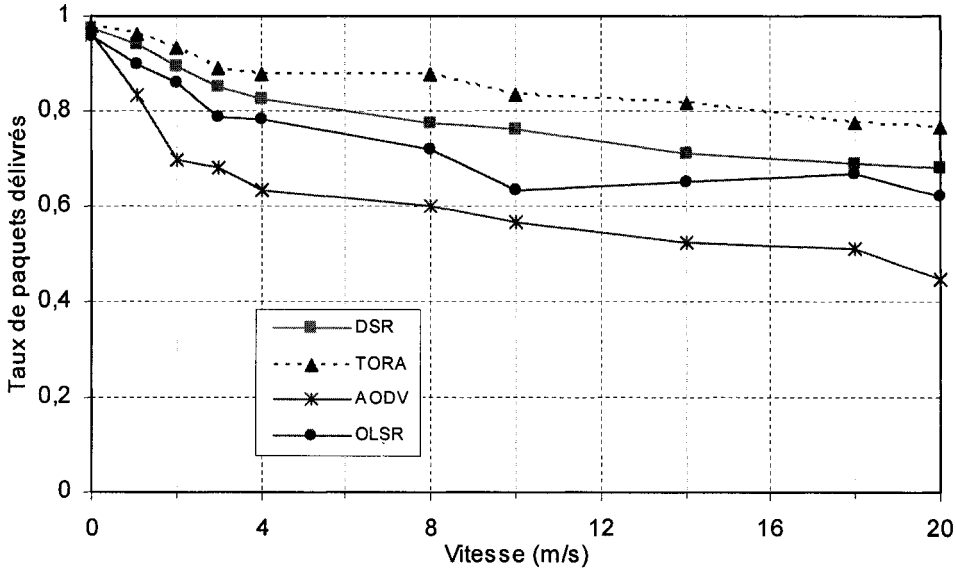


Figure 53 Taux de paquets délivrés en fonction de la mobilité

Le protocole AODV éprouve de la difficulté quand les nœuds se déplacent à une vitesse importante. Le routage basé sur la source (utilisé par DSR) révèle plus d'informations

dans la découverte d'une route que dans l'AODV. Par conséquent, pour une même période de temps, il y aura plus de routes qui sont découvertes et par la suite plus de paquets seront délivrés.

Le protocole OLSR est sensible à la mobilité, les messages de mise à jour sont échangés périodiquement et non à la suite d'un changement de situation. Les routes deviennent moins précises quand la mobilité augmente. Certaines informations sur l'état des liens maintenus dans les tables de routage sont imprécises. L'ajustement de la période de mise à jour pourra résoudre ce problème.

6.2.2 Nombre de sauts

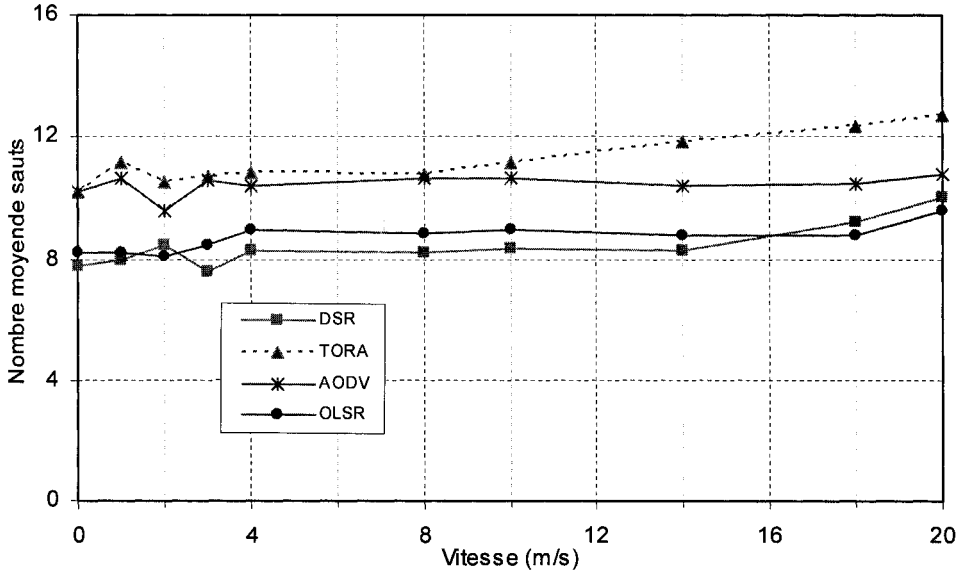


Figure 54 Nombre moyen de sauts en fonction de la mobilité

Comme nous l'avons mentionné, le nombre moyen de sauts ne concerne que les paquets délivrés avec succès jusqu'à destination. À l'exception du protocole TORA qui ne minimise pas le nombre de sauts pour choisir le meilleur chemin, la figure 54 révèle que les protocoles qui délivrent plus de paquets de données (comme indiqué à la figure 53) possèdent un nombre de sauts moins important. Ceci s'explique par le fait que si la

distance entre source et destination est importante, le nombre de nœuds intermédiaires que les paquets de données vont traverser augmente et par la suite la probabilité qu'un paquet soit abandonné devient plus importante. Le nombre de sauts dépend principalement de la manière dont le protocole de routage a implémenté son mécanisme de maintenance d'une route pour trouver une route alternative.

6.2.3 Nombre de paquets transmis par le nombre de paquets de données délivrés

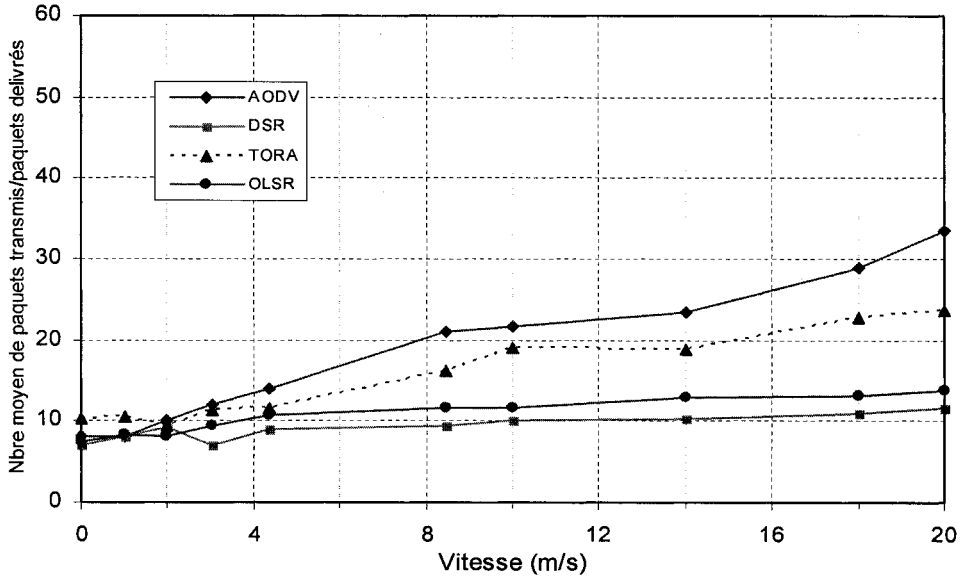


Figure 55 Nombre moyen de paquets transmis par paquets délivrés en fonction de la mobilité

Parmi les protocoles réactifs, DSR performe mieux en ce qui a trait au nombre de paquets de données transmis par le nombre de paquets de données délivrés (figure 55). En effet, DSR possède un dispositif de récupération (*salvation*) efficace où un nœud qui détecte une route brisée récupère les données et les envoie via une autre route vers la destination, une route connue et enregistrée dans la cache. Ainsi, les paquets sont abandonnés moins fréquemment comparé aux autres protocoles.

Le protocole OLSR présente des résultats comparables à ceux de DSR. La sélection des relais multipoint et l'échange périodique des tables de routage permettent de minimiser la retransmission des paquets de données dans le réseau. Pour TORA et AODV, le nombre de retransmissions croît en fonction de la mobilité. Le protocole AODV possède les pires résultats ce qui explique, en partie, la dégradation de son taux de paquets délivrés (figure 53).

6.2.4 Nombre d'octets utilisés pour le trafic de contrôle

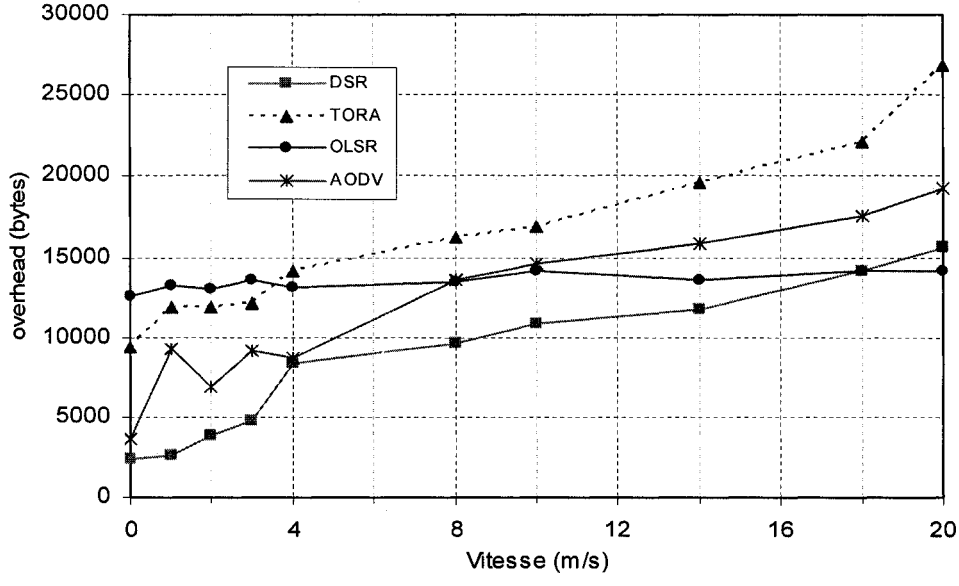


Figure 56 Nombre d'octets utilisés pour le trafic de contrôle en fonction de la mobilité

D'après les courbes de la figure 56, nous pourrons remarquer que, pour une mobilité réduite, les protocoles de routage réactifs (DSR, AODV et TORA) utilisent moins de trafic de contrôle comparé au protocole proactif (OLSR). Le trafic de contrôle utilisé par le protocole OLSR est quasiment constant et dépend peu de la mobilité; la mise à jour dans OLSR se fait de façon périodique et non pas à la suite d'un déplacement des nœuds. Pour un niveau de mobilité élevé, les protocoles de routage réactifs possèdent un trafic de contrôle important. Ceci est dû principalement aux procédures répétitives pour

déterminer une nouvelle route utilisées par les protocoles de routage. Le protocole DSR performe mieux que les autres protocoles réactifs. En effet, dès qu'il détecte qu'une route n'est plus valide. Dans DSR *l'overhead* provient principalement de la séquence des nœuds intermédiaires inscrite dans le paquet pour atteindre la destination.

6.2.5 Délai moyen de bout en bout

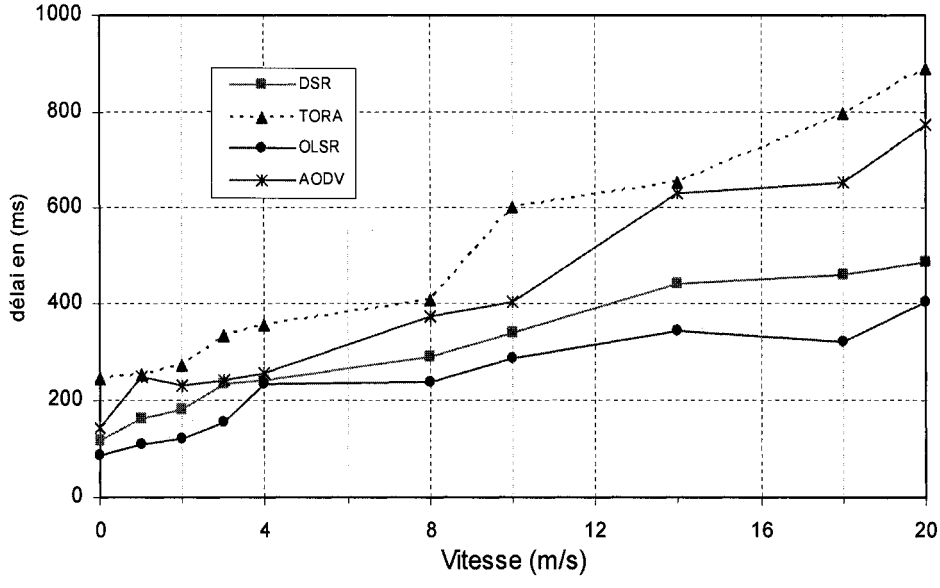


Figure 57 Délai moyen en fonction de la mobilité

La figure 57 illustre le délai moyen de bout en bout entre source et destination. Comme prévu, le protocole OLSR possède un délai plus faible comparé aux autres protocoles réactifs OLSR est un protocole proactif : quand un paquet arrive à un nœud il sera retransmis immédiatement ou bien abandonné. Pour les protocoles réactifs, il n'existe pas de route vers une destination donnée, les paquets seront sauvegardé dans un tampon jusqu'à ce qu'une nouvelle route soit déterminée. De plus, nous remarquons que pour le protocole OLSR le délai ne varie pas beaucoup avec la mobilité.

