

Titre: Routage hybride avec qualité de service pour réseaux mobiles ad hoc
Title: hoc

Auteur: Amine Alaoui
Author:

Date: 2005

Type: Mémoire ou thèse / Dissertation or Thesis

Référence: Alaoui, A. (2005). Routage hybride avec qualité de service pour réseaux mobiles ad hoc [Mémoire de maîtrise, École Polytechnique de Montréal]. PolyPublie.
Citation: <https://publications.polymtl.ca/7582/>

Document en libre accès dans PolyPublie

Open Access document in PolyPublie

URL de PolyPublie: <https://publications.polymtl.ca/7582/>
PolyPublie URL:

Directeurs de recherche: Alejandro Quintero
Advisors:

Programme: Non spécifié
Program:

UNIVERSITÉ DE MONTRÉAL

ROUTAGE HYBRIDE AVEC QUALITÉ DE SERVICE
POUR RÉSEAUX MOBILES AD HOC

AMINE ALAOUI

DÉPARTEMENT DE GÉNIE INFORMATIQUE
ÉCOLE POLYTECHNIQUE DE MONTRÉAL

MÉMOIRE PRÉSENTÉ EN VUE DE L'OBTENTION
DU DIPLÔME DE MAÎTRISE ÈS SCIENCES APPLIQUÉES
(GÉNIE INFORMATIQUE)

AOÛT 2005



Library and
Archives Canada

Published Heritage
Branch

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque et
Archives Canada

Direction du
Patrimoine de l'édition

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file *Votre référence*
ISBN: 978-0-494-16748-9

Our file *Notre référence*
ISBN: 978-0-494-16748-9

NOTICE:

The author has granted a non-exclusive license allowing Library and Archives Canada to reproduce, publish, archive, preserve, conserve, communicate to the public by telecommunication or on the Internet, loan, distribute and sell theses worldwide, for commercial or non-commercial purposes, in microform, paper, electronic and/or any other formats.

The author retains copyright ownership and moral rights in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

In compliance with the Canadian Privacy Act some supporting forms may have been removed from this thesis.

While these forms may be included in the document page count, their removal does not represent any loss of content from the thesis.

AVIS:

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque et Archives Canada de reproduire, publier, archiver, sauvegarder, conserver, transmettre au public par télécommunication ou par l'Internet, prêter, distribuer et vendre des thèses partout dans le monde, à des fins commerciales ou autres, sur support microforme, papier, électronique et/ou autres formats.

L'auteur conserve la propriété du droit d'auteur et des droits moraux qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

Conformément à la loi canadienne sur la protection de la vie privée, quelques formulaires secondaires ont été enlevés de cette thèse.

Bien que ces formulaires aient inclus dans la pagination, il n'y aura aucun contenu manquant.

**
Canada

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Ce mémoire intitulé:

ROUTAGE HYBRIDE AVEC QUALITÉ DE SERVICE
POUR RÉSEAUX MOBILES AD HOC

présenté par: ALAoui Amine

en vue de l'obtention du diplôme de: Maîtrise ès sciences appliquées
a été dûment accepté par le jury d'examen constitué de:

M. BOUCHENEK, Hanifa, Doctorat, présidente

M. QUINTERO, Alejandro, Doct., membre et directeur de recherche

M. NICOLESCU, Gabriela, Doct., membre

REMERCIEMENTS

J'aimerais en premier lieu remercier M. Alejandro Quintero, mon directeur de recherche, pour son encadrement, son écoute et sa compréhension.

Je tiens également à remercier M. Samuel Pierre, directeur du Laboratoire de Recherche en Réseautique et en Informatique Mobile à l'École Polytechnique de Montréal pour tout ce que j'y ai appris.

Un merci particulier à mes amis du LARIM : Nabil, Gabriel, Abdellatif et Lotfi pour leurs conseils et leurs encouragements. Je tiens aussi à remercier Khaled, Sabine, Abdelhamid et Ali ainsi que l'ensemble des membres du laboratoire pour l'ambiance d'entraide qui y règne.

J'aimerais aussi remercier mon oncle Abdallah pour son soutien sans faille.

Un grand merci à M. Patrick Hétu, mon superviseur au travail, pour ses encouragements et son aide.

Mes derniers remerciements mais non les moindres vont à ceux à qui je dois tout : mes parents ainsi qu'à mes deux frères Youssef et Badr. Sans eux, je n'aurais rien pu faire !

RÉSUMÉ

Grâce à la flexibilité qu'ils offrent aux usagers, les réseaux mobiles sont en train de devenir de plus en plus populaires auprès des utilisateurs.

Bien que mobiles, ces réseaux ont généralement des composantes fixes qui gèrent et coordonnent les communications. Il existe cependant, des réseaux mobiles sans aucune infrastructure fixe. Dans ces réseaux sans fils appelés *ad hoc* ou MANET, les unités mobiles jouent à la fois les rôles d'émetteur / récepteur et celui de routeur.

Les techniques de routage dans ces réseaux peuvent être classées en trois catégories : proactives, réactives et hybrides. Dans la première, les différents nœuds s'échangent périodiquement les tables de routage de façon à ce que toute unité mobile source sache comment joindre un nœud destination à tout moment. Pour ce qui est de la seconde catégorie, un nœud source ne cherche la route vers un nœud destination que si il y a une demande de connexion entre ces deux nœuds. Le protocole ZRP appartient quant à lui à la dernière catégorie et tente de trouver un juste équilibre entre les avantages de la première (efficacité et rapidité) et de la seconde (minimum de signalisation) en limitant l'échange périodique des tables de routage pour chaque nœud à l'intérieur d'une zone de voisins de ρ sauts.

Par ailleurs, les applications informatiques sont de plus en plus exigeantes en matière de qualité de service (VoIP, vidéo ...). Et si les problèmes reliés à celle-ci sont en cours de résolution pour les réseaux filaires, l'intégration de la qualité de service pour les réseaux ad hoc demeure un défi. En effet, cela est principalement relié aux caractéristiques intrinsèques des réseaux MANET (bande passante limitée, contraintes d'énergie, topologie changeante...).

Dans ce travail, nous proposons un protocole de routage hybride qui prend en charge deux critères de la qualité de service : le délai et la bande passante. L'idée de base est de maintenir en plus de la zone de routage *initiale* de ZRP, une zone de routage étendue qui elle est formée à partir des contraintes de qualité de service. Les unités mobiles tenteront

ainsi de garder en mémoire les routes les plus performantes en matière de qualité de service de manière à les utiliser plus rapidement au besoin.

De plus, notre protocole intègre un mécanisme de réparation de routes lequel deviendra actif si au cours d'une connexion, la route devient désuète suite à la rupture d'un lien.

La validité de notre solution peut être confirmée par les résultats encourageants obtenus suite à la simulation.

En effet, si dans la plupart des cas les performances de notre protocole sont comparables à ceux de ZRP, dans le cas des réseaux relativement denses (une centaine de nœuds), nos résultats surpassent clairement à la fois ceux des protocoles proactifs et réactifs.

ABSTRACT

Because of their flexibility, mobile networks are becoming more and more popular between users. In general, this networks functions with fixed infrastructure. The role of this infrastructure is to coordinate communications between different mobile nodes.

However, Ad hoc networks (MANETs) are a kind of mobile wireless networks that works without any fixed infrastructure. In this context, the mobile nodes have to assume by themselves the process of routing.

We have three categories of routings protocols for MANETs : proactive, reactive and hybrid. In the first case, mobile nodes always maintain a route to all destinations. In contrast, reactive protocols find a route only when it is needed to deliver a message. Finally, hybrid protocols like ZRP try to combine the benefits of proactive protocols (effectiveness, little delay) and reactive ones (low control traffic). In fact, ZRP limits the scope of proactive procedure to node local neighbourhood zone (ρ hops).

On the other hand, for more and more applications quality of service is necessary (VoIP, video ...). However, because of the characteristics of ad hoc networks (limited bandwidth, energy constraints, changing topology), providing quality of service for MANETs is a real challenge.

In this work, we present a hybrid routing protocol with quality of service support for ad hoc networks (delay and bandwidth). The main idea of this protocol is to add an extended routing zone to ZRP initial routing zone. This extended zone will be formed specially to help quality of service support. Thus, the mobile node will try to keep also in memory good routes in term of quality of service.

Moreover, our protocol includes a route repair mechanism which will be active if the route between a source and destination break during communication.

We can confirm the validity of our solution by the encouraging results. In fact, the simulation shows that performance of our protocol exceeds at the same time those of proactive and reactive protocols especially for dense networks (more than 100 nodes).

TABLE DES MATIÈRES

REMERCIEMENTS	iv
RÉSUMÉ	v
ABSTRACT	vii
TABLE DES MATIÈRES	ix
LISTE DES FIGURES.....	xii
LISTE DES TABLEAUX.....	xv
LISTE DES SIGLES ET ABRÉVIATIONS	xvi
CHAPITRE I NTRODUCTION	1
1.1 Concepts et définitions	2
1.2 Description de la problématique	3
1.3 Buts et objectifs de la recherche.....	5
1.4 Plan du mémoire	6
CHAPITRE II ROUTAGE ET QUALITÉ DE SERVICES DANS LES RÉSEAUX AD-HOC.....	7
2.1 Protocoles de routage dans les réseaux ad hoc.....	8
2.1.1 Routage Plat	10
2.1.2 Routage hiérarchique	22
2.2 Routage et qualité de service	32
2.2.1 Modèles de qualités de service	33
2.2.2 Mécanismes de niveau 2	36

2.2.3 Routage avec qualité de service	38
2.2.4 Protocoles de signalisation.....	46
2.3 Réparation de routes	48
2.3.1 Re-routage global	48
2.3.2 Re-routage partiel.....	49
2.3.3 Re-routage partiel tenant compte de la densité des chemins.....	49
2.3.4 Réparation de route localisée (LRR).....	50

CHAPITRE III PROTOCOLE DE ROUTAGE

HYBRIDE AVEC QUALITÉ DE SERVICE.....	52
3.1 Le protocole hybride de routage ZRP	53
3.2 Protocole de routage hybride avec qualité de service	56
3.2.1 Routage avec contrainte de délai.....	56
3.2.2 Routage avec contrainte de bande passante	66
3.3 Réparation de route avec HPQS.....	76

CHAPITRE IV IMPLÉMENTATION, SIMULATION

ET ANALYSE DE PERFORMANCE	79
4.1 Environnement de simulation	79
4.1.1 L'environnement de simulation	80
4.1.2 Présentation du simulateur Qualnet	81
4.2 Implémentation : structures de données et algorithmes	83
4.2.1 Implémentation de HPQS-D	83
4.2.2 Implémentation de HPQS-BP	89
4.3 Plan d'expérience et scénarios	94
4.4 Résultats et analyse	96

4.4.1 Délai moyen bout-en-bout.....	97
4.4.2 Variation du délai.....	102
4.4.3 Débit.....	106
4.4.4 Surcharge de signalisation	110
4.4.4 Synthèse des résultats.....	113
CHAPITRE V CONCLUSION.....	114
5.1 Synthèse des travaux et originalité des contributions	114
5.2 Limitations des travaux.....	116
5.3 Travaux futurs	117
BIBLIOGRAPHIE	118

LISTE DES FIGURES

Figure 1.1 Réseau ad hoc mobile	2
Figure 2.1 MOBILE IP	9
Figure 2.2 Protocoles de routage Ad hoc	10
Figure 2.3 Le protocole OLSR.....	13
Figure 2.4 Le protocole DSR	19
Figure 2.5 Le Protocole CGSR	24
Figure 2.6 Le Protocole HSR.....	26
Figure 2.7 Niveaux de détection de requêtes ZRP.....	30
Figure 2.8 Diffusion sélective ZRP.....	31
Figure 2.9 Le protocole Ticket Based Protocol	43
Figure 2.10 Réparation de route LRR	50
Figure 3.1 Protocole ZRP zone de routage du nœud A.....	53
Figure 3.2 Composantes de ZRP.....	55
Figure 3.4 Algorithme HPQS-D	61
Figure 3.5 Exemple HPQS-D (Topologie globale du réseau).....	63
Figure 3.6 Exemple HPQS-D (Zones du nœud A)	63
Figure 3.7 Exemple HPQS-D (Zones du nœud F)	64
Figure 3.8 Exemple HPQS-D (Zones du nœud O)	65
Figure 3.9 Exemple HPQS-D (Zones du nœud D)	65
Figure 3.10 Zones de routages initiale et étendue du nœud A (HPQS-BP).....	69
Figure 3.11 Algorithme HPQS-BP	71
Figure 3.12 Exemple HPQS-BP (Topologie globale du réseau).....	73
Figure 3.13 Exemple HPQS-BP (Zones du nœud A)	74
Figure 3.14 Exemple HPQS-BP (Zones du nœud I).....	75
Figure 3.15 Exemple HPQS-BP (Zones du nœud J).....	75
Figure 3.16 Réparation de route HPQS	78

Figure 4.1 Structure de métrique IARP pour HPQS-D.....	84
Figure 4.2 Structure de la table de routage IARP pour HPQS-D	84
Figure 4.3 Structure de la table des états des liens IARP pour HPQS-D.....	85
Figure 4.4 Structure de l'en-tête des paquets IARP pour HPQS-D	85
Figure 4.5 Structure générale IARP pour HPQS-D	86
Figure 4.6 Structure de la table de routage IERP pour HPQS-D	86
Figure 4.7 Structure de l'en-tête des paquets IERP pour HPQS-D.....	87
Figure 4.8 Structure générale de IERP pour HPQS-D.....	87
Figure 4.9 Structure de métrique IARP pour HPQS-BP.....	89
Figure 4.10 Structure de la table de routage IARP pour HPQS-BP	89
Figure 4.11 Structure de la table des états des liens IARP pour HPQS-BP.....	90
Figure 4.12 Structure de l'en-tête des paquets IARP pour HPQS-BP	90
Figure 4.13 Structure générale de IARP pour HPQS-BP	91
Figure 4.14 Structure de la table de routage IERP pour HPQS-BP	92
Figure 4.15 Structure de l'en-tête des paquets IERP pour HPQS-BP.....	92
Figure 4.16 Structure générale IERP pour HPQS-BP.....	93
Figure 4.17 Délai bout-en-bout (30 nœuds)	98
Figure 4.18 Délai bout-en-bout (50 nœuds)	99
Figure 4.19 Délai bout-en-bout (80 nœuds)	100
Figure 4.20 Délai bout-en-bout (120 nœuds)	101
Figure 4.21 Variation du délai (30 nœuds)	103
Figure 4.22 Variation du délai (50 nœuds)	104
Figure 4.23 Variation du délai (80 nœuds)	105
Figure 4.24 Variation du délai (120 nœuds)	106
Figure 4.25 Débit (30 nœuds)	107
Figure 4.26 Débit (50 nœuds)	108
Figure 4.27 Débit (80 nœuds)	108
Figure 4.28 Débit (120 nœuds)	109
Figure 4.29 Surcharge de signalisation (absence de mobilité).....	110

Figure 4.30 Surcharge de signalisation ($V = 10 \text{ m/s}$)	111
Figure 4.31 Surcharge de signalisation ($V = 30 \text{ m/s}$)	112
Figure 4.32 Surcharge de signalisation ($V = 70 \text{ m/s}$)	112

LISTE DES TABLEAUX

Tableau 2.1 Comparaison entre les protocoles proactifs	17
Tableau 2.2 Comparaison entre les protocoles réactifs.....	21
Tableau 2.3 Comparaison entre les protocoles hiéarchiques	31
Tableau 2.4 Comparaison entre MACA / PR et IEEE 802.11 avec QdS.....	38
Tableau 2.5 Comparaison des protocoles de routage avec QdS	46
Tableau 3.1 Exemple de table de routage HPQS-D.....	59
Tableau 3.2 Exemple de table de routage HPQS-BP	70
Tableau 4.1 Spécifications de l'environnement de simulation	81
Tableau 4.2 Plan d'expérience	96

LISTE DES SIGLES ET ABRÉVIATIONS

AODV	Ad hoc On-demand Distance Vector
BF	Bellman-Ford
BRP	Broadcast Resolution Protocol
CEDAR	Core-Extraction Distributed Ad hoc Routing
CGSR	Clusterhead Gateway Switch Routing
CTS	Clear-To-Send
DCF	Distributed Coordination Function
DIFS	Distributed Inter-Frame Space
DiffServ	Differentiated Services
DSDV	Destination Sequenced Distance Vector
DSR	Dynamic Source Routing
FQMM	Flexible QoS Model for MANETs
HPQS	Hybrid Protocol With Quality of Service
IARP	Intrazone Routing Protocol
IEEE	Institute of Electrical and Electronics Engineers
IERP	Interzone Routing Protocol
IETF	Internet Engineering Task Force
IntServ	Integrated Services
IP	Internet Protocol
LAR	Location-Aided Routing
MAC	Medium Access Control
MACA	Multiple Access Collision Avoidance
MACA/PR	MACA with Piggyback Reservation
MANET	Mobile Ad hoc NETwork
PDA	Personal Digital Assistant

QdS	Qualité de Service
RAM	Random Access Memory
RERR	Route ERRob
RFC	Request For Comment
RREP	Route REPl
RREQ	Route REQu
RSVP	Resource reSerVation Protocol
RTS	Request-to-Send
SIFS	Short Inter-Frame Space
TBP	Ticket-Based Probing
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
ZRP	Zone Routing protocol

CHAPITRE I

INTRODUCTION

De nos jours, les télécommunications jouent un rôle de plus en plus important. Les contextes d'utilisation des réseaux n'ont jamais été aussi variés ainsi que les besoins des usagers. Et si la variété des environnements où on utilise les technologies de communication suppose une certaine flexibilité au niveau de l'architecture utilisée, les besoins grandissant des usagers impliquent le souci d'intégrer des contraintes supplémentaires dans la conception des réseaux en vue de satisfaire ces mêmes besoins. Les réseaux mobiles ad hoc constituent par ailleurs un bon exemple d'une structure flexible qui peut s'adapter à divers environnements. En effet, contrairement aux réseaux traditionnels, il s'agit de réseaux où tous les nœuds sont mobiles et communiquent sans aucune infrastructure fixe.

S'agissant des besoins des utilisateurs, il va sans dire que les utilisations possibles des réseaux étant variés (application multimédia, vidéoconférence, voix sur IP, etc..), le concept du *best-effort* sur lequel repose Internet devient inadéquat. Ce concept consiste en effet à l'utilisation optimale des liens du réseau en dépit de l'introduction de tout délai supplémentaire. Pour pallier ce problème, il devient impératif que les réseaux intègrent la qualité de service.

Le présent mémoire s'intéresse à l'intégration de la qualité de service dans le fonctionnement des réseaux mobiles ad hoc. Si plusieurs solutions ont été proposées pour les réseaux filaires classiques, l'intégration de la QoS pour les réseaux ad hoc constitue aujourd'hui un défi.

Dans ce premier chapitre, nous définirons dans un premier temps les principales notions traitées dans ce travail. Dans un second temps, nous présenterons les problèmes entourant l'intégration de la qualité de service dans les réseaux ad hoc. Nous présenterons en fin de chapitre le plan du mémoire.

1.1 Concepts et définitions

Contrairement aux réseaux mobiles traditionnels tels que les réseaux cellulaires, les réseaux ad hoc sont des réseaux sans fil et sans aucune infrastructure fixe. Un tel réseau est en effet constitué d'unités mobiles se déplaçant dans un environnement donné et n'utilisant que le médium sans fil pour communiquer. Elles ne font donc appel à aucune infrastructure fixe et ne sont gérées par aucune entité centrale. Ainsi, dans un réseau ad hoc, deux unités mobiles hors portée l'une par rapport à l'autre communiquent via des nœuds mobiles intermédiaires, lesquels jouent le rôle de routeurs.

On représente généralement un réseau ad hoc par un graphe $G_t = (V_t, E_t)$ où V_t modélise l'ensemble des unités mobiles qui forment le réseau à l'instant t et E_t l'ensemble des liens existant dans le réseau au même instant.

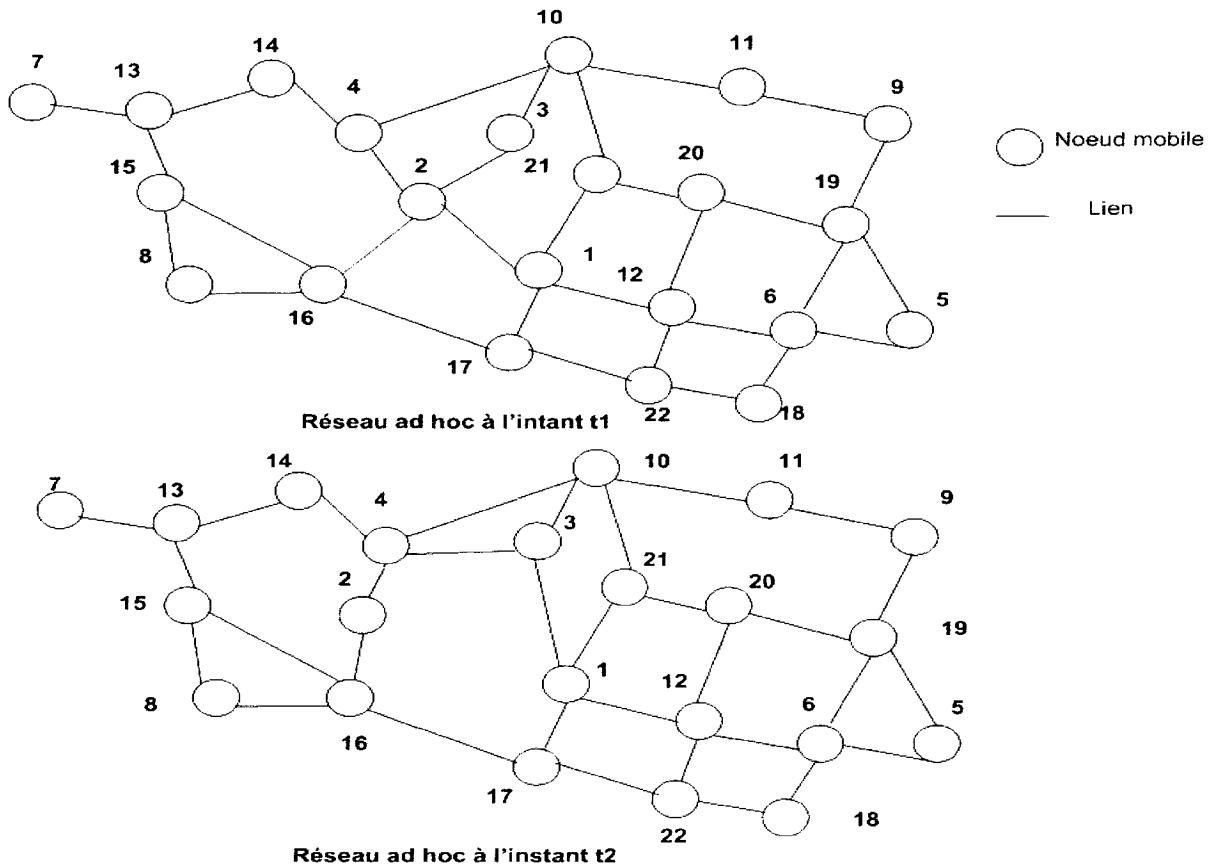


Figure 1.1 Réseau ad hoc mobile

La figure 1.1 présente un réseau ad hoc composé de 22 nœuds à deux instants distincts t_1 et t_2 . Entre ces deux moments, le nœud mobile 2 s'est déplacé provoquant ainsi la rupture du lien 2->3. Néanmoins, la communication reste possible entre ces deux unités à l'instant t_2 à travers le nœud 4 qui jouera le rôle de nœud intermédiaire. Ainsi, on dira que les nœuds 2 et 3 dans le réseau ad hoc communiquent à travers la route 2->4->3. De même, les nœuds 8 et 18 peuvent communiquer via la route 8->16->17->22->18. Plus rigoureusement, on définit une *route* dans un réseau ad hoc comme étant l'ensemble des nœuds mobiles contribuant à l'acheminement des données de la source à la destination. Ainsi, le *routage* est le processus permettant de découvrir et de sélectionner une ou plusieurs routes entre deux nœuds mobiles.

Par ailleurs, on entend par *qualité de service* l'ensemble des caractéristiques, des contraintes ou des exigences auxquelles doit satisfaire la connexion entre une source et une destination données afin de répondre aux besoins de l'application. Ces exigences se rapportent généralement à la bande passante ou au délai bout-en-bout. Il peut s'agir aussi de gigue, de taux de perte de paquets ou d'autres contraintes du même type.

Ainsi, le routage avec qualité de service pour les réseaux ad hoc consiste non seulement à trouver une route possible entre la source et la destination mais il faut aussi que la route trouvée vérifie les contraintes de QoS.

À cela s'ajoute le concept de réparation et maintenance de route. En effet, dans un réseau ad hoc une route peut devenir invalide. C'est le cas notamment quand un lien est rompu à cause de la mobilité d'un des nœuds qui formait la route. Dans un tel cas, il faut réparer la route en trouvant une alternative qui assure la continuité de la connexion.

1.2 Description de la problématique :

Garantir une certaine qualité de service pour une transmission entre deux nœuds mobiles dans un réseau ad hoc n'est pas facile. En effet, l'atteinte de cet objectif peut être compromise par de nombreuses difficultés. Celles-ci sont généralement liées à la nature même des réseaux ad hoc :

- Le caractère dynamique de la topologie : Cette contrainte est due à la mobilité des nœuds. En effet, les nœuds mobiles peuvent se déplacer arbitrairement et de façon rapide dans un territoire donné. Par conséquent, les liens existants à un instant donné peuvent disparaître;
- L'inexistence d'infrastructure fixe : Comme nous l'avons mentionné précédemment cette caractéristique distingue les réseaux ad hoc des autres réseaux mobiles. Ainsi, les nœuds mobiles sont appelés à jouer à la fois les rôles d'hôte et de routeur;
- Contrainte de bande passante : De part leur nature, les réseaux ad hoc mobiles ont une bande passante est limitée. Garantir la QoS est donc plus difficile que pour les réseaux filaires;
- Contraintes énergétiques : Les unités mobiles doivent être autonomes en matière d'énergie. Elles sont donc alimentées par des sources d'énergie portables tels que les batteries. Par conséquent, le fonctionnement des nœuds mobiles est sujet à la contrainte d'énergie imposée par ces mêmes sources;
- Contraintes de sécurité : Étant donné que les réseaux ad hoc mobiles utilisent l'interface sans fil, il va sans dire que la sécurité est plus difficile à garantir que dans un contexte de réseau filaire. Plus encore, le fait que les paquets transitent par des nœuds intermédiaires rend l'atteinte de cet objectif encore plus ardu d'autant plus que l'introduction de mécanisme de codage n'est pas conseillé dans un contexte où la bande passante est limitée;
- Réseau hétérogène : Un réseau ad hoc est très rarement homogène. Les nœuds n'ont donc pas tous les mêmes capacités (mémoire, temps de calcul ...). Les mécanismes conçus pour ces réseaux doivent donc prendre cette caractéristique en considération.

Les aspects précédents montrent des caractéristiques intrinsèques des réseaux ad hoc mobiles. L'introduction de contraintes de qualité de service ajoute un nouveau niveau de

difficulté au problème. Il est donc nécessaire de réétudier les mécanismes de QoS pour les réseaux filaires dans le but de voir comment on pourrait les adapter au contexte des réseaux mobiles ad hoc.

Ainsi, il faudra en premier lieu définir un modèle de QoS compatible avec les réseaux ad hoc. Tout comme *IntServ* et *DiffServ* pour les réseaux filaires classiques, celui-ci devra présenter les services fournis et indiquer les mécanismes qui seront mis en place pour les assurer.

De plus, l'étude devra prendre en considération la particularité de la couche MAC pour les réseaux sans-fil. En effet, cette couche est essentielle pour la QoS. Cela dit, la modélisation actuelle du médium radio ne tient pas compte de certaines particularités (telle que l'asymétrie des liens).

Par ailleurs, il est primordial que la solution bout-en-bout suggérée assure la compatibilité des réseaux ad hoc avec les autres catégories de réseaux. En effet, les différents réseaux sont de plus en plus appelés à communiquer ensemble.

1.3 Buts et objectifs de la recherche :

Le principal but de la présente recherche est de proposer un mécanisme de routage adapté aux réseaux ad hoc mobiles permettant la communication entre les différents nœuds du réseau tout en tenant compte de la qualité de service. Ce mécanisme devra aussi inclure une procédure de réparation et de maintenance de routes dans le réseau. Pour ce faire, nous nous proposons de :

- ➔ Présenter, étudier, comparer et critiquer les différents protocoles et mécanismes présents dans la littérature;
- ➔ Concevoir et détailler le mécanisme proposé notamment en ce qui concerne les algorithmes, stratégies et systèmes utilisés;
- ➔ Implémenter ce mécanisme et comparer les résultats obtenus avec les autres protocoles présents dans la littérature en ce qui a trait à la performance.

1.4 Plan du mémoire :

En plus du présent chapitre, ce mémoire comprend quatre autres.

Ainsi, dans le second chapitre, nous présenterons un état de l'art du sujet. Nous mettrons l'emphase sur la comparaison des différents mécanismes et protocoles de routage sans et avec qualité de service en vue de souligner les principales faiblesses de chaque mécanisme.

Dans le chapitre trois, nous détaillerons la solution que nous proposons pour la prise en charge de la qualité de service lors du processus de routage dans les réseaux ad hoc. Ainsi, nous préciserons les algorithmes et les protocoles auxquels fera appel la solution suggérée.

Dans le chapitre quatre, nous présenterons l'implémentation de notre solution ainsi qu'une analyse de performance relative à cette dernière. Pour ce faire, nous utiliserons des outils de simulation qui nous permettront de comparer notre solution aux différents mécanismes étudiés dans le chapitre deux.

Nous achèverons enfin avec une conclusion dans laquelle nous synthétiserons la contribution apportée dans le cadre de cette maîtrise et proposerons des pistes pour rechercher d'autres solutions éventuelles.

CHAPITRE II

ROUTAGE ET QUALITÉ DE SERVICES DANS LES RÉSEAUX AD-HOC

De plus en plus présents dans des contextes très variés, les réseaux téléinformatiques sont appelés à devenir plus flexibles. Actuellement, les réseaux mobiles sans-fil tentent d'élargir cette notion en permettant aux utilisateurs de se déplacer tout en conservant leur connectivité réseau.

Cette flexibilité s'étend cependant bien au-delà de la simple mobilité des usagers. En effet, les réseaux mobiles doivent désormais non seulement gérer la mobilité, mais aussi, de plus en plus, supporter bons nombres d'applications qui exigent une certaine qualité de service (délai et/ou bande passante).

Plus que dans les réseaux mobiles avec infrastructure fixe, où des stations de base jouent le rôle de routeurs acheminant les paquets vers nœuds mobiles, les réseaux sans infrastructure fixe, dits *ad hoc* ou MANET, de part leur nature rendent la satisfaction ces contraintes particulièrement complexe. En effet, dans ces réseaux, les nœuds mobiles jouent à la fois le rôle de stations mais aussi celui de routeurs intermédiaires à travers lesquels doit transiter l'information. De plus, la nature mobile de ces réseaux fait en sorte que leur topologie change d'une manière difficilement prévisible avec le temps. Ainsi, le routage des paquets de la source vers la destination à travers les nœuds mobiles voisins doit se faire selon un protocole qui tient compte de la topologie changeante du réseau. De plus, des liens établis lors du routage initial peuvent devenir invalides au cours d'acheminement à cause de la mobilité d'un ou de plusieurs nœuds qui formaient la route. Dans ce cas un re-routage sera nécessaire afin d'acheminer les paquets restant à destination.

À ces contraintes viennent s'ajouter d'autres. Ainsi, les nœuds du réseau doivent être autonomes en matière d'énergie. Et il est difficile d'assurer la sécurité des réseaux MANET d'une part, parce qu'il s'agit de réseaux sans-fil et d'autre part parce qu'il est

difficile d'interdire aux nœuds intermédiaires d'accéder à l'information qu'ils acheminent.

Cependant, l'avantage d'utiliser un réseau ad hoc par rapport à un réseau mobile avec infrastructure réside principalement dans le fait que le premier ne nécessite pas d'infrastructure, que son installation soit rapide et qu'il requiert un minimum d'investissement.

L'objet du présent chapitre est de présenter les différents protocoles, algorithmes et travaux qui traitent du sujet. Nous commencerons par présenter les protocoles ou algorithmes de routage les plus répandus existants dans la littérature. Après quoi, nous nous attarderons à présenter quelques travaux qui traitent de la qualité de service dans les réseaux ad hoc. Finalement, nous nous intéresserons au problème de re-routage dans ce type de réseaux.

2.1 Protocoles de routage dans les réseaux ad hoc

Dans les réseaux mobiles avec infrastructure fixe, un nœud mobile source s qui veut communiquer avec un nœud destination d commence par acheminer les paquets vers la station de base fixe à laquelle il est relié. Celle-ci par un protocole bien déterminé, se charge d'acheminer l'information vers d .

C'est le cas par exemple du protocole MOBILE-IP [1], où tout nœud mobile possède deux adresses IP. L'une appelée *home address* est statique et l'identifie à l'intérieur de son réseau domicile. L'autre dite *care-of address* est dynamique et change au fur et à mesure que le nœud mobile se déplace. De plus, chaque nœud mobile possède un *home agent* (HA) et possiblement un *foreign agent* (FA) dont le rôle est d'acheminer les paquets vers le nœud mobile selon sa localisation.

