

Titre: Routage géographique dans les réseaux mobiles ad hoc
Title:

Auteur: Valérie Alandzi
Author:

Date: 2005

Type: Mémoire ou thèse / Dissertation or Thesis

Référence: Alandzi, V. (2005). Routage géographique dans les réseaux mobiles ad hoc
[Master's thesis, École Polytechnique de Montréal]. PolyPublie.
Citation: <https://publications.polymtl.ca/7580/>

Document en libre accès dans PolyPublie

Open Access document in PolyPublie

URL de PolyPublie: <https://publications.polymtl.ca/7580/>
PolyPublie URL:

Directeurs de recherche: Alejandro Quintero
Advisors:

Programme: Unspecified
Program:

UNIVERSITÉ DE MONTRÉAL

ROUTAGE GÉOGRAPHIQUE
DANS LES RÉSEAUX MOBILES AD HOC

VALÉRIE ALANDZI

DÉPARTEMENT DE GÉNIE INFORMATIQUE
ÉCOLE POLYTECHNIQUE DE MONTRÉAL

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

DÉCEMBRE 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: 978-0-494-16749-6

Our file *Notre référence*
ISBN: 978-0-494-16749-6

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 GÉOGRAPHIQUE
DANS LES RÉSEAUX MOBILES AD HOC

Présenté par : VALÉRIE ALANDZI

En vue de l'obtention du diplôme de : Maîtrise ès Sciences Appliquées

A été dûment accepté par le jury d'examen constitué de :

M. FERNANDEZ José, Ph.D., président

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

M. PIERRE Samuel, Ph.D., membre

DÉDICACE

A ma mère Christiane,

A mon père Jean-Marie,

A mon frère Patrick,

A mon compagnon Patrice.

REMERCIEMENTS

Mes premiers remerciements vont à mon directeur de recherche, Monsieur Alejandro Quintero, professeur du département de génie informatique, pour ses encouragements, ses suggestions, ainsi que sa disponibilité.

Je tiens aussi à remercier Monsieur Samuel Pierre, professeur du département de génie informatique et directeur du LARIM (Laboratoire de recherche en réseautique et informatique mobile), pour son accueil.

J'aimerais par ailleurs remercier les membres du LARIM pour leur soutien et l'ambiance de travail.

Finalement, je voudrais exprimer toute ma gratitude à ma famille, à Melissa et surtout à Patrice, pour leur patience et le soutien moral qu'ils m'ont apporté tout au long de la réalisation de ces travaux.

RÉSUMÉ

La plupart des algorithmes de routage existant pour les réseaux ad hoc sont destinés à être utilisés dans des réseaux contenant quelques centaines de nœuds. Ces algorithmes reposent sur le stockage des informations sur tous les liens du réseau ou sur les liens d'une route d'une source à une destination. Les performances de ces approches sont par conséquent altérées dans les réseaux de grande dimension ou à mobilité élevée. L'utilisation de la position géographique dans le routage est souvent proposée comme moyen de garantir l'efficacité du routage malgré les changements d'échelle du réseau. Le routage géographique s'avère cependant difficile dans un réseau dispersé avec des "trous" dans sa topologie.

L'acheminement géographique, dans sa forme de base, est en effet susceptible d'atteindre un optimum local lorsqu'un noeud n'a pas de voisin plus proche de la destination que lui-même. Le noeud est alors dit concave. Diverses approches ont été proposées pour retrouver un contexte où le routage géographique est possible. La plus utilisée est le recouvrement par mode périmètre, qui consiste en un détour autour du noeud concave. Si cette approche s'avère relativement efficace dans un réseau dense, elle engendre souvent la formation de boucles et la longueur des routes est loin d'être optimale. Par ailleurs, ses performances sont grandement altérées dans les réseaux dispersés.

Nous proposons un nouvel algorithme d'acheminement géographique, PARA (*Proximity Aware Routing for Ad-hoc networks*), pour remédier à ce problème. PARA choisit le prochain nœud transmetteur d'un paquet de données, non seulement en fonction de son voisinage direct, mais aussi en fonction de son voisinage à deux sauts. Nous introduisons ainsi le concept de "découverte de proximité". L'étude du voisinage à deux sauts permet d'anticiper la présence, mais surtout de minimiser le risque d'atteinte de "trous" dans le réseau, où l'avancée géographique du paquet vers la destination n'est plus possible. PARA se veut aussi un compromis entre le routage géographique et

ABSTRACT

Most of the existing routing protocols for ad hoc networks are designed to scale in networks of a few hundred nodes. They rely on state concerning all links of the network or links on a route between a source and a destination. This may result in poor scaling properties in larger mobile networks or when nodes mobility is high. Using location information to help routing is often proposed as a means to achieve scalability in large mobile networks. However, location-based routing is difficult when there are holes in the network topology.

Thus, the power of position-based forwarding in its basic form comes with one drawback: when no node is closer to the destination than the current node, a local maximum is reached. Then the current node is said concave. Many approaches were proposed to recover from this local optimum. GPSR uses perimeter forwarding to route the packet around the problem region. However, perimeter mode may give a very bad path in networks where the source and the destination are not well connected along a straight line. Furthermore, perimeter mode may lead to routing loops and thus to packet drops.

We propose a novel position-based routing protocol, PARA (*Proximity Aware Routing for Ad-hoc networks*), to address these issues. PARA selects the next hop of a packet not only based on its direct neighborhood, but also on its 2-hops neighborhood. Thus we introduce the concept of “proximity discovery”. The knowledge of 2-hops neighborhood enables the anticipation of concave node and helps to reduce the risks to reach a hole in the network. PARA also stands for a compromise between position and topology-based routing as packets that cannot be forwarded with PARA geographical mode are routed using topology-based routing.

Our simulation results show that PARA performance is optimal in sparse networks with little congestion. Then, PARA significantly outperforms GPSR for the delivery ratio, transmission delay and path length. Furthermore, in this case, PARA delivers even more data packets than AODV.

topologique. Les paquets ne pouvant être acheminés selon le mode géographique de PARA le sont en effet avec un routage topologique.

Les résultats de nos simulations montrent que les performances de PARA sont optimales dans les réseaux dispersés et moyennement congestionnés. PARA surpasse alors largement GPSR pour ce qui est du taux de livraison et du délai moyen de transmission des paquets de données, ainsi que de la longueur moyenne des routes. Par ailleurs, dans de tels contextes, PARA délivre plus de paquets que AODV.

TABLE DES MATIÈRES

DÉDICACE	IV
REMERCIEMENTS	V
RÉSUMÉ	VI
ABSTRACT	VIII
TABLE DES MATIÈRES	IX
LISTE DES TABLEAUX.....	XII
LISTE DES FIGURES.....	XIV
LISTE DES ANNEXES.....	XVII
LISTE DES SIGLES ET ABRÉVIATIONS	XVIII
CHAPITRE I INTRODUCTION.....	1
1.1 Définitions et concepts de base	2
1.2 Éléments de la problématique	4
1.3 Objectifs de la recherche	6
1.4 Plan du mémoire	7
CHAPITRE II ROUTAGE DANS LES RÉSEAUX AD HOC	8
2.1 Routage basé sur la topologie du réseau	10
2.1.1 Routage proactif.....	10
2.1.2 Routage réactif	12
2.2 Prise en compte de la position géographique dans le routage.....	14
2.2.1 Concepts de base	15
2.2.2 Stratégies d'acheminement	17
2.3 Routage géographique.....	21
2.3.1 Routage géographique greedy.....	22
2.3.2 Routage dirigé et multi-chemins (<i>directed forwarding</i>)	25

2.3.3	Routage hiérarchique	30
2.3.4	Routage géographique sans beacons.....	31
2.4	Récapitulatif	33
	CHAPITRE III PROTOCOLE DE ROUTAGE LOCALISÉ PROPOSÉ	36
3.1	Motivations et objectifs.....	36
3.2	Hypothèses	38
3.3	Principe	38
3.3.1	Concepts de base	41
3.3.2	Stockage des informations	43
3.3.3	Initialisation de la phase de transmission.....	43
3.3.4	Déroulement d'une découverte de proximité.....	45
3.3.5	Sélection du nœud transmetteur	57
3.3.6	Transmission des paquets de données.....	59
3.3.7	Mise à jour de la table de meilleurs voisins	61
3.3.8	Atteinte de la destination.....	62
3.3.9	Détection des nœuds concaves.....	67
3.3.10	Gestion des nœuds concaves.....	69
3.4	Analyse mathématique	75
3.4.1	Modèle du réseau	75
3.4.2	GPSR.....	75
3.4.3	AODV	75
3.4.4	PARA	77
3.4.5	Ebauche de résultats.....	80
	CHAPITRE IV IMPLÉMENTATION ET ANALYSE DE PERFORMANCE.....	82
4.1	Le simulateur NS.....	82
4.2	Implémentation de PARA	83
4.2.1	Classes implémentées pour le routage PARA.....	83
4.2.2	Entête des paquets	83

4.2.3	Implémentation des temporiseurs.....	84
4.2.4	Extensions apportées à l'interface Tcl de NS	85
4.2.5	Choix significatifs	86
4.3	Evaluation de performance de PARA	90
4.3.1	Environnement de simulation	90
4.3.2	Résultats des simulations	95
4.4	Interprétation des résultats	107
CHAPITRE V CONCLUSION		115
5.1	Synthèse des travaux	115
5.2	Limitation des travaux.....	116
5.3	Travaux futurs	117
BIBLIOGRAPHIE		119
ANNEXES		123

LISTE DES TABLEAUX

Tableau 3.1: Contenu de la table de meilleurs voisins de N pour la Figure 3.8.....	68
Tableau 3.2 : Paramètres globaux utilisés pour l'analyse numérique	76
Tableau 3.3: Paramètres utilisés pour l'analyse numérique de AODV	76
Tableau 3.4 : Paramètres utilisés pour l'analyse numérique de PARA	78
Tableau 4.1 : Valeurs de gigues et délais.....	89
Tableau 4.2 : Paramètres globaux de simulation	91
Tableau 4.3: Scénarios de connectivité.....	93
Tableau 4.4: Paramètres de PARA.....	94
Tableau 4.5 : Paramètres de GPSR	95
Tableau E.1 : Format d'une entrée du cache de PREQ1s au niveau d'un noeud N.....	143
Tableau E.2 : Format d'une entrée de la table de meilleurs voisins	144
Tableau E.3 : Format d'une entrée du cache de nœud concave	144
Tableau E.4 : Format d'une entrée du cache de PREQ2s	144
Tableau F.1 : Format de l'entête PARA des paquets de données	145
Tableau F.2 : Format de l'entête d'une PREQ1	145
Tableau F.3 : Format de l'entête d'une PREQ2_ONE.....	146
Tableau F.4 : Format de l'entête d'une PREQ2	146
Tableau F.5 : Format de l'entête d'une PREP2	146
Tableau F.6 : Format de l'entête d'une PREP1	147
Tableau G.1 : Contenu des caches de PREP1s pour la Figure 3.4.....	148
Tableau G.2 : Contenu de la table de meilleurs voisins de N pour la Figure 3.4.....	149
Tableau H.1 : Contenu des caches de PREP1s initialisés pour la Figure annexée D.1 .	150
Tableau H.2 : Contenu de l'entête d'une PREQ2 pour la Figure D.1.....	151
Tableau H.3 : Contenu de l'entête de la PREP2 diffusée par G pour la Figure D.1	151
Tableau H.4 : Contenu des caches de PREP1s pour la Figure D.1	152
Tableau H.5 : Contenu de l'entête de la PREP1 diffusée par A pour la Figure D.1	152
Tableau H.6 : Contenu de la table de meilleurs voisins de N pour la Figure D.1	153

Tableau I.1: Contenu de la table de meilleurs voisins de N pour la Figure 3.8	154
Tableau M.1 : Résultats obtenus dans un réseau de connectivité 21 (1500x300m)	166
Tableau M.2 : Résultats obtenus dans un réseau de connectivité 13 (500x500m)	167
Tableau M.3 : Résultats obtenus dans un réseau de connectivité 5 (1500x1300m)	168

LISTE DES FIGURES

Figure 1.1 : Exemple d'environnement mobile avec infrastructure fixe	3
Figure 1.2 : Effet de la mobilité d'un nœud sur la topologie d'un réseau ad hoc	4
Figure 2.1 : Représentation en graphe unitaire d'un réseau ad hoc	8
Figure 2.2 : Découverte de route dans AODV	15
Figure 2.3 : Stratégies de routage greedy.....	19
Figure 2.4 : Exemple de boucle dans le routage directionnel	20
Figure 2.5 : Echec du routage greedy.....	21
Figure 2.6 : Exemple d'acheminement par périmètre.....	23
Figure 2.7 : Exemple de nœud concave dans GEDIR.....	26
Figure 2.8 : Exemple de région.....	28
Figure 2.9 : Exemple de régions supposée et inspectée dans LAR.....	30
Figure 2.10 : Secteur de 60 degrés dans BLR.....	32
Figure 3.1 : Acheminement d'un paquet avec PARA.....	40
Figure 3.2 : Nœud concave par anticipation	43
Figure 3.3 : Algorithme décrivant le comportement d'un noeud initiant une communication.....	45
Figure 3.4 : Échanges de paquets de contrôle dans une découverte de proximité à destination unique	55
Figure 3.6 : Rallongement de la route vers la destination.....	59
Figure 3.7: PARA, algorithme de routage de base.....	61
Figure 3.8 : Exemple de déplacement de meilleurs voisins.....	68
Figure 3.9 : Passage en mode Recouvrement.....	70
Figure 3.10 : Changements de mode de routage	74
Figure 4.1: Structure <i>hdr_para</i>	83
Figure 4.2 : Classe <i>PrequestTimer</i>	84
Figure 4.3: Extension apportée à l'interface Tcl.....	86

Figure 4.4 : Impact de la notification de la couche Mac sur le taux de réception des paquets de données.....	88
Figure 4.5 : Taux de réception des paquets de données dans un réseau de 50 nœuds dispersés (5 nœuds en moyenne) pour (a) 10, (b) 20 et (c) 30 connexions CBR.....	96
Figure 4.6 : Trafic de contrôle normalisé dans un réseau de 50 nœuds dispersés (5 nœuds en moyenne) pour (a) 10, (b) 20 et (c) 30 connexions CBR.....	97
Figure 4.7 : Délai de transmission moyen dans un réseau de 50 nœuds dispersés (5 nœuds en moyenne) pour (a) 10, (b) 20 et (c) 30 connexions CBR.....	98
Figure 4.8 : Longueur moyenne des routes dans un réseau de 50 nœuds dispersés (5 nœuds en moyenne) pour (a) 10, (b) 20 et (c) 30 connexions CBR.....	99
Figure 4.9 : Taux de réception des paquets de données dans un réseau de 50 nœuds moyennement connectés (13 nœuds en moyenne) pour (a) 10, (b) 20 et (c) 30 connexions CBR	101
Figure 4.10 : Délai moyen de transmission dans un réseau de 50 nœuds moyennement connectés (13 nœuds en moyenne) pour (a) 10, (b) 20 et (c) 30 connexions CBR	102
Figure 4.11: Longueur moyenne des routes dans un réseau de 50 nœuds moyennement connectés (13 nœuds en moyenne) pour (a) 10, (b) 20 et (c) 30 connexions CBR.....	103
Figure 4.12 : Taux de réception dans un réseau dense de 50 nœuds (21 nœuds en moyenne) pour (a) 10, (b) 20 et (c) 30 connexions CBR.....	104
Figure 4.13 : Délai moyen de transmission dans un réseau dense de 50 nœuds (21 nœuds en moyenne) pour (a) 10, (b) 20 et (c) 30 connexions CBR.....	105
Figure 4.14 : Longueur des routes dans un réseau dense de 50 nœuds (21 nœuds en moyenne) pour (a) 10, (b) 20 et (c) 30 connexions CBR.....	106
Figure 4.15 : Proportion de paquet délivrée par PARA par type d'acheminement dans un réseau de 50 noeuds dispersé (5 noeuds) pour (a) 10, (b) 20 et (c) 30 connexions CBR	109

Figure 4.16 : Proportion de paquet délivrée par PARA par type d'acheminement dans un réseau dense de 50 nœuds (21 noeuds) pour (a) 10, (b) 20 et (c) 30 connexions CBR	110
Figure 4.17 : Trafic de contrôle normalisé dans un réseau de 100 nœuds avec 10 sources	113
Figure B.1 : Déroulement des opérations de AGPF	128
Figure B.2 : Un exemple de découverte de route dans GRID.....	130
Figure D.1 : Échanges de paquets lors d'une découverte de proximité à destinations multiples.....	141
Figure D.2 : Déroulement d'une découverte de proximité à destinations multiples.....	142
Figure K.1 : Diagramme de classes de PARA	158
Figure L.1 : Structure <i>hdr_para_preq1</i> décrivant les champs de l'entête d'une PREQ1	162
Figure L.2 : Structure <i>hdr_para_prep1</i> décrivant les champs de l'entête d'une PREP1	163
Figure L.3 : Structure <i>hdr_para_data</i> décrivant les champs de l'entête PARA d'un paquet de données	164
Figure N.1 : Taux de réception des paquets de données dans un réseau de 100 nœuds dispersés (5 nœuds en moyenne) pour (a) 10 et (b) 30 connexions CBR	169
Figure N.2 : Délai de transmission moyen dans un réseau de 100 nœuds dispersés (5 nœuds en moyenne) pour (a) 10 et (b) 30 connexions CBR	169
Figure N.3 : Longueur moyenne des routes dans un réseau de 100 nœuds dispersés (5 nœuds en moyenne) pour (a) 10 et (b) 30 connexions CBR	170

LISTE DES ANNEXES

Annexe A : Services de localisation.....	124
Annexe B : Protocoles de routage géographique hiérarchiques.....	127
Annexe C : Structures de stockages mises en œuvre par PARA	132
Annexe D : Déroulement d'une découverte de proximité à destinations multiples.....	134
Annexe E : Format du stockage des informations avec PARA	143
Annexe F : Format des entêtes PARA	145
Annexe G : Exemples de stockage de données dans le cadre de l'exemple de la Figure 3.4	148
Annexe H : Exemples de stockage de données et d'entêtes PARA dans le cadre de l'exemple de la Figure annexée D.1.....	150
Annexe I : Contenu de la table de meilleurs voisins de N pour l'exemple de la Figure 3.6	154
Annexe J : Détails d'implémentation du simulateur NS.....	155
Annexe K : Classes implémentées pour le routage PARA dans NS.....	158
Annexe L : Structures décrivant les entêtes des paquets PARA.....	162
Annexe M : Evaluation de l'impact des paramètres discv_delay et reply_delay sur la performance de PARA	165
Annexe N : Résultats de quelques simulations dans des réseaux ad hoc de 100 nœuds	169

LISTE DES SIGLES ET ABRÉVIATIONS

AGPF	: Anchored Geodesic Packet Forwarding
AODV	: Ad hoc on-demand distance vector
BLR	: Beacon Less Routing
BS	: Base Station
BSC	: Base Station Controller
BTS	: Base Transceiver Station
CBF	: Contention-Based Forwarding
DFS	: Depth First Search
DLS	: DREAM location service
DREAM	: Distance routing effect algorithm for mobility
DSDV	: Destination-sequenced distance vector
DSR	: Dynamic Source Routing
GEDIR	: <u>Geographic Distance</u> Routing
GLS	: Grid Location Service
GPS	: Global Positioning System
GPSR	: Greedy Perimeter Stateless Routing
GRA	: Geographical Routing Algorithm

GRSS	:	Geographic Region Summary Service
HLR	:	Home Location Register
IP	:	Internet Protocol
LAR	:	Location-aided routing
MANET	:	Mobile Service Switching Center
MFR	:	Most Forward within R
MSC	:	Mobile Service Switching Center
NFP	:	Nearest with Forward Progress
PARA	:	Proximity Aware Routing for Ad-hoc networks
PREP	:	Proximity Reply
PREQ	:	Proximity Request
RERR	:	Route <u>Error</u>
RREP	:	Route <u>Reply</u>
RREQ	:	Route <u>Request</u>
RTPC	:	Réseau téléphonique public commuté
TLR	:	Terminode Local Routing
TRR	:	Terminode Remote Routing
VHR	:	Virtual Home Region

VLR : Visitor Location Register

CHAPITRE I

Introduction

Les technologies de communication sans fil connaissent aujourd’hui un essor remarquable. La quantité de terminaux mobiles et d’équipements de réseautage sans fil disponible sur le marché témoigne en effet de l’importance de ces technologies dans la vie de tous les jours. Les technologies sans fil répondent à un besoin réel : les usagers veulent être en mesure d’accéder à des informations données à partir de diverses plateformes électroniques et ce, n’importe où et n’importe quand. Les réseaux informatiques actuels sont par conséquent désormais axés vers l’ubiquité. L’intégration des réseaux sans fil aux réseaux traditionnels filaires tels qu’Internet est donc cruciale, notamment pour le support de différents types de trafic de données (voix, vidéo,...). Les utilisateurs mobiles peuvent cependant vouloir communiquer dans des contextes où de telles infrastructures fixes ne sont pas disponibles, que ce soit parce qu’il est économiquement ou physiquement impossible de fournir l’infrastructure nécessaire ou parce que l’urgence de la situation ne permet pas son installation. Une classe d’étudiants peut ainsi avoir besoin d’interagir durant un cours, des hommes d’affaire sont susceptibles de vouloir partager des fichiers dans le hall d’un aéroport ou encore un groupe de sauveteurs doit avoir la possibilité de se déployer rapidement après un tremblement de terre ou une inondation. Dans de telles situations, un ensemble de terminaux mobiles avec des interfaces réseau sans fil peut former un réseau sans l’aide de la moindre infrastructure ou administration centralisée. Ce type de réseau est appelé réseau “ad hoc”. La gestion des communications dans un environnement sans fil où les terminaux sont mobiles est très différente et plus complexe que dans un réseau filaire où les terminaux sont fixes. Dans ce chapitre d’introduction, nous présenterons, dans un premier temps, les définitions et concepts nécessaires à la bonne compréhension du mémoire. Nous préciserons ensuite les éléments de la problématique et les objectifs de la recherche avant de terminer avec une ébauche du plan du mémoire.

1.1 DÉFINITIONS ET CONCEPTS DE BASE

Les systèmes cellulaires constituent le modèle traditionnel pour les réseaux mobiles sans fil. Le territoire couvert dans de tels réseaux est divisé en cellules, chacune étant associée à une station de base ou BS (*Base Station*). Les communications sont alors établies par des commutateurs ou MSC (*Mobile Service Switching Center*) et peuvent être relayées par le réseau téléphonique public commuté (RTPC). Les réseaux cellulaires reposent donc sur une infrastructure fixe, comme le montre la Figure 1.1, qui peut parfois s'avérer coûteuse et mal adaptée dans certains contextes.

Un réseau ad hoc est quant à lui un réseau mobile sans fil, temporaire, constitué d'un ensemble de terminaux mobiles susceptibles d'établir des communications les uns avec les autres et ce, sans infrastructure préexistante quelconque. Les terminaux mobiles qui constituent les nœuds d'un réseau ad hoc sont libres de se déplacer arbitrairement et combinent les rôles de routeur et de station d'hôte. L'acheminement des données d'un terminal à un autre se fait ainsi de façon autonome. Cette capacité d'organisation autonome fait des réseaux ad hoc les systèmes les mieux adaptés à des contextes spécifiques, tels que les interventions sur des sites de désastres, les conférences, les opérations militaires ou encore le déploiement de réseaux de capteurs.

Une route entre un nœud source et un nœud destination d'un réseau ad hoc est constituée de l'ensemble de nœuds intermédiaires du réseau par lesquels doivent transiter les données pour atteindre le nœud destinataire. Le routage combine les opérations de découverte de route et les opérations de maintenance de route.

Dans le routage géographique, les décisions de routage sont effectuées selon la position géographique des nœuds du réseau ad hoc. Les paquets de données sont acheminés vers le prochain nœud dans la direction du nœud destinataire jusqu'à ce que les paquets atteignent ce dernier. De fait, les nœuds doivent être en mesure de déterminer leur position.

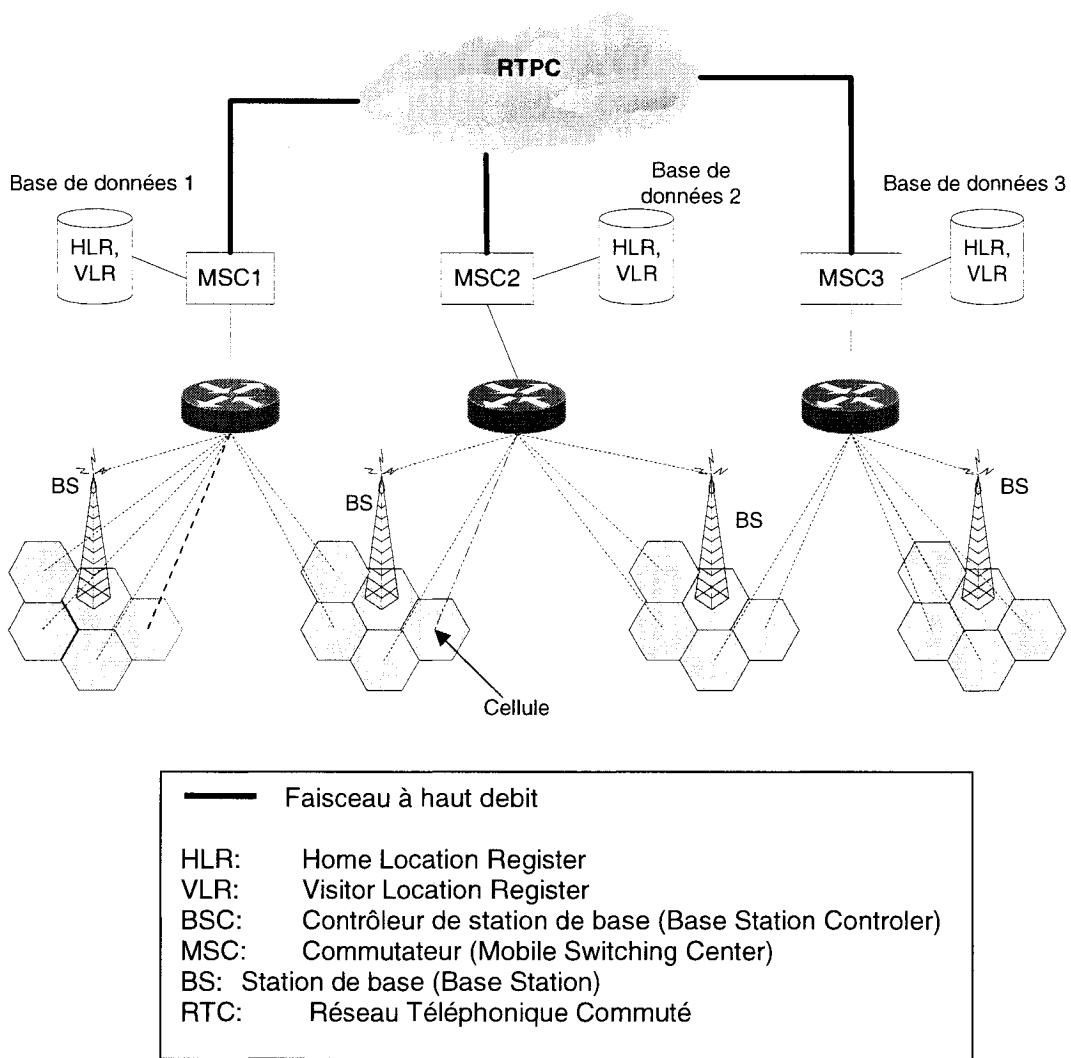


Figure 1.1 : Exemple d'environnement mobile avec infrastructure fixe

La disponibilité de petits récepteurs de systèmes de positionnement global ou GPS (*Global Positioning System*) à faible coût a eu un impact décisif dans le développement du routage géographique dans les réseaux ad hoc. Un tel récepteur permet en effet au terminal qui en est doté de connaître sa position géographique, mais aussi sa vitesse et sa direction. Par ailleurs, un nœud est en mesure de déterminer la position géographique d'un autre nœud du réseau par le biais d'un service de localisation, tel que VHR [24] ou GLS [25]. Ce type de service a pour fonction de faire correspondre dynamiquement

l'adresse logique d'un terminal mobile à sa position géographique courante dans le réseau.

1.2 ÉLÉMENTS DE LA PROBLÉMATIQUE

Le routage dans les réseaux ad hoc fait face à de nombreuses contraintes liées à la nature même des réseaux ad hoc. En premier lieu, la mobilité imprévisible des nœuds rend la topologie du réseau instable et sujette à des changements fréquents comme le montre la Figure 1.2. De plus, le fait que les terminaux mobiles disposent de sources d'énergie autonome engendre des contraintes d'énergie. En outre, la bande passante est limitée et n'autorise donc pas de lourds paquets de contrôle.

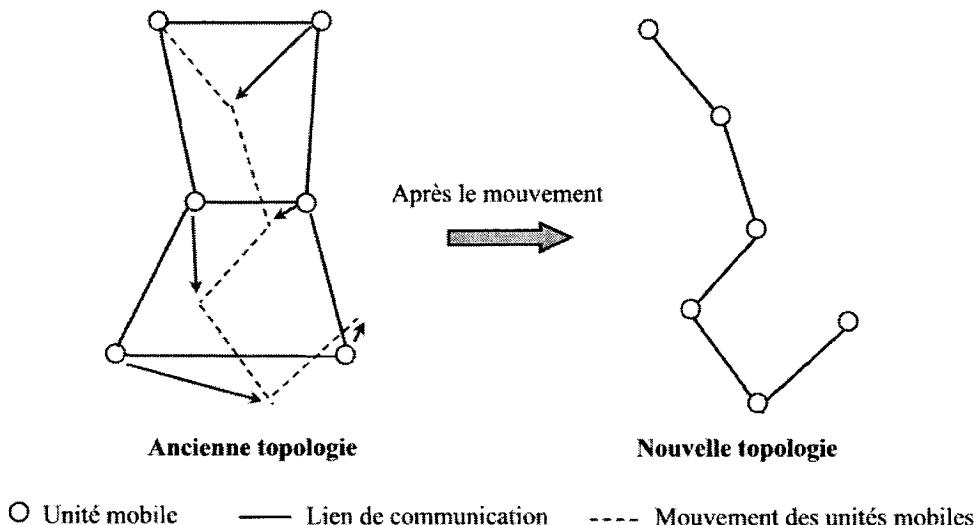


Figure 1.2 : Effet de la mobilité d'un nœud sur la topologie d'un réseau ad hoc

Deux approches de routage basées sur la topologie du réseau tentant de pallier à ces contraintes ont été élaborées : il s'agit du routage proactif et du routage réactif.

Dans un routage proactif, chaque nœud maintient constamment, soit pro activement, une route vers tous les autres nœuds du réseau. La création et la maintenance des routes sont réalisées par des échanges de paquets à la fois périodiques et discrets, déclenchés par des événements (par exemple la rupture d'un lien). Le principal inconvénient de cette

approche est que la maintenance des routes nécessite un trafic de contrôle important lorsque la mobilité du réseau augmente.

Au contraire, dans un routage réactif, une route entre deux nœuds n'est découverte que lorsqu'une communication doit être établie. Certes, le trafic de contrôle est ainsi réduit, mais la découverte des routes engendre un délai avant la transmission effective des données. D'autre part, bien que la maintenance des routes n'ait lieu que lorsqu'une communication est établie, le trafic engendré demeure important.

Ainsi, dans les deux approches, le trafic de contrôle est susceptible, selon la dimension du réseau, de consommer une grande partie de la bande passante au détriment de paquets de données. Réduire le trafic de contrôle de routage devient donc un enjeu primordial pour le routage dans les réseaux ad hoc à grande échelle.

Le routage géographique élimine pour sa part certaines limitations des approches de routage réactif et proactif. Les algorithmes de routage géographique sont distribués par nature et se rapprochent ainsi des algorithmes dits "gloutons", dans la mesure où un comportement local permet d'atteindre un objectif global. En effet, dans un routage géographique, chaque nœud choisit le prochain nœud transmetteur d'un paquet de données en se basant uniquement sur sa propre position géographique, celles de ses nœuds voisins, et celle de la destination. Selon le mode de sélection du prochain nœud transmetteur, trois stratégies peuvent être définies, offrant chacune une optimisation différente :

- La minimisation du nombre de nœuds transmetteurs : le prochain nœud transmetteur est celui le plus proche du nœud destinataire,
- La minimisation de la distance: le prochain nœud est le nœud voisin le plus proche de la ligne virtuelle entre les nœuds source et destinataire,
- La minimisation de la consommation d'énergie : cette approche consiste à choisir le prochain saut comme étant le nœud le plus proche dans la direction de la destination. La puissance nécessaire pour la transmission est alors réduite.

Les algorithmes de routage géographique permettent de limiter le trafic de contrôle dans la mesure où seules les positions géographiques des nœuds sont nécessaires pour

effectuer le routage. Par ailleurs, les informations de localisation peuvent être utilisées dans un réseau sensible au contexte ou dans des applications spécifiques à des zones géographiques telles que la découverte de services et de ressources. Le routage géographique constitue par conséquent une alternative intéressante de routage pour les réseaux ad hoc, bien que comportant quelques faiblesses. En premier lieu, l'utilisation de la position des nœuds pose des problèmes évidents de fiabilité. La précision de la position du nœud destinataire est donc à considérer. De plus, comme tous les algorithmes “gloutons”, les algorithmes de routage géographique sont susceptibles d'atteindre des optimums locaux, dans lesquels un nœud n'a aucun nœud voisin plus proche que lui-même du nœud destinataire. Les paquets sont alors bloqués au nœud courant. Bien que des stratégies de recouvrement aient été proposées pour sortir de ces optimums, le problème est toujours sujet à recherche. Enfin, il faut développer des algorithmes de routage géographique prenant en compte des critères de qualité de service, tels que la bande passante disponible, le délai de transmission et la congestion du réseau.

1.3 OBJECTIFS DE LA RECHERCHE

L'objectif principal de ce mémoire est de proposer un nouveau mécanisme de routage géographique dans les réseaux ad hoc adapté à des systèmes évolutifs à grande échelle. Nos objectifs sont les suivants:

- Analyser les solutions existantes en matière de routage afin d'en souligner les faiblesses;
- Concevoir et implémenter un protocole permettant une meilleure utilisation de la position géographique des nœuds dans les réseaux ad hoc;
- Évaluer la performance de ce protocole en le comparant aux meilleurs algorithmes répertoriés dans la littérature.

1.4 PLAN DU MÉMOIRE

Ce mémoire est organisé comme suit. Le Chapitre 2 est consacré au survol des divers concepts ayant trait à notre sujet, ainsi que des principales approches de routage relatives aux réseaux ad hoc mobiles. Le Chapitre 3 présente une description détaillée de notre protocole de routage et discute notamment les apports du dit protocole par rapport aux récentes études. Le Chapitre 4 traite quant à lui de tous les éléments liés à l'implémentation de notre protocole, ainsi que des expériences de simulations réalisées et finalement des résultats qui en découlent. Enfin, dans le Chapitre 5, nous concluons ce travail par une discussion sur les limites et apports de notre protocole et une ébauche de futures pistes de recherche.

CHAPITRE II

Routage dans les réseaux ad hoc

Les réseaux ad hoc mobiles ou MANETs (*Mobile ad hoc networks*) étendent la mobilité à toutes les composantes de l'environnement. Ce type de réseau ne nécessite en effet aucune infrastructure réseautique particulière, contrairement aux réseaux cellulaires où les stations de base constituent les relais nécessaires pour toute communication. Dans un réseau ad hoc, les nœuds s'auto organisent de façon autonome, jouant ainsi chacun le rôle de routeur afin d'acheminer les paquets de données.

Les réseaux ad hoc sont généralement modélisés par un graphe unitaire. Selon ce graphe, deux nœuds A et B du réseau sont voisins, et donc reliés par un arc, si la distance euclidienne entre leurs coordonnées physiques dans le réseau est au plus R , où R est le rayon de transmission d'un nœud. Chaque nœud est en mesure de communiquer directement avec tous les nœuds situés dans son rayon de transmission. Cependant, pour communiquer avec des nœuds hors de sa portée, un nœud doit utiliser des nœuds intermédiaires pour relayer les paquets de données saut par saut. La Figure 2.1 illustre la représentation en graphe unitaire d'un réseau ad hoc de 8 unités mobiles.

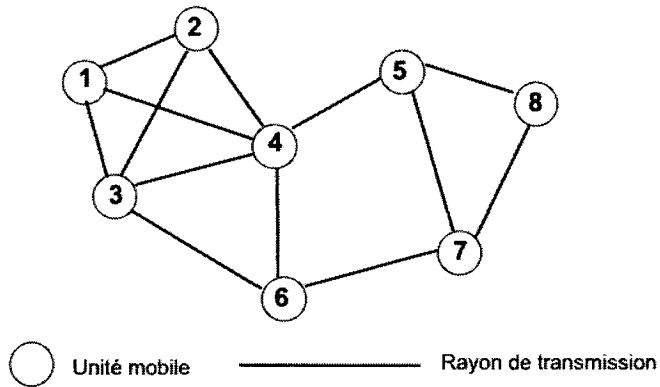


Figure 2.1 : Représentation en graphe unitaire d'un réseau ad hoc

Des variations de ce modèle existent, telles que les graphes unitaires prenant en compte les obstacles ou encore les graphes de puissance minimale, où chaque nœud est doté d'un rayon de transmission propre. Certes, ces graphes sont plus réalistes, mais il n'y a pas eu à ce jour dans la littérature de recherche crédible utilisant d'autres modèles que le graphe unitaire.

Les réseaux ad hoc possèdent les caractéristiques suivantes :

- Une topologie dynamique : les unités mobiles pouvant se déplacer arbitrairement, la topologie d'un réseau ad hoc est sujette à des changements fréquents et imprévisibles;
- Une absence totale d'infrastructures: les réseaux ad hoc, contrairement aux autres réseaux mobiles, ne reposent sur aucune infrastructure préexistante ou tout genre d'administration centralisée. La gestion du réseau est distribuée à travers les nœuds du réseau;
- Une bande passante limitée : le medium de communication sans fil est partagé et constitue donc une contrainte;
- Des contraintes d'énergie : les unités mobiles sont alimentées par des sources d'énergie autonomes et leur puissance énergétique est donc limitée. La consommation énergétique des terminaux doit être minimisée;
- Une sécurité physique limitée : Les réseaux mobiles sans fil sont en général plus vulnérables aux menaces sécuritaires que les réseaux filaires. La vulnérabilité physique des canaux et des nœuds, le manque d'infrastructure et les changements de topologie rendent la sécurisation des réseaux ad hoc d'autant plus difficile.

Le routage est le processus d'acheminement des informations à la bonne destination à travers un réseau. De part la nature dynamique des réseaux ad hoc, le routage dans ce type de réseaux est plus difficile et complexe que dans les réseaux fixes. De plus, les ressources limitées des terminaux mobiles font de l'efficacité du routage un élément d'autant plus critique de toute communication. Dépendamment des informations utilisées pour l'acheminement des paquets de données, les protocoles de routage peuvent être divisés en deux catégories :

- Les protocoles basés sur la topologie du réseau : les informations sur les liens du réseau sont utilisées,
- Les protocoles géographiques : seules les positions des nœuds sont prises en compte.