Grâce à la manière dont le protocole DSR détecte la non-validité d'une route, il possède le plus faible délai parmi les protocoles réactifs. Avec DSR, la procédure d'acquisition

d'une route dans DSR permet de déterminer plus qu'une route et de la stocker dans la cache, alors que dans AODV, une seule route est déterminée par RREQ. Les protocoles TORA et AODV possèdent les pires résultats en termes de délai. Pour le protocole TORA, le délai est dû principalement par la perte des informations sur la distance au fur et à mesure qu'il a progression dans la route. Tandis que pour le protocole AODV, le délai est dû au fait que ce protocole maintient les routes dans un état mou (*soft state*), et elles expireront après un *time-out*. Ainsi, une nouvelle procédure de découverte de route est initiée. Dans un environnement où la mobilité est importante, les routes sélectionnées par les messages *RREQ* peuvent ne plus être valables quand la source commence l'envoi des paquets de données ou même quand les messages *RREP* sont retournés. Ainsi, le délai engendré par la procédure de découverte d'une route augmente.

6.2.6 Interprétation des résultats de simulation

D'après les résultats de simulation d'un protocole de routage réactif (OLSR) et trois protocoles de routages réactifs (AODV, DSR et TORA) nous pouvons déduire les remarques suivantes :

- La performance de ces protocoles se dégrade en présence de la mobilité. Quand la mobilité augmente, le taux de paquets délivrés diminue, le trafic de contrôle augmente, le délai augmente, etc. Toutefois, le protocole OLSR performe mieux que les autres dans un environnement dynamique en termes de trafic de contrôle et de délai. Le trafic du contrôle est presque constant pour OLSR alors qu'il augmente rapidement pour les autres protocoles. Pour OLSR, l'échange du trafic de contrôle se fait de façon périodique et ne dépend que de la taille de l'ensemble MPR qui dépend de la taille du réseau.

- Le délai croît avec la mobilité pour tous les protocoles. Cependant, pour OLSR cette croissance est moins rapide. La dynamique du réseau engendrée par le mouvement des nœuds fait en sorte que l'information dans la cache n'est pas valide. OLSR qui effectue un échange périodique aura une information plus

fraîche. La période de mise à jour peut améliorer encore le délai pour OLSR. Toutefois, le trafic de contrôle sera plus important.

- La mobilité des nœuds a un effet direct sur le taux de paquets délivrés à la destination. En effet, il décroît rapidement dès que le niveau de mobilité augmente (moins de 90% pour une vitesse moyenne de 2 *m/s*). Cependant, les protocoles réactifs TORA et DSR résistent mieux face à la mobilité et délivrent plus de paquets de données comparé à OLSR et AODV.

6.3 Protocoles de routage avec dorsale

Nous proposons dans cette section de vérifier au moyen des simulations comment l'utilisation une dorsale améliore les performances de certains protocoles. Nous limitons notre étude à trois protocoles c'est-à-dire DSR, AODV et TORA pour les raisons suivantes :

- ces trois protocoles sont déjà implémentés dans OPNET (version 11),
- le protocole OLSR utilise le principe de relais multipoint MPR.

Des travaux de recherche, menés par [Adjih *et al.* (2005)] et [Wu et Lou (2004)], ont été approfondis pour approximer l'ensemble de domination connexe à partir de l'ensemble des relais multipoints. De plus, les résultats de simulation du chapitre 4 illustrent que la diffusion qui utilise les relais multipoint et la dorsale est comparable dans certaines mesures. Nous résumons les principaux changements apportés aux protocoles de routage par rapport à leur version originale.

6.3.1 *DSR over Backbone* (DSRoB)

Comme dans la description du protocole DSR, DSRoB utilise les messages RREQ, RREP et RERR dans la découverte d'une route. Toutefois, la diffusion ne se fait que

dans les nœuds qui forment la dorsale jusqu'au nœud dominant voisin de la destination. En d'autres termes, les nœuds ordinaires n'auront pas à retransmettre ces messages même s'ils les reçoivent. Dans l'exemple de la figure 58, la diffusion du message RREQ se fait uniquement par les nœuds de la dorsale, la dorsale étant formée par l'ensemble des nœuds {1, 3, 5 et 8}. Rappelons qu'avec la dorsale, chaque nœud qui n'en est pas membre y possède au moins un voisin. La source, nœud 1, qui désire envoyer du trafic à la destination, nœud 9, lance la recherche d'une route par l'envoi du message RREQ au nœud de la dorsale 3. Le message RREQ va alors se propager dans la dorsale uniquement jusqu'à destination. Les autres nœuds {2, 4, 6 et 7} ne diffusent pas le message même s'ils le reçoivent.

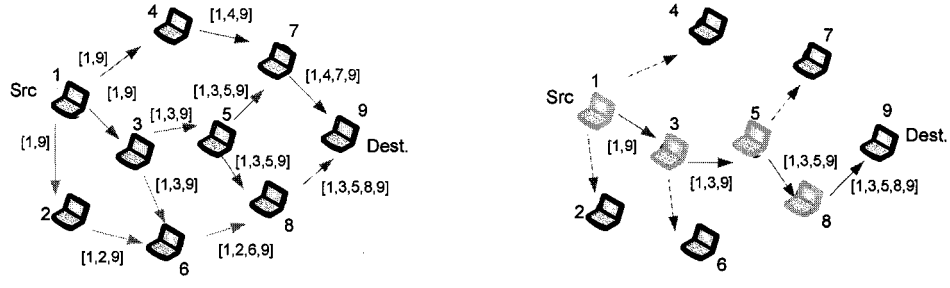


Figure 58 Propagation du message RREQ dans DSRoB

L'utilisation d'une dorsale avec le protocole DSR, permet de :

- minimiser les messages RREQ diffusés dans le réseau et par la suite minimiser les messages RREP dans le chemin de retour. Un seul chemin de retour sera envoyé, non pas comme dans le DSR classique où chaque nœud qui peut joindre la destination peut retourner au moins avec un chemin réalisable,
- éliminer la procédure *Ring Zero Search*. En effet avec les informations que possède chaque nœud de la dorsale cette procédure ne sera pas utile. La source va toujours envoyer sa demande au nœud dominant le plus proche et c'est le rôle de ce dernier de vérifier si la destination est voisine. En effet, le nœud de la dorsale possède l'information sur tous les nœuds qu'il peut desservir. Rappelons

que cette procédure consiste à envoyer le message RREQ à un saut afin de vérifier si la destination se retrouve au voisinage,

- améliorer la procédure de récupération de données (*salvation mechanism*) utilisée par DSR. Dans DSR, les paquets non délivrés sont récupérés aux nœuds intermédiaires en utilisant des routes alternatives qui existent dans la cache. En plus de ce mécanisme pour la récupération de données, DSRoB utilise aussi des informations obtenues lors de la maintenance de la dorsale. Un mécanisme simple pour réparer une route qui n'est plus disponible consiste à examiner la séquence de nœuds dans le paquet et à localiser un nœud que nous connaissons avec le chemin basé sur l'information lors d'une maintenance. La séquence de nœuds sera alors modifiée. Si le même nœud se retrouve deux fois dans la nouvelle séquence ceci indique qu'il y a une boucle et le paquet sera abandonné.

6.3.2 *AODV over Backbone* (AODVoB)

Comme présenté à la section 2.3.2.2, le protocole AODV repose sur trois principales composantes :

- l'initiation et la propagation des messages RREQ,
- l'initiation et la propagation des messages RREP,
- la maintenance de tables de vecteurs de distance.

Dans AODVoB, seule la propagation des messages RREQ est différente. En effet, la diffusion de RREQ se fait comme dans DSRoB. Quand une source envoie le message RREQ pour une route vers une destination, le message ne se propage que dans les nœuds qui forment la dorsale jusqu'au nœud dominant directement connecté à la destination. La destination répond alors par un message RREP. La propagation du message RREP est la même comme dans AODV, le RREP arrivant à la source avec une mise à jour du numéro de séquence de la source. Tous les nœuds intermédiaires recevant le message RREP mettent à jour leurs tables de routage avec le nouveau numéro de séquence de la

source. A contrario dans le protocole AODV, il y aura moins de RREP en retour étant donné que le nœud de destination possède un voisin direct dominant de la dorsale et la route passera par ce dominant.

L'utilisation d'une dorsale avec le protocole AODV permet de :

- minimiser l'*overhead* dans RREQ, la propagation des messages RREQ dans la dorsale réduit le calcul de l'*overhead* dans le message RREQ. Alors qu'AODV diffuse le message RREQ dans pratiquement tous les nœuds du réseau, AODVoB restreint cette diffusion dans la dorsale uniquement,
- éliminer la procédure *Ring Zero Search*, comme dans DSRoB. La source va toujours envoyer sa demande au nœud dominant le plus proche de la dorsale et c'est le rôle de ce nœud de vérifier si la destination est dans son voisinage,
- maintenir une route AODVoB utilise le même principe que DSRoB pour la maintenance d'une route, suite à un changement dans la topologie, à l'exception de la séquence des nœuds qui n'existent pas dans DSRoB. Dans AODVoB, la dorsale maintient des informations explicites pour trouver une nouvelle route quand l'ancienne route n'est plus valable. Quand le message RREP est retourné vers la source, chaque nœud intermédiaire de la dorsale qui reçoit ce message va ajouter l'identificateur du nœud prochain à la dorsale dans le chemin vers la destination. Quand une route n'est plus valable, le dominant en amont va tenter, avec l'information qu'il possède après la maintenance, de trouver le prochain nœud pour la destination.

6.3.3 *TORA over Backbone* (TORAoB)

Le protocole TORA a été conçu dans le but d'éliminer les boucles de routage en utilisant le principe des graphes acycliques orientés (DGA). Le même principe est utilisé dans TORAoB, cependant un DGA de taille plus petite sera formé ; seuls les membres de la

dorsale participent à la formation d'un DGA vers une destination donnée. Le nombre de messages QRY/UPD qui transitent dans le réseau sera alors minimisé. Par le même principe, les messages CLR utilisés lors d'une détection de défaillance ne seront pas diffusés dans tout le réseau.

La détermination des chemins optimaux ne fait pas partie des principales tâches du protocole TORA. Toutefois, ceci peut être amélioré en présence de la dorsale étant donné que les nœuds qui forment la dorsale sont connectés de façon minimale.

6.3.4 Évaluation des performances

Dans cette section, nous allons utiliser les mêmes critères de performances tels que définis dans la section 6.2. Autrement dit, nous allons examiner le taux de paquets délivrés, le nombre moyen de sauts, le nombre de paquets de données transmis par rapport au nombre de paquets de données délivrés, le nombre d'octets utilisés pour le trafic de contrôle et le délai moyen de bout en bout. D'autre part, nous allons également utiliser les mêmes paramètres de simulation définis dans la section précédente pour évaluer les performances des protocoles de routage en présence d'une dorsale (DSRoB, AODVoB et TORAoB) et en fonction de la mobilité.

En plus de la mobilité, nous allons prendre en considération la charge dans le réseau. A cet effet, nous allons examiner dans un premier temps la variation du taux de paquets délivrés et le délai de bout en bout en fonction de la charge exprimée en termes du nombre de communications simultanés (session).

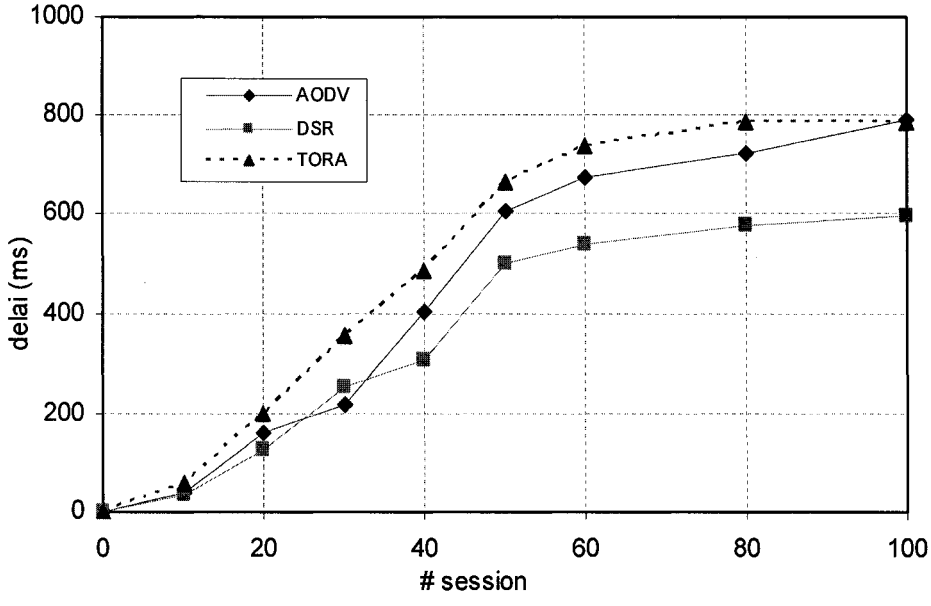
Nous allons considérer un réseau de taille 50 nœuds et une vitesse moyenne des nœuds de 10 *m/s*. Les nœuds sont placés aléatoirement dans un plan de taille 1.5 *km* x 1.5 *km* et la durée de simulation est de 250*s*. Un trafic de type CBR est généré entre deux paires de nœuds (sources, destinations). Chaque source possède un taux de transmissions entre 1 à 8 paquets/s et la taille des paquets est égale à 512 octets. Afin de varier la charge dans le

réseau, nous allons varier le nombre de session de 10 à 100. Cette série de simulation est effectuée dans un réseau sans dorsale.