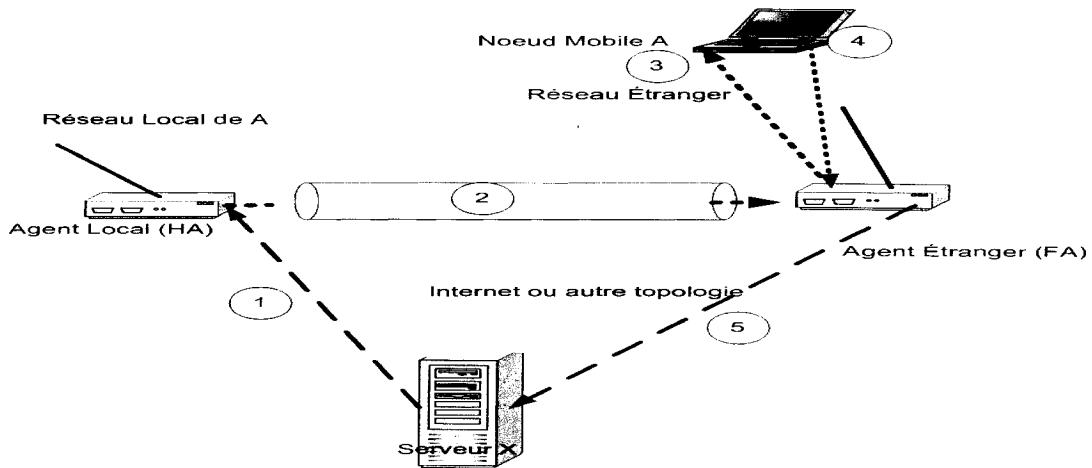


Figure 2.1 MOBILE IP

Dans la figure 2.1, tout échange entre le nœud mobile A et un hôte donné (serveur X dans le cas présent) a lieu selon les étapes suivantes :

1. Le serveur X transmet un datagramme IP destiné au noeud mobile A avec l'adresse domicile (*home address*) de A dans l'en-tête IP. Ainsi, le datagramme IP sera routé vers le réseau domicile de A;
2. Une fois arrivé dans ce réseau, le datagramme IP est intercepté par le HA. Celui-ci encapsule le datagramme dans un nouveau paquet IP avec la *care-of address* de A en en-tête. Ainsi, le paquet pourra être envoyé vers le réseau étranger;
3. Le FA intercepte le datagramme et élimine l'en-tête IP puis délivre le datagramme initial ainsi récupéré au noeud mobile A;
4. Quand le noeud A envoie un paquet IP au serveur X, il utilise l'adresse IP de ce dernier. Dans cet exemple, le serveur X n'est pas mobile son adresse est donc fixe. Les datagrammes IP envoyés de A vers le serveur X seront routé par le réseau étranger;
5. Les datagrammes IP seront acheminés de A vers X à travers Internet.

Ainsi, les nœuds mobiles n'ont pas à gérer eux-mêmes l'information relative au routage. Le cas des réseaux ad hoc est différent. En effet, en absence de routeurs fixes ou de stations de base, les nœuds mobiles de ces réseaux doivent eux-mêmes se charger du

processus de routage et d'acheminement de l'information. Pour ce faire, plusieurs protocoles adaptés aux réseaux ad hoc mobiles ont été proposés. Tel qu'illustré dans la figure 2.2, on s'accorde généralement pour classer les approches de routages en trois grandes catégories [2]:

- Routage plat : lui-même sous divisé en deux catégories : proactif et réactif;
- Routage hiérarchique;
- Routage basé sur la position géographique.

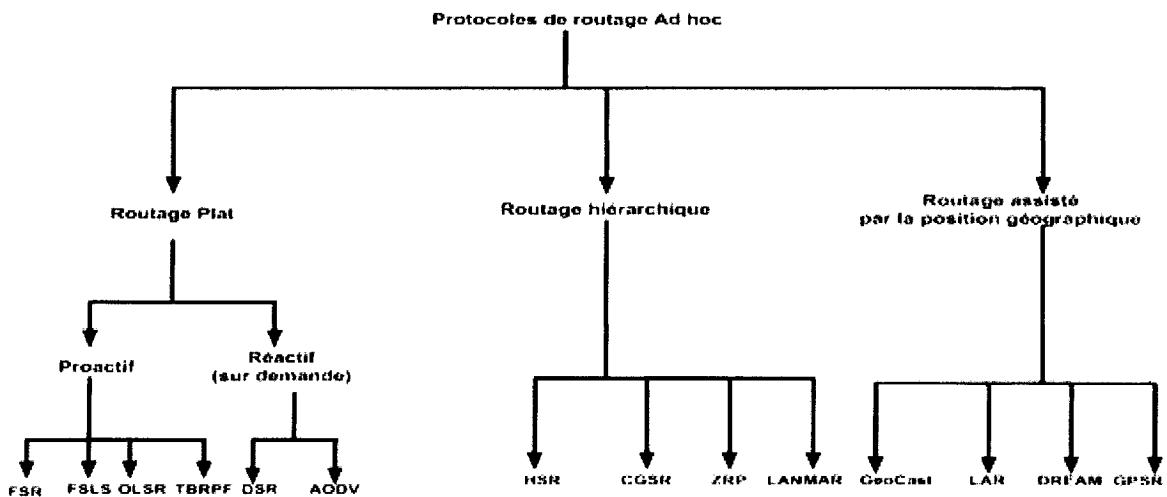


Figure 2.2 Protocoles de routage Ad hoc

Dans le routage plat, tous les nœuds ont la même importance et jouent des rôles similaires. Par contre, dans les protocoles de routage hiérarchique, on assigne à certains nœuds des rôles particuliers dans le réseau. Enfin, le routage géographique s'appuie sur le GPS qui permet à la source de connaître la position de la destination avec une précision raisonnable.

Dans ce chapitre cependant, nous nous attarderons uniquement sur les deux premières catégories de routage.

2.1.1 Routage Plat

Comme mentionné précédemment, le routage plat se subdivise en deux principales catégories : proactive et réactive. Le routage proactif tend à mettre à jour sur

une base régulière les routes entre toutes les paires de nœuds formant le réseau. La plupart des protocoles de cette catégorie sont inspirés des protocoles basés sur les états des liens dans les réseaux filaires.

Les protocoles réactifs, qu'à eux, aussi appelés protocoles de routage sur demande, constituent une nouvelle approche de routage inexistante dans les réseaux fixes. En effet, ceux-ci diffèrent des protocoles de routage conventionnel en ce sens qu'ils ne sont actifs que lors des communications.

Routage proactif

La recherche de routes dans ces protocoles se fait de manière continue. En effet, les nœuds du réseau échangent de manière régulière l'information (tables) de routage et ce, qu'il ait des requêtes de communication ou non. En outre, les protocoles proactifs ont plusieurs propriétés intéressantes particulièrement pour les applications qui exigent une certaine qualité de service ou les applications temps réel. Il s'agit de protocoles où pour chaque paire de nœuds dans le réseau, une route est disponible presque à chaque instant.

Dans ce qui suit, nous présenterons le fonctionnement des protocoles proactifs les plus répandus.

Le protocole OLSR

Comme son nom l'indique : *Optimized Link State Protocol*, agit comme protocole à état de lien. Il convient particulièrement pour les réseaux ad hoc denses (une centaine de nœuds ou plus). Ainsi, plus le réseau est dense, plus l'usage d'OLSR est approprié.

Le protocole OLSR maintient la topologie globale du réseau au niveau de chaque nœud. Pour ce faire, les différents nœuds échangent périodiquement de l'information concernant la topologie du réseau. De plus, OLSR utilise les MPRs (ou relais multipoint) pour réduire non seulement le nombre de diffusions (*broadcast*) mais aussi la taille des

paquets de contrôle utilisés. Les MPRs sont des nœuds particuliers choisis pour diffuser les messages durant le processus de *flooding*.

Le protocole OLSR fonctionne comme suit : tout nœud N du réseau ad hoc diffuse périodiquement à ses voisins immédiats des paquets *HELLO* afin d'échanger l'information concernant le voisinage. Pour identifier les relais multipoint, le nœud N détermine le minimum de voisins qui lui permettront de rejoindre tous les nœuds qui sont à une distance de deux sauts. Les nœuds qui satisfont cette condition seront les MPRs de N. Cependant, à l'échelle globale du réseau, déterminer le nombre de MPRs minimum pour chaque nœud est un problème NP-difficile. Pour pallier ce problème, des heuristiques efficaces sont utilisées (telle que l'heuristique de calcul des MPRs proposée dans [3]).

Une fois ses MPRs déterminés, tout nœud du réseau ad hoc en informe ses voisins dans des paquets *HELLO*. À la réception de ces paquets, chaque nœud enregistre le nœud MPR à partir duquel il pourra joindre l'émetteur.

Il convient aussi de noter les différences qui existent entre OLSR et un protocole classique basé sur l'état des liens. En effet, dans OLSR c'est uniquement les MPRs d'un nœud N qui transmettent la mise du jour de l'état de liens de N. De plus, la mise à jour de l'état des liens de N est réduite pour inclure uniquement les nœuds dont N est un MPR. Ainsi, une topologie partielle est propagée à travers le réseau et par conséquent, N ne pourra être joint qu'à partir de l'un de ses MPRs. C'est donc à partir de cette topologie partielle qu'OLSR arrive à trouver une route pour une destination donnée.

L'utilisation des MPRs représente donc un avantage certain puisque la diffusion périodique des paquets de contrôle ne passe que par des noeuds particuliers évitant ainsi le plus possible la surcharge du réseau. De plus, l'utilisation d'heuristiques pour déterminer les MPRs nous permet de nous rapprocher le plus possible des routes optimales. Cependant, OLSR demeure inadéquat dans le cas des réseaux peu denses. En effet, le choix des MPRs sera très restreint dans un tel cas.

La figure 2.3 présente un exemple de réseau ad hoc où le protocole OLSR est utilisé. Dans cette figure les noeuds E,F et G sont les MPRs de A qui lui permettent de communiquer avec les autres noeuds du réseau.

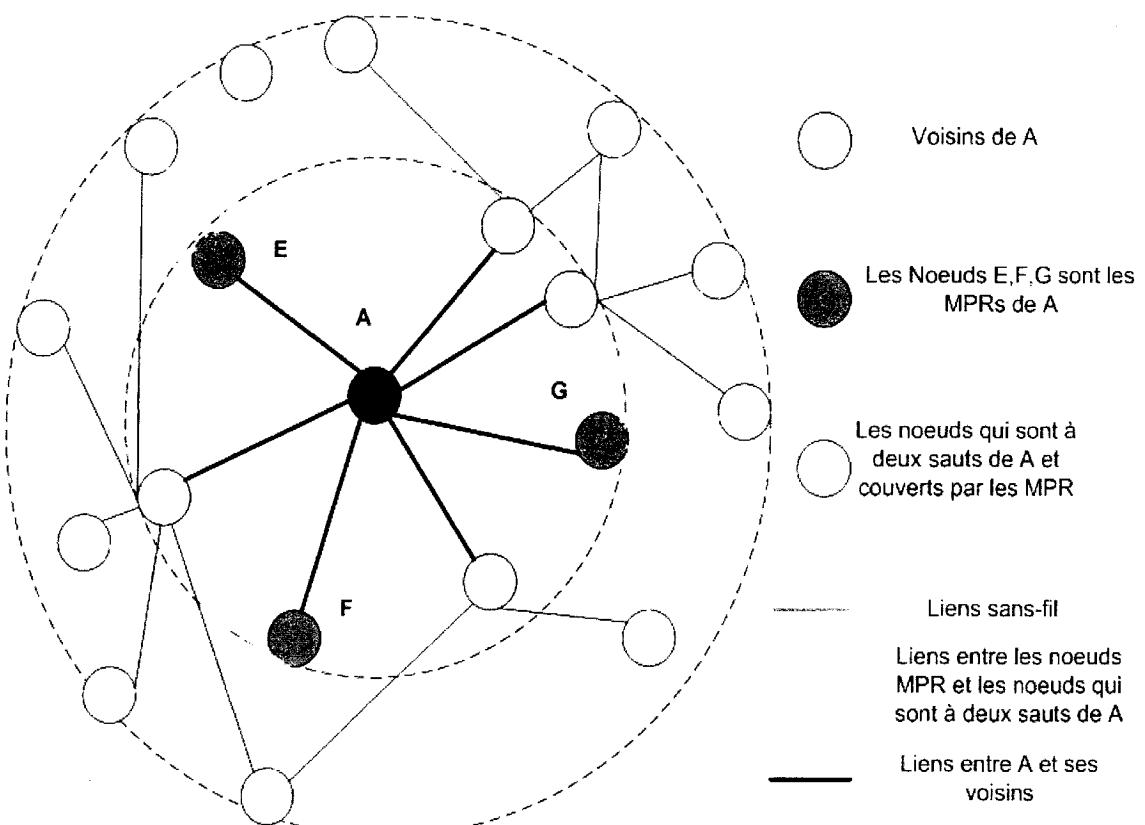


Figure 2.3 Le protocole OLSR

Le protocole FSR

Comme OLSR, *Fisheye State Routing* [4] est aussi un protocole basé sur l'état des liens. FSR conserve la topologie du réseau au niveau de chaque nœud et propage la mise à jour de l'état des liens à travers le réseau. Cependant, FSR intègre les particularités suivantes :

- Tout nœud échange l'information complète relative à son état des liens uniquement avec ses voisins, évitant ainsi la propagation jusqu'aux nœuds

éloignés. Les états de liens sont donc mis à jour à partir de l'information envoyée par les voisins;

- L'échange périodique de l'état des liens peut aussi se faire à la suite d'un événement, pour éviter ainsi un grand nombre de mises à jour qui peuvent avoir lieu suite à un lien brisé à cause de la mobilité d'un nœud par exemple;
- Les diffusions (*broadcast*) périodiques de l'information relative à l'état des liens se font selon différentes fréquences. La diffusion se fait selon une fréquence plus importante pour les entrées qui correspondent à des destinations proches. Et au contraire, plus la destination est éloignée, plus la diffusion se fait selon une fréquence faible. Par conséquent, les nœuds mobiles voisins ont une connaissance des états de lien plus récente que les nœuds éloignés.

Ainsi, FSR limite le surdébit de contrôle dans le réseau comparativement au protocole classique d'états de liens. Néanmoins, son principal désavantage réside dans le fait que les nœuds ont une faible connaissance des segments qui leur sont éloignés.

Le protocole TBRPF

« *Topology Broadcast on Reverse Path Forwarding* » [5] est aussi un protocole basé sur les états des liens. Il inclut les deux modules suivants :

- Le module de découverte des voisins (TND) qui opère à travers l'envoi de paquets *HELLO* qui ne contiennent que les changements survenus tels la perte ou la détection d'un voisin (information différentielle);
- Le modèle de routage qui fonctionne en se basant sur une l'information de topologie partielle obtenue grâce aux mises à jour périodiques et différentielles (contenant les changements uniquement).

En outre, TBRPF offre la possibilité d'opérer en utilisant la topologie globale du réseau en ajoutant de l'information additionnelle dans les mises à jour.

TBRPF fonctionne comme suit : soit s la source des mises à jour. Chaque nœud i dans le réseau choisit un nœud p qui est son prochain saut vers s selon la route avec un nombre minimum de sauts. De cette manière se constitue un arbre qui se compose des différentes

routes qui relient tous les nœuds i à la source s . Ainsi, au lieu de surcharger le réseau en entier, TBRPF propage les mises à jour des états de liens uniquement dans la direction opposée de l'arbre formé des routes à sauts minimum, c'est-à-dire de la source s vers les différents nœuds du réseau. Donc, chaque nœud i n'accepte les mises à jour de s que si elles proviennent de p . Plus encore, seuls les liens qui appartiennent à l'arbre et qui ont subis des changements seront inclus dans la mise à jour.

Ainsi, TBRF est bien adapté aux réseaux ad hoc dont le changement de topologie est rapide et tente d'éviter de propager inutilement l'information de mise à jour en générant un minimum de surdébit de routage et en utilisant des paquets de mise à jour plus petits qu'un algorithme à état de liens traditionnel. Cependant, le calcul de l'arbre peut devenir fastidieux pour des réseaux moyennement complexes.

Le protocole DSDV

Contrairement aux protocoles précédents, DSDV [6] (ou *Dynamic Destination-Sequenced Distance-Vector*) est un protocole proactif basé sur le vecteur de distance. Il s'agit d'un protocole basé sur l'algorithme de Bellman-Ford où chaque nœud du réseau peut communiquer avec n'importe quelle autre station mobile en utilisant sa table de routage. Celle-ci contient le nombre de sauts nécessaires pour atteindre une destination donnée et un numéro de séquence permettant de différencier les anciennes routes des routes plus récentes. Cette table est alimentée par les données provenant des nœuds voisins. En effet, chaque nœud envoie la mise à jour de sa table de routage à tous ses voisins directs. La mise à jour se fait périodiquement ou suite à un événement tel que l'apparition ou la disparition d'un nœud mobile. De cette manière, le nœud source sera, en principe, toujours en mesure de localiser la destination. La mise à jour peut se faire au complet (envoie de toute l'information concernant le routage) ou d'une façon incrémentale (avis des changements par rapport à la mise à jour précédente uniquement). Plus le réseau est stable, plus on utilisera la mise à jour incrémentale. Et au contraire, dans le cas d'un réseau où la topologie change très fréquemment, c'est-à-dire que les

changements surviennent plus rapidement que parviennent les mises à jour, on privilégiera la mise à jour complète.

Lors de la mise à jour, l'émetteur envoie un paquet contenant l'adresse des nœuds destination, le nombre de sauts vers chaque destination, le numéro de séquence incrémenté et le plus récent numéro de séquence connu pour chaque destination. Chaque nœud qui reçoit le paquet de mise à jour compare les numéros des séquences dans sa table de routage avec ceux du paquet pour vérifier si il doit ou non procéder à une mise à jour. Ainsi, il écrase les routes obsolètes par les nouvelles, s'il y a lieu. Si deux routes ont le même numéro de séquence, on ne considère que celle avec la meilleure (plus petite) métrique. Généralement, la métrique correspond au nombre de nœuds dans le chemin. Dès qu'une route est choisie, on incrémente la métrique qui lui correspond. Naturellement, on assigne à un lien brisé ou rompu une très grande valeur (infinie).

Le protocole DSDV a tendance, de part sa pro activité à générer un trafic de routage important, d'où la lenteur de ce protocole qui constitue sa principale faiblesse. Cependant, le fait que la source puisse localiser préalablement la destination constitue l'avantage de DSDV.

Le tableau 2.1 ci-dessous compare les protocoles proactifs où N est le nombre de nœuds mobiles dans le réseau. Notons également que la complexité de mise en mémoire mesure l'ordre de grandeur des tables de routage utilisées par les différents protocoles alors que la complexité de communication calcule le nombre de messages nécessaires lors d'une mise à jour.

Tableau 2.1 Comparaison entre les protocoles proactifs

	FSR	OLSR	TBRPF	DSDV
Type de routage	Proactif, État des liens	Proactif, État des liens	Proactif, État des liens	Proactif, Vecteur de distance
Métrique utilisée	Plus court Chemin	Plus court chemin	Plus court chemin	Plus court chemin
Fréquence des mises à jour	Périodiquement	Périodiquement	Périodiquement	Périodiquement ou suite à un événement
Nombres de séquence utilisés	Oui	Oui	Oui	Oui
Sans boucle	Oui	Oui	Oui	Oui
Pire cas existe	Non	Oui (État des liens pur)	Non	Non
Chemins multiples supportés	Oui	Non	Non	Non
Complexité de mise en mémoire	$O(N)$	$O(N)$	$O(N)$	$O(N)$
Complexité de communication	$O(N)$	$O(N)$	$O(N)$	$O(N)$

Routage réactif

Très utilisé dans les réseaux ad hoc, le routage réactif tend à minimiser le surdébit dû au routage. En effet, l'idée de base des protocoles réactifs est de ne chercher une route entre deux nœuds n et m que quand il existe une demande de communication entre ces deux nœuds.

Il existe plusieurs protocoles réactifs tels que *Ad Hoc On Demand Distance Vector Routing* (AODV), *Associativity-Based Routing* (ABR), *Dynamic Source Routing* (DSR), *Temporally Ordered Routing Algorithms* (TORA). Parmi tous ces protocoles, le groupe de travail MANET de l'IETF considère que AODV et DSR sont les véritables candidats

pour une standardisation éventuelle. C'est la raison pour laquelle, nous nous limiterons à présenter ces deux protocoles dans la présente section.

Le protocole DSR

S'agissant d'un protocole réactif, le Routage à Source Dynamique [7] détermine la séquence de nœuds par lesquels transitera l'information de la source jusqu'à la destination et ce, uniquement suite à la réception par la source s d'une demande de connexion vers un nœud destination d . Quand il reçoit une telle requête, s déclenche le processus de découverte de la route vers le nœud d . Ce processus permet au réseau ad hoc de découvrir dynamiquement une route vers d . Il se déroule comme suit :

Le nœud s initie l'opération en propageant un paquet de requête de route qui comprend l'adresse de la destination. Ce paquet est composé également de l'adresse de s , un champ de route pour accumuler les nœuds transitoires visités et un numéro de route. Quand un tel paquet est reçu par nœud intermédiaire du réseau, trois cas sont possibles :

1. Si le nœud a déjà reçu une requête provenant de s avec le même identificateur ou si l'adresse de ce même nœud est présente dans le champ du chemin, alors le traitement de cette requête a déjà été effectué. Le paquet est alors ignoré empêchant ainsi la création de boucles.
2. Si le nœud qui reçoit le paquet est d alors le chemin a déjà été trouvé auquel cas on envoie le chemin dans un paquet réponse à s .
3. Si aucune des deux situations précédentes ne se présente alors le nœud ajoute son adresse dans le champ correspondant et diffuse la requête à ses voisins.

À la fin du processus de découverte, s reçoit un paquet de réponse de route qui a été envoyé par d et qui contient la séquence de nœuds par lesquels va transiter l'information jusqu'à la destination d . Notons que le paquet de réponse a emprunté le chemin inverse de la route trouvée, en supposant que les liens sont bidirectionnels.

Dans la figure 2.4 ci-dessous, le nœud source A recherche une route vers la destination E. Le nœud A commence alors par propager la requête vers E. Au début, le paquet de requête transite par les nœuds B et C. Ces derniers retransmettent la requête après avoir

ajouté leurs adresses dans le champ approprié. Les nœuds suivants (le nœud D dans ce cas) en font de même jusqu'à la destination. Le processus de construction de route est terminé quand la requête parvient au nœud destination E. Celui-ci utilise le chemin inverse à celui spécifié dans le champs de route pour signifier à A qu'une route est désormais disponible (cela presuppose que les liens sont bidirectionnels).

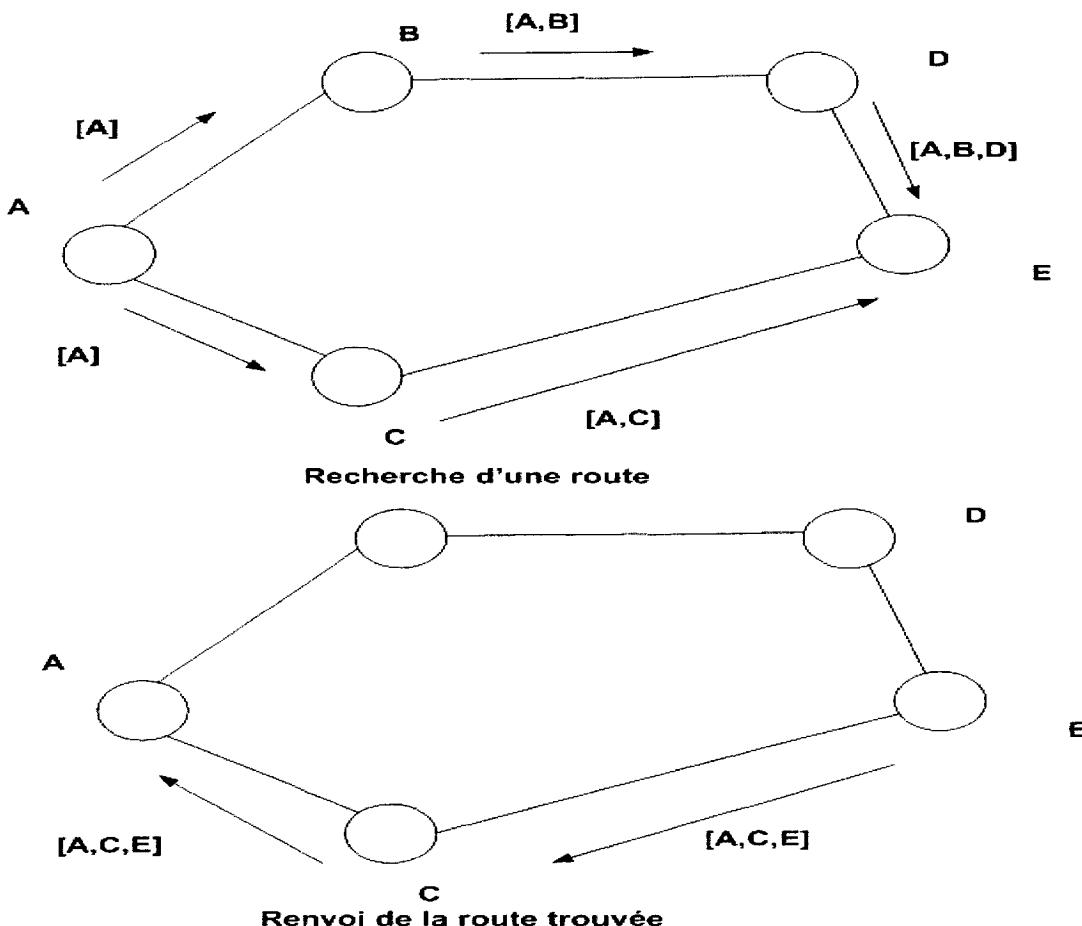


Figure 2.4 Le protocole DSR

Ainsi, le principal avantage de DSR est sa réactivité. En effet, le protocole DSR réduit les coûts en bande passante puisque la route n'est recherchée que si une communication doit être établie. Par contre, cette communication ne pourra se faire qu'après un certain délai rattaché justement à la durée nécessaire pour la recherche du chemin.

Le protocole AODV

Comme le protocole précédent, AODV [8] diffuse une requête de découverte de route pour identifier les nœuds par lesquels va transiter l'échange entre la source et la destination. Cependant, dans ce protocole, tous les nœuds transitaires mettent à jour leurs tables de routage quand une réponse à une requête de route leur parvient. De même, si un nœud transitoire connaît déjà un chemin vers la destination, il l'utilisera et arrêtera par la même occasion la diffusion de la requête de route.

Le fonctionnement d'AODV est le suivant : quand un nœud source s voudra communiquer avec un nœud destination d et que le premier ne possède pas de chemin vers d , il diffusera une requête de route dans le réseau. Cette requête est un paquet particulier appelé *Route Request* ou RREQ. En plus de l'adresse de la source, RREQ contient un numéro de séquence identifiant la requête courante, un identificateur de diffusion ainsi que le numéro de séquence du dernier paquet reçu par la source et qui l'informe sur la destination. Tout nœud recevant RREQ met à jour sa connaissance du nœud s et selon le cas comme suit :

- Si il s'agit de la destination d elle-même ou d'un nœud connaissant un chemin vers celle-ci (avec un numéro de séquence plus récent que celui contenu dans le RREQ) alors le nœud courant émet une réponse sous forme d'un paquet appelé *Route Reply* ou RREP;
- Si le nœud a déjà traité la requête (même source et même identificateur de diffusion) alors celui-ci sera ignoré;
- Sinon le nœud diffuse le RREQ.

Les nœuds intermédiaires profitent de l'acheminement des paquets RREP vers s pour établir un pointeur vers d . Une fois qu'un chemin entre une source et une destination a été trouvé, il sera utilisé pendant les communications. Cependant, si la source reçoit un RREP plus récent avec un nombre de sauts moins importants, elle mettra à jour le chemin de routage et utilisera la meilleure route.

En cas de bris d'un lien appartenant à la route active, la source sera avertie par un paquet *Route Error RERR* et une autre requête de route sera lancée si nécessaire. À la fin des communications, les liens qui ont été utilisés seront effacés des tables de routage.

AODV supporte également le routage *multicast* en constituant des arborescences. Dans ce cas, les nœuds appartenant à l'arborescence *multicast* seront en mesure d'émettre un paquet RREP vers la source. Après un certain délai servant à obtenir un nombre raisonnable de réponses, la source choisit le meilleur chemin et répond en émettant à son tour un message d'activation du routage *multicast* appelé MACT.

Ainsi, l'avantage d'AODV réside non seulement dans le fait qu'il limite l'utilisation de la bande passante de part sa réactivité mais aussi dans le fait qu'il supporte aussi bien le routage *unicast* que *multicast*. Son principal inconvénient réside cependant dans le temps de découverte et de calcul de routes.

Le tableau 2.2 compare DSR et AODV. La notation utilisée est identique à celle du tableau 2.1 sauf pour e qui est le nombre de communication entre les paires de nœuds :

Tableau 2.2 Comparaison entre les protocoles réactifs

	AODV	DSR
Type de routage	Sur demande	Sur demande
Métrique utilisée	Plus court chemin	Plus court chemin
Fréquence des mises à jour	Quand nécessaire	Quand nécessaire
Nombres de séquence utilisés	Oui	Non
Sans boucle	Oui	Oui
Pire cas existe	Oui (<i>flooding total</i>)	Oui (<i>flooding total</i>)
Chemins multiples supportés	Non	Oui
Complexité de mise en mémoire	$O(e)$	$O(e)$
Complexité de communication	$O(2N)$	$O(2N)$

À la lumière de ce qui a précédé, nous pouvons affirmer qu'en terme de trafic réseau et de mobilité, les protocoles proactifs construisent des routes plus efficaces que les protocoles réactifs. De plus, les routes sont formées d'avances et donc, il n'y a pratiquement aucun délai avant le début des échanges. Cependant, ces protocoles consomment beaucoup de bande passante et d'énergie à cause de leurs mises à jour périodiques. Ainsi, les protocoles proactifs sont inadaptés pour les applications critiques. Par ailleurs, du point de vue qualité de service, les protocoles proactifs tels que FSR, OLSR et TBRRPF peuvent facilement prendre en considération des contraintes sur le délai ou la bande passante en ajoutant une entrée correspondante dans les tables de routage. Par contre, pour les protocoles sur demande, les contraintes de qualité de service doivent intervenir lors du processus de découverte de route.

2.1.2 Routage hiérarchique :

Lorsque le réseau ad hoc devient relativement grand (plus d'une centaine de nœuds), les techniques de routage plat présentées ci haut deviennent difficilement réalisables. En effet, de telles techniques seraient inadéquates principalement parce qu'elles nécessiterait beaucoup plus de ressources qu'en possèdent les nœuds mobiles (mémoires, CPU etc..).

C'est pour pallier à ces lacunes que plusieurs algorithmes de routage hiérarchique ont été développés. L'idée principale de ces protocoles est de séparer les nœuds du réseau en plusieurs groupes et d'assigner différents rôles aux nœuds constituant chaque groupe. Ainsi, on réduit considérablement la taille des tables de routage de la plupart des nœuds puisqu'elles ne comprennent que les informations concernant leurs groupes respectifs. La manière la plus commune pour bâtir l'hiérarchie consiste à grouper les nœuds selon leur disposition géographique formant ainsi une grappe. Chaque grappe aura un nœud tête-de-grappe qui permet la communication à l'intérieur de celle-ci. Une alternative est d'implanter une hiérarchie implicite de sorte que chaque nœud ait sa propre portée

locale. La technique de routage utilisée sera différente selon que la destination appartient à la portée locale de la source ou pas.

Le protocole CGSR

Le protocole de Changement de Routage Grappe-Passerelle [9] (ou *Clusterhead-Gateway Switch Routing*) est un protocole classique de routage hiérarchique. Il consiste en premier lieu à partager le réseau en grappes stables à partir de l'algorithme LCC (*Least Clusterhead Change*) et à élire un nœud tête-de-grappe dans chaque grappe. Un nœud mobile appartenant à plusieurs grappes jouera le rôle de passerelles entre celles-ci. Ainsi, entre toute paire de nœuds source s et destination d deux cas sont possibles :

- Si s et d appartiennent à la même grappe alors les paquets de données seront routés selon le chemin suivant : $s \rightarrow t \rightarrow d$ où t est la tête-de-grappe.
- Si s et d appartiennent aux grappes différentes G_0 et G_n respectivement alors le chemin emprunté par les paquets de données sera le suivant :

$$s \rightarrow t_0 \rightarrow p(G_0 - G_1) \rightarrow t_1 \rightarrow p(G_1 - G_2) \rightarrow t_2 \rightarrow \dots \rightarrow p(G_{i-1} - G_i) \rightarrow t_i \rightarrow p(G_i - G_{i+1}) \rightarrow \dots \rightarrow t_n \rightarrow d$$

Où t_i est tête-de-grappe de la grappe G_i et $p(G_{i-1} - G_i)$ est la passerelle entre les grappes G_{i-1} et G_i .

Dans CGSR, chaque nœud mobile maintient deux tables : une table de membres de chaque grappe et une table de routage basé sur le vecteur de distance. La première table enregistre les têtes-de-grappes de chaque nœud dans le réseau. Cette table est diffusée à intervalle de temps régulier. À chaque fois qu'il reçoit un paquet de données, un nœud mobile met à jour s'il y a lieu, la table de membres. La table de routage comprend quant à elle, la route vers chaque tête-de-grappe sans égard au nombre de nœuds que comprend la grappe en question.

Pour router un paquet de données, le nœud mobile source commence par identifier dans la table de membres, le nœud tête-de-grappe de la grappe de destination. Ensuite, il consulte sa table de routage afin d'identifier le prochain saut vers la grappe destination et route les paquets de données à travers les différentes têtes-de-grappes jusqu'à ce que

les paquets arrivent vers la tête-de-grappe de destination qui les achemine vers le nœud mobile destination.

La figure 2.5 ci-dessous illustre le fonctionnement de CGSR.

Le principal avantage de CGSR réside dans le fait qu'il réduit énormément la taille des tables de routage des nœuds. Par conséquent, la diffusion de ces tables dans le réseau ne consomme pas autant de ressources que les protocoles classiques basés sur le vecteur à distance. Par ailleurs, les inconvénients de ce protocole se résument à la difficulté de maintenir une grappe dans un contexte de mobilité. De ce fait, le LCC utilisé pour construire les grappes, produit un surdébit et complexifie la formation et la maintenance des grappes.