Dans ce chapitre, nous évoquerons dans un premier temps les approches de routage basées sur la topologie des réseaux ad hoc. Nous exposerons par la suite les différentes manières de prendre en compte la position géographique des unités mobiles dans de tels réseaux. Enfin, nous nous attarderons sur l'utilisation de la position géographique des nœuds dans le routage en étudiant le routage géographique.

2.1 ROUTAGE BASÉ SUR LA TOPOLOGIE DU RÉSEAU

Selon le mode de création et de maintenance de routes lors de l'acheminement des données, les protocoles de routage utilisant l'état des liens du réseau peuvent être classés en deux catégories :

- Les protocoles proactifs, où les routes sont établies à l'avance en se basant sur l'échange périodique des tables de routage;
- Les protocoles réactifs, où la recherche de route se fait à la demande.

De nombreux protocoles de routage basés sur la topologie du réseau existent dans la littérature. Nous nous contenterons cependant de présenter les protocoles qui nous semblent les plus intéressants.

2.1.1 Routage proactif

Dans les protocoles de routage proactifs, chaque nœud maintient des informations de routage vers chacun des autres nœuds dans le réseau. Ces protocoles sont dérivés des protocoles à état de lien et vecteur distant. Les informations de routage étant généralement stockées dans des tables, on parle aussi de protocoles orientés tables (*table-driven protocols*). Ces tables sont mises à jour périodiquement et / ou si la topologie du réseau change.

Le protocole DSDV

L'algorithme de routage DSDV (*Destination-sequenced distance vector*) [7] est basé sur l'algorithme de vecteur distant de Bellman-Ford (DBF), mais garantie l'absence de boucles. Chaque unité mobile maintient une table de routage contenant:

- Toutes les destinations possibles;
- Le nombre de nœuds ou sauts nécessaires pour atteindre chaque destination;
- Un numéro de séquence correspondant à un nœud destination. Ce dernier est utilisé pour distinguer les nouvelles routes des anciennes, évitant ainsi la formation de boucles de routage.

Lorsqu'un nœud désire transmettre un paquet à un nœud destinataire, il utilise sa table de routage pour obtenir un chemin vers la destination du paquet.

Pour assurer le rafraîchissement des tables de routage, chaque nœud diffuse régulièrement une mise à jour de sa table de routage à ses voisins. Ces mises à jour sont transmises sur une base périodique ou si des changements de topologie du réseau sont détectés (apparition, disparition d'un nœud, etc.). Pour réduire la bande passante consommée par les mises à jour, DSDV définit deux types de mise à jour, les mises à jour complètes et les mises à jour incrémentales. Lors d'une mise à jour complète, un nœud transmet la totalité de sa table de routage à ses voisins. Au contraire, dans une mise à jour incrémentale, seules les entrées qui ont subi un changement par rapport à la dernière mise à jour sont échangées. Ce type de mise à jour prévaut dans un réseau où la mobilité des nœuds est moindre. Lorsque les mouvements des nœuds mobiles deviennent plus fréquents, des mises à jour complètes sont alors utilisées.

Chaque paquet de mise à jour contient le nouveau numéro de séquence incrémenté du nœud émetteur et, pour chaque nouvelle route, l'adresse de la destination, le nombre de nœuds (ou de sauts) séparant le nœud de la destination et le numéro de séquence (des données reçues de la destination) tel qu'il a été estampillé par la destination. Lorsqu'un nœud reçoit un paquet de mise à jour, il procède effectivement à une mise à jour des routes de sa table dans deux cas : si les routes reçues dans le paquet ont un numéro de

séquence plus récent ou parce qu'elles ont une meilleure métrique que celles contenues dans sa table.

Lorsqu'un lien est rompu, toute route incluant le nœud concerné doit se faire assigner une métrique infinie ainsi qu'un nouveau numéro de séquence. Ainsi, lorsqu'un nœud reçoit une métrique infinie, et qu'il a un numéro de séquence moins récent avec une métrique finie, il propage lui aussi un paquet de mise à jour concernant le lien brisé.

Le protocole DSDV induit, de part sa nature proactive, une grande charge de trafic de contrôle liée à l'envoi périodique de paquets de mises à jour et cette charge croît avec le nombre de nœuds du réseau. Le principal avantage des protocoles de routage proactifs est la disponibilité immédiate des routes vers toutes les destinations pour tout nœud source. Les mises à jour périodiques et permanentes, ainsi que le stockage des informations les rendent toutefois peu appropriés pour les réseaux de grande dimension.

2.1.2 Routage réactif

Les protocoles de routage à la demande, ou réactifs, ont été proposés dans le but de réduire le trafic induit par les protocoles proactifs en maintenant les informations de routage uniquement pour les routes actives. Les routes ne sont ainsi déterminées et maintenues que pour les nœuds qui doivent envoyer des données à une destination particulière.

Lorsqu'un nœud doit transmettre un paquet de données, il diffuse un paquet de requête de route à travers le réseau. Ce processus constitue la phase de découverte de route. Lorsqu'un nœud intermédiaire possédant une route vers la destination, ou la destination elle-même, reçoit une requête de route, une réponse de route est renvoyée au nœud source. Le nœud source utilise ensuite cette information pour acheminer son paquet à la destination.

Le protocole AODV

Le protocole de routage AODV (*Ad hoc on-demand distance vector*) [3] est basé sur les protocoles DSDV et DSR [4]. AODV réduit le nombre de diffusions de paquets en créant les routes au besoin, contrairement à DSDV qui maintient la totalité des routes au

niveau de chaque nœud. AODV utilise aussi un mécanisme de découverte de route semblable à celui utilisé dans l'algorithme de routage DSR. Cependant, AODV s'appuie sur un établissement dynamique des entrées dans les tables de routage au niveau des nœuds intermédiaires.

Lorsqu'un nœud source a besoin de communiquer avec un autre nœud pour lequel il n'a pas d'informations dans sa table de routage, il diffuse un paquet de requête de route (RREQ) à travers le réseau. Supposons que le nœud S détienne des paquets destinés au nœud D . Le nœud S lance alors une découverte de route (Figure 2.2 (a)).

A la réception d'une RREQ, les nœuds intermédiaires conservent dans leur cache l'adresse du noeud qui leur a relayé la requête. Cette dernière constitue l'adresse du saut suivant pour la route vers la source initiale. Des pointeurs de retour vers la source sont ainsi établis dans leur table de routage. Chaque RREQ contient, outre l'adresse de la source, le numéro de séquence de la source, l'identificateur de diffusion de la requête et le numéro de séquence de la destination le plus récent connu de la source. Le numéro de séquence du nœud source est utilisé pour maintenir la fraîcheur de l'information pour le chemin inverse vers la source, tandis que le numéro de séquence du nœud destination spécifie le degré de fraîcheur de la route vers la destination nécessaire pour être acceptée par la source. Un nœud recevant une RREQ émet un paquet de réponse de route (RREP) si :

- Il est le nœud destination;
- Il possède une route vers la destination avec un numéro de séquence supérieur ou égal à celui spécifié dans la RREQ.

La RREP est alors envoyée vers la source. Sinon, le nœud diffuse la RREQ. Ainsi, sur la Figure 2.2 (b), les nœuds A , B et C diffusent à leur tour la RREQ reçue de S . Par ailleurs, les nœuds conservent les informations permettant d'identifier une RREQ, c'est-à-dire l'adresse de la source et l'identificateur de diffusion, pour éviter de diffuser une nouvelle fois une RREQ qu'ils ont déjà traitée. La Figure 2.2 (c) illustre les pointeurs inverses établis lors de la découverte de route.

Les RREPs sont propagées vers la source en utilisant les pointeurs inverses établis lors de la diffusion des RREQs correspondantes vers la destination, comme le montre la Figure 2.2 (d). Lorsque la source reçoit la première RREP, elle peut commencer à émettre des paquets de données vers la destination (Figure 2.2 (e)) en utilisant le chemin emprunté par la PREP. Cependant, si ultérieurement elle reçoit une autre RREP, elle mettra à jour son information de routage si la route concernée est meilleure.

Une route est maintenue active tant que des paquets de données transitent de la source à la destination selon ce chemin. Si un lien est rompu alors qu'une route est active, le nœud en amont sur le chemin diffuse un paquet d'erreur de route (RRER) vers la source pour l'aviser que la destination est désormais inaccessible par cette route. Après avoir reçu la RRER, le nœud source peut initier une nouvelle procédure de découverte de route pour communiquer avec la destination.

L'utilisation des numéros de séquence permet de prévenir la formation de boucles de routage. Un autre point fort du protocole AODV est sa capacité d'adaptation aux réseaux très dynamiques.

Si les protocoles de routage à la demande permettent d'éliminer le trafic de contrôle lié aux mises à jour permanentes des approches proactives, ils comportent des faiblesses. En effet, le routage réactif induit une certaine latence au niveau des nœuds liée à la construction des routes. D'autre part, les pannes sur les liens initialisant une nouvelle découverte de route introduisent un délai supplémentaire et consomment plus de largeur de bande lorsque la taille du réseau augmente.

2.2 PRISE EN COMPTE DE LA POSITION GÉOGRAPHIQUE DANS LE ROUTAGE

Dans cette section, nous définirons dans un premier temps quelques concepts liés à la prise en compte de la position géographique des unités mobiles dans le routage. Nous présenterons ensuite les différentes manières dont ces informations de localisation peuvent être utilisées dans l'acheminement des paquets dans les réseaux ad hoc.

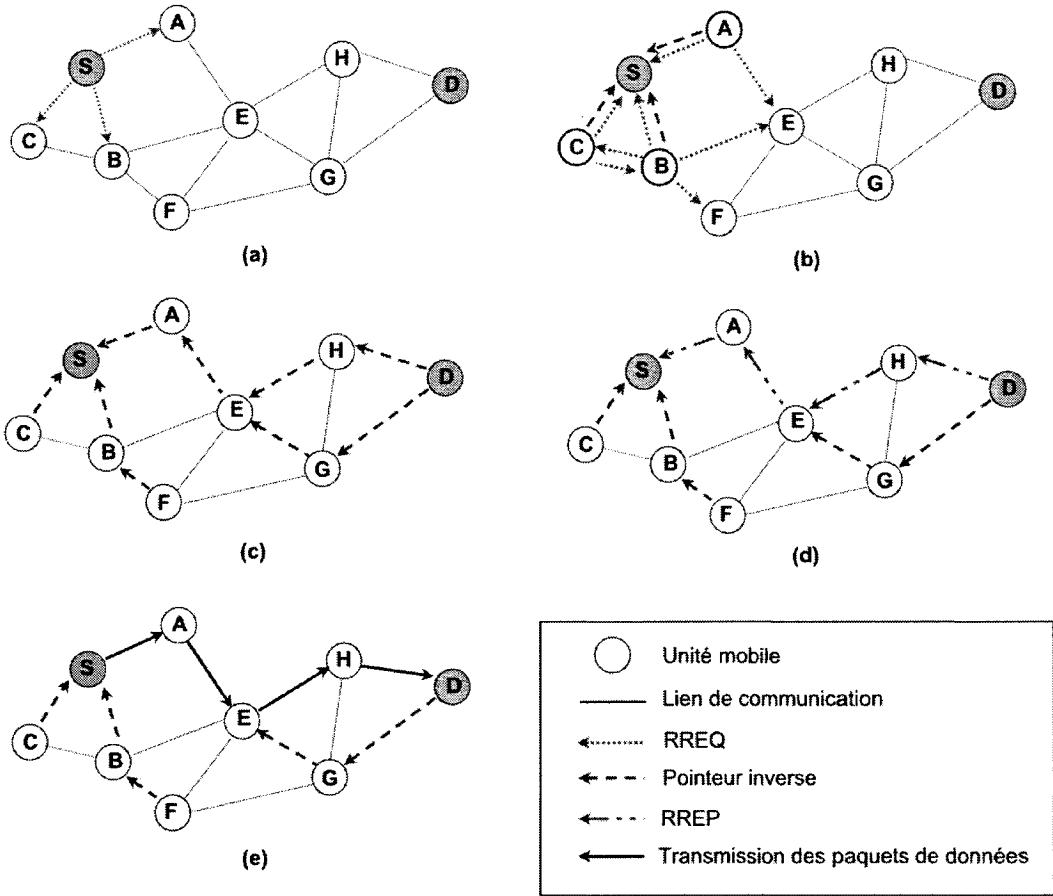


Figure 2.2 : Découverte de route dans AODV

2.2.1 Concepts de base

Diverses méthodes permettent d'évaluer la position géographique des nœuds dans un réseau sans fil.

➤ **Evaluer sa position ou celle d'un noeud voisin**

La distance entre des nœuds voisins peut être estimée sur la base de la puissance des signaux reçus. Les coordonnées relatives de nœuds voisins peuvent alors être obtenues en échangeant ces informations entre les nœuds [21]. D'autres méthodes de localisation peuvent aussi être utilisées, en fonction du temps d'arrivée (TOA, *Time Of Arrival*), de la différence de temps d'arrivée (TDOA) ou de la phase du signal reçu. La précision de

ces calculs dépend de la précision des mesures sur lesquelles ils sont basés. Les approximations obtenues sont donc souvent corrompues par des erreurs.

La position des nœuds peut aussi être obtenue directement en communiquant via un satellite en utilisant un système de communication global ou GPS, si les nœuds sont équipés de petits récepteurs GPS à faible consommation énergétique. Cette option se révèle être la plus efficace, la plus précise, mais aussi la plus simple par rapport aux calculs complexes nécessaires au niveau des unités mobiles du réseau avec les autres méthodes de localisation.

➤ **Evaluer la position d'un nœud spécifique**

Afin de déterminer la position courante d'un nœud spécifique, un service de localisation s'avère nécessaire. Les unités mobiles enregistrent constamment leur position courante auprès d'un tel service. Par la suite, lorsqu'un nœud ne connaît pas la position d'un éventuel partenaire de communication, il contacte le service de localisation pour l'obtenir.

Dans [14], les services de localisation sont classifiés en fonction d'une part, du nombre de nœuds hébergeant le service et d'autre part, du nombre de positions de nœuds qu'un serveur de localisation maintient. On parle alors de service de localisation “plusieurs-pour-plusieurs”, “plusieurs-pour-tous”, “tous-pour-plusieurs”, “tous-pour-tous”. Nous nommerons cette classification, classification de Mauve. Dans les réseaux cellulaires, les services de localisation sont des serveurs dédiés dont l'adresse est bien connue. On parle donc dans ce cas d'une approche “plusieurs-pour-tous” puisque les serveurs sont des nœuds spécifiques qui maintiennent les informations de position de tous les nœuds mobiles. Dans un réseau ad hoc mobile, une telle approche centralisée n'est pas viable [14]. Différents systèmes de service de localisation distribués ont donc été proposés dans la littérature.

Les services de localisation peuvent par ailleurs être classifiés en deux types : les services de localisation proactifs et réactifs [27]. Les services de localisation proactifs ne nécessitent pas d'échanges périodiques d'informations de localisation par les noeuds,

contrairement aux services de localisation réactifs. Nous nous intéresserons donc ici aux services de localisation proactifs. On distingue encore deux classes de services de localisation dans cette catégorie :

- les systèmes de bases de données de localisation, où des nœuds spécifiques servent de base de données de localisation pour d'autres nœuds. Selon la classification de Mauve, la plupart de ces systèmes sont classifiés *tous-pour-plusieurs*;
- les systèmes de dissémination de localisation, qui peuvent être considérés comme un cas spécial de système de bases de données de localisation où chaque nœud du réseau est un serveur de localisation pour tous les autres. Il s'agit donc d'une approche *tous-pour-tous*.

L'Annexe A présente brièvement les services de localisations proactifs les plus discutés.

2.2.2 Stratégies d'acheminement

Un nœud peut utiliser la position de ses nœuds voisins, ainsi que celle de la destination, pour déterminer le prochain nœud transmetteur d'un paquet de données. Différentes stratégies peuvent alors être appliquée pour effectuer ce choix: stratégies basées sur le progrès, la distance ou la direction.

➤ Stratégies basées sur le progrès

La notion de progrès est un concept clé pour plusieurs méthodes basées sur les GPS proposées entre 1984 et 1986. Etant donné un nœud source S , le progrès du nœud A est défini comme la projection de A sur la ligne reliant S et la destination finale.

Une stratégie intuitive consiste à transmettre le paquet au nœud qui réalise le plus grand progrès vers la destination. Un voisin est dans la direction de transmission si son progrès est positif. Ainsi sur la Figure 2.3, le prochain saut serait le nœud C . Cette stratégie est connue sous le nom de “Plus en Avant dans r ” (**MFR**, *most forward within r*) [6], où r est le rayon de transmission du nœud source.

Dans **NFP** (*nearest with forward progress*) [8], la méthode a été modifiée en proposant d'ajuster la puissance de transmission à la distance entre les deux nœuds. Le paquet est envoyé à celui des voisins qui est d'une part, plus proche de la destination que le nœud émetteur et d'autre part, le plus proche du nœud émetteur. Sur la Figure 2.3, ce nœud serait donc le nœud *A*. La transmission peut alors être accomplie avec une puissance minimale, réduisant ainsi les interférences avec les autres nœuds et augmentant par conséquent la probabilité d'une transmission réussie.

Finalement, dans la méthode de **progrès aléatoire** (*random progress method*) [7], le paquet est transmis aléatoirement, avec une probabilité égale, à un nœud intermédiaire qui a un progrès positif. Cette stratégie minimise la précision de l'information nécessaire sur la position des voisins et réduit le nombre d'opérations nécessaires pour la transmission des paquets.

➤ Stratégies basées sur la distance

En 1987, Finn [9] a proposé le principe **greedy**: il s'agit de sélectionner le nœud voisin le plus proche de la destination. Lorsque l'on atteint un nœud concave *C*, c'est-à-dire qu'aucun des voisins de *C* n'est plus proche de la destination que *C*, Finn propose de chercher parmi les voisins à *n* sauts en inondant le réseau jusqu'à ce qu'un nœud plus proche de la destination que *C* soit trouvé. L'algorithme ne garantit pas la livraison des paquets et n'optimise pas le taux de d'inondation. L'auteur affirme cependant que son algorithme ne crée pas de boucles, puisqu'il force le paquet à toujours se rapprocher de la destination pas à pas.

Une variante de l'algorithme greedy, appelée **GEDIR**, est proposée dans [10]. Dans la méthode **GEDIR originale**, chaque nœud intermédiaire transmet le paquet à son meilleur voisin (celui le plus proche de la destination). Le transfert s'arrête lorsque le meilleur voisin est le voisin d'où provient le paquet, soit au nœud dit concave.

Dans la méthode **GEDIR disjointe**, chaque nœud intermédiaire transmet le paquet à ses meilleurs voisins, c'est-à-dire ses voisins plus proches de la destination que lui-même, parmi ses voisins qui n'ont jamais reçu et transmis le même paquet auparavant.

Après avoir transmis le paquet, un nœud A devient inactif par rapport à ce paquet et rejette les futures copies de ce paquet. Tous les voisins de A qui ont reçu le paquet éliminent A de leur liste de voisins candidats pour la transmission des paquets. Un nœud cesse de transmettre le paquet (devient concave) s'il n'y a plus de voisins respectant la condition de transmission. Les boucles ne peuvent pas apparaître dans cet algorithme.

Ces méthodes réduisent le taux d'échec par rapport à la méthode greedy en mémorisant le trafic antérieur.

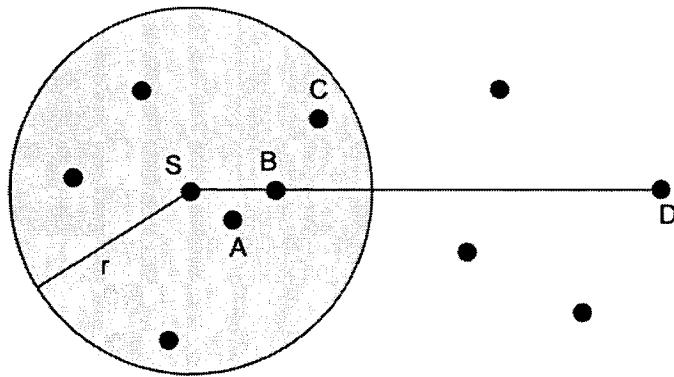


Figure 2.3 : Stratégies de routage greedy

Dans la méthode **greedy** à 2 sauts [10], le nœud A sélectionne le meilleur nœud candidat C parmi ses voisins à 1 saut et ses voisins à 2 sauts, selon le critère correspondant à l'algorithme à améliorer (distance ou progrès). A transmet ensuite le paquet à son meilleur voisin à 1 saut dans l'ensemble des voisins de A et C , soit le meilleur voisin lui permettant d'atteindre C . Ce principe basique peut être appliqué à la plupart des méthodes de routage présentées dans cette section.

➤ Stratégies basées sur la direction

Le **routage compas** (*compass routing*) [11], ou méthode **DIR**, constitue une autre stratégie d'acheminement qui sélectionne le voisin le plus proche de la ligne droite entre l'émetteur et le récepteur du paquet. Dans notre exemple, le nœud sélectionné serait donc B . Cependant, la méthode DIR, comme toutes les méthodes qui acheminent les

paquets au nœud voisin avec la direction la plus proche, telles que DREAM [1], peut occasionner la formation de boucles, comme le montre [10], utilisant le contre-exemple présenté sur la Figure 2.4.

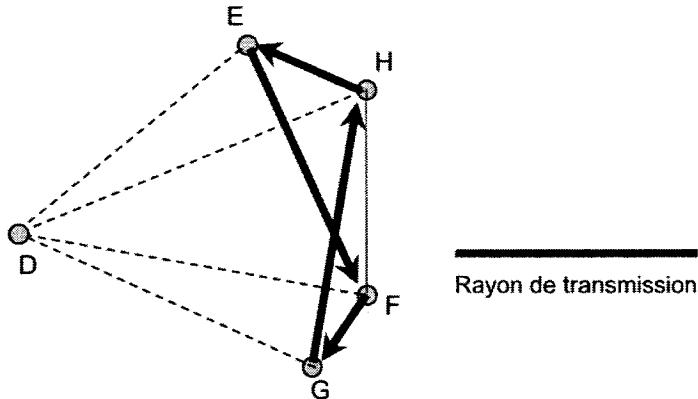


Figure 2.4 : Exemple de boucle dans le routage directionnel

La boucle consiste en quatre nœuds nommés E , F , G et H . Supposons que la source soit le nœud E . Le nœud E sélectionne le nœud F pour transmettre le paquet, car la direction de F est plus proche de la destination D que celle de son autre voisin H . De la même façon, F sélectionne G , puis le nœud G sélectionne H et H sélectionne E .

Les méthodes d’acheminement *greedy* et MFR fournissent, dans la plupart des cas, le même chemin vers la destination. Toutes les méthodes d’acheminement *greedy* ont des taux de livraison des paquets élevés pour les graphes denses, mais faibles pour les graphes clairsemés (environ la moitié des paquets pour des nœuds de degré moyen inférieur à quatre ne sont pas délivrés) [20]. Lorsque les paquets sont livrés, le nombre de sauts des méthodes *greedy* et MFR approche la performance de l’algorithme du plus court chemin.

Les stratégies d’acheminement *greedy* ne parviennent cependant pas toujours à trouver un chemin entre l’émetteur et la destination, même s’il en existe un. Le paquet est alors coincé au niveau du nœud transmetteur. La Figure 2.5 illustre ce problème. Le cercle autour du nœud destination D représente la distance entre le nœud source S et D , tandis que le cercle autour de S représente le rayon de transmission de S . Dans ce cas, il

existe bel et bien un chemin valide de S à D . Le problème est que S est plus proche de D que tous les autres nœuds qui sont dans son rayon de transmission. L'algorithme *greedy* a donc atteint un optimum local.

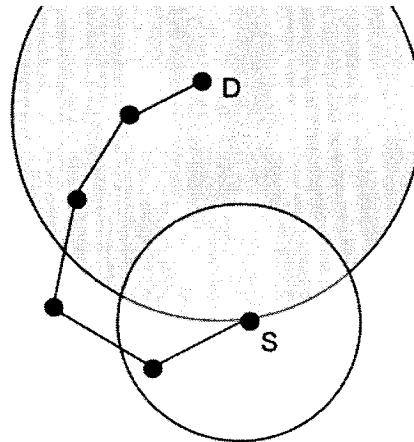


Figure 2.5 : Echec du routage greedy

Pour sortir de ce blocage, plusieurs approches ont été proposées. Dans [6], il est suggéré de transmettre le paquet au nœud avec le plus petit progrès en arrière (négatif), si aucun nœud n'est trouvé dans la direction avant. Cette solution peut cependant induire des boucles qui ne peuvent pas exister lorsque les paquets sont uniquement transmis vers la destination avec un progrès positif. Une autre solution consiste à abandonner les paquets qui arrivent à un optimum local [8].

2.3 ROUTAGE GÉOGRAPHIQUE

Les algorithmes de routage géographique éliminent certaines des limitations du routage basé sur la topologie du réseau. Ce type de routage utilise les positions physiques des nœuds du réseau. Chaque nœud détermine sa propre position via un récepteur de système de positionnement global GPS ou tout autre type de service de localisation [18]. Un nœud émetteur d'un paquet utilise par la suite un service de localisation pour déterminer la position de la destination et l'inclure dans l'en-tête du paquet. Les décisions de routage se font ainsi au niveau de chaque nœud et sont

uniquement basées sur la position de la destination contenue dans l'entête du paquet et les positions des nœuds voisins. De la sorte, contrairement au routage topologique, le routage géographique ne nécessite ni l'établissement, ni la maintenance de routes. On peut identifier trois stratégies principales dans le routage géographique [14] : l'acheminement “greedy”, l'acheminement dirigé et le routage hiérarchique. Les acheminements greedy et dirigé acheminent respectivement les paquets à un ou plusieurs voisins. Les algorithmes de routage hiérarchiques sont une combinaison des algorithmes de routage topologique et géographique. Le routage géographique est alors typiquement utilisé pour les grandes distances (lorsque les nœuds transmetteur et receveur sont éloignés), tandis qu'un algorithme de routage basé sur la topologie est utilisé à un niveau local. Dans toutes ces approches, chaque nœud du réseau diffuse périodiquement des paquets de mise à jour de sa position (*beacons* en anglais) à ses voisins. D'autres stratégies récemment proposées ont par ailleurs pour objectif de fournir un routage géographique sans *beacons*.

Dans cette section, nous présenterons les différentes approches de routage géographique évoquées ci-dessus.

2.3.1 Routage géographique greedy

Nous présentons ici GPSR et GRA, deux protocoles de routage populaires basés sur un acheminement greedy.

Le protocole GPSR

Le protocole **GPSR** (*Greedy Perimeter Stateless Routing*) [15], comme les autres protocoles présentés dans cette section, assume que chaque nœud est muni d'un système GPS. Deux méthodes sont utilisées pour l'acheminement des paquets de données :

- L'acheminement greedy, tel que nous l'avons présenté précédemment, qui est utilisé partout où cela est possible;
- L'acheminement par périmètre, qui est utilisé dans les régions où l'acheminement greedy ne s'applique pas, c'est-à-dire au niveau des nœuds concaves.

Chaque nœud du réseau maintient une table de ses voisins contenant les adresses et les positions de ces derniers. Les auteurs définissent la règle de la main droite utilisée dans l'acheminement par périmètre comme suit : lorsqu'un paquet arrive au nœud x (le nœud concave) par le nœud y , le prochain lien traversé est le prochain lien issu de x rencontré dans le sens contraire des aiguilles d'une montre à partir du lien (x,y) .

L'entête d'un paquet GPSR contient un champ indiquant si le paquet est en mode d'acheminement greedy ou périmètre, ainsi que la position géographique de la destination du paquet. Tous les paquets sont marqués greedy par le nœud émetteur. Dans le mode greedy, un nœud transmetteur cherche dans sa table de voisins, le voisin le plus proche de la destination. Si ce voisin est plus proche de la destination que le nœud transmetteur, alors celui-ci lui transmet le paquet. Sinon, le nœud transmetteur change le mode greedy du paquet en mode périmètre.

Lorsqu'un paquet entre en mode périmètre, GPSR enregistre dans le paquet la position L_p , soit la position où le paquet est entré en mode périmètre. En mode périmètre, l'acheminement du paquet suit la règle de la main droite, comme l'illustre la Figure 2.6. Lors de la réception d'un paquet en mode périmètre, GPSR compare la position L_p dans l'entête du paquet avec la position du nœud courant. GPSR remet le paquet en mode greedy dès que la distance séparant le nœud intermédiaire de la destination est inférieure à L_p . En effet, l'acheminement par périmètre n'est utilisé que pour sortir d'un optimum local de l'algorithme greedy.

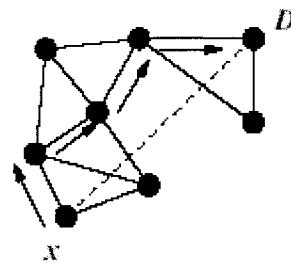


Figure 2.6 : Exemple d'acheminement par périmètre

Les auteurs ont comparé les performances de GPSR avec le protocole DSR [4] en supposant que les informations de localisation de la destination étaient précises. GPSR

délivre dès lors plus de 94% des paquets de données avec succès. Il entre donc en compétition avec DSR pour des réseaux de cinquante nœuds et devient plus efficace que DSR lorsque le nombre de nœuds augmente. Par ailleurs, le trafic de contrôle généré par GPSR est constant lorsque la mobilité augmente, tandis que DSR doit rechercher des routes plus longues avec un plus grand diamètre et ce d'autant plus souvent que la mobilité augmente (avec moins de cache effectif). DSR génère donc drastiquement plus de trafic de routage dans les simulations avec plus de cent nœuds. L'évolutivité et la garantie de livraison se révèlent ainsi être les avantages majeurs de GPSR sur les protocoles basés sur le routage de source.

Le protocole GRA

Jain, Puri et Sengupta [13] proposent une stratégie de routage géographique qui nécessite que les nœuds maintiennent de façon partielle les routes vers certaines destinations dans des tables de routage. GRA applique une stratégie d'acheminement greedy des paquets. Chaque nœud maintient une table de routage contenant une liste de nœuds ainsi que leurs positions géographiques.

Au départ, la table d'un nœud donné ne contient que les informations concernant les nœuds voisins de ce nœud. Lorsqu'un nœud a un paquet pour une destination, il choisit parmi les nœuds présents dans sa table, le nœud qui est le plus proche de la destination, et lui transmet le paquet. Le processus se répète jusqu'à ce que le paquet atteigne la destination.

Il peut arriver qu'un nœud S découvre qu'il est plus proche de la destination que chacun de ses voisins, l'acheminement *greedy* atteint là un optimum local. Le paquet est alors bloqué en S . Dans ce cas, le nœud S lance un processus de découverte de route. La découverte de route trouve un chemin de S à D et met à jour les tables de routage vers D au niveau de chaque nœud sur le chemin avec cette nouvelle information. La table des nœuds concernés contient alors des informations sur leurs nœuds voisins, mais aussi sur d'autres nœuds du réseau. Une fois que la découverte de route est complétée avec succès, le paquet peut être routé de S vers D .

Les auteurs proposent deux stratégies de découverte de route : *breadth first search* (qui est équivalente à l'inondation) et *depth first search* (DFS). DFS fournit un chemin acyclique de S à D . Chaque nœud met son identificateur et son adresse dans le paquet de découverte de route. Il transmet ensuite ce paquet à un voisin qui n'a pas encore reçu ce paquet auparavant. Ce voisin est celui de tous les voisins qui minimise $d(S,y) + d(y,D)$ où $d(x,y)$ est la distance euclidienne entre les nœuds x et y . Si un nœud n'a pas la possibilité de transmettre le paquet, il retire son identificateur et son adresse du paquet et retourne le paquet au nœud dont il l'a reçu précédemment. Les paquets de découverte de route sont maintenus pour un certain temps.

Le protocole GRA fournit un chemin unique vers chaque destination possible et garantie la livraison des paquets à la destination puisqu'il apporte une solution au problème des nœuds concaves de l'approche greedy. Il est donc très utile pour supporter le trafic sensible aux pertes.

2.3.2 Routage dirigé et multi-chemins (*directed forwarding*)

Dans un acheminement dirigé, le paquet est transmis à plusieurs voisins localisés dans la direction de la destination. On parle aussi d'inondation partielle, car celle-ci n'est dirigée que vers un secteur limité du réseau (comme dans les protocoles DREAM et LAR) ou parce qu'elle est arrêtée après un certain nombre de sauts (par exemple dans les familles d'algorithmes *flooding GEDIR*). Par ailleurs, l'inondation partielle peut être utilisée uniquement pour la découverte de route, comme dans LAR, ou pour l'acheminement des paquets, comme dans DREAM. Afin de contrôler l'inondation, ces méthodes nécessitent que les nœuds mémorisent le trafic précédent, ce pour éviter de transmettre le même paquet plus d'une fois.

Pour éviter l'abandon des paquets aux nœuds concaves, [10] propose une modification aux algorithmes greedy / GEDIR et MFR. En effet, les approches basées sur la distance, le progrès ou la direction mènent à un arrêt de l'algorithme aux nœuds concaves. Rappelons qu'un nœud concave est un nœud dont le meilleur voisin pour la transmission d'un paquet selon le critère d'acheminement spécifié est précisément le nœud

dont il a reçu le paquet. Il y a alors formation de boucles. La Figure 2.7 illustre un exemple de nœud concave dans l'algorithme GEDIR.

Le nœud S doit transmettre un paquet dont le destinataire est le nœud D . S détermine alors que le nœud B est son voisin le plus proche de D . S transmet donc le paquet à B . Il est à noter que dans l'algorithme GEDIR, le nœud B ne compare pas sa distance avec celles de ses voisins. Ainsi, même si le nœud B est plus proche de la destination que le nœud S , le paquet est quand même transmis au nœud S puisque celui-ci est le nœud le plus proche de D dans son rayon de transmission. Le nœud B constitue donc un nœud concave. Par la suite le nœud S retourne le paquet à B et une boucle locale entre les nœuds B et S est créée.

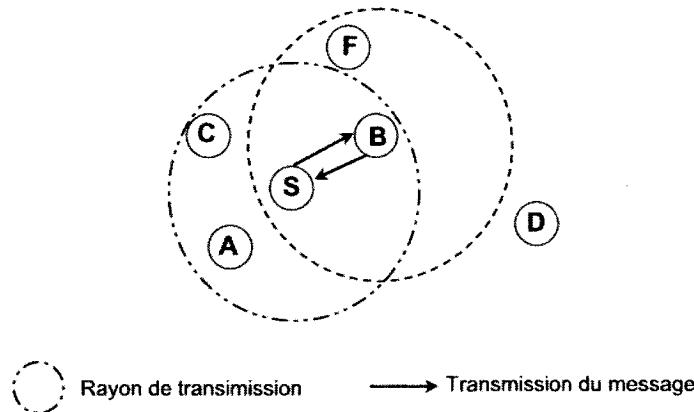


Figure 2.7 : Exemple de nœud concave dans GEDIR

Les auteurs proposent donc l'approche suivante : lorsque les algorithmes de base abandonneraient le paquet au nœud concave B , A diffuse le paquet à tous ses voisins. Le nœud A se retire ensuite du réseau pour les copies suivantes du paquet m , c'est-à-dire que ses voisins ne transmettront pas m à A dans les décisions futures. Cette méthode s'appelle *flooding greedy/GEDIR* et *flooding MFR* (sous l'abréviation **f-greedy**, **f-GEDIR**, **f-DIR** et **f-MFR**)[10].

En plus de garantir la livraison des paquets et un comportement sans boucle, les expériences réalisées dans [10] montrent que ces algorithmes réduisent le taux d'inondation par rapport aux protocoles DREAM et LAR. Pour les graphes denses, ces

méthodes approchent les performances de l'approche *greedy*. Pour les graphes clairsemés, elles induisent une inondation partielle.

Dans la méthode **multi-chemins** [12], le nœud source S transmet le paquet à ses c meilleurs voisins par rapport à la distance à D . Chacune des c copies créées par la suite suit les méthodes *greedy*, alternative ou disjointe. Ces méthodes sont ainsi appelées *c-greedy*, *c-alternate* et *c-disjoint*. Les expériences indiquent un gain significatif dans le taux de livraison des paquets pour $c=2$, un simple gain pour $c=3$ et pas de gain significatif pour $c>3$. Le taux d'inondation augmente avec c , et il semble donc que la seule valeur de c qui justifie l'utilisation cet algorithme soit 2.

Le protocole DREAM

DREAM (*Distance routing effect algorithm for mobility*) [1] est un protocole de routage proactif utilisant la position géographique des nœuds pour restreindre l'inondation de données à un sous-ensemble du réseau. Les nœuds échangent leurs coordonnées de façon périodique dans des paquets de contrôle. Ces coordonnées sont stockées dans une table de routage (appelée table de localisation). La fréquence à laquelle les paquets de contrôle sont transmis est déterminée par :

- L'effet de distance: plus la distance séparant deux nœuds est grande, plus ils semblent se déplacer lentement l'un par rapport à l'autre. Ainsi, si les nœuds sont éloignés géographiquement l'un de l'autre, leur position est mise à jour moins fréquemment que s'ils étaient proches ;
- Le taux de mobilité : plus un nœud se déplace rapidement, plus il doit communiquer sa position fréquemment.

Lorsqu'un nœud S veut envoyer un paquet m au nœud D , il transmet m à tous ses voisins à un saut dans la “direction de D ”. Pour obtenir cette direction, le nœud A détermine une région susceptible de contenir la destination D (zone supposée). Cette région représente la zone où devrait se situer le nœud D en prenant en compte sa mobilité depuis sa dernière position connue. Il s'agit en fait d'un cercle centré à la dernière position connue du nœud destination. Les nœuds intermédiaires ne

retransmettront alors le paquet que s'ils sont dans la “direction de D ”, soit s'ils sont situés dans la région définie par la ligne entre S et D et l'angle φ (voir Figure 2.8). Chaque nœud intermédiaire répète la même procédure, jusqu'à ce que D , si possible, soit atteint.

L'échange des informations de localisation permet de réduire considérablement la consommation de largeur de bande par rapport à l'échange complet d'informations d'états de lien ou de vecteurs de distance dans les protocoles de routage topologique. DREAM est donc mieux adapté aux changements d'échelles du réseau. D'autre part, le trafic de contrôle de routage est réduit, en rendant la fréquence à laquelle les paquets de mises à jour sont diffusés proportionnelle à la mobilité et à l'effet de distance. En d'autres termes, les nœuds stationnaires n'ont pas besoin d'envoyer de paquets de mises à jour.

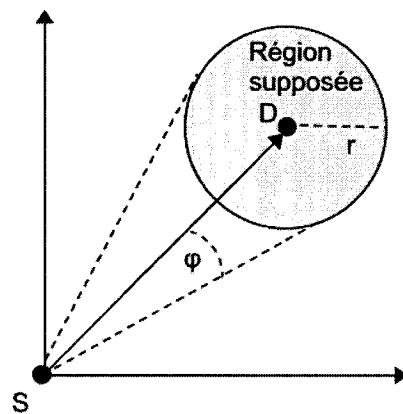


Figure 2.8 : Exemple de région

Le protocole LAR

Le but du protocole LAR (*Location-aided routing*) [5] est d'utiliser la position des nœuds pour améliorer la phase de découverte de route des approches de routage topologique réactif. L'inondation des paquets de requête de route est limitée à une certaine portion du réseau de façon similaire au protocole DREAM. Les auteurs proposent toutefois deux approches.