Les résultats de simulation, figure 59-a, nous montrent que le délai de bout en bout augmente graduellement avec la charge dans le réseau. La charge du réseau est exprimée en nombre de session simultané. Il est important de voir que pour un nombre de session supérieure à 70, on arrive à une charge où le réseau commence à être congestionné. La figure 59-b confirme ce que nous venons de dire, en effet, avec un nombre de session élevé (>70), les trois protocoles n'arrivent pas à livrer plus que la moitié du trafic. Ceci est expliqué par le fait que le nombre *d'overhead* de contrôle généré augmente ce qui laisse moins de bande passante disponible pour le trafic de données et ce qui augmente la chance des paquets droppés dû à la collision et au débordement des files d'attente dans les nœuds.

Dans la suite, nous allons considérer deux niveaux de charge ; une charge moyenne (zone linéaire figure 59-a) réalisée par 40 sessions et un niveau de charge élevée (réseau congestionné) pour un nombre de session égale à 80.

(a)



(b)

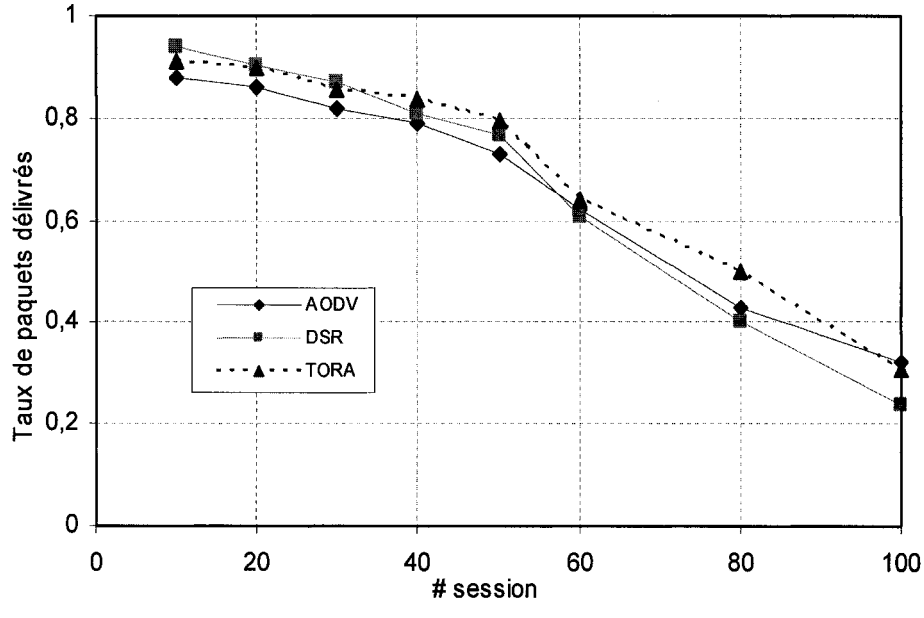


Figure 59 Délai de bout en bout et taux de paquets délivrés en fonction de la charge du réseau (# session)

6.3.4.1 Taux de paquets délivrés

Dans cette section, le taux de paquets délivrés est illustré dans la figure 60 pour les trois protocoles de routage (AODV, DSR et TORA) avec et sans dorsale. Rappelons que ce taux représente le nombre de paquets de données délivrés aux destinations par le nombre de paquets générés par les sources. Nous examinons deux niveaux de charge, soit une charge moyenne (40 sessions) et une charge élevée (80 sessions). Sur la figure 60, le nombre entre parenthèses indique le nombre de session.

D'après les résultats de simulations, nous pouvons remarquer qu'il y a une amélioration des performances des protocoles de routage en présence de la dorsale pour les niveaux de charge moyen et élevé. Pour un niveau de charge moyen, le taux de paquets délivrés peut atteindre les 80% pour les trois protocoles et pour une mobilité assez élevée (20m/s). Cette amélioration est de l'ordre de 17 % pour AODV, 12 % pour DSR et 4 % pour TORA par rapport à leur version originale. Cette amélioration est à l'origine principalement de la réduction des messages de contrôle (RREQ et RREP dans DSR et AODV et *Query* et *Update* dans TORA). En effet, les messages de contrôle ne sont plus diffusés dans tout le réseau, seuls les nœuds dominants échangent ces messages. Une réduction des messages de contrôle engendre une bande passante plus disponible pour le trafic de données.

Pour un niveau de charge élevé (80 session), les trois protocoles éprouvent la difficulté à délivrer plus que 80% des paquets pour une mobilité faible. Un nombre de session élevé engendre une augmentation du trafic de contrôle et par la suite une bande passante moins disponible pour le trafic de données. Un grand nombre de paquets sera droppé dans les nœuds congestionnés. Le protocole DSR performe mieux que les deux autres protocoles (AODV et TORA) pour un niveau de charge moyen et élevé en présence de la dorsale.

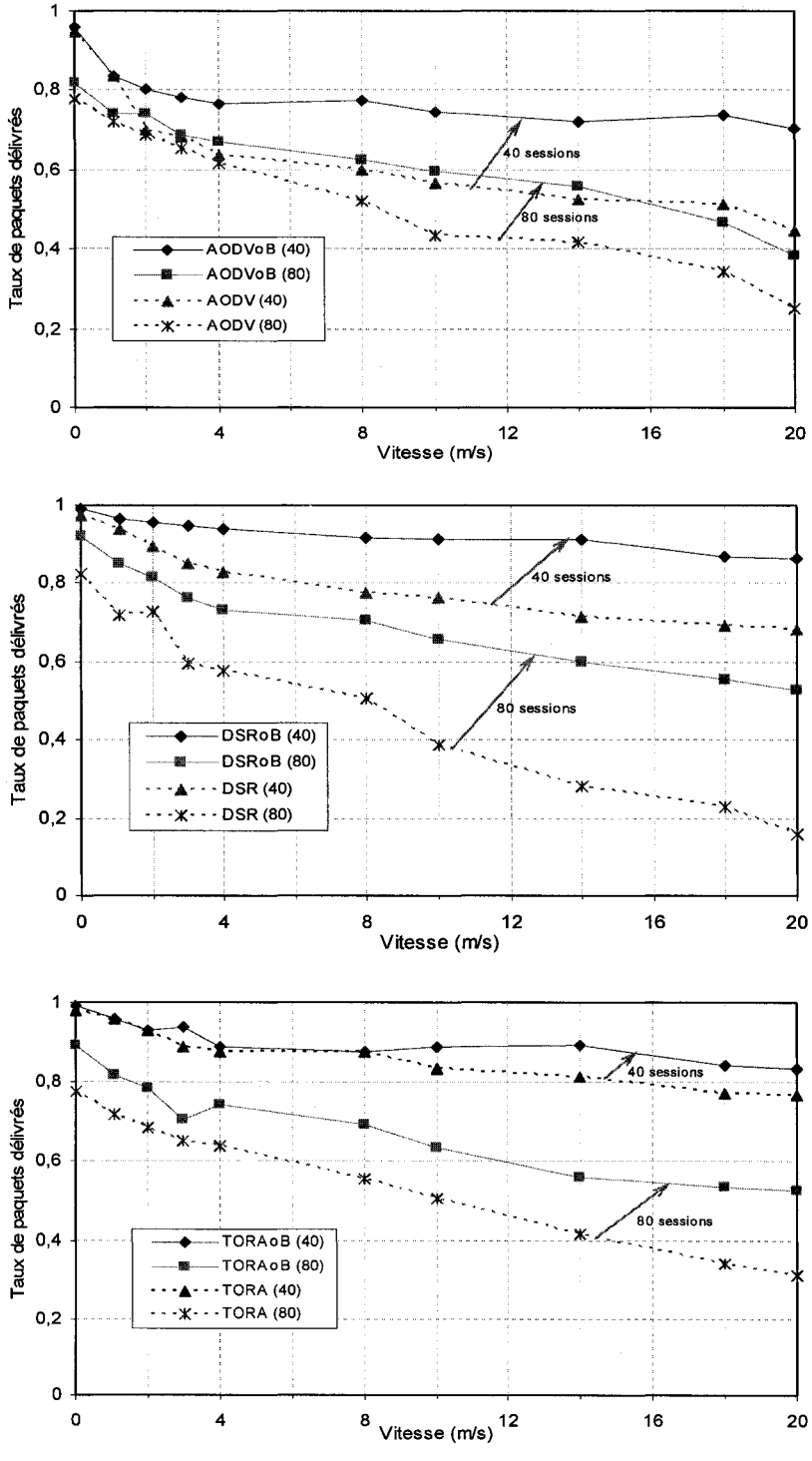


Figure 60 Nombre de paquets délivrés par rapport au nombre de paquets envoyés en fonction de la mobilité

6.3.4.2 Nombre de sauts

La figure 61 illustre les résultats de simulation du nombre de sauts moyen en fonction de la mobilité. Pour des raisons de clarté graphique, nous avons représenté uniquement les résultats pour une charge moyenne (40 sessions). En effet, les résultats de simulations obtenus pour une charge élevée (80 sessions) sont très proches de ceux pour une charge moyenne. Ceci est tout à fait raisonnable étant donné que le nombre de sauts est calculé uniquement pour les paquets qui se rendent à la destination et il ne dépend que de la disponibilité de la route entre la source et la destination.

D'après les résultats de simulation, nous ne remarquons pas beaucoup d'amélioration des performances des protocoles de routage en termes de nombre de sauts quand la dorsale est utilisée. Seul le protocole TORA a pu bénéficier de la présence de la dorsale et le nombre de sauts a diminué d'un peu plus de 10 %. Rappelons que le protocole TORA ne possède pas de mécanisme qui lui permet de déterminer les routes optimales. TORAoB bénéficie des informations sur la topologie lors de la construction et de la maintenance de la dorsale pour diminuer le nombre moyen de sauts entre la source et la destination.

Le protocole AODV se comporte pratiquement de la même façon en absence ou en la présence d'une dorsale, alors que le protocole DSR ne tire pas d'avantage de la présence de la dorsale en ce qui concerne le nombre moyen de sauts. Les routes sont plus longues pour DSRoB. En utilisant la dorsale, les routes doivent passer par les nœuds dominants qui, dans certains cas, ne sont pas minimales.

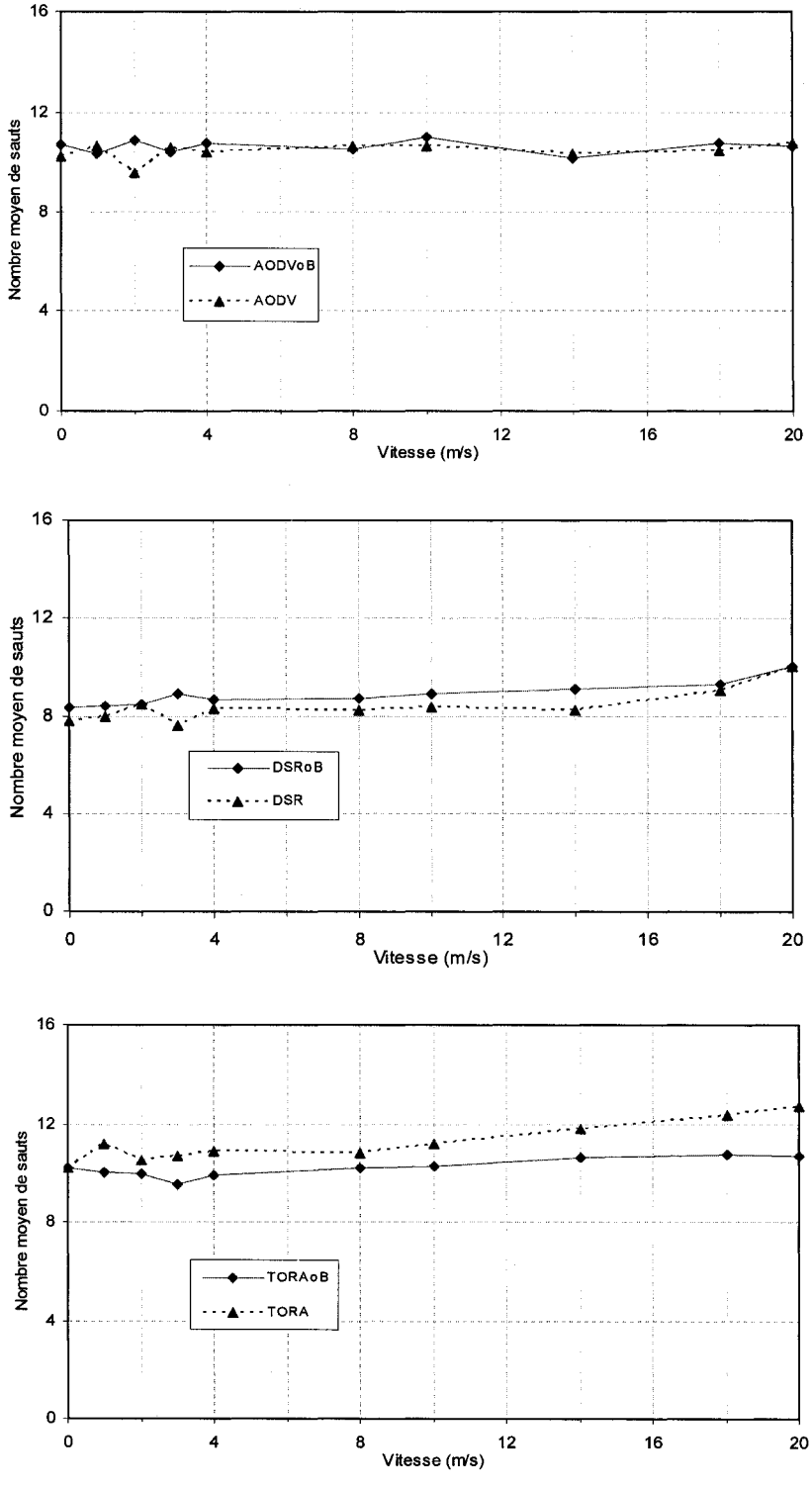


Figure 61 Nombre moyen de sauts en fonction de la mobilité

6.3.4.3 Nombre de paquets de données transmis par rapport au nombre de paquets de données délivrés

Dans cette simulation, nous avons évalué le nombre de paquets de données retransmis par tous les nœuds intermédiaires par rapport au nombre de paquets délivrés aux destinations. Ce nombre nous permet d'obtenir l'information sur le nombre de retransmissions nécessaires des paquets de données pour se rendre à la destination. Comme nous l'avons mentionné dans la section 6.2, il est préférable d'avoir un nombre de l'ordre du nombre de sauts moyen. En effet, plus ce nombre dépasse le nombre de sauts moyen, plus il y a des paquets de données redondants dans le réseau.