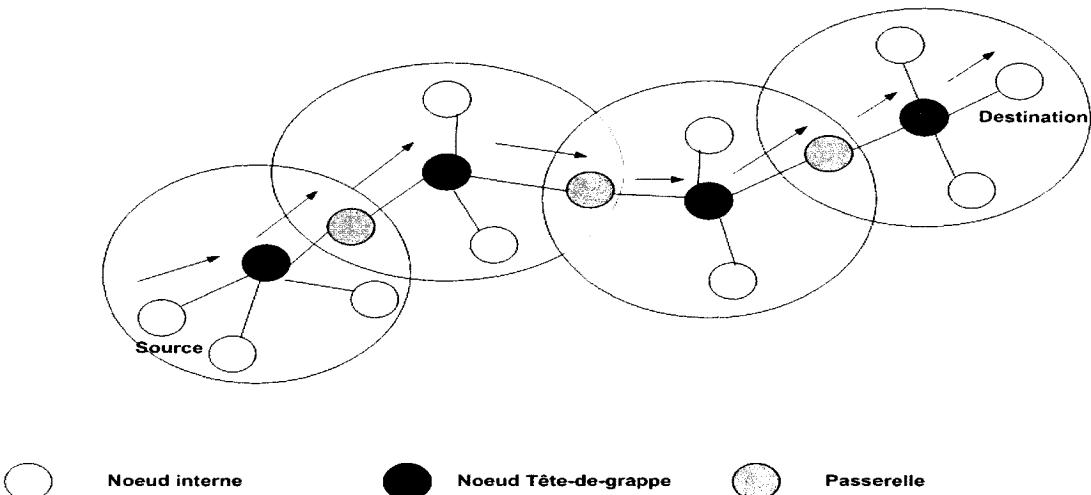


Figure 2.5 Protocole CGSR

Le protocole HSR

Le protocole de Routage à État Hiérarchique [10] (ou *Hierarchical State Routing*) est un protocole de routage hiérarchique multi-niveaux basé sur l'état des liens. Il construit une topologie hiérarchique logique à plusieurs niveaux. Les nœuds de même niveau sont regroupés en grappes. Dans chaque grappe, les nœuds élisent une tête-de-grappe qui deviendra membre du niveau suivant. Les nœuds ainsi élus de chaque grappe s'organisent en grappes de niveaux supérieurs et à leur tour choisissent des têtes-de-

grappes qui formeront le niveau suivant et ainsi de suite. Le but de la procédure est à la fois de réduire la taille des tables de routage, les temps de traitement et de transmission à chaque niveau.

La figure 2.6 est un exemple du routage HSR à trois niveaux. Nous remarquons la présence de trois types de nœuds

- Les têtes-de-grappes qui jouent le rôle de coordonnateur à l'intérieur des grappes;
- Les nœuds passerelles qui lient les différentes grappes;
- Les nœuds internes.

Au premier niveau, chaque nœud diffuse ses états de liens à l'intérieur de sa grappe. La tête-de-grappe regroupe l'information concernant l'état des liens de la grappe et la diffuse aux têtes-de-grappes voisines (à travers les passerelles).

On considère l'existence d'une connectivité logique de niveau 2 entre les têtes-de-grappes voisines. En effet, dans la figure 2.6 les têtes-de-grappes 1 et 2 deviennent membres de la grappe G2 de niveau 2. Ainsi, au niveau 2, les états de liens comprennent des liens virtuels comme celui qui existe entre les nœuds 1 et 2. Ce lien virtuel, se traduit au niveau 1 par le chemin regroupant les deux têtes-de-grappes (le chemin 1->6->2).

Après avoir obtenu, l'information sur les états de liens, les nœuds de niveaux supérieurs diffusent l'information vers les nœuds de niveaux inférieurs de sorte que chaque nœud de niveau 1 puisse avoir une information hiérarchique sur la topologie du réseau.

Cette hiérarchie se traduit principalement au niveau des adresses. L'adresse hiérarchique du nœud n est une séquence formée des numéros des nœuds du chemin qui amène du niveau supérieur vers le nœud n . Par exemple, l'adresse du nœud 5 sera $<1,1,5>$.

L'avantage d'utiliser l'adressage HSR réside dans le fait que les nœuds peuvent mettre à jour leurs adresses dynamiquement quand ils reçoivent l'avis de nœuds de niveau supérieur.

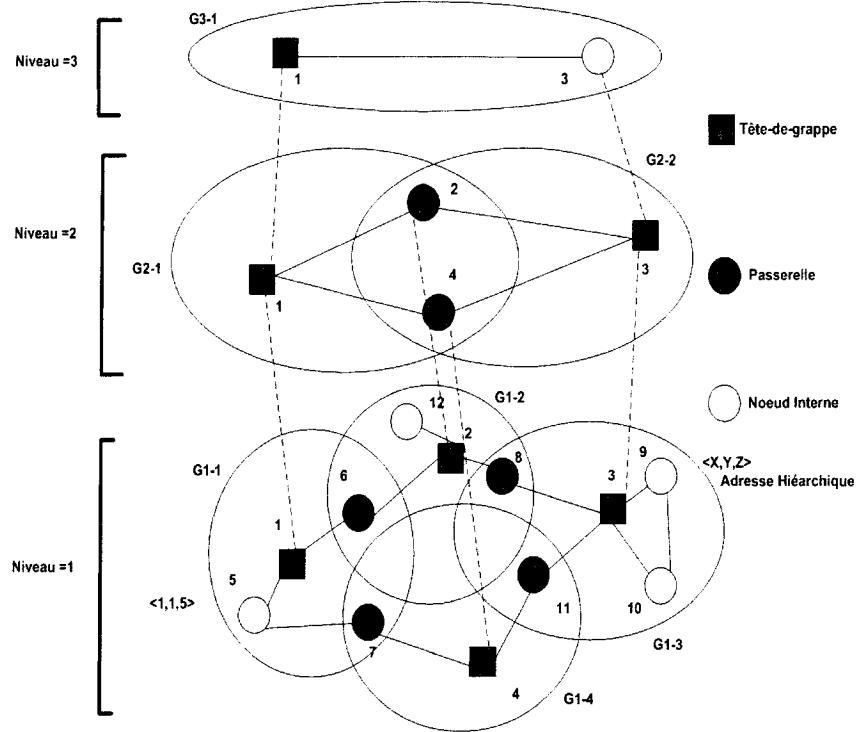


Figure 2.6 Le Protocole HSR

De plus, l'adressage hiérarchique permet d'acheminer les paquets quelque soit la destination à partir des tables de routage hiérarchique. Notons que les passerelles peuvent communiquer avec deux ou plusieurs têtes-de-grappes et donc avoir plus d'une adresse hiérarchique (exemple $\langle 3,3,11 \rangle = \langle 3,4,11 \rangle$). Par ailleurs, l'inconvénient de HSR se résume aux changements fréquents des adresses hiérarchiques dû à la mobilité des nœuds ce qui peut rendre la localisation difficile.

Le Protocole ZRP

Le protocole de zone de routage (ou *Zone Routing Protocol*) [11] est un protocole de routage hybride proactif / réactif, qui tente d'hériter des avantages de ces deux catégories de protocoles. Dans ZRP, chaque nœud mobile possède une *zone* dont il est le centre et dont le rayon est fixé à un nombre de saut donné. Par exemple, si le

nombre de sauts est fixé à trois, alors la zone comprendra tous les voisins du nœud qui sont à une distance de trois sauts ou moins.

Le protocole ZRP comprend trois principales composantes :

- À l'intérieur de la zone, le protocole de routage intrazone (*IARP Intrazone Routing Protocol*) conserve l'information relative au routage. Si le nœud destination est à l'intérieur de la zone du nœud source, alors IARP connaît obligatoirement une route pour transmettre les paquets;
- Pour les nœuds hors zone, le protocole de routage interzone (*IERP Interzone Routing Protocol*) sera utilisé. Le protocole IERP est un protocole réactif qui utilisera des requêtes/réponses de routes (RREQ/RREP) afin de trouver le chemin vers la destination. Soulignons que IERP débute la recherche de route à partir de la frontière de la zone du nœud source. Ainsi les paquets RREQ seront diffusés à partir de la périphérie de la zone de routage;
- Les requêtes de routes RREQ sont diffusées par le protocole de diffusion BRP (*Broadcast Resolution Protocol*) jusqu'à ce qu'un nœud connaisse le chemin vers la destination (c'est-à-dire que le nœud destination soit dans sa zone).

Soulignons que IARP et IERP ne sont pas des protocoles qu'on peut implémenter en tant que tels mais bien des cadres généraux qui devront être mis en place en utilisant selon le cas un des protocoles proactifs ou réactifs présentés ci-dessus. Ainsi, selon l'implémentation choisie, tout protocole de routage proactif basé sur l'état des liens ou sur le vecteur de distance pourra agir comme IARP. De même, tout protocole réactif pourra jouer le rôle de IERP.

Dans ce qui suit, nous détaillerons chacune des composantes du protocole :

Le protocole de routage intrazone (IARP)

Le protocole intrazone de routage (IARP) [27] est le protocole utilisé pour la communication à l'intérieur de la zone ZRP. Son utilisation est donc limitée à l'intérieur des zones de routage.

Attendu que le voisinage d'un nœud dans un réseau ad hoc est sujet à rapide changement et que celui-ci aura un impact beaucoup plus important que celui d'une modification de la topologie lointaine, le protocole IARP devra être proactif afin de réagir rapidement et efficacement devant un changement potentiel. Pour ce faire, les nœuds mobiles devront continuellement mettre à jour l'information de routage. Le protocole IARP optimise localement les chemins sélectionnés en évitant les routes qui sont relativement longues en terme de nombre de sauts. De plus, le protocole IARP tente d'éviter les liens en pannes en optant pour des tronçons alternatifs.

Notons que IARP devra supporter aussi bien les liens unidirectionnels que bidirectionnels qui peuvent exister dans le réseau.

Aussi de part sa nature proactive, le protocole IARP est très performant en matière de découverte de route. En effet, les routes vers destinations locales sont immédiatement disponibles. Afin toutefois de ne pas surutiliser la bande passante du réseau, la portée du protocole est limitée dans l'espace (zone de routage). Ainsi, comme nous l'avons expliqué précédemment et afin de pouvoir utiliser un protocole traditionnel à états des liens comme IARP, les zones de routage devront être limitées à ρ sauts. Cette limitation peut être implantée en ajoutant un paramètre de temps de vie (TTL) aux requêtes de découvertes de routes qui sera initialisé à $\rho - 1$ et décrémenter à chaque saut jusqu'à ce qu'il atteigne zéro.

La communication avec les nœuds hors zone est faite quant à elle à travers les nœuds de la périphérie des zones de routage.

Le protocole de routage interzone (IERP)

Dans le cas où le nœud destination ne se trouve pas dans la zone de routage du nœud source, le protocole de routage interzone [28] est alors utilisé.

Les requêtes de routes IERP sont faites uniquement sur demande. Par conséquent, ZRP limite l'utilisation de la bande passante hors zone de routage en envoyant la requête de route uniquement aux nœuds de la périphérie de la zone de routage, évitant ainsi une diffusion à tous les noeuds locaux. De plus, aucun nœud, même ceux de la périphérie,

n'envoie d'accusé de réception vers la source. Cependant, étant donné que le chemin n'est pas connu d'avance, la découverte de route nécessite un délai supplémentaire.

Il est important de noter que le protocole IERP profite aussi bien des connaissances acquises par le protocole de routage intrazone IARP que par le protocole de diffusion (BRP), ce dernier sera étudié plus loin.

Le protocole de diffusion (BRP)

Le protocole de diffusion, *Bordercast Resolution Protocol* [29], est utilisé dans ZRP pour diriger les paquets de requêtes initiées par le protocole de routage interzone IERP en vue de maximiser la performance. Ce protocole de diffusion peut être vu comme un pont entre IARP et IERP. Le protocole BRP construit un arbre de diffusion basé sur l'information fourni par le protocole proactif de routage intrazone IARP. Contrairement à IERP et IARP, le protocole BRP est plus un protocole de livraison de paquets qu'un protocole de routage.

Le protocole BRP tente de conserver la liste des nœuds qui ont déjà reçu une requête donnée dans le but d'éviter de leur retransmettre cette même requête. Quand un nœud i reçoit un paquet de requête de route pour le diffuser vers un nœud j qui n'appartient pas à sa zone de routage, BRP construit un arbre de diffusion pour que le noeud i puisse acheminer la requête à travers ses voisins. Chacun de ces derniers reçoit cette requête et reconstruit l'arbre de diffusion et détermine les nœuds qui n'appartiennent pas à l'arbre de diffusion du nœud source pour que la requête puisse leur être acheminée. Le processus continue ainsi en marquant les nœuds qui ont déjà reçu la requête jusqu'à ce qu'un nœud retrouve le nœud j dans sa zone de routage.

Dans le but de déterminer si la zone de routage a été couverte par la diffusion, BRP inclut deux niveaux de *détection de requête*. Ainsi, lors de la diffusion vers les nœuds de la périphérie, les nœuds internes transmettant la requête vers la périphérie détectent la requête et sont par conséquent marqués comme étant couverts par celle-ci. Il s'agit du premier niveau de détection (QD1).

De plus, un seul canal est utilisé pour la diffusion. Ainsi, un nœud peut être informé d'une requête par l'écoute du canal en question. Il s'agit du second niveau de détection (QD2).

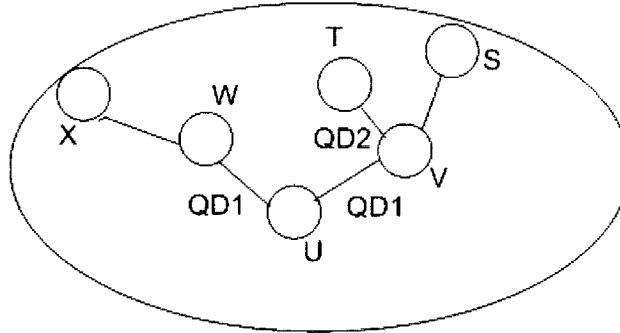


Figure 2.7 Niveaux de détection de requêtes ZRP

Dans la figure 2.7, le nœud U tente de transmettre la requête de route vers les nœuds de la périphérie S, T et X. Ainsi, lors de la diffusion vers le noeud S, le nœud V est marqué par le niveau (QD1) alors que le nœud T est marqué par le niveau (QD2).

Bien que la détection de la requête permet de marquer un nœud, celle-ci ne suffit pas à elle seule. En effet, on devra en plus éviter d'acheminer les paquets vers les nœuds couverts qui donc les ont déjà reçus. Cela s'appelle la *terminaison hâtive* (ET) et dépend aussi bien de la *détection de requête* que de la *terminaison de boucle* (TL). Cette dernière consiste à éliminer toutes les routes formant des boucles dans les zones locales de routage, puisqu'il s'agit de nœuds qu'on peut joindre localement avec IARP.

Par ailleurs, afin d'éviter toute diffusion inutile, le protocole BRP implante une *Diffusion Sélective*. Pour ce faire, tout nœud doit connaître la topologie autour d'une zone plus grande que sa zone initiale de taille $2 \rho - 1$. Grâce à cela, un nœud peut éliminer un nœud de la périphérie de sa liste de diffusion à condition que la zone locale de routage de celui-ci soit couverte par d'autres nœuds de la périphérie.

C'est notamment le cas dans la figure 2.8, où le nœud 1 pourra retirer le nœud 3 de sa liste de diffusion étant donné que les nœuds 7 et 8 sont successivement couverts par les nœuds 2 et 4. En effet, le nœud 3 couvre les nœuds 7 (couvert également par le nœud 4) et 8 (qui est couvert par le nœud 2).

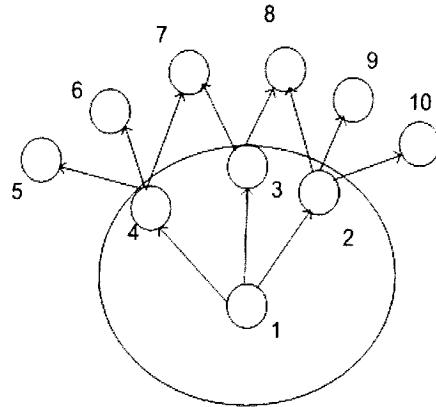


Figure 2.8 Diffusion sélective ZRP

Ainsi, ZRP limite le surdébit présent dans les protocoles proactifs à une seule zone. De plus, il limite la procédure de recherche de routes aux nœuds hors zone. Cependant, ZRP peut s'avérer inefficace en cas de *flooding* total du réseau par les paquets de requêtes (grande taille de la zone de routage).

Le tableau 2.3 ci-dessous compare les protocoles CGSR, HSR et ZRP. Les significations de N et e ont été indiquées précédemment. De plus, M est le nombre moyen de nœuds dans les grappes, L le nombre moyen de nœuds dans la zone et H le nombre de niveaux pour HSR :

Tableau 2.3 Comparaison des protocoles hiérarchiques

	CGSR	HSR	ZRP
Hiérarchie	Deux niveaux explicites	Multi niveaux explicites	Deux niveaux implicites
Type de routage	Proactif, vecteur de distance	Proactif, états des liens	Hybride, VD et ÉL
Sans boucle	Oui	Oui	Oui
Métrique de routage	Via nœuds critiques	Via nœuds critiques	Plus courts chemins
Nœuds critiques	Oui (grappe)	Oui (grappe)	Non
Complexité de mise en mémoire	$O(N/M)$	$O(M*H)$	$O(L) + O(e)$
Complexité de communication	$O(N)$	$O(M*H)$	$O(N)$

2.2 Routage et qualité de service :

Les protocoles présentés dans la section précédente sont dits de moindre effort (*Best Effort*). En effet, leur unique but est d'optimiser l'utilisation des ressources réseaux. Or, de plus en plus d'applications nécessitent une certaine qualité de service. Celle-ci est définie comme étant la garantie que les connexions établies répondent à certains critères (généralement concernant les délais ou la bande passante).

Par ailleurs, garantir une qualité de service pour les réseaux ad hoc constitue un sujet de recherche fort complexe. En effet, la capacité du réseau mobile d'assurer la qualité de service dépend grandement des caractéristiques intrinsèques de chaque nœud formant le réseau MANET.

Comme nous l'avons déjà vu, la QoS est encore plus difficile à atteindre principalement pour les raisons suivantes :

- La faiblesse et la grande variabilité des capacités des liens sans-fil;
- Le faible débit des liens;
- La topologie changeante du réseau;
- Les protocoles MAC les plus répandus (tel que 802.11b) n'intègrent pas de support pour la qualité de service.

De plus, il serait important d'ajouter que dans un contexte où les nœuds du réseau sont très mobiles, garantir la qualité de service peut devenir non seulement difficile mais statistiquement impossible. En effet, les mises à jour propagées peuvent alors rencontrer un lien rompu ou qui ne satisfait pas au critère de qualité de service demandé. Par contre, si la topologie du réseau ad hoc change relativement lentement pour permettre la diffusion des mises à jour à travers le réseau, celui-ci est dit *combinatoirement stable* faute de quoi on ne peut parler de qualité de service.

Il va sans dire que la qualité de service ne peut être complètement assurée par une seule couche du réseau. En effet, toutes les couches doivent inclure une composante qui garantie la qualité de service. Parmi celles-ci, les plus importantes sont: QoS pour MAC,

QoS pour routage ainsi qu'un mécanisme de signalisation et de réservation de ressources.

Dans ce qui suit, nous présenterons en premier lieu les modèles de QoS les plus répandus, modèles qui définissent l'architecture globale dans laquelle les contraintes de services doivent être respectées. Ensuite, nous enchaînerons avec les protocoles de QoS de niveau MAC qui tentent de faire bénéficier les protocoles de haut niveau des mécanismes offerts pour garantir la qualité de service. En troisième lieu, nous examinerons en détail quelques protocoles de routage avec QoS pour finalement présenter les mécanismes de signalisation qui tentent de réserver les ressources pour les protocoles de routage.

2.2.1 Modèles de qualités de service :

Avant d'entamer l'étude sur l'intégration de la qualité de service pour les réseaux ad hoc, nous jugeons approprié de résumer quelques mécanismes et approches suggérés pour les réseaux filaires. En effet, l'IETF propose principalement deux modèles pour satisfaire les besoins des applications en matière de qualité de services :

- ❖ IntServ [12] : qui est basé sur la réservation par flot. Il s'agit d'un modèle orienté délai qui suggère trois classes de services : *le service garanti, la charge contrôlée et le moindre effort*. Ainsi, la première classe IntServ garantit à la fois le débit et le délai maximal sans pour autant offrir une assurance sur la gigue ou le taux de perte de paquets. Pour la seconde catégorie, ce mécanisme n'offre aucune garantie quantifiable mais plutôt l'assurance que le trafic ne subira aucune congestion. Pour la dernière classe enfin, le mécanisme agit tout comme les protocoles de moindre effort. Ajoutons que IntServ utilise une signalisation de type RSVP;
- ❖ DiffServ [13]: Il s'agit d'un modèle à gestion distribué par domaine conçu pour les cœurs de réseaux. Il agrège les flots en fonction de leur qualité de service et marque les paquets IP. Aussi, DiffServ fonctionne sans signalisation échangée

et sans aucun mécanisme de réservation ou de contrôle de congestion. Il différencie trois types de nœuds : nœuds d'entrée, nœuds de sortie et nœuds intermédiaires. De plus, le modèle DiffServ classe le trafic en différentes catégories ou priorités dans le but d'appliquer un traitement différencié aux différentes classes de trafic. Pour ce faire, DiffServ utilise le champ TOS (*Type of Service*) défini dans IPv4 que l'IETF a rebaptisé champ DS (*DiffServ*).

Ces deux modèles demeurent cependant inadéquats pour les réseaux ad hoc. En effet, IntServ exige une capacité de traitement incompatible avec les contraintes d'énergie des nœuds des réseaux MANET. De plus, le protocole de signalisation RSVP requiert une demande de bande passante difficile à satisfaire dans le cas d'un réseau sans fil. Et si le modèle DiffServ peut paraître plus adapté au contexte des réseaux ad hoc, il demeure néanmoins conçu pour des coeurs de réseaux qui ne sont pas sujet à mobilité et qui ont une importante bande passante. En effet, DiffServ prévoit des routeurs périphériques appelés *Edge Routers* qui effectuent les opérations de classification, de contrôle et de marquage ainsi que des routeurs centraux ou *Core Routers* qui traitent les paquets en fonction du champ DS de l'en-tête d'IP. Ce type d'architecture est pour le moins difficile à adapter aux réseaux ad hoc.

FQMM

FQMM [14] est un modèle de qualité de services conçu pour les réseaux ad hoc de taille petite à moyenne (jusqu'à 50 nœuds) utilisant une topologie plate non hiérarchique. De plus, FQMM est assez proche des mécanismes des réseaux filaires pour permettre d'interfacer ceux-ci avec les réseaux ad hoc. Ce modèle définit plusieurs classes de services dont la plus prioritaire permet aux flux de spécifier leurs propres contraintes.

Tout comme DiffServ, FQMM définit trois catégories de nœuds : émetteurs, récepteurs et intermédiaires. De part leur nature, les nœuds mobiles du réseau ad hoc peuvent appartenir à différentes catégories pour différents flux.

Le modèle FQMM tente de concilier les modèles IntServ et DiffServ en optant pour un approvisionnement hybride par-flots (comme IntServ) et par-classes (comme DiffServ). Ainsi, tout comme DiffServ, FQMM possède un mécanisme de différentiation. De plus, pour certaines classes trafic, FQMM pourra convertir la granularité par classes en granularité par flots. Cependant, il est difficile de garantir la granularité par flots à l'échelle du réseau et ce, compte tenu de la bande passante limitée. Par conséquent, FQMM tente de préserver celle-ci pour une petite portion de trafic dans le réseau MANET.

L'un des avantages de FQMM est qu'il adapté aussi bien pour les réseau ad hoc que pour les réseaux filaires classiques. Cependant, si les problèmes reliés à l'extensibilité du modèle IntServ paraissent résolus, ce n'est malheureusement pas le cas des problèmes liés à la nature des réseaux ad hoc (contraintes d'énergie, de bande passante, etc.) qui sont délégués au protocole de routage de couche inférieure. En effet, FQMM ne peut garantir seul la satisfaction des contraintes de qualité de service. Ainsi, FQMM requiert la mise en œuvre d'un protocole de routage capable de rechercher les routes qui satisfont aux contraintes de qualité de service.

Modèle QoS basé sur la prédiction de la performance locale

Ce modèle proposé dans [15] a été spécialement conçu pour les réseaux ad hoc. Comme son nom l'indique, il tente de prédire la performance locale future du réseau pour adapter le routage aux ressources disponibles.

Pour ce faire, le modèle lance une procédure de prédiction locale sur chaque lien. Ainsi, la performance sera estimée en traduisant les paramètres des couches de bas niveau en information sur l'état des liens.

Par ailleurs, ce mécanisme se base sur un modèle analytique pour calculer les différents délais qui existent dans le réseau. En effet, il utilise un modèle de Markov et ce, dans le but de prédire le délai pour chaque nœud.

L'avantage de ce modèle réside dans le fait qu'il tente tenir compte des différentes évolutions possibles du réseau (perte de nœuds, mouvement d'un nœud etc..). Cependant, il ne s'agit que d'un modèle probabiliste qui peut dans certains cas ne pas refléter l'évolution réelle voulue.

2.2.2 Mécanismes de niveau 2

Les rôles des mécanismes de ce niveau sont divers :

- Éviter les collisions;
- Partager la bande passante;
- Résoudre les problèmes spécifiques aux transmissions hertziennes (tels que les stations cachées ou exposées).

Cela n'empêche pas bon nombre de protocoles de routage avec QoS d'espérer profiter de certains mécanismes de niveau MAC pour assurer la qualité de service.

Le protocole MACA / PR :

MACA / PR [16] (ou *Multiple Access Collision Avoidance with Piggy back Reservation*) différencie la politique d'accès au médium selon la nature du flux.

Ainsi, le flux non privilégié sera traité de façon standard (*Best Effort*). Quant au flux privilégié (flux temps réel), une demande d'autorisation unique (échange RTS-CTS) est transmise au début du flux. Les paquets qui suivent seront ensuite acheminés directement. À leur arrivée à destination, un acquittement sera envoyé par le récepteur. Si l'émetteur ne reçoit pas d'acquittement, une nouvelle demande d'autorisation sera transmise.

Par ailleurs, dans le but de réserver la bande passante nécessaire, l'émetteur inclut préalablement les informations de l'ordonnancement d'un paquet dans celui qui le précède. De cette manière, tous les voisins de la destination sauront, en écoutant l'acquittement envoyé par celle-ci, le moment où arrivera le paquet suivant et pourront ainsi, le cas échéant, différer leurs propres transmissions afin d'éviter la collision.

L'avantage de MACA/PR est qu'il permet de résoudre la problématique des stations cachées¹. D'autre part, sa faiblesse réside dans le fait qu'aucune transmission de données en temps réel n'est possible avant la réception de l'autorisation.

Intégration de la QoS pour la norme IEEE 802.11

Le but proposé par les auteurs dans [17] est d'intégrer un mécanisme de différenciation de services dans le protocole 802.11. Pour ce faire, on établit un système de priorités entre les trames, lequel peut être intégré en adaptant certains paramètres de la fonction de coordination diffusée (DCF) à la priorité des paquets.

Cette fonction est basée sur la détection de la porteuse (CSMA). Pour pouvoir émettre, le nœud doit être sûr que le médium est libre pendant au moins (DIFS – CDF *Inter Frame Spacing*) unités de temps et ce, afin de prioriser la transmission de certains paquets de signalisation, laquelle peut s'effectuer dès que le canal a été libre pendant un temps SIFS (*Short Inter Spacing*) plus court que DIFS.

En ajoutant au DIFS (qui est constant) un délai aléatoire, on empêche deux nœuds mobiles de commencer à émettre en même temps. Si malgré tout une collision survient, on réinitialisera le processus et le délai aléatoire sera plus grand.

Certains paramètres pourront être adaptés de manière dynamique permettant l'intégration des priorités dans le protocole 802.11 :

¹ Problème qui survient quand deux nœuds sources S_1 et S_2 tentent de communiquer en même temps avec un nœud destination d . À cause des collisions, les données transmises par l'une des deux sources ne sont pas reçues.

- Utiliser différentes valeurs de délai de silence (DIFS) selon la priorité;
- Incrémenter les délais aléatoires de manières différentes selon la priorité (suite à une collision);
- Limiter la longueur des trames selon la priorité (les trames prioritaires occuperont le médium plus longtemps que les trames moins prioritaires).

Suite à plusieurs tests avec des flots UDP et TCP, la première solution paraît plus stable.

Ainsi, le principal avantage de ce mécanisme tient au fait qu'il ne s'agit pas d'un nouveau protocole mais plutôt d'une adaptation du standard 802.11. De plus, il résout le problème des stations cachées. Par ailleurs, plus les collisions se produisent, plus le délai aléatoire d'attente sera important.

Le tableau suivant compare les deux derniers mécanismes présentés :

Tableau 2.3 Comparaison entre MACA / PR et IEEE 802.11 avec QdS

	MACA/PR	IEEE 802.11 avec QdS
Demande d'autorisation au début de l'échange	Oui	Non
Ajout par rapport au 802.11	Nouveau protocole	Simple adaptation
Différenciation des flux	Oui	Oui
Problème de stations cachées résolu	Oui	Oui

2.2.3 Routage avec qualité de service :

L'objectif du routage avec qualité de service dans les réseaux ad hoc est de rechercher une route entre la source et la destination qui satisfait aux contraintes de qualité de service. Dès lors, il ne suffit plus de trouver une route quelconque mais la

route trouvée devra respecter certains critères tels que le délai, la bande passante ou le coût de transmission.

Ces conditions supplémentaires font en sorte que le routage avec QoS coûte généralement plus cher en terme de ressources que le routage classique. En effet, la réservation des ressources change la topologie et doit de ce fait être diffusée dans le réseau.

Dans cette section, nous tenterons de présenter quelques mécanismes de routage avec qualité de service tels que proposés dans la littérature.

Routage avec qualité de service à partir de la source

Ce protocole réactif fonctionne de la manière suivante :

- La source initie une découverte de route (qui doit satisfaire les contraintes de qualité de service) en diffusant la requête vers ses voisins;
- Chaque demande de route est unique et est identifiée par l'adresse de la source ainsi qu'un numéro de séquence;
- Chaque nœud intermédiaire propage la requête vers ses voisins qui respectent les critères de qualité de service;
- Tout nœud intermédiaire ignore les requêtes qu'il a déjà traitées (même identification);
- Tout nœud intermédiaire enregistre le nœud qui lui a acheminé la demande (prédécesseur). Il l'effacera après un certain temps de manière à ne pas utiliser inutilement ses ressources mémoires;
- Quand la requête arrive à destination, le chemin parcouru par la requête sera conforme aux contraintes de QoS. Grâce à l'enregistrement des prédécesseurs, un message de confirmation sera envoyé vers la source en empruntant le chemin inverse. À son arrivée à la source, la connexion deviendra active;

- Si à cause des changements de topologies le message de confirmation se perd en cours de transmission, le nœud destination se chargera de la libération des ressources.

L'avantage de ce protocole tient au fait qu'on élimine les nœuds qui ne se satisfont pas les contraintes de QoS dès qu'on les rencontre. Cependant, le principal inconvénient réside dans le trafic important qui est généré, surtout dans un contexte où la bande passante est limitée, comme c'est le cas dans les réseaux ad hoc.

Dans [18], l'auteur tente de remédier à cela principalement en proposant de :

- Limiter la diffusion des requêtes de routes vers les nœuds qui n'ont pas encore reçu celle-ci (inutile de réacheminer la requête vers le prédecesseur);
- Faire en sorte que le nœud achemine la requête vers tous ses voisins par un *multicast local radio*. Bien entendu, seuls ceux qui respectent les contraintes de qualité de service la prendront en considération. En effet, cela permet de limiter l'utilisation de la bande passante ainsi que l'envoi de requêtes;
- Permettre à chaque nœud de garder en mémoire pour un temps donné le nombre de sauts qui le séparent d'autres nœuds. Cela s'avérera particulièrement intéressant si un nœud intermédiaire connaît la distance qui le sépare de la destination. En effet, il pourra limiter la diffusion de la demande au nombre de sauts qu'il connaît (éventuellement il pourra incrémenter ce nombre quelque peu pour tenir compte de la topologie changeante). Au delà de cette limite, il va sans dire que toute propagation est inutile.

Ces techniques permettent donc de minimiser l'utilisation des ressources du réseau en terme de bande passante. Toutefois, cela présuppose que les nœuds possèdent suffisamment de ressources mémoire ce qui n'est pas toujours le cas dans un réseau ad hoc surtout que les nœuds peuvent être très différents (ordinateur portable, téléphone cellulaire, PDA ..).

Routage avec qualité de service à partir de la destination :

Contrairement au protocole précédent, dans celui-ci c'est la destination qui initie le routage après avoir reçue une première demande de connexion de la source. Cette première demande est routée selon un protocole de type moindre effort. Au cours de son acheminement, on estime la distance qui sépare la source de la destination en incrémentant un champ compteur à chaque saut.

À la réception de cette demande, la destination commence à rechercher une route qui satisfait les contraintes de qualité de service. Il est aussi à noter que même si on utilise un protocole de moindre effort au cours de ce premier échange cela n'empêche pas de vérifier si la route empruntée respecte les critères de QdS. La destination pourra ainsi directement vérifier si le chemin emprunté par cette première demande de connexion satisfait déjà aux contraintes. Si c'est le cas, la recherche de route est terminée et la destination envoie un message de confirmation à la source qui se charge de réserver les ressources nécessaires pour la connexion. Il va sans dire que l'utilisation d'un protocole de moindre effort supportant les routes multiples augmente considérablement nos chances d'obtenir une route respectant les critères de QdS du premier coup.

Dans le cas contraire, la destination recherche le chemin qui répond aux contraintes en utilisant un multicast local. En plus de l'adresse de la source et les critères de QdS, la requête inclut un champ TTL lequel est initialisé à l'aide de l'estimation du nombre de sauts entre la destination et la source (légèrement augmentée). Ainsi, la demande sera acheminée de voisins en voisins, seuls les nœuds qui satisfont aux critères et qui ne l'ont pas encore reçue la prennent en considération. Par ailleurs, à chaque saut le champ TTL est décrémenté jusqu'à ce que le message arrive à la source ou que le champ devienne nul auquel cas la requête n'est pas réacheminée.

Ce protocole présente plusieurs avantages. En effet, en supportant les chemins multiples il maximise les chances de trouver une routes respectant les critères de QdS dès le début. Si ce n'est pas le cas, le routage au mieux fournit à la destination une estimation qui lui

permettra de trouver la route avec les contraintes de QoS voulues. Cependant, le temps nécessaire pour effectuer le routage est relativement long.

Le protocole Ticket Based Probing:

Ticket Based Probing [19] entend limiter les coûts supplémentaires dus à la recherche d'un chemin qui satisfait les contraintes de qualités de services. Il suppose que la mobilité est assez restreinte pour que le routage avec qualité de service reste possible. Ce protocole diffère fondamentalement de ceux qui ont été présentés plus haut, en ce sens que chaque nœud prend en considération non seulement l'état local de ses liens mais aussi un état global. En effet, tout nœud i conserve l'information sur chacun de ses états de liens (i,j) comprenant : le délai (i,j) , la bande passante (i,j) encore disponible ainsi que le coût (i,j) .