Dans la première approche, LAR1, deux types de zones sont définies : les zones supposées et les zones inspectées.

La zone supposée d'un nœud destination D , du point de vue d'un nœud source S au temps t_l , correspond à la région dans laquelle le nœud S s'attend à trouver le nœud D au temps t_l . La zone supposée du nœud destination D est en fait une région circulaire dont la position et la taille sont estimées par la source S en se basant sur :

- La position de la destination, telle que connue par la source ;
- L'instant qui correspond à cette position ;
- La vitesse moyenne du mouvement de la destination.

La zone inspectée du nœud D correspond quant à elle à la région dans laquelle est effectivement diffusée la requête de route vers le nœud D . Un nœud ne propage la requête de route que s'il appartient à cette zone de d'inspection. Afin d'augmenter la probabilité que la requête de route réussisse à atteindre le nœud D , la zone de requête devrait inclure la zone supposée du nœud D . Le nœud source S définit la zone d'inspection de D comme le plus petit rectangle incluant sa propre position et la zone supposée D (voir Figure 2.9). Le nœud S inclut ensuite les coordonnées de cette zone dans le paquet de requête de route transmis lors du lancement d'une découverte de route. Lors de la réception d'un paquet de requête de route, un nœud ne propage ce paquet que s'il est situé dans la zone de requête. Lorsque le nœud destination D reçoit un paquet de requête de route, il répond en envoyant un paquet de réponse de route vers la source. Cependant, dans le cas de LAR1, le nœud D inclut sa position courante et le temps courant dans le paquet de réponse de route.

Dans la deuxième approche, LAR2, le nœud S calcule la distance qui le sépare de la destination et l'inclut dans le paquet de requête de route en plus des coordonnées de la destination. Lorsqu'un nœud reçoit le paquet de requête de route, il calcule la distance qui le sépare de la destination et la compare avec la distance contenue dans le paquet reçu. Dans le cas où la distance calculée est inférieure ou égale à la distance reçue, le nœud transmet le paquet de requête de route reçu. Lors de l'envoi, le nœud met à jour le champ de distance avec sa propre distance qui le sépare du nœud destination.

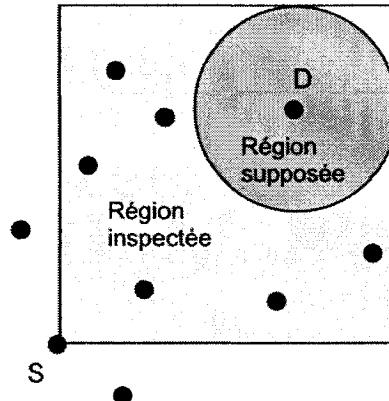


Figure 2.9 : Exemple de régions supposée et inspectée dans LAR

Les deux méthodes limitent le trafic de contrôle transmis à travers le réseau et donc préservent la bande passante. Elles déterminent aussi, dans la plupart des cas, le plus court chemin vers la destination. Cependant, la première approche aura tendance à réagir comme les protocoles d'inondation (comme DSR et AODV) dans un contexte à mobilité élevée.

Il est à noter que les protocoles DREAM et LAR, comme tous les protocoles utilisant la direction ou la distance dans les décisions d'acheminement, sont susceptibles de créer des boucles.

2.3.3 Routage hiérarchique

Les protocoles Terminode [16] et GRID [17] sont des protocoles de routage géographiques hiérarchiques dans lesquels le routage est structuré en niveaux. Ces protocoles appliquent différentes règles au routage longue distance et courte distance. Le routage géographique est utilisé pour le routage longue distance, tandis que lorsque le paquet arrive proche de la destination, un algorithme de routage topologique proactif est adopté. L'Annexe B décrit le fonctionnement de ces deux protocoles.

2.3.4 Routage géographique sans beacons

Dans la plupart des approches de routage géographique, les informations minimales qu'un nœud doit posséder pour prendre ses décisions d'acheminement sont sa propre position, la position de ses voisins et finalement la position de la destination (obtenue par l'intermédiaire d'un service de localisation). Les *beacons* sont des paquets de contrôle diffusés périodiquement par un nœud pour informer ses voisins de sa position. Évidemment, les échanges périodiques de beacons consomment une certaine proportion de la bande passante disponible, ainsi que la puissance énergétique des unités mobiles.

Dans [22], pour remédier à ce gaspillage de ressources, les auteurs proposent un algorithme de routage, BLR (*beacon less routing*), dont l'originalité est l'absence de beacons. Un nœud n'a donc en conséquence aucune information sur ses voisins. L'algorithme prend cependant soin que le paquet ne soit transmis qu'à un seul nœud voisin approprié.

Lorsqu'un nœud source a un paquet à transmettre, il inclut dans le paquet sa propre position, ainsi que celle de la destination du paquet, puis diffuse ce paquet à tous les nœuds dans son rayon de transmission.

Pour retransmettre le paquet, les nœuds intermédiaires doivent être situés dans l'arc de cercle de 60 degrés formé à partir de la position du nœud précédent vers la position de la destination et dans un rayon égal au rayon de transmission r commun à tous les nœuds du réseau (par exemple les nœuds A et B sur la Figure 2.10). Le choix de l'angle de 60 degrés doit permettre à chaque nœud situé dans ce secteur de percevoir les transmissions d'un autre nœud dans ce secteur.

A la réception d'un paquet, un nœud connaît la position du nœud précédent (le nœud qui a transmis le paquet), la position de la destination du paquet, ainsi que sa propre position. Ce nœud peut alors aisément déterminer s'il est situé dans le secteur de 60 degrés décrit ci-dessus.

Parmi tous les nœuds situés dans le secteur de 60 degrés, seul celui qui réalise le plus grand progrès, soit le nœud A sur la Figure 2.10, sera choisi via un processus de compétition. Chaque nœud évalue en effet un délai qui devra être introduit avant qu'il ne

transmette son paquet. Ce délai est inversement proportionnel au progrès réalisé par le nœud. Ainsi, le nœud réalisant le plus grand progrès vers la destination sera le premier à retransmettre le paquet. Par ailleurs, lorsque les autres nœuds situés dans le secteur de 60 degrés percevront cette transmission, ils annuleront leur propre transmission.

Le protocole BLR permet certes d'éliminer le trafic engendré par les beacons dans le routage géographique classique, mais le réseau doit avoir une densité de nœuds élevée pour garantir qu'il y ait au moins un nœud dans le secteur de 60 degrés.

Selon [23], les informations de localisation utilisées dans les approches de routage géographique ne sont souvent pas à jour à cause de la mobilité des nœuds. Les auteurs partent du constat que pour augmenter la précision de ces informations, il faudrait augmenter la fréquence de diffusion des beacons et donc encore augmenter leur consommation de bande passante. Les auteurs proposent donc un algorithme de routage géographique, CBF (*Contention-Based Forwarding*), dans lequel les décisions d'acheminement se font selon l'état courant réel du voisinage et non celui perçu par le nœud transmetteur. Tout comme BLR, cet algorithme se distingue des approches classiques de routage géographique par le fait que l'acheminement des paquets se fait sans l'aide de beacons et sans maintenance d'informations sur les nœuds voisins.

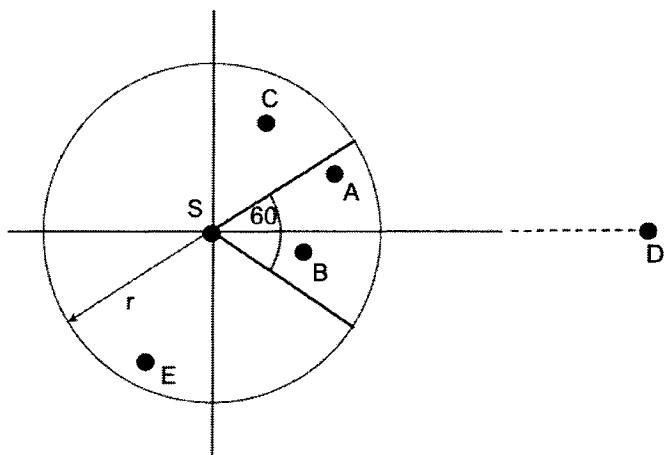


Figure 2.10 : Secteur de 60 degrés dans BLR

Lorsqu'un nœud source veut transmettre un paquet, il inclut dans un premier temps sa propre position ainsi que celle de la destination D dans l'entête du paquet puis le diffuse à tous ses voisins.

Un processus de concurrence basé sur le temps, similaire au protocole BLR, entre alors en jeu entre les nœuds voisins ayant reçu le paquet afin de déterminer quel nœud pourra retransmettre le paquet. Cependant, contrairement à BLR, tous les nœuds voisins du nœud source participent à ce processus de concurrence. Chaque nœud qui reçoit le paquet vérifie s'il est plus proche de la destination que le nœud qui lui a transmis le paquet. Une fonction de délai assure que le nœud réalisant le plus grand progrès sera le premier à retransmettre le paquet. Lorsque les autres nœuds perçoivent cette transmission, ils annulent leur propre transmission.

Il se peut toutefois que les nœuds ne soient pas en mesure de percevoir les transmissions d'autres nœuds si ceux-ci sont hors de leur rayon de transmission et donc que plusieurs paquets soient retransmis indépendamment. Pour remédier à ce problème, un processus de suppression est mis en place. Il s'agit de s'assurer que les nœuds transmetteurs potentiels soient dans le rayon de transmission l'un de l'autre, soit dans une zone que l'on appelle zone de suppression. Seuls les nœuds contenus dans cette zone de suppression peuvent être candidats pour retransmettre le paquet. Les autres nœuds ne sont pas considérés.

2.4 RÉCAPITULATIF

Nous avons, dans ce chapitre, effectué un survol des différentes stratégies de routage pour les réseaux ad hoc existantes dans la littérature et dégagé leurs forces et faiblesses. Deux types d'algorithmes de routage ont ainsi été étudiés: le routage topologique et le routage géographique.

Le routage topologique utilise des informations sur les liens existants dans le réseau pour acheminer les paquets. On distingue par ailleurs le routage proactif et le routage réactif, selon la manière dont les routes sont établies. Les algorithmes proactifs maintiennent des informations de routage sur les chemins disponibles et ce, même si ces

chemins ne sont pas couramment utilisés. Le principal inconvénient de cette approche est que la maintenance de ces chemins consomme une part significative de la bande passante lorsque la topologie du réseau change fréquemment. En réponse à cette observation, les algorithmes de routage réactifs maintiennent uniquement les routes actives. Ces algorithmes comportent toutefois certaines limitations. En effet, il est nécessaire d'exécuter une découverte de route avant toute nouvelle transmission et ce processus engendre un délai avant que la communication ne soit établie. Par ailleurs, bien que la maintenance de route ne soit ici limitée qu'aux routes actives, elle demeure susceptible de générer un trafic de contrôle important dans un contexte de mobilité élevée.

Le routage géographique, quant à lui, élimine certains inconvénients du routage basé sur la topologie en n'utilisant que la position des noeuds dans l'acheminement des paquets. Généralement, chaque nœud détermine sa position à l'aide d'un récepteur GPS. Un service de localisation permet à tout nœud émetteur d'un paquet de déterminer la position de la destination. Ce dernier inclut ensuite l'adresse ainsi obtenue dans l'en-tête du paquet. La décision de routage au niveau de chaque nœud est par la suite basée sur la position de la destination contenue dans l'en-tête du paquet de données et la position des noeuds voisins. Le routage géographique ne nécessite donc ni l'établissement, ni la maintenance de routes. La position des nœuds voisins est généralement communiquée via des diffusions sur 1 saut. Ces beacons sont envoyés périodiquement par tous les nœuds. Le trafic induit par le routage géographique est donc relativement faible et surtout indépendant de la mobilité des nœuds. L'évolutivité est par conséquent un avantage majeur de cette approche. Comme tout algorithme "glouton", le routage géographique est cependant susceptible d'atteindre un optimum local, lorsque aucun voisin n'est plus proche de la destination que le nœud courant, ou encore que le nœud plus proche de la destination est le nœud dont le paquet de données a été reçu (il y a alors formation de boucles). La stratégie de recouvrement la plus reconnue pour sortir de ce blocage est le routage par périmètre utilisée par GPSR. Si le mode périmètre s'avère

relativement efficace dans les réseaux denses, le nombre de sauts n'est pas garanti et les pertes de paquets liées à la formation de boucles peuvent s'avérer importantes.

Dans ce qui suit, nous proposons un nouvel algorithme de routage permettant de pallier à certaines des lacunes évoquées dans ce chapitre.

Chapitre III

Protocole de routage localisé proposé

Dans le chapitre précédent, nous avons présenté diverses approches de routage dans les réseaux ad hoc, ainsi que les problèmes inhérents à chacune d'entre elles. Nous allons maintenant décrire le principe et les mécanismes que nous proposons pour pallier à certaines de ces lacunes. Dans ce chapitre, nous introduisons donc l'algorithme de routage géographique que nous avons élaboré: PARA (*Proximity Aware Routing for Ad-hoc networks*). PARA choisit le prochain nœud transmetteur d'un paquet de données, non seulement en fonction de son voisinage direct, mais aussi en fonction de son voisinage à deux sauts. Nous introduisons ainsi le concept de “découverte de proximité”. L'étude du voisinage à deux sauts doit permettre d'anticiper la présence, mais surtout de minimiser le risque d'atteinte de “trous” dans le réseau, où l'avancée géographique du paquet vers la destination n'est plus possible. Nous présenterons dans un premier temps les motivations et les objectifs de PARA. Nous énoncerons par la suite les hypothèses préalables à la mise en œuvre de notre protocole avant de fournir une description détaillée de PARA.

3.1 MOTIVATIONS ET OBJECTIFS

Les algorithmes de routage réactif, tel qu'AODV, offrent de bonnes performances dans des réseaux ad hoc où la mobilité des nœuds est moindre, en réduisant notamment le trafic de contrôle par rapport aux approches proactive et en utilisant le plus court chemin. Toutefois, lorsque la mobilité des nœuds augmente, la charge sur le réseau croît rapidement, car les routes doivent être mises à jour beaucoup plus souvent, engendrant des diffusions de paquets à travers tout le réseau. Par ailleurs, le routage AODV de base utilise un mécanisme d'échange périodique de paquets *HELLO* pour maintenir la connectivité locale des nœuds, ce qui ajoute encore à la charge de contrôle.

Dans la plupart des algorithmes de routage géographique, chaque nœud du réseau ad hoc diffuse sa position à ses voisins directs sous la forme de paquets périodiques (*beacons* en anglais). Tout nœud du réseau stocke les informations contenues dans les *beacons* dans une table et mémorise donc constamment les positions géographiques de ses voisins. Un nœud détenant un paquet de données à transmettre sélectionne alors le prochain nœud transmetteur du paquet dans sa table de voisins selon sa propre position, celle de ses voisins et celle de la destination du paquet. Dans un acheminement *greedy*, le voisin qui minimise la distance restant à parcourir pour atteindre la destination est choisi pour la transmission du paquet. Si l'acheminement *greedy* apparaît comme la meilleure approche quant au choix du prochain nœud transmetteur, il comporte dans sa forme de base certaines lacunes liées au contexte des réseaux ad hoc.

En effet, l'acheminement *greedy*, dans sa forme de base, est susceptible d'atteindre un optimum local lorsqu'il n'y a plus de nœuds physiquement plus proche de la destination que le nœud détenteur du paquet. De telles situations se présentent notamment lorsque le réseau est partitionné par des obstacles. L'acheminement géographique vers la destination du paquet de données n'est alors plus possible. Le paquet est donc littéralement bloqué au niveau du nœud détenteur, qui est dit *concave* (voir Section 2.2). Entrent alors en jeu diverses techniques de recouvrement, destinées à “débloquer” le paquet de données du nœud concave. Le mécanisme de recouvrement le plus populaire est le routage par périmètre, utilisé par le protocole GPSR, qui est basé sur la traversée d'un graphe planaire. Cette approche comporte cependant deux inconvénients majeurs. D'une part, le nombre de sauts nécessaires pour retrouver un contexte où le routage géographique est possible peut s'avérer important. Par ailleurs, il a été montré dans [33] que le tiers des paquets perdus avec GPSR le sont à cause de boucles de routage dans le mode périmètre.

PARA se veut un compromis entre les protocoles AODV et GPSR, exploitant uniquement les forces de chaque approche. Nos objectifs sont les suivants :

- Proposer une nouvelle stratégie d'acheminement des paquets basée sur la position des nœuds permettant d'anticiper les risques de blocage au niveau de nœuds concaves.
- Garantir l'absence de boucles de routage, susceptibles de déclencher l'abandon de paquets de données et engendrant un certain gaspillage de ressources du réseau en terme de délai, d'énergie et de bande passante.
- Réduire la perte de paquets de données dans un contexte dispersé et hautement mobile.

3.2 HYPOTHÈSES

Dans le cadre de la mise en œuvre de PARA, nous poserons les hypothèses suivantes :

- Nous considérerons que chacune des unités mobiles du réseau ad hoc est équipée d'un récepteur GPS lui permettant d'évaluer ses coordonnées géographiques à tout instant t .
- Nous négligerons les erreurs liées à la précision des positions fournies par les récepteurs GPS pour des raisons évidentes de simplicité.
- Nous assumerons la présence dans le réseau ad hoc d'un service de localisation distribué, tel que GLS [30] ou GRSS [32] (voir Annexe A). Chaque unité mobile ayant besoin d'établir une communication avec un nœud donné du réseau sera donc en mesure d'obtenir la position géographique de ce nœud par le biais de ce service.
- Afin de simplifier la mise en œuvre de notre approche, les liens de transmission entre nœuds voisins du réseau ad hoc seront considérés symétriques. Par ailleurs, nous supposerons que les nœuds du réseau ont le même rayon de transmission R .

3.3 PRINCIPE

Nous cherchons à proposer une nouvelle stratégie d'acheminement géographique permettant d'anticiper l'atteinte d'un optimum local, tout en supprimant les boucles de

routages engendrées par GPSR en adoptant un routage topologique là où le routage géographique échoue.

PARA repose ainsi sur quelques principes majeurs, dont la suppression du mécanisme d'échanges périodiques de *beacons* et *HELLOs* utilisé dans les approches de routage évoquées précédemment. De fait, chaque unité mobile du réseau n'a donc au départ aucune information sur son voisinage.

Le second principe concerne l'établissement des communications : lorsqu'une unité mobile S doit communiquer avec une autre unité du réseau, elle initie une *découverte de proximité* en diffusant des paquets, que nous appellerons des requêtes de proximité. La découverte de proximité doit permettre au nœud S d'obtenir des informations d'une part, sur ses voisins directs et d'autre part, sur le voisinage de ses voisins directs, soit son voisinage à deux sauts. En effet, la stratégie d'acheminement déployée par PARA consiste à choisir le prochain nœud transmetteur, non seulement en fonction de sa distance à la destination du paquet, mais aussi en fonction de celle de ses voisins à la destination. Nous partons du constat qu'à priori, plus un noeud a de voisins physiquement plus proches de la destination que lui-même, moins il y a de chances que ce nœud soit concave. Ainsi, plutôt que de transmettre le paquet au nœud voisin minimisant la distance restante à parcourir pour atteindre la destination, nous le transmettons au voisin qui :

- Permet de faire moindrement progresser géographiquement le paquet vers la destination;
- Possède lui-même le plus de voisins physiquement plus proches de la destination que lui-même.

Cette approche doit nous permettre d'anticiper, et donc de minimiser le risque d'atteindre un noeud concave afin d'exploiter autant que possible le routage géographique.

Supposons un réseau ad hoc tel qu'illustré sur la Figure 3.1. Le nœud S génère un paquet de données destiné au nœud D . Dans un algorithme géographique classique, le paquet serait envoyé au nœud voisin $V1$ qui minimise la distance à parcourir pour

atteindre D . Cependant, ce nœud se révèlerait par la suite concave et il faudrait alors entreprendre une procédure de recouvrement pour atteindre D . Selon la stratégie d'acheminement de PARA, c'est le nœud $V2$ qui est choisi comme prochain nœud transmetteur du paquet. Permettant une avancée géographique vers la destination moins importante que le nœud $V1$, le nœud $V2$ permet néanmoins d'atteindre la destination sans procédure de recouvrement.

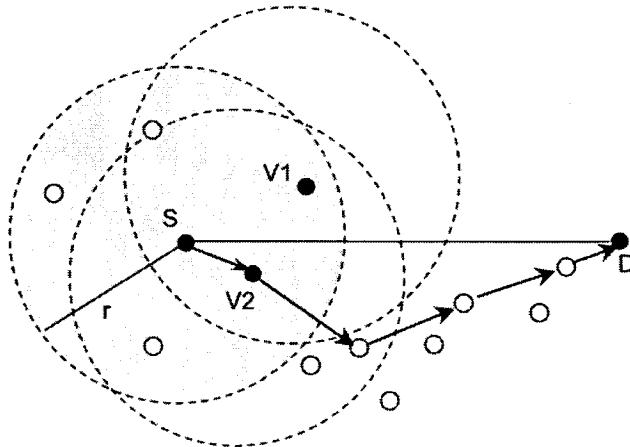


Figure 3.1 : Acheminement d'un paquet avec PARA

Nous proposons aussi un mécanisme de recouvrement permettant de gérer l'atteinte, ou encore l'anticipation, de noeuds concaves lors de la transmission d'un paquet de données. En effet, le mode périmètre, utilisé par GPSR pour retrouver un contexte où l'acheminement *greedy* est possible après un nœud concave, est fortement sujet à la formation de boucles. Or, ces boucles sont susceptibles d'entraîner un nombre important d'abandon de paquets et de rallonger considérablement le nombre de sauts effectués par les paquets.

Enfin, lorsqu'un nœud concave est atteint et que la procédure de recouvrement de PARA n'est pas efficace, les paquets de données sont acheminés avec un routage topologique, AODV, évitant les boucles de routages engendrées par GPSR et utilisant le plus court chemin pour atteindre la destination.

Avant d'approfondir les mécanismes mis en jeu par PARA, nous commencerons par définir les concepts sur lesquels ce nouvel algorithme repose. Par la suite, nous présenterons les différentes structures de stockage mises en œuvre par PARA, après quoi, nous développerons en détail l'algorithme de routage.

3.3.1 Concepts de base

Un nœud sera dit *actif* s'il génère, ou du moins transmet, des paquets de données. Nous nommerons *destinations actives* d'un nœud, l'ensemble des nœuds destinataires des paquets de données détenus par ce nœud. Le nombre de destination actives du nœud N sera désigné par $NDst(N)$.

Par ailleurs, nous noterons $DIST(I, D)$, la distance euclidienne du nœud I au nœud D .

Soient le nœud D , la destination d'un paquet de données p , et le nœud N , le nœud détenteur du paquet p . Un nœud V , voisin de N et vérifiant $DIST(V, D) < DIST(N, D)$, est appelé *meilleur voisin* de N pour la destination D et noté $MV_{N, D}(V)$. L'ensemble des meilleurs voisins du nœud N pour la destination D est désigné par $MV_{N, D}$. En outre, un nœud I , meilleur voisin du nœud $MV_{N, D}(V)$ pour la destination D , est quant à lui noté $MV_{N, D}(V, I)$. $MV_{N, D}(V, I)$ constitue donc un meilleur voisin à 2 sauts de N pour la destination D via son meilleur voisin direct $MV_{N, D}(V)$.

La notation $MV_N(J)$ désigne un nœud J , meilleur voisin de N pour au moins l'une de ses destinations actives, peu importe cette destination. Nous nommerons l'ensemble des destinations pour lesquelles $MV_N(J)$ est un meilleur voisin de N , *meilleures destinations* de $MV_N(J)$, tandis que le nombre de ces destinations est $N_MDst(MV_N(J))$. L'ensemble constitué de tels meilleurs voisins de N est noté MV_N .

L'indice d'accessibilité, noté $IsDest(MV_{N, D}(V), D)$, évalue, comme son nom l'indique, l'accessibilité de la destination D via le meilleur voisin $MV_{N, D}(V)$. Ce paramètre peut prendre trois valeurs:

- $Isdest(MV_{N, D}(V), D) = 2$, le meilleur voisin $MV_{N, D}(V)$ est en fait la destination D elle-même: $V = D$. La destination est donc située dans le voisinage direct du nœud détenteur des paquets, N .

- $IsDest(MV_{N, D}(V), D) = 1$, la destination D est alors dans le voisinage direct du meilleur voisin $MV_{N, D}(V)$. Le nœud N peut alors acheminer les paquets à D en deux sauts via $MV_{N, D}(V)$.
- $IsDest(MV_{N, D}(V), D) = 0$, la destination D n'a pas été détectée dans le voisinage direct de $MV_{N, D}(V)$. Le meilleur voisin $MV_{N, D}(V)$ permet donc au nœud N de rapprocher, géographiquement parlant, les paquets de données de la destination, sans toutefois l'atteindre.

Il faut ajouter à la distance à la destination D et à l'indice d'accessibilité, un autre paramètre d'importance utilisé par PARA: l'*indice de proximité*. L'indice de proximité du nœud N pour une destination donnée D , noté $IProx(N, D)$, est défini comme étant le nombre de voisins directs de ce nœud qui sont physiquement plus proches de D que N , soit le nombre de ses meilleurs voisins pour D . De fait, ce paramètre reflète les chances de progrès géographique via le nœud N d'un paquet de données vers son nœud destinataire D . Plus le nœud N a de meilleurs voisins, plus le risque d'atteindre un nœud concave est à priori réduit. En effet, le choix de nœuds permettant de faire “avancer” le paquet vers la destination est alors plus grand.

Si, par contre, le nœud N a un indice de proximité nul, c'est qu'il n'y a aucun nœud dans son rayon de transmission susceptible de faire progresser géographiquement un paquet de données vers D . Le nœud N est alors dit *concave*. Autrement dit, N ne devrait pas être choisi pour transmettre le paquet. Un nœud sera dit *concave par anticipation* si chacun de ses meilleurs voisins est concave. La Figure 3.2 illustre un exemple de nœud concave par anticipation.

Le nœud M détient un paquet destiné au nœud D . Les nœuds $V1$ et $V2$ sont géographiquement plus proches de D que M . Ils constituent donc les meilleurs voisins de M la destination D . Aucun de ces nœuds ne possède cependant de voisins susceptibles de faire progresser physiquement le paquet vers D . Ces nœuds sont donc concaves et, par conséquent, le nœud M est concave par anticipation.

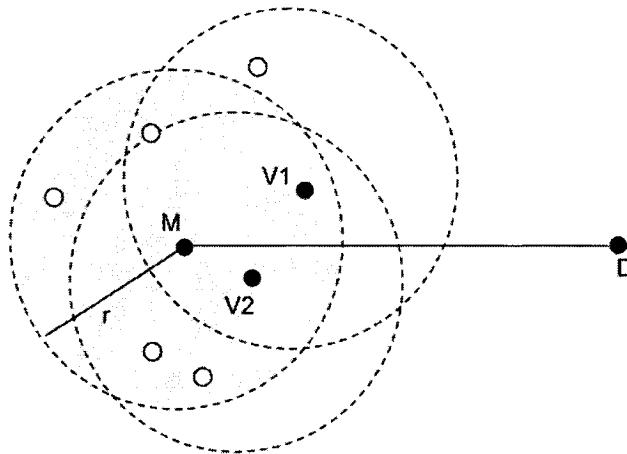


Figure 3.2 : Nœud concave par anticipation

3.3.2 Stockage des informations

Différents types de stockage, tous temporaires, sont déployés par PARA. Dépendamment de leur état, actif ou inactif participant à une découverte de proximité, les nœuds du réseau utilisent une ou plusieurs des structures de stockages suivantes : une table de meilleurs voisins, un cache de découverte, un cache de nœuds concaves, un cache réponses de proximité et un cache de requêtes de proximité. L'Annexe C décrit ces structures de stockage de manière détaillée.

3.3.3 Initialisation de la phase de transmission

Un nœud N initie une phase de transmission avec un nœud D s'il débute une communication en générant un premier paquet de données destiné à D ou encore s'il reçoit pour la première fois d'un autre nœud un paquet de données adressé à D .

Le nœud N devient alors actif. Il est cependant incapable de transmettre le paquet de données, car il ne dispose d'aucune information sur les noeuds qui l'entourent. En effet, dans les deux cas évoqués précédemment, la table de meilleurs voisins du nœud N pour la destination D est vide. Celle-ci sera peuplée lors de la découverte de proximité initiale pour la destination D .

Les paquets de données générés ou reçus entre-temps pour D seront stockés dans un tampon jusqu'à l'issu de cette découverte, après quoi ils seront envoyés au prochain nœud transmetteur.

Le routage PARA nécessite l'ajout d'une entête spécifique PARA aux paquets de données. Le format de cette entête est illustré par le Tableau annexé F.1. Chaque fois qu'il génère un paquet de données, un nœud N détermine, dans un premier temps, la position de la destination D des paquets de données via le service de localisation du réseau. Ce service permet en effet d'obtenir dynamiquement la position géographique d'un nœud à partir de son adresse. Tout paquet de données généré voit aussi le champ *data_mode* de son entête PARA initialisé à *géographique*, soit le mode d'acheminement normal de PARA.

Une fois le paquet de données inséré dans le tampon, le nœud N initie une découverte de proximité, dite *initiale*, car N débute une phase de transmission pour la destination D . A l'issu de cette découverte de proximité initiale, le nœud N aura en sa possession les informations qui lui permettront de déterminer le prochain nœud transmetteur du paquet de données. Parallèlement, le nœud N mémorise les découvertes initiales en cours dans son cache de découvertes, au cas où d'autres paquets seraient générés ou reçus pour les destinations concernées. Ces paquets sont eux aussi insérés dans le tampon en attendant la fin de la découverte initiale correspondante.

Soient :

- N le nœud initiateur de la transmission;
- p un paquet de données;
- S la source et D la destination du paquet p ;
- $Nsauts(p)$ le nombre de sauts effectués jusque là par le paquet p ;

Le comportement d'un nœud N lorsqu'il initie une transmission avec D est défini par l'algorithme présenté par la Figure 3.3.

- Obtenir la position de D
- Si $S = N$ et $Nsauts = 0$, le nœud N génère le paquet p , alors
 - Ajouter l'entête PARA au paquet p
- Si N n'est pas actif, alors
 - Changer l'état de N à actif dans le cache de découvertes
 - Lancer le déclencheur pour les découvertes de proximité périodiques
- Si N n'a pas encore enregistré D dans son cache de découverte, alors
 - Enregistrer D et ses coordonnées dans le cache de découverte
- Sinon
 - Mettre à jour les coordonnées de D dans le cache découvertes
- Si N n'a pas de découverte de proximité initiale pour D en cours, alors
 - Si $MV_{N,D} = \emptyset$, alors
 - Insérer p dans le tampon de paquets
 - Lancer une découverte de proximité initiale pour D
 - Démarrer le compteur pour le délai de la découverte de proximité initiale, $discv_delay$
 - Sinon
 - Choisir le prochain nœud transmetteur pour D
 - Transmettre le paquet p au prochain nœud transmetteur
- Sinon
 - Insérer p dans le tampon de paquets

Figure 3.3 : Algorithme décrivant le comportement d'un noeud initiant une communication

3.3.4 Déroulement d'une découverte de proximité

La découverte de proximité est le processus permettant à un nœud N de découvrir son voisinage direct, ainsi que son voisinage à deux sauts. Comme nous l'avons déjà évoqué, PARA adopte une approche dite réactive, dans la mesure où il n'y a aucun trafic de contrôle à moins qu'une communication ne soit nécessaire entre deux nœuds du réseau ad hoc. Chaque nœud du réseau n'a donc à priori aucune connaissance sur les

nœuds qui l'entourent. Ainsi, tout nœud débutant une transmission doit initier une découverte de proximité avant de choisir le prochain nœud transmetteur des paquets de données qu'il détient. Par la suite, des découvertes de proximité périodiques lui permettront de maintenir sa table de meilleurs voisins à jour.

Globalement, à l'issu d'une découverte de proximité, le nœud N :

- Possède une table de meilleurs voisins à jour;
- Est en mesure d'évaluer son propre indice de proximité;
- Est apte à choisir le prochain nœud transmetteur d'un paquet de données.

Une découverte de proximité peut être :

- Initiale: elle constitue alors un événement discret qui est déclenché dès qu'une première transmission pour une destination spécifique est initiée.
- Périodique: lorsqu'un nœud passe de l'état inactif à l'état actif, il démarre un déclencheur qui assurera le lancement périodique de découvertes de proximité. Ces découvertes de proximité périodiques, dont la période est fixée par le paramètre *preq_int*, permettent à chaque nœud actif du réseau de mettre à jour sa table de meilleurs voisins pour chacune de ses destinations actives.

Par ailleurs, les échanges réalisés dans le cadre de toute découverte de proximité permettent de dégager deux phases logiques, mettant en jeu deux couples d'acteurs:

- Durant la phase 1, ce sont le nœud initiateur de la découverte et ses voisins directs qui communiquent. Les paquets de contrôle échangés sont alors des requêtes et des réponses de proximité de niveau 1, que nous noterons respectivement PREQ1 et PREP1.
- La phase 2 constitue une sous-étape de la phase 1 et implique quant à elle les voisins directs du nœud initiateur et ses voisins à 2 sauts. Des requêtes et des réponses de proximité de niveau 2, notées respectivement PREQ2 et PREP2, sont ici transmises.

Il convient de le spécifier, et nous l'expliciterons ultérieurement dans cette section, les découvertes de proximités sont sujettes à des contraintes temporelles. Plus particulièrement, la durée globale d'une découverte de proximité est initialement fixée

au paramètre *discv_delay*, tandis que la durée de la phase 2 est régie par le paramètre *reply_delay*.

Bien que leur objectif et leur déroulement global soient le même, nous distinguerons deux types de découvertes de proximité selon le nombre de destinations concernées :

- Les découvertes de proximité à destination unique, qu'il s'agisse d'une découverte initiale ou d'une découverte périodique pour un nœud n'ayant qu'une destination active.
- Les découvertes de proximité périodiques à destinations multiples.

Nous commencerons notre étude du déroulement des découvertes de proximité par les découvertes destination unique, avant de décrire les découvertes de proximité périodiques à destinations multiples.

3.3.4.1 Peuplement de la table de meilleurs voisins pour une destination spécifique et découverte de proximité à destination unique

Une découverte de proximité initiée par un nœud N est à destination D unique si:

- Le nœud N initie une transmission avec D , que ce soit parce qu'il génère ou qu'il reçoit pour la première fois un paquet de données pour D . Le nœud N lance alors une découverte de proximité initiale pour D . La destination D sera dite *nouvelle*, dans la mesure où le nœud N ne dispose encore d'aucune information concernant son voisinage pour cette destination.
- Le nœud N a déjà transmis des paquets de données à la destination D . Il lance une découverte périodique de mise à jour de sa table de meilleurs voisins et ne possède qu'une seule destination active, D .

Le déroulement d'une découverte de proximité à destination unique est le même, peu importe que la destination D soit une destination active ou une nouvelle destination, et se décline en plusieurs étapes.

❖ Étape 1 de la découverte: Diffusion d'une PREQ1 par le nœud initiateur

Pour initier une découverte de proximité, le nœud N diffuse une requête de proximité de niveau 1, ou PREQ1, à ses voisins directs (voir Figure 3.4 (a)). Cette PREQ1 contient l'adresse de l'unique destination D , ses coordonnées et la distance du nœud initiateur N à D .

Cependant, dans un contexte général de découverte de proximité, l'entête de la PREQ1 contiendra ces mêmes informations pour chaque destination active du nœud initiateur, ainsi que le nombre de ses destinations actives, comme le montre le Tableau annexé F.2. Ici, le nombre de destinations actives inséré dans l'entête de la PREQ1 est 1.

❖ Étape 2 de la découverte: Réception de la PREQ1 par les voisins directs

A la réception d'une PREQ1, chaque voisin direct de N , V , enregistre dans un premier temps la PREQ1 dans son cache de réponses de proximité. Il enregistre notamment :

- Le nœud initiateur de la découverte, soit N ;
- Le nombre de destinations pour cette découverte, ici 1;
- La destination correspondant à la découverte, D .

Chaque nœud V vérifie ensuite s'il est la destination D de cette découverte. Si c'est le cas, le nœud va insérer de nouvelles informations dans son cache et exécuter directement l'étape 7 de la découverte de proximité. Cette étape consiste en l'envoi d'une réponse de proximité de niveau 1, ou PREP1. Dans le cas présent, on parlera alors de PREP1 *anticipée*. Entre temps, les informations nouvellement insérées par le nœud V concerné sont :

- L'adresse de la destination (même s'il s'agit de sa propre adresse) ;
- Son indice d'accessibilité, dont la valeur est 2 puisque V est la destination;
- Son indice de proximité, qui vaut 1;
- Sa distance à D , qui est nulle.

Si le nœud V n'est pas la destination D , il récupère ses propres coordonnées et évalue sa distance à la destination D à partir des informations contenues dans l'entête de la PREQ1 :

- Si $DIST(V, D) < DIST(N, D)$, alors le nœud V est un meilleur voisin de N pour D et sera désormais, en tant que tel, noté $MV_{N,D}(V)$. Il s'avère intéressant d'investir le voisinage de ce nœud puisqu'il permet déjà une avancée géographique vers D . Pour sa part, D constitue une meilleure destination pour le nœud $MV_{N,D}(V)$. Celui-ci va donc initialiser ses caractéristiques pour la destination D dans son cache de PREP1s.

Le nœud $MV_{N,D}(V)$ ajoute ainsi dans son cache de réponse les champs suivants :

- L'adresse de D ;
- Son indice d'accessibilité initialisé, $IsDest(MV_{N,D}(V), D) = 0$;
- Son indice de proximité initialisé, $IProx(MV_{N,D}(V), D) = 0$;
- Sa distance à D , $DIST(MV_{N,D}(V), D)$.

Chaque meilleur voisin $MV_{N,D}(V)$ passe ensuite à l'exécution de la phase 2.

- Sinon, le nœud V ne constitue pas un meilleur voisin de N pour la destination D . Dans ce cas, il ne fait rien de plus.

Les étapes 3, 4, 5 et 6 de la découverte de proximité vont permettre d'évaluer les possibilités d'avancée géographique vers la destination à deux sauts de N et constituent une deuxième phase logique de la découverte de proximité, la phase 2. Seuls les meilleurs voisins du nœud N exécutent cette phase, car les autres voisins, ne permettant pas le moindre progrès géographique vers la destination, ne sont pas considérés dans le choix du prochain nœud transmetteur. Sur la Figure 3.4 (b) : il s'agit des nœuds A , B et E . A l'issu de ces étapes, chaque meilleur voisin $MV_{N,D}(V)$ de N connaîtra ses attributs pour sa meilleure destination D .

- ❖ Étape 3 de la découverte: Diffusion d'une PREQ2_ONE par chaque meilleur voisin
A leur tour, les meilleurs voisins $MV_{S,D}$ diffusent chacun une requête de proximité, de niveau 2 cette fois-ci (Figure 3.4 (b)). Traitant ici d'une seule destination active, la requête de proximité diffusée est une PREQ2_ONE. Elle contient l'adresse et les

coordonnées de la destination D , ainsi que l'adresse du noeud initiateur de la découverte, N . Le Tableau annexé F.3 illustre le format de l'entête des PREQ2_ONEs. L'usage des PREQ2s, qui jouent le même rôle et interviennent au même moment dans la découverte de proximité que les PREQ2_ONEs, sera décrit dans l'Annexe D qui présente le déroulement des découvertes de proximité à destinations multiples.