Les résultats de la figure 62 montrent que le nombre de retransmission des paquets augmente quand la charge dans le réseau augmente. Cette augmentation est due principalement au non disponibilité des routes entre la source et la destination ; les nœuds intermédiaires sont surchargés. Avec l'utilisation de la dorsale, nous avons remarqué que le nombre de retransmissions dans le réseau a été considérablement réduit particulièrement pour les deux protocoles AODV et TORA, qui possèdent des résultats déplorables en absence de la dorsale. Cette réduction est due essentiellement à l'information disponible dans les nœuds dominants. Cette information sera alors utilisée par le protocole de routage en question pour acheminer ses paquets de données vers la destination.

Pour une charge moyenne ou élevée, la présence d'une dorsale a permis de réduire le nombre de retransmissions non utiles dans le réseau : une réduction en nombre de retransmissions de l'ordre de 20 % et 30 % pour les protocoles AODV et TORA respectivement. Pour le protocole DSR, il n'y a pas de réduction significative (2 %) dans le cas d'une charge moyenne, alors que cette réduction est de l'ordre de 20% pour une charge élevée. Rappelons que DSR utilise une technique de récupération qui lui permet d'arrêter la retransmission une fois qu'il détecte qu'une route n'est plus valide.

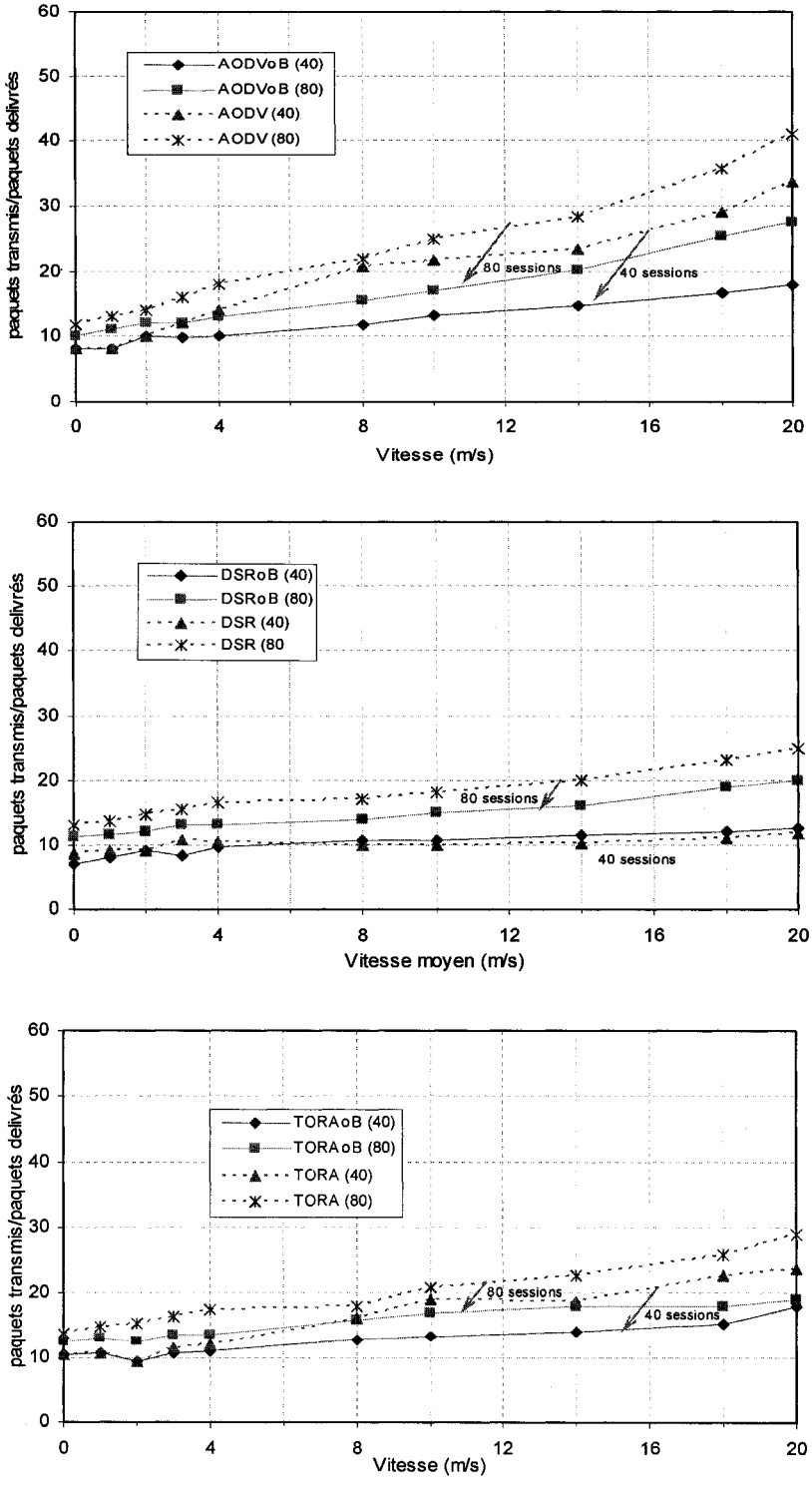


Figure 62 Nombre moyen de paquets transmis par paquets délivrés en fonction de la mobilité

6.3.4.4 Nombre d'octets utilisés pour le trafic de contrôle

Comme illustré sur la figure 63, le trafic de contrôle pour tous les protocoles réactifs a été considérablement réduit quand la dorsale est utilisée et il croît légèrement en fonction de la mobilité. Pour un système statique (vitesse nulle) ou une mobilité faible nous remarquons que le trafic de contrôle en présence de la dorsale est plus important que celui sans dorsale pour les trois protocoles de routage. Le trafic additionnel provient du trafic utilisé par la maintenance de la dorsale. Ceci est observé pour les deux niveaux de la charge dans le réseau et pour la plupart des protocoles de routage.

Quand la mobilité augmente, le trafic de contrôle augmente légèrement. Cette légère augmentation du trafic de contrôle peut être expliquée par le fait que la taille de la dorsale augmente en fonction de la mobilité (comme nous l'avons discuté dans le chapitre précèdent). Rappelons que le trafic de contrôle est échangé uniquement par les nœuds qui forment la dorsale. Nous notons aussi que les protocoles DSRoB et AODVoB génèrent moins de trafic de contrôle que le protocole OLSR (figure 56) même pour un taux de mobilité élevé. Le protocole DSRoB génère moins de trafic de contrôle comparé aux deux autres protocoles dans un environnement dynamique.

Pour une charge élevée, la présence de la dorsale a permis de diminuer le trafic de contrôle. Toutefois, cette réduction est beaucoup moins importante que dans le cas d'une charge moyenne.

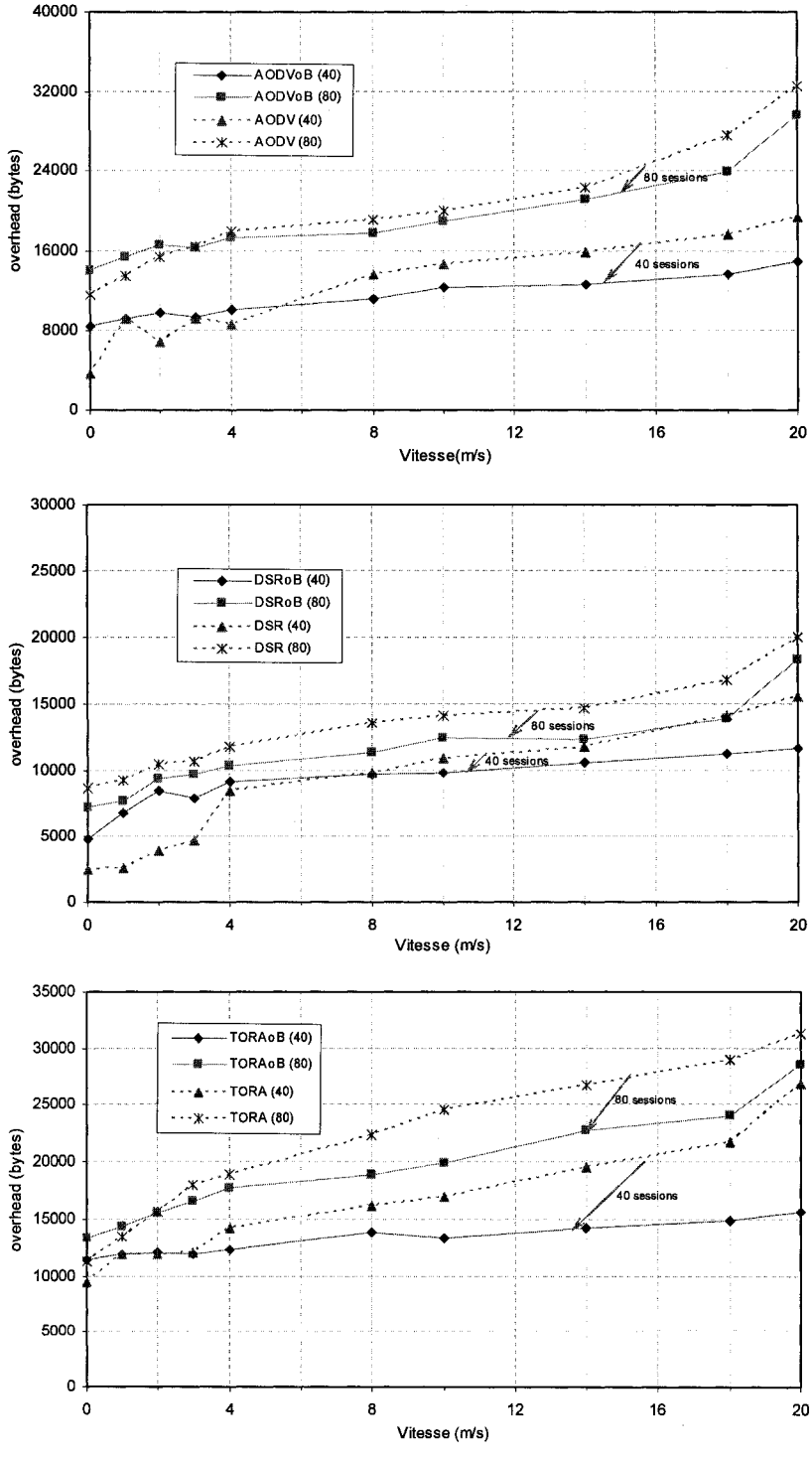


Figure 63 Nombre d'octets utilisés pour le trafic de contrôle en fonction de la mobilité

6.3.4.5 Délai moyen de bout en bout

La figure 64 illustre la variation du délai moyen de bout en bout en fonction de la mobilité pour les protocoles de routage AODV, DSR et TORA. Quand la mobilité des nœuds augmente, nous observons bien une amélioration pour les trois protocoles réactifs DSRoB, AODVoB et TORAoB, elle est de l'ordre de 27 %, 19 % et 30 % respectivement. La réduction du nombre de retransmission dans le réseau et le trafic de contrôle permet de diminuer le temps de séjour dans les tampons et le temps de traitement dans les nœuds intermédiaires.

Pour une mobilité réduite, les résultats de simulation montrent que le délai moyen de bout en bout en présence de la dorsale est légèrement supérieur comparé à celui dans le cas où la dorsale n'est pas employée. Le délai additionnel provient du fait que les routes sélectionnées par les protocoles en présence de la dorsale sont plus longues que celles des protocoles dans leurs versions originales, figure 56. Les résultats de simulations montrent que le protocole DSRoB performe mieux que les autres protocoles pour un niveau de mobilité élevé.

Tous les protocoles de routage possèdent un délai important quand la charge dans le réseau est élevée (80 sessions). Ceci est expliqué par le fait que les tampons dans les nœuds intermédiaires sont pleins et le temps de séjour des paquets dans ces tampons va augmenter. Nous remarquons aussi que le protocole qui délivre plus de paquets (DSR) possède le délai le plus faible. La présence d'une dorsale permet de réduire le délai quand la mobilité augmente. Toutefois, cette réduction est moins importante que dans le cas d'une charge moyenne.