De cette façon, nous pourrons déterminer chacun de ces trois paramètres pour tout chemin $P = i \rightarrow j \rightarrow \dots \rightarrow k \rightarrow l$:

$$\text{délai } (P) = \text{délai } (i,j) + \dots + \text{délai } (k,l) \quad (2.1)$$

$$\text{bande passante}(P) = \min \{ \text{bande passante}(i,j), \dots, \text{bande passante}(k,l) \} \quad (2.2)$$

$$\text{coût } (P) = \text{coût } (i,j) + \dots + \text{coût } (k,l) \quad (2.3)$$

Ainsi, le problème de routage avec QoS revient à trouver un chemin P entre une source s et une destination t et qui satisfait une condition donnée (par exemple $\text{délai } (P) \leq D$).

Pour ce faire, chaque nœud i conserve l'information reliée aux états de liens entre i et toute destination possible t . Cette information est périodiquement mise à jour par un protocole de vecteur de distance. Elle contient les éléments suivants :

- Le délai : $D_i(t)$ conserve le délai minimum bout-en-bout de i à t ;
- La bande passante : $B_i(t)$ conserve la plus grande bande passante disponible bout-en-bout;
- Le coût : $C_i(t)$ conserve le coût minimum de i à t ;
- La variation du délai : $\Delta D_i(t)$ conserve la variation du délai entre les mises à jour ce qui est utile dans le contexte d'une topologie changeante;

- La variation de la bande passante : $\Delta B_i(t)$ conserve la variation de la bande passante.

Ticket Based Probing est un protocole réactif où la source envoie des demandes de connexions pour trouver un chemin vers la destination respectant les contraintes de qualité de service. La source limite le processus de découverte de route en assignant à la demande de route un nombre déterminé de tickets qui seront diffusé à travers le réseau. Ainsi, le nombre de messages de routage est limité à tout moment durant la procédure de routage. En effet, le nombre maximum de routes recherchées correspond au nombre total de tickets envoyés.

Tel que montré dans la figure 2.9, la source s envoie deux demandes de routes $p1$ et $p2$. Le nombre entre parenthèse représente le nombre de tickets assigné à chaque demande. Au nœud j , $p2$ est divisé en $p3$ et $p4$ avec un ticket chacune. Il y a trois messages maximum à chaque instant qui correspondent aux routes recherchées : $s \rightarrow i \rightarrow t$, $s \rightarrow j \rightarrow t$ et $s \rightarrow j \rightarrow k \rightarrow t$. En effet, le protocole supporte la recherche multi chemins.

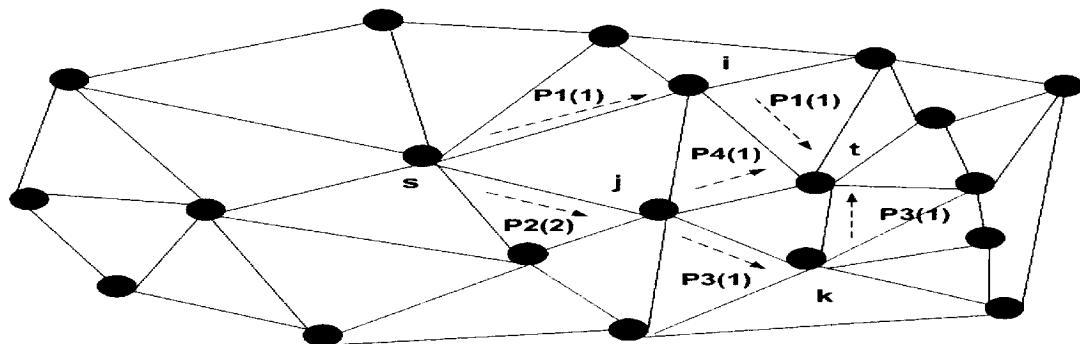


Figure 2.9 Le protocole Ticket Based Protocol

Par ailleurs, le routage avec tickets présente plusieurs avantages :

1. Le surdébit de routage est directement lié au nombre de tickets utilisés. Plus on utilise de tickets, plus on recherche de chemins et plus la probabilité de trouver une route faisable sera élevée. Par contre, le coût de surdébit sera aussi plus élevé;

2. La présence de $\Delta D_i(t)$ et $\Delta B_i(t)$ dans le modèle, le fait tenir compte des imprécisions et cela a un impact direct sur le nombre de tickets assignés;
3. Le processus de routage distribué est utilisé afin d'annuler tout calcul centralisé de chemins, lequel peut être coûteux particulièrement pour les grands réseaux;
4. Maintenir l'information locale et globale dans les nœuds intermédiaires permet de l'utiliser pour diriger les tickets vers des chemins faisables et peu coûteux.

L'inconvénient tient par ailleurs de la complexité du calcul du nombre tickets qui devront être transmis. De plus, le temps de routage peut être relativement long.

Le protocole CEDAR:

Il s'agit d'un protocole réactif avec QdS [20] qui, comme son nom l'indique (*Core-Extraction Distributed Ad hoc Routing algorithm*), se base sur l'élection dynamique d'un cœur de réseau stable.

Les nœuds qui appartiennent au cœur du réseau jouent un rôle particulier. En effet, ils :

- ✓ Limitent la diffusion des messages de contrôle pour n'impliquer qu'un nombre limité de nœuds dans le processus de routage;
- ✓ Propagent de manière efficace l'information requise sur la bande passante disponible dans les liens;
- ✓ Effectuent la découverte de route (tout protocole de routage réactif pourra être utilisé). Celle-ci n'est faite qu'entre les nœuds du cœur.

CEDAR tente de minimiser les collisions dans le canal en utilisant le mécanisme CSMA/CA. De plus, le protocole conserve des chemins secondaires qu'il utilisera si le chemin principal devient désuet. Cependant, l'élection d'un cœur de réseau stable n'est pas toujours possible dans le cas de la forte mobilité des nœuds.

Routage base sur la prédition de la localisation:

Ce protocole, défini dans [21], tente de prédire la localisation des nœuds ainsi que le délai bout-en-bout entre ceux-ci à un instant futur. Par conséquent, on pourra commencer à réparer la route (trouver des chemins alternatifs) avant même que la route de la connexion courante ne devienne désuète.

Afin de pouvoir prédire l'emplacement futur des nœuds, le protocole présuppose que chaque nœud du réseau peut obtenir sa localisation actuelle par GPS.

Dans ce protocole, tout nœud a détient une connaissance topologique de tout le réseau et peut ainsi, trouver un chemin vers tout autre nœud du réseau. Pour ce faire, le nœud a maintient deux table : une *table de routage* et une *table de mise à jour*. La première contient l'information concernant les chemins dont a fait partie. Elle est mise à jour quand un ces chemins est rompu (à cause du mouvement d'un nœud ou du fait que le chemin ne satisfait plus les contraintes de QoS). La seconde sert à stocker les informations pertinentes concernant les nœuds qui transmettent des paquets de mise à jour vers a .

Soit b un nœud qui envoie une mise à jour vers a . Après la réception du paquet, le nœud a ajoute les informations suivantes dans sa table de mise à jour : adresse de b , temps d'envoi du paquet, temps de réception du paquet, les coordonnées du nœud b tel que mentionnées dans le paquet, la vitesse de b tel que mentionnée dans le paquet, les paramètres décrivant l'état des ressources de b et optionnellement la direction du mouvement du nœud b . De plus, la table des mises à jour contient également la liste des nœuds de proximité.

L'avantage de ce protocole est qu'il anticipe les pannes dans le réseau. Par contre, il presuppose que le nœud connaît exactement sa position GPS ce qui peut ne pas être le cas.

Le tableau 2.4 ci-dessous compare les protocoles de routage présentés :

Tableau 2.5 Comparaison des protocoles de routage avec QdS

	Routage initié par la source	Routage initié par la destination	TBP	CEDAR	Routage par prédition de la localisation
Type de routage	Réactif	Réactif	Réactif	Réactif	GPS
Nœud(s) de contrôle	Source	Destination	Source	Cœur stable	Source
Chemins multiples	Non	Oui	Oui	Oui	Non
Contraintes de délai	Oui	Oui	Oui	Oui	Oui
Contraintes bande passante	Oui	Oui	Oui	Oui	Non

2.2.4 Protocoles de signalisation :

Les protocoles de signalisation définissent comment les messages de contrôles seront échangés. Ces messages peuvent contenir des demandes de routes, des informations concernant la topologie du réseau etc.

Le protocole INSIGNIA:

Il s'agit d'un protocole de signalisation dit *in-band* [22] c'est-à-dire que les informations de signalisation sont transmises dans l'en-tête des paquets. Ces informations incluent :

- Le privilège accordé au paquet;
- La qualité de service désirée par l'application;

- L'importance relative du paquet dans le flux de données.
- La demande de réservation de bande passante s'il s'agit du premier paquet du flux.

Ces informations sont initialisées par l'émetteur et mis à jour durant le parcours des paquets pour refléter l'état actualisé du réseau. De plus, des mises à jour sont périodiquement émises par récepteur pour permettre à l'émetteur d'adapter la transmission à l'état des ressources disponibles du réseau.

L'objectif de ce protocole est d'effectuer des réservations de bande passante dans les réseaux MANET. Il n'est lié à aucun protocole particulier de routage et a été conçu pour réagir efficacement et rapidement aux changements topologiques.

INSIGNIA présuppose que les applications s'adaptent selon la disponibilité de la bande passante. En effet, le protocole devra définir deux niveaux de QoS :

- Le niveau de base : qui correspond à la quantité minimum de bande passante nécessaire au bon fonctionnement de l'application;
- Le niveau supérieur : qui correspond à la disponibilité optimale de la bande passante.

Le protocole dQoS / dRSVP:

Ce protocole [23] entend tenir compte de la variabilité des besoins d'une application en terme de qualité de service. En effet, une application peut réviser à la baisse ou à la hausse son besoin en bande passante (par exemple) durant une connexion. Pour tenir compte de cette variabilité, les auteurs proposent que l'application suggère une borne inférieure pour la bande passante qui correspond au fonctionnement minimum et une borne supérieure qui correspond au cas optimal. Au cours de la connexion, la bande passante utilisée peut varier selon les disponibilités des ressources réseau et les besoins ponctuels de l'application.

2.3 Réparation de routes :

De part la nature même des réseaux ad hoc, les routes qui y s'établissent sont fragiles. En effet, les changements de topologie du réseau, principalement dus à la mobilité des nœuds, peuvent faire en sorte que les routes préétablies pour les connexions en cours deviennent obsolètes.

Ainsi, concevoir des protocoles de routage efficaces pour les réseaux MANET va de pair avec l'intégration de mécanismes de réparation de route dans ces protocoles. Par conséquent, il va sans dire que les processus de réparation de routes sont très étroitement liés aux protocoles de routage. Pour preuve, il suffit de se rappeler que plusieurs protocoles étudiés dans la section précédente offrent intrinsèquement des mécanismes de réparation de route. C'est notamment le cas du protocole *Ticket Based Probing* qui tente de conserver en plus de la route principale, des chemins secondaires qui remplaceront la première si celle-ci tombe en panne.

Par ailleurs, pour entreprendre la réparation de la route actuelle, il faut préalablement détecter que celle-ci est tombée en panne. Cela se produit en effet quand un nœud i n'arrive pas à rejoindre le nœud qu'il le suit dans la route courante (détection du bris du lien par la couche de niveau 2). Le nœud i conclut alors que lien a été brisé et déclenche le mécanisme de réparation de route.

Le but de cette section est de présenter et de comparer différents mécanismes de maintenance de route qui existent dans la littérature.

2.3.1 Re-routage global:

Dans le re-routage global, une nouvelle route est recherchée à partir de la source quand l'ancien chemin tombe en panne. Le nœud qui détecte la rupture en avise la source. Par la suite, celle-ci déclenche une nouvelle découverte de route pour trouver un nouveau chemin. Celui-ci doit, bien entendu, satisfaire tous les critères de QoS.

Cette façon de faire présente plusieurs inconvénients :

- Le délai de recherche d'une nouvelle route peut être assez long, ce qui risque de perturber le fonctionnement de l'application particulièrement si celle-ci exige un délai maximum;
- Gaspillage de bande passante causé par les messages de contrôle.

2.3.2 Re-routage partiel:

Il s'agit d'une amélioration par rapport au re-routage global. Il diffère de celui-ci en ce sens que le nœud intermédiaire tente lui-même de réparer la route en essayant de rejoindre la destination. Ainsi, il garde les paquets qu'il reçoit en mémoire et agit comme si il en était la source. En effet, il diffuse une demande de route de la même façon que l'aurait fait la source elle-même. Dès qu'il a une réponse de la destination, il commence à envoyer les paquets en mémoire et la communication continue ainsi. Ici encore, le nouveau chemin entre le nœud intermédiaire et la destination doit répondre aux exigences de la QoS.

Si par contre la réparation de route partielle échoue, alors le nœud intermédiaire avise la source qui fait un re-routage global. Ainsi, ce protocole est plus performant que le précédent si le re-routage partiel réussi (puisque le nœud intermédiaire est en général plus près de la destination que la source) mais moins performant s'il échoue (perte de temps avant le re-routage global).

2.3.3 Re-routage partiel tenant compte de la densité des chemins:

Ce mécanisme proposé par A. Quintero et al. [24], tente d'améliorer le re-routage partiel en privilégiant le choix d'un chemin facilement réparable (peut-être réparé par un re-routage local) lors du routage initial. Pour ce faire, on favorisera les routes formées par des nœuds de grande densité, c'est-à-dire des nœuds avec un nombre maximum de voisins.

Ainsi, on minimisera le risque d'échec du re-routage partiel. Cependant, si malgré tout celui-ci échoue on aura un délai de routage plus important que les deux types de re-routage précédents. En effet, le temps de recherche du chemin initial est plus long pour ce protocole.

2.3.4 Réparation de route localisée (LRR):

Proposé par R. Duggirala, ce mécanisme [25] est basé sur le fait que le plus souvent, une route ad hoc est rompue à cause de la mobilité d'un seul nœud. Ainsi, il est conçu pour faire une réparation localisée à l'endroit de la rupture. Cette zone de rupture est limitée à deux sauts.

Dans ce protocole, chaque nœud i conserve en mémoire le nœud qui est après deux sauts de lui dans la route (nœud j). Si une rupture se produit au niveau du nœud i il tentera rejoindre j en envoyant une demande de route limitée à ses voisins. Si l'un de ceux-ci a j comme voisin direct, alors il répondra à la demande de route de i , la route sera ainsi réparée. Dans le cas contraire, c'est-à-dire si i ne reçoit aucune réponse après un *timeout* alors LRR laissera au protocole adjacent le soin de réparer la route.

Dans cette figure, la route initiale passait par A , B et D . Quand la connexion entre A et B a été rompue à cause de la mobilité de B , LRR a réparé le chemin tel qu'indiqué.

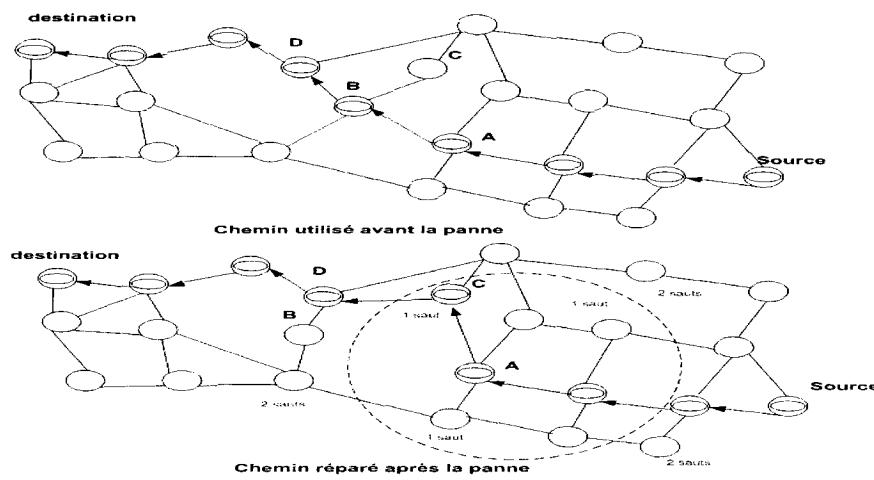


Figure 2.10 Réparation de route LRR

De part leur nature, les réseaux ad hoc présentent des difficultés certaines au niveau du routage. En effet, comme nous l'avons montré tout au long de chapitre, le routage dans ce type de réseau est confronté à la mobilité des nœuds et donc à un réseau à topologie changeante. De plus, de nombreuses applications nécessitent une certaine qualité de service, ce qui représente un nouveau défi pour ce type de réseau. Finalement, la maintenance des routes est très importante dans ce contexte, puisqu'elle permet la reprise de la communication après que la route initiale soit tombée en panne. Toutefois, comme nous l'avons vu, plusieurs solutions ont été développées chacune d'elle tentant de résoudre une partie du problème. Cela rend la comparaison entre les différents protocoles difficile et dépendante du type et de l'environnement dans lequel évolue le réseau ad hoc.

Néanmoins, comme le précise l'auteur dans [26], l'absence de La Solution fait en sorte que le routage dans le réseau ad hoc avec QoS demeure un défi.

CHAPITRE III

PROTOCOLE DE ROUTAGE HYBRIDE AVEC QUALITÉ DE SERVICE

Dans ce chapitre, nous présenterons les détails algorithmiques de la solution proposée. En plus d'assurer le routage dans le contexte dynamique d'un réseau ad hoc, les chemins trouvés par celle-ci devront satisfaire aux critères de qualités de services préalablement établit. Aussi, le protocole proposé devra inclure des mécanismes de réparation de routes lesquels deviennent actifs si un chemin devient invalide en cours d'utilisation entre autres à cause de la mobilité des nœuds.

Notre solution s'inspire du protocole de routage ZRP (*Zone Routing Protocol*) présenté dans le chapitre précédent. Rappelons que ZRP est un protocole de routage hybride proactif / réactif. En effet, ZRP est proactif pour les nœuds relativement rapprochées et réactif pour les nœuds lointains. Ainsi, le critère de pro activité dans le cas de ZRP est la distance qui sépare les différents nœuds qui entrent ou veulent entrer en communication (distance mesurée en nombre de sauts).

La principale différence entre ZRP et le protocole que nous proposons est que ce dernier inclut les critères de qualité de service comme critères de pro activité, le but étant de répondre le plus rapidement possible à une demande de connexion exigeant une certaine qualité de service (délai ou bande passante).

Nous commencerons ce chapitre par faire un rappel du protocole ZRP, proposé par Pearlman et al.[11], lequel est à la base de notre proposition. Dans un second temps, nous présenterons un aperçu de la solution que nous proposons. Après quoi, nous étudierons deux cas de figure : routage exigeant un délai maximum et routage avec contraintes en bande passante. Pour chacun de ces cas, nous présenterons une adaptation de la solution que nous proposons ainsi que des exemples qui illustrent son utilisation. Enfin, nous discuterons de l'intégration de mécanismes de réparation de routes dans notre protocole.

3.1 Le protocole hybride de routage ZRP

Tout comme les protocoles purement proactifs, les protocoles purement réactifs ont des désavantages certains. Le protocole ZRP tente de concilier ces deux philosophies de routage afin de limiter au maximum les faiblesses de chacune. Il serait très important de souligner que ZRP est plus un cadre qui regroupe plusieurs protocoles, qu'un protocole proprement dit.

Force est de constater que dans les réseaux ad hoc, les communications ont en général lieu entre des nœuds (source et destination) relativement rapprochés. Par conséquent, l'impact du départ ou du mouvement d'un nœud se limite le plus souvent à son voisinage. C'est pourquoi le protocole ZRP est basé sur la séparation des nœuds du voisinage de la topologie globale du réseau. Ce voisinage est appelé *zone*. Chaque nœud du réseau en possède une, d'où le chevauchement entre les différentes zones du réseau. Le but de cette séparation est d'utiliser des mécanismes de routage différents à l'intérieur et à l'extérieur de la zone de routage de chaque noeud. Ainsi, le protocole qui régit le routage à l'intérieur d'une zone est différent de celui qui prévaut à l'extérieur de celle-ci. La taille des zones ρ , aussi appelé rayon, correspond aux nombres de sauts vers la périphérie de la zone.

En divisant le réseau en plusieurs zones de tailles possiblement différentes qui se chevauchent, ZRP évite le surdébit relié à la structure hiérarchique par grappe.

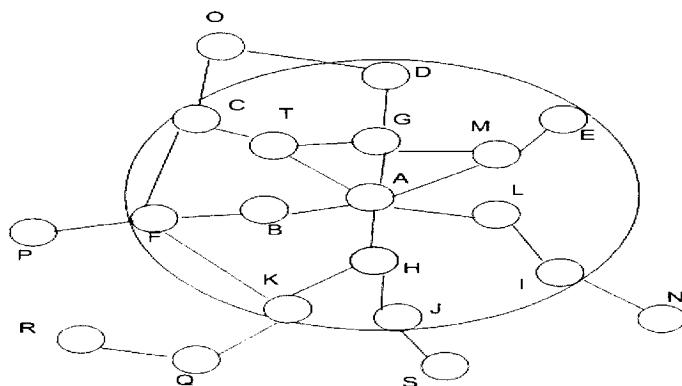


Figure 3.1 Protocole ZRP – zone de routage du nœud A

La figure 3.1 ci-dessus représente la zone de routage du nœud A ($\rho = 2$). À titre d'exemple, il existe plusieurs routes qui relient A à F incluant une route dont la longueur c est supérieure au rayon de la zone ($c > \rho$). Cependant, étant donné qu'il existe une route dont la longueur est inférieure ou égale au rayon de la zone, le nœud F appartient tout de même à la zone de routage de A ($c \leq \rho$).

Le protocole ZRP définit les nœuds de la périphérie comme étant ceux dont la longueur de la route qui les relient à A est égale au rayon de la zone ($c = \rho$) tels que les nœuds J, I et K. Par ailleurs, les nœuds O, P, R, Q, S et N sont à l'extérieur de la zone de routage du nœud A ($c > \rho$).

Il va sans dire que dans un réseau ad hoc, les nœuds mobiles doivent d'abord reconnaître leurs voisins afin de pouvoir constituer leurs zones de routage respectives. Pour ce faire, les nœuds doivent utiliser un protocole de découverte de voisins NDP *Neighbor Discovery Protocol*. Notons que ZRP ne spécifie aucun protocole particulier pour planter NDP. Cela dit, il s'agit généralement d'un protocole de niveau 2. Celui-ci fonctionne en faisant en sorte que chaque nœud envoie régulièrement des messages *HELLO* vers son voisinage. Quand le nœud émetteur reçoit une réponse, il conclut à l'existence d'une connexion directe point-à-point vers son voisin. Le protocole NDP est libre d'établir les critères de son choix pour la sélection des voisins (fréquence du signal, puissance du signal, délai..).

Même après avoir collecter l'information locale nécessaire pour le routage, la diffusion périodique des messages NDP doit être maintenue dans le but de garder à jour l'information qui concerne les voisins.

Si par ailleurs la couche de niveau 2 ne peut planter NDP alors c'est le protocole de routage intrazone *Intrazone Routing Protocol* (IARP) qui devra prendre en charge la découverte de voisins. Le protocole de routage intrazone est un protocole proactif responsable du routage à l'intérieur d'une même zone. Quant au routage à l'extérieur de cette zone, il est pris en charge par le protocole de routage interzone *Interzone Routing Protocol* (IERP). Il s'agit d'un protocole réactif auquel on fait notamment appel quand la destination se trouve à l'extérieur de la zone de routage de la source et que par

conséquent la route n'est pas connue d'avance. Dans ce dernier cas, la demande est transmise aux nœuds de la périphérie qui la prennent en charge.

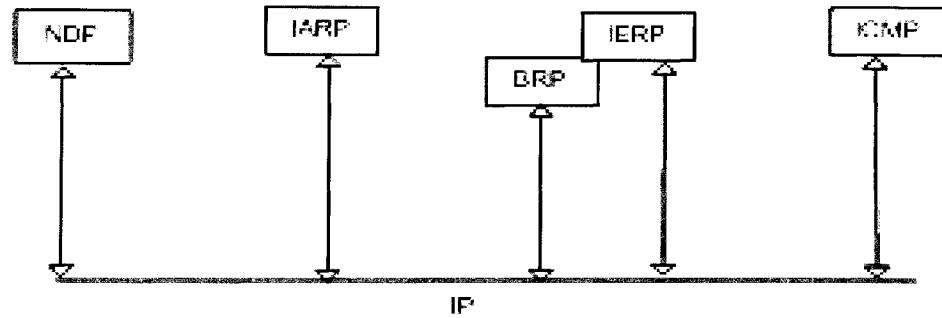


Figure 3.2 Composantes de ZRP

Le protocole ZRP est composé de plusieurs protocoles lesquels ensemble permettent de tirer profits des avantages des protocoles réactifs et proactifs. Cependant, chacune de ces composantes (voir figure 3.2) fonctionne indépendamment des autres et on pourra choisir des technologies différentes pour chacune d'elle afin de maximiser la performance. À titre d'exemple, un protocole proactif comme OLSR pourra être utilisé comme IARP alors qu'un protocole réactif tel que AODV pourra être utilisé comme iERP.

Même si la nature hybride de ZRP semble indiquer qu'il s'agit d'un protocole hiérarchique comme nous l'avions d'ailleurs mentionné au chapitre précédent, certains auteurs considèrent qu'il est plus approprié de considérer ZRP comme un protocole de routage plat. En effet, contrairement à celles des protocoles de routage hiérarchiques, les zones de routage de ZRP se chevauchent. De plus, contrairement aux protocoles hiérarchiques, le protocole ZRP possède autant de zones qu'il y a de nœuds. Ainsi, tous les nœuds du réseau ont la même importance.

3.2 Protocole de routage hybride avec qualité de service

Baptisé HPQS (*Hybrid Protocol with Quality of Service*), le protocole que nous proposons s'appuie sur l'hybridité de ZRP et tente d'y ajouter la prise en charge de deux critères de la qualité de service : le délai et la bande passante.

Comme nous le verrons, dans un cas comme dans l'autre HPQS prend en considération, en plus de la zone de routage initiale (ou interne) de ZRP, une zone de routage dite *étendue* (ou externe) qui elle est constituée à partir des contraintes de qualité de service. Dans ce qui suit, nous détaillerons les algorithmes HPQS aussi bien pour le délai (HPQS-D) que pour la bande passante (HPQS-BP). De plus, nous présenterons des exemples qui illustrent chaque cas.

3.2.1 Routage avec contrainte de délai

Dans le cas du routage avec contrainte de délai, la demande de connexion peut être formulée comme suit : « Trouver un chemin opérationnel entre une source a et une destination b dont le délai total bout-en-bout ne dépasse pas un certain délai D ».

Pour répondre à cette demande, nous proposons d'utiliser le protocole de routage hybride avec QoS (HPQS-D) adapté à la contrainte de délai fonctionnant de la manière suivante:

Chaque nœud mobile i appartenant au réseau ad hoc, conserve pro activement l'information concernant les nœuds qui appartiennent à sa zone proactive de routage appelée $ZP(i)$.

Afin de constituer la zone $ZP(i)$, on doit d'abord trouver les valeurs des variables $d(i,j)$ et $t(i,j)$ définies comme suit:

Soit $d(i,j)$ la distance en nombre de sauts séparant le nœud mobile i et un nœud mobile j quelconque et $t(i,j)$ le délai bout-en-bout minimum possible entre les nœuds i et j . Afin de déterminer les valeurs de $d(i,j)$ et de $t(i,j)$, il suffit de considérer le chemin $C(i,j)$ qui lie les nœuds i et j et dont le délai bout-en-bout est minimum. Dans le cas où il

y aurait deux chemins ou plus liant i et j et dont les délais bout-en-bout sont égaux, on considère alors $C(i,j)$ comme étant le chemin le plus court (plus petit nombre de sauts). Une fois que le chemin $C(i,j)$ a été déterminé, on pose alors les valeurs de $d(i,j)$ et de $t(i,j)$ comme égales aux valeurs successives du nombre de sauts et au délai bout-en-bout du chemin C .

Si par ailleurs il n'existe aucun chemin possible entre les nœuds mobiles i et j , les valeurs de $d(i,j)$ et de $t(i,j)$ seront considérées comme infinies.

Les sous zones $ZP_{interne}(i)$ et $ZP_{externe}(i)$ sont alors constituées à partir des règles suivantes :

- Si pour un nœud j quelconque la distance qui sépare les nœuds mobiles i et j est inférieure ou égale à $DIST_1$ sauts :

$$d(i,j) \leq DIST_1 \quad (3.1)$$

Alors le nœud j appartient à la sous zone $ZP_{interne}(i)$ laquelle correspond à la zone de routage initiale ZRP;

- Si pour un nœud j quelconque la distance entre les nœuds mobiles i et j se situe entre $DIST_1$ et $DIST_2$ sauts et le délai total minimum bout-en-bout entre les deux nœuds i et j est inférieur ou égal à DEL :

$$DIST_1 \leq d(i,j) \leq DIST_2 \quad (3.2)$$

$$\text{Et : } t(i,j) \leq DEL \quad (3.3)$$

Alors le nœud j appartient à la sous zone $ZP_{externe}(i)$ qu'on appellera aussi zone étendue HPQS-D.

Notons que les valeurs de $DIST_1$, $DIST_2$ et DEL sont constantes et déterminées au début de l'utilisation du protocole et peuvent être différentes pour chacun des nœuds du réseau ad hoc. En effet, compte tenu du fait que le réseau ad hoc peut être hétérogène, chacun des nœuds pourra se fixer des valeurs qui tiennent compte de ses capacités (mémoire, CPU, etc...).

Comme nous le verrons plus loin, la zone $ZP(i)$ est constituée à partir des sous zones $ZP_{interne}(i)$ et $ZP_{externe}(i)$ selon la contrainte de délai imposée lors de la demande de connexion.

Notons également que dans tous les cas, si un nœud j appartient à la zone $ZP(i)$, alors tout les nœuds intermédiaires du chemin $C(i,j)$ appartiennent également à $ZP(i)$.

Rappelons que le chemin C correspond toujours au chemin dont le délai est minimum et ce aussi bien pour les nœuds appartenant à la sous zone $ZP_{extreme}(i)$ qu'à ceux appartenant à $ZP_{interne}(i)$.

De plus soulignons que chaque nœud i conserve une table de routage (voir le tableau 3.1 pour un exemple de table de routage HPQS-D).

La figure 3.3 ci-dessous, illustre un exemple de fonctionnement du protocole HPQS-D. Les frontières des zones interne et externe du nœud A sont successivement en trait continu et en trait discontinu. Les valeurs indiquées correspondent au délai de chaque lien.

Les valeurs des variables $DIST_1$, $DIST_2$ et DEL sont successivement 2 sauts, 3 sauts et 20 unités de temps.

Afin de constituer la zone de routage, on calcule la distance et le délai bout-en-bout entre chacun des nœuds et le nœud A. Ce dernier correspond à la somme des délais de chacun des tronçons du chemin C qui lie le dit nœud au nœud A (équation (2.1)).

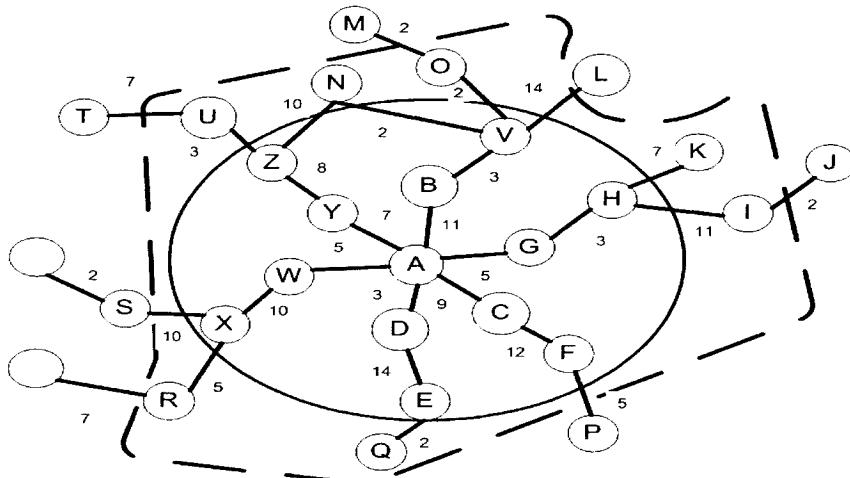


Figure 3.3 Zones de routage initiale et étendue du nœud A (HPQS-D)

Ainsi, par exemple le délai bout en bout entre A et le nœud R est $5 + 10 + 5 = 20$.

De cette manière, nous trouvons que les nœuds B, C, D, E, F, G, H, V, W, X, Y et Z appartiennent à la zone initiale du nœud A puisque la distance qui les sépare de ce dernier est inférieure ou égale à $DIST_1 = 2$ sauts.

De plus, les nœuds R, Q, I, K, O, U et N appartiennent à la zone étendue du nœud A puisque la distance qui les sépare de ce dernier est entre $DIST_1$ et $DIST_2$ (3 sauts) et que le délai bout-en-bout entre ces nœuds et le nœud A est inférieur ou égal à DEL .

Dans le cas particulier du nœud N, celui-ci peut être relié au nœud A à travers les chemins : A->B->V->N de délai 16 et A->Y->Z->N de délai 25. Or, il suffit que le délai bout-en-bout d'un deux chemins soit inférieur ou égal à DEL pour que le nœud N appartienne à la zone étendue du nœud A. Par ailleurs, les nœuds L et P n'appartiennent pas à la zone de routage étendue compte tenu du délai bout-en-bout du chemin qui les lie au nœud A et qui est supérieur à DEL .

Les ensembles $P_{interne}(A) = \{Z, V, H, F, E, X\}$ et $P_{externe}(A) = \{R, Q, I, K, O, N, U\}$ représentent successivement l'ensemble des noeuds de la périphérie des zones interne (initiale) et externe (étendue) du nœud A. Les nœuds appartenant à l'un ou l'autre des deux ensembles jouent, comme nous le verrons un rôle important lors du routage.