Chaque PREQ2_ONE, tout comme chaque PREQ2, est associée à la PREQ1 reçue auparavant. Une découverte de proximité doit en effet être identifiée de façon unique. Cette unicité est garantie par la combinaison des deux champs suivants dans l'entête des PREQ2_ONE et des réponses de proximité qui en découlent :

- L'adresse du nœud initiateur de la découverte;
- L'adresse de la destination.

❖ Étape 4 de la découverte: Réception des PREQ2_ONEs par les voisins des meilleurs voisins

Lorsqu'il reçoit une PREQ2_ONE, chaque voisin I des meilleurs voisins $MV_{S,D}$ évalue sa distance à la destination D après avoir récupéré sa propre position. Il procède ensuite à l'étape 5 de la découverte de proximité.

Un même nœud I est susceptible de recevoir plusieurs PREQ2_ONEs pour une même découverte. Pour éviter la diffusion à plusieurs reprises d'une même PREP2, lorsqu'un nœud reçoit pour la première fois une PREQ2_ONE correspondant à une découverte spécifique, il vérifie s'il a déjà mémorisé cette découverte dans son cache de PREQ2s :

- Si l'il n'a pas d'entrée dans son cache de PREQ2s pour cette découverte, le nœud I évalue s'il est un meilleur voisin pour le noeud émetteur de la PREQ2_ONE, comme nous venons de le voir. Si c'est le cas, alors le nœud I mémorise cette découverte (adresse du nœud initiateur, nombre de destinations actives et adresse de la destination active s'il y a lieu). Il diffuse ensuite une PREP2 (étape 5 de la découverte de proximité).
- Si la découverte associée à la PREQ2_ONE est déjà mémorisée dans son cache de PREQ2s, le nœud I ne traite pas la PREQ2_ONE.

D'autre part, les nœuds ayant reçu une PREQ1 relative à la même découverte que la PREQ2_ONE, soient les voisins directs de N , ignorent la PREQ2_ONEs puisque celle-ci a pour objectif de découvrir leur voisinage à 2 sauts. Ainsi, les meilleurs voisins $MV_{N,D}(A)$, $MV_{N,D}(B)$ et $MV_{N,D}$ ne traitent pas les PREQ2_ONEs émises pour la découverte initiée par N . Par ailleurs, le nœud initiateur de la découverte, N , ignore aussi ces PREQ2_ONEs pour des raisons évidentes.

❖ Étape 5 de la découverte: Diffusion des PREP2s par les meilleurs voisins à 2 sauts

A la réception d'une PREQ2_ONE, chaque voisin à 2 sauts I de N diffuse une PREP2. Dans le cadre d'une découverte à destination unique, cette PREP2 contient l'adresse de la destination active et la distance du meilleur voisin émetteur $MV2_{S,D}(V, I)$ à cette destination. Encore une fois, nous traitons ici le cas où la découverte ne concerne qu'une seule destination. Autrement, les informations insérées dans l'en-tête de la PREP2 seraient déclinées pour toutes les destinations incluses dans l'en-tête de la PREQ2 correspondante comme le montre le Tableau annexé E.4.

Considérant notre exemple, (voir Figure 3.4 (c)), les nœuds R , C , H , F et G diffusent donc une PREP2.

❖ Étape 6 de la découverte: Réception des PREP2s par les meilleurs voisins

Les PREP2s ne sont traitées que par les nœuds qui ont émis une PREQ2_ONE pour la découverte correspondante, soient les meilleurs voisins $MV_{N,D}$ du nœud initiateur N pour la destination D . Lorsqu'un meilleur voisin $MV_{N,D}(V)$ reçoit une PREP2 d'un de ses propres voisins, il vérifie d'abord si le nœud I , émetteur de la PREP2, est la destination :

- Si I est la destination, $MV_{N,D}(V)$ change ses attributs pour sa meilleure destination D : il change la valeur de son indice d'accessibilité à 1 et incrémenté son indice de proximité.
- Sinon, $MV_{N,D}(V)$ compare sa distance à la destination D , qu'il a, rappelons le, mémorisée dans son cache de PREP1s, à celle du nœud émetteur de la PREP2 :

- Si $DIST(I, D) < DIST(MV_{N,D}(V), D)$, le nœud I est un meilleur voisin pour $MV_{N,D}(V)$. Le noeud $MV_{N,D}(V)$ met à jour ses caractéristiques pour cette découverte à jour dans son cache de PREP1s, à savoir , il incrémente son indice de proximité.
- Sinon, $MV_{N,D}(V)$ ne fait rien.

Le traitement des PREP2s se poursuit ainsi tant que le délai *reply_delay* alloué à la phase 2 de la découverte de proximité (constituée des étapes 3, 4, 5 et 6) ne s'est pas écoulé. Ce délai doit en effet permettre aux meilleurs voisins de N de recevoir toutes les PREP2s générées par leurs propres meilleurs voisins pour cette découverte.

Comme le montre la Figure 3.4 (d), le nœud F est un meilleur voisin de $MV_{N,D}(A)$, tandis que les nœuds G et H sont des meilleurs voisins de $MV_{N,D}(B)$. Ces nœuds constituent donc des meilleurs voisins à 2 sauts de N et sont notés $MV2_{N,D}(F)$, $MV2_{N,D}(G)$, et $MV2_{N,D}(H)$. La valeur de l'indice de proximité des meilleurs voisins de N , $MV_{N,D}(A)$ et $MV_{N,D}(B)$ est donc respectivement de 1 et 2, tandis que celle de $MV_{N,D}(E)$ est nulle.

Une fois le délai *reply_delay* expiré, la phase 2 de la découverte de proximité est terminée : les meilleurs voisins $MV_{N,D}$ du nœud initiateur N connaissent leurs caractéristiques pour cette découverte, soient leur indice d'accessibilité, leur indice de proximité et leur distance à la destination. Les meilleurs voisins $MV_{N,D}$ peuvent alors passer à l'exécution de l'étape suivante.

L'Annexe G.1 illustre les informations stockées dans le cache de PREP1s de chaque meilleur voisin $MV_{N,D}(V)$ de N à la fin de la phase 2. Les coordonnées de chaque meilleure destination sont aussi mémorisées.

❖ Étape 7 de la découverte : Envoi des PREP1s par les meilleurs voisins

Lorsque le délai alloué à la phase 2 d'une découverte spécifique expire, les meilleurs voisins $MV_{N,D}$ du nœud initiateur de la découverte, N , sont susceptibles de diffuser une réponse de proximité de niveau 1 ou PREP1 destinée au nœud N . Susceptibles, car seuls les meilleurs voisins de N dont l'indice de proximité est supérieur

à 0 diffuseront une PREP1, puisqu'ils permettent une avancée géographique vers D sur au moins 2 sauts. Les meilleurs voisins dont l'indice de proximité est nul, comme le nœud $MV_{N,D}(E)$, sont, de fait, concaves, et ne présentent donc pas d'intérêt pour PARA (Figure 3.4 (d)). Le format général de l'entête d'une PREP1 est donné par le Tableau annexé.F.6. Dans le cadre d'une découverte de proximité à destination unique, les informations insérées dans l'entête d'une PREP1 par un meilleur voisin $MV_{N,D}(V)$ non concave sont tirées de son cache de PREP1s, soient dans notre exemple:

- L'adresse du nœud initiateur de la découverte, N ;
- Le nombre de meilleures destinations, qui vaut ici 1;
- L'adresse de la destination de la découverte, D ;
- Son indice d'accessibilité, $IsDest(MV_{N,D}(V), D)$;
- Son indice de proximité, $IProx(MV_{N,D}(V), D)$;
- Sa distance à D , $DIST(MV_{N,D}(V), D)$.

Si le nœud émetteur de la PREP1 est la destination de la découverte, il s'agit d'une PREP1 anticipée. Le délai *reply_delay* n'est alors à priori alors pas écoulé. Le nœud émetteur d'une PREP1 anticipée insère dans l'entête de celle-ci les mêmes champs que précédemment, mais avec les valeurs spécifiques suivantes :

- $IsDest(MV_{N,D}(V), D) = 2$;
- $IProx(MV_{N,D}(V), D) = 1$;
- $DIST(MV_{N,D}(V), D)$.

Globalement, lorsque le délai *reply_delay* de la découverte de proximité arrive à terme, tous les voisins du nœud initiateur suppriment l'entrée correspondante de leur cache de PREP1s tandis que les nœuds ayant mémorisé une PREQ2 pour cette découverte font de même dans leur cache de PREQ2s.

❖ Étape 8 de la découverte: Réception des PREP1s par le nœud initiateur

Les PREP1s permettent :

- Aux nœuds nouvellement actifs et en cours de découverte de proximité initiale de peupler leur table de meilleurs voisins;

- Aux nœuds déjà actifs et disposant d'une table de meilleurs voisins peuplée, de mettre à jour les entrées correspondantes aux nœuds émetteurs des PREP1s.

Le nœud initiateur de la découverte de proximité, N , identifie les PREP1s qui lui sont destinées en vérifiant que l'adresse contenue dans le champ *discv_id* est bien la sienne. Si c'est le cas, il traite les informations contenues dans la PREP1 et met à jour sa table de meilleurs voisins pour cette destination. Les nœuds recevant une PREP1 qui ne leur est pas adressée la suppriment.

A la réception d'une PREP1, le nœud N va donc insérer les informations concernant le meilleur voisin correspondant, s'il s'agit d'une découverte initiale, dans sa table de meilleurs voisins pour D , ou encore le mettre à jour, s'il s'agit d'une découverte périodique. Les meilleurs voisins sont insérés dans la table selon un ordre bien précis basé sur la valeur de leurs trois attributs, soient, en ordre prioritaire :

- L'indice d'accessibilité : la priorité du meilleur voisin est proportionnelle à la valeur de cet attribut, puisqu'il reflète la possibilité d'atteindre la destination via ce meilleur voisin;
- L'indice de proximité : à indice d'accessibilité égaux, le meilleur voisin privilégié sera celui dont l'indice de proximité est le plus élevé, c'est-à-dire celui maximisant les chances de progrès géographique vers la destination;
- La distance à la destination : pour des valeurs d'indices d'accessibilité et de proximité égales, le meilleur voisin prioritaire sera celui minimisant la distance à la destination.

L'Annexe G.2 illustre la table ainsi ordonnée des meilleurs voisins du nœud N pour la destination D à l'issu de sa découverte de proximité initiale avec $DIST(B, D) > DIST(A, D)$. Chaque entrée de la table de meilleurs voisins est valide pour un délai fixé à 3 fois l'intervalle des découvertes périodiques, *prequest_int*, à partir du moment de son insertion ou de sa mise à jour. Lorsque ce délai expire, l'entrée est supprimée de la table.

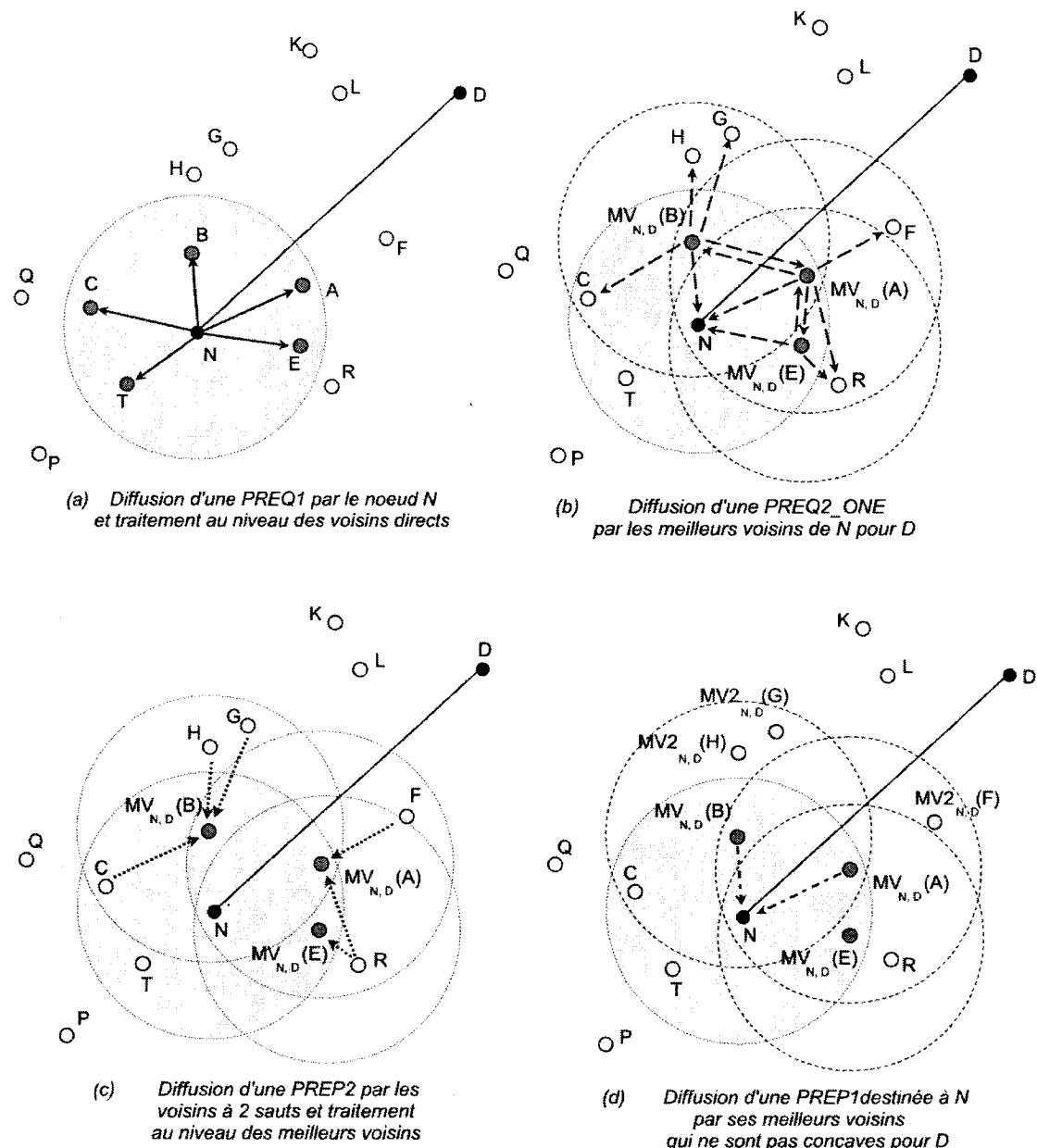


Figure 3.4 : Échanges de paquets de contrôle dans une découverte de proximité à destination unique

En définitive, les différentes étapes d'une découverte de proximité à destination unique peuvent être résumées par la Figure 3.5.

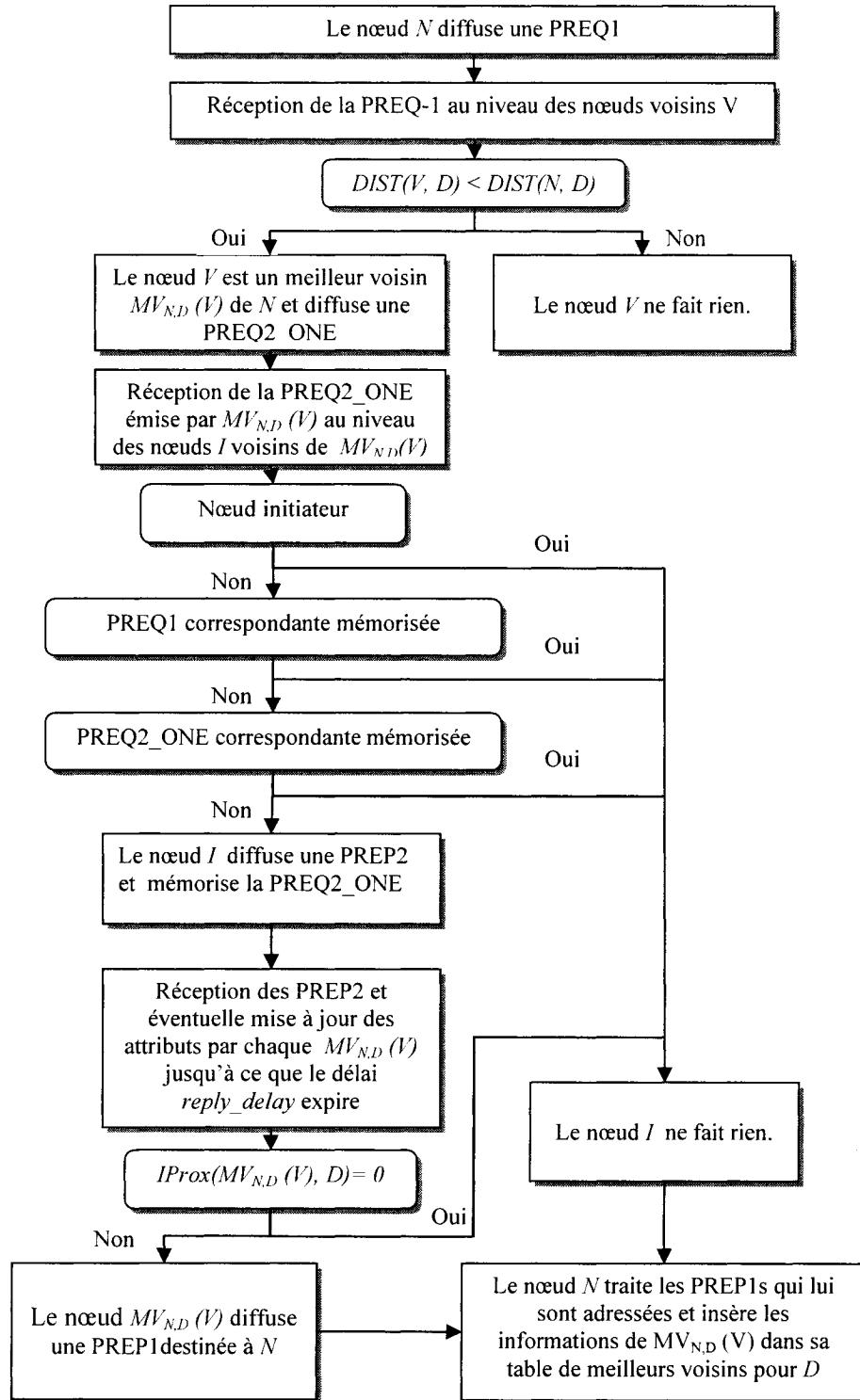


Figure 3.5 : Déroulement d'une découverte de proximité à destination unique

3.3.4.2 Découverte de proximité à destinations multiples

Seules les découvertes de proximité périodiques sont susceptibles d'impliquer plusieurs destinations, car elles doivent permettre au noeud initiateur de mettre à jour sa table de meilleurs voisins pour chacune de ses destinations actives. Le déroulement global de ce type de découverte est semblable à celui décrit dans la section précédente et comprend les mêmes 8 étapes. Les principales différences par rapport à une découverte de proximité pour une destination unique résident dans la manière dont les requêtes de proximité sont traitées. L'Annexe D décrit de manière détaillée le déroulement d'une découverte de proximité mettant en jeu plusieurs destinations.

Nous venons de décrire les différentes étapes permettant à un noeud:

- De peupler sa table de voisin pour une destination spécifique, par une découverte de proximité initiale ponctuelle;
- De mettre à jour sa table de meilleurs pour chacune de ses destinations, par des découvertes de proximité périodiques.

A l'issu de toute découverte de proximité, le noeud initiateur a ainsi une vision globale des avancées géographiques possibles des paquets de données qu'il détient vers leur destination via chacun de ses meilleurs voisins. Doté d'une table de meilleurs voisins peuplée et à jour pour chacune de ses destinations actives, le noeud initiateur est alors en mesure de choisir le prochain saut des paquets en fonction de leur destination.

La section suivante s'attache à décrire le choix du prochain noeud transmetteur des paquets de données.

3.3.5 Sélection du noeud transmetteur

PARA a pour objectif notable de réduire le risque d'atteindre un noeud concave dans un acheminement géographique. Soit le noeud N , détenteur d'un paquet de données à transmettre à une destination D . Au lieu de transmettre le paquet au noeud voisin de N minimisant la distance géographique à parcourir pour atteindre D , comme dans la plupart des stratégies de routage localisé, PARA transmet le paquet au noeud V qui :

- Constitue un meilleur voisin pour D , c'est-à-dire permet de faire moindrement progresser géographiquement le paquet vers D :

$$DIST(V, D) < DIST(N, D) \quad (3.1)$$

- Possède lui-même, parmi tous les meilleurs voisins de N , le plus grand nombre de meilleurs voisins pour D , soit le plus grand indice de proximité :

$$IProx(V) = \max(IProx(I))_{\substack{I \in MV \\ N, D}} \quad (3.2)$$

PARA anticipe la présence de nœuds concaves en transmettant le paquet au nœud possédant le plus grand nombre de meilleurs voisins, maximisant ainsi les chances de progrès géographique vers la destination. Cette approche favorise le nombre de meilleurs voisins au détriment de l'avancée géographique dans le choix du nœud transmetteur. Cette stratégie peut donc, dans sa conception de base, présenter des lacunes dans le cas où la destination du paquet est atteignable en deux sauts via un meilleur voisin qui n'a pas l'indice de proximité le plus élevé. En effet, si on se contente dans ce cas de transmettre le paquet au meilleur voisin dont l'indice de proximité est le plus élevé, on rallonge le chemin vers la destination. La Figure 3.6 illustre une telle situation. Les meilleurs voisins de N pour la destination D sont $MV_{N,D}(J)$ et $MV_{N,D}(I)$ avec $IProx(MV_{N,D}(J), D) = 1$ et $IProx(MV_{N,D}(I), D) = 2$. Si on applique notre algorithme tel que présenté jusqu'ici, les paquets de données seront acheminés à leur destination via le nœud $MV_{N,D}(I)$ qui a l'indice de proximité le plus élevé. L'acheminement se fait alors en 3 sauts, alors que la destination est atteignable via le nœud $MV_{N,D}(J)$ en 2 sauts.

Ici entre en jeu l'indice d'accessibilité du meilleur voisin, qui reflète l'accessibilité d'une destination D via le meilleur voisin concerné. Comme nous l'avons vu dans la section précédente, pour chaque destination active, les meilleurs voisins sont ordonnés et maintenus tels quel dans la table de meilleurs voisins en fonction de leur priorité, établie sur leur indice d'accessibilité, leur indice de proximité et leur distance à la destination. Ainsi, la table de meilleurs voisins du nœud N pour la topologie illustrée par la Figure 3.6 est donnée par l'Annexe I.

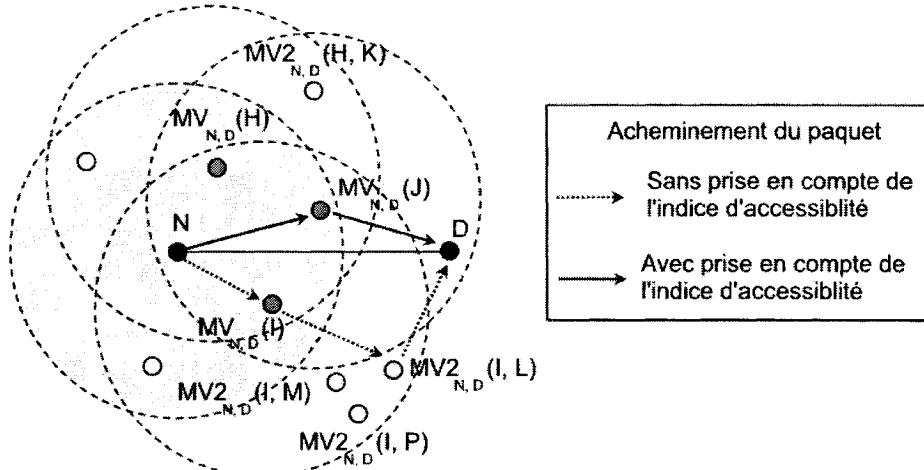


Figure 3.6 : Rallongement de la route vers la destination

Quand vient le temps de sélectionner le prochain saut d'un paquet, le nœud transmetteur choisit donc le meilleur voisin en tête de sa table de meilleurs voisins pour la destination du paquet, dans ce cas, *J*. De la même manière, dans le cadre de la découverte de proximité à destination unique illustrée dans la Section 3.3.4.1, le prochain nœud transmetteur des paquets à destinations de *D* sera *B*.

3.3.6 Transmission des paquets de données

Nous l'avons évoqué dans la Section 3.3.3, lorsqu'un nœud *N* génère ou reçoit pour la première fois un paquet de données destiné à un nœud pour lequel il n'a pas d'informations dans sa table de meilleurs voisins, il lance une découverte de proximité initiale pour cette destination et insère le paquet dans un tampon. De même, les paquets de données reçus ou générés pour cette destination avant la fin de la découverte initiale sont insérés dans le tampon.

C'est ici que la durée prédéterminée et fixée par le paramètre *discv_delay* de la découverte de proximité prend toute son importance. Ce délai doit permettre, d'une part, aux meilleurs voisins d'exécuter la phase 2 de la découverte (il est donc plus grand que le délai alloué à la phase 2, *reply_delay*) et d'autre part, au nœud initiateur de pleinement

peupler sa table de meilleurs voisins en « attendant » la réception de toutes les PREP1s émises par ses meilleurs voisins.

Nous estimons donc que lorsque le délai alloué à la découverte est expiré, le nœud initiateur connaît la totalité de son entourage et est alors en mesure de transmettre les paquets de données. Certes, cette estimation reste du domaine de la supposition, mais la table de meilleurs demeure, à priori, après ce délai, suffisamment peuplée pour permettre au nœud N de faire un choix judicieux quant au prochain nœud transmetteur des paquets de données. A la fin de la découverte initiale, le nœud N transmet donc tous les paquets pour cette destination contenus dans son tampon au prochain saut.

Autrement, dans le cadre de découvertes périodiques, le délai alloué à la découverte est de moindre importance, puisque le nœud initiateur dispose déjà d'une table de meilleurs voisins pour chacune de ses destinations actives. Les entrées de sa table de meilleurs voisins sont alors mises à jour, et les paquets reçus ou générés entre temps sont transmis en utilisant les données de la table de meilleurs voisins telles qu'elles sont.

Avant de transmettre un paquet de données, le nœud N insère son indice de proximité dans l'entête PARA du paquet. Ainsi, dans une topologie telle que celle illustrée sur la Figure 3.6 dans la section précédente, le nœud évalue, en faisant le décompte de ses meilleurs voisins pour D , son indice de proximité à 3 et insère cette valeur dans tous les paquets de données destinés à D qu'il transmet au noeud J .

A la réception d'un paquet de données, chaque nœud intermédiaire vérifie à son tour s'il a une table de meilleurs voisins pour la destination du paquet. Si oui, il retransmet directement le paquet de données. Sinon, il lance à son tour une découverte de proximité initiale. De plus, si le nœud n'est pas actif au moment où il reçoit le paquet de données, il démarre son compteur pour les découvertes de proximité périodiques. Le processus se poursuit ainsi jusqu'à l'atteinte de la destination par le paquet de données, tel que l'illustre la Figure 3.7.

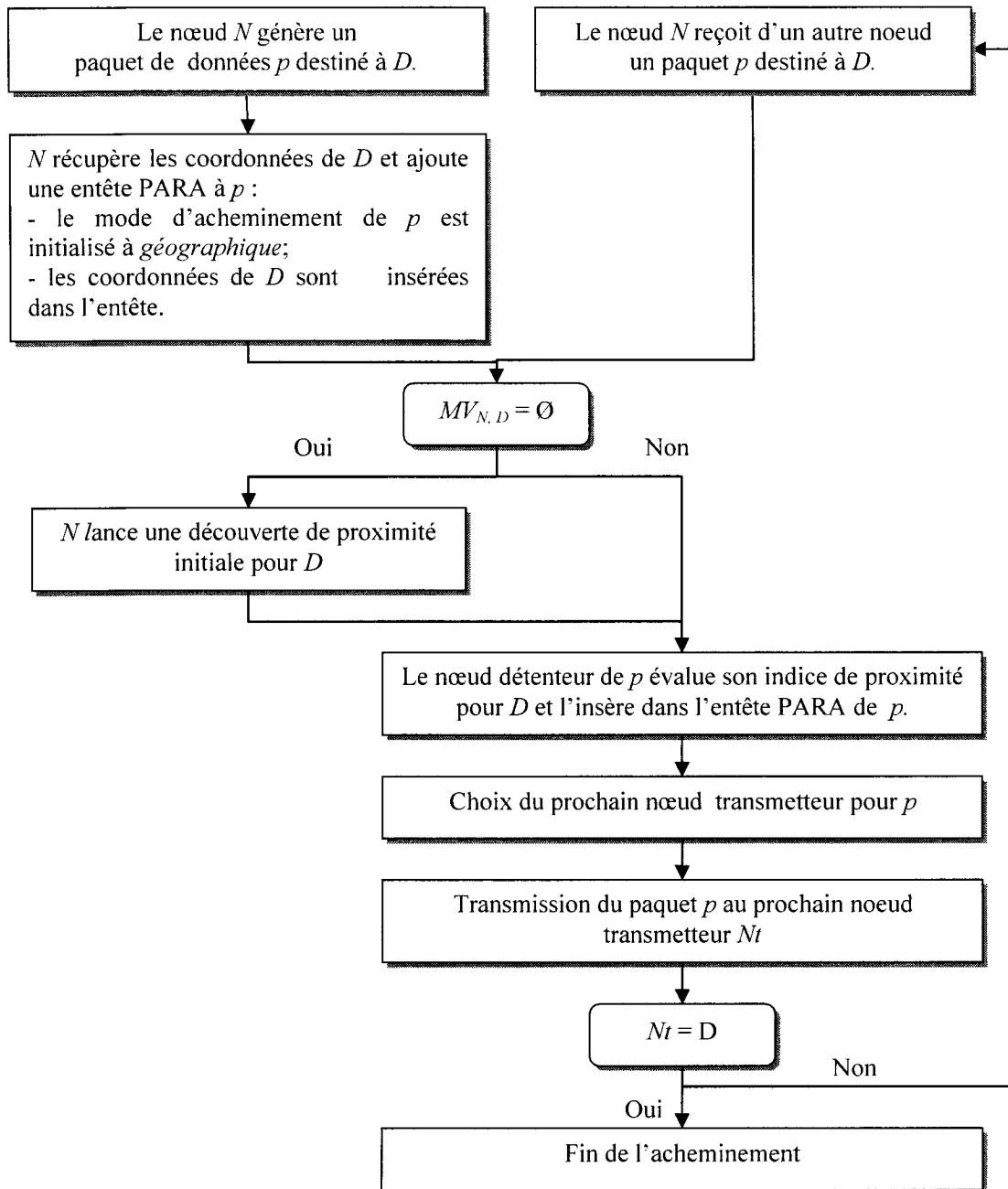


Figure 3.7: PARA, algorithme de routage de base

3.3.7 Mise à jour de la table de meilleurs voisins

Les réseaux ad hoc sont, par définition, mobiles. Un nœud ne peut donc considérer perpétuellement valides les informations obtenues à l'issu d'une découverte de

proximité initiale. Ces informations doivent en effet être rafraîchies régulièrement afin de maintenir à jour les données du nœud sur son voisinage en dépit de sa mobilité.

Dans un routage topologique, la mise à jour de la table de routage d'un nœud est effectuée dans le cadre de la maintenance de route, qui sous-tend l'établissement d'une route de bout-en-bout de la connexion et assume que cette route est relativement stable. La maintenance n'a alors lieu que lorsqu'un bris est détecté sur la route.

Avec un routage géographique tel que PARA, le concept de route n'existe pas. Les mises à jour se font de manière isolée au niveau de chaque nœud actif du réseau via des découvertes de proximité périodiques évoquées dans la Section 3.3.4.2. Ces mises à jour périodiques sont lancées dès que le nœud devient actif. Si le nœud cesse de générer ou de transmettre des paquets de données, elles sont interrompues. L'intervalle d'exécution des découvertes de proximité périodiques est prédéterminé et fixé par le paramètre par *prequest_interval*.

Une entrée dans la table de meilleurs voisins sera par ailleurs considérée valide durant un intervalle équivalent à 3 fois *prequest_interval* depuis sa dernière mise à jour. Chaque entrée se voit ainsi attribuée un temps d'expiration. Ce temps est renouvelé chaque fois que le nœud insère ou met à jour une entrée dans la table de meilleurs voisins à la réception d'une PREP1. Par la suite, lorsqu'une découverte de proximité arrive à son terme, le nœud initiateur parcourt sa table de meilleurs voisins et supprime les entrées qui ont dépassé leur temps d'expiration.

3.3.8 Atteinte de la destination

Comme nous l'avons vu jusqu'ici, une découverte de proximité initiale lancée par un nœud N engendre une certaine latence au niveau des voisins de N d'une part, et au niveau même de N d'autre part, puisqu'il faut "attendre" les réceptions respectives de la totalité supposée des PREP2s et PREP1s.

Pour limiter cette latence lorsque la destination est détectée dans un rayon de 2 sauts, un certain nombre de mécanismes, que nous décrirons dans cette section, entrent en jeu.

Nous traitons encore une fois les découvertes de proximité à destination unique et à destinations multiples différemment.

3.3.8.1 Découverte de proximité à destination unique

Reprenons le déroulement classique d'une découverte de proximité initiale. Le nœud N lance une découverte de proximité initiale pour la destination D . Deux cas de figure se posent en ce qui concerne la détection de D : D peut être détectée dans le voisinage direct de N ou dans le voisinage à 2 sauts de N .

➤ Détection de la destination dans le voisinage direct de N

A la réception d'une PREQ1, chaque voisin de N vérifie s'il est la destination, auquel cas il envoie une PREP1, dite *anticipée*, au nœud N . Cette PREP1 est anticipée, dans la mesure où, contrairement aux autres voisins qui, éventuellement attendront la fin de la phase2 de la découverte de proximité pour envoyer leur PREP1, le nœud destinataire D envoie sa PREP1 immédiatement. Rappelons que cette PREP1 contient notamment:

- L'indice d'accessibilité de D , qui vaut 2;
- L'indice de proximité de D , égal à 1;
- La distance du nœud émetteur (D) à D , valant 0.

Chaque fois que le nœud N reçoit une PREP1, il insère le meilleur voisin dans sa table de meilleurs voisins pour la destination D . Ce n'est alors qu'au moment où le délai alloué à la découverte de proximité, *discv_delay*, expire, que le nœud N envoie les paquets destinés à D présents dans le tampon. Toutefois, lorsque N insère un meilleur voisin dans sa table avec un indice d'accessibilité égal à 2, c'est que ce meilleur voisin est la destination D pour laquelle il a lancé une découverte initiale. Dans ce cas, le nœud N n'attend pas la fin de la découverte : les paquets du tampon sont transmis directement et immédiatement à la destination. Les PREP1s reçues ultérieurement des autres meilleurs voisins seront insérées dans la table normalement.

➤ **Détection de la destination à 2 sauts de N**

Lorsqu'un meilleur voisin $MV_{N,D}(V)$ de N détecte la destination D dans son voisinage, la PREP1 qu'il envoie à N à la fin de la phase 2 de la découverte de proximité contient, entre autres:

- Son indice d'accessibilité, $IsDest(MV_{N,D}(V), D) = 1$;
- Son indice de proximité, $IProx(MV_{N,D}(V), D)$;
- Sa distance à D , $DIST(MV_{N,D}(V), D)$.

Si la destination D est dans le voisinage direct d'un meilleur voisin $MV_{N,D}(V)$, ce meilleur voisin a de fortes chances d'être choisi comme prochain nœud transmetteur des paquets de données pour D . Pour éviter de lancer à son tour une découverte initiale, au cas où il recevrait effectivement des paquets de données pour D et sachant que D est dans son voisinage direct, le meilleur voisin $MV_{N,D}(V)$ insère une entrée anticipée pour la destination D dans sa table de meilleurs voisins. Cette entrée équivaut à celle qui aurait été insérée si $MV_{N,D}(V)$ avait reçu une PREP1 proactive de D , soit :

- $isDest(MV2_{N,D}(V, D), D) = 2$;
- $IProx(MV2_{N,D}(V, D), D) = 1$;
- $DIST(MV2_{N,D}(V, D), D) = 0.0$.

Si une entrée dans la table de meilleurs voisins a normalement un délai d'expiration équivalent à 3 fois l'intervalle des découvertes périodiques, une entrée insérée de manière proactive pour une destination a une durée de validité beaucoup plus courte, environ égale au tiers du même intervalle. En effet, tous les meilleurs voisins ayant détecté la destination D effectuent une insertion anticipée de D dans leur table de meilleurs voisins, mais seul le nœud qui sera choisi comme prochain nœud transmetteur tirera parti de cette insertion. Lorsque le meilleur voisin en question reçoit effectivement un paquet de données pour D , il :

- Met à jour le délai d'expiration de l'entrée de D dans sa table de meilleurs voisins. Le délai est alors le délai normal, soit 3 fois *prequest_interval*;
- S'il passe à l'état actif, il démarre le déclencheur pour les découvertes de proximité périodiques;

- Transmet le paquet de données directement à D , sans lancer de découverte de proximité initiale.

Les autres meilleurs voisins, ne recevant pas de paquet pour D , supprimeront l'entrée pour D de leur table de meilleurs voisins.

3.3.8.2 Découverte de proximité à destinations multiples

Dans le cadre d'une découverte de proximité périodique à destinations multiples, le nœud initiateur N diffuse une PREQ1 contenant les informations de chacune de ses destinations actives Di , avec $1 \leq i \leq NDst$ où $NDst$ est le nombre de destinations actives de N . Décrivons les processus engagés selon que l'une des destinations actives se situe dans le voisinage direct de N ou dans son voisinage à 2 sauts.

➤ **Détection d'une destination active dans le voisinage direct de N**

Lorsque les voisins directs de N reçoivent une PREQ1, ils évaluent pour quelles destinations incluses dans l'entête de la PREQ1, ils constituent un meilleur voisin de N . Les meilleurs voisins MV_N , qui sont meilleurs voisins de N pour au moins une de ses destinations actives, initient alors la phase 2 de la découverte de proximité en diffusant des PREQ2s. Le nœud N reçoit inévitablement les PREQ2s diffusées par ses meilleurs voisins. Chaque fois qu'il reçoit une PREQ2, le nœud initiateur de la découverte vérifie si le nœud émetteur est l'une de ses destinations actives. Si c'est le cas, il insère une entrée dans sa table de meilleurs voisins pour cette destination ou met à jour l'entrée correspondante.

Par ailleurs, tous les nœuds actifs du réseau analysent en permanence le trafic dans leur rayon de transmission. S'ils perçoivent un paquet de contrôle dont le nœud émetteur est l'une de leurs destinations actives, ils mettent à jour leur table de meilleurs voisins pour cette destination en conséquence.

➤ Détection d'une destination active à 2 sauts du nœud N

A la fin de la phase 2 de la découverte de proximité, chaque meilleur voisin $MV_N(V)$ parcourt son cache de PREP1s et pour chacune de ses meilleures destinations, insère ses caractéristiques dans l'entête de la PREP1 qu'il va envoyer au nœud N . Lors de ce processus, il vérifie aussi son indice d'accessibilité pour chaque meilleure destination notée MDj , avec $1 \leq j \leq N_MDst(MV_N(V))$, où $N_MDst(MV_N(V))$ est le nombre de meilleures destinations de $MV_N(V)$:

- Si $IsDest(MV_N(V), MDj) = 1$, la destination MDj est située dans l'entourage direct de $MV_N(V)$. $MV_N(V)$ procède alors de la même manière que pour une découverte de proximité à destination unique, c'est-à-dire qu'il insère une entrée anticipée pour MDj dans sa table de meilleurs voisins, avec :

- $IsDest(MV2_N(V, MDj), MDj) = 2$;
- $IProx(MV2_N(V, MDj), MDj) = 1$;
- $DIST(MV2_N(V, MDj), MDj) = 0.0$;
- $t_{ex} = .3 \times prequest_interval$.