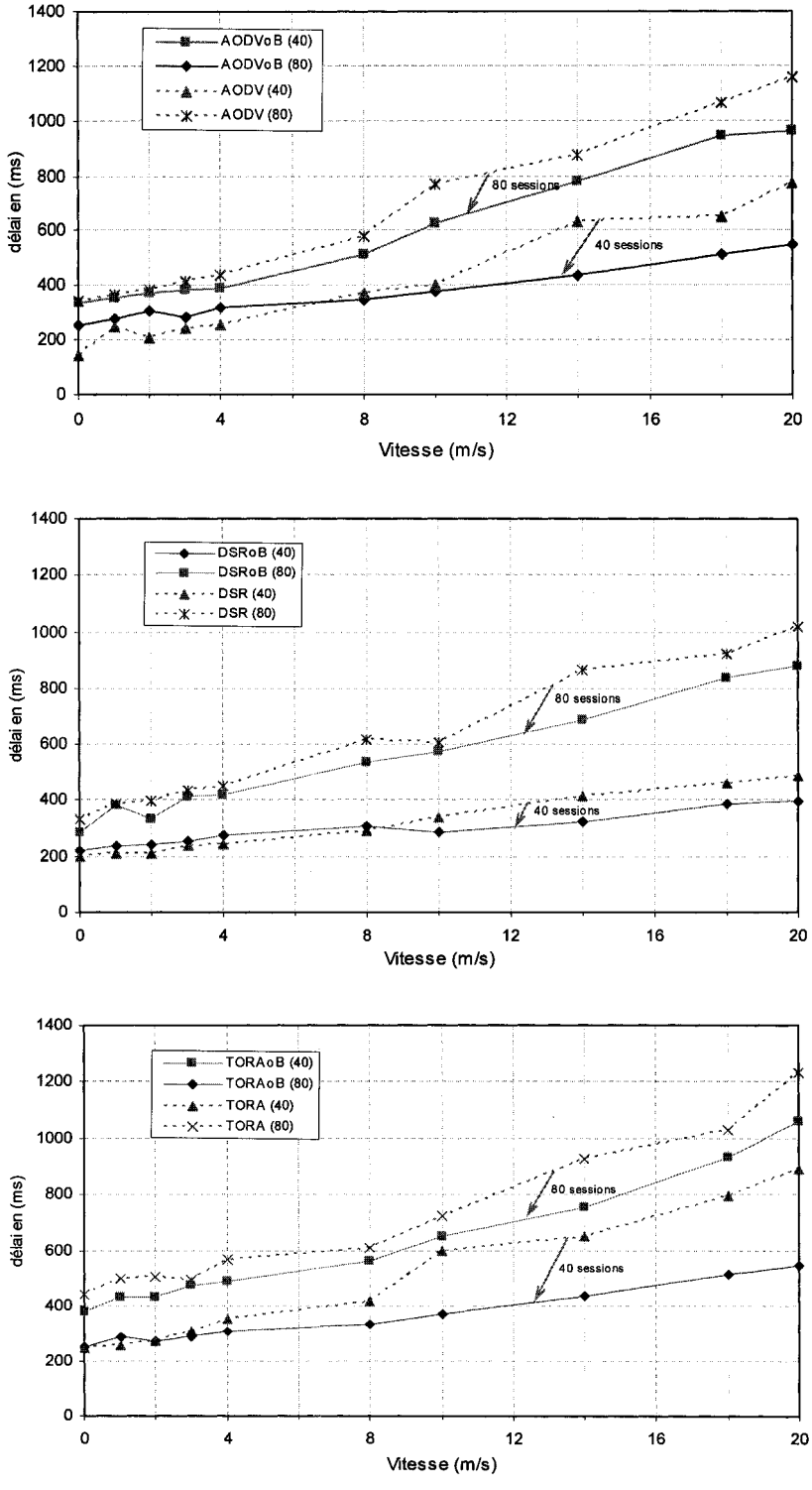


Figure 64 Délai moyen en fonction de la mobilité

Tableau VI

Sommaire des résultats obtenus

Mobilité faible

	Charge moyenne				Charge élevée			
	taux de paquets délivrés	retransmission	overhead	délai	taux de paquets délivrés	retransmission	overhead	délai
AODV	+12%	-5%	+18%	+15%	+10%	-10%	+8%	-3%
DSR	+8%	-4%	+25%	+8%	+16%	-12%	-10%	-4%
TORA	+4%	-2%	+7%	+4%	+12%	-6%	+12%	-4%

Mobilité elevée

	Charge moyenne				Charge élevée			
	taux de paquets délivrés	retransmission	overhead	délai	taux de paquets délivrés	retransmission	overhead	délai
AODV	+25%	-37%	-23%	-10%	+21%	-32%	-11%	-8%
DSR	+28%	-2%	-18%	-5%	+36%	-12%	-15%	-4%
TORA	+17%	-22%	-12%	+4%	+25%	-6%	-18%	-10%

6.3.5 Interprétation des résultats

Le tableau VI récapitule l'écart relatif entre les résultats de simulation pour les protocoles de routage (DSR, AODV et TORA) avec et sans la dorsale pour un niveau de charge moyen (40 sessions) et élevé (80 session) et pour une mobilité faible [0 ~ 4m/s] et élevée [14 ~ 20m/s]. Dans ce tableau, le signe (+) indique une amélioration pour le taux de paquets délivrés. De même, le signe (-) indique une réduction de retransmission, *overhead* et délai. Nous avons intérêt à augmenter le taux de paquet délivré et diminuer le nombre de retransmission, *l'overhead* et le délai. Par exemple, le +12% pour le taux de paquet délivré (protocole TORA, charge moyenne, mobilité faible) indique qu'en utilisant la dorsale, TORAoB, il y a une amélioration de l'ordre de 12% de plus de paquets délivrés par rapport aux résultats de TORA (sans dorsale). Les cases en gris

représentent les cas où l'utilisation de la dorsale est moins efficace qu'avec le protocole en question. Par exemple, le protocole AODV génère plus de paquet de contrôle en présence d'une dorsale. En effet, pour une charge moyenne et pour une mobilité faible, les résultats de simulation du protocole AODVoB montrent qu'il y a 18% de trafic de contrôle additionnel comparés à celui d'AODV.

En se basant sur l'analyse des résultats de simulation, nous avons remarqué que l'utilisation de la dorsale est avantageuse et permet d'améliorer la performance des protocoles de routages (AODV, DSR et TORA) pour un niveau de mobilité élevée (peu importe le niveau de la charge dans le réseau). Ceci est expliqué par le fait que la construction de la dorsale a permis de limiter la zone de diffusion des messages de contrôle et le nombre de retransmission dans le réseau. Il s'en suit que plus de bande passante disponible pour le trafic de donnée. Aussi, notre procédure de maintenance a prouvé son efficacité dans un environnement dynamique. En effet, elle est de nature distribuée et elle s'adapte bien avec tout changement de la topologie, ce qui permet d'avoir des informations pertinentes qui seront utiles pour les protocoles de routage. Ces informations concernent essentiellement la topologie du réseau. À un instant donné, chaque dominant connaît à priori la liste des dominés qui doit desservir.

Pour une mobilité faible, la présence d'une dorsale n'apporte pas beaucoup d'amélioration. Ceci est dû au fait que la construction et la maintenance de la dorsale engendre un *overhead* additionnel sans qu'il soit compensé par le niveau de mobilité. Surtout pour une charge moyenne, la performance des trois protocoles se dégrade en termes de trafic de contrôle et de délai. Cependant, nous observons une légère amélioration au niveau du taux de paquets délivrés et le nombre de retransmission. Dans le cas où nous avons une charge élevée, les performances des protocoles AODV et TORA se dégradent légèrement en termes de trafic de contrôle additionnel. Pour les autres paramètres (délai, taux de paquets délivrés et nombre de retransmission), la

présence de la dorsale contribue à une légère amélioration pour les trois protocoles de routage.

Tableau VII

proportion du trafic de contrôle par rapport au trafic global

	Mobilité faible				Mobilité élevée			
	Charge moyenne (40 sessions)		Charge élevée (80 sessions)		Charge moyenne (40 sessions)		Charge élevée (80 sessions)	
	sans dorsale	avec dorsale	sans dorsale	avec dorsale	sans dorsale	avec dorsale	sans dorsale	avec dorsale
AODV	33%	37%	41%	37%	48%	27%	61%	32%
DSR	28%	34%	37%	29%	46%	22%	56%	28%
TORA	31%	33%	42%	44%	51%	33%	54%	30%

Le trafic de contrôle est présenté dans l'unité absolue et pas relatif par rapport au trafic total parce que nous avons voulu étudier la variation (évolution) du trafic de contrôle en fonction de la variation de la mobilité. Le tableau VII illustre le pourcentage du trafic de contrôle par rapport au trafic global pour les trois protocoles de routage AODV, DSR et TORA. Le trafic de contrôle consomme une bonne partie de la bande passante ; entre 30% et 50% de la bande passante est utilisé pour le trafic de contrôle. La proportion du trafic de contrôle par rapport au trafic global varie en fonction du protocole et en fonction du niveau de la mobilité. Ces résultats montrent bien que la présence de la dorsale permet de réduire cette proportion de la bande passante utilisée par le trafic de contrôle surtout pour une mobilité élevée.

Si nous comparons le protocole OLSR, qui utilise la technique de relais multipoint pour diffuser les messages contrôles dans le réseau, nous pourrons dire que pour un même niveau de charge le taux de paquets délivrés pour les protocoles de routage réactifs DSR et TORA, sans la dorsale, dépasse celui de OLSR et la présence de la dorsale permet

d'avantage d'améliorer ce taux. En ce qui concerne le nombre de retransmission OLSR présente de meilleurs résultats comparés aux autres protocoles sans la dorsale. Toutefois, en présence de la dorsale celle-ci diminue d'avantage le nombre de retransmission. OLSR et les trois protocoles réactifs en présence de la dorsale possèdent le même ordre de grandeur en ce qui concerne le trafic de contrôle (*overhead*). Le délai pour DSRoB est faible par rapport à celui d'OLSR, alors que pour AODVoB et TORAoB ce délai est toujours important comparé à celui de OLSR.

6.4 Conclusion

Dans ce chapitre, nous avons comparé, au moyen des simulations, quatre protocoles de routage ; un protocole proactif (OLSR) et trois protocoles réactifs (AODV, DSR et TORA). Nous nous sommes intéressés principalement à leurs comportements face à la mobilité des nœuds dans le réseau. Pour cela nous avons défini cinq critères (ou métriques) pour effectuer cette comparaison. Les résultats de simulation montrent que les performances de ces protocoles se dégradent quand la mobilité augmente : le taux de paquets délivrés diminue, la retransmission inutile augmente, le trafic de contrôle augmente, le délai augmente etc.

Nous avons par la suite repris les mêmes protocoles en présence d'une dorsale et refait les mêmes simulations. De plus, nous avons tenu compte du niveau de la charge du trafic dans le réseau en faisant varier le nombre de sessions simultanées durant la simulation. Nous avons remarqué qu'il y a une amélioration considérable surtout quand le niveau de mobilité augmente. En effet, les trois protocoles réactifs, pris en considération, performent mieux en présence d'une dorsale pour une charge moyenne et élevée dans le réseau. Cette amélioration varie et elle est de 10 à 30 % pour certaines métriques. Cette amélioration est due principalement aux facteurs suivant :

- la réduction de la zone de diffusion des paquets de contrôle dans le réseau et la limitation de nombre de retransmissions inutiles assure une meilleure utilisation de la bande passante dans le réseau,
- l'information, même locale, disponible dans la dorsale sur la position des nœuds dominés à servir par le dominant. Cette information est utile pour le protocole de routage dans la phase de la découverte d'une route,
- la maintenance distribuée qui assure, dans certaines mesures, la stabilité et la connectivité de la topologie même pour une mobilité élevée.

Dans un réseau *ad hoc*, nous avons constaté que la présence d'une dorsale a permis de contribuer à l'amélioration des performances des protocoles de routage en présence de la mobilité. Pour une mobilité élevée et une charge moyenne, cette amélioration des performances est remarquable comparativement à celle des protocoles sans dorsale. En effet, la dorsale a permis de minimiser le trafic de contrôle, de réduire les retransmissions inutiles et de réduire l'effet de la mobilité. Et par conséquent, la consommation de la bande passante a été réduite. Pour une charge élevée (et peu importe le niveau de la mobilité), la dorsale a permis d'améliorer les performances mais pas autant que celles pour une charge moyenne.

Pour un système statique ou à faible mobilité, la présence d'une dorsale n'est pas si avantageuse que dans le cas à forte mobilité. Nous avons remarqué que les protocoles de routage sont capables de s'adapter dans un environnement peu dynamique. En effet, le délai et le trafic de contrôle sont plus faibles quand la dorsale est utilisée. Pour une mobilité réduite, le trafic de contrôle généré pour la maintenance de la dorsale ne sera pas compensé et affecte la performance des protocoles.

Dans nos simulations, nous avons considéré une topologie uniforme et un trafic homogène et nous avons remarqué que la présence d'une dorsale améliore les

performances des protocoles de routage existants. Pour une topologie non-uniforme, c'est la connectivité des nœuds qui va être modifié et par la suite la taille de la dorsale qui va changer et nous présumons que la présence d'une dorsale va toujours apporter des améliorations. Pour un trafic hétérogène la performance va dépendre principalement de la bande passante disponible dans le réseau qui augmentera en présence de la dorsale. Nous avons vérifié que la présence de la dorsale permet de réduit le trafic de contrôle dans le réseau et par la suite il y aura plus de bande passante disponible pour le trafic de données.

En conclusion, dans un réseau *ad hoc*, l'utilisation d'une dorsale avec un protocole de routage est avantageuse pour un environnement dynamique (forte mobilité) peu importe le niveau de la charge dans le réseau. Tandis que pour un environnement statique et une charge élevée, l'utilisation d'une dorsale est peu avantageuse. Pour un environnement statique et une charge moyenne, l'utilisation d'une dorsale ne présente aucun avantage. Ainsi, le choix d'utilisation d'une dorsale ou non dépend de la nature de l'application en question.