Le tableau suivant est un extrait de ce que pourrait être la table de routage HPQS-D d'unité mobile A :

Tableau 3.1 – Exemple de table de routage HPQS-D

Nœud	Chemin C	Nombre de sauts	Délai
B	A->B	1	11
X	A->W->X	2	15
R	A->W->X->R	3	20
D	A->D	1	3
E	A->D->E	2	17
Q	A->D->E->Q	3	19

Revenons maintenant à notre demande : « Trouver un chemin opérationnel entre une source a et une destination b dont le délai total bout en bout ne dépasse pas D ».

La figure 3.4 illustre l’algorithme HPQS-D utilisé pour une demande de connexion entre une source et une destination avec une exigence en matière de délai.

Tel qu’illustré, afin de constituer la zone $ZP(a)$, le nœud source a vérifie si D est inférieure à DEL . Si cette condition est vérifiée alors il pose:

$$ZP(a) = ZP_{interne}(a) \cup ZP_{externe}(a) \quad (3.4).$$

Si par contre cette condition n’est pas vérifiée alors $ZP(a)$ est limitée à $ZP_{interne}(a)$:

$$ZP(a) = ZP_{interne}(a) \quad (3.5).$$

Pour répondre à la demande de connexion, le nœud mobile a commence par consulter sa table de routage, deux cas sont alors possibles :

- Le nœud b appartient à $ZP(a)$;
- Le nœud b n’appartient pas $ZP(a)$.

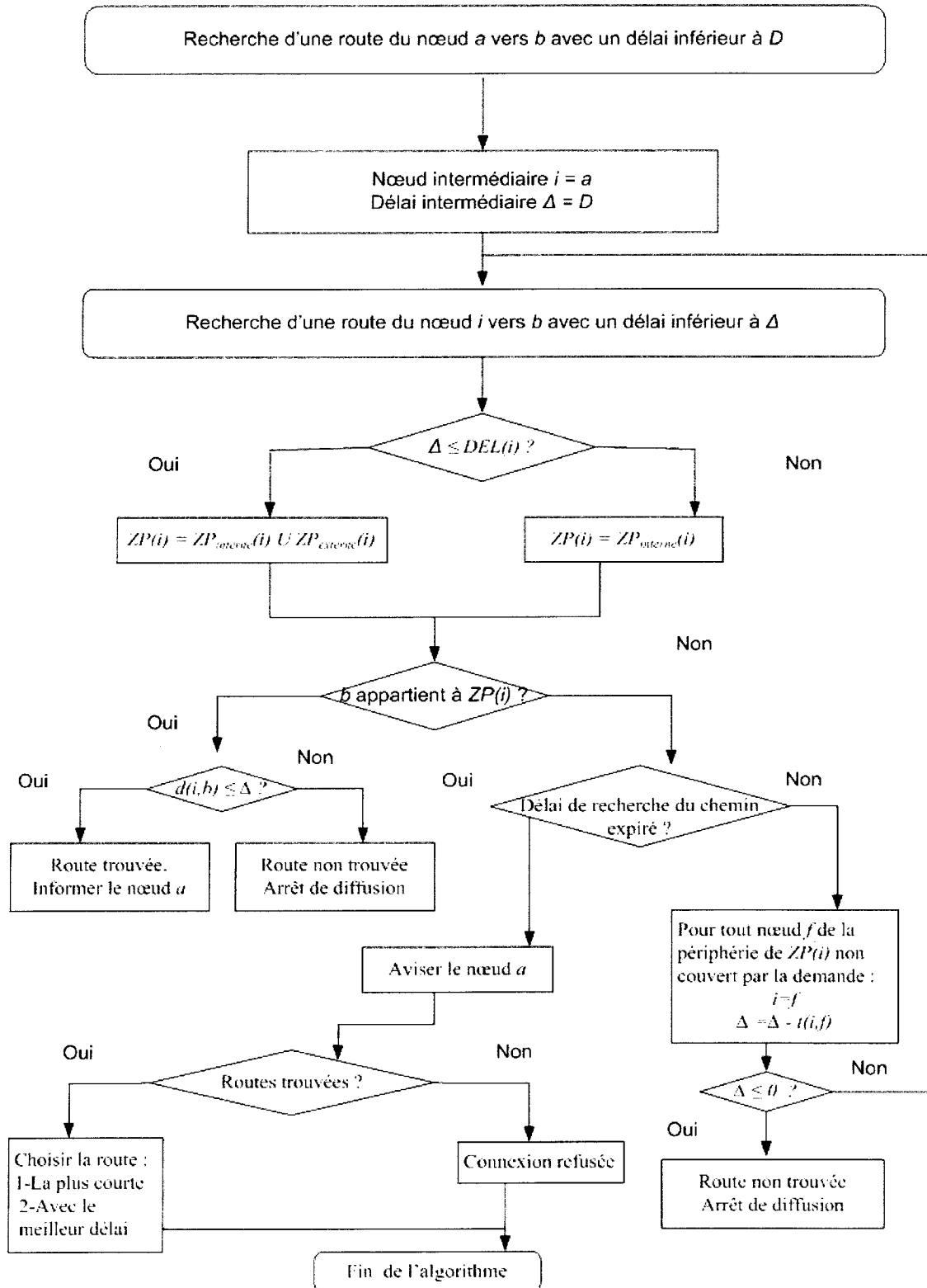


Figure 3.4 Algorithme HPQS-D

Le premier cas est relativement simple. En effet, il suffit que le nœud a vérifie dans sa table de routage si $d(a,b)$ est inférieure ou égale à D . Si c'est le cas alors le chemin C correspond à la route recherchée. Dans le cas contraire, la connexion est refusée puisque le critère de qualité de service (délai) n'a pas été satisfait.

En ce qui concerne le second cas, le nœud a transmet la demande de connexion vers tous les nœuds périphérique f_k qui sont à la frontière de la zone $ZP(a)$. Chacun des nœuds f_k tente de trouver un chemin opérationnel vers le nœud b de sorte que le délai bout-en-bout entre les nœuds f_k et b soit inférieur ou égale à $D - t(a,f)$:

$$t(f_k, b) \leq D - t(a,f) \quad (3.6)$$

Pour ce faire, chaque nœud frontière agit comme s'il était l'initiateur de la demande. La recherche continue ainsi jusqu'à ce qu'une route soit trouvée ou qu'un temps limite (TTL) soit atteint.

À la fin de l'algorithme, le nœud a reçoit les différentes routes qui ont été trouvées. Il en choisit une et ce en appliquant dans l'ordre les critères suivants :

- La route la plus courte : Notre algorithme priorise les routes courtes puisque celles-ci seront en général plus robustes. En effet, vu qu'il y a moins de nœuds, la probabilité que le chemin devienne non fonctionnel est moindre ;
- La route dont le délai est moins important : En effet, de cette manière, nous aurons plus d'assurance quant à la satisfaction du critère de qualité de service.

Pour illustrer le fonctionnement de HPQS-D, nous proposons l'exemple du réseau de la figure 3.5, où le nœud A tente de communiquer avec le nœud Z avec un délai bout-en-bout inférieur à 50 unités de temps.

Dans le but de simplifier, nous considérerons que les valeurs des variables $DIST_1$, $DIST_2$ et DEL sont identiques pour tous les nœuds du réseau et égales successivement à 2 sauts, 3 sauts et 30 unités de temps. De plus, nous considérons tous les liens du réseau comme bidirectionnels.

Le nœud A commence par initier la requête. Il constate que le délai exigé par la connexion est supérieur à DEL ($50 > 30$). Donc, pour cette demande le nœud A considère uniquement sa zone de routage interne pour cette demande.

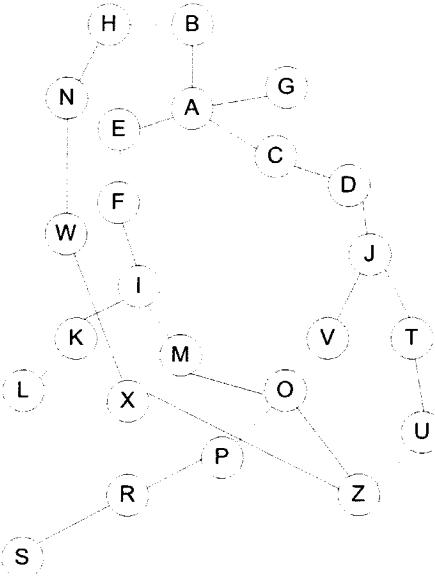


Figure 3.5 Exemple HPQS-D (Topologie globale du réseau)

Ceci explique pourquoi la limite de la zone de routage externe est en lignes pointillées dans la figure 3.6. De plus, puisque le nœud Z n'appartient pas à la zone de routage du nœud A, ce dernier doit donc transmettre la demande aux nœuds qui sont à la périphérie de sa zone de routage (nœuds D, F, G, H) en mettant à jour le délai tel que décrit dans l'algorithme.

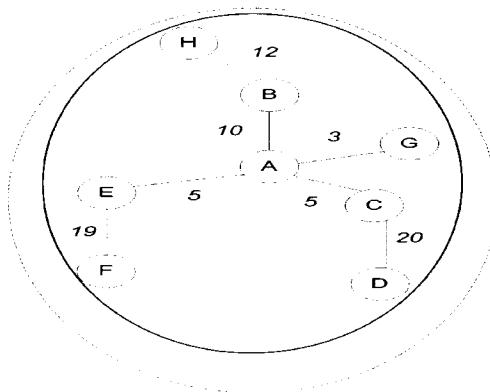


Figure 3.6 Exemple HPQS-D (Zones du nœud A)

Ainsi, les nœuds D, F, G et H se retrouvent avec une demande de connexion vers Z avec un délai inférieur ou égale successivement à 25, 26, 47 et 28 unités de temps.

Pour que ce qui est du nœud G, étant donné que le nœud A est son unique voisin, la diffusion s'arrête à son niveau.

Le nœud H quant à lui retransmet la demande au nœud unique appartenant à sa zone de routage et non encore couvert : le nœud N. Cependant, étant donné que la contrainte de délai mise à jour est négative, la diffusion de la requête s'arrête au nœud N.

En ce qui concerne le nœud F, celui-ci reçoit la requête et forme sa zone de routage (figure 3.7). Puisque le nœud F reçoit une demande de connexion avec un délai inférieur ou égal à 26 unités, la zone de routage de F inclut également la zone de routage externe. Ensuite, le nœud F transmet la requête vers les nœuds de la périphérie non marquée : le noeud O avec un délai de 10 et le nœud L avec un délai de 2. Étant donné que le nœud L n'a pas de voisin non marqué, la diffusion s'arrête à son niveau.

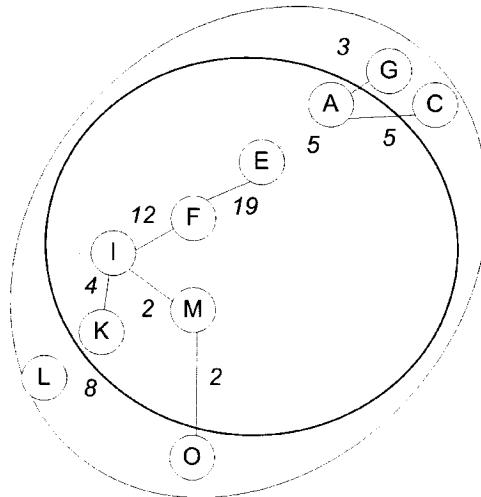


Figure 3.7 Exemple HPQS-D (Zones du nœud F)

Quant au nœud O, il construit sa zone de routage tel que montré dans la figure 3.8. Celle-ci inclut également la zone externe. De plus, le nœud O s'aperçoit que la destination Z appartient à sa zone de routage. Une première route est constituée et le nœud source A en est avisé.

Par ailleurs, le nœud D construit sa zone de routage qui inclut la zone externe de routage (figure 3.9). Ensuite, il transmet la requête aux nœuds de la périphérie non marquée : le nœud V avec un délai de 9 unités et le nœud U avec un délai de 12 unités.

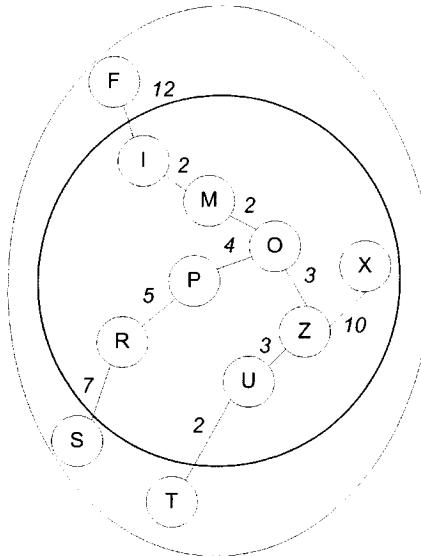


Figure 3.8 Exemple HPQS-D (Zones du nœud O)

Une fois encore étant donné que le nœud V n'a pas de voisin non marqué, la diffusion s'arrête à son niveau. Pour ce qui est du nœud U celui-ci sa zone de routage et s'aperçoit que la destination Z y appartient. Une seconde route est ainsi formée et le nœud source A en est avisé.

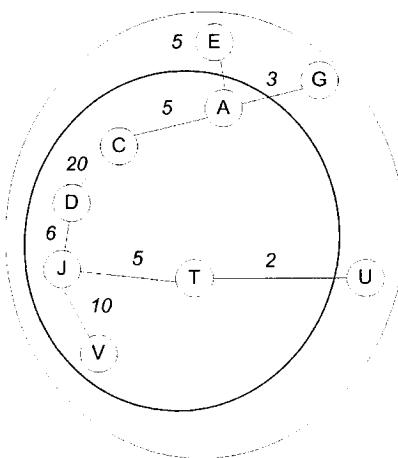


Figure 3.9 Exemple HPQS-D (Zones du nœud D)

Le nœud A se retrouve donc avec deux routes possibles :

- ✓ La route A->E->F->I->M->O->Z avec un délai bout en bout de 43 unités ;
- ✓ La route A->C->D->J->T->U->Z avec un délai bout en bout de 41 unités.

Étant donné que la longueur des deux chemins est identique (6 sauts), le nœud A choisit le chemin ayant le meilleur délai (route **A->C->D->J->T->U->Z**). Le nœud A envoie alors un message de réservation à travers la route trouvée. La connexion est désormais établie.

Notons par ailleurs que le chemin A->H->N->W->X->Z n'a pas été considéré par notre algorithme puisqu'il ne satisfait pas aux contraintes de délai.

3.2.2 Routage avec contrainte de bande passante:

Ce cas est semblable à celui de routage avec contrainte de délai. En effet, la demande de connexion peut être formulée comme suit : « Trouver un chemin opérationnel entre une source a et une destination b avec une bande passante bout-en-bout supérieure ou égale à B ».

Pour répondre à cette demande, nous proposons d'utiliser le protocole de routage hybride avec QdS (HPQS-BP) adapté à la contrainte de bande passante fonctionnant de la manière suivante:

Chaque nœud mobile i appartenant au réseau ad hoc, conserve proactive activement l'information concernant les nœuds qui appartiennent à sa zone proactive de routage appelée $ZP(i)$.

Afin de constituer la zone $ZP(i)$, il faut d'abord trouver les valeurs des variables $d(i,j)$ et de $p(i,j)$ elles-mêmes définies comme suit:

Soit $d(i,j)$ la distance séparant le nœud mobile i et un nœud mobile j quelconque et $p(i,j)$ la bande passante bout-en-bout maximum possible entre les nœuds i et j . Afin de déterminer les valeurs de $d(i,j)$ et de $p(i,j)$, il suffit de considérer le chemin $C(i,j)$ qui lie les nœuds i et j avec une bande passante maximum. Dans le cas où il y aurait deux

chemins ou plus liant i et j avec la même bande passante maximum, on considère alors $C(i,j)$ comme étant le chemin le plus court (plus petit nombre de sauts). Une fois que le chemin $C(i,j)$ a été déterminé, on pose alors les valeurs de $d(i,j)$ et de $p(i,j)$ comme égales aux valeurs successives du nombre de sauts et la bande passante bout-en-bout du chemin C .

Si par ailleurs il n'existe aucun chemin possible entre les nœuds mobiles i et j , la valeur de $d(i,j)$ sera considérée comme infinie et celle de $p(i,j)$ comme étant nulle.

Ainsi, les sous zones $ZP_{interne}(i)$ et $ZP_{externe}(i)$ sont constitués à partir des règles suivantes :

- Si pour un nœud j quelconque la distance qui sépare les nœuds mobiles i et j est inférieure ou égale à $DIST_1$ sauts :

$$d(i,j) \leq DIST_1 \quad (3.7)$$

Alors le nœud j appartient à la sous zone $ZP_{interne}(i)$ laquelle correspond à la zone de routage initiale ZRP;

- Si pour un nœud j quelconque la distance entre les nœuds mobiles i et j se situe entre $DIST_1$ et $DIST_2$ sauts et la bande passante bout-en-bout entre les deux nœuds i et j est supérieure ou égale à BP :

$$DIST_1 \leq d(i,j) \leq DIST_2 \quad (3.8)$$

$$\text{Et : } p(i,j) \geq BP \quad (3.9)$$

Alors le nœud j appartient à la sous zone $ZP_{externe}(i)$ qu'on appellera aussi zone étendue HPQS-BP.

Notons que les valeurs de $DIST_1$, $DIST_2$ et BP sont constantes et déterminées au début de l'utilisation du protocole et peuvent être différentes pour chacun des nœuds du réseau ad hoc.

En effet, compte tenu du fait que le réseau ad hoc peut être hétérogène, chacun des nœuds pourra se fixer des valeurs qui tiennent compte de ses capacités (mémoire, CPU, etc...).

Comme nous le verrons plus loin, la zone $ZP(i)$ est constituée à partir des sous zones $ZP_{interne}(i)$ et $ZP_{externe}(i)$ selon la contrainte de bande passante imposée lors de la demande de connexion.

Notons également que dans tous les cas, si un nœud j appartient à la zone $ZP(i)$, alors tout les nœuds intermédiaires de $C(i,j)$ appartiennent à $ZP(i)$.

Rappelons que le chemin C correspond toujours au chemin dont la bande passante est maximum et ce aussi bien pour les nœuds appartenant à la sous zone $ZP_{externe}(i)$ qu'à ceux appartenant à $ZP_{interne}(i)$.

De plus soulignons que chaque nœud i conserve une table de routage (voir le tableau 3.2 pour un exemple de table de routage HPQS-BP).

La figure 3.10 ci-dessous, illustre un exemple de fonctionnement du protocole HPQS-BP. Les frontières des zones interne et externe du nœud A sont successivement en trait continu et en trait discontinu. Les valeurs indiquées correspondent à la bande passante de chaque lien.

Les valeurs de $DIST_1$, $DIST_2$ et BP correspondent successivement à 2 sauts, 3 sauts et 25 unités de bande passante. Afin de constituer la zone de routage, on calcule la distance et la bande passante bout-en-bout entre chacun des nœuds et le nœud A. Cette dernière correspond au minimum des bandes passantes de chacun des tronçons du chemin C qui lie le dit nœud au nœud A (équation (2.2)).

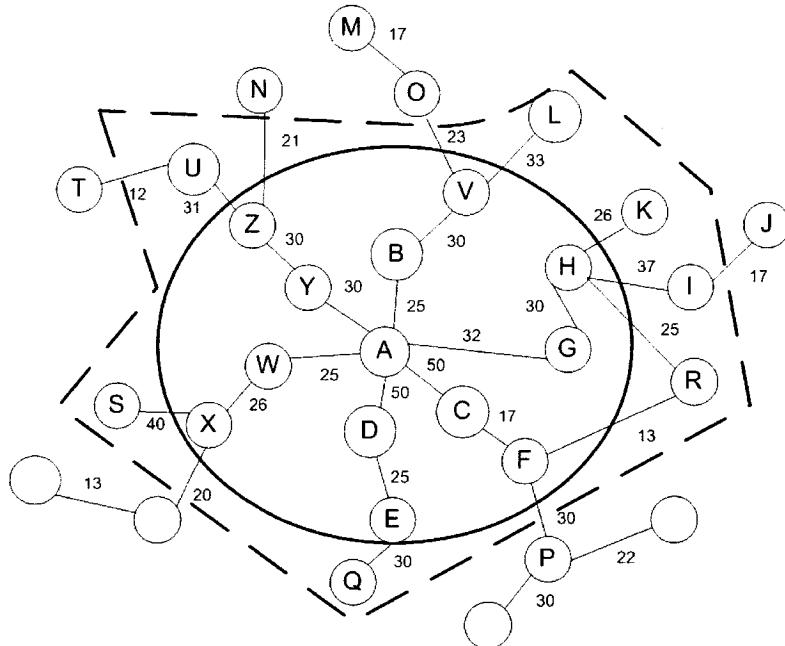


Figure 3.10 Zones de routages initiale et étendue du nœud A (HPQS-BP)

Ainsi, par exemple la bande passante bout-en-bout entre A et le nœud O est $\min(25, 30, 23) = 23$.

De cette manière nous trouvons que les nœuds B, C, D, E, F, G, H, V, W, X, Y et Z appartiennent à la zone initiale du nœud A puisque la distance qui les sépare de ce dernier est inférieure ou égale à $DIST_1 = 2$ sauts. De plus, les nœuds U, R, L, K, I, Q et S appartiennent à la zone étendue du nœud A puisque la distance qui les sépare de ce dernier est entre $DIST_1$ et $DIST_2$ (3 sauts) et que la bande passante bout-en-bout entre ces nœuds et le nœud A est supérieure ou égale à BP . Dans le cas particulier du noeud R, celui-ci peut être relié au nœud A à travers les chemins : A->G->H->R de bande passante 25 et A->C->F->R de bande passante 13. Or, il suffit que la bande passante bout-en-bout d'un deux chemins soit supérieure ou égale à BP pour que le nœud R appartienne à la zone étendue du nœud A.

Par ailleurs, les nœuds O et P n'appartiennent pas à la zone de routage étendue compte tenu que la bande passante bout-en-bout du chemin qui les lient au nœud A et qui est inférieure à BP .

Les ensembles $P_{interne}(A) = \{Z, V, H, F, E, X\}$ et $P_{externe}(A) = \{S, U, L, I, R, Q, K\}$ représentent successivement l'ensemble des nœuds de la périphérie des zones interne (initiale) et externe (étendue) du nœud A. Les nœuds appartenant à l'un ou l'autre des deux ensembles jouent, comme nous le verrons, un rôle important lors du routage.

Le tableau suivant est un extrait de ce que pourrait être la table de routage HPQS-BP l'unité mobile A :

Tableau 3.2 – Exemple de table de routage HPQS-BP

Nœud	Chemin C	Nombre de sauts	Bandé passante
B	A ->B	1	25
X	A->W->X	2	25
D	A->D	1	50
E	A->D->E	2	25
Q	A->D->E->Q	3	25

Revenons maintenant à notre demande : « Trouver un chemin opérationnel entre une source a et une destination b dont la bande passante est supérieure ou égale à B ».

La figure 3.11 illustre l'algorithme HPQS-BP utilisé pour une demande de connexion entre une source et une destination avec une exigence en matière de bande passante.

Tel qu'illustré, afin de constituer la zone $ZP(a)$, le nœud source a vérifie si B est supérieure à BP . Si cette condition est vérifiée alors il pose:

$$ZP(a) = ZP_{interne}(a) \cup ZP_{externe}(a) \quad (3.10).$$

Si par contre cette condition n'est pas vérifiée alors $ZP(a)$ est limitée à $ZP_{interne}(a)$:

$$ZP(a) = ZP_{interne}(a) \quad (3.11).$$

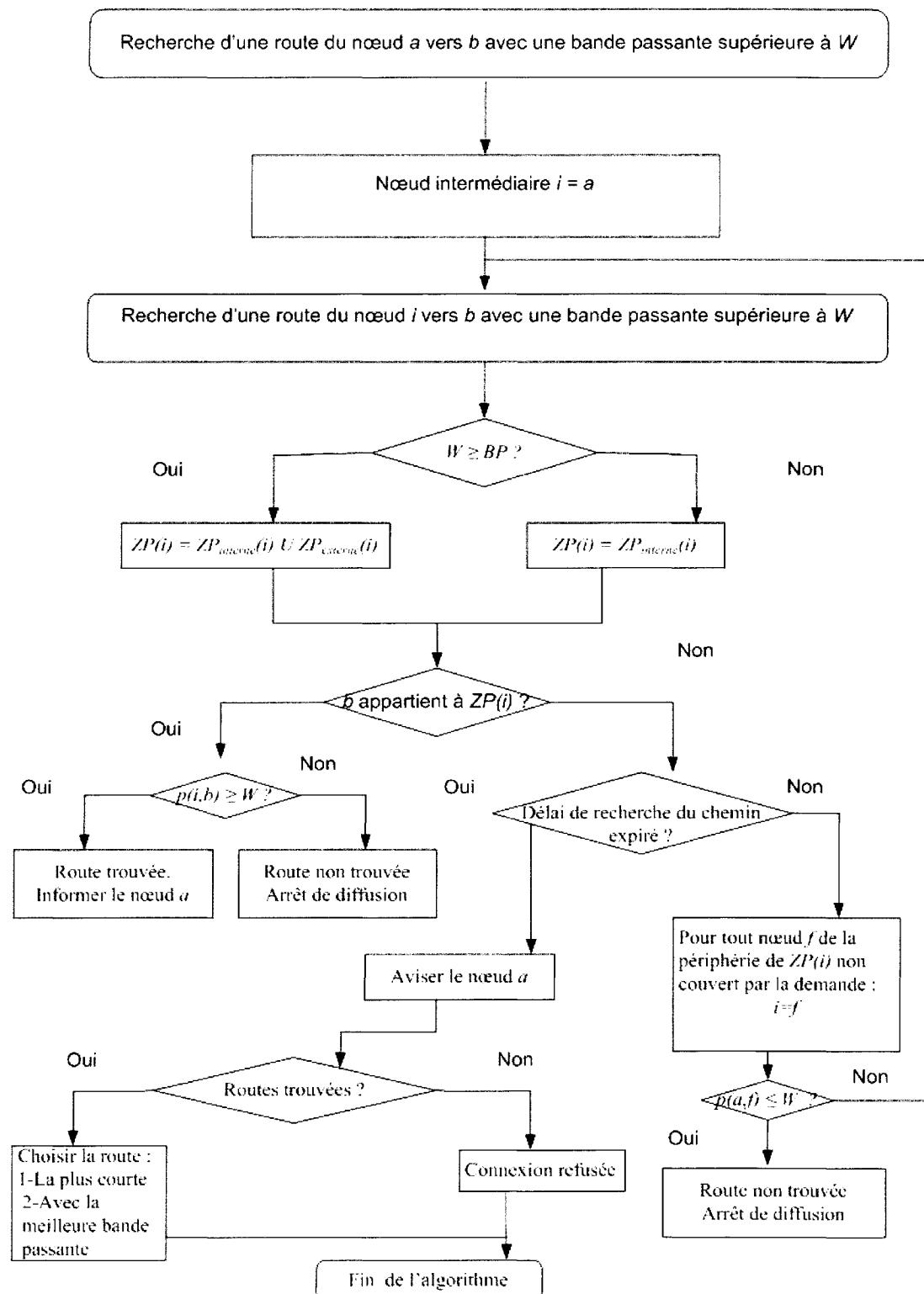


Figure 3.11 Algorithme HPQS-BP

Pour répondre à la demande de connexion, le nœud mobile a commence par consulter sa table de routage, deux cas sont alors possibles :

- Le nœud b appartient à $ZP(a)$;
- Le nœud b n'appartient pas $ZP(a)$.

Le premier cas est relativement simple. En effet, il suffit que le nœud a vérifie dans sa table de routage si $p(a,b)$ est supérieure ou égale à B . Si c'est le cas alors le chemin C correspond à la route recherchée. Dans le cas contraire, la connexion est refusée puisque le critère de qualité de service (bande passante) n'a pas été satisfait.

En ce qui concerne le second cas, le nœud a transmet la demande de connexion vers tous les nœuds (périphériques) f_k qui sont à la frontière de la zone $ZP(a)$. Chacun des nœuds f_k tente de trouver un chemin opérationnel vers le nœud b de sorte que la bande passante bout-en-bout entre les nœuds f_k et b soit supérieure ou égale à B :

$$p(b,f_k) \geq B \quad (3.12)$$

Pour ce faire, chaque nœud frontière agit comme s'il était l'initiateur de la demande. La recherche continue ainsi jusqu'à ce qu'une route soit trouvée ou qu'un temps limite (TTL) soit atteint.

À la fin de l'algorithme, le nœud a reçoit les différentes routes qui ont été trouvées. Il en choisit une et ce en appliquant dans l'ordre les critères suivants :

- La route la plus courte : Notre algorithme priorise les routes courtes puisque celles-ci seront en général plus robustes. En effet, vu qu'il y a moins de nœuds, la probabilité que le chemin devienne non fonctionnel est moindre ;
- La route dont la bande passante est la plus importante : En effet, de cette manière, nous aurons plus d'assurance quant à la satisfaction du critère de qualité de service.

Pour illustrer le fonctionnement de HPQS-BP, nous proposons l'exemple du réseau de la figure 3.12 où nœud A tente de communiquer avec le nœud Z avec une bande passante bout-en-bout supérieure ou égale à 50 unités.

Dans le but de simplifier, nous considérerons que les valeurs des variables $DIST_1$, $DIST_2$ et BP sont identiques pour tous les nœuds du réseau et égales successivement à 2 sauts, 3 sauts et 40 unités bande passante. De plus, nous considérons tous les liens du réseau comme bidirectionnels.

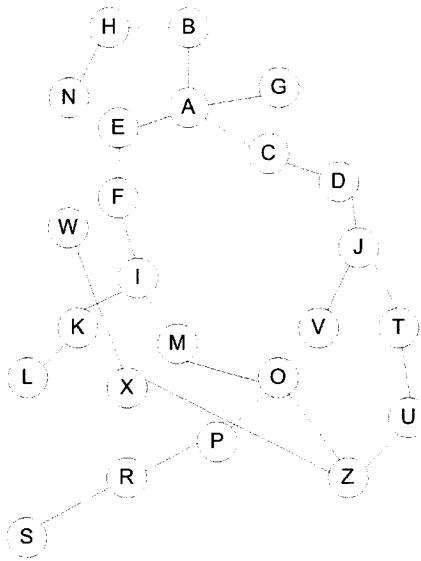


Figure 3.12 Exemple HPQS-BP (Topologie globale du réseau)

Le nœud A commence par initier la requête. Il constate que la bande passante exigée par la connexion est supérieure à BP ($50 > 30$). Donc, la zone de routage de A inclut la zone de routage externe (figure 3.13). De plus, puisque le nœud Z n'appartient pas à la zone de routage du nœud A, ce dernier doit transmettre la demande aux nœuds qui sont à la périphérie de sa zone de routage (nœuds I, J, G, H).

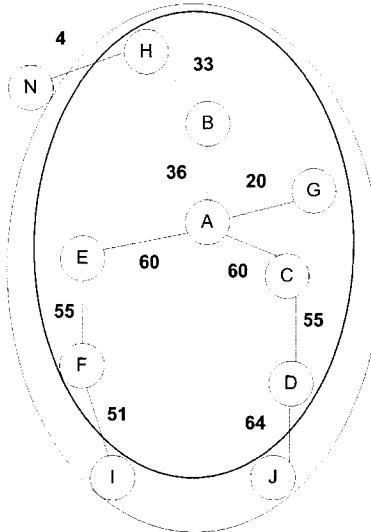


Figure 3.13 Exemple HPQS-BP (Zones du nœud A)

Ainsi, les nœuds I, J, G et H se retrouvent avec une demande de connexion vers Z avec une bande passante supérieure ou égale à 50 unités.

Pour que ce qui est du nœud G, étant donné que le nœud A est son unique voisin, la diffusion s'arrête à son niveau.

Le nœud H n'a pas à retransmettre la demande. En effet, la bande passante bout-en-bout de A vers H est de 33 unités, valeur inférieure à la bande passante exigée.

Quant au nœud I, il construit sa zone de routage tel que montré dans la figure 3.14. Celle-ci inclut également la zone externe pour les raisons expliquées précédemment. Le nœud I s'aperçoit que la destination Z appartient à sa zone de routage. Une première route est constituée et le nœud source A en est avisé.

De même, le nœud J construit sa zone de routage qui inclut la zone externe (figure 3.15).

De plus, étant donné que le nœud destination Z y appartient, une seconde route est formée et le nœud A en est avisé.

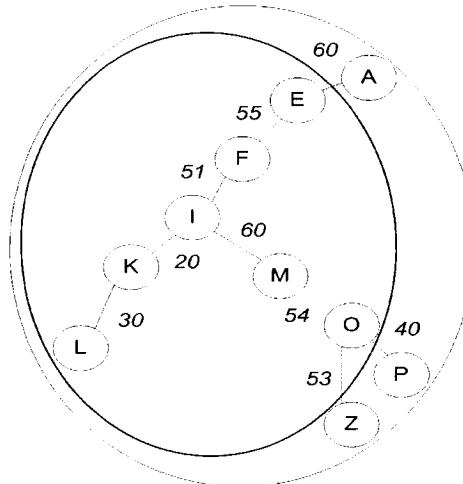


Figure 3.14 Exemple HPQS-BP (Zones du nœud I)

Le nœud A se retrouve donc avec deux routes possibles :

- ✓ La route A->E->F->I->M->O->Z avec une bande passante bout-en-bout de 51 unités ;
- ✓ La route A->C->D->J->T->U->Z avec une bande passante bout-en-bout de 55 unités.

Étant donné que la longueur des deux chemins est identique (6 sauts), le nœud A choisit le chemin ayant la meilleure bande passante. La route **A->C->D->J->T->U->Z** est donc choisie.

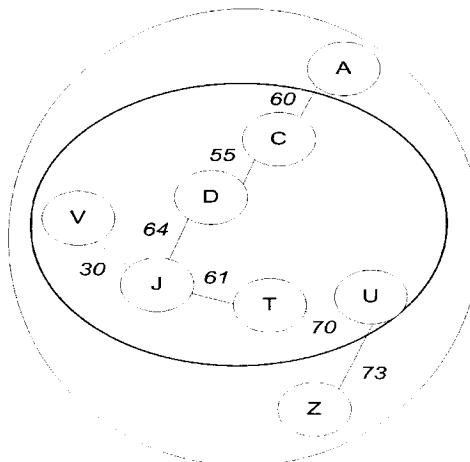


Figure 3.15 Exemple HPQS-BP (Zones du nœud J)

Notons par ailleurs que le chemin A->H->N->W->X->Z n'a pas été considéré par notre algorithme puisqu'il ne satisfait pas aux contraintes de bande passante.