Puisque nous nous plaçons dans le cadre d'une découverte de proximité périodique, au moins une transmission de paquets a déjà eu lieu pour toutes les destinations actives du nœud N . Il se peut donc que le meilleur voisin $MV_N(V)$ ait déjà une entrée pour MDj dans sa table de meilleurs voisins. Dans ce cas, il effectue simplement une mise à jour du délai d'expiration.

- Sinon, il ne fait rien.

Plusieurs meilleurs voisins MV_N sont ainsi susceptibles d'avoir dans leur table de meilleurs voisins des entrées anticipées pour une ou plusieurs destinations. Nous nous intéressons ici au cas où les meilleurs voisins MV_N n'avaient pas d'entrée dans leur table concernant l'une des destinations actives de N , que nous noterons D . Comme pour une découverte à destination unique, seul le nœud choisi comme prochain nœud transmetteur des paquets destinés à D conservera cette entrée à jour, les autres la supprimeront. Cette logique vaut pour toutes les destinations actives.

3.3.9 Détection des nœuds concaves

Il peut arriver malgré tout qu'un paquet de données soit transmis à un nœud concave, soit un nœud via lequel le paquet ne peut plus progresser physiquement vers la destination. Comme nous allons le voir, divers cas de figures peuvent se présenter. Dans la section suivante, nous décrirons les mécanismes que PARA met en œuvre pour faire face à de telles situations.

➤ Absence de meilleurs voisins et meilleurs voisins concaves par anticipation

Si, à la fin d'une découverte de proximité à destination unique, un nœud N ne reçoit aucune PREP1, alors le nœud N est considéré concave pour cette destination. Cependant, dans les faits, l'absence de PREP1 peut s'expliquer de deux manières :

- Soit le nœud N est réellement concave;
- Soit ses meilleurs voisins sont eux-mêmes concaves pour cette destination. En effet, nous l'avons évoqué dans la Section 3.3.4, de tels meilleurs voisins ne diffusent pas de PREP1, puisqu'ils ne permettent pas la progression du paquet de données sur 2 sauts. Du point de vue de N , ils sont concaves par anticipation.

Une telle situation peut aussi se produire dans le cadre d'une découverte de proximité périodique à destinations multiples.

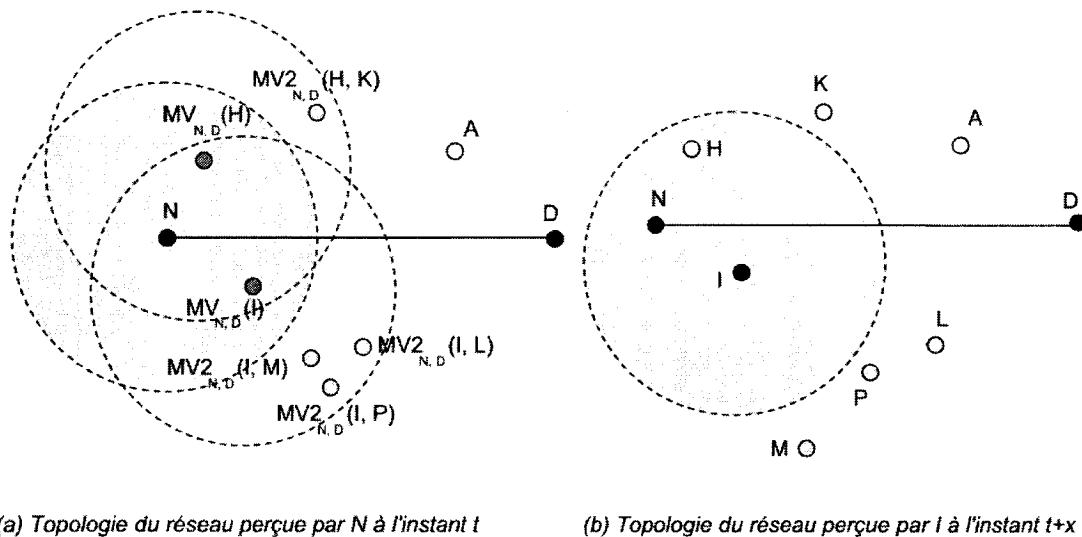
➤ Déplacement des meilleurs voisins

Supposons que le nœud N ait un paquet de données au nœud D à transmettre. La Figure 3.8 (a) présente la topologie du réseau, telle que perçue par le nœud N , à l'instant t . N est le seul nœud actif du réseau. A l'issu de sa découverte de proximité initiale, à l'instant t , la table de meilleurs voisins du nœud N pour D contient les données présentées dans le Tableau 3.1.

Le nœud N transmet donc le paquet de données au nœud I , qui est alors supposé avoir trois meilleurs voisins pour la destination D : les nœuds L , P et M (Figure 3.8 (a)).

Tableau 3.1: Contenu de la table de meilleurs voisins de N pour la Figure 3.8

Dst_id	$MV_N(V)$	$IsDest$ ($MV_N(V), MDi$)	$IProx$ ($MV_N(V), MDi$)	$DIST(MV_N(V), MDi)$	t_{ex}
D	I	0	3	$DIST(I, D)$	T
	H	0	1	$DIST(H, D)$	T

**Figure 3.8 :** Exemple de déplacement de meilleurs voisins

A la réception du paquet, le nœud I lance une découverte de proximité initiale pour la destination D . Cependant, le nœud I ne reçoit alors aucune PREP1. Les nœuds M, P, L se sont en effet déplacés durant l'intervalle de temps x séparant l'envoi des PREP1s pour la découverte de N et la réception du paquet par I . Ces nœuds peuvent :

- Etre demeurés dans le rayon de transmission de I , mais s'être éloignés de la destination;
- S'être déplacés hors de la portée du nœud I .

La Figure 3.8 (b) illustre la topologie du réseau à l'instant $t+x$ dans ce dernier cas. En définitive, le nœud I s'avère concave pour la destination D , contrairement à ce que démontrait la découverte de N .

Dans la section suivante, nous présentons les mécanismes de gestion de nœuds concaves de PARA.

3.3.10 Gestion des nœuds concaves

Lorsqu'un paquet de données atteint un noeud concave, il faut tenter de retrouver un contexte favorable au progrès physique du paquet vers sa destination. Rappelons que le nœud N ayant transmis le paquet au nœud concave C a inclus son indice de proximité dans l'entête du paquet. Différents cas sont envisageables selon la valeur prise par l'indice de proximité de N inclus dans l'entête du paquet.

3.3.10.1 Mode recouvrement

Un indice de proximité inclus dans le paquet supérieur à 1 signifie que le nœud N avait plus d'un meilleur voisin au moment du choix du prochain nœud transmetteur du paquet. Ainsi, à moins que le nœud N , ou ses meilleurs voisins, ne se soient déplacés, il y a à priori au moins un autre nœud, mis à part le nœud concave C , permettant de faire progresser géographiquement le paquet vers la destination. Un mécanisme de *recouvrement* est alors mis en place: il s'agit de renvoyer le paquet à son saut précédent, N , en espérant que celui-ci soit en mesure de le transmettre à un autre meilleur voisin. Le processus de recouvrement, initié par un nœud concave C , se déroule comme suit:

➤ **Passage en mode recouvrement :** A l'issu de sa découverte de proximité, le nœud concave C renvoie le paquet au nœud N , dont il l'a reçu, après avoir modifié son entête PARA. En effet, le nœud C :

- Change le mode d'acheminement du paquet de *géographique* à *recouvrement*;
- Insère son adresse et sa distance à D , $DIST(C, D)$ dans l'entête du paquet;
- Enregistre sa propre adresse dans son cache de nœuds concaves pour éviter de recevoir de nouveau des paquets pour la destination D . Ainsi, par la suite,

comme nous le verrons plus tard, il n'interviendra pas dans les découvertes de proximité à destination unique pour D .

- **Réception d'un paquet de données en mode recouvrement :** A la réception d'un paquet de données, le nœud N vérifie le mode d'acheminement du paquet. Si le paquet est en mode géographique, le paquet est routé selon le principe que nous avons décrit jusqu'ici. Si paquet est en mode recouvrement, le nœud concave, soit C :

- Est retiré de la table de meilleurs voisins de N pour la destination du paquet;
- Est enregistré dans le cache de nœuds concaves de N .

Le nœud N transmet alors le paquet au prochain nœud transmetteur choisi de sa table de meilleurs voisins, soit, dans l'exemple illustré par la Figure 3.8, le nœud H . Par la suite, chaque nœud recevant le paquet fera de même. La Figure 3.9 illustre le processus de recouvrement utilisé dans l'exemple.

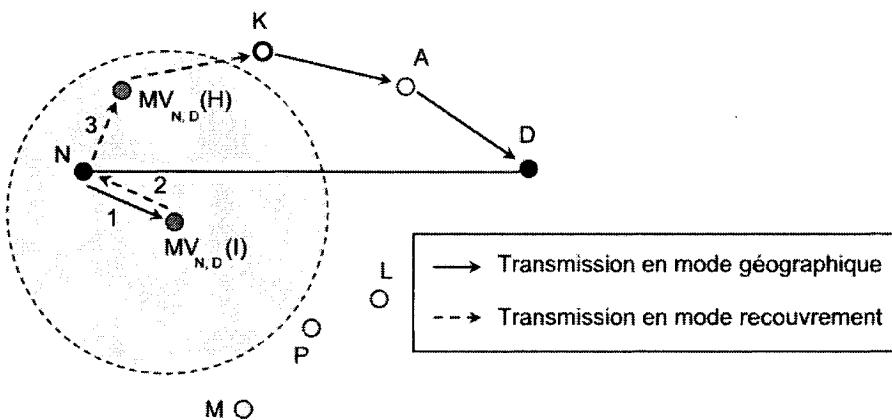


Figure 3.9 : Passage en mode Recouvrement

L'adresse du nœud concave doit être mémorisée par les noeuds pour éviter l'envoi d'un paquet de données destiné à D à un nœud justement concave pour D . En effet, chaque fois qu'un nœud V reçoit un paquet de contrôle PARA, il effectue les vérifications suivantes, afin de ne pas considérer un nœud concave dans le choix du prochain nœud transmetteur d'un paquet :

- Si le paquet de contrôle est une PREQ1 à destination unique ou une PREQ2_ONE, le nœud V vérifie qu'il n'est pas concave pour la destination concernée avant de traiter le paquet. S'il est concave pour cette destination, il ne diffuse selon le cas, ni PREQ2_ONE, ni PREP2.
- S'il s'agit d'une PREP2 ou d'une PREP1, le nœud V vérifie lors de l'insertion des informations dans son cache de PREP1s ou dans sa table de meilleurs voisins, qu'il n'a pas d'entrée pour le nœud émetteur du paquet dans son cache de nœuds concaves. En effet, si le nœud C , émetteur du paquet, est enregistré concave pour une destination D_i , ses attributs pour cette destination en particulier ne seront pas insérés.
- A la réception d'une PREQ1 à destinations multiples, le nœud V n'insère parmi ses meilleures destinations, que celles pour lesquelles il ne s'est pas mémorisé concave.

Une PREQ2 à destinations multiples, quant à elle, ne nécessite pas de vérification, puisque son traitement n'est pas sujet à condition (voir Annexe D).

➤ **Abandon du mode recouvrement :** Un paquet demeure en mode recouvrement tant qu'il n'a pas atteint un nœud dont la distance à la destination est inférieure à celle du nœud concave, C , incluse dans son entête (afin d'éviter les boucles de routage). Une fois qu'un tel noeud est atteint, le mode géographique est réactivé. Ainsi, chaque fois que le paquet est transmis à un nœud intermédiaire V :

- Si $DIST(V, D) < DIST(C, D)$, le nœud V change le mode d'acheminement dans l'entête du paquet de *recouvrement à géographique* avant de l'envoyer au prochain nœud transmetteur.
- Sinon, le mode d'acheminement du paquet reste inchangé.

Ainsi, sur la Figure 3.9, le mode recouvrement est conservé jusqu'à ce que le paquet atteigne le nœud A , car $DIST(A, D) < DIST(I, D)$. Par la suite, le routage géographique PARA est utilisé pour acheminer le paquet à sa destination D .

3.3.10.2 Mode topologique

Le routage topologique doit permettre de sortir d'un blocage au niveau d'un noeud concave lorsque le processus de recouvrement échoue ou n'est pas réalisable. Le paquet est alors routé en mode topologique jusqu'à sa destination.

➤ **Faiblesses du mode recouvrement :** Le mécanisme de recouvrement de PARA ne peut être utilisé dans les trois situations suivantes :

- Supposons que le noeud C reçoive un paquet de données en mode géographique destiné à D du noeud N . A la suite de sa découverte de proximité initiale pour D , le noeud C s'avère concave. Si l'indice de proximité de N inclus dans l'entête du paquet est égal à 1, le mode recouvrement n'est pas applicable. En effet, dans ce cas, le noeud N n'a à priori pas d'autres meilleurs voisins que le noeud concave C auquel il a déjà transmis le paquet. Le noeud C va donc initier le passage en mode topologique du paquet.
- La procédure de recouvrement ne peut être utilisée que pour un paquet en mode géographique. Si un paquet est acheminé en mode recouvrement et qu'un noeud intermédiaire C est concave pour la destination de ce paquet, alors le mode d'acheminement du paquet est changé à *topologique* au niveau du noeud intermédiaire.
- Si un noeud génère un paquet de données pour le noeud D et qu'il s'avère concave à l'issu de sa découverte de proximité, il utilise un routage topologique pour acheminer le paquet à sa destination.

➤ **Passage en mode topologique :** Le protocole de routage topologique AODV est utilisé pour sortir de l'optimum local atteint par PARA. Ce protocole a été décrit dans la Section 2.1.2. Notre choix s'est porté sur AODV en raison de sa résistance aux boucles de routage et son trafic de contrôle relativement faible par rapport à certaines approches de routage réactif telle que DSR. D'autre part, AODV réagit

relativement rapidement aux ruptures de liens et ne met à jour que les nœuds affectés par ce bri. Pour initier le passage en mode topologique d'un paquet de données destiné à D , un nœud V :

- Change le mode d'acheminement dans l'entête PARA du paquet à *topologique*;
- Diffuse des requêtes de routes ou RRQs pour la destination du paquet.

Une fois la route établie entre le nœud intermédiaire et la destination, le paquet, ainsi que ceux reçus ultérieurement par le nœud V pour cette destination, sont acheminés via cette route et ce, tant que le nœud V est concave. Lorsqu'il reçoit un paquet de données, un noeud intermédiaire I utilise donc, selon le mode d'acheminement du paquet, sa table de meilleurs voisins, ou sa table de routage AODV, s'il y a lieu. Cependant, si le nœud I est inactif lorsqu'il reçoit un paquet de données en mode topologique, il utilise uniquement les mécanismes du protocole AODV pour transmettre le paquet.

Parallèlement, les découvertes périodiques permettent au nœud V , initiateur du mode topologique, de vérifier s'il est toujours concave pour cette destination. Si ce n'est plus le cas, les paquets de données sont à nouveau routés avec PARA.

La Figure 3.10 illustre les changements de mode de routage de notre algorithme. S est le nœud ayant transmis le paquet au nœud N . Le processus global de routage a été simplifié pour ne mettre en évidence que les changements de mode d'acheminement de PARA.

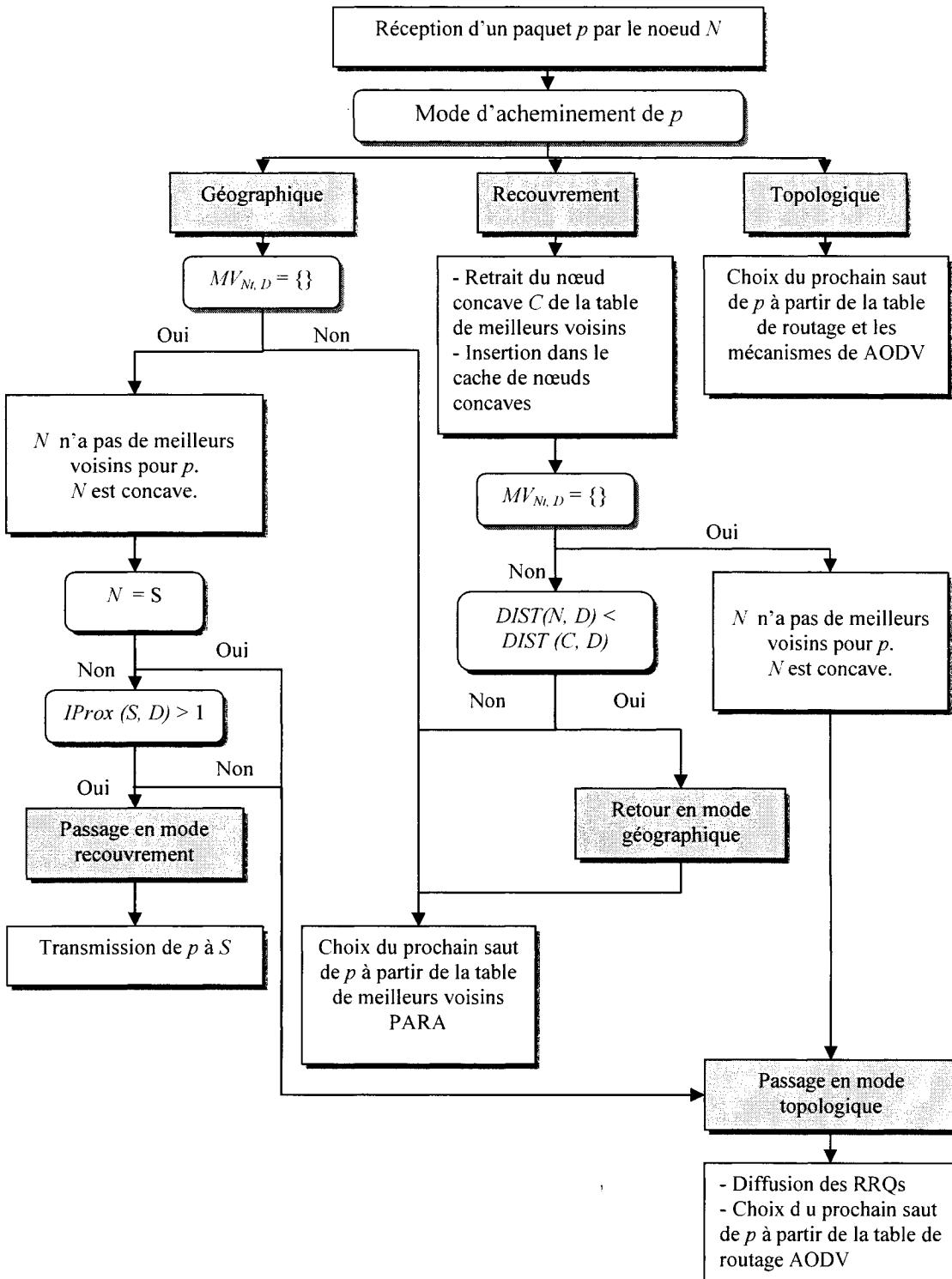


Figure 3.10 : Changements de mode de routage

3.4 ANALYSE MATHÉMATIQUE

PARA se veut un compromis entre un protocole de routage topologique réactif, tel qu'AODV, et un protocole de routage géographique comme GPSR. Dans cette section, nous analysons mathématiquement parlant, la charge engendrée par le trafic de contrôle par PARA, AODV et GPSR, notée respectivement L_{PARA} , L_{AODV} et L_{GPSR} .

3.4.1 Modèle du réseau

Nous supposerons dans le cadre de cette analyse que les noeuds mobiles sont uniformément distribués avec une densité de nœuds δ au sein d'une aire A . Nous assumerons aussi que le réseau compte N noeuds. La surface de transmission R d'un nœud est $2\pi r^2$, où r est le rayon de transmission des nœuds. La connectivité moyenne φ de chaque nœud est en conséquence donnée par RN/A , soit $R\delta$. Nous supposerons d'autre part que la longueur moyenne des routes (en terme de sauts) est L_r . Par ailleurs, le nombre de connexions actives par nœud sera noté A_c et S est le nombre de sources dans le réseau. Ces paramètres communs sont résumés dans le Tableau 3.2.

3.4.2 GPSR

Les seuls paquets de contrôle utilisés par GPSR sont les *beacons*, diffusés de manière périodique par tous les nœuds N du réseau, afin de communiquer leur position géographique à leurs voisins. Ce trafic représente l'équivalent d'un trafic proactif dans la mesure où il est permanent qu'il y ait des transmissions ou non. On a donc :

$$L_{\text{GPSR}} = TN \quad (3.3)$$

3.4.3 AODV

AODV, dans sa forme de base, génère quatre types de paquets de contrôle : les HELLOS, les requêtes de routes ou RRQs, les réponses de routes ou RRP, et les paquets d'erreur de route, RERRs. Les paramètres spécifiques à AODV sont présentés dans le Tableau 3.3.

Tableau 3.2 : Paramètres globaux utilisés pour l'analyse numérique

N	Nombre de nœuds du réseau
Δ	Densité du réseau
A	Aire sur laquelle le réseau est réparti
Φ	Connectivité moyenne des nœuds
S	Nombre de sources dans le réseau
T	Taux de diffusion de paquets HELLO pour AODV, <i>beacons</i> pour GPSR et taux d'exécution de découvertes périodiques de proximité pour PARA
μ	Taux de rupture des liens
L_R	Longueur moyenne d'une route
A_c	Nombre de connections actives par nœud
$LGPSR$	Nombre total de paquets de contrôle générés par GPSR
$LAODV$	Nombre total de paquets de contrôle générés par AODV
$L PARA$	Nombre total de paquets de contrôle générés par PARA

Tableau 3.3: Paramètres utilisés pour l'analyse numérique de AODV

L_e	Longueur moyenne d'une route utilisée de la source au nœud où le lien est brisé.
L_{HELLO}	Nombre de paquets HELLO générés par AODV
L_{RRQ}	Nombre de paquets RRQ générés par AODV
L_{RRP}	Nombre de paquets RRP générés par AODV
L_{RERR}	Nombre de paquets RERR générés par AODV

Trafic induit par les HELLOS

AODV utilise un mécanisme d'échange de paquets HELLOS pour évaluer la connectivité locale des nœuds. Le trafic conséquent est donc τN paquets par seconde.

$$L_{HELLO} = TN \quad (3.4)$$

Trafic induit par les RRQs

Les RRQs sont générées par les sources et diffusées par inondation dans tout le réseau. Si l'on suppose que chacune des S sources du réseau diffuse une RRQ toutes les λ_s

secondes, on alors un trafic de $SN\lambda_s$. La fréquence des découvertes des routes est liée aux ruptures de liens par $\lambda_s = \mu L_R$. Le trafic induit par les RRQs est donc, en définitive :

$$L_{RRQ} = \mu L_R SN \quad (3.5)$$

Trafic induit par les RRP

Les RRP cheminent le long des L_R sauts de la destination à la source de la RRQ. Comme le taux de RRP est le même que le taux d'envoi de RRQs, le trafic lié aux RRP est :

$$L_{RRP} = \mu L_R^2 \quad (3.6)$$

Trafic induit par les RERR

Lorsqu'un lien est brisé, AODV produit un paquet d'erreur qui est renvoyé à la source pour signaler la rupture du lien. Rappelons que L_e est la longueur du chemin à partir du lien brisé jusqu'à la source ($L_e < L_R$). Le paquet d'erreur devant traverser L_e liens pour atteindre la source, L_e paquets d'erreurs sont produits par route brisée. Pour chaque nœud, le taux de rupture des routes actives est μA_c . Dans un réseau de N nœuds, le nombre de paquets par seconde, dus aux erreurs de routes est $\mu L_e A_c N$. D'où :

$$L_{RERR} = \mu L_e A_c N \quad (3.7)$$

Finalement, le trafic de contrôle total généré par AODV est :

$$\begin{aligned} L_{AODV} &= L_{HELLO} + L_{RRQ} + L_{RRP} + L_{RERR} \\ &= \tau N + \mu L_R SN + \mu L_R^2 + \mu L_e A_c N \end{aligned} \quad (3.8)$$

Ce résultat confirme le fait que le trafic de contrôle de AODV est proportionnel à la mobilité des nœuds, au nombre total de nœuds du réseau, ainsi qu'au nombre de noeuds actifs.

3.4.4 PARA

PARA génère 5 types de paquets : les requêtes de proximités de niveau 1 et 2 (respectivement PREQ1, PREQ2, et PREQ2_ONE) et les réponses de proximité de niveau 1 et 2 (PREP1 et PREP2). Le Tableau 3.4 présente les paramètres spécifiques que nous utiliserons pour analyser le trafic induit par PARA. Nous nous contenterons ici

d'analyser le trafic de contrôle lié à PARA dans un contexte où tous les paquets sont acheminés en mode géographique.

Tableau 3.4 : Paramètres utilisés pour l'analyse numérique de PARA

P	Probabilité qu'un nœud constitue un meilleur voisin pour un autre nœud pour une destination quelconque
p_r	La probabilité qu'à la réception d'une PREQ2, un nœud ait déjà reçu une PREQ2 pour cette même découverte.
N	Nombre de meilleurs voisins d'un nœud
$Ldiscv$	Le nombre de paquets de contrôle générés lors d'une découverte de proximité
$Linit$	Le nombre total de paquets de contrôle générés par l'ensemble des nœuds du réseau lors de leurs découvertes de proximité initiales
$Lperiod$	Le nombre total de paquets de contrôle générés par l'ensemble des nœuds du réseau lors de leurs découvertes de proximité périodiques
$Lreq1$	Nombre de PREQ1 générées lors d'une découverte de proximité
$Lreq2$	Nombre de PREQ2 générées lors d'une découverte de proximité
$Lprep1$	Nombre de PREP1s générées lors d'une découverte de proximité
$Lprep2$	Nombre de PREP2s générées lors d'une découverte de proximité

Charge induite par une découverte de proximité

Etudions le trafic induit par chaque type de paquet de contrôle par chaque nœud lors d'une découverte de proximité.

Dans un premier temps, le nœud initiateur diffuse une PREQ1.

$$Lreq1 = 1 \quad (3.9)$$

Par la suite, chacun des meilleurs voisins du nœud initiateur diffuse, à son tour, une PREQ2. Le nombre de PREQ2s diffusées est donc équivalent au nombre de meilleurs voisins du noeud initiateur :

$$Lreq2 = n = p\varphi \quad (3.10)$$

Lorsqu'un voisin d'un de ces meilleurs voisins reçoit pour la première fois une PREQ2, il diffuse une PREP2. Les meilleurs voisins, toutefois, ne traitent pas ces PREQ2. Le trafic lié à la diffusion des PREP2s est donc donné par :

$$L_{prep2} = (p\varphi^2)(1-p_r) - n = (p\varphi^2)(1-p_r) - p\varphi \quad (3.11)$$

Enfin, à l'issu de la phase 2 de la découverte de proximité, chaque meilleur voisin diffuse une PREP1, s'il n'est toutefois pas concave. La probabilité p_c qu'un meilleur voisin soit concave est la probabilité p_{c1} que son nombre de meilleurs voisins soit nul, ajoutée à la probabilité p_{c2} que ses meilleurs voisins, s'il en a, soient eux-mêmes concaves:

$$p_{c1} = (1-p)\varphi \quad (3.12)$$

$$p_{c2} = (1-p)n = (1-p)p\varphi \quad (3.13)$$

$$\begin{aligned} p_c &= p_{c1} + p_{c2} \\ &= \varphi(1-p)(1+p) \end{aligned} \quad (3.14)$$

Finalement, le trafic lié aux PREP1s est obtenu par :

$$L_{prep1} = (1 - p_c)n = p\varphi(1 - \varphi(1-p)(1+p)) \quad (3.15)$$

En définitive, le trafic de contrôle généré par une découverte de proximité est obtenu par :

$$\begin{aligned} L_{discv} &= L_{req1} + L_{req2} + L_{prep2} + L_{prep1} \\ &= 1 + p\varphi + (p\varphi^2)(1-p_r) - p\varphi + p\varphi(1 - \varphi(1-p)(1+p)) \\ &= 1 + (p\varphi^2)(1-p_r) + p\varphi(1 - \varphi(1-p)(1+p)) \end{aligned} \quad (3.16)$$

Charge induite par les découvertes de proximité initiales

Les SL_R noeuds actifs du réseau initient chacune de leur transmission avec leur A_c destinations actives en lançant une découverte de proximité initiale. Le trafic correspondant est donné par :

$$L_{discv_init} = SL_R A_c L_{discv} \quad (3.17)$$

Charge induite par les découvertes de proximité périodiques

Sachant que chacun des SL_R noeuds actifs du réseau lancera une découverte de proximité périodique toutes les T secondes, le trafic généré par les découvertes de proximité périodique est :

$$Ldiscv_perio = TL_R Ldiscv \quad (3.18)$$

Charge totale de PARA

Le trafic de contrôle total généré par PARA, dans un contexte favorable à l'acheminement géographique des paquets est obtenu avec:

$$\begin{aligned} L_{PARA} &= Ldiscv_init + Ldiscv_perio \\ &= TL_R Ldiscv + L_R A_c Ldiscv \\ &= L_R (T + A_c) (1 + (p\varphi^2)(1-p_r) + p\varphi (1 - \varphi (1-p)) (1+p)) \end{aligned} \quad (3.19)$$

3.4.5 Ebauche de résultats

Les performances de PARA en terme de paquets de contrôle sont évidemment influencées par certains paramètres de la topologie du réseau.

Au niveau local d'une découverte de proximité, le nombre de paquets échangés est directement proportionnel à la connectivité des nœuds. Ce paramètre a en effet une incidence certaine sur la proportion de meilleurs voisins d'un nœud. Dans un réseau ad hoc dense, le trafic induit par une découverte de proximité individuelle devrait donc s'avérer relativement important. Toutefois, PARA ne s'adresse pas à de tels réseaux, dans lesquels l'algorithme d'acheminement *greedy* est de toute manière particulièrement performant.

Par ailleurs, le trafic de contrôle généré par PARA est, contrairement à GPSR et AODV, indépendant de la taille du réseau ad hoc. La charge de contrôle, à l'échelle globale du réseau, est uniquement déterminée par la proportion de nœuds actifs du réseau, soit par le nombre de connections et la longueur des routes. La mobilité des nœuds n'intervient pas directement sur cette métrique.

L'intervalle des découvertes périodiques et des *beacons*, joue aussi un rôle critique. En effet, la validité des informations stockées est liée à la fréquence des mises à jour et est donc obtenue au détriment du trafic de contrôle.

Certes, cette analyse se limite au cas idéal où tous les paquets sont acheminés en mode géographique mais, PARA anticipant la présence des nœuds concaves, les passages en

mode topologique devraient être de moindre importance. PARA devrait donc s'avérer particulièrement performant :

- dans un réseau à faible connectivité, par rapport à GPSR;
- dans un réseau à haute mobilité, par rapport à AODV.

En définitive, le principal inconvénient de PARA se situe au niveau de la latence induite par les découvertes de proximité. Il ne faut pas s'attendre à ce que PARA surpassse AODV de ce point de vue.

Dans le Chapitre 4, l'analyse de performance de PARA à travers des simulations va nous permettre de conforter cette ébauche de résultats.

Chapitre IV

Implémentation et analyse de performance

Nous avons, dans le chapitre précédent, détaillé le protocole de routage PARA. L'objectif du présent chapitre est d'évaluer dans quelle mesure PARA atteint les objectifs fixés. Pour des raisons évidentes de faisabilité et de coût, il est difficile de valider un protocole de routage dans un contexte réel. Afin d'évaluer la performance de PARA par rapport à d'autres protocoles de routage, nous nous contenterons donc d'effectuer un certain nombre de simulations dans diverses topologies de réseaux. Pour ce faire, nous utiliserons NS, *Network Simulator*, un outil logiciel de simulation de réseaux informatiques. Après avoir introduit ce simulateur et ses principales caractéristiques, nous décrirons la mise en oeuvre de PARA dans ce simulateur. Finalement, nous présenterons le plan d'expérience de notre analyse de performance, les résultats obtenus, ainsi que les interprétations qui en découlent.

4.1 LE SIMULATEUR NS

NS, *Network Simulator*, est un simulateur à événements discrets orienté objet pour les réseaux informatiques. Initialement développé à Berkeley pour la recherche dans l'environnement Internet, il permet maintenant de simuler une grande variété de réseaux, dont les réseaux sans-fil. Bien que l'usage de ce simulateur soit largement répandu dans les milieux de recherche, notre choix se justifie principalement par la disponibilité du code du protocole GPSR pour ce simulateur. NS est, de plus, gratuit et libre, facilitant l'extension du code existant selon nos besoins. Pour plus de détails concernant l'implémentation de NS, veuillez consulter l'Annexe J.

4.2 IMPLÉMENTATION DE PARA

L'implémentation de PARA dans NS a nécessité l'ajout de modules au code existant du simulateur. Certains choix quant à la mise en œuvre de PARA ont par ailleurs du être effectués.

4.2.1 Classes implémentées pour le routage PARA

Les classes implémentées pour le routage PARA, ainsi que leurs interactions avec les classes existantes de NS, sont présentées dans l'Annexe K.

4.2.2 Entête des paquets

La structure *hdr_para* définit l'entête PARA générale, partagée par tous les formats de paquets, de manière à faciliter leur traitement en C++. Le champ *ph_type* identifie le type de l'entête. La Figure 4.1 illustre le format de cette structure.

```
Struct hdr_para {
    u_int8_t    ph_type;
    static int  offset_;
    inline static int& offset() { return offset_; }
    inline static hdr_para* access(const Packet* p) {
        return (hdr_para*) p->access(offset_);
    }
};
```

Figure 4.1: Structure *hdr_para*

Les champs de l'entête de chaque format de paquet sont par la suite définis dans une structure spécifique. Il est à noter qu'il n'existe pas de relation directe avec la taille réelle du paquet simulé. Celle-ci est en effet explicitement calculée dans la fonction *hdr_size()* de chaque structure d'entête. Ainsi, la structure *hdr_para_preq1* décrit les champs d'une requête de proximité de niveau 1 (voir Figure annexée L.1).

Les informations sur les destinations actives du nœud émetteur sont représentées par un tableau de type *hdr_dst_preq1* qui encapsule l'adresse et les coordonnées de la destination, ainsi que la distance du nœud émetteur à cette destination. Le nombre maximal de destinations, *MAX_DST*, est fixé à 40. De la même manière, la structure

hdr_para_preq2 décrit les champs de l'entête d'une requête de proximité de niveau 2. La principale différence avec la structure *hdr_para_preq1* tient au fait que le tableau de destinations est de type *hdr_dst_preq2* et ne contient pas d'informations concernant la distance du noeud émetteur aux destinations.

Le concept est le même pour les entêtes des PREP2s et des PREP1s, décrites respectivement par les structures *hdr_para_prep2* et *hdr_para_prep1*. A titre d'exemple, la Figure annexée L.2 illustre le code de la structure *hdr_para_prep1*.

Enfin, les champs de l'entête PARA ajoutée aux paquets de données sont spécifiés par la structure *hdr_para_data* (voir Figure annexée L.3).

4.2.3 Implémentation des temporisateurs

Comme nous l'avons décrit dans le Chapitre 3, les découvertes de proximité sont soumises à des contraintes temporelles. La mise en œuvre de ces contraintes dans l'implémentation de PARA passe par la définition de temporisateurs. Ces temporisateurs sont implémentés par diverses classes faisant appel à l'ordonnanceur de NS.

La classe *PRequestTimer* hérite de la classe *TimerHandler* de NS et définit le temporisateur visant à assurer le lancement périodique des découvertes de proximité. L'intervalle des découvertes est fixé par le paramètre *prequest_interval*. Ce temporisateur est initialisé dans *ParaAgent:recv()* lorsqu'un noeud devient actif. Si le nœud n'est plus actif, le temporisateur est annulé. La Figure 4.2 illustre la déclaration de la classe *PRequestTimer*.

```
class PRequestTimer2 : public TimerHandler {
public:
    PRequestTimer2(Para2_Agent *a) : TimerHandler() {agent = a; }
    void expire(Event* );
private:
    Para2_Agent    *agent;
};
```

Figure 4.2 : Classe *PrequestTimer*

La classe *ForwardTimer* hérite elle aussi de la classe *TimerHandler* et introduit le délai *discv_delay* alloué à chaque découverte de proximité pour parcourir la table de meilleurs voisins et supprimer les entrées non valides. Ce temporisateur est initialisé à chaque début de découverte de proximité périodique.

La classe *ReplyTimer* implémente le temporisateur retardant l'envoi d'une PREP1 par un noeud ayant reçu une PREQ1 d'un délai défini par *reply_delay*.

Le temporisateur implémenté par la classe *InitDiscvTimer* permet quant à lui de retarder l'envoi des paquets de données destinés à un nœud pour lequel une découverte initiale est en cours. Ce temporisateur est initialisé dans *ParaAgent:recv()* lorsqu'une découverte initiale est lancée. Lorsque le délai *discv_delay* expire, la découverte initiale est terminée et les paquets sont envoyés.

D'autres temporiseurs entrent en jeu dans la mis en oeuvre de PARA. Ces derniers sont implémentés par les classes *ConcavTimer* et *Preq2Timer* destinés respectivement à assurer la validité des entrées du cache de nœuds concaves et du cache de PREQ2. Lorsque le délai de validité de l'entrée expire, l'entrée correspondante est supprimée.

4.2.4 Extensions apportées à l'interface Tcl de NS

Dans NS, l'exécution d'une simulation est généralement contrôlée par un script de simulation écrit en Tcl. En plus des actions communes, telles que la spécification du scénario de simulation ou le lancement même du simulateur, il est possible d'exporter des fonctions d'un agent de routage (écrit en C++) à un environnement Tcl. Ce mécanisme est utilisé lors de la simulation de PARA, d'une part, pour initialiser le service de localisation, dans *LocDBase:command()*, et d'autre part, pour enregistrer un pointeur sur le service de localisation dans chaque agent de routage PARA, dans *ParaAgent:command()*. La Figure 4.3 montre par exemple le code C++ implémentant la fonction Tcl *ldb* de l'agent PARA.

```

Int Para_Agent::command(int argc, const char*const* argv) {
[...]
if (strcmp(argv[1], "ldb") == 0) {
    if ((obj = TclObject::lookup(argv[2])) == 0) {
        trace( "%s: %s lookup of %s failed\n", __FILE__, argv[1],
               argv[2]);
        return TCL_ERROR;
    }
    ldb_ = (LocDbase *) obj;
    return TCL_OK;
}
}

```

Figure 4.3: Extension apportée à l’interface Tcl

4.2.5 Choix significatifs

Nous avons été amenés, lors de la mise en œuvre de PARA à faire un certain nombre de choix dans notre implémentation :

- **Notifications de rupture de lien par la couche Mac**

Avant tout, nous avons choisi de prendre en compte les ruptures de liens détectées au niveau de la couche Mac. Tout comme GPSR, nous recevons donc une notification de la couche Mac 802.11 lorsqu’un paquet excède son nombre maximal de retransmissions. En effet, si les échecs de transmission dépassent ce nombre, c’est que le noeud destinataire a quitté le rayon de transmission du nœud. L’utilisation de ces notifications permet à PARA de détecter le départ de meilleurs voisins plus rapidement qu’à travers l’expiration du temps de validité d’un meilleur voisin ($3 * \text{Prequest_interval}$).