CONCLUSION

Au cours des dernières années, la recherche dans le domaine des réseaux *ad hoc* a connu une croissance remarquable. Le perfectionnement des systèmes d'économie de puissance et la miniaturisation des dispositifs portatifs ont accéléré la croissance des environnements sans fil de réseau. Les réseaux *ad hoc* offrent la commodité d'un rapide déploiement, d'une mobilité sans contrainte, etc. Ils sont caractérisés en particulier par l'absence d'une infrastructure fixe, une structure dynamique et des ressources limitées. Les ressources comprennent entre autres l'énergie de la batterie, la puissance de transmission et la bande passante de communication.

Nous sommes partis de l'hypothèse qu'un bon contrôle de la topologie dans un réseau *ad hoc* permettrait, d'une part, d'améliorer la performance des protocoles de routage et, d'autre part, d'optimiser l'utilisation des ressources (bande passante et énergie) dans le réseau. Nous avons alors procédé à la comparaison des différentes techniques pour réaliser le contrôle de la topologie dans un réseau *ad hoc*. Ces techniques sont les suivantes : technique basée sur le contrôle de la puissance de transmission, technique basée sur la formation des *clusters* et technique basée sur la construction d'une dorsale. En se basant sur la littérature des différentes techniques, nous avons remarqué que la technique basée sur la construction d'une dorsale est meilleure que les deux autres techniques selon les critères suivants : la connectivité, la procédure de construction et la maintenance. En effet, le contrôle basé sur la puissance de transmission ne garantit pas la connectivité du réseau. De plus, nous avons considéré le cas des réseaux *ad hoc* homogènes où les terminaux possèdent le même type de source d'énergie et la même autonomie. Quant à la construction et à la maintenance, la formation des *clusters* pose un problème au niveau du critère pour choisir les *clusterheads* ainsi que le délai engendré dans le processus de réélection une fois que le *clusterhead* change de position.

La motivation de notre recherche était de proposer une nouvelle approche pour construire et maintenir une dorsale dans un réseau *ad hoc* mobile. D'abord et avant tout, nous avons revu les approches proposées pour construire la dorsale. Nous avons trouvé que la plupart des propositions ne s'intéressent pas à la maintenance et pour ceux qui considèrent la maintenance, un seul algorithme est proposé couvrant la construction et la maintenance. Nous nous sommes basés sur le fait qu'il est plus efficace de maintenir une dorsale de taille minimale, nous avons alors proposé un algorithme pour la construction et une procédure de maintenance. L'algorithme qui construit la dorsale dans la phase d'établissement (*setup*) permet de garantir une taille minimale de la dorsale. La procédure de maintenance est distribuée et sera appliquée par le terminal qui change de position : une maintenance locale.

La construction de la dorsale dans la phase d'établissement est basée sur la détermination de l'ensemble MCDS dans un graphe. Nous avons établi une formulation qui garantit une taille minimale à cet ensemble. Nous avons validé la taille de la solution finale fournie par notre algorithme au moyen d'une analyse basée sur un modèle probabiliste. De plus, nous avons trouvé que notre algorithme performe mieux que deux autres algorithmes populaires fréquemment utilisés dans la littérature.

La maintenance de la dorsale quant à elle est assurée par une procédure distribuée. Au moyen d'un échange de quelques paquets *hello*, un nœud qui change de position va essayer de se connecter au nœud le plus proche et qui fait partir de la dorsale. En réalité, il s'agit de deux maintenances locales : la première dans la nouvelle région où le nœud va se retrouver et la deuxième dans la région d'où le nœud venait.

Au moyen des simulations, nous avons vérifié deux critères soit :

- **l'efficacité** : une dorsale de taille minimale signifie qu'un nombre limité de nœuds dans le réseau doivent participer pour diffuser un message. Ainsi, le nombre de messages est minimisé ce qui se traduit par une économie de la bande

passante et par l'énergie dissipée, ce qui permet également de contourner le problème *broadcast storm*,

- **la robustesse** : une procédure distribuée garantit la maintenance de la dorsale suite à un changement de la topologie dans le temps causé par le déplacement des nœuds. Cette procédure garantit aussi le passage à l'échelle (*scalability*) du fait que la maintenance se fait localement. En effet, nous avons considéré deux paramètres : (i) le pourcentage de connectivité versus la mobilité et (ii) la variation du nombre de nœuds dans le réseau. Plus de 90 % des terminaux restent connectés pour une mobilité forte ($30m/s$), alors que ce pourcentage est quasiment constant quand le nombre de nœuds varie dans le réseau. Ces résultats prouvent que la procédure de maintenance assure une certaine stabilité de la topologie du réseau pour une mobilité élevée. De plus, la procédure de maintenance garantit une certaine distribution uniforme d'énergie étant donné qu'à long terme tous les nœuds auront la même probabilité pour devenir dominant ; plus de 70 % des nœuds vont passer entre 20 et 26 fois à l'état dominant.

Dans le but de valider notre approche, nous avons comparé la performance de trois protocoles de routage face à la mobilité, sans et avec une dorsale. Les résultats de simulation ont démontré qu'avec une dorsale les protocoles réactifs performent mieux. Une amélioration des performances variant de 10 à 30 %, suivant le protocole en question, a été illustrée lors de différents scénarios de simulation.

Le routage s'appuyant sur une structure virtuelle est plus stable que celui basé sur la topologie physique. Cette constatation est faite suite à l'évaluation des paramètres comme le taux de paquets délivrés, le nombre de sauts, le nombre de retransmissions et le délai. Cette structure offre au réseau d'augmenter ses performances dans un environnement dynamique et un taux de mobilité élevé. La dorsale permet de limiter, de façon considérable, les paquets de contrôle induis par les procédures de découverte

utilisées par les protocoles de routage. En effet, seuls les nœuds de cette dorsale relaient un paquet de contrôle. La dorsale formée par l'ensemble des nœuds connectés remplit parfaitement une zone de service pour le trafic de contrôle, problème majeur dans les réseaux *ad hoc*.

RECOMMANDATIONS

Dans ce travail de recherche, nous nous sommes intéressés au contrôle de la topologie dans les réseaux *ad hoc* MANET. Nous avons montré que le contrôle de la topologie basé sur la construction d'une dorsale virtuelle permet d'améliorer les performances de certains protocoles (DSR, AODV et TORA) de routage conçus pour les réseaux ad hoc. Toutefois, d'autres protocoles point à point et même point à multipoint peuvent être utilisés et bénéficier ainsi de l'information fournie par la dorsale.

Il serait intéressant aussi de vérifier le comportement de notre solution avec des scénarios plus réalistes. Voici différents scénarios possibles qui illustrent des événements réalistes comme ceux présentés dans [Johansson *et al.* (1999)] :

- dans une salle de conférence, où il n'y a pas beaucoup de mouvement pendant une longue durée de temps, soudainement tout le monde, ou presque, bouge en même temps pendant une pause,
- lors d'un désastre : un certain nombre de nœuds qui représentent des véhicules d'urgence se déplacent à une vitesse élevée et d'autres représentent les humains (secouristes) qui se déplacent avec une faible vitesse (1 à 2m/s).

Nous avons considéré une topographie carrée ($1500m$x$1500m$) pour effectuer les simulations, il serait intéressant aussi de considérer une topographie rectangulaire. Un changement dans la géométrie de la topographie engendre une variation dans la densité des nœuds, pour un même nombre de nœuds. De plus, les routes sont plus longues pour une topographie rectangulaire comparées à celles pour une topographie carrée pour une même densité des nœuds.

Dans les simulations pour la construction et la maintenance, nous avons tenu compte de deux facteurs : la mobilité des nœuds et la mise à l'échelle (*scalability*). Les résultats ont démontré que la procédure de maintenance est invariable par rapport à la taille du réseau du fait qu'elle est distribuée. Dans les simulations que nous avons effectuées pour étudier la performance des protocoles de routage, nous n'avons considéré que la mobilité. Il serait intéressent de vérifier la mise à l'échelle pour les protocoles de routage en présence de la dorsale. Aussi, nous n'avons considéré qu'un seul type de trafic (CBR). Il serait intéressent de considérer d'autre type de trafic tel que le TCP par exemple [Clausen T. (2004)]. Deux types de problèmes qui peuvent être identifiés quand TCP est utilisé dans un réseau *ad hoc* : *i-* la topologie change, les routes sont interrompus et le flux TCP subira un nombre important de timeout ce qui dégrade sa performance. *ii-* la performance du TCP dans un environnement *ad hoc* dépend fortement de la taille de la fenêtre de congestion utilisée. Plus la taille de la fenêtre augmente plus le nombre de paquet dans la route augmente et ils seront en compétition dans le médium partagé [Cordeiro et Agrawal (2006)].

Nous avons pris comme hypothèse que les nœuds sont homogènes : les nœuds possèdent la même énergie au départ et tous les nœuds ont le même rayon de transmission. Avec le modèle de mobilité que nous avons utilisé, RWP modifié, et avec la procédure de maintenance proposée, nous avons trouvé que la distribution des nœuds qui vont être dominants, et par la suite ceux qui vont dissiper plus d'énergie, est quasi uniforme. À long terme, tous les nœuds auront la même probabilité de devenirs dominants. Toutefois, il serait aussi intéressant de vérifier le cas où nous avons des nœuds hétérogènes : le rayon de transmission n'est pas constant.

ANNEXE A

Liste des publications

Journal

K. Mnif, M. Kadoch et R. Bo « Virtual Backbone Based on MCDS for Topology Control in Wireless Ad hoc Networks », *WSEAS Transactions on Communications*, Vol. 5, N°3. p. 395-400, mars 2006.

ABSTRACT

This paper proposes to use virtual backbone structure to handle control messages in *ad hoc* networks. This structure is effective in reducing the overhead of disseminating control information. In the first part, the approach to build the virtual backbone based on the Minimum Connected Dominating Set (MCDS) is presented. The novelty is in the way on computing the MCDS. A Linear Programming approach is used to find a Minimum Dominating Set (MDS). Then, a spanning tree of the MDS is computed to provide the MCDS. And, the performance of this approach is compared with other approaches. Simulation results show that our approach outperforms the others. In the second part, a theoretical analysis based on probabilistic approach is introduced to evaluate MCDS size. Finally, different techniques of diffusion in *ad hoc* networks are presented and compared. The flooding technique is simple and efficient, but it is expensive in term of bandwidth and causes excessive flows of messages. Simulation results show that technique using virtual backbone performs flooding and it is compared to MP relay.

Keywords: Minimum Connected Dominating Set, Virtual backbone, flooding

Conférences Internationales

K. Mnif and M. Kadoch « Virtual Backbone for Routing protocols Enhancement in Mobile Ad hoc Networks », présenté au *13th IEEE ICCS'06*. Singapore. 30 octobre -1 novembre 2006.

ABSTRACT

This paper proposes to use virtual backbone structure to handle control messages and reduce the effect of node mobility in mobile ad hoc networks. This structure is effective in reducing the overhead of disseminating control information and can be useful for routing protocols to enhance their performance. This paper presents a new approach to construct and maintain a

virtual backbone for ad hoc mobile networks. The construction of backbone is based on the Minimum Connected Dominating Set (MCDS). A distributed procedure is presented to maintain the backbone when the mobility is introduced. Terminals in ad hoc networks are free to move anywhere and anytime. Each terminal which changes position has to execute a maintenance procedure to connect to the backbone. Simulations results show the performance of this procedure when mobility and scalability is considered. To validate our approach, we used simulation to show that using backbone, routing protocols can enhance their performances in high dynamic environment. Three reactive protocols (AODV, DSR and TORA) have been used in this simulation.

K. Mnif and M. Kadoch « Performance Enhancement of Routing Protocols using Virtual Backbone in Mobile Ad hoc Networks », présenté au *8th OPNETWORKS'06.* Washington. 28-31 Août 2006.

ABSTRACT

Several routing protocols for mobile ad hoc networks (MANET) have been proposed in the last decade. Mobility and infrastructure-less represent two of several issues related to mobile ad hoc network. In order to enhance Ad hoc routing protocols, a virtual backbone can be used. This paper proposes to use virtual backbone to handle traffic control and hence reduce bandwidth consumption. The backbone enables routing protocols to use only a subset of nodes in the network for route management and avoid the use of flooding technique to broadcast control traffic. By using Opnet simulator, we present the performance evaluation of in constructing and maintaining the backbone with a new approach. The performance of routing protocols improvement based on the backbone is then compared to that of their standard versions.

Keywords: virtual backbone, routing protocols, ad hoc networks

K. Mnif, R. Bo et M. Kadoch « *Virtual Backbone Based on MCDS for Topology Control in Wireless Ad hoc Networks* », *2nd ACM PE-WASUN/MSWIM'05*, Montréal, p. 230-233, 10-13 Octobre, 2005.

ABSTRACT

This paper proposes to utilize virtual backbone to handle control messages in *ad hoc* networks. The virtual backbone is built by using the Minimum Connected Dominating Set (MCDS) on a graph. The first part of this paper presents a new algorithm to construct the MCDS. The construction of the MCDS is formulated using the linear programming approach. We compared the performance of this procedure with those other previous approaches, and we find that our approach is less complex and gives the nearest solution to the minimal one. The second part of this paper presents different techniques of diffusion in *ad hoc* networks such as flooding, clustering, MP relay, and backbone based on MCDS, etc. The flooding technique is simple and efficient, but it is expensive in term of bandwidth, and causes excessive flows of message etc. Simulation results show that the approach of virtual backbone based MCDS outperforms flooding and MP relay.

Keywords: Minimum Connected dominating Set, Integer programming, Virtual backbone.