3.3 Réparation de route avec HPQS:

Dans un réseau ad hoc, même après trouvé une route entre la source et la destination, celle-ci peut devenir désuète à cause principalement de la mobilité d'un des nœuds qui la forment. Dans ce cas, un mécanisme de la réparation de route doit être activé afin de remédier à la situation.

Dans le chapitre précédent, nous avons vu qu'il avait plusieurs techniques de réparation de route mais que ceux-ci peuvent être classés en deux catégories principales :

- Re-routage global : où c'est la source qui a pour mandat de trouver la nouvelle route;
- Re-routage local : où c'est le nœud à partir duquel la route a été brisée qui cherche une route alternative. Si toutefois cette tentative s'avère infructueuse, alors un re-routage global sera nécessaire.

Ainsi, bien que le re-routage local nécessite à priori moins de temps que le re-routage global, ce dernier devient plus intéressant dans le cas d'échec du premier. En effet, si le re-routage local réussit, alors il y aura un gain en terme de temps de découverte de route par rapport au re-routage global. Cependant, si le re-routage local échoue alors il y aura perte par rapport au re-routage global puisque ce dernier sera finalement utilisé précédé d'une tentative de réparation locale de route infructueuse et coûteuse en terme de temps de calcul.

Le mécanisme de réparation de route inclut dans le protocole HPQS hérite ses avantages de celui de ZRP [30].

Ce mécanisme tente de minimiser le temps nécessaire pour savoir si une réparation locale de route portera fruits ou non. Pour ce faire, nous nous appuyons sur le concept de

zone de routage HPQS précédemment détaillé. Ainsi, chacun des nœuds qui constituent la route conserve en mémoire les prochains sauts du chemin à l'intérieur de sa zone de routage.

Considérons une connexion établie entre un nœud source N_s et un nœud destination N_d à travers la route suivante :

$$N_s \rightarrow N_1 \rightarrow N_2 \rightarrow \dots \rightarrow N_{i-1} \rightarrow N_i \rightarrow N_{i+1} \rightarrow N_{i+2} \rightarrow \dots \rightarrow N_{i+j} \rightarrow \dots \rightarrow N_d$$

Si le lien $N_i \rightarrow N_{i+1}$ est rompu en cours d'utilisation, alors c'est le nœud N_i qui procède à la réparation locale de route. Pour ce faire, il commence par reconstruire sa zone de routage, selon les mêmes critères que si il avait reçu la demande du nœud N_{i-1} . En supposant que le nœud N_{i+j} correspond au dernier saut conservé en mémoire par le nœud N_i , ce dernier tente de trouver une route à l'intérieur de sa zone de routage vers l'un des nœuds ($N_{i+2}, N_{i+3}, \dots, N_{i+j}$) qui vérifie les deux conditions suivantes :

- ✓ Peut remplacer le lien brisé $N_i \rightarrow N_{i+1}$;
- ✓ Les exigences de QoS (délai, bande passante) restent respectés bout-en-bout de N_s vers N_d après le remplacement du lien brisé.

Notons que cette vérification peut se faire d'une manière quasi-instantanée puisque il s'agit d'une vérification que le nœud N_i peut faire en consultant sa table de routage locale.

Si aucune route de remplacement n'est trouvée alors un re-routage global sera nécessaire.

La figure 3.16 illustre le mécanisme de réparation de route HPQS. Avant la panne, le chemin établit entre la source et la destination incluait entre autres, le tronçon $A \rightarrow B \rightarrow D$. Considérant que la taille de la zone de routage du nœud A était de 2 sauts, celui-ci doit garder en mémoire les nœuds B et D .

Suite au déplacement du nœud B , le lien $A \rightarrow B$ devient inutilisable. Le nœud A déclenche alors le protocole de réparation de route HPQS. Pour ce faire, le nœud A tente de trouver un tronçon de route à l'intérieur de sa zone de routage (délimitée en gras) pour joindre le nœud D . En plus de remplacer $A \rightarrow B \rightarrow D$, ce tronçon devra répondre aux

exigences QoS. Si le tronçon A->C->D satisfait à ces deux conditions, on pourra considérer que le ré-routage local s'est terminé avec succès.

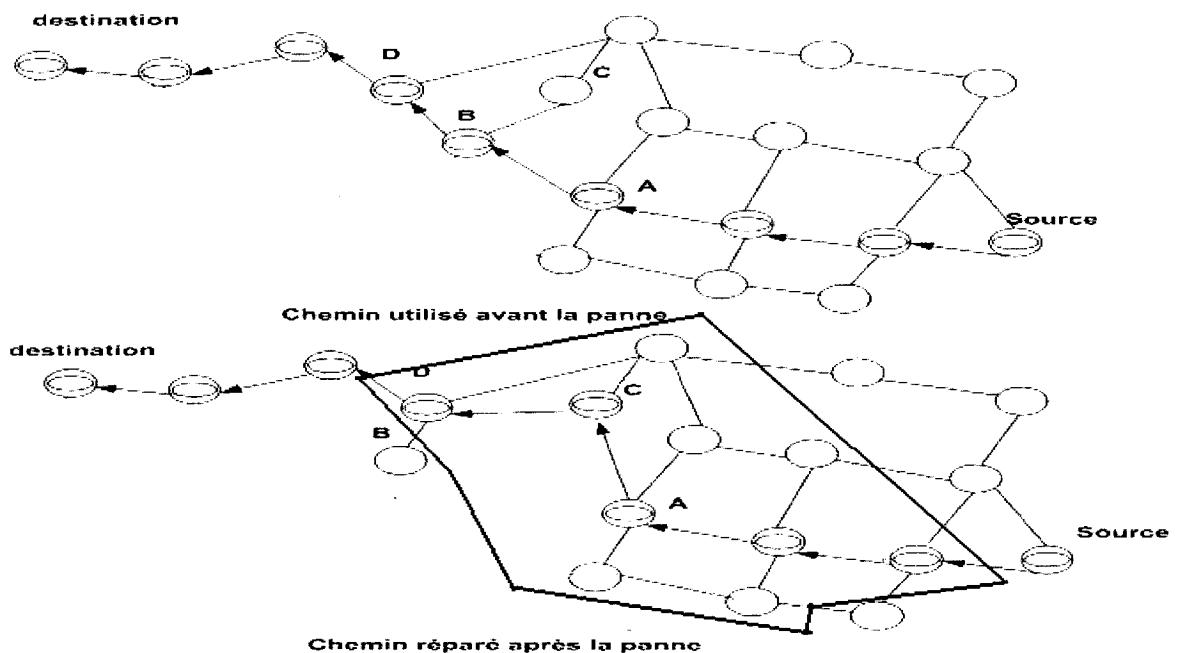


Figure 3.16 Réparation de route HPQS

CHAPITRE IV

IMPLÉMENTATION, SIMULATION ET ANALYSE DE PERFORMANCE

Pour faire suite au chapitre précédent où nous avons détaillé notre solution, laquelle consiste en un protocole de routage hybride avec qualité de service pour réseaux ad hoc, nous nous proposons dans ce chapitre de procéder à l'implémentation et à l'évaluation de performance du protocole proposé.

Étant donné que l'implémentation réelle est irréalisable dans notre contexte et que la vérification formelle semble inadaptée au cas présent, la simulation demeure la meilleure option pour l'évaluation de performance dans notre cas.

Ainsi, nous commencerons par présenter l'environnement ainsi que le simulateur sur lesquels nous travaillerons. Ensuite, nous présenterons les détails de l'implémentation : structures de données et algorithmes. Dans un troisième temps, nous poursuivrons avec notre plan d'expérience ainsi que les différents scénarios qui lui sont reliés.

Enfin, nous terminerons ce chapitre en présentant et analysant les résultats obtenus et en les comparants à ceux des meilleurs protocoles et mécanismes présents dans la littérature.

4.1 Environnement de simulation

Il existe plusieurs simulateurs supportant les réseaux ad hoc mobiles. Chacun d'eux possède ces propres caractéristiques. On peut citer entre autres :

- ❖ **NS-2 [31]:** Simulateur développé par plusieurs universités dont : *USC – University of Southern California et Berkeley University*. Il est codé en C++ et OTCL et disponible gratuitement sur Internet;

- ❖ **OPNET [32]:** Logiciel commercial de modélisation et de simulation de réseaux largement utilisé. Il supporte de nombreux type de réseaux dont les réseaux mobiles ad hoc;
- ❖ **GloMoSim [33] :** Simulateur développé par *University of California at Los Angeles*. Il a été codé sous *Parsec* qui est un langage basé sur le C et facilitant la prise en charge de l'exécution séquentielle et parallèle des modèles de simulations à événements discrets. Le fonctionnement de GloMoSim est basé sur les couches du modèle OSI. Ce simulateur supporte aussi bien les réseaux filaires que les réseaux mobiles sans fil;
- ❖ **Qualnet [34]:** Il s'agit de la version commerciale de GloMoSim développée par *Scalable Network Technologies*. Plusieurs modules et fonctionnalités ont été ajoutés dont notamment : le support des réseaux WiFi et des réseaux satellite ainsi qu'une meilleure interface graphique. Le simulateur Qualnet est codé en C++ et tout comme GloMoSim, il est conçu pour la simulation parallèle (*Parsec*). Une présentation plus détaillée de Qualnet sera faite dans les lignes qui suivent.

Nous pensons, après comparaison, que Qualnet répond mieux à nos besoins d'une part parce qu'il est adapté pour la conception de protocoles de routages pour les réseaux ad hoc et d'autre part, comme nous le verrons plus loin, pour le grand nombre d'outils et de possibilités qu'il offre.

4.1.1 L'environnement de simulation :

Nous présentons ici l'environnement de simulation utilisé. Le tableau 4.1 ci-dessous résume les spécifications techniques de cet environnement :

Tableau 4.1 – Spécifications de l'environnement de simulation

Microprocesseur	Intel Pentium IV – HT
Horloge du microprocesseur	3.2 GHz
Mémoire vive	512 Mo
Système d'exploitation	Microsoft Windows XP Professional – SP1
Environnement de programmation	Microsoft Visual C++
Outil de compilation	<i>nmake</i>
Préalable à l'utilisation de Qualnet	Java 2 SDK 1.4.0
Version de Qualnet utilisée	3.8

L'exécutable *qualnet.exe* sera généré et c'est à partir de celui-ci que nous effectuerons nos simulations.

4.1.2 Présentation du simulateur Qualnet

La version 3.8 du simulateur Qualnet regroupe principalement les éléments suivants :

- **Qualnet Designer** : Outil permettant de créer graphiquement de nouveaux protocoles. Il peut s'agir aussi bien de protocoles du niveau d'application, du niveau transport, du niveau liaison ou encore des protocoles de routage. Chaque protocole est conçu comme une machine à états finis. Cet outil peut également servir à créer de nouveaux scénarios en vue de lancer des simulations;
- **Qualnet Animator** : Permet de lancer et visualiser en temps réel le déroulement des simulations;
- **Qualnet Analyser** : Cet outil permet de visualiser les différentes statistiques et résultats de la simulation. Il s'agit entre autres, de paquets (ou d'octets) envoyés et reçus à différents niveaux, de délai et

du débit bout-en-bout. C'est à partir de ces résultats que nous pourrons analyser la performance des différents protocoles et comparer le comportement de divers mécanismes dans des contextes ou scénarios similaires;

- **Qualnet Tracer :** Permet de tracer les différents paquets qui transitent à travers le réseau;
- **Qualnet Importer :** Permet d'importer des modèles de réseaux vers Qualnet.

En somme, Qualnet est un simulateur à événements discrets. En effet, tout peut y être vu comme résultat se produisant à la suite d'un événement. De plus, comme nous l'avons mentionné précédemment, les différents protocoles fonctionnent tel que des machines à états finis où on passe d'un état à un autre suite à un événement. Par ailleurs, Qualnet est conçu pour être conforme au modèle OSI. Ainsi, chaque couche peut être associée à un ou plusieurs protocoles ou mécanismes implémentés dans Qualnet:

- **Couche d'application :** FTP, HTTP, Telnet, VoIP, ... ;
- **Couche transport :** TCP, UDP ;
- **Couche réseau (IP & routage) :** OSPF, BGP, ICMP ... ;
- **Couche liaison :** 802.11a, 802.11b, 802.11g, CSMA ... ;
- **Couche physique :** communication radio, communication point-à-point.

L'interface entre les différentes couches se fait aussi via des événements. Ces événements sont de deux sortes : *événements de paquets* et *événements du timer*. Les premiers sont utilisés afin de simuler le passage des paquets à travers le réseau. Quant aux seconds, ils sont utilisés comme fonctions d'alarmes et permettent ainsi aux applications de programmer par elles-mêmes le déclenchement de leurs événements dans un instant futur.

4.2 Implémentation : structures de données et algorithmes

Nous détaillerons dans ce qui suit les modifications et ajouts que nous avons effectués dans Qualnet afin de simuler le fonctionnement du protocole HPQS présenté dans le chapitre précédent.

Pour ce faire, nous présenterons l'implémentation de deux versions distinctes de HPQS. La première, nommée HPQS-D correspond à l'adaptation du protocole HPQS pour les contraintes de délai. La seconde HPQS-BP est l'adaptation de HPQS pour les contraintes de bande passante. Les deux versions sont implémentées indépendamment l'une de l'autre tel que décrit dans le chapitre trois.

4.2.1 Implémentation de HPQS-D

Étant donné que le protocole hybride de routage avec qualité de service HPQS-D peut-être considéré comme une extension ou une généralisation du protocole ZRP, nous avons commencé par identifier les structures de données et algorithmes de ce protocole. Ils sont principalement regroupés dans les six fichiers suivants:

- *zrp.h* et *zrp.cpp* : Ces deux fichiers implémentent cadre générique du protocole ZRP. Ils contiennent notamment les algorithmes et structures de données de l'interfaçage entre IARP et IERP;
- *iarp.h* et *iarp.cpp* : Ces fichiers regroupent l'information sur le protocole interne proactif de routage (intrazone);
- *ierp.h* et *ierp.cpp* : Ces fichiers regroupent l'information sur le protocole externe réactif de routage (interzone).

Nos modifications affecteront principalement la composante IARP puisqu'il s'agit d'étendre la zone de routage proactif. Cela dit, pour assurer une certaine cohérence,

quelques ajouts du côté du protocole de routage réactif interzone IERP, se sont également avérés essentiels.

Ainsi, nous commencerons par définir une nouvelle métrique pour IARP. Celle-ci servira à identifier les demandes de connexion exigeant un délai maximum à respecter. Une première métrique concernant le nombre de sauts étant déjà définie dans *IarpMetric*, nous y ajoutons la nouvelle métrique ***IARP_METRIC_DELAI*** tel que le montre la figure 4.1 :

```
typedef enum enum_iarp_metric_type {
    IARP_METRIC_HOP_COUNT = 0,
    IARP_METRIC_DELAI = 1           // Métrique QoS du délai
} IarpMetric;
```

Figure 4.1 Structure de métrique IARP pour HPQS-D

De plus, comme l'illustre la figure 4.2, nous ajoutons une nouvelle entrée ***delaiVersNoeud*** dans la table de routage du protocole IARP représentée par la structure *IarpRoutingEntry*. Celle-ci servira à conserver le délai bout-en-bout vers chaque nœud appartenant à la zone de routage dans la table de routage.

```
typedef struct iarp_routing_entry_struct {
    NodeAddress destinationAddress;
    NodeAddress nextHopAddress;
    NodeAddress subnetMask;
    unsigned int hopCount;
    unsigned int delaiVersNoeud;    // délai bout-en-bout vers le nœud
    int outGoingInterface;
    clocktype lastHard;
    NodeAddress* path;
} IarpRoutingEntry;
```

Figure 4.2 Structure de la table de routage IARP pour HPQS-D

Nous ajoutons aussi une nouvelle entrée dénommée ***zoneRadiusExtra*** laquelle correspond à la taille maximale de la zone de routage, zone externe comprise. Cette variable également appelée rayon de la zone de routage externe, correspond précisément à la variable *DIST₂* décrite dans le chapitre trois. Cette nouvelle entrée sera ajoutée à

trois structures de données : la table des états des liens (*IarpLinkStateTableEntry*) (figure 4.3), l’entête des paquets IARP (*IarpGeneralHeader*) (figure 4.4) ainsi que la structure principale du protocole IARP (*struct_iarp_str*) (figure 4.5).

```
typedef struct link_state_table_entry_struct {
    NodeAddress linkSource;
    unsigned int zoneRadius ;           // Rayon de la zone interne
    unsigned int zoneRadiusExtra ;      // Rayon de la zone externe
    NodeAddress linkStateId;
    clocktype insertTime;
    IarpLinkStateInfo* linkStateinfo;
    struct link_state_table_entry_struct* next;
    struct link_state_table_entry_struct* prev;
} IarpLinkStateTableEntry ;
```

Figure 4.3– Structure de la table des états des liens IARP pour HPQS-D

L’entrée *délaiExige* représente quant à elle la différence entre le délai initialement requis par la connexion et le délai bout-en-bout de la source jusqu’au nœud courant. Il s’agit donc du délai transmis vers les nœuds périphériques tel que le stipule notre algorithme. Cette entrée est ajoutée dans les structures *struct_iarp_str* (figure 4.5) et *IarpGeneralHeader* (figure 4.4). En effet, l’ajout d’un champ supplémentaire *délaiExige* dans l’entête des paquets IARP permet de transmettre cette valeur via le protocole de routage interzone IERP afin d’éviter toute propagation inutile (*délaiExige≤0*).

```
typedef struct iarp_gen_header_struct {
    NodeAddress linkSourceAddress;
    unsigned short linkStateSeqNum;
    unsigned char zoneRadius ;           // Rayon de la zone interne
    unsigned char zoneRadiusExtra ;     // Rayon de la zone externe
    unsigned char ttl ;
    int délaiExige ;                  // délai transmis à la périphérie
    unsigned char reserved[3];
    unsigned char linkDestCnt;
} IarpGeneralHeader;
```

Figure 4.4 Structure de l’en-tête des paquets IARP pour HPQS-D

Finalement, l'entrée *délaiConserve* qui correspond à la variable *DEL* définie dans le chapitre précédent laquelle permet de délimiter notre zone proactive de routage est également ajoutée dans la structure *struct_iarp_str* (figure 4.5).

```
struct struct_iarp_str
{
    int state;
    IarpRoutingTable routeTable ; //IARP routing table structure.
    unsigned zoneRadius ;           // Rayon de la zone interne
    unsigned zoneRadiusExtra ;      // Rayon de la zone externe
    int délaiConserve ;            // DEL
    int délaiExige ;                // délai transmis à la périphérie
    clocktype iarpBroadcastTimer ; //IARP broadcast timer
    clocktype iarpRefreshTimer ;
    clocktype linkStateLifeTime;
    IarpLinkStateTable iarpLinkStateTable;
    unsigned int numOfMyNeighbor;
    NodeAddress myLinkSourceAddress;
    unsigned short linkStateSeqNum;
    SeqTable* seqTable;
    IerpUpdateFunc ierpUpdateFunc;
    IarpStatsType stats;
    unsigned short seed[3];
    int initStats;
    int printStats ;
    BOOL statsPrinted ;
} ;
```

Figure 4.5 Structure générale IARP pour HPQS-D

Notons que toutes les variables de délai sont exprimées en millisecondes (ms).

Au niveau de IERP, tel que le montre la figure 4.6, l'entrée *délaiChemin* est ajoutée à la structure de donnée de la table de routage *IerpRoutingTableEntry*. Cette entrée correspond au délai bout-en-bout de chacun des chemins figurant dans la table de routage de IERP.

```
typedef struct {
    NodeAddress destinationAddress;
    NodeAddress subnetMask;
    clocktype lastUsed;
    unsigned int hopCount;
    unsigned int délaiChemin ; // délai bout-en-bout
    NodeAddress* pathList;
} IerpRoutingTableEntry ;
```

Figure 4.6 Structure de la table de routage IERP pour HPQS-D

De plus, tout comme pour IARP et pour les mêmes raisons, nous ajoutons le champ ***delaiExige*** dans l’entête des paquets du protocole IERP (*IerpGeneralHeader*). Cet ajout est illustré par la figure 4.7 :

```
typedef struct ierp_general_header_struct {
    unsigned char type;
    unsigned char length;
    unsigned int delaiExige ;      // délai transmis à la périphérie
    unsigned char nodePtr;
    unsigned char reserved1;
    unsigned short queryId;
    unsigned short reserved2;
} IerpGeneralHeader ;
```

Figure 4.7 Structure de l’en-tête des paquets IERP pour HPQS-D

De même, tel qu’illustré dans la figure 4.8, l’entrée ***zoneRadiusExtra*** est ajoutée dans la structure principale de IERP (*struct_ierp_str*).

```
struct struct_ierp_str
{
    int state;
    IerpRoutingTable ierpRoutingTable;
    unsigned short queryId;
    unsigned int zoneRadius;
    unsigned int zoneRadiusExtra ;    // Rayon de la zone externe
    IerpSeqTable* seqTable;
    clocktype ierpRouteRequestTimer;
    IerpMessageBuffer msgBuffer;
    int maxBufferSize;
    IerpStatsType stats;
    unsigned short seed[3] ;
    int initStats ;
    int printStats ;
    BOOL statsPrinted ;
};
```

Figure 4.8 Structure générale de IERP pour HPQS-D

En plus de ces ajouts, des modifications algorithmes ont été effectuées au niveau des fonctions du protocole IARP.

Ainsi, nous modifions la fonction *IarpUpdateIerpRoutingTable()*, laquelle se charge de mettre à jour la table de routage du protocole IERP à partir de l'information locale collecté par IARP. Nous l'avons modifiée pour qu'elle mette également à jour l'entrée correspondante au délai que nous avons ajoutée aux tables de routage. Cette fonction est d'autant plus importante puisqu'elle sert de lien entre les deux protocoles IARP et IERP. De plus, la fonction *IarpReceiveNdpUpdate()* qui permet à chaque nœud de recevoir périodiquement l'information de son voisinage sera également modifiée. En effet, cette modification permettra au nœud qui reçoit le paquet de mise à jour de son voisin de calculer le délai bout-en-bout qui le sépare de ce dernier. Ce calcul se fait selon la formule suivante :

$$d = d_{min} + rand(d_{max} - d_{min}) \quad (4.1)$$

où d_{min} et d_{max} représentent successivement le délai minimum et le délai maximum entre deux nœuds voisins. La composante aléatoire permet d'accorder un certain réalisme au calcul.

Aussi, la fonction *IarpCheckForInZoneNode()* vérifie si un nœud donné appartient à la zone de routage proactive. Nous avons modifié cette fonction afin d'inclure la zone de routage externe telle que nous l'avons définie.

Pour la fonction *IarpBuildHeader()* qui construit l'en-tête des paquets IARP, une modification a été aussi nécessaire afin d'y ajouter les champs **zoneRadiusExtra** et **delaiExige** tel qu'expliqué ci-dessus.

Par ailleurs, la fonction *IarpFindShortestPathBFS()* se charge de trouver le plus court chemin d'une source s vers une destination t donnée à travers un chemin $s \rightarrow n_0 \rightarrow n_1 \rightarrow \dots \rightarrow n_{m+l} \rightarrow t$ où le nœud t ainsi que tous les nœuds intermédiaires n_i appartiennent à la zone de routage proactif du nœud s . Nous la modifions pour pouvoir calculer le délai bout-en-bout entre s et t comme suit :

$$d(s, t) = d(s, n_0) + \sum_{i=0}^m d(n_i, n_{i+1}) + d(n_{m+1}, t) \quad (4.2)$$

Finalement, nous modifions la fonction *IarpHandleProtocolPacket()* d'une part afin de tenir en compte l'extension de la zone de routage et d'autre part afin de délimiter la diffusion dans le cas où elle serait inutile (contrainte de délai non respectée).

4.2.2 Implémentation de HPQS-BP

L'implémentation de HPQS adapté aux contraintes de bande passante est sensiblement semblable à celle de HPQS-D.

En effet, du côté de IARP nous définissons une nouvelle métrique pour la bande passante que nous appellerons ***IARP_METRIC_BANDE_PASSANTE***. La figure 4.9 suivante illustre cet ajout :

```
typedef enum enum_iarp_metric_type {
    IARP_METRIC_HOP_COUNT = 0,
    IARP_METRIC_BANDE_PASSANTE = 1 // Métrique QoS de bande passante
} IarpMetric;
```

Figure 4.9 Structure de métrique IARP pour HPQS-BP

Aussi, nous ajoutons une nouvelle entrée ***bandePassanteVersNoeud*** dans la table de routage du protocole IARP représentée par la structure *IarpRoutingEntry* (figure 4.10). Cette entrée servira à conserver la bande passante bout-en-bout vers tous les nœuds qui appartiennent à la zone de routage.

```
typedef struct iarp_routing_entry_struct {
    NodeAddress destinationAddress;
    NodeAddress nextHopAddress;
    NodeAddress subnetMask;
    unsigned int hopCount;
    unsigned int bandePassanteVersNoeud; // bande passante bout-en-bout
    int outGoingInterface;
    clocktype lastHard;
    NodeAddress* path;
} IarpRoutingEntry;
```

Figure 4.10 Structure de la table de routage IARP pour HPQS-BP

Par ailleurs, ici encore nous ajoutons l'entrée ***zoneRadiusExtra*** dans la table des états des liens (*IarpLinkStateTableEntry*), dans l'en-tête des paquets IARP (*IarpGeneralHeader*) ainsi que dans la structure principale du protocole IARP (*struct_iarp_str*) (figures 4.11, 4.12 et 4.13).

```
typedef struct link_state_table_entry_struct {
    NodeAddress linkSource;
    unsigned int zoneRadius;
    unsigned int zoneRadiusExtra; // Rayon de la zone externe
    NodeAddress linkStateId;
    clocktype insertTime;
    IarpLinkStateInfo* linkStateinfo;
    struct link_state_table_entry_struct* next;
    struct link_state_table_entry_struct* prev;
} IarpLinkStateTableEntry;
```

Figure 4.11 Structure de la table des états des liens IARP pour HPQS-BP

La nouvelle entrée ***bandePassanteExigee*** représente la bande passante requise par la connexion. Cette entrée est ajoutée aux deux structures *struct_iarp_str* (figure 4.13) et *IarpGeneralHeader* (figure 4.12). L'ajout d'un champ supplémentaire ***bandePassanteExigee*** dans l'en-tête des paquets IARP permet de transmettre cette valeur via le protocole de routage interzone IERP afin d'éviter toute propagation inutile (*bandePassanteVersNoeud*<*bandePassanteExigee*).

```
typedef struct iarp_gen_header_struct {
    NodeAddress linkSourceAddress;
    unsigned short linkStateSeqNum;
    unsigned char zoneRadius;
    unsigned char zoneRadiusExtra; // Rayon de la zone externe
    unsigned int bandePassanteExigee; // Bande Passante exigée QoS
    unsigned char ttl;
    unsigned char reserved[3];
    unsigned char linkDestCnt;
} IarpGeneralHeader;
```

Figure 4.12 Structure de l'en-tête des paquets IARP pour HPQS-BP

Finalement, les entrées ***bandePassanteConservee*** et ***bandePassanteDisponible*** correspondant successivement à la variable *BP* définie dans le chapitre précédent et à la bande passante encore disponible pour un nœud donné sont également ajoutées à la structure *struct_iarp_str* (figure 4.13).

Notons que toutes les entrées et variables de bande passante sont exprimées en bits par secondes (bps).

```
struct struct_iarp_str
{
    int state;
    IarpRoutingTable routeTable; //IARP routing table structure.
    unsigned zoneRadius;           // Rayon de la zone interne
    unsigned zoneRadiusExtra;      // Rayon de la zone externe
    unsigned int bandePassanteConservee; // BP
    unsigned int bandePassanteDisponible; // bande passante disponible
    unsigned int bandePassanteExigee;    // Bande Passante exigée QoS
    clocktype iarpBroadcastTimer; //IARP broadcast timer
    clocktype iarpRefreshTimer;
    clocktype linkStateLifeTime;
    IarpLinkStateTable iarpLinkStateTable;
    unsigned int numOfMyNeighbor;
    NodeAddress myLinkSourceAddress;
    unsigned short linkStateSeqNum;
    SeqTable* seqTable;
    IerpUpdateFunc ierpUpdateFunc;
    IarpStatsType stats;
    unsigned short seed[3];
    int initStats ;
    int printStats ;
    BOOL statsPrinted ;
} ;
```

Figure 4.13 Structure générale de IARP pour HPQS-BP

Au niveau de IERP, l'entrée ***bandePassanteChemin*** est ajoutée à la structure de donnée de la table de routage *IerpRoutingTableEntry*. Cette entrée correspond à la bande passante bout-en-bout de chacun des chemins figurant dans la table de routage de IERP (figure 4.14).

```

typedef struct {
    NodeAddress destinationAddress;
    NodeAddress subnetMask;
    clocktype lastUsed;
    unsigned int hopCount;
    unsigned int bandePassanteChemin; // bande passante bout-en-bout
    NodeAddress* pathList;
} IerpRoutingTableEntry;

```

Figure 4.14 Structure de la table de routage IERP pour HPQS-BP

Encore là, un champ ***bandePassante*** est ajoutée dans l’entête des paquets du protocole IERP (*IerpGeneralHeader*) (figure 4.15).

```

typedef struct ierp_general_header_struct {
    unsigned char type;
    unsigned char length;
    unsigned int bandePassante; // bande passante exigée par la
                                // connexion
    unsigned char nodePtr;
    unsigned char reserved1;
    unsigned short queryId;
    unsigned short reserved2;
} IerpGeneralHeader;

```

Figure 4.15 Structure de l’en-tête des paquets IERP pour HPQS-BP

De même, l’entrée ***zoneRadiusExtra*** est ajoutée dans la structure principale de IERP (*struct_ierp_str*) (figure 4.16).

Les changements algorithmes sont également similaires à ceux qui ont été effectués pour HPQS-BP.

Ainsi, nous modifions la fonction *IarpUpdateIerpRoutingTable()* pour qu’elle mette également à jour l’entrée correspondante à la bande passante que nous avons ajoutée aux tables de routage.

```

struct struct_ierp_str
{
    int state;
    IerpRoutingTable ierpRoutingTable;
    unsigned short queryId;
    unsigned int zoneRadius;
    unsigned int zoneRadiusExtra; // Rayon zone externe
    IerpSeqTable* seqTable;
    clocktype ierpRouteRequestTimer;
    IerpMessageBuffer msgBuffer;
    int maxBufferSize;
    IerpStatsType stats;
    unsigned short seed[3];
    int initStats;
    int printStats;
    BOOL statsPrinted;
};

```

Figure 4.16 Structure générale IERP pour HPQS-BP

Nous modifions également la fonction *IarpReceiveNdpUpdate()* de manière à estimer la bande passante des liens vers les nœuds du voisinage immédiat. Pour ce faire, nous utilisons la formule :

$$bp = bp_{min} + rand(bp_{min} - bp_{max}) \quad (4.3)$$

où bp_{min} et bp_{max} représentent successivement la bande passante minimum et la bande passante maximum entre deux nœuds voisins. Comme ce fut le cas pour le délai, la composante aléatoire permet d'accorder un certain réalisme au calcul.

Ici encore, la fonction *IarpCheckForInZoneNode()* a été modifiée pour inclure la zone de routage externe telle que nous l'avons définie pour HPQS-BP.

Pour la fonction *IarpBuildHeader()*, une modification a été aussi nécessaire afin d'y ajouter les champs **zoneRadiusExtra** et **bandePassanteExigee**.

Par ailleurs, nous modifions la fonction *IarpFindShortestPathBFS()* pour pouvoir calculer la bande passante bout-en-bout entre une source s et une destination t à travers un chemin $s \rightarrow n_0 \rightarrow n_1 \rightarrow \dots \rightarrow n_{m+1} \rightarrow t$ où le nœud t ainsi que tous les nœuds

intermédiaires n_i appartiennent à la zone de routage proactif du nœud s . Pour ce faire, nous utilisons la formule suivante :

$$bp(s,t) = \text{MIN}(bp(s,n_0), \underset{i=0}{\overset{m}{\text{MIN}}}(n_i, n_{i+1}), bp(n_{m+1}, t)) \quad (4.4)$$

Finalement, nous modifions la fonction *IarpHandleProtocolPacket()* d'une part afin de tenir compte de l'extension de la zone de routage et d'autre part afin de délimiter la diffusion dans le cas où elle serait inutile (contrainte de bande passante non respectée ou bande passante non suffisante pour répondre adéquatement à la demande de connexion).

4.3 Plan d'expérience et scénarios

Dans cette section, nous présenterons les différents paramètres de simulation utilisés qui nous permettront de comparer les meilleurs protocoles présents dans la littérature avec le nôtre.

En effet, nous proposons de comparer les performances des deux versions HPQS-D et HPQS-BP avec celles des deux protocoles réactifs AODV et DSR, celles du protocole proactif Bellman-Ford (BF) et enfin celles du protocole hybride ZRP.

Pour ce faire, il nous faut établir plusieurs scénarios de manière à analyser le fonctionnement des protocoles dans des contextes similaires. Dans chacun de ces scénarios, nous devrons introduire du trafic entre différentes paires (source, destination). Afin d'y parvenir nous utilisons des connexions de type CBR relativement exigeantes en terme de QoS, une connexion FTP usuelle pour échange de fichiers ainsi qu'une connexion usuelle de type Telnet. Notons que les paires (source, destination) seront distinctes pour chacune des connexions.

Ainsi, nous avons conçu quatre scénarios :

- Scénario 1 : 30 nœuds mobiles avec une connexion CBR;
- Scénario 2 : 50 nœuds mobiles avec une connexion CBR, une connexion Telnet et une connexion FTP;

- Scénario 3 : 80 nœuds mobiles avec une connexion CBR, une connexion Telnet et une connexion FTP;
- Scénario 4 : 120 nœuds mobiles avec une connexion CBR, une connexion Telnet et une connexion FTP.

Dans tous les cas, les nœuds se déplaceront à l'intérieur d'une surface de 1250 m x 1250 m. De plus, pour pouvoir étudier l'impact du mouvement des nœuds sur nos critères de performance, les nœuds se déplaceront :

- À 0 m/s (aucune mobilité ou mobilité faible) ;
- Aléatoirement selon la loi uniforme dans l'intervalle [0,10] m/s (vitesse d'une voiture en ville) ;
- Aléatoirement selon la loi uniforme dans l'intervalle [0,30] m/s (vitesse d'une voiture dans une autoroute urbaine) ;
- Aléatoirement selon la loi uniforme dans l'intervalle [0,70] m/s (train à grande vitesse).