D’autre part, nous avons implémenté une fonctionnalité permettant de traverser la file d’attente de l’interface. Ce mécanisme va de pair avec les notifications de la couche mac et a un effet marqué sur nos résultats, comme le montre la Figure 4.5. Lorsque l’interface tente de façon répétée de retransmettre un paquet situé en tête de sa file, la tête de file est bloquée dans l’attente d’une confirmation (ACK) du nœud récepteur via la couche liaison. Le blocage de la file d’attente réduit de façon significative la capacité de transmission de l’interface. Pour cette raison, lorsque nous recevons une notification

d'échec de retransmission d'un paquet de données de la couche Mac, nous agissons en fonction du mode d'acheminement du paquet:

- Si le paquet est en mode géographique ou recouvrement,
 - le meilleur voisin devenu inaccessible est supprimé de la table de meilleurs voisins;
 - le paquet est passé à l'agent de routage Para qui assurera son envoi au prochain nœud transmetteur choisi à partir de la table de meilleurs voisins ainsi mise à jour;
 - la file d'attente de l'interface est parcourue, et tous les paquets dont le prochain saut était le nœud inaccessible sont retransmis de la même manière au prochain nœud transmetteur.
- Si le paquet est en mode topologique, le mécanisme traditionnellement mis en œuvre par AODV est appliqué : si la route est active, le nœud détectant la rupture de lien, émet un paquet *route error* (RERR) vers la source. Après réception de la RERR, si le lien rompu est plus proche (en terme de sauts) de la source que de la destination, la source lance un nouveau processus de découverte de route. Sinon, le paquet est abandonné.

Pour vérifier l'impact de ce choix sur nos résultats, nous avons simulé un réseau ad hoc de 50 nœuds dont les mouvements sont régis par différents temps de pause. La durée de ces simulations est de 200 secondes, durant lesquelles 10 connexions sont établies. Pour chaque topologie, 5 itérations ont été réalisées. La Figure 4.4 illustre le taux de réception des paquets de données en fonction du temps de pause des nœuds selon que le mécanisme de notification de la couche Mac est utilisé ou non. L'implémentation de PARA avec notification de ruptures de lien (PARA-LL) permet de délivrer jusqu'à 35% plus de paquets que l'implémentation dépourvue de cette fonctionnalité (PARA) lorsque la mobilité des nœuds est maximale. Il va de soi que la notification de ruptures de lien ne permet un gain de performance que lorsque les ruptures de lien sont fréquentes, c'est-à-dire dans un contexte où la mobilité des nœuds est élevée. Ainsi, lorsque le temps de pause des nœuds atteint 80s, les performances des deux implémentations sont

comparables. Les résultats obtenus parlent donc d'eux-mêmes et justifient ce choix d'implémentation.

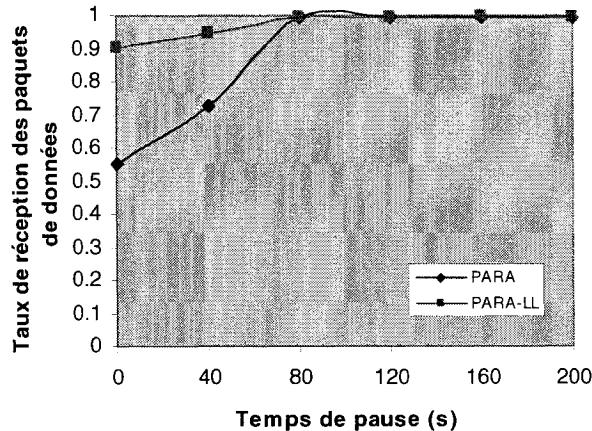


Figure 4.4 : Impact de la notification de la couche Mac sur le taux de réception des paquets de données

➤ Choix des gagues et délais

Dans le contexte d'une découverte de proximité, les nœuds peuvent être amenés à diffuser une PREQ2, une PREP2 et une PREP1. Pour éviter la diffusion simultanée de ces paquets, et donc limiter le risque de collision, nous introduisons un facteur temporel aléatoire lors de la diffusion de chaque type de paquets de contrôle. Pour ce faire, nous utilisons une variable aléatoire uniforme, dont la valeur maximale est fixée par les paramètres *jitter_preq2*, *jitter_prep2* et *jitter_prep1*, correspondant respectivement au délai aléatoire introduit lors de la diffusion d'une PREQ2, d'une PREP2 et d'une PREP1. La combinaison de ces 3 facteurs temporels a une forte influence sur nos résultats. Afin d'approcher la meilleure combinaison, nous avons effectué des simulations d'une durée de 200s sur un réseau de 50 nœuds statiques avec 30 connexions. Pour ces simulations, nous avons affecté des valeurs vraisemblablement élevées aux paramètres *discv_delay* et *reply_delay* afin qu'ils n'influencent pas les résultats, soit respectivement 120ms et 90ms. Les valeurs testées sont 0s, 0.005s et

0.010s. Par ailleurs, pour chaque combinaison, nous avons effectué une série de 6 simulations.

Après avoir déterminé la combinaison de valeur apportant les meilleurs résultats en terme de taux de livraison de paquets et de délai moyen, nous avons procédé de la même manière pour évaluer les délais à allouer à chaque phase de la découverte de proximité. En effet, le choix des valeurs des paramètres *discv_delay* et *reply_delay* a une influence certaine sur la performance de PARA, comme le montre l'Annexe M. Les valeurs que nous avons testées varient ici entre 0.3s et 0.13s (par intervalle de 10 ms). Nous avons effectué ces simulations dans un réseau de 50 nœuds statiques avec 30 connexions établies. Par ailleurs, cette série de tests a été réalisée en faisant varier la connectivité du réseau afin de s'assurer un choix de valeur pertinent dans tout type de réseau. Finalement, le Tableau 4.1 résume les choix que nous avons faits. Nous avons décidé d'opter pour les valeur assurant le meilleur taux de réception des paquets, quelle que soit la connectivité du réseau.

Tableau 4.1 : Valeurs de gîgues et délais

Paramètre	Valeur (secondes)
Discv_delay	0.07
Reply_delay	0.05
Jitter_preq2	0.01
Jitter_prep2	0.005
Jitter_prep1	0.005

➤ **Implémentation de ARP**

L'implémentation de base dans le simulateur NS du protocole de résolution d'adresse, ARP, entraîne l'abandon de nombreux paquets car, rappelons qu'un seul paquet peut être traité par destination. Ainsi, si un autre paquet arrive pour la même destination, le premier paquet est abandonné. Certes, l'utilisation de ARP dans les simulations semble plus réaliste, mais l'étude des performances des protocoles de

routage n'en est que plus délicate, car certains effets, notamment les pertes de paquets, ne sont pas directement liés aux protocoles de routage, mais au code du module ARP de NS. Nous avons donc décidé d'utiliser une autre version de ARP, qui effectue la conversion d'adresses immédiatement, sans envoi de requêtes et attente de réponse. Cette opération est possible, car dans NS, les adresses des noeuds sont les mêmes que les adresses physiques. Pour obtenir ce comportement, nous avons modifié la fonction *int ARPTable:: arpsolve()*. Cette modification garantie que les résultats obtenus sont uniquement le fait des protocoles de routage même.

4.3 EVALUATION DE PERFORMANCE DE PARA

Pour évaluer dans quelle mesure PARA atteint les objectifs escomptés, nous simulerons l'algorithme de routage dans diverses topologies de réseau ad hoc. Par ailleurs, afin de déterminer quantitativement les améliorations apportées par PARA aux autres approches de routage, nous simulerons aussi le protocole GPSR, pour l'aspect géographique, ainsi que le protocole réactif AODV, qui est largement reconnu pour le routage topologique. Après avoir présenté la configuration générale mise en œuvre pour les simulations, nous décrirons notre plan d'expérience. Finalement, nous exposerons et analyserons les résultats obtenus par PARA.

4.3.1 Environnement de simulation

Nous présentons dans cette section les différents paramètres entrant en jeu dans le cadre de la configuration de notre environnement de simulation.

4.3.1.1 Configuration globale

Nous avons réalisé les simulations sur la version 2.27 du simulateur NS avec les paramètres globaux présentés dans le Tableau 4.2.

4.3.1.2 Métriques de performance

Nous évaluons les protocoles de routage selon les cinq métriques suivantes :

- Le taux de livraison des paquets, soit le ratio du nombre de paquets de données délivrés aux destinations par le nombre de paquets générés par les sources.
- Le délai moyen de bout en bout. Cette métrique inclut tous les types de délais possibles causés par les files d'attente, les retransmissions par la couche MAC et la propagation des paquets. Dans le cas de PARA et AODV, cette métrique inclut aussi le délai causé par la mise en tampon des paquets durant une découverte de route, ou une découverte de proximité initiale.
- Le trafic de contrôle normalisé, qui correspond au nombre de paquets de contrôle transmis par paquet de données délivré aux destinations. Chaque saut est compté comme une transmission.
- La longueur des routes, soit le nombre moyen de sauts des paquets délivrés aux destinations.

Tableau 4.2 : Paramètres globaux de simulation

Paramètre	Valeur
Couche Mac	IEEE 802.11
Rayon de transmission	250m
Capacité du canal	2Mb/s
Modèle de propagation	<i>Two-ray</i>
Taille des paquets de données	64 bytes
Vitesse maximale des nœuds	20 m/s
Durée des simulations	900s

4.3.1.3 Modèle de mobilité

Les nœuds simulés en utilisant l'extension sans fil de NS se déplacent sur un plan sans obstacles. Le modèle de mobilité utilisé est *Random Waypoint*. Dans ce modèle, chaque nœud commence par être statique durant un temps de pause prédéfini au début

de la simulation. Une fois ce temps écoulé, chaque nœud choisit aléatoirement une destination dans la surface de simulation et se déplace vers elle avec une vitesse uniformément distribuée entre 0m/s et, dans notre cas, 20m/s. Lorsqu'il atteint sa destination, le nœud marque un temps de pause, puis sélectionne une autre destination, et le processus se répète ainsi durant toute la durée de la simulation.

4.3.1.4 Facteurs environnementaux

Divers facteurs vont influencer les performances des protocoles évalués. Nous faisons donc varier ces conditions expérimentales afin d'étudier leur impact sur nos résultats.

➤ Dimension du réseau

Le trafic de contrôle généré par AODV et GPSR est fonction de la dimension du réseau. Nous étudions donc dans un premier temps les performances des protocoles de routage dans des réseaux ad hoc de 50 noeuds. Par ailleurs, faute de temps, nous n'effectuerons que quelques simulations dans des réseaux de 100 nœuds.

➤ Degré de connectivité des nœuds

Dans la plupart des études basées sur des simulations de réseaux ad hoc, les nœuds sont connectés de manière dense. Ainsi dans [15], GPSR démontre des résultats particulièrement bons, car le mode périmètre n'est que très peu utilisé. En effet, dans un réseau hautement connecté, quasiment chaque nœud a un chemin vers tout autre nœud du réseau, généralement de quelque sauts. Parallèlement, le trafic de contrôle génère alors souvent de la congestion. Un réseau dispersé possède d'autres caractéristiques. Dans de tels réseaux, les chemins entre deux nœuds, d'autant plus si on utilise un acheminement géographique, n'existent pas toujours, et les choix de routage sont de toute évidence plus affectés par la mobilité des nœuds.

Afin d'étudier le comportement de PARA, nous effectuons des simulations dans les deux types de réseaux. Nous étendons la surface de simulation selon le nombre de

nœuds du réseau afin d'obtenir trois valeurs approximatives de connectivités, comme le montre le Tableau 4.3.

Pour obtenir ces diverses valeurs de connectivité, nous procémons comme suit. Sachant que chaque nœud du réseau a un rayon de transmission de 250m, nous commençons par évaluer la surface de transmission chaque nœud, $At = \pi * 250^2 \approx 196\ 349\ m^2$. Soient D , la densité du réseau, C , la connectivité du réseau, A , la surface du réseau et N , le nombre de nœuds du réseau. On a :

$$D = \frac{A}{N} \quad (4.1)$$

$$C = At \times D \quad (4.2)$$

Supposons que nous voulons une connectivité approximative de 5 nœuds. Selon (4.2), la densité des nœuds doit alors être de $2.56 \cdot 10^{-5}$ nœud / m^2 , ce qui équivaut à avoir $D = 1$ nœud par $39\ 000\ m^2$. On utilise ensuite (4.1) pour déterminer la surface du réseau permettant d'obtenir une telle connectivité sachant que le réseau compte 50 nœuds.

Tableau 4.3: Scénarios de connectivité

Nombre de nœuds	Région	Densité	Connectivité
50	1500m x 300m	1 nœud / 9000m ²	21
	1500m x 500m	1 nœud / 15000m ²	13
	1500m x 1300m	1 nœud / 39000 m ²	5
100	2250m x 1730m	1 nœud / 39000 m ²	5

➤ Degré de mobilité

Varier le degré de mobilité, ou la vitesse des nœuds dans le réseau, est un moyen efficace de tester le comportement d'un protocole de routage dans un environnement dynamique. Dans nos simulations, nous fixons la vitesse minimale des nœuds à 0m/s, et leur vitesse maximale à 20m/s. Nous faisons varier les temps de pause selon les valeurs suivantes : 0s, 40s, 80s, 120s, 160s, 200s et 300s.

➤ **Nombre de trafics de données**

Les protocoles réactifs ne sont actifs que lorsqu'une communication est nécessaire entre deux nœuds du réseau. Le nombre de trafic de données a donc une influence certaine sur le trafic de contrôle généré par AODV et PARA. Nous simulons un trafic CBR et faisons varier le nombre de connexions établies: 10, 20 et 30. Les connexions CBR sont initiées à des instants uniformément distribués entre 100 et 330 secondes après le début de la simulation.

4.3.1.5 Paramètres des protocoles

L'intervalle d'exécution des découvertes de proximité périodiques de PARA, *prequest_interval*, est susceptible d'influencer les performances du protocole. Cependant, afin de limiter le nombre de simulations à réaliser, nous avons décidé de nous limiter à une seule valeur pour ce paramètre : *prequest_interval* = 1.5s. De la même manière, pour GPSR, la période des beacons est elle aussi fixée à 1.5s. Par ailleurs, les valeurs des paramètres spécifiques à AODV sont celles recommandées par le RFC 3561 [26]. Les Tableaux 4.4 et 4.5 résument les valeurs des paramètres utilisés par PARA et GPSR.

Tableau 4.4: Paramètres de PARA

Paramètre	Valeur (s)
<i>Prequest_interval</i>	1.5
<i>Discv_delay</i>	0.07
<i>Reply_delay</i>	0.05
<i>Jitter_preq2</i>	0.01
<i>Jitter_prep2</i>	0.005
<i>Jitter_prep1</i>	0.005
Délai de validité des meilleurs voisins	4.5

Tableau 4.5 : Paramètres de GPSR

Paramètre	Valeur (s)
<i>Beacon_interval</i>	1.5
Beacons implicates	Oui
Mode perimeter	Oui

4.3.2 Résultats des simulations

Pour chaque scénario étudié, 6 topologies aléatoires sont générées. Elles se caractérisent par un même scénario de trafic, mais des scénarios de mobilité différents. Les résultats que nous discutons dans cette section représentent la moyenne des résultats obtenus pour les 6 topologies étudiées et ce, pour chaque protocole évalué. Nous décrivons donc dans un premier temps les résultats selon la densité du réseau et les métriques définies précédemment. Une interprétation détaillée des résultats obtenus sera présentée dans la section suivante.

4.3.2.1 Scénario 1 : Réseau ad hoc dispersé

Rappelons que nous avons utilisé 10, 20 et 30 sources dans nos expériences, tout en faisant varier le temps de pause des nœuds. Le but de ce scénario est de comparer PARA à AODV et GPSR dans un réseau relativement dispersé (avec une connectivité moyenne de 5 nœuds) pour évaluer la capacité de PARA à gérer et anticiper les nœuds concaves.

Taux de réception. La Figure 4.5 illustre le taux de réception des trois protocoles évalué en fonction du temps de pause avec divers nombres de sources. D'une part, les performances de PARA et AODV sont relativement proches. PARA offre le meilleur taux de réception pour 10 et 20 sources (Figure 4.5 (a) et (b)) et délivre jusqu'à 12% plus de paquets que AODV dans les cas de mobilité élevée. Pour un temps de pause de 300s, les différences sont atténuées. On note cependant qu'avec 30 sources (Figure 4.5 (c)), le taux de réception d'AODV est légèrement supérieur à celui de PARA.

D'autre part, dans tous les cas, la performance de PARA dépasse de manière significative celle de GPSR. Pour 10 connexions, PARA délivre en moyenne 20% plus de paquets que GPSR, tandis que pour 20 et 30 sources, PARA délivre environ deux fois plus de paquets que GPSR.

Si le taux de réception de AODV demeure plus ou moins constant lorsque le nombre de sources augmente, les performances de PARA et GPSR sont, quant à elles, altérées. Ainsi, avec 10 sources, PARA délivre en moyenne 75% des paquets contre 60% avec 30 sources.

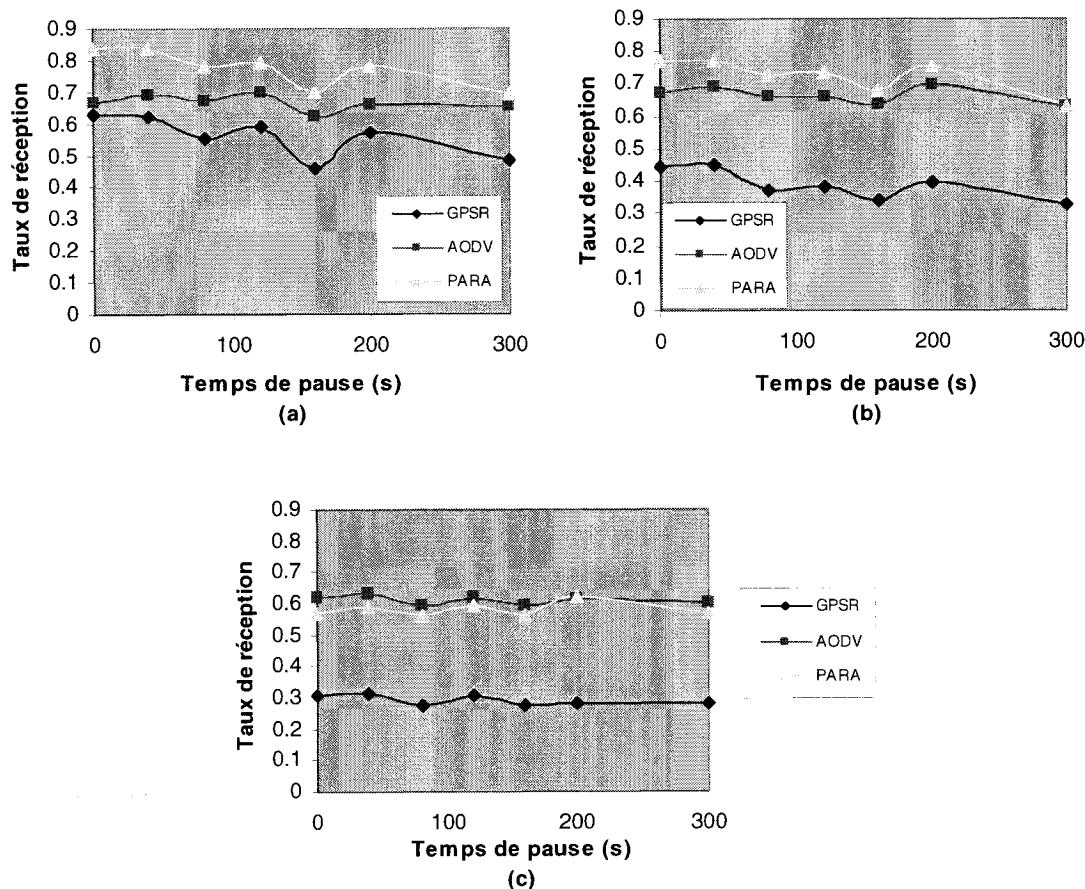


Figure 4.5 : Taux de réception des paquets de données dans un réseau de 50 nœuds dispersés (5 nœuds en moyenne) pour (a) 10, (b) 20 et (c) 30 connexions CBR

Trafic de contrôle normalisé. Dans tous les cas, GPSR surpassé AODV d'environ 50% (Figure 4.6). Par ailleurs, le trafic de contrôle de PARA est significativement supérieur à celui de GPSR et AODV. En effet, le trafic de PARA est 1.75 – 2 fois plus important que celui d'AODV, ce facteur augmentant avec le nombre de source.

Le trafic de contrôle de GPSR est constant quelle que soit la mobilité des nœuds. Le nombre de paquets de contrôle générés par PARA, quant à lui, augmente avec la mobilité et le nombre de sources. Le trafic de contrôle de AODV suit la même tendance, certes moins marquée, par rapport à la mobilité des nœuds, et diminue légèrement lorsque le nombre de sources augmente.

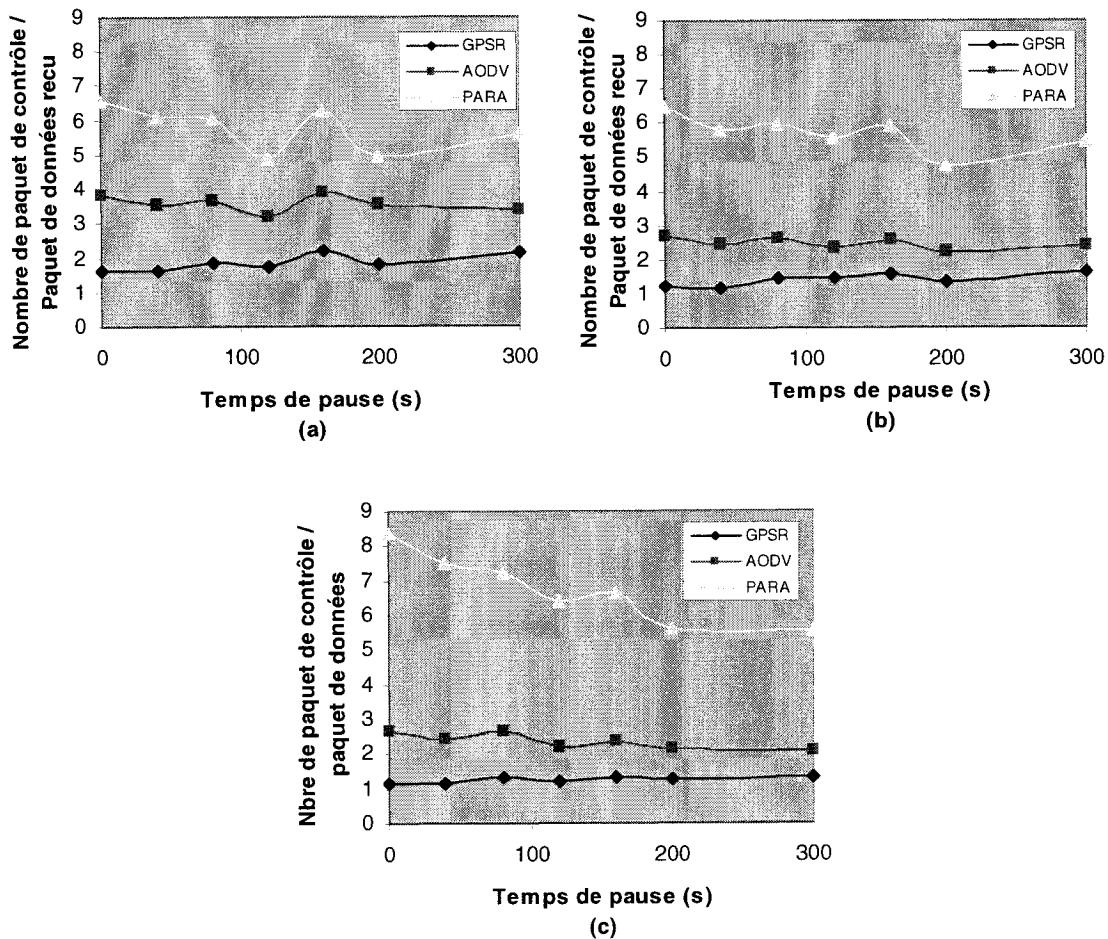


Figure 4.6 : Trafic de contrôle normalisé dans un réseau de 50 nœuds dispersés (5 nœuds en moyenne) pour (a) 10, (b) 20 et (c) 30 connexions CBR

Délai moyen de transmission. Dans tous les cas, AODV offre le meilleur délai de transmission (Figure 4.7). Pour 10 connexions, les performances des trois protocoles sont relativement similaires. PARA a un délai de transmission proche de celui de GPSR et supérieur à celui d'AODV d'environ 20 ms. Avec 20 et 30 sources, les différences de performance s'accentuent. PARA surpassé largement GPSR. Le délai de transmission moyen de GPSR est en fait alors environ 1.5-2 fois plus élevé que ce lui de PARA. Le délai de transmission moyen d'AODV est relativement constant, quelle que soit la mobilité, tandis que les performances de GPSR et PARA sont altérées.

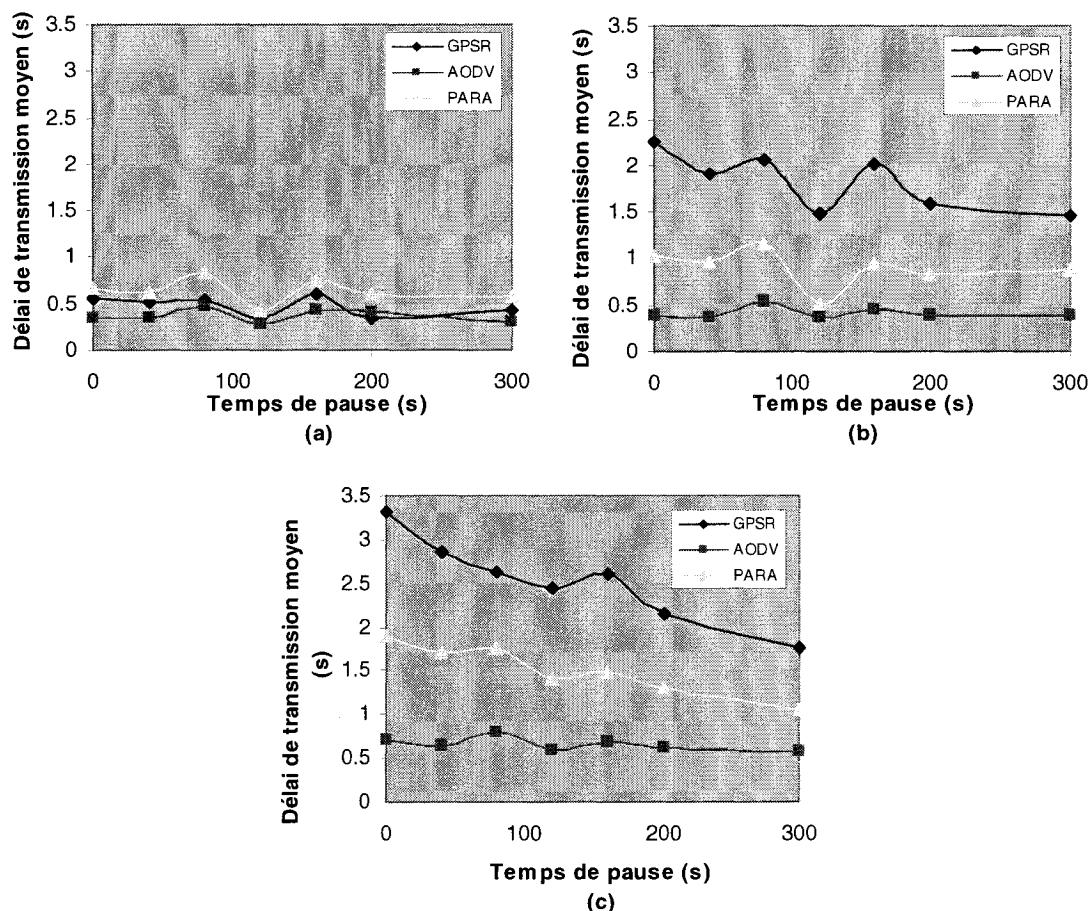


Figure 4.7 : Délai de transmission moyen dans un réseau de 50 nœuds dispersés (5 nœuds en moyenne) pour (a) 10, (b) 20 et (c) 30 connexions CBR

Longueur des routes. Avec 10 et 20 sources, la performance de PARA surpassé celle de GPSR de manière significative: les routes de GPSR sont alors environ 45% et 20 % respectivement plus longues que celles de PARA (Figure 4.8 (a) et (b)). La performance d'AODV est légèrement meilleure que celle de PARA.

Avec 30 sources, les résultats sont moins disparates (Figure 4.8 (c)). Les performances de GPSR et PARA sont similaires pour une mobilité élevée, mais avec un temps de pause de 300s, PARA surpassé légèrement GPSR. Par ailleurs avec 0s et 40s de temps de pause, les performances de PARA et GPSR surpassent celles d'AODV.

On remarque aussi que, quel que soit le protocole, le nombre moyen de sauts des routes atteint sa valeur maximale avec un temps de pause de 120s.

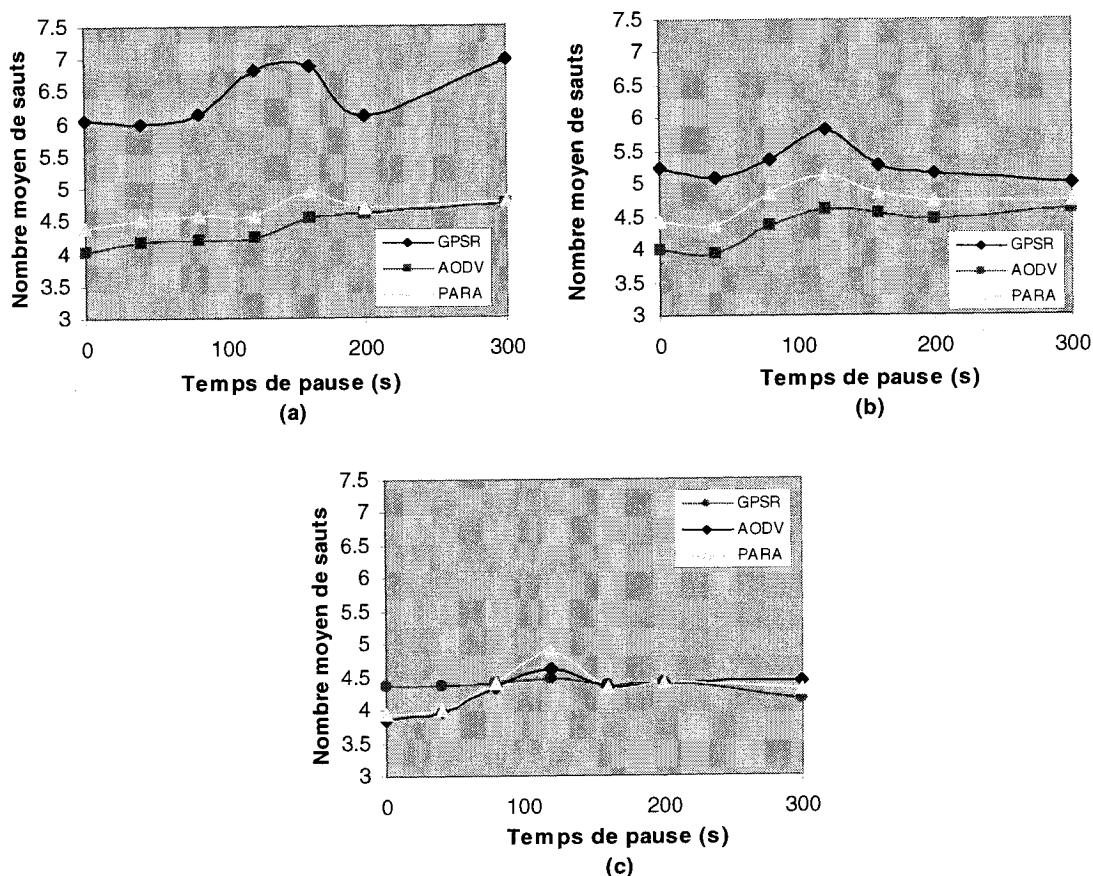


Figure 4.8 : Longueur moyenne des routes dans un réseau de 50 nœuds dispersés (5 nœuds en moyenne) pour (a) 10, (b) 20 et (c) 30 connexions CBR

4.3.2.2 Scénario 2 : Réseau ad hoc moyennement connecté (1500x500m)

Dans ce scénario, chaque nœud du réseau a environ 13 voisins immédiats. Nous qualifierons un tel réseau de moyennement connecté.

Taux de réception. Avec 10 sources, PARA délivre quasiment 100% des paquets et surpasse AODV et GPSR (Figure 4.9 (a)). La performance de GPSR se rapproche de celle de PARA, tandis qu'AODV ne délivre en moyenne que 85% des paquets. Par ailleurs, lorsque la mobilité diminue, le taux de réception de AODV augmente. PARA suit la même tendance, mais de manière moins significative. Avec 20 sources, les résultats sont plus mitigés, toutefois PARA domine en moyenne AODV et GPSR, bien que les différences soient légères (Figure 4.9 (b)). Avec 30 sources, la performance de PARA est nettement moins bonne que celle de GPSR et AODV (Figure 4.9 (c)). AODV délivre 10% plus de paquets en moyenne que GPSR. Lorsque la mobilité est élevée, GPSR surpasse PARA d'environ 10%, mais lorsque le temps de pause augmente, cet écart diminue. Finalement, avec un temps de pause de 300s, les performances sont équivalentes, car le taux de réception de PARA augmente avec la sédentarité des nœuds. Les performances de PARA et GPSR sont altérées lorsque le nombre de sources augmente, tandis que celle de AODV demeure équivalente.

Trafic de contrôle normalisé. Encore une fois, le trafic de contrôle de PARA est 2 fois plus important que celui généré par AODV. GPSR offre toujours la meilleure performance. Par ailleurs, le trafic de PARA est plus important que dans le scénario précédent. Le nombre de paquets de contrôles de PARA augmente donc comme escompté avec la connectivité et le nombre de sources.

Délai moyen de transmission. La Figure 4.10 montre que AODV offre la meilleure performance dans tous les cas. PARA a un délai de transmission supérieur à celui de AODV. La différence entre les performances de AODV et PARA augmente avec le nombre de sources. Pour 10 sources, lorsque la mobilité des nœuds est réduite, à partir

d'un temps de pause de 80s, la performance de PARA dépasse significativement celle de GPSR. Pour 20 et 30 sources, le délai de transmission moyen de GPSR est environ 2 fois supérieur en moyenne à celui de PARA. En outre, le délai moyen de transmission de PARA diminue avec la mobilité des nœuds.

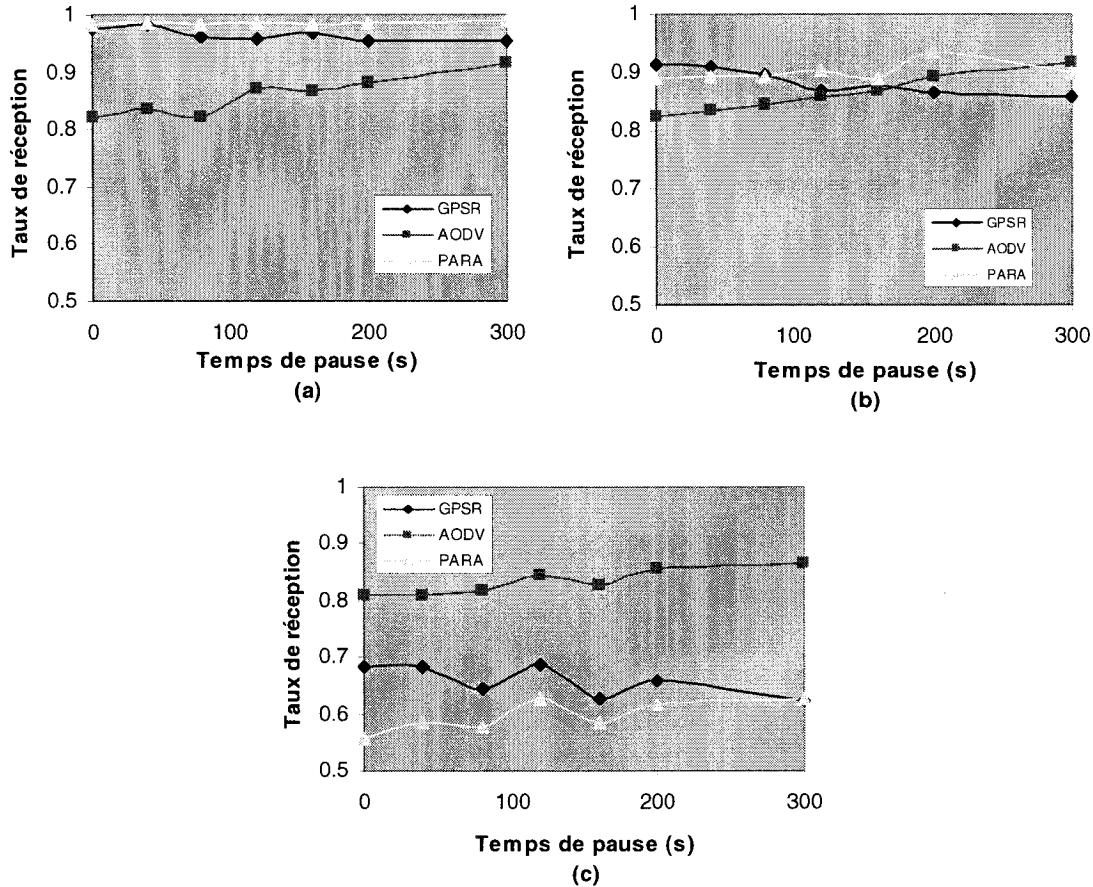


Figure 4.9 : Taux de réception des paquets de données dans un réseau de 50 nœuds moyennement connectés (13 nœuds en moyenne) pour (a) 10, (b) 20 et (c) 30 connexions CBR

Longueur des routes. Dans tous les cas, PARA domine GPSR (Figure 4.11). Avec 10 connexions, les performances de PARA et AODV sont relativement similaires. Toutefois, avec 20 sources, la longueur des routes de AODV est inférieure à celle de PARA. Avec 30 sources, la performance de PARA est meilleure que celle d'AODV.

Quel que soit le protocole, le nombre de sauts augmente avec la sédentarité des nœuds.