K. Mnif, R. Bo et M. Kadoch « *A Distributed Approach for Computing the Minimum Dominating Connected Set for Efficient Routing in Ad hoc Networks* », IEEE *18th Canadian Conference on Electrical and Computer Engineering*, Saskatoon, p. 2065-2068, 1-5 mai 2005.

ABSTRACT

An optimized way of flooding packets, in *Ad hoc* networks, is to find the Minimum Connected Dominating Set (MCDS). Nodes belonging to the MCDS set are responsible for relaying messages, while other nodes are not. Most of previous methods, such as CDS-based and Weakly-CDS, have followed combinatorial approach or graph coloration technique to find an approximate solution. However, these methods are centralized. This paper proposes a new approach of employing integer linear program. To evaluate the performance of our approach, we compute the size of MCDS in a variety of graphs, Simulation results show that our approach has a very good performance in parameters such as size of MCDS and computing complexity compared with CDS-B and WCDS approaches.

Keywords: *Ad hoc* networks, Routing, Minimum Connected Dominating Set Integer Programming.

ANNEXE B

Compléments mathématiques

B.1 Résolution des problèmes en programmation linéaire de grande taille

La résolution d'un problème de programmation linéaire a été l'objet de plusieurs travaux de recherche pendant plusieurs décennies et de nombreuses méthodes ont été proposées à cette fin. Pendant 40 ans la méthode du simplex est demeurée la référence absolue des algorithmes de programmation linéaire. En 1979, Khachian a proposé un algorithme de programmation linéaire appelé méthode des ellipses de Khachian. Bien que cette méthode converge polynomialement en théorie, cet algorithme convergeait en pratique moins vite que le simplex [Glineur (1998)]. En 1984, Karmarkar a proposé une nouvelle recherche en présentant des algorithmes de points intérieurs à convergence polynomial et surpassant la méthode du simplex pour les problèmes d'optimisation linéaire de grande taille [Bazarra et al. (2005)]. Contrairement à la méthode du Simplex qui explore le polytope réalisable en suivant ses bords pour trouver l'optimum du problème, ces algorithmes produisent une suite d'itérées à l'intérieur de l'ensemble admissible. Ces méthodes ont été aussi été appliquées avec succès à la résolution de problèmes quadratiques de grande taille et très récemment à la résolution de problèmes non linéaires généraux.

Décrivons brièvement les méthodes de points intérieurs. Considérons le problème suivant :

$$(P)\begin{cases} \min \quad f(x) \\ c(x) \geq 0 \end{cases}$$

Ou f: $\Re^n \rightarrow \Re$ est la fonction à minimiser, $c(x) \geq 0$ signifie que les fonctions c_i: $\Re^n \rightarrow \Re$ $(1 \leq i \leq m)$, composantes de c doivent être positives à la solution. On suppose que les fonctions f et c_i $(1 \leq i \leq m)$ sont au moins deux fois différentiables et convexes. Le problème barrière associé au problème (P) est le suivant :

$$(P_u) \quad \min \varphi_u(x) = f(x) - \mu \log(c(x))$$

où φ_μ est la fonction barrière définie par Frisch [Glineur (1998)] et où μ appelé paramètre barrière, est strictement positif. Comme le logarithme n'est défini que pour des arguments strictement positifs, la fonction barrière logarithmique n'est définie que sur l'ensemble noté $\mathcal{F}^0$ des points strictement réalisables (c'est-à-dire vérifiant c(x) > 0). Par convexité de la fonction f, des fonctions c_i ($1 \leq i \leq m$), par stricte concavité et croissance de la fonction log, la fonction φ_μ est une fonction convexe.

Le théorème suivant présente les propriétés de la fonction barrière. Soit *opt*(*P*) l'ensemble des solutions de (*P*) et f^* la valeur optimale de f sur cet ensemble. Une preuve de ce théorème est présentée dans [Wright (1992)].

Théorème 1

Supposons que les fonctions f et c du problème (*P*) soient convexes et que l'ensemble des points strictement réalisables noté $\mathcal{F}^0 \neq \varnothing$. Soit $\{\mu_k\}$ une suite décroissante de paramètres positifs convergeant vers 0. Supposons que l'ensemble des solutions de (*P*) soit non vide et borné. On a alors les assertions suivantes :

(i) Pour tout $\mu > 0$, φ_μ est connexe dans $\mathcal{F}^0$ et possède un minimum noté x_μ (non nécessairement unique) sur $\mathcal{F}^0$. Tout minimum local x_μ est aussi un minimum global de φ_μ,

(ii) Toute suite de minima x_{μ_k} de φ_μ possède une sous-suite convergente, et tout point limite possible de telles suites est dans *opt*(*P*),

(iii) $f(x_{\mu_k}) \rightarrow f^*$ et $\varphi_\mu(x_{\mu_k}) \rightarrow f^*$, pour toute suite convergente de minima $\{x_{\mu_k}\}$.

Les méthodes des points intérieurs se déroulent comme suit. À partir d'une valeur initiale de μ, on résout de manière approchée le problème barrière $\{P_\mu\}$. On obtient alors un itère dans un voisinage d'une solution du problème $\{P_\mu\}$. Ensuite, on calcule un nouveau paramètre de perturbation $\mu > 0$ inférieur au précédent. On obtient alors un nouveau problème $\{P_\mu\}$ à résoudre. On résout ce dernier et on fait de nouveau décroître μ de telle manière que la suite de ces paramètres tende vers zéro. On va donc résoudre une suite de problèmes barrières à μ fixé de manière approchée jusqu'à l'obtention d'une solution du problème initial (P). Le théorème suivant assure, sous certaines conditions, la convergence de ces solutions approchées vers une solution de problème (P). Une preuve de ce théorème est présentée aussi dans [Wright (1992)].

Théorème 2

Supposons que l'ensemble $\mathcal{F}^0 \neq \varnothing$ et que x^* soit une solution de (P) pour laquelle les conditions d'optimalité de *Karush-Kuln-Tucker* soient satisfaisantes pour un multiplicateur λ^*, supposons aussi que les gradients des contraintes actives sont linéairement indépendantes et que les conditions de complémentarité stricte et du second ordre soient satisfaites en (x^*, λ). Considérons la suite des problèmes barrières P_{μ_k} ou $\{\mu_k\}$ est une suite de valeurs positives convergeant de façon monotone vers zéro quand $k \to \infty$. Alors :

(i) il existe une sous-suite de minima des fonctions barrières $\phi_{u_k}(x)$ convergeant vers x^*,

(ii) pour k suffisamment grand, le hessien $\nabla^2 \phi_{u_k}(x_k)$ est défini positif,

(iii) il existe une unique fonction continûment différentiable $x(\mu)$ des minima de $\phi_u(x)$ dans un voisinage de μ,

(iv) $\lim_{\mu \to 0} x(\mu) = x^*$.

La méthode des points intérieurs est largement implémentée dans différents logiciels qui sont conçus pour résoudre les problèmes de grande taille. Le logiciel Xpress-MP proposé par l'entreprise Dash Optimization, NOPTIQ (*Nonlinear Minimization Problems for Large Scale*) proposé par FIST (*France Innovation Scientifique et Transfert*) et CPLEX proposé par ILOG sont des exemples de logiciels capables de résoudre des problèmes de grande taille (le nombre des variables peut atteindre des dizaines de milliers de variables et de contraintes) qui utilisent la méthode des points intérieurs.

BIBLIOGRAPHIE

ADJIH, C., S. BOUDJIT, P. JACQUET, A. LAOUITI et P. MUHLETHALER(2005). « An Advanced Configuration and Duplicate Address Detection mechanism for a multi-interface OLSR Network » *INRIA Research Report RR-7547*. 34 pages, novembre.

ADJIH, C., P. JACQUET et L. VIENNOT (2005). « Computing Dominating Set with Multipoint Relays » *Ad Hoc and Sensor Wireless Networks*, p. 27-39, mars.

AHUJA, R.K., T.L. MAGNANTI et J.B. ORLIN (1993). *Network Flows: Theory, Algorithms, and Applications*, Prentice-Hall Inc., New Jersey.

ALZOUBI, K.M., P.-J. WAN et O. FRIEDER (2002). « New Distributed Algorithm for Connected Dominating Set in Wireless Ad Hoc Networks », *34th Annual Hawaii International Conference on System Sciences HICSS'02*, Hawaii, p. 3881-3887.

ARORA, S. et G. KARAKOSTAS (2000). « A 2+ε approximation algorithm for the *k*-MST problem », *11th annual ACM-SIAM symposium on Discrete algorithms*, CA, USA. p. 754-759, janvier.

BAKER D.J. et A. EPHREMIDES (1981). « The Architectural Organization of a Mobile Radio Network Via a Distributed Algorithm », *IEEE Transactions on Communications*, vol. 29, n° 11, p. 1694-1701.

BANGMAN, X., S. HISCHKE et B. WALKE (2003). « The Role of Ad Hoc Networking in Future Wireless Communications », *International Conference on Communication Technology Proceedings, 2003 (ICCT 2003)*, vol. 2, p. 1353-1358, 9-11 avril.

BASAGNI, S., I. CHLAMTAC, V.R. SYROTIUK et B.A. WOODWARD (1998). « A Distance Routing Effect Algorithm for Mobility (DREAM) », *ACM/IEEE Mobicom'98*, p. 76-84.

BASAGNI, S. (1999). « Distributed Clustering for Ad Hoc Networks », *International Symposium on Parallel Architectures, Algorithms and Networks (ISPAN '99)*, p. 310-315.

BAZARRA, M.S., J.J. JARVIS et H.D. SHERALI (2005). *Linear Programming and Network Flows*, Wiley-Interscience, Third Edition, New Jersey.

BENINI, L., G. CASTELLI, A. MACII et E. MACII (2000), « A Discrete-time Battery Model for High Level Power Estimation, Design, Automation and Test», *Proceedings of the Conference on Design, Automation and Test in Europe*, Paris,. p. 35-39.

BETTSTETTER, C., G. RESTA et P. SANTI (2003). « The Node Distribution of the Random Waypoint Mobility Model for Wireless Ad Hoc Networks », *IEEE Transaction on Mobile Computing* (*MobiHoc'03*), vol. 2, n° 3, p. 257-269, septembre.

BOUKERCHE, A. (2004). « Performance Evaluation of Routing Protocols for Ad Hoc Wireless Networks, Mobile Networks and Applications », *Mobile Networks and Applications*, vol. 9, n° 4, p. 333-342, août.

CALINESCU, G., I. MANDOIU, P.-J. WAN et A. ZELIKOVSKY (2004). « *Selecting Forwarding Neighbors in Wireless Ad Hoc Networks* », University of California Postprint n° 232, URL: http://repositories.cdlib.org/postprints/232.

CAMP, T., J. BOLENG, B. WILLIAMS, L. WILCOX et W. NAVIDI (2002). « Performance Comparison of Two Location Based Routing Protocols for Ad Hoc Networks », *Proceedings of INFOCOM'02*, vol. 3, p. 1678-1687.

CAPKUN, S., M. HAMDI et J.-P. HUBAUX (2003) « GPS-free positioning in mobile Ad-Hoc networks », *Proceedings of the 34th Hawaii International Conference on System Sciences – 2003*, p. 1-10, janvier.

CARUSO ANTONIO, STEFANO CHESSA, SWADES DE, ALESSANDRO URPI (2005) « GPS Free Coordinate Assignment and Routing in Wireless Sensor Networks », INFOCOM'05. Miami, Mars.

CHELIUS, G. et E. FLEURY (2002). « Ananas: A New Ad Hoc Network Architecture», *INRIA Research Report RR-4354*. Janvier.

CHEN, Y.P. et A.L. LIESTMAN (2002). « Approximating Minimum Size Weakly-Connected Dominating Sets for Clustering Mobile Ad Hoc Networks », *Third ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc'02)*, p. 165-172, juin.

CLAUSEN, T. (2004). « Comparative Study of Routing Protocols for Mobile Ad Hoc Networks », Rapport de recherche n° 5135. *INRIA*, 25 pages, septembre.

CORDEIRO C. M. et D. P. Agrawal (2006) *Ad hoc Networks and Sensor Networks : Theory and Application,* World Scientific Publishing Co.N.J.

CORMEN, T., C. LEISERSON et R. RIVEST (2002). *Introduction à l'algorithmique*, 2e éd., Cazin, Dunod, Paris.

CORSON, M.S. et J. MACKER (1999). « Mobile Ad Hoc Networking (MANET) Routing Protocol Performance Issues and Evaluation Considerations », *IETF*, RFC, n° 2501.

DAS, B. et V. BHARGHAVAN (1997). « Routing in Ad Hoc Networks Using Minimum Connected Dominating Sets », *International Conference on Communications (ICC'97)*, Montréal, p. 376–380, juin.

DIMITRI, M. et G. Dudek (2006). « Probabilistic Self-Localization for Sensor Networks », *American Association for Artificial Intelligence*, 6 pages.

GLINEUR, F. (1998). « Interior-Point Methods for Linear Programming: A Guided Tour », *Belgian Journal of Operations Research, Statistics and Computer Science*, 28 pages.

HAAS, Z.J. (1997). « The Routing Algorithm for the Reconfigurable Wireless Networks », *Proceeding of IEEE International Conference on Universal Personal Communications (ICUPC'97)*, San Diego (CA), vol. 2, p. 562-566, octobre.

HEUSE, M., F. ROUSSEAU, G. BERGER-SABBATEL et A. DUDA (2003). « Performance Anomaly of 802.11b », *Proceedings of INFOCOM'03*, San Francisco (CA), vol. 3, p. 836–843, avril.

HOU, T. et V.O.K. LI (1986). « Transmission Range Control in Multihop Packet Radio Networks », *IEEE Transactions on Communications*, vol. 34, n°1, p. 38-44, janvier.