Dans les cas de mobilité, nous utiliserons le patron *RANDOM_WAYPOINT*. En effet, bien que ce patron ne soit pas tout à fait représentatif de la réalité, il demeure le mieux adapté pour l'analyse de performance puisqu'il est le plus largement utilisé dans la littérature. Mentionnons qu'après chaque déplacement, les nœuds s'arrêtent pendant ce qu'on appelle un taux de pause lequel sera fixé à 30 secondes pour toutes nos simulations.

Par ailleurs, nous ferons varier le rayon de la zone de routage de protocole ZRP. À cet effet, nous utiliserons les valeurs 2 et 3 sauts. De même, pour HPQS-D et HPQS-BP les valeurs des zones internes et externes varieront successivement à (1,2) et (2,3).

Aussi, pour le protocole HPQS-D nous ferons changer la valeur de *DEL* dite du délai conservé, les valeurs de 500 ms et d'une seconde seront utilisées.

De même, pour HPQS-BP, nous ferons varier la valeur BP (bande passante conservée) pour laquelle nous utiliserons les valeurs de 3 et 10 Mbps.

Notons que le temps de simulation sera de 30 secondes pour chaque expérience ce qui correspond à temps réel de 30 minutes en moyenne selon la complexité du scénario. De plus, dans le but de nous assurer de la validité des résultats obtenus, nous répéterons plusieurs fois la même expérience.

Le tableau suivant résume notre plan d'expérience :

Tableau 4.2 Plan d'expérience

Paramètre	Protocole(s) concerné(s)	Valeur(s)
Terrain	Tous	1250 x 1250 m ²
Nombre de nœuds	Tous	30, 50, 80 et 120
Patron de mobilité	Tous	Random_waypoint
Vitesse	Tous	0, [0,10], [0,30] et [0,70] m/s
Rayon de zone de routage	ZRP	2 et 3
Rayon interne, Rayon externe	HPQS-D, HPQS-BP	(1,2) et (2,3)
Délai conservé (DEL)	HPQS-D	500 et 1000 ms
Bandé passante conservée (BP)	HPQS-BP	3 et 10 Mbps

4.4 Résultats et analyse

Maintenant que nous avons établit notre plan d'expérience, il nous reste à choisir les différentes métriques qui nous permettrons de comparer les performances des différents protocoles.

Ainsi, notre comparaison se fera sur la base des métriques de performance suivantes :

- ✓ Le délai bout-en-bout : Cette métrique indique le temps en secondes nécessaire pour qu'un paquet donné arrive à destination à partir de la source;

- ✓ La variation du délai : Cette métrique n'est nulle autre que la variation de la précédente. Elle permet de quantifier la stabilité du délai pendant la durée des connexions;
- ✓ Le débit : Cette métrique correspond aux nombres de bits par secondes que reçoit la destination;
- ✓ La surcharge de signalisation : Cette métrique permet de mesurer le nombre de paquets de signalisation nécessaire pour assurer le bon fonctionnement de chaque protocole.

Après avoir réalisé la série des simulations de notre plan d'expérience, nous présenterons et analyserons dans ce qui suit les résultats obtenus. Notons toutefois qu'en raison de la grande quantité d'expériences et des résultats générés, nous ne pourrons présenter la totalité de ceux-ci. Cependant, les résultats présentés seront suffisamment complets pour permettre une analyse de performance valide pour notre solution. En effet, les autres résultats n'apporteront aucun changement à notre étude.

4.4.1 Délai moyen bout-en-bout

Dans ce qui suit, nous étudierons le délai bout-en-bout des différents protocoles pour les quatre scénarios (30, 50, 80 et 120 nœuds).

Ainsi, Les figures 4.17 (a) et (b) représentent le délai moyen bout-en-bout des connexions CBR en fonction de la vitesse des nœuds pour le premier scénario.

Notons que pour ce cas le rayon de la zone de routage pour le protocole ZRP est de 2 sauts. De plus, pour les protocoles HPQS-D et HPQS-BP les valeurs des zones de routage interne et externe sont successivement fixées à 2 et à 3 sauts. Aussi, les valeurs de *DEL* pour HPQS-D et *BP* pour HPQS-BP seront successivement fixées à 500 ms et 10 Mbps.

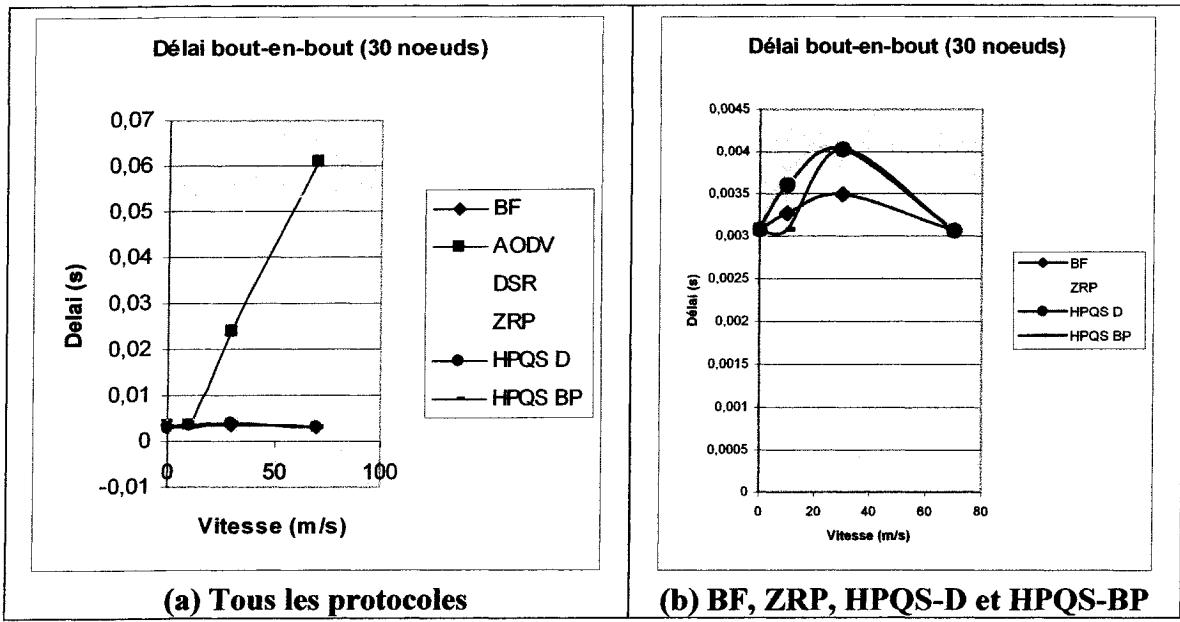


Figure 4.17 Délai bout-en-bout (30 noeuds)

La figure 4.17 (a) illustre le délai de l'ensemble des protocoles. Ainsi, nous constatons que jusqu'à $v=10 \text{ m/s}$ le délai bout-en-bout des connexions est sensiblement le même pour tous les protocoles et se situe à peu près à 3 ms .

Cependant, à partir de $v=30 \text{ m/s}$ un avantage est constaté au profit des protocoles proactif (BF) et hybrides (ZRP, HPQS-D et HPQS-BP) par rapport aux protocoles réactifs AODV et DSR. En effet, alors que le délai bout-en-bout croît très faiblement pour les protocoles proactif et hybrides, il atteint 47 ms pour DSR et 24 ms pour AODV. Cela peut s'expliquer par le fait que les protocoles proactifs (et hybrides) maintiennent en tout temps une table de routage complète (ou partielle) ce qu'ils leur permet de communiquer d'une manière plus rapide.

Ainsi, le gain des protocoles HPQS-D et HPQS-BP par rapport aux protocoles réactifs DSR et AODV dépasse 80% pour les vitesses élevées.

La figure 4.17 (b) nous permet par ailleurs de comparer de plus près le comportement des protocoles HPQS-D et HPQS-BP par rapport aux protocoles BF et ZRP. Nous remarquons que le délai varie faiblement pour ces quatre protocoles (environ 1 ms). La

différence entre ces protocoles peut donc être considérée à toute fin pratique comme non significative.

Les figures 4.18 (a) et (b) illustre le délai bout-en-bout en fonction de la vitesse pour un réseau de 50 unités mobiles :

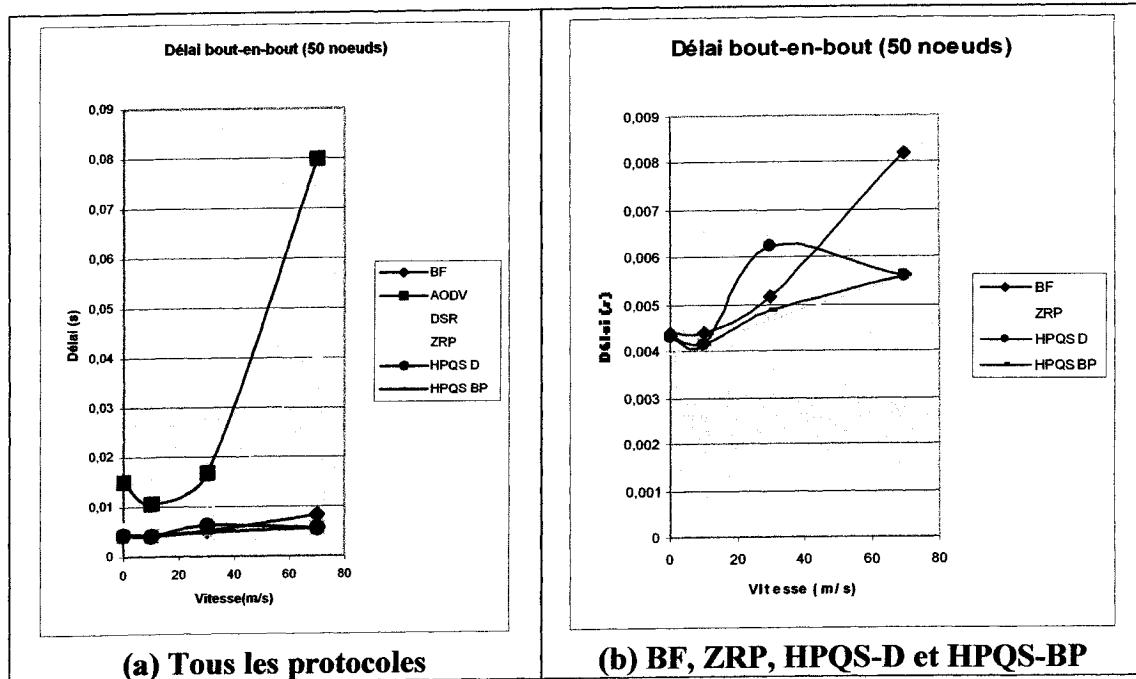


Figure 4.18 Délai bout-en-bout (50 nœuds)

Notons que les paramètres utilisés pour les protocoles hybrides (rayons des zones de routage, *DEL* et *BP*) sont les mêmes que ceux du cas précédent.

Comme pour le réseau de 30 nœuds, il y a une différence importante entre d'un côté, les délais des protocoles réactifs et d'un autre, ceux des protocoles hybrides et proactif. Ainsi, les gains réalisés par HPQS-D et HPQS-BP par rapport aux protocoles réactifs AODV et DSR varient en moyenne entre 27% et 74%.

La figure 4.18 (b) montre par ailleurs que les délais bout-en-bout des protocoles BF, ZRP, HPQS-D et HPQS-BP varient à l'intérieur d'un intervalle de 4 ms. Ce qui n'est encore une fois pas assez significatif pour en tirer des conclusions.

Les figures 4.19 (a) et (b) représentent le délai bout-en-bout en fonction de la vitesse pour un réseau de 80 unités mobiles :

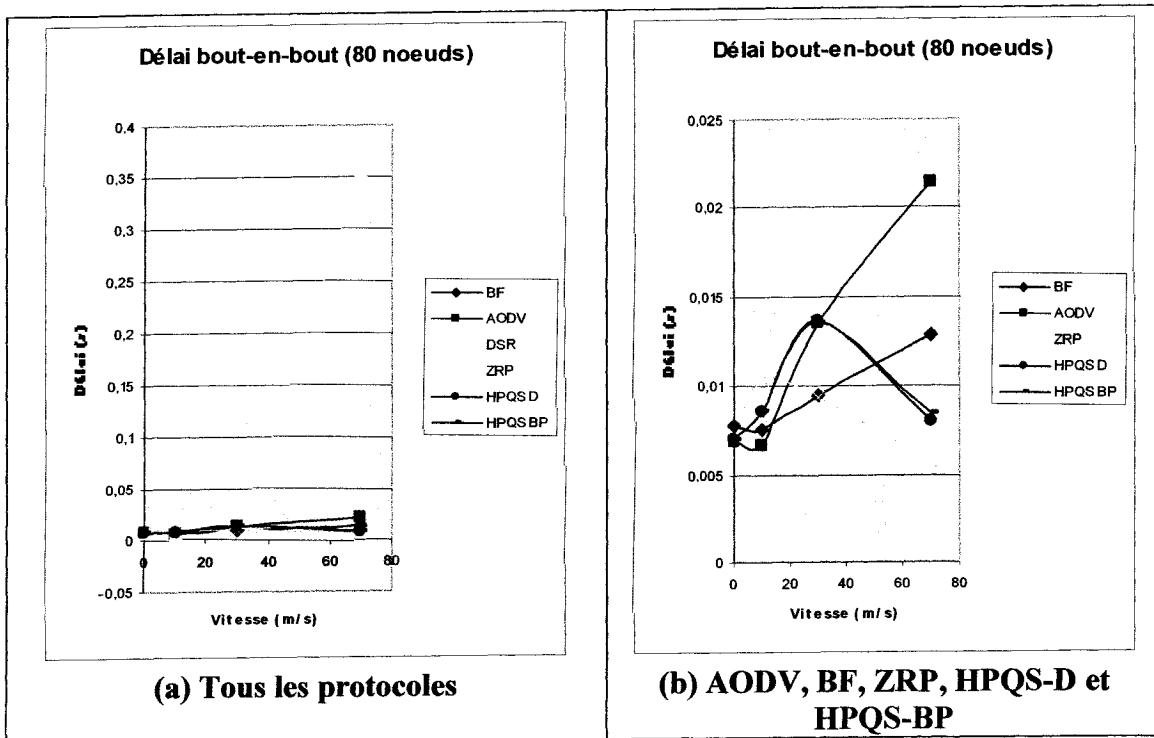


Figure 4.19 Délai bout-en-bout (80 noeuds)

Pour ce cas, la taille de la zone de routage du protocole ZRP reste fixée à 2 sauts. Par ailleurs, les tailles des zones de routages interne et externe sont fixées successivement à 1 et 2 sauts pour les protocoles HPQS-D et HPQS-BP. Quant aux valeurs de *DEL* pour HPQS-D et *BP* pour HPQS-BP elles sont fixées successivement à 1 s et 3 Mbps.

Cette fois, nous constatons que seul le protocole DSR se distingue par un délai bout-en-bout élevé pour les vitesses supérieures à $v=30 \text{ m/s}$. En effet, le délai du protocole AODV demeure dans l'ordre de grandeur des protocoles hybrides et proactif.

Les protocoles HPQS-D et HPQS-BP suivent sensiblement la même courbe. On notera que dans ce cas, les protocoles HPQS-D et HPQS-BP semblent avoir un comportement plus proche de celui des protocoles réactifs. En effet, si le gain moyen en terme de délai des protocoles HPQS-D et HPQS-BP est de 26% par rapport à DSR, il n'est que de 7% par rapport à AODV. En fait, pour une faible mobilité AODV est meilleur et ce n'est qu'à partir de $v=30 \text{ m/s}$ que les protocoles HPQS-D et HPQS-BP l'emportent. Ceci

s'explique par la taille réduite des zones interne et externe. En effet, plus on réduit la taille des zones de routage, plus HPQS-D et HPQS-BP auront tendance à utiliser leur protocole de routage réactif interzone IERP au dépend du protocole de routage proactif intrazone IARP.

Pour ce qui est du quatrième scénario (120 nœuds), nous utiliserons la même configuration pour les protocoles hybrides que celle utilisée pour le cas de 30 et de 50 nœuds.

Les figures 4.20 (a) et (b) illustrent les résultats obtenus :

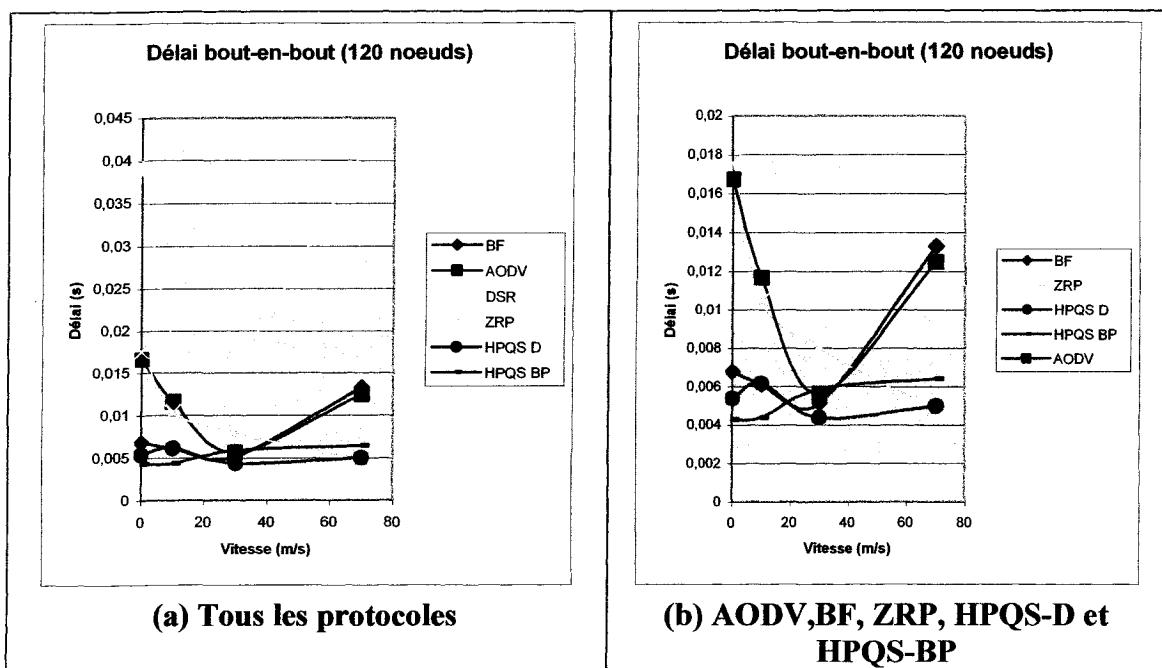


Figure 4.20 Délai bout-en-bout (120 nœuds)

Ici encore, le protocole DSR se distingue par son délai élevé par rapport aux protocoles. De plus, la figure 4.20 (b) nous montrent que nos protocoles réalisent les meilleures performances. En effet, HPQS-D améliore en moyenne le délai bout-en-bout de 24%, 49%, 78% et 38% successivement pour les protocoles BF, AODV, DSR et ZRP. Quant à HPQS-BP, l'amélioration moyenne est de 25%, 45%, 78% et 31%.

En résumé, nous pouvons affirmer que le délai bout-en-bout des protocoles HPQS-D et HPQS-BP dépend de la taille des zones internes et externes ainsi que de la densité du réseau (nombre de nœuds). Ainsi, nous constatons que plus la taille des zones sera grande, plus HPQS-D et HPQS-BP auront tendance à se comporter comme des protocoles proactifs (délai bout-en-bout plus faible). Inversement, plus la taille des zones sera restreinte, plus grand sera le délai bout-en-bout des protocoles HPQS-D et HPQS-BP.

De plus, en comparant le délai pour les différents scénarios, nous constatons que les performances de HPQS-D et HPQS-BP sont meilleures pour les réseaux denses pour lesquels nos protocoles surpassent à la fois les protocoles réactifs et proactifs.

4.4.2 Variation du délai

Dans ce qui suit, nous analyserons la variation du délai pour les protocoles AODV, DSR, BF, ZRP, HPQS-D et HPQS-BP. Ainsi, nous présenterons les résultats pour les réseaux de 30, 50, 80 et 120 unités mobiles.

Les figures 4.21 (a) et (b) illustrent le cas de 30 nœuds.

Notons que pour ce cas le rayon de la zone de routage pour le protocole ZRP est de 2 sauts. Les valeurs des zones de routage interne et externe pour les protocoles HPQS-D et HPQS-BP sont successivement fixées à 2 et à 3 sauts. De plus, les valeurs de *DEL* pour HPQS-D et *BP* pour HPQS-BP seront fixées à 500 ms et 10 Mbps.

La figure 4.21 (a) nous montre qu'à partir de $v=10 \text{ m/s}$, les variations des délais des protocoles réactifs AODV et DSR deviennent beaucoup plus grandes que celles des protocoles proactif et hybrides. Ainsi, s'il est difficile de comparer la performance des protocoles entre 0 et 10 m/s étant donné que leurs variations de délai dans cet intervalle sont inférieures à 2 ms, on constate en revanche que pour les grandes vitesses, il y a un écart important entre les protocoles réactifs d'un côté et les protocoles proactif et hybrides de l'autre.

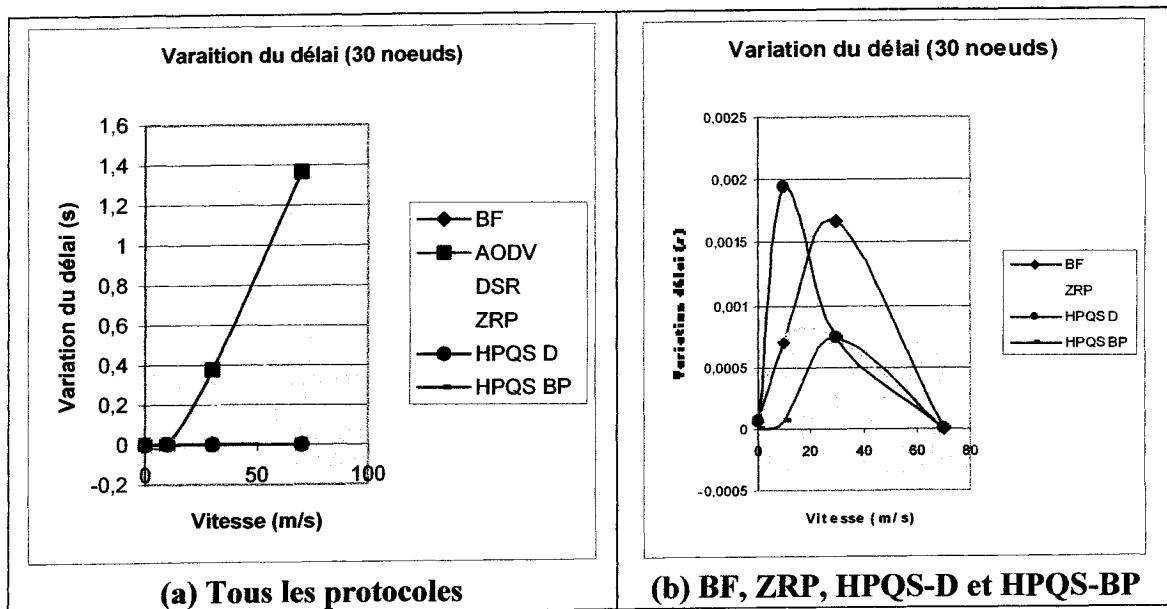


Figure 4.21 Variation du délai (30 noeuds)

Concernant la figure 4.21 (b), la faible différence (moins de 2 ms) qui existe entre les protocoles HPQS-D, HPQS-BP, ZRP et BF sur la totalité de l'intervalle étudié nous ne permet pas de tirer de conclusions sinon que de constater que la différence de variation de délai entre ces protocoles n'est pas significative.

Les figures 4.22 (a) et (b) illustrent la variation du délai en fonction de la vitesse pour le second scénario.

Les paramètres de ZRP, HPQS-D et HPQS-BP sont les mêmes que pour le cas précédent. Encore là, les protocoles réactifs se distinguent par leur variation de délai élevée pour les vitesses supérieures à $v=10$ m/s.

Par ailleurs, la figure 4.22 (b) révèle que la variation du délai du protocole HPQS-D est en général supérieure à celle des autres protocoles hybrides et du protocole proactif. En effet, la différence atteint son maximum (environ 14 ms) en absence de mobilité. La variation du délai de HPQS-D atteint alors environ 16 ms. Cependant, malgré cette différence on peut considérer que la variation du délai de HPQS-D demeure acceptable puisqu'elle demeure au pire des cas comparable et souvent meilleure que celle des protocoles réactifs.

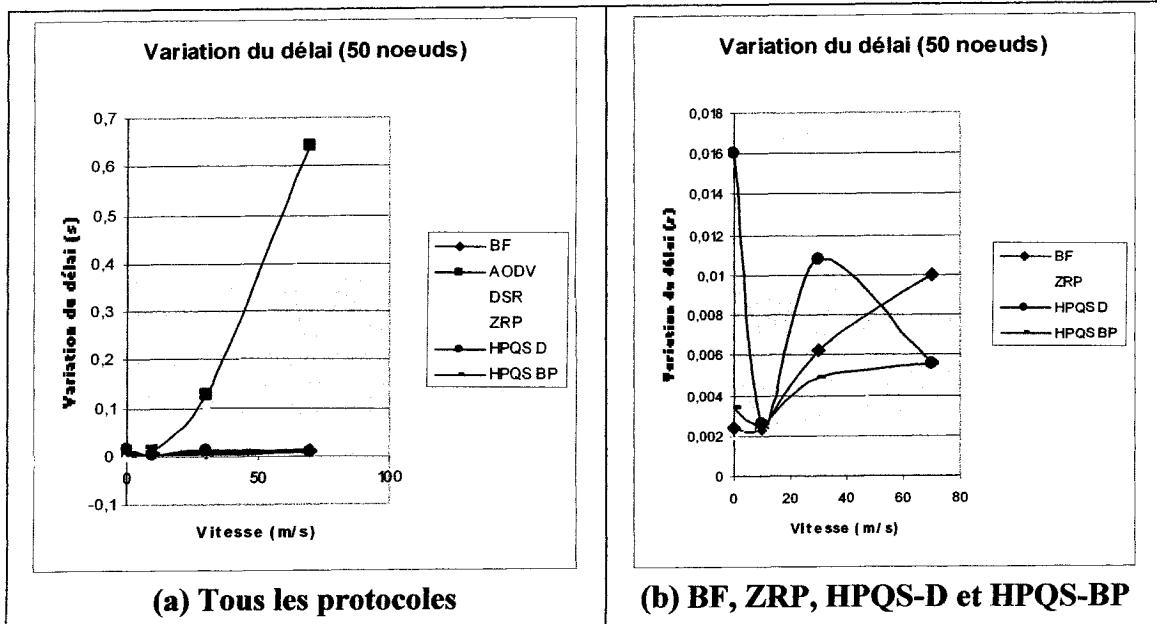


Figure 4.22 Variation du délai (50 noeuds)

Quant au protocole HPQS-BP, nous pouvons affirmer que sa variation de délai est comparable à celle des protocoles ZRP et BF.

Pour le troisième scénario (80 nœuds), nous présenterons les résultats obtenus pour les paramètres suivants : taille de la zone de routage du protocole ZRP de 2 sauts, tailles des zones de routages interne et externe successivement de 1 et 2 sauts pour les protocoles HPQS-D et HPQS-BP. Quant aux valeurs de *DEL* pour HPQS-D et *BP* pour HPQS-BP elles sont fixées successivement à 1 s et 3 Mbps.

Les figures 4.23 (a) et (b) illustrent les résultats obtenus.

Comme dans les cas précédents, la variation de délai est plus importante pour les protocoles AODV et DSR. De plus, la figure 4.23 (b) montre que les deux protocoles HPQS-D et HPQS-BP suivent sensiblement la même courbe. Aussi, leurs variations de délai sont en général en deçà de celles des autres protocoles aussi bien réactifs (AODV, DSR) que hybride (ZRP). En effet, si la variation de délai des protocoles HPQS-D et HPQS-BP varie entre 6 et 11 ms, celles de AODV, DSR et ZRP peuvent atteindre successivement les valeurs de 219, 969 et 19 ms.

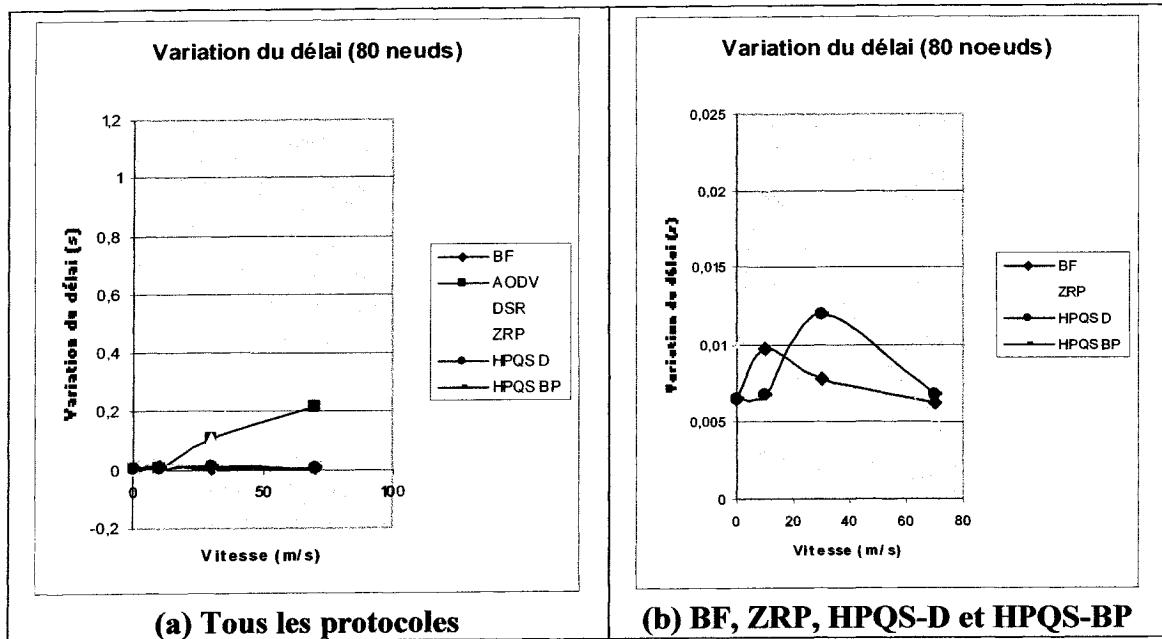


Figure 4.23 Variation du délai (80 noeuds)

Pour le cas du quatrième scénario (120 unités mobiles), le rayon de la zone de routage pour le protocole ZRP est de 2 sauts. De plus, pour les protocoles HPQS-D et HPQS-BP les valeurs des zones de routage interne et externe sont successivement fixées à 2 et à 3 sauts. Aussi, les valeurs de *DEL* pour HPQS-D et *BP* pour HPQS-BP seront fixées à 500 ms et 10 Mbps.

Les figures 4.24 (a) et (b) qui suivent illustrent les résultats obtenus. Nous constatons que nos deux protocoles ont en moyenne la variation de délai la moins importante en comparaison avec les autres protocoles étudiés. En effet, la variation du délai pour HPQS-D et HPQS-BP se situe entre 1 et 3 ms alors qu'elle dépasse à grande vitesse 250 ms pour les protocoles réactifs AODV et DSR et avoisine par moments les 30 ms pour ZRP et BF.

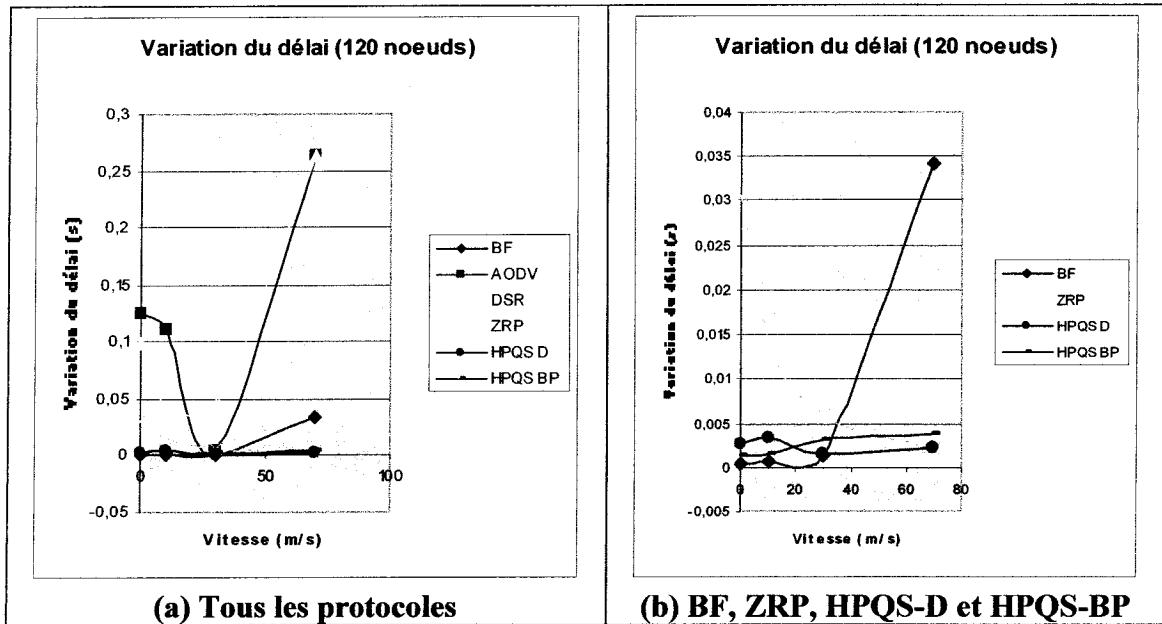


Figure 4.24 Variation du délai (120 noeuds)

En résumé, nous constatons que la variation de délai des protocoles HPQS-D et HPQS-BP est dans le pire des cas comparable à celles des protocoles réactifs AODV et DSR. Par ailleurs, les résultats semblent indiquer qu'en terme de variation de délai, les protocoles HPQS-D et HPQS-BP semblent mieux adaptés aux réseaux plus denses. En effet, si les résultats sont acceptables pour les réseaux de faible densité, les résultats obtenus pour les réseaux de grande taille sont plus intéressants.