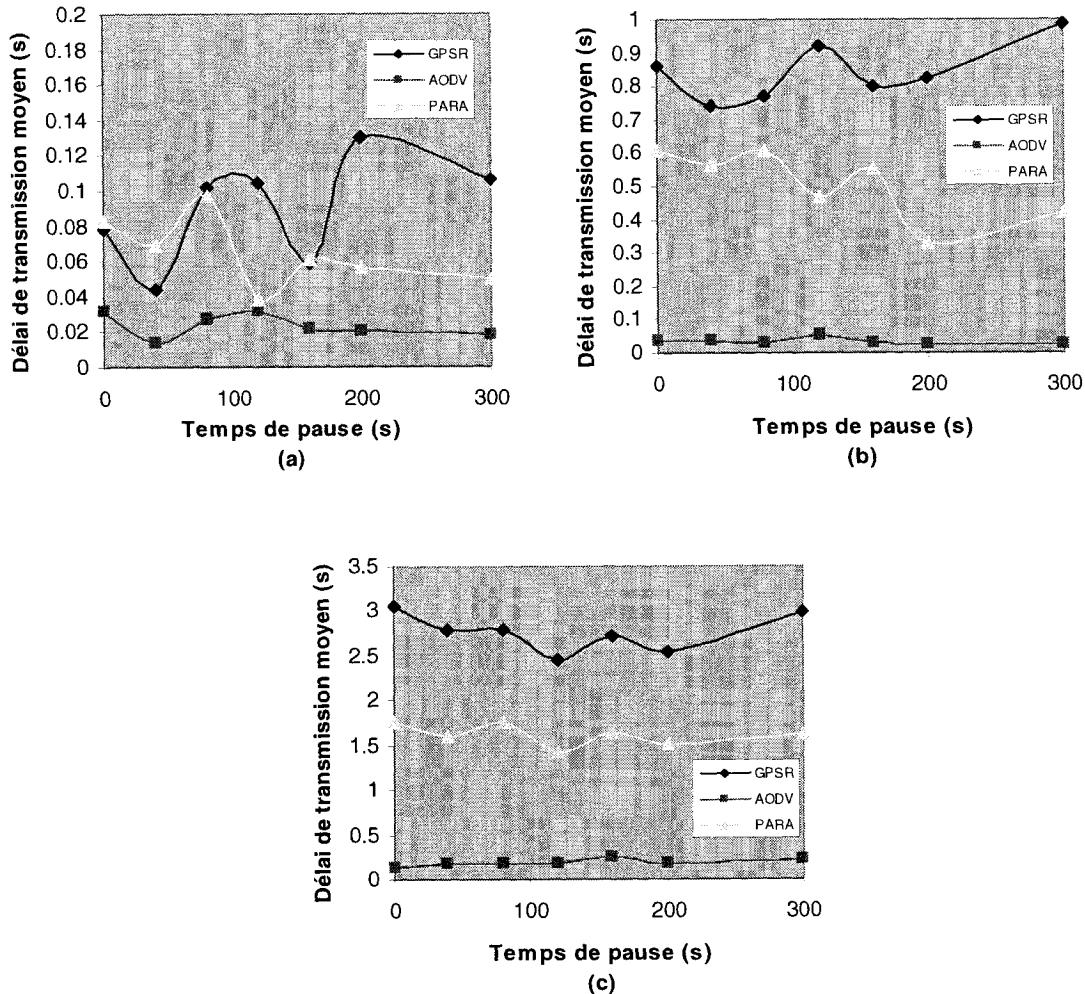


Figure 4.10 : Délai moyen de transmission dans un réseau de 50 nœuds moyennement connectés (13 nœuds en moyenne) pour (a) 10, (b) 20 et (c) 30 connexions CBR

4.3.2.3 Scénario 3 : réseau ad hoc dense (1500x300m)

Dans ce scénario, chaque nœud a une connectivité moyenne de 20 noeuds. Dans un tel réseau, quasiment chaque nœud a au moins un chemin vers tout autre nœud du réseau, généralement à quelques sauts. Le trafic de contrôle est alors en général important et le réseau est souvent congestionné.

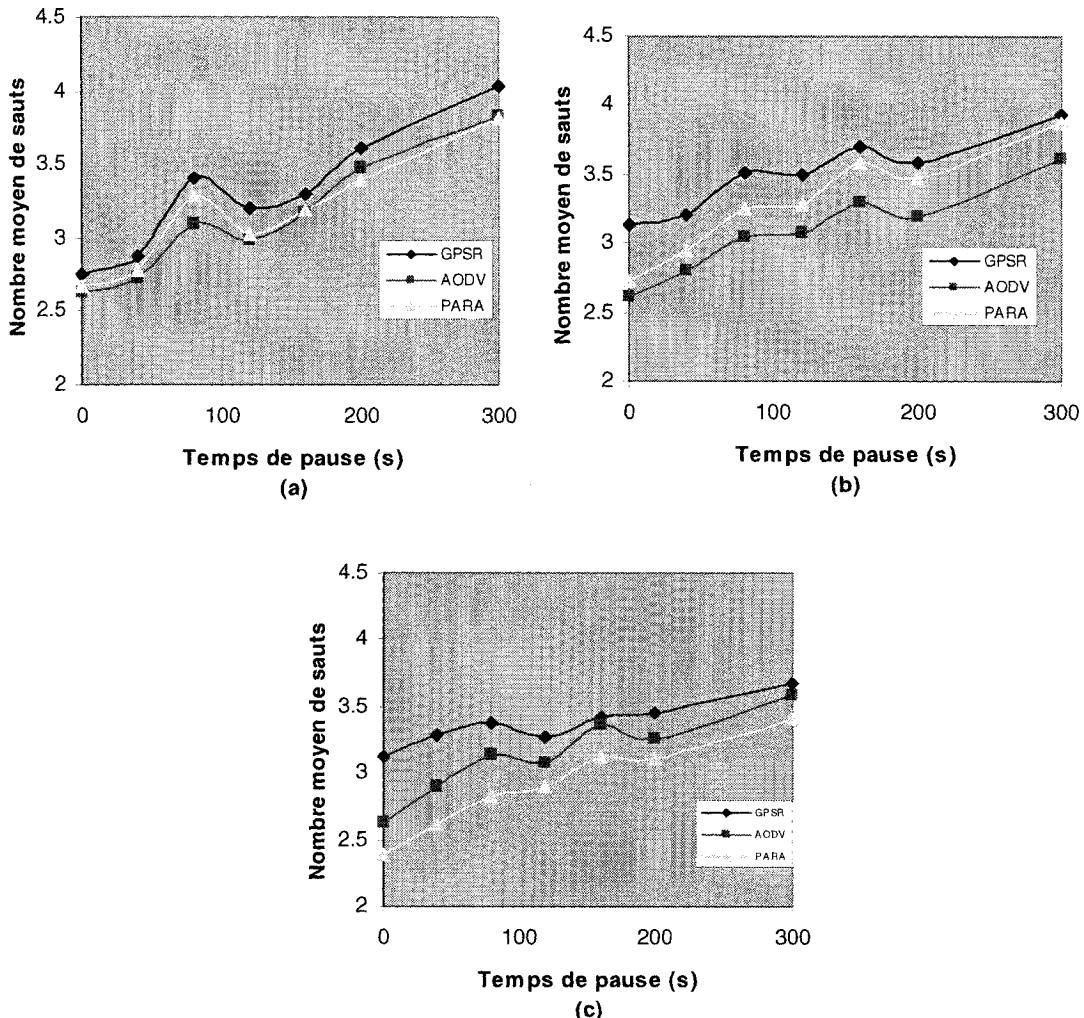


Figure 4.11: Longueur moyenne des routes dans un réseau de 50 nœuds moyennement connectés (13 nœuds en moyenne) pour (a) 10, (b) 20 et (c) 30 connexions CBR

Taux de réception. Pour 10 connexions, comme le montre la Figure 4.12, PARA délivre la quasi totalité des paquets et offre la meilleure performance. La performance de GPSR talonne celle de PARA, tandis qu'AODV délivre entre 85 et 92% des paquets. Le taux de réception d'AODV est altéré par la mobilité des nœuds.

Pour 20 connexions, GPSR apporte la meilleure performance, avec un taux moyen de réception de 95%, non affectée par la mobilité des nœuds. La performance de PARA

vient ensuite (5% de différence de performance) et s'améliore avec la sédentarité des nœuds pour atteindre celle de GPSR au temps de pause de 300s. AODV délivre en moyenne environ 5% moins de paquet que PARA, et sa performance s'améliore avec la sédentarité des nœuds.

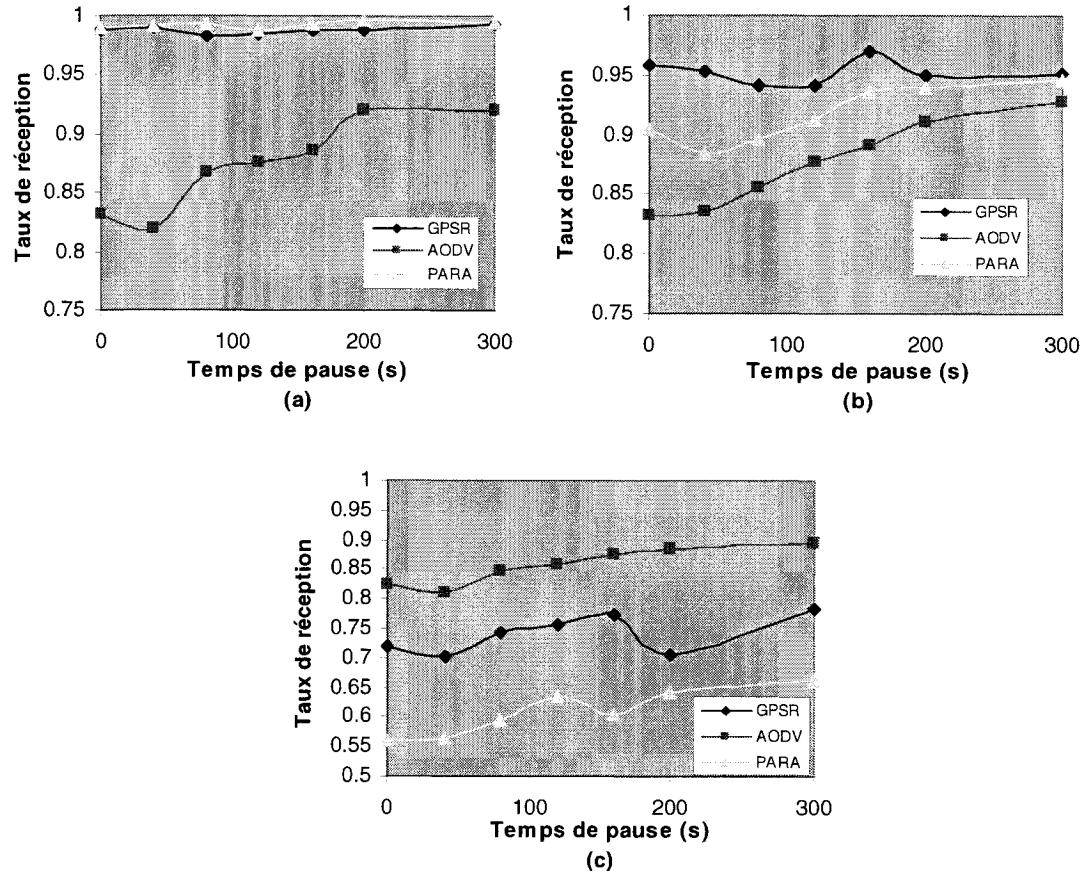


Figure 4.12 : Taux de réception dans un réseau dense de 50 nœuds (21 nœuds en moyenne) pour (a) 10, (b) 20 et (c) 30 connexions CBR

Le taux de réception de PARA est le plus bas lorsqu'il y a 30 sources, avec une valeur moyenne de 60% de paquets délivrés. Ce phénomène est lié à la congestion du réseau résultant d'une part, du trafic de contrôle important généré par PARA (proportionnel à la densité du réseau et au trafic) et d'autre part, au nombre de sources importants dans le

réseau. AODV domine, délivrant 85% de paquets, suivi de GPSR, dont le taux de réception est de 75%.

Le taux de réception de PARA diminue donc de façon marquée lorsque le nombre de sources augmente.

Délai moyen de transmission. Encore une fois, AODV offre le meilleur délai de transmission (Figure 4.13).

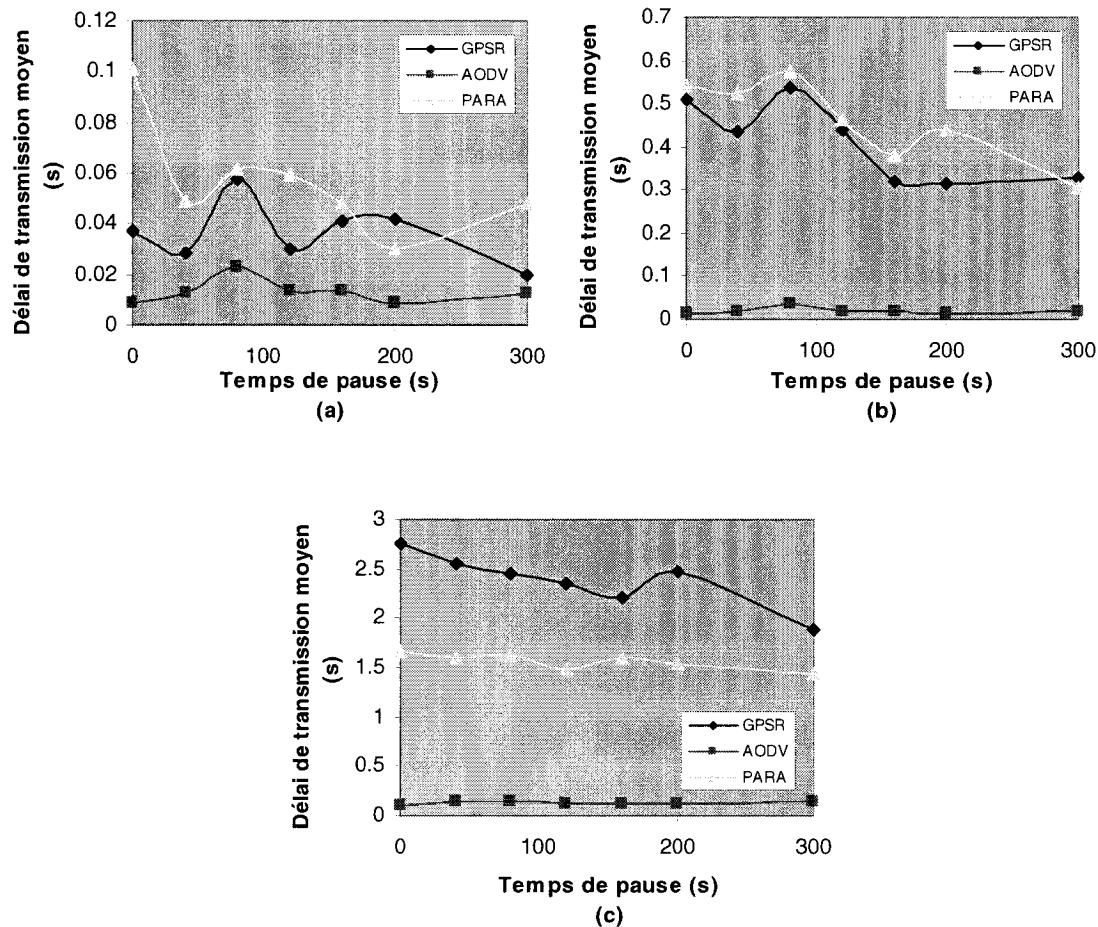


Figure 4.13 : Délai moyen de transmission dans un réseau dense de 50 nœuds (21 nœuds en moyenne) pour (a) 10, (b) 20 et (c) 30 connexions CBR

Avec 10 sources, le délai de transmission de PARA est en moyenne supérieur à celui de GPSR, notamment lorsque le temps de pause est de 0s. Dans les autres cas, leurs délais

demeurent proches. L'écart entre les performances des protocoles se creuse lorsque le nombre de sources augmente. Avec 20 sources, GPSR domine légèrement PARA, avec un délai plus de six fois supérieur à celui d'AODV. Cependant, avec 30 sources, la performance de PARA surpassé celle de GPSR de 1 seconde en moyenne. La performance de PARA demeure en deçà de celle d'AODV de 1.5 seconde.

Par ailleurs, le délai de transmission d'AODV augmente légèrement avec le temps de pause, tandis que celui de PARA et GPSR diminue avec la sédentarité des noeuds.

Longueur des routes. La Figure 4.14 illustre la longueur moyenne des routes en fonction de la mobilité des nœuds et différents nombres de sources.

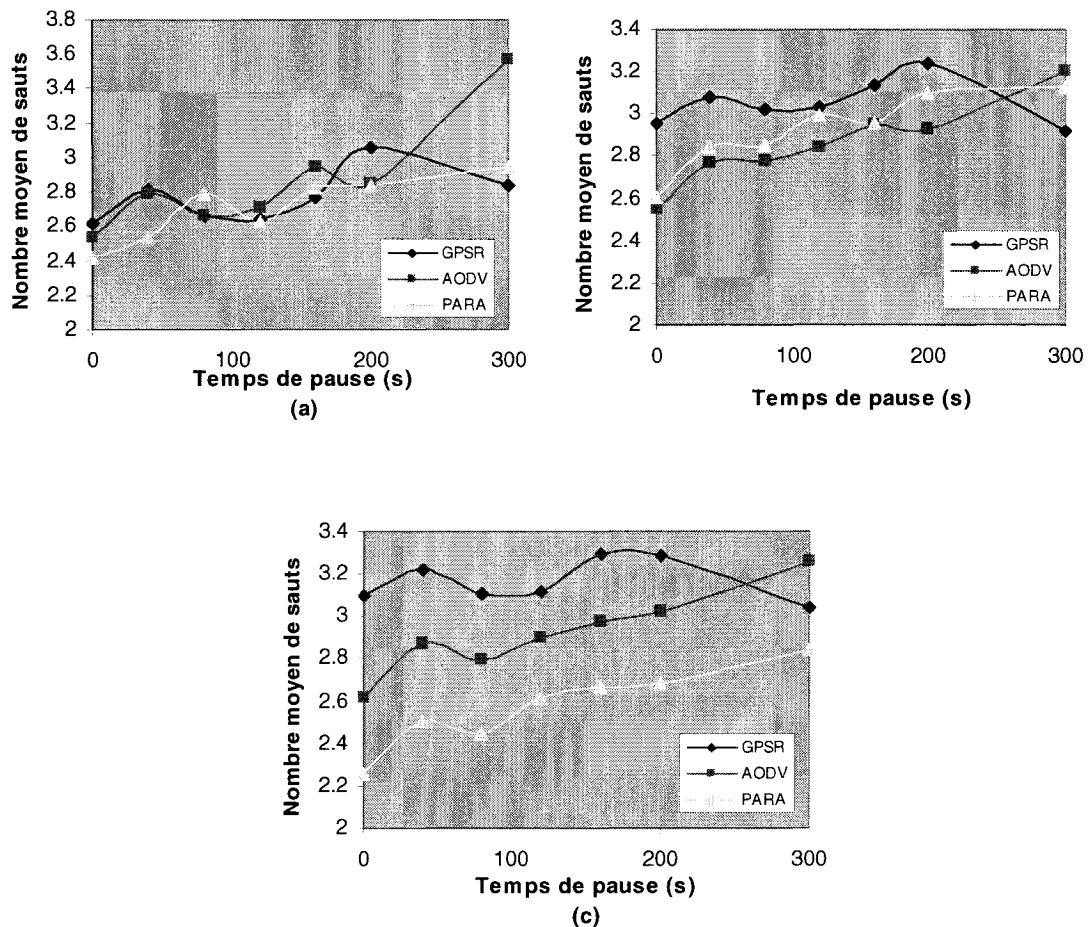


Figure 4.14 : Longueur des routes dans un réseau dense de 50 nœuds (21 nœuds en moyenne) pour (a) 10, (b) 20 et (c) 30 connexions CBR

Avec 10 sources, PARA apporte la meilleure performance lorsque la mobilité est élevée. Les résultats des protocoles sont plus ou moins similaires par la suite, excepté au temps de pause de 300s où la longueur des routes de PARA et GPSR est environ 20% inférieure à celle d'AODV.

Avec 20 sources, PARA domine GPSR, excepté avec un temps de pause de 300s. AODV fournit la meilleure performance, mais elle demeure proche de celle de PARA. PARA offre les routes les plus courtes avec 30 sources. La longueur moyenne des routes de AODV est en effet environ 18% plus grande que celle de PARA. GPSR a la pire performance, bien qu'au temps de pause de 300s, il domine AODV. De manière globale, la longueur des routes augmente avec la sédentarité des nœuds.

4.4 INTERPRÉTATION DES RÉSULTATS

Nous allons maintenant analyser les résultats présentés dans la section précédente.

➤ Taux de réception.

Tel qu'escompté, les performances de PARA sont optimales dans des réseaux dispersés et peu congestionnés. En effet, dans des réseaux faiblement ou moyennement congestionnés (Figure 4.5 (a) et (b) et Figure 4.9 (a) et (b)), PARA délivre plus de paquets que GPSR et AODV. Toutefois, les résultats obtenus par AODV en terme de taux de réception de paquets demeurent plus proches de ceux de PARA que GPSR, ce qui s'explique par le fait que la position géographique des noeuds n'est pas prise en compte par AODV pour acheminer les paquets. AODV n'est donc pas sensible à la connectivité des nœuds, contrairement à GPSR, dont les performances sont ici beaucoup moins bonnes, révélant les lacunes du mode périphérique.

D'autre part, dans des réseaux denses, les performances de PARA surpassent celles d'AODV et GPSR tant que le réseau n'est pas congestionné (Figure 4.12 (a)). Cependant, lorsque le nombre de source atteint 30, le taux de réception des paquets de PARA est bien inférieur aux autres. En effet, le trafic de contrôle de PARA est alors

relativement important, d'autant que la connectivité du réseau est élevée, et génère un nombre élevé de collisions.

La mobilité des nœuds n'a pas d'influence notable sur le taux de réception des paquets de PARA. En effet, comme dans tout protocole de routage géographique, les choix d'acheminement du paquet sont effectués indépendamment au niveau de chaque nœud actif, le concept de route n'existe donc pas. Dans un tel contexte, le routage est moins affecté par la mobilité des nœuds que pour AODV où une route est établie entre la source et la destination du paquet. Dans ce cas, toute rupture de lien nécessite l'intervention de plusieurs nœuds pour trouver une nouvelle route. Dans un contexte à mobilité élevée, AODV souffre donc plus des ruptures de liens que GPSR et AODV.

Par ailleurs, de manière globale, plus le réseau est congestionné, moins PARA est performant. En effet, comme le montre les Figures 4.15 et 4.16, lorsque la congestion du réseau augmente, la proportion de paquets en mode topologique délivrée par PARA croît aussi. Ce phénomène est tout simplement lié au trafic de contrôle généré par PARA qui est proportionnel au nombre de sources dans le réseau d'une part et à la connectivité des nœuds d'autre part. Ce trafic est d'autant plus sujet à des collisions que la congestion du réseau augmente. Par conséquent, PARA perd un certain nombre d'informations nécessaires au routage géographique. Les décisions d'acheminement sont alors moins précises et les paquets ont alors tendance à être acheminés en mode topologique.

D'autre part, avec 10 connexions et une connectivité moyenne de 5 noeuds, environ 30% des paquets acheminés par PARA le sont en mode topologique (Figure 4.15 (a)). Cette proportion relativement importante reflète la topologie dispersée du réseau. En effet, dans un tel réseau, les "trous" en terme d'acheminement géographique sont relativement nombreux et le routage géographique de bout en bout en est d'autant plus difficile. La faculté de PARA à anticiper la présence de nœuds concaves dans le réseau est donc ici révélée par le taux de réception des paquets de données largement supérieur à celui de GPSR. Cependant, dans un réseau dense, où la connectivité des nœuds est élevée, PARA délivre la majorité des paquets en mode géographique (Figure 4.16). Ainsi, point

commun à tous les protocoles évalués, le taux de réception des paquets augmente avec la densité du réseau.

Le taux de réception de GPSR est aussi altéré par un nombre de sources croissant. Nous supposons que les collisions sont alors plus nombreuses et que le mode périphérique est alors plus utilisé. Il y aurait donc formation de boucles et par conséquent, perte de paquets.

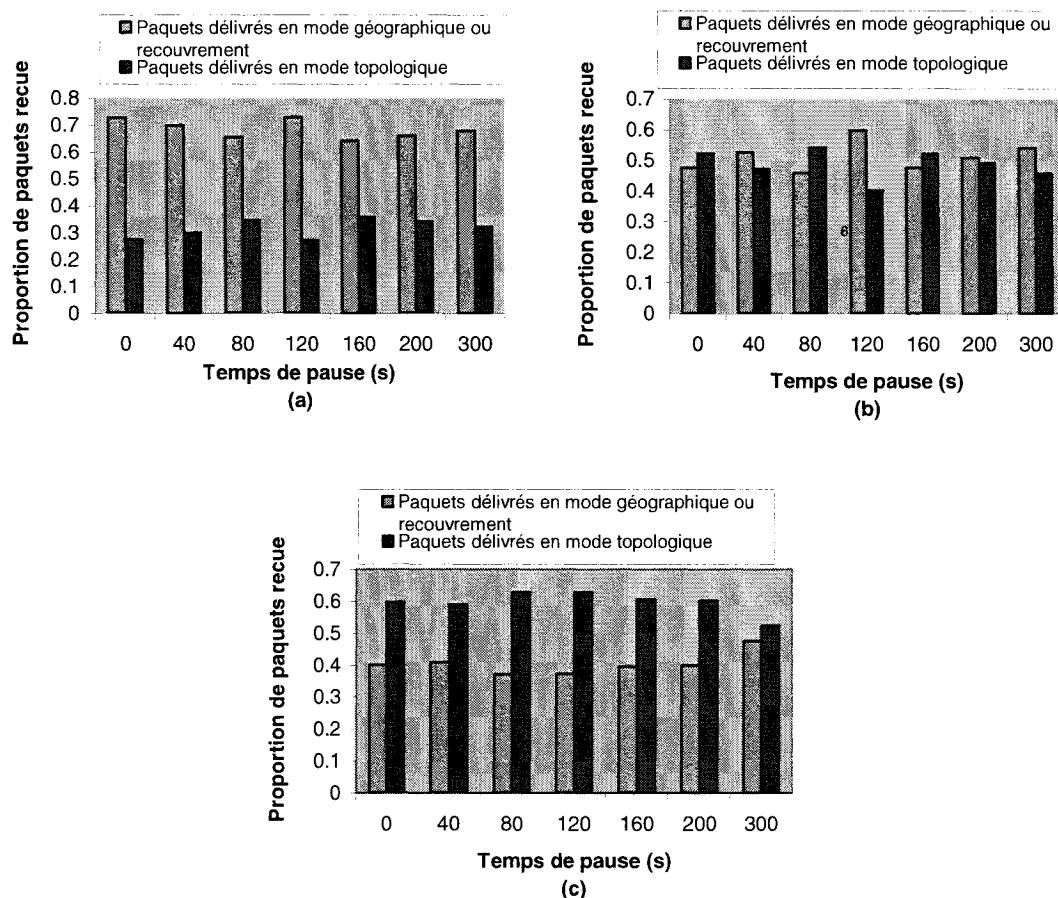


Figure 4.15 : Proportion de paquet délivrée par PARA par type d'acheminement dans un réseau de 50 noeuds dispersé (5 noeuds) pour (a) 10, (b) 20 et (c) 30 connexions CBR

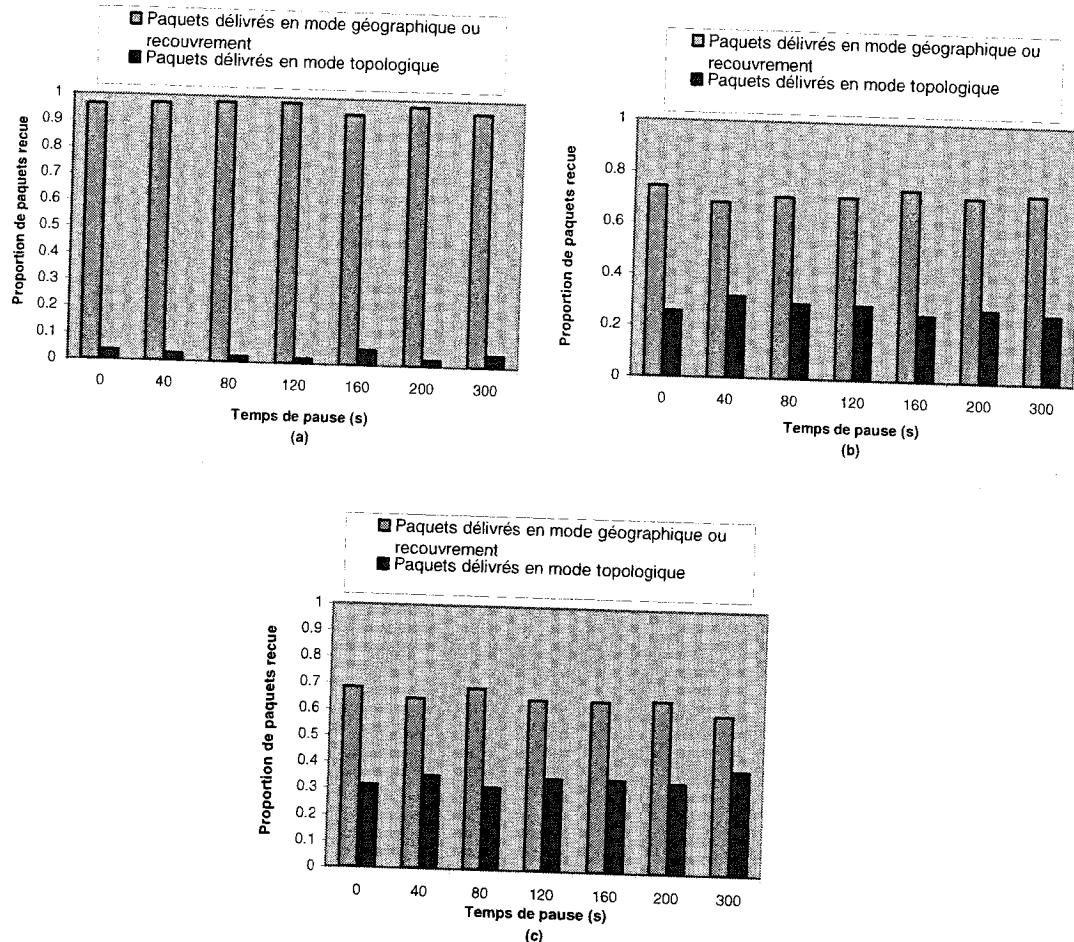


Figure 4.16 : Proportion de paquet délivrée par PARA par type d'acheminement dans un réseau dense de 50 nœuds (21 noeuds) pour (a) 10, (b) 20 et (c) 30 connexions CBR

➤ Délai moyen de transmission

Globalement, le délai de transmission moyen de PARA est toujours supérieur à celui d'AODV. AODV offre le meilleur délai, entre autres, car bien que les découvertes de routes occasionnent une certaine latence, la destination ne répond qu'à la première RREQ qu'elle reçoit. Ce processus favorise la route la moins congestionnée au détriment du plus court chemin. De plus, avec PARA, chaque découverte de proximité initiale induit une certaine latence au niveau de chaque noeud lorsque les paquets sont mis en cache. Malgré cette latence, le délai de transmission de PARA est quasiment toujours

inférieur à celui de GPSR. Le délai de transmission de GPSR est vraisemblablement lié au nombre de sauts importants que le mode périmètre induit.

Lorsque la mobilité des noeuds diminue, le délai de transmission de PARA est légèrement réduit. En effet, dans un réseau stable, les décisions de routage de PARA sont optimales et garantissent l'acheminement du paquet dans les meilleures conditions. Les mises à jour des tables de meilleurs voisins suite à des ruptures de liens sont alors réduites.

En outre, les performances de PARA sont altérées lorsque le nombre de connexions augmente. Encore une fois, lorsque le réseau est congestionné, les collisions sont plus fréquentes. Les retransmissions au niveau de la couche liaison sont alors plus nombreuses et les redirections de paquets de données augmentent le nombre de découvertes de proximité initiales. Le délai de transmission moyen s'en trouve donc augmenté.

➤ Longueur moyenne des routes.

De manière globale, la longueur moyenne des routes des paquets délivrés avec succès par PARA s'approche de celle d'AODV, tout en demeurant légèrement supérieure. En effet, rappelons qu'AODV achemine à priori les paquets par le plus court chemin, tandis que PARA choisit les chemins minimisant les risques de noeuds concaves. Les paquets délivrés par PARA le sont donc via des routes dont la longueur approche le nombre optimal de sauts. D'autre part, la longueur moyenne des routes avec GPSR est en moyenne beaucoup plus grande qu'avec PARA et AODV. Comme nous l'avons décrit dans les sections précédentes, le mode périmètre employé par GPSR pour sortir d'un optimum local consiste en un détour pour retrouver un contexte où le routage géographique est possible, ce qui explique que la longueur des routes soit plus importante.

Encore une fois, c'est dans les réseaux dispersés et moyennement connectés (Figure 4.8 et 4.11) que PARA réalise le meilleur gain de performance quant au nombre moyen de sauts des paquets. Curieusement, avec 30 sources, et notamment dans des réseaux denses

et moyennement connectés, la longueur des routes de PARA est inférieure à celle d'AODV (Figure 4.11 (c) et 4.14 (c)). Nous émettons deux hypothèses par rapport à cette tendance. D'une part, comme nous l'avons évoqué précédemment, AODV a tendance à choisir les routes les moins congestionnées au détriment des plus courts chemins, ce qui pourrait expliquer que PARA surpassé AODV dans des réseaux avec 30 sources. D'autre part, le taux de réception des paquets de données de PARA étant plus faible que celui d'AODV dans ce contexte précis, la métrique concernant le nombre moyen de sauts des paquets reçus par PARA peut s'avérer biaisée, puisque évaluée sur un nombre d'échantillons plus petit.

➤ **Trafic de contrôle normalisé.**

En terme de trafic de contrôle, contrairement à nos attentes, la performance de PARA ne surpassé pas celle de AODV et GPSR dans des réseaux de 50 noeuds, au contraire. Le trafic de contrôle généré par GPSR est stable et relativement faible puisqu'il ne s'agit que des beacons périodiques diffusés par chaque noeud. Comme dans tout protocole de routage réactif, le trafic de contrôle de PARA augmente par contre avec la mobilité des nœuds. En effet, plus les nœuds sont mobiles, plus les découvertes de proximité initiales résultant des ruptures de liens sont nombreuses. AODV suit d'ailleurs la même tendance. De plus, comme nous l'avions prévu, le trafic de contrôle généré par PARA croît considérablement lorsque la congestion et la connectivité du réseau augmente, ce qui explique les baisses de performances de PARA dans de tels contextes. En effet, une densité élevée implique que les requêtes et réponses de proximité sont beaucoup plus nombreuses à chaque découverte de proximité, tandis qu'un nombre croissant de sources dans le réseau est susceptible d'augmenter le nombre de découvertes initiales et la taille des paquets de contrôle.

Toutefois, l'écart de performance entre PARA et les autres protocoles, notamment AODV, semble se réduire dans des réseaux de 100 nœuds, comme le montre la Figure 4.17. En effet, la dimension du réseau, comme nous l'avons discuté dans la Section 3.4.5, n'a pas d'influence directe sur le trafic de contrôle de PARA, contrairement à

GPSR et AODV. Certes, la longueur des routes devrait croître avec le nombre de nœuds, ayant ainsi une certaine incidence sur le nombre de paquet de contrôle, mais cette corrélation n'est pas évidente.

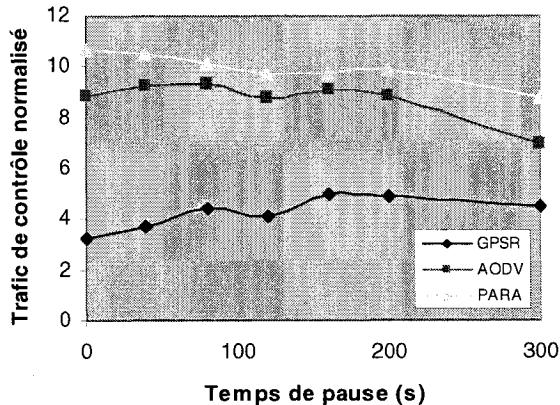


Figure 4.17 : Trafic de contrôle normalisé dans un réseau de 100 nœuds avec 10 sources

➤ Conclusion

D'après les résultats obtenus, PARA offre généralement de meilleures performances en terme de taux de réception, de délai moyen de transmission et nombre moyen de sauts des paquets que GPSR dans des réseaux dispersés ou moyennement connectés et peu congestionnés. D'autre part, la mobilité des nœuds ne semble pas avoir de réelle influence sur les métriques précédentes. Nous pouvons donc affirmer que l'approche de gestion des noeuds concaves de PARA s'avère particulièrement efficace. Les performances de PARA par rapport à AODV sont plus mitigées. Toutefois, dans les contextes favorables à PARA, le taux de réception de paquets de PARA peut s'avérer supérieur à celui d'AODV. Par ailleurs, les quelques résultats obtenus dans des réseaux de 100 nœuds (voir Annexe N) semblent conforter la tendance observée dans les réseaux de 50 nœuds.

La principale lacune de PARA est le trafic de contrôle relativement important qu'il génère. Il faut cependant noter que les différences entre la performance de PARA sur cet aspect et celles d'AODV et GPSR devraient être réduites dans des réseaux à grande

échelle. Les résultats positifs obtenus avec les autres métriques prouvent la validité du concept sur lequel repose PARA, soit le choix du prochain nœud transmetteur en fonction de son indice de proximité. En outre, les baisses de performance de PARA dans les réseaux denses ou congestionnés peuvent être imputées à ce trafic de contrôle. Certes, PARA, dans sa forme actuelle, génère beaucoup de trafic de contrôle, mais il ne s'agit là que d'un problème d'implémentation. Aux vues des résultats obtenus, tout laisse en effet penser qu'une implémentation de PARA assurant un trafic de contrôle moindre apporterait de bons résultats quelle que soit la topologie du réseau. Une approche proactive, basée sur la diffusion périodique permanente d'informations sur le voisinage à deux sauts pourrait éliminer ce problème, tout en réduisant le délai de transmission de PARA.

Chapitre V

Conclusion

Dans cet ultime chapitre, nous présentons dans un premier temps une synthèse de notre travail. Par la suite, nous évoquerons les limitations de notre protocole de routage avant de conclure avec une ébauche des travaux futurs à réaliser.

5.1 SYNTHÈSE DES TRAVAUX

Notre travail a été axé autour du routage géographique dans les réseaux ad hoc. Nous nous sommes en effet fixé pour objectif de proposer une nouvelle stratégie d'acheminement géographique qui soit innovatrice du point de vue de la gestion des nœuds concaves, notamment par rapport au protocole GPSR.

Le protocole de routage PARA que nous avons élaboré repose sur l'anticipation des noeuds concaves dans le réseau. En effet, PARA choisit le prochain nœud transmetteur d'un paquet de données, non seulement en fonction de son voisinage direct, mais aussi en fonction de son voisinage à deux sauts. Nous introduisons ainsi le concept de "découverte de proximité". L'étude du voisinage à deux sauts doit permettre d'anticiper la présence, mais surtout de minimiser le risque d'atteinte de "trous" dans le réseau où l'avancée géographique du message vers la destination n'est plus possible. Une découverte de proximité permet donc à un nœud de découvrir, d'une part, ses meilleurs voisins pour chacune de ses destinations actives et, d'autre part, le voisinage de ses meilleurs voisins. La stratégie d'acheminement déployée par PARA consiste en effet à choisir comme prochain nœud transmetteur celui des ses meilleurs voisins qui a l'indice de proximité le plus élevé. Au cas où cette approche ne permettrait pas d'éviter l'atteinte d'un noeud concave, PARA est aussi doté d'une stratégie de recouvrement. Finalement, si cette dernière échoue, les paquets sont acheminés avec un routage topologique, AODV.