HU, L. (1993). « Topology Control for Multihop Packet Radio Networks », *IEEE Transactions on Communications*, vol. 41, n° 10, p. 1474-1781.

HUANG, Z., C.C. SHEN, C. SCRISATHAPORNPHAT et C. JAIKAEO (2002). « Topology Control for Ad Hoc Networks with Directional Antennas », *IEEE 11th Conference on Computer Communications and Networks*, Miami (FL), p. 16-21, mai.

JACQUET, P., A. LAOUITI, P. MINET, et L. VIENNOT (2002) « Performance of multipoint relaying in ad hoc mobile routing protocols » *Networking 2002*, Pise (Italy) pp 387-398.

JACQUET, P. et T. CLAUSEN (2003). « Optimized Link State Routing Proocol », *IETF*, RFC, n° 3626.

JOHANSSON, P., T. LARSSON, N. HEDMAN, B. MIELCZAREK et M. DEGERMARK (1999). « Scenario-Based Performance Analysis of Routing Protocols for Mobile Ad Hoc Networks », *Proceedings of ACM/IEEE MOBICOM'99*, Seattle (WA), p. 195-206.

JOHNSON, D.B. et D.A. MALTZ (1996). « Dynamic Source Routing in Ad Hoc Wireless Networks », *Mobile Computing*, Kluwer Academic Publishers, p. 153-181.

KAWADIA, V. et P.R. KUMAR (2003). « Power Control and Clustering in Ad Hoc Networks », *IEEE INFOCOM'03*, vol. 1, p. 459-469, mars.

KO, Y. et N.H. VAIDYA (1998). « Location-Aided Routing in Mobile Ad Hoc Networks », *Proceeding of ACM MOBICOM'98*, p. 66-76.

KRISHNA, P., N. VAIDYA, M. CHATTERJEE et D. PRADHAN (1997). « A Cluster-based Approach for Routing in Dynamic Networks », *ACM SIGCOMM Computer Communication*, p. 49-65.

LASSOUS I.G. (2004) «*Les Réseeaux Ad hoc*» Chapitre du livre RÉSEAUX MOBILES ET SANS FIL, Hermes Science Publications. p 163-188.

LI, L., J.Y. HALPERN, P. BAHL, Y.M. WANG et R. WATTENHOFER (2001). « Analysis of a Cone-based Distributed Topology Control Algorithms for Wireless Multi-Hop Networks », *ACM Symposium Principle of Distributed Computing*.

LIM, H. et C. KIM (2000). « Multicast Tree Construction and Flooding in Wireless Ad Hoc Networks », *3rd ACM International Workshop on Modeling, Analysis and Simulation of Wireless and Mobile Systems'00*, p.61-68, avril.

LLOYD, E.L., R. LIU, M.V. MARATHE, R. RAMANATHAN et S.S. RAVI (2002). « Algorithmic Aspects of Topology Control Problems for Ad Hoc Networks », *IEEE Mobile Ad Hoc Networking and Computing* (*MOBIHOC'02*), p. 304-308.

LUNDGREN, H., E. NORDSTROM et C. TSCHUDIN (2002). « The Gray Zone Problem in IEEE 802.11b Based Ad Hoc Networks », *Proceedings of the ACM SIGMOBILE Mobile Computing and Communications*, vol. 6, n°3, p.104-105.

MALPANI, M., J. WELCH et VAIDYA N. (2000) « Leader Election Algorithms for Mobile Ad hoc Networks », *4th International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications*, p. 96-103.

MARSAN, M.A., C.F. CHIASSERINI, A. NUCCI, G. CARELLO et L.D. GIOVANNI (2002). « Optimizing the Topology of Bluetooth Wireless Personal Area Networks », *IEEE INFOCOM'02*, p. 435-441.

MEGIDDO, N. (1987). « On the Complexity of Linear Programming », *Advances in Economic Theory: Fifth World Congress*, T. Bewley (éd.), Cambridge University Press, p. 225–268.

MNIF, K., R. BO et M. KADOCH (2005a). « A Distributed Approach for Computing the Minimum Connected Dominating Set in Ad Hoc Networks », *IEEE 18th Canadian Conference on Electrical and Computer Engineering* (*CCECE'05*), Saskatoon, p. 2065-2068, mai.

MNIF, K., R. BO et M. KADOCH (2005b). « Virtual Backbone Based on MCDS for Topology Control in Wireless Ad Hoc Networks », *2nd ACM Performance Evaluation of Wireless Ad Hoc, Sensor, and Ubiquitous Networks (WASUN/MSWIM'05)*, Montreal, p. 230-233, octobre.

MNIF K., M. KADOCH et R. BO (2006). « Virtual Backbone Based on MCDS for Topology Control in Wireless Ad Hoc Networks », *WSEAS Transactions on Communications*, vol. 5, n° 3, p. 395-400, mars.

MNIF, K. et M. KADOCH (2006a). « Performance Enhancement Routing Protocols of using Virtual Backbone in Mobile Ad hoc Networks », presented at *OPNETWORKS'06*, Washington, 28-31 Août.

MNIF, K. et M. KADOCH (2006b). « Construction and Maintenance of Virtual Backbone in Mobile Ad Hoc Networks », accepted in the 13th *IEEE ICCS'06*, Singapore, 5 pages, novembre.

MURTHY, C.S.R. et B.S. MANOJ (2004). *Ad Hoc Wireless Networks: Architecture and Protocols*, Prentice Hall.

MURTHY, S. et J.J. GARCIA-LUNA-ACEVES (1996). « An Efficient Routing Protocol for Wireless Networks », *Journal ACM Mobile Networks and Applications*, Special Issue on Routing in Mobile Communications Networks, vol. 1, n° 2, p. 183-197.

NESGARI, S. et R. PRAKASH. (2002). « ManetConf : Configuration of Hosts in a Mobile Ad Hoc Network », *IEEE INFOCOM*, vol. 2, 2002, p. 1059-1068, juin.

NICULESCU D. et B. NATH. (2003). « Ad-hoc positioning system », In the Proceedings of IEEE Global Communication Conference (Globcom'03), p. 2926-2931, novembre.

OGIER, R., F. TEMPLIN et M. LEWIS (2004). « Topology Dissemination Based on Reverse-Path Forwarding (TBRPF) », *IETF*, RFC n° 3684, février.

PARK, V. et S. CORSON (2001). « Temporally -Ordered Routing Algorithm (TORA) – Version 1 », draft-ietf-manet-tora-spec-03.txt, work in progress.

PARK, V.D. et M.S. CORSON (1998). « A Performance Comparison of the Temporally-ordered Routing Algorithm and Ideal Link-state Routing », *3rd IEEE Symposium on Computers and Communications ISCC'98*, p. 592-598.

PARUCHURI, V.S., A. DURRESI, D.S. DASH et R. JAIN (2002). Optimal Flooding Protocol for Routing in Ad Hoc Networks », Technical report, Ohio State University.

PERKINS, C. et P. BHAGWAT (1994). « Highly Dynamic Destination-sequenced Distance-vector (DSDV) Routing for Mobile Computers », *Proceedings of ACM SIGCOMM'94*, p. 234-244.

PERKINS, C. E., E.M. ROYER, S.R. DAS et M.K. MARINA (2001). « Performance Comparison of Two On-demand Routing Protocols for Ad Hoc Networks », *IEEE Wireless Communications*, vol. 8, n° 1, p. 16-28.

PERKINS, C., J. MALINEN, R. WALIKAWA and E. M. ROYER (2001) « IP Address Autocon-figuration for Ad Hoc Networks », Internet Draft : draft-ietf-manet-autoconf-01.txt, November 2001.

PERKINS, E., E. ROYER et S. DAS (2003). « Ad Hoc On-demand Distance Vector (AODV) Routing », *IETF*, RFC, n° 3561.

PUJOLLE, G. (2003). *Les réseaux*, 4e éd., Eyrolles, Paris.

QAYYUM, A., L. VIENNOT et A. LAOUITI (2002). « Multipoint Relaying for Flooding Broadcast Messages in Mobile Wireless Networks », *Proceedings of the 35th Annual Hawaii International Conference*, p. 3866-3875, 7-10 janvier.

RAMANATHAN, R. et R. ROSALES-HAIN (2000). « Topology Control of Multihop Wireless Networks Using Transmit Power Adjustment », *INFOCOM'00*, p. 404-413, mars.

RAPPAPORT, T.S. (1996). *Wireless Communications: Principles and Practices*, 2nd ed., Prentice Hall PTR, New Jersey.

RUBIN, I. et P. VINCENT (2001). « Topological Synthesis of Mobile Backbone Networks for Managing Ad Hoc Wireless Networks », *IFIP/IEEE International Conference on Management of Multimedia Networks and Services'01*, Chicago (IL), p. 215-221, novembre.

SASSON Y., D. CAVIN et A. SCHIPER (2002) « Probabilistic Broadcast for Flooding in Wireless Mobile Ad hoc Networks », *Technical Report IC/2002/54* EPFL.

SESAY, S., Z. YANG, B. QI et J. HE (2004). « Simulation Comparison of Four Wireless Ad Hoc Routing Protocols », *Information Technology Journal*, vol. 3, n° 3, p. 219-226.

SINGH, S., M. WOO et C.S. RAGHAVENDRA (1998). « Power-aware Routing in Mobile Ad Hoc Networks », *ACM MOBICOM'98*, Dallas (TX), p. 181-190, octobre.

SINHA, P., R. SIVAKUMAR et V. BHAGHAVAN (1999). « CEDAR: A Core Extraction Distributed Ad Hoc Routing Algorithm », *IEEE Journal of Selected Areas in Communications (JSAC)*, vol. 17, n° 8, p. 1454-1466.

SINHA, P., R. SIVAKUMAR et V. BHARGHVAN (2001). « Enhancing Ad Hoc Routing with Dynamic Virtual Infrastructure », *Proceedings of INFOCOM'01*, Alaska, vol. 3, p.1763-1772, avril.

SLIJEPCEVIC, S. et M. POTKONJAK (2001). « Power Efficient Organization of Wireless Sensor Networks », *IEEE International Conference on Communications ICC'01*, Helsinki, juillet.

SPIEGEL M.R. et J. LIU (2000). *Formules et tables de mathématiques*, 2e éd., McGraw-Hill International, Royaume-Uni.

SRIVASTAVA, S. et R.K. GHOSH (2002). « Cluster Based Routing Using a K-tree Core Backbone for Mobile Ad Hoc Networks », *Proceedings of the 6th International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications'02*, Atlanta (GA), p. 14-23, 28 septembre.

STALLINGS, W. (2005). *Réseaux et communications sans fil*, Pearson Education, Paris.

STOJMENOVIC, I., M. SEDDIGH et J. ZUNIC (2001). « Internal Nodes Based Broadcasting in Wireless Networks », *Proceedings of the 34th Annual Hawaii International Conference on System Science*).

TOH, C.K. (1997). « Associativity-based Routing for Ad Hoc Mobile Networks », *Wireless Personal Communications*, vol. 4, n° 2, p. 1-36.

TSENG, Y.-C., S.-Y. NI, Y.-S. CHEN et J.-P. SHEU (2002). « The Broadcast Storm Problem in Mobile Ad Hoc Networks », *Wireless Networks*, vol. 8, p. 153-167.

VIVIER, E. (2004) « Architecture et protocoles des réseaux GSM, GRPS, UMTS » chapitre du livre *Réseaux sans fils et mobiles*, Editions Lavoisier, p. 21-55.

YOON, J., M. LIU et B. NOBLE (2003). « Waypoint Considered Harmful », *Proceedings of INFOCOM'03*, p. 1312-1321.

WAN, P.-J., G. CALINESCU, X. LI et O. FREIDER (2001). « Minimum-energy Broadcast Routing in Static Ad Hoc Wireless Networks », *Proceedings of INFOCOM(2001)*, p. 1162-1171.

WATTENHOFER, R., L. LI, P. BAHL et Y.-M. WANG (2001). « Distributed Topology Control for Power Efficient Operation in Multihop Wireless Ad Hoc Networks », *Proceedings of INFOCOM'01*,

WIESELTHER, J.E., G.D. NGUYEN et A. EPHREMIDES (2000). « On the Construction of Energy-efficeint Broadcast and Multicast Tress in Wireless Networks », *Proceedings of INFOCOM'00*, p. 585-594.

WU J. et H. LI (1999). « On Calculating Connected Dominating Set for Efficient Routing in Ad Hoc Wireless Networks », *3rd International Workshop on Discrete Algorithms and Methods for MOBILE Computing and Communications*, Seattle (WA), p. 7-14.

WU J. et H. LI (2001). « A Dominating Set Based Routing Scheme in Ad Hoc Wireless Networks », *Telecommunications Systems (2001)*, p. 13-36.

WU J. et W. LOU (2004). « Extended Multipoint Relays to Determine Connected Dominating Sets in MANETs », *IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks (SECON'04)*, Santa Clara (CA), p. 621-630, octobre.

WRIGHT, M.H. (1992). « Interior Methods for Constrained Optimization », *Acta Numerica'92*, vol. 1, p. 341-407.

YOUSSEF, A., YOUNIS M., et AGRAWALA A. (2005). « Accurate Anchor-Free Node Localization in Wireless Sensor Networks », IEEE Workshop on Information Assurance in Wireless Sensor Networks (WSNIA), Pheonix, AZ. avril 2005

ZHAO, Y.J., R. GOVIDAN et D. ESTRAIN (2002). « Residual Energy Scans for Monitoring Wireless Sensor Networks », *IEEE Wireless Communications and Networking Conference*, Orlando (FL), mai.