4.4.3 Débit

Nous traiterons dans cette partie les résultats concernant le débit des connexions CBR pour les quatre scénarios étudiés.

Pour chacun de ces scénarios, le rayon de la zone de routage pour le protocole ZRP est fixé à 2 sauts. De plus, pour les protocoles HPQS-D et HPQS-BP les valeurs des zones de routage interne et externe sont successivement fixées à 2 et à 3 sauts. Par ailleurs, les valeurs de *DEL* pour HPQS-D et *BP* pour HPQS-BP seront successivement fixées à 500 ms et 10 Mbps.

La figure 4.25 illustre le cas du réseau de 30 nœuds. Elle montre que jusqu'à $v = 10 \text{ m/s}$, le débit des différents protocoles est quasiment identique. En effet, tous les protocoles assurent un débit de 4275 bits/s en absence de mobilité et la même valeur jusqu'à $v = 10 \text{ m/s}$. Au-delà de cette vitesse, même si le débit de tous les protocoles tend à la baisse, nous constatons que les comportements des protocoles réactifs et les protocoles proactifs (ou hybrides) se distinguent les uns des autres.

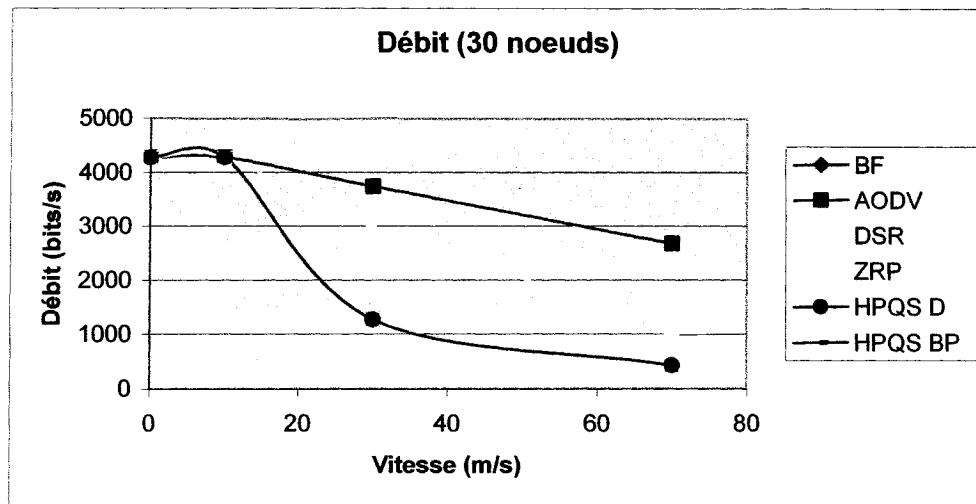


Figure 4.25 Débit (30 nœuds)

En effet, pour les grandes vitesses, les protocoles BF, ZRP, HPQS-D et HPQS-BP évoluant d'une manière très similaire ont un débit moindre que les protocoles AODV et DSR. Cela peut s'expliquer par la difficulté de maintenir la table de routage quand les nœuds se déplacent à grande vitesse. Ainsi, si à $v = 30 \text{ m/s}$, les débits d'AODV et de DSR sont successivement de 3739 et 2967 bits/s, ils n'atteignent que 1271 bits/s pour les autres protocoles (écart de 2468 bits/s et 1696 bits/s).

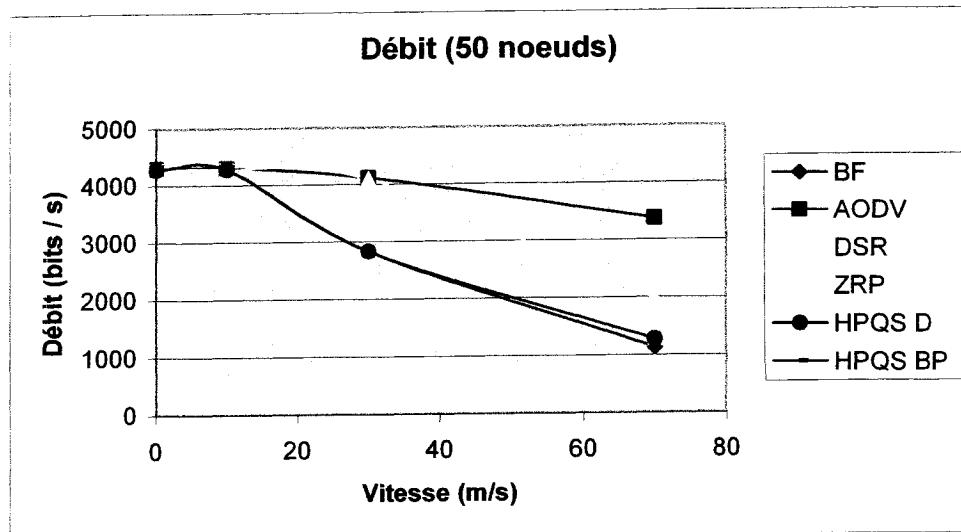


Figure 4.26 Débit (50 noeuds)

Le même constat peut être fait pour le réseau de 50 nœuds, illustré dans la figure 4.26 ci-dessus. En effet, à $v = 30 \text{ m/s}$, les débits d'AODV et de DSR sont successivement de 4119 et 4097 bits/s, ils n'atteignent que 2825 bits/s pour les autres protocoles (écart de 1294 bits/s et 1272 bits/s).

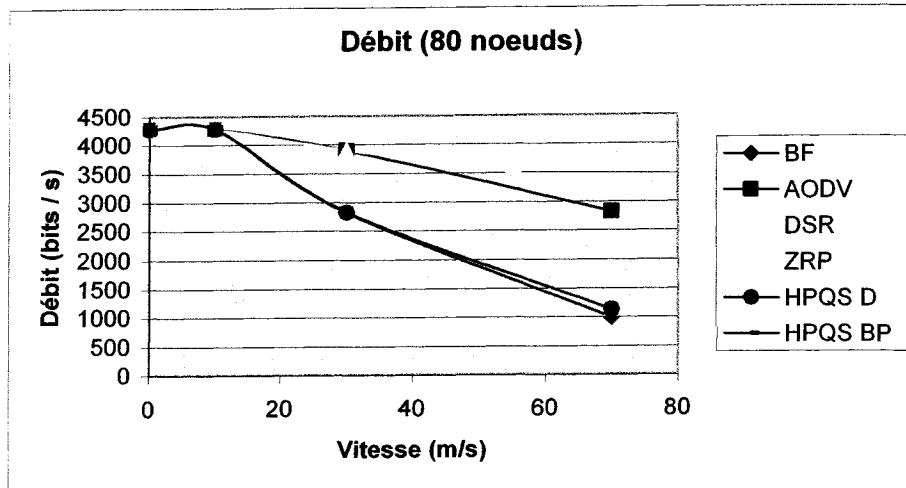


Figure 4.27 Débit (80 noeuds)

Notons toutefois que l'écart entre les protocoles réactifs et hybrides diminue comparativement au scénario précédent. Cet écart diminue encore plus dans le cas du réseau à 80 nœuds dont les résultats sont illustrés dans la figure 4.27.

En effet, à $v = 30 \text{ m/s}$, les débits d'AODV et de DSR sont successivement de 3897 et 3904 bits/s, ils n'atteignent que 2825 bits/s pour les autres protocoles (écart de 1072 bits/s et 1079 bits/s).

Finalement, la figure 4.28 illustre le débit en fonction de la vitesse pour le réseau de 120 nœuds :

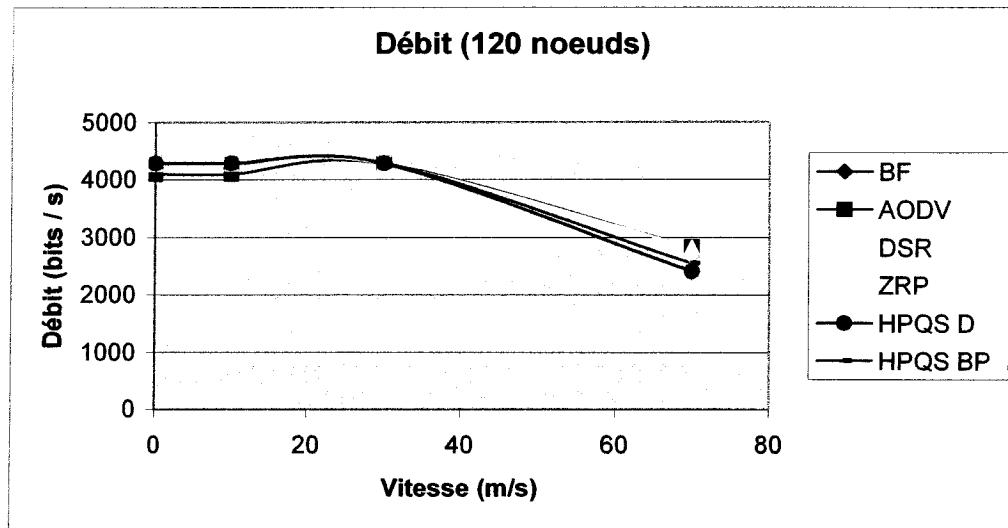


Figure 4.28 Débit (120 nœuds)

Nous constatons que les protocoles proactif et hybrides assurent un meilleur débit dans ce cas. En fait, jusqu'à $v = 30 \text{ m/s}$ les protocoles hybrides ont un débit légèrement supérieur à celui des protocoles AODV et DSR. En effet, HPQS-D réalise un gain de 3% en moyenne par rapport à AODV. Les gains réalisés par HPQS-BP sont inférieurs à 1 %. En résumé, nous constatons que la performance en matière de débit pour les protocoles HPQS-D et HPQS-BP tout comme celle de ZRP est reliée à la densité du réseau. En effet, plus le réseau est dense meilleure sera la performance de ces deux protocoles. Cependant, pour les réseaux moins denses et pour des vitesses élevées le débit de ces protocoles décroît rapidement mais reste toujours supérieur ou égal à celui du protocole proactif (BF).

4.4.4 Surcharge de signalisation

Dans ce qui suit nous comparerons la charge de signalisation des protocoles HPQS-D et HPQS-BP avec celle des autres protocoles. À cette fin, la taille de la zone de routage du protocole ZRP sera fixée à 2 sauts. Par ailleurs, les tailles des zones de routages interne et externe seront fixées successivement à 1 et 2 sauts pour les protocoles HPQS-D et HPQS-BP. Quant aux valeurs de *DEL* pour HPQS-D et *BP* pour HPQS-BP elles seront fixées successivement à 1 s et 3 Mbps.

La figure 4.29 représente la charge de signalisation pour les différents scénarios en absence de mobilité :

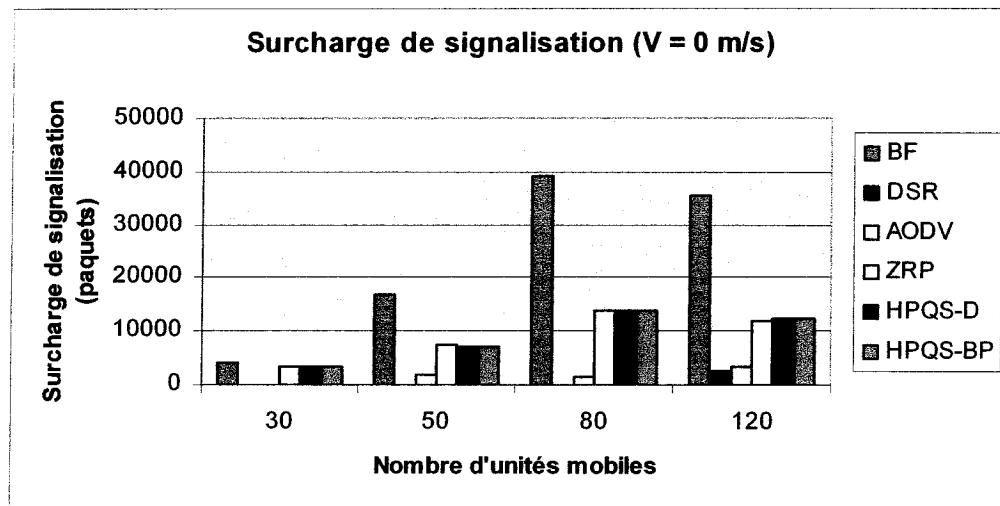


Figure 4.29 Surcharge de signalisation (absence de mobilité)

Nous constatons que la surcharge des protocoles HPQS-D et HPQS-BP est en général sensiblement égale à celle du protocole ZRP qui elle-même est toujours inférieure à celle du protocole proactif (BF). Cela veut dire que l'ajout d'un champ supplémentaire QdS dans les paquets de recherche de route n'a pas eu un effet néfaste sur le réseau.

En fait, pour les cas de faible mobilité (figure 4.29) ou de mobilité moyenne (figure 4.30), nous notons que la surcharge de HPQS-D et HPQS-BP est légèrement supérieure à celle de ZRP. Cependant, cette surcharge supplémentaire ne dépasse pas 0,8 % en moyenne.

Pour les grandes vitesses, nous faisons la constatation inverse. En effet, nous remarquons dans ces cas (figures 4.31 et 4.32) un avantage en faveur de nos deux protocoles aux détriments de ZRP. Cet avantage est dû au fait que nous comparons le protocole ZRP avec une zone de routage de taille 2 avec les protocoles HPQS-D et HPQS-BP dont les tailles des zones de routages internes et externes sont successivement de 1 et 2 sauts. Par conséquent, les zones de routage de HPQS-D (et HPQS-BP) n'incluent pas tous les nœuds dont le nombre de sauts est égal à 2, encore faut-il que la contrainte de délai *DEL* (ou de bande passante *BP*) soit respectée.

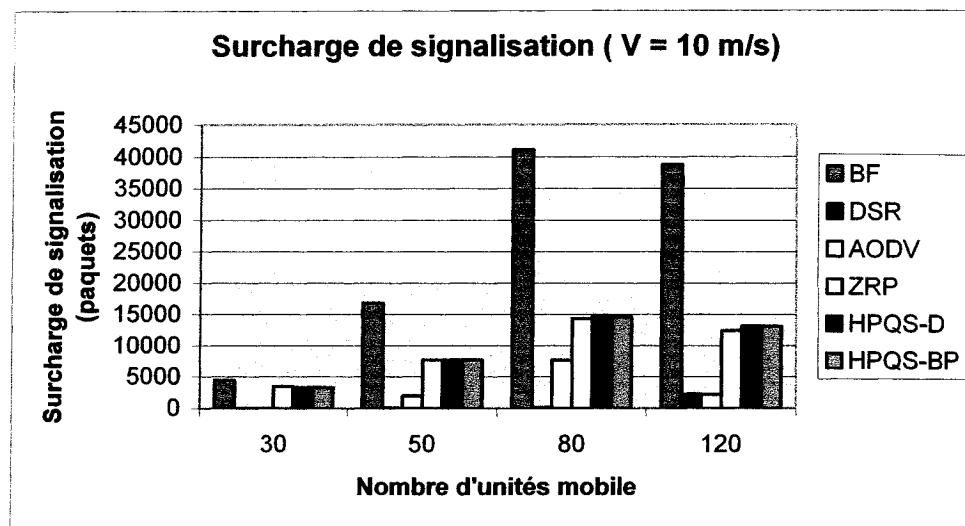


Figure 4.30 Surcharge de signalisation (V = 10 m/s)

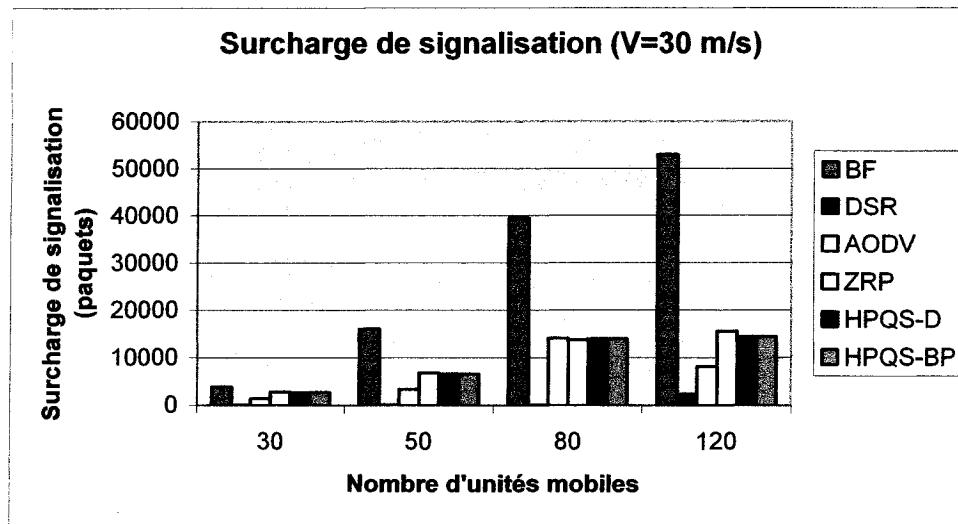


Figure 4.31 Surcharge de signalisation (V=30 m/s)

Ainsi, dans ces deux derniers cas, le gain des protocoles HPQS-D et HPQS-BP par rapport à ZRP avoisine les 3%.

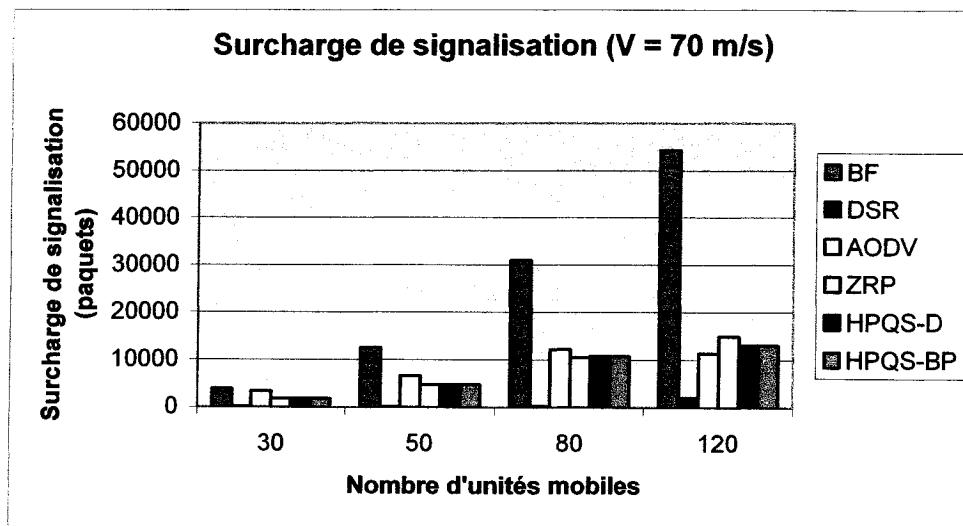


Figure 4.32 Surcharge de signalisation (V=70 m/s)

En résumé, nous pouvons affirmer que la surcharge de HPQS-D et HPQS-BP est en général comparable à celle de ZRP. Ainsi, si pour les cas de faible mobilité nos protocoles semblent augmenter très légèrement la surcharge de signalisation, il en est

autrement pour les cas de mobilité à grande vitesse où nos deux protocoles ont des meilleures performances que ZRP en terme de surcharge de signalisation.

4.4.4 Synthèse des résultats

Suite aux résultats présentés, nous constatons que les protocoles HPQS-D et HPQS-BP ont dans le pire des cas des performances comparables à ceux du protocole ZRP et peuvent souvent les améliorer.

En effet, nous avons pu constater qu'en matière de délai et de variation de délai les performances de nos protocoles sont meilleurs que ceux des protocoles réactifs AODV et DSR. Pour ce qui est du débit, les performances sont quasiment les mêmes que celle ZRP. De plus, nous avons constaté que pour ces trois métriques les performances de notre solution sont meilleures pour les réseaux relativement denses.

Concernant la surcharge de signalisation nous avons vu que nos deux protocoles diminuent la surcharge comparativement à ZRP quand les nœuds se déplacent à grande vitesse.

Pour l'équité, il faut garder à l'esprit que parmi tous les protocoles étudiés dans ce chapitre, HPQS-D et HPQS-BP sont les seuls qui exigent que leurs routes satisfassent à des critères de qualité de service. En effet, les autres protocoles ne procurent aucune garantie de délai ou de bande passante.

Par ailleurs, rappelons que pour pouvoir maximiser les performances de HPQS-D et HPQS-BP, il faudra les configurer de façon optimale. En effet, le choix de la taille des zones de routage interne et externe ainsi que celui des paramètres *DEL* et *BP* rend notre solution flexible et peut influencer les performances de nos protocoles.

CHAPITRE V

CONCLUSION

Après avoir introduit le problème de la gestion de la qualité de service dans les réseaux ad hoc mobiles, nous avons présenté un résumé des recherches antérieures qui traitent du sujet. Ensuite, nous avons proposé un nouveau protocole qui marie le concept d'hybridité proactif / réactif avec celui de la qualité de services. Les résultats que nous avons obtenus suite à la simulation sont encourageants.

Dans ce dernier chapitre, nous commencerons par synthétiser le travail réalisé dans le cadre de ce mémoire tout en mettant en valeur son caractère original. Puis, nous discuterons des limites de notre solution. Cependant, étant donné que le problème de la qualité de service dans les réseaux ad hoc reste un problème ouvert, nous terminerons ce mémoire en proposant de nouvelles pistes pour des recherches futures.

5.1 Synthèse des travaux et originalité des contributions

En ce début de siècle, les réseaux téléinformatiques sont appelés à jouer des rôles de plus en plus importants et ce dans des contextes très variés (éducation, recherches scientifiques, domaine bancaire, affaires ...).

D'un côté les usagers veulent des réseaux mobiles et flexibles qui s'adaptent facilement à leur besoin. De l'autre, les applications utilisées sont de plus en plus exigeantes en matière de qualité de service surtout en ce qui a trait au délai et à la bande passante. Il se trouve que si les réseaux ad hoc peuvent répondre adéquatement aux besoins des usagers pour ce qui concerne la mobilité et la flexibilité, il en est malheureusement autrement pour ce qui est des exigences en matière de qualité de service. Ainsi, bien que plusieurs recherches ont déjà abordé le sujet, l'intégration de la qualité de service dans les réseaux ad hoc demeure un problème ouvert et constitue toujours un défi.

En effet, les caractéristiques intrinsèques des réseaux ad hoc, rendent la prise en charge de la qualité de service relativement difficile (topologie changeante, contraintes d'énergies, bandes passante limitée ..).

Ce travail a pour objectif de proposer une nouvelle solution quant à la satisfaction des critères de qualité de service lors du processus de routage dans les réseaux ad hoc mobiles. Ainsi, il faut non seulement trouver une route entre une source et une destination dans le contexte dynamique des réseaux ad hoc, mais il faut en plus que cette route réponde aux critères de qualité de service définis lors de la demande de connexion.

Pour ce faire, nous avons proposé un protocole hybride HPQS inspiré du protocole ZRP et qui gère deux critères de qualité de service : le délai et la bande passante. En plus de la zone de routage *initiale* de ZRP, notre protocole intègre une zone de routage *externe* qui, elle est bâtie en tenant compte de la qualité de service. Le choix de ZRP comme protocole à la base de notre solution s'est fait à la lumière de la revue de littérature dans laquelle nous avons tenté de ressortir les avantages et les inconvénients de divers protocole. Nous avons alors constaté que le protocole ZRP tente de concilier les avantages des protocoles proactifs et réactifs.

Tout comme ZRP, le protocole HPQS tente donc de trouver un juste équilibre entre un processus de routage efficace et la limitation de la surcharge de signalisation du réseau.

L'idée de base du protocole HPQS et qui fait son originalité est de maintenir de manière proactive non seulement les routes vers les nœuds proches en terme de nombre de sauts comme le ferait ZRP mais également les routes vers les nœuds relativement éloignés si celles-ci respectent certains critères de qualité de service. Cela permet d'une part de répondre plus rapidement aux demandes qui exigent une certaine qualité de service et d'autre part de réparer pro activement et efficacement les routes respectant les critères QoS.

Dans les faits, nous avons présenté deux versions distinctes du protocole HPQS :

- HPQS-D : Version de HPQS qui tente de répondre aux demandes de connexion exigeant un délai maximum à ne pas dépasser ;

- HPQS-BP : Version de HPQS qui tente de répondre aux demandes de connexion exigeant une bande passante minimum.

Dans un cas comme dans l'autre, nous avons détaillé les algorithmes qui permettent de constituer la zone de routage externe et ceux qui permettent de répondre adéquatement à une demande de connexion avec des contraintes de QoS.

Afin de valider notre raisonnement théorique, nous avons utilisé le simulateur Qualnet dans lequel nous avons implémenté HPQS-D et HPQS-BP. Par la suite, nous avons comparé les performances de notre solution avec celles du protocole proactif Bellman-Ford (BF), du protocole hybride ZRP et des deux protocoles réactifs AODV et DSR.

Il en ressort que dans l'ensemble, notre solution est un bon compromis entre les protocoles proactifs et les protocoles réactifs. Plus encore, nous avons constaté que pour les réseaux denses (120 nœuds et plus), notre solution est meilleure en règle générale que les tous autres protocoles étudiés.

Nous pouvons donc affirmer que nous avons apporté une nouvelle solution au problème de la qualité de services dans les réseaux ad hoc.

5.2 Limitations des travaux

En dépit de l'intérêt de la solution que nous proposons, le travail réalisé dans le cadre de cette maîtrise, comporte comme tout œuvre, quelques limitations.

La première tient au fait que durant tout ce travail nous avons considéré les liens entre les nœuds comme étant symétriques ce qui ne correspond pas toujours à la réalité. De même, le délai (ou la bande passante) bout-en-bout n'est pas toujours égale dans un sens et dans l'autre. Cela implique que la route trouvée de la source vers la destination n'est pas forcément valide dans le sens inverse. Ainsi, pour tenir compte de cette réalité, il faudra que le protocole HPQS puisse gérer deux routes différentes pour chacun des sens. La seconde limitation est liée au patron de mobilité utilisé lors de la simulation. En effet, le patron *RANDOM_WAYPOINT* bien qu'il soit le plus largement utilisé dans la littérature ne correspond pas à la réalité.

La troisième limitation que nous pouvons soulever relève également de la simulation. Nous avons en effet considéré tous les nœuds comme ayant la même puissance CPU, la même taille des files d'attente, la même capacité en matière d'énergie... Or en réalité, les réseaux ad hoc sont généralement hétérogènes.

5.3 Travaux futurs

Notre travail ouvre la porte à plusieurs recherches possibles. Dans cette section, nous présenterons quelques pistes et suggestions pour les travaux futurs.

La première suggestion consiste à mesurer plus précisément l'impact de chacun des paramètres $DIST_1$, $DIST_2$, DEL et BP sur la performance des protocoles HPQS-D et HPQS-BP. En effet, nous avons vu dans l'analyse de performance que la taille de la zone de routage peut influencer le délai bout-en-bout des connexions. Il faudra approfondir cette étude et mesurer plus exactement comment chaque paramètre affecte les différentes métriques de performance. Cette étude sera en quelque sorte équivalente à celle effectuée pour ZRP par Pearlman et al. [35]. Il s'agira dans les faits de trouver pour chacun des paramètres les valeurs qui optimisent la performance des protocoles HPQS-D et HPQS-BP.

Une seconde proposition consiste à personnaliser pour chaque nœud, les valeurs des paramètres ci-dessus. En effet, si le réseau est hétérogène, il va sans dire que les mêmes valeurs ne peuvent correspondre à des nœuds de différentes natures. Une étude semblable a été réalisée pour le protocole ZRP [36].

Dans un autre ordre d'idée, on pourrait également comparer les performances de HPQS-D et HPQS-BP avec d'autres protocoles avec QoS tels que TBF.

Une quatrième et dernière suggestion reviendrait à comparer la technique de réparation de route de HPQS (héritée de celle du protocole ZRP) par rapport aux autres techniques recensées dans la littérature.

BIBLIOGRAPHIE

- [1] C. Perkins "IP Mobility Support," ed., IETF RFC 2002, 1996
- [2] X. Hong, K. Xu et M. Gerla "Scalable Routing Protocols for Mobile Ad Hoc Networks", *IEEE Network*, 2002
- [3] T. Clausen, P.Jacquet et al., "Optimized Link State Protocol", draft-ietf-manet-11.txt, Internet Draft, IETF MANET Working Group, 2003
- [4] G. Pei, M.Gerla, et T.-W. Chen, "Fisheye State Routing: A Routing Sheme for Ad Hoc Wireless Network", Proc. *ICC 2000*, New Orleans, LA, 2000
- [5] R. G. Ogier, F. L. Templin, B. Bellur, et M. G. Lewis. "Topology broadcast based on reverse-path forwarding (tbrpf) ",draft-ietf-manet-tbrpf-05.txt, Internet Draft, IETF MANET Working Group, 2002.
- [6] C. Perkins, T.J. Watson, "Highly dynamic destination sequenced distance vector routing (DSDV) for mobile computers", *ACM SIGCOMM_94 Conference on Communications Architectures*, 1994.
- [7] D.B. Jonhson et D.A Maltz, "Dynamic Source Routing in Ad Hoc Wireless Network", *Mobile Computing*, T. Imielinski et H. Korth, Chapitre 5, Kluwer, 1996, pp. 88-153
- [8] C. Perkins et E.M Royer, "Ad-hoc on Demand Distance Vector Routing", Proc. *IEEE WMCSA*, LA, 1999, pp 90-100.
- [9] C. C. Chiang, H-K Wu, Winston Liu, et Mario Gerla, "Routing in clustered multihop, mobile wireless networks", *The IEEE Singapore International Conference on Networks*, pp 197-211, 1997.
- [10] G. Pei et al. "A Wireless Hierarchical Routing Protocol with Group Mobility", Proc. *IEEE WCNC '99*, 1999
- [11] Z.J. Hass, R. Pearlman, "Zone routing protocol for ad-hoc networks", Internet Draft, draft-ietf-manet-zrp-02.txt, 1999.
- [12] R Braden., D. Clark. et S. Shenker, "Integrated services in the internet architecture: on overview", RFC 1633, IETF, 1994.

- [13] K. Nichols, S. Blake, F. Baker, D. Black, "Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers", RFC 2474, 1998.
- [14] H. Xiao, W.K.G. Seah, A. Lo, et K.C. Chua. "A flexible quality of service model for mobile ad hoc networks", *IEEE Vehicular Technology Conference*, pp. 445-449, 2000.
- [15] S. Hongxia et H. Hughes "Adaptive QoS routing based on prediction of local performance in ad hoc network", *IEEE International Conference on Personal Wireless Communications*, 2000 pp. 340-344
- [16] M. Gerla et C.R. Lin. "MACA/PR: An asynchronous multimedia multihop wireless network", *IEEE INFOCOM 97*, 1997.
- [17] I. Aad and C. Castelluccia. "Differentiation mechanisms for IEEE 802.11.IEEE", *Infocom 2001*, 2001.
- [18] S. Chen. "Routing support for providing guaranteed end-to-end quality of service", *thèse de doctorat*, University of Illinois at Urbana-Champaign, 1999.
- [19] S. Chen et K. Nahrstedt, "Distributed Quality-of-Service Routing in Ad Hoc Networks", *IEEE Journal on selected areas in communications*, vol. 17, No. 8, pp 1488-1505, 1999.
- [20] P. Sinha, R. Sivakumar, and V. Bharghavan. "CEDAR : a core extraction distributed ad hoc routing algorithm". *IEEE Journal on selected areas in communications*, vol. 17, No. 8, pp 1454-1465, 1999
- [21] S.H Shah et K. Nahrstedt, "Predictive location-based QoS routing in mobile ad hoc networks", *IEEE International Conference in communications*, 2002, pp 1022-1027.
- [22] S.B. Lee, G.S. Ahn, X. Zhang, and A.T. Campbell. "INSIGNIA : An ip-based quality of service framework for mobile ad hoc network". *Journal of Parallel and Distributed Computing (Academic Press)*, vol. 60, No.4, pp. 374-406, 2000
- [23] M. Mirhakkak, N. Schult, and D. Thomson. "Dynamic quality-of-service for mobile ad hoc networks". *IEEE MobiHoc 2000*, 2000.
- [24] A. Quintero, S. Pierre, B. Macabéo, "A routing protocol based on node density for ad hoc networks", *Ad Hoc Networks*, Vol. 2, No. 3, 2004, pp.335-349.
- [25] R. Duggirala, R. Gupta, Q. Zeng et D.P Agrawal, "Performance Enhancements of Ad Hoc Networks with Localized Route Repair", *IEEE Transactions on Computers*, vol. 52, No. 7, 2003.

- [26] C. Chaudet, *Qualité de service et réseaux ad-hoc, un état de l'art*, INRIA, 2001
- [27] Z.J. Haas, M.R. Pearlman, and P. Samar, "Intrazone Routing Protocol (IARP)," IETF Internet Draft, draft-ietf-manet-iarp-02.txt, 2002.
- [28] Z.J. Haas, M.R. Pearlman, and P. Samar, "Interzone Routing Protocol (IERP)," IETF Internet Draft, draft-ietf-manet-ierp-02.txt, 2002.
- [29] Z.J. Haas, M.R. Pearlman, and P. Samar, "Bordercasting Resolution Protocol (BRP)," IETF Internet Draft, draft-ietf-manet-brp-02.txt, 2002.
- [30] M.R Pearlman, Z.J Haas, and S.I. Mir, "Using Routing Zones to Support Route Maintenance in Ad Hoc Networks," IEEE WCNC 2000, Chicago, IL, 2000
- [31] The Network Simulator—NS-2, <http://www.isi.edu/nsnam/ns>, 2005
- [32] OPNET Technologies, Inc. <http://www.opnet.com/>, 2005
- [33] M. Gerla, R. Bagrodia, X. Zeng, "GloMoSim: A Library for Parallel Simulation of Large-Scale Wireless Networks", *12th ACM Workshop on Parallel And Distributed Simulations (PADS)*, pp. 154-161, 1998.
- [34] Scalable Network Technologies (SNT), "QualNet 3.8 Product Tour", 2004, 93 pages, <http://www.scalable-networks.com/>.
- [35] Pearlman, M.R. and Haas, Z.J., "Determining the Optimal Configuration for the Zone Routing Protocol," IEEE JSAC, 1999.
- [36] P. Samar, M.R. Pearlman and Z.J. Haas, "Independent zone routing: an adaptive hybrid routing framework for ad hoc wireless networks", IEEE/ACM Transactions on Networking (TON), vol. 12, No. 4, pp 595-608.