Afin d'évaluer dans quelle mesure PARA atteint les objectifs escomptés, nous avons procédé à des séries de simulations avec l'outil logiciel *NS Network Simulator*. Par ailleurs, pour mesurer quantitativement les améliorations apportées par PARA, nous avons également simulé les protocoles GPSR et AODV. Les métriques que nous avons utilisées pour évaluer la performance des protocoles sont le taux de réception, le trafic de contrôle normalisé, le délai moyen de transmission ainsi que la longueur moyenne des routes. Nous avons aussi simulé diverses densités de réseaux afin, notamment, de déterminer le comportement des protocoles dans des réseaux dispersés.

Les résultats obtenus prouvent l'efficacité de PARA par rapport aux nœuds concaves. En effet, dans des réseaux dispersés, PARA surpassé largement GPSR pour ce qui est du taux de réception, du délai moyen de transmission et de la longueur moyenne des routes. Dans un tel contexte, PARA offre d'ailleurs la meilleure performance quant au taux de réception des paquets. En outre, quelle que soit la topologie du réseau, PARA domine généralement GPSR en terme de délai moyen de transmission et longueur moyenne des routes. D'autre part, dans les réseaux denses et moyennement connectés, PARA délivre avec succès plus de paquets que AODV tant que le réseau n'est pas congestionné. Par ailleurs, les performances de PARA ne sont pas altérées par la mobilité des nœuds. Finalement, PARA ne génère pas de boucles de routage.

PARA constitue donc un protocole de routage particulièrement efficace dans les réseaux dispersés pour le trafic sensible aux pertes.

5.2 LIMITATION DES TRAVAUX

La principale lacune de PARA est le trafic de contrôle important qu'il génère. En effet, les résultats obtenus lors des simulations montrent que ce trafic est, d'une manière générale, largement supérieur à celui de GPSR et AODV. Bien que les différences s'amenuisent lorsque la dimension du réseau augmente, la performance de PARA demeure en deçà de celle des autres protocoles, contrairement à nos attentes. Comme nous l'avions prévu, le nombre de paquets de contrôle générés par PARA augmente avec le nombre de sources et la densité du réseau. Ce trafic de contrôle important a pour

conséquence une baisse de performance significative des performances de PARA dans des réseaux denses et / où congestionnés où les collisions sont alors beaucoup plus nombreuses. L'approche réactive sur laquelle est basée PARA n'a donc pas eu l'effet escompté, au contraire. Rappelons toutefois que PARA n'a pas été élaboré pour les réseaux denses, car l'acheminement géographique est alors très efficace.

Le délai de transmission de PARA, certes meilleur que celui de GPSR, demeure supérieur à celui de AODV. Cette latence est elle aussi imputable à l'aspect "on-demand" de PARA, car ce délai inclut notamment les temps de mise en tampon des paquets lors des découvertes de proximité initiales.

Nous avons par ailleurs considéré un service de localisation idéal dans le cadre de nos simulations de PARA et GPSR. En effet, notre objectif était ici de prouver la validité du concept d'acheminement de PARA. Nous avons donc volontairement négligé le trafic qui pourrait être induit par les requêtes au service de localisation.

Finalement, nous avons stipulé dans l'une de nos hypothèses que nous négligions les erreurs de localisations liées au Service de Positionnement Global. Nous sommes conscients que cette hypothèse idéalise nos résultats et qu'il s'avérerait pertinent d'étudier l'impact des erreurs de positionnement sur les performances de PARA.

5.3 TRAVAUX FUTURS

Après avoir décrit les limitations de notre travail, nous allons maintenant exposer quelques pistes de recherche relative à notre sujet. Dans un premier temps, une suite logique à notre travail serait de proposer une nouvelle implémentation de PARA qui réduirait le trafic de contrôle. Une approche semblable aux beacons périodiques de GPSR pourrait par exemple être adoptée, en étendant cependant les informations échangées au voisinage à deux sauts. Une implémentation de PARA économisant la bande passante devrait par ailleurs permettre une amélioration significative des performances de PARA dans les réseaux denses et congestionnés. La latence induite par les découvertes de proximité serait de plus éliminée.

PARA ne semble pas, dans l'état actuel des choses, démontrer de sensibilité à la mobilité des nœuds, toutefois il pourrait s'avérer intéressant d'inclure des facteurs de prédiction de la position des nœuds dans le choix du prochain nœud transmetteur. On pourrait ainsi prendre en compte, en plus de sa distance à la destination, la vitesse et la direction du noeud dans l'évaluation de la validité d'un meilleur voisin.

Finalement, un autre point intéressant serait d'inclure des aspects de qualité de service à PARA.

BIBLIOGRAPHIE

- [1] S. Basagni, I.Chlamtac, V.R. Syrotivk, B.A. Woodward, “A distance effect algorithm for mobility (DREAM) ”, Proceedings of the Fourth Annual ACM/IEEE International Conference on Mobile Computing and Networking (Mobicom’98), Dallas, TX, 1998
- [2] M.S. Corson, A. Ephremides, “A distributed routing algorithm for mobile wireless networks”, ACM/Baltzer Wireless Networks 1 (1) (1995), pp. 61-81
- [3] C. Perkins et E. Royer, “Ad hoc on Demand Distance Vector Routing”, Proc. IEEE WMCSA, LA, Février 1999, pp. 90-100
- [4] D.B. Johnson et D. Maltz, “Dynamic Source Routing in Ad Hoc Wireless Network”, *Mobile Computing*, T. Imielinski et H. Korth, Chapitre 5, Kluwer, 1996, pp. 88-153
- [5] Y.-B. Ko et N.H. Vaidya, “Location-aided routing (LAR) in mobile ad hoc networks”, Proceedings of the Fourth Annual ACM/IEEE International Conference on Mobile Computing and Networking (Mobicom’98), Dallas, TX, 1998
- [6] H. Takagi et L. Kleinrock, “Optimal transmission ranges for randomly distributed packet radio terminals”, IEEE Transactions on Communications, 32, 3, 1984, pp. 246-257
- [7] R. Nelson et L. Kleinrock, “The spatial capacity of a slotted ALOHA multihop packet radio network with capture”, IEEE Transactions on Communications, 32, 6, 1984, pp. 684-694
- [8] T.C. Hou et V.O.K. Li, “Transmission range control in multihop packet radio networks”, IEEE Transactions on Communications 34 (1) (1986), pp. 38-44

- [9] G.G. Finn, "Routing and addressing problems in large metropolitan-scale internetworks", ISI Research Report ISU/RR-87-180, Mars 1987
- [10] I. Stojmenovic et X.Lin, "Loop-free hybrid single-path/flooding routing algorithms with guaranteed delivery for wireless networks", IEEE Transactions on Parallel and Distributed Systems, Volume 12, Issue 10 (October 2001), pp. 1023-1032
- [11] E. Kranakis, H. Singh et J. Urrutia, "Compass routing on geometric networks", Proceedings of the 11th Canadian Conference on Computational Geometry, Vancouver, Août 1999
- [12] X. Lin, M. Lakshdisi et I. Stojmenovic, "Location based localized alternate, disjoint, multi-path and component routing algorithms for wireless networks", Proc. ACM Symposium on Mobile Ad Hoc Networking & Computing MobiHoc, Long Beach, California, USA, 4-5 Octobre, 2001
- [13] R. Jain, A. Puri et R. Sengupta, "Geographical routing using partial information for wireless ad hoc networks", IEEE Personal Communications, Février 2001, pp. 48-57
- [14] M. Mauve et J. Widmer, "A survey on Position-Based Routing in Mobile Ad Hoc Networks", IEEE Network, 15, Novembre 2001, pp. 30-39
- [15] B.Karp et H.T. Kung, "GPSR: Greedy Perimeter stateless routing for wireless networks", Proc. MOBICOM, Août 2000, pp. 243-254
- [16] Lj. Blazevic, S. Giordano et J.Y. Le Boudec, "Self-organized terminode routing", TR DSC/2000/040, Swiss Federal Institute of Technology, Lausanne, Décembre 2000

- [17] W.H. Liao, Y.C. Tseng, J.P. Sheu, “GRID: A fully location-aware routing protocol for mobile ad hoc networks”, Proc. IEEE HICSS, Janvier 2000
- [18] J. Hightower et G. Borriello, “Location Systems for Ubiquitous Computing”, Computer, vol.34, no. 8, Août 2001, pp. 57-66
- [19] Z. J. Haas et B. Liang, “Ad Hoc Mobility Management with Uniform Quorum Systems”, IEEE/ACM Trans. Net., vol. 7, no. 2, Avril 1999, pp.228-40
- [20] S. Giordano, I. Stojmenovic et L. Blazevic, “Position based algorithms for ad hoc networks: a taxonomy”, Juillet 2001, <http://www.site.uottawa.ca/~ivan/routing-survey.pdf>
- [21] S. Capkun, M. Hamdi et J.P Hubaux, “GPS-free positioning in mobile ad-hoc networks”, Proc. Hawai Int. Conf. On System Sciences, Janvier 2001
- [22] M. Heissenbüttel et T. Braun, “A Novel Position-based and Beacon-less Routing Algorithm for Mobile Ad-Hoc Networks ”, 3rd IEEE Workshop on Applications and Services in Wireless Networks, Bern, Switzerland, Juillet 2-4, 2003
- [23] H.Füßler, J. Widmer, M. Mauve et H. Hartenstein, “A Novel Forwarding Paradigm for Position-Based Routing (with Implicit Addressing)”, Proceedings of IEEE 18th Annual Workshop on Computer Communications, 20-21 Octobre 2003, pp. 194-200
- [24] Giordano S., Stojmenovic L. et Blazevic L., “The Virtual Home Region”, Technical Report No. SSC/1999/037, EPFL, Switzerland, Octobre 1999
- [25] Li J., Jannotti J., De Couto D., Karger D. et Morris R., “A scalable location service for geographic ad hoc routing”, Proceedings of ACM MOBICOM, ACM, Août 2000

- [26] RFC 3561 - Ad hoc On-Demand Distance Vector (AODV) Routing, Juillet 2003, <http://www.faqs.org/rfcs/rfc3561.html>
- [27] T.Camp, "Location Information Services in Mobile Ad Hoc Networks", Technical Report MCS-03-15, The Colorado School of Mines, October 2003, <http://toilers.mines.edu>
- [28] S. Giordano et M. Hamdi, "Mobility management: The virtual home region", Technical Report SSC/1999/037, Ecole Polytechnique Federale de Lausanne, October 1999.
- [29] I. Stojmenovic, "Home agent based location update and destination search schemes in ad hoc wireless networks", dans *Advances in Information Science and Soft Computing*, Editeurs A. Zemliak et N.E. Mastorakis, pages 6.11., WSEAS Press, 2002. Aussi, Technical Report, University of Ottawa, TR-99-10, September 1999.
- [30] J. Li, J. Jannotti, D. De Couto, D. Karger et R. Morris., "A scalable location service for geographic ad hoc routing", dans *Proceedings of the ACM/IEEE International Conference onMobile Computing and Networking (MOBICOM)*, pages 120.130, 2000.
- [31] T. Camp, J. Boleng et L. Wilcox, "Location information services in mobile ad hoc networks", dans *Proceedings of the IEEE International Conference on Communications (ICC)*, pages 3318.3324, 2001.
- [32] P.-H. Hsiao., "Geographical region summary service for geographical routing", dans *Proceedings of the ACM International Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC)*, pages 263.266, 2001.
- [33] L. Blazevic, "Scalable Routing Protocols with Applications to Mobility", PhD thesis, Ecole Polytechnique Federale de Lausanne, 2001.

ANNEXES

ANNEXE A

Services de localisation

Nous présentons ici divers services de localisation proactifs fréquemment discutés dans la littérature.

➤ **Systèmes de bases de données de localisation**

Diverses approches de systèmes de bases de données de localisation existent. Nous décrirons brièvement ici les plus importantes. Rappelons que dans de tels systèmes, des nœuds spécifiques servent de bases de données de localisation pour d'autres nœuds spécifique du réseau. Généralement, un nœud met à jour sa position dans ses serveurs de localisation lorsqu'il se déplace vers une nouvelle position. Par la suite, les serveurs de localisation peuvent être interrogés lorsque la position du nœud est nécessaire.

Plusieurs **services de localisation basés sur la région d'origine** (*home region location services*), similaires à Mobile IP, ont été proposés. Dans ces protocoles, chaque nœud du réseau est associé à une région d'origine. Une région d'origine (appelée région d'origine virtuelle dans [28] et agent d'origine dans [29]) peut être définie par un rectangle ou une position et un cercle de rayon R autour de cette position. Tous les nœuds à l'intérieur de la région d'origine d'un noeud donné, soient tous les nœuds à l'intérieur du cercle ou du rectangle, maintiennent les informations de la localisation de ce nœud. Ainsi, lorsque le nœud se déplace, il envoie sa position courante à tous les nœuds de sa région d'origine.

Le paquet de mise à jour de position est acheminé à la région d'origine via un protocole de routage géographique. A la réception du paquet de mise à jour de position, chaque nœud à l'intérieur de la région d'origine rediffuse le paquet à tous les nœuds de la région d'origine.

Par la suite, lorsqu'une source désire transmettre un paquet à une destination, la source interroge l'agent d'origine pour obtenir la position actuelle de la destination. Plus spécifiquement, le paquet de requête de position est envoyé par la source au centre de la

région d'origine en suivant un algorithme d'acheminement géographique. Chaque nœud recevant cette requête dans la région d'origine répond avec la position de la destination.

Les services de localisations basés sur les **quorums** reposent sur la réPLICATION DES INFORMATIONS DE LOCALISATION AU NIVEAU DE PLUSIEURS NŒUDS, QUI SERVENT DE DÉPÔTS D'INFORMATIONS. En d'autres termes, des quorums (des sous-ensembles de noeuds) de lecture et d'écriture sont définis dans le réseau. Lorsqu'un noeud a besoin de mettre à jour sa position, il transmet sa mise à jour à un quorum d'écriture. Quand une source désire, pour sa part, transmettre un paquet à une destination, il interroge un quorum de lecture pour la position de la destination. Le principal défi dans un système basé sur les quorums consiste à définir les quorums de lecture et d'écriture de manière à maximiser la probabilité de succès d'une requête. L'objectif est en effet de définir les quorums de telle sorte que le quorum de lecture d'un noeud croise le quorum d'écriture d'un autre noeud. Ainsi, les mises à jour peuvent être obtenues pour toutes les destinations. **GLS** (*Grid Location Service*) [30] est un système de localisation basé sur les quorums particulièrement reconnu.

➤ Systèmes de dissémination de localisation

Dans un système de dissémination de localisation, tous les nœuds du réseau reçoivent périodiquement les mises à jour de localisation d'un nœud spécifique. Ainsi, chaque nœud du réseau maintient les positions de tous les autres nœuds du réseau.

Le service de localisation de DREAM [31], **DLS** (*DREAM location service*), est similaire à celui proposé par les auteurs de DREAM [1]. Chaque paquet de localisation, permettant la mise à jour des tables de localisation, contient les coordonnées et la vitesse de la source, ainsi que l'instant de transmission du paquet. Chaque nœud du réseau transmet un tel paquet à un taux différents selon que les nœuds auxquels il est destiné sont proches ou éloignés. Comme nous le verrons dans la section 2.3, les nœuds éloignés semblent se déplacer plus lentement que les nœuds proches. Il n'est donc pas nécessaire de maintenir des informations à jour pour les nœuds éloignés. Ce phénomène est appelé *effet de distance*. DREAM contrôle donc les informations de localisation dans le réseau

via la fréquence à laquelle les paquets de localisation sont transmis et le nombre de sauts qu'ils sont autorisés à effectuer.

Un autre service de localisation, GRSS (Geographic Region Summary Service), proposé dans [32] est aussi souvent cité dans la littérature.

ANNEXE B

Protocoles de routage géographique hiérarchiques

Les protocoles de routage géographique Terminode et DIR sont dits hiérarchiques, dans la mesure où ils appliquent différentes règles au routage longue distance et courte distance. Le routage géographique est utilisé pour le routage longue distance, tandis que lorsque le paquet arrive proche de la destination, un algorithme de routage topologique proactif est adopté.

Routage Terminode

Le routage hiérarchique **Terminode** [16] utilise à la fois l'inondation et le routage multi-chemins. Ce mode de routage est la combinaison de deux protocoles appelés *Terminode Local Routing* (TLR) et *Terminode Remote Routing* (TRR).

TLR est un mécanisme qui permet de joindre des destinations aux alentours d'un terminode, ou unité mobile, et n'utilise pas les informations de localisation pour prendre les décisions d'acheminement des paquets.

TRR, quant à lui, est utilisé pour envoyer des données aux destinations et utilise les informations géographiques des noeuds. Il constitue l'élément clé pour assurer l'évolutivité et réduire la dépendance vis-à-vis des systèmes intermédiaires. La nouveauté principale de cet algorithme est le mode d'acheminement géodésique basé sur les ancre ou AGPF (*Anchored Geodesic Packet Forwarding*) de TRR. Il s'agit d'une méthode basée sur un routage de source, mais au lieu d'inclure un chemin traditionnel dans le paquet, c'est-à-dire une liste de noeuds, il utilise une liste d'ancre. Les ancre sont des points géographiques fixes. Le paquet est acheminé en suivant le chemin d'ancre. En effet, à chaque noeud intermédiaire, le paquet est envoyé dans la direction de la prochaine ancre sur le chemin d'ancre en appliquant l'acheminement géodésique des paquets. Lorsqu'un terminode, ou unité mobile, détermine que la prochaine ancre se situe dans son rayon de transmission, il l'efface du chemin d'ancre et envoie le paquet

dans la direction de la nouvelle prochaine ancre. Le processus se répète jusqu'à ce que le paquet soit envoyé dans la direction de la destination finale.

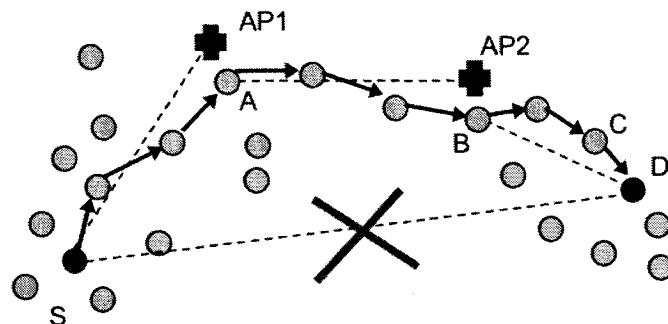


Figure B.1 : Déroulement des opérations de AGPF

La Figure B.1 illustre le fonctionnement de AGPF lorsqu'une source S doit transmettre des paquets au nœud D et qu'il n'y a pas de connectivité le long de la ligne droite entre S et D . S dispose d'un chemin d'ancres vers D constitué d'une liste de positions géographiques appelées points d'ancrage : $\{AP1, AP2\}$. Dans un premier temps, l'acheminement géodésique de paquet dans la direction d' $AP1$ est utilisé. Après quelques sauts, le paquet arrive au nœud A qui découvre qu'il est proche de $AP1$. En A , le paquet est transmis en utilisant l'acheminement géodésique de paquet en direction de $AP2$. Par la suite, lorsque le paquet arrive en B , c'est-à-dire proche de $AP2$, le nœud B commence à transmettre le paquet vers D . Finalement, lorsque le paquet arrive au niveau du nœud C , celui découvre que D est accessible avec le mécanisme TLR et transmet le paquet à D en utilisant TLR.

TRR possède de plus une composante, la découverte de chemin d'ancres ou APD (*Anchored Path Discovery*), offrant deux méthodes distinctes pour obtenir les chemins d'ancres.

Les résultats des simulations pour des réseaux ad hoc mobiles composés de plusieurs centaines d'unités mobiles démontrent les bénéfices de la combinaison de TLR

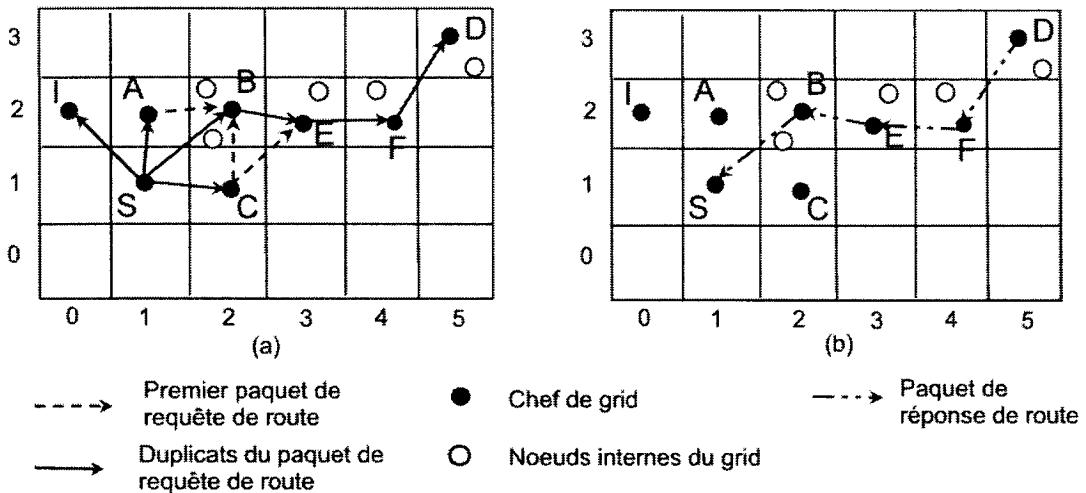
et TRR par rapport aux protocoles existants utilisant les informations géographiques pour le routage.

Le protocole GRID

Le protocole GRID [17] est un protocole hiérarchique qui exploite les informations de localisation des nœuds fournies par un système GPS dans la découverte de routes, l'acheminement des paquets, ainsi que la maintenance des routes.

La zone géographique du réseau est divisée en un certain nombre de carrés ou *grids*, une correspondance étant définie au préalable entre les positions des nœuds et les coordonnées des *grids*. Dans chaque *grid*, un nœud mobile, s'il y en a, est élu chef de *grid* par un protocole d'élection de chef de *grid*. Le routage est ensuite performé *grid* par *grid* à travers les chefs de *grid*, les autres nœuds n'ayant pas cette responsabilité. Les tables de routage contiennent donc les identificateurs (IDs) des *grids* et non des nœuds. La taille d de chaque *grid* dépend du rayon de transmission R des nœuds et plusieurs options sont proposées, dans l'idée de faire en sorte que chaque chef de *grid* puisse communiquer avec les chefs des *grids* voisins. Les chefs de *grid* constituent donc un ensemble dominant. Un rayon de recherche est aussi défini. Les auteurs proposent notamment plusieurs options pour la géométrie du rayon de recherche (rectangle, cône, etc.).

Le protocole AODV a été adapté pour développer GRID. Ainsi, lorsqu'un nœud S a besoin d'une route vers le nœud D , il diffuse un paquet de requête de route contenant, outre les adresses des nœuds source et destination, les numéros de séquence correspondants, un identificateur de la requête et le rayon de recherche. Lorsqu'un chef de *grid* reçoit une requête de route, il vérifie d'abord s'il est localisé dans la zone spécifiée par le rayon de recherche ou non. Si, oui, il fixe un pointeur inverse pour la réponse à la requête et propage le paquet. Sinon, il ignore la requête. La requête est ensuite propagée *grid* par *grid* jusqu'à ce que la destination soit atteinte. La suite du processus est similaire à AODV (voir Figure B.2).



- (a) Propagation des paquets de requête de route de la source *S* vers la destination *D*. Seuls les nœuds *A*, *B*, et *C* sont dans le rayon de recherche, et donc propagent le paquet.
 (b) Propagation du paquet de réponse de route de la destination *D* vers la source *S*.

Figure B.2 : Un exemple de découverte de route dans GRID

Le choix des chefs de *grid* repose sur un protocole d'élection. Les chefs de *grid* annoncent leur existence en diffusant des paquets GATE contenant les coordonnées du *grid* correspondant et leur position courante. Chaque nœud d'un *grid* reçoit les paquets GATE du chef de *grid* courant. Cependant, si au bout d'une période prédefinie aucun paquet GATE n'est reçu, chaque nœud du *grid* peut diffuser un paquet BID contenant les coordonnées du *grid*, ainsi que sa position afin de réclamer le statut de chef de *grid*. Si un chef de *grid* existe, il répond par un paquet GATE pour refuser sa demande. Dans un *grid* sans chef, le nœud le plus proche du centre du *grid* peut rejeter l'offre du nœud précédent en envoyant son propre paquet BID. Si aucune de ces situations ne se produit, alors le nœud ayant émis le premier BID s'auto élit chef de *grid*. C'est donc le nœud le plus proche du centre du *grid* qui est élu chef de *grid*. Lorsqu'un chef de *grid* quitte le *grid* courant, un autre nœud de ce *grid* le remplace.

Les auteurs de GRID utilisent le protocole LAR pour la découverte de route, bien que de meilleures options soient disponibles, comme nous l'avons déjà discuté.

L'utilisation de LAR peut en effet induire un trafic de contrôle important et créer des boucles. D'autre part, la maintenance des routes ne prend pas en compte la situation où un *grid* devient vide après que son chef et unique nœud l'ait quitté. Enfin, le partitionnement du réseau en *grids* repose sur une carte statique du réseau préalablement établie.

ANNEXE C

Structures de stockages mises en œuvre par PARA

- **La table de meilleurs voisins :** Les nœuds actifs stockent des informations sur leurs meilleurs voisins dans leur table de meilleurs voisins et ce, pour chacune de leurs destinations actives. Pour chaque destination active Di sont ainsi stockés l'adresse, la distance à Di , l'indice de proximité et l'indice d'accessibilité de chaque meilleur voisin (voir Tableau annexé E.2). Les entrées des meilleurs voisins pour chaque destination sont maintenues en ordre décroissant d'importance par une fonction de tri que nous évoquerons plus en détail dans les sections suivantes. Pour déterminer le prochain nœud transmetteur d'un paquet ayant pour destination le nœud Di , un nœud émetteur récupérera donc l'adresse du premier meilleur voisin pour Di dans sa table de meilleurs voisins.
- **Le cache de découvertes de proximité :** Le cache de découvertes de proximité est utilisé par les nœuds actifs pour mémoriser leur état, leurs destinations actives et les découvertes de proximité initiales en cours. Nous reviendrons sur ce sujet dans les sections suivantes.
- **Le cache de nœuds concaves :** Pour éviter la formation de boucles de routage, les nœuds actifs mémorisent temporairement les nœuds qui se révèlent concaves pour une destination, afin de ne plus leur transmettre de paquets pour cette destination (voir Tableau annexé E.3).
- **Le cache de réponse de proximité :** Un autre type de stockage est le *cache de réponses de proximité*. Contrairement aux structures précédentes qui ne sont maintenues que par les nœuds actifs, ce cache est utilisé par les voisins d'un nœud initiateur d'une découverte de proximité. Ce cache joue en fait deux rôles, selon la nature des nœuds voisins :
 - Tous les nœuds voisins y mémorisent les requêtes de proximité de niveau 1 qu'ils ont reçues;

- Les nœuds voisins qui s'avèrent être des meilleurs voisins y stockent aussi les informations issues de la phase 2 de la découverte de proximité, avant de les transmettre au nœud initiateur de la découverte.

Ce processus sera explicité dans la Section 3.3.4. Pour simplifier la dénomination de ce cache, nous parlerons de *cache de PREP1s*. Chaque entrée du cache a une durée de vie fixée par le paramètre *reply_delay*, qui spécifie aussi la durée de la phase 2 d'une découverte de proximité. Le Tableau annexé E.1 décrit le format des données stockées dans ce cache.

- **Le cache de requêtes de proximité :** Les nœuds recevant des requêtes de proximité de niveau 2 lors de la phase 2 d'une découverte de proximité mémorisent les requêtes qu'ils ont reçues dans un *cache découvertes de proximité*, ou *cache de PREQ2s* et ce, le temps que la phase 2 soit achevée. Le délai de stockage est donc fixé par le paramètre *reply_delay*. Le format des informations mémorisées dans le cache de PREQ2s est décrit par le Tableau annexé E.4.

ANNEXE D

Déroulement d'une découverte de proximité à destinations multiples

Le nœud N , actif, initie une découverte de proximité périodique à intervalle régulier fixé par le paramètre *prequest_interval*. Le processus est le même que pour une découverte de proximité à destination unique, avec quelques subtilités que nous allons détailler.

❖ Étape 1: Diffusion d'une PREQ1 par le nœud initiateur

Un nœud initie toujours une découverte de proximité en diffusant une PREQ1. Le format de l'entête d'une PREQ1 est décrit par le Tableau annexé F.2. Le nœud N évalue d'abord sa position, puis la distance qui le sépare de chacune de ses destinations actives, après quoi il insère son nombre de destinations actives dans l'entête de la PREQ1 et pour chacune d'entre elles les informations suivantes:

- L'adresse et les coordonnées de la destination;
- Sa distance à la destination en question.

La PREQ1 est ensuite diffusée (Figure D.1 (a)).

❖ Étape 2 : Réception de la PREQ1 par les voisins directs

Chaque voisin direct de N , V , lorsqu'il reçoit une PREQ1, enregistre, comme auparavant, la PREQ1 dans son cache de réponses de proximité. Supposons que le nœud N ait $NDst$ destinations actives et notons Di la i -ème destination présente dans l'entête de la PREQ1, avec $1 \leq i \leq NDst$. Le comportement du nœud V est alors fonction du résultat de la comparaison suivante, effectuée pour chaque destination Di :

- Si le nœud V est la destination Di , le nœud va mettre à jour ses informations pour cette destination dans son cache de PREP1s :
 - L'adresse de la destination Di (même s'il s'agit de sa propre adresse) ;

- Son indice d'accessibilité, dont la valeur est 2 puisque V est la destination;
 - Son indice de proximité, qui vaut 1.
 - Sa distance à D_i , qui est nulle.
- Sinon, le nœud V récupère sa position, puis évalue sa distance à D_i , dont les coordonnées sont incluses dans la PREQ1. Il compare ensuite sa distance à celle du nœud initiateur de la découverte contenue dans l'entête de la PREQ1 :
- Si $DIST(V, D_i) < DIST(N, D_i)$, alors le nœud V est un meilleur voisin de N pour D_i et D_i constitue une meilleure destination pour V . Le nœud V va donc initialiser ses attributs pour la destination D_i dans son cache de PREP1s. Il ajoute les champs suivants à l'entrée correspondant à cette découverte:
 - L'adresse de D_i ;
 - Son indice d'accessibilité initialisé, $IsDest(V, D_i) = 0$;
 - Son indice de proximité initialisé, $IProx(V, D_i) = 0$;
 - Sa distance à D_i , $DIST(V, D_i)$.
- Le nœud V incrémentera aussi le nombre de ses meilleures destinations, c'est-à-dire le nombre de destinations pour lesquelles il est un meilleur voisin pour N . Nous noterons cet attribut $N_MDst(V)$.
- Sinon, le nœud V ne constitue pas un meilleur voisin de N pour la destination D_i .

A l'issu du traitement de la PREQ1, chaque voisin V vérifie s'il est un meilleur voisin pour au moins une destination incluse dans la PREQ1 :

- Si $N_MDst(V) > 0$, le nœud V est un meilleur voisin de N pour au moins une de ses destinations actives. Nous le noterons désormais $MV_N(V)$. $MV_N(V)$ va donc procéder à la phase 2 de la découverte de proximité, afin de déterminer s'il possède lui-même des meilleurs voisins pour ses meilleures destinations.
- Sinon, le nœud V ne permet d'avancée géographique vers aucune des destinations actives du nœud initiateur N . Il ne fait rien de plus.

Illustrons le comportement des nœuds par un exemple. Supposons que les destinations actives du nœud N sont P et D . Les noeuds A et E sont des meilleurs voisins de N pour la destination D, tandis que le nœud C est un meilleur voisin de N pour la destination P . Le nœud B , quant à lui, est un meilleur voisin de N pour chacune de ses destinations actives. Ces nœuds insèrent donc une entrée pour leurs meilleures destinations respectives. L'Annexe H.1 présente le contenu des caches de PREP1s ainsi initialisés par ces meilleurs voisins. Chaque meilleur voisin stocke aussi les coordonnées de chacune de ses meilleures destinations. Le nœud T quand à lui ne permet d'avancée géographique des paquets de données ni vers P , ni vers D . Contrairement aux meilleurs voisins de N , il n'exécute donc pas la phase 2 de la découverte de proximité.

Pour toute découverte de proximité, la phase 2 a pour objectif d'évaluer les possibilités d'avancée géographique à deux sauts de N vers une destination et englobe les étapes 3, 4, 5 et 6 de la découverte. Encore une fois, seuls les meilleurs voisins du nœud N pour au moins une destination active exécutent cette phase. A l'issu de cette phase, chacun de ces meilleurs voisins, connaîtra ses attributs pour ses meilleures destinations.

Dans le cadre d'une découverte de proximité à destinations multiples, nous noterons :

- MV_N , l'ensemble des nœuds qui constituent des meilleurs voisins de N pour au moins une de ses destinations actives;
- $MV_N(V)$, un nœud spécifique V qui est un meilleur voisin de N pour au moins une de ses destinations actives.

❖ Étape 3 de la découverte: Diffusion des PREQ2s par les meilleurs voisins

Tout comme dans une découverte de proximité à destination unique, les meilleurs voisins de N , MV_N , vont diffuser une requête de proximité de niveau 2 ou PREQ2 (Figure D.1 (b)). Cependant, le format de l'en-tête d'une PREQ2 est différent de celui de l'en-tête d'une PREQ2_ONE. La différence entre ces deux entêtes tient au fait qu'une PREQ2 ne contient aucune information concernant la distance du nœud émetteur aux destinations, comme le montre le Tableau annexé F.4. En effet, le nœud émetteur d'une

PREQ2 va insérer dans l'entête de celle-ci, d'une part, l'adresse du nœud initiateur de la découverte, N , d'autre part, l'adresse et les coordonnées de chacune des destinations actives du nœud initiateur, ainsi que le nombre de destinations actives de ce dernier. En définitive, chaque meilleur voisin MV_N copie ces informations directement de la PREQ1 correspondante reçue auparavant, avant de diffuser la PREQ2.

L'Annexe H.2 présente l'entête de la PREQ2 diffusée par les meilleurs voisins de N dans notre exemple.

❖ Étape 4 de la découverte: Réception des PREQ2s par les voisins à 2 sauts

Lorsqu'un nœud I reçoit pour la première fois une PREQ2 pour une découverte spécifique, il la mémorise, pour ne pas traiter les éventuelles PREQ2s pour cette même découverte qu'il est susceptible de recevoir ultérieurement.

Chaque nœud I vérifie ensuite s'il a mémorisé des informations concernant cette découverte dans son cache de PREP1. Si c'est le cas, il ne fait rien de plus, puisqu'il est un voisin direct de N . Sinon, le nœud I passe à l'étape suivante.

Bien sûr, le nœud initiateur de la découverte ne traite pas les PREQ2s de ses voisins.

❖ Étape 5 de la découverte: Diffusion des PREP2s par les voisins à 2 sauts

Les nœuds I ayant reçu une PREQ2 et remplissant les critères spécifiés à l'étape précédente, vont tous diffuser une PREP2 (Figure D.1 (c)). Le format d'une PREP2 est le même, qu'il s'agisse d'une découverte de proximité à destination unique ou à destinations multiples (voir Tableau annexé F.5). Seul le champ concernant le nombre de destinations relatif à la découverte est différent: pour une découverte à destination unique, il vaut 1, tandis que pour une découverte à destinations multiples, il sera égal à $NDst$.

Le nœud I évalue donc sa position et insère systématiquement l'adresse de chacune des destinations qui était incluse dans la PREQ2 reçue, avec sa distance à cette destination. Les champs $Discv_id$ et $NDst$ ont les mêmes valeurs que ceux de la PREQ2 correspondante et permettent d'identifier de façon unique cette découverte de proximité.

L'entête de la PREP2 diffusée par le nœud G contiendra donc les valeurs présentées dans l'Annexe H.3. Le nœud I diffuse ensuite sa PREP2.

❖ Étape 6 de la découverte: Réception des PREP2s par les meilleurs voisins

Encore une fois, seuls les nœuds qui ont émis une PREQ2 pour la découverte correspondante à une PREP2, traitent cette PREP2, soient les meilleurs voisins MV_N du nœud initiateur N . Comme nous l'avons vu précédemment, toute PREP2 contient les adresses des destinations actives du nœud initiateur et la distance du nœud émetteur I à chacune d'entre elles.

A la réception d'une PREP2, les meilleurs voisins MV_N sont donc en mesure de mettre à jour leur caractéristiques pour chacune de leurs meilleures destinations. Chaque meilleur voisin $MV_N(V)$ parcourt alors ses meilleures destinations dans son cache de PREP1s et, pour chacune de ses meilleures destinations, notée MDj avec $1 < j < N_MDst(MV_{N,D}(V))$, vérifie si le nœud émetteur de la PREP2 est la destination MDj :

- Si oui, il met à jour son indice d'accessibilité en conséquence, $IsDest(MV_N(V), MDj) = 1$.
- Sinon, le nœud $MV_N(V)$ compare sa distance à MDj avec la distance du nœud émetteur de la PREP2, I :
 - Si $DIST(I, MDj) < DIST(MV_N(V), MDj)$, alors le nœud I constitue un meilleur voisin de $MV_N(V)$ pour sa meilleure destination MDj . $MV_N(V)$ incrémente donc son indice de proximité, $IProx(MV_N(V), MDj) += 1$.
 - Sinon, $MV_N(V)$ ne fait rien de plus.

Le traitement des PREP2s est toujours soumis au délai *reply_delay* alloué la phase 2 de la découverte de proximité. Une fois ce délai expiré, la phase 2 de la découverte de proximité est terminée: les meilleurs voisins MV_N du nœud initiateur N connaissent leurs caractéristiques pour chacune de leurs meilleures destinations, c'est-à-dire leur indice d'accessibilité, leur indice de proximité et leur distance à la destination. Les meilleurs voisins MV_N peuvent passer alors à l'exécution de l'étape suivante.

L'Annexe H.4 présente le contenu du cache de PREP1s de chaque meilleur voisin de N à la fin de la phase 2 de la découverte de proximité. T est équivalent au délai *reply_delay* ajouté au temps d'insertion de l'entrée.

❖ Étape 7 de la découverte: Envoi des PREP1s par les meilleurs voisins

Lorsque le délai alloué à la phase 2 d'une découverte spécifique expire, seuls les meilleurs voisins MV_N du nœud initiateur de la découverte N , sont susceptibles de diffuser une réponse de proximité de niveau 1 ou PREP1 destinée au nœud N (Figure D.1 (d)). En effet, si son indice de proximité est nul pour chacune de ses meilleures destinations, un meilleur voisin $MV_N(V)$ ne diffusera pas de PREP1, car il est concave pour toutes ces destinations.

Le format de l'entête des PREP1s est décrit dans le Tableau annexé F.6. Les informations contenues dans l'entête d'une PREP1 sont issues du cache de PREP1s du nœud émetteur. Pour chacune de ses meilleures destinations MDj pour lesquelles il n'est pas concave (son indice de proximité n'est pas nul), chaque meilleur voisin $MV_N(V)$ insère, entre autres, les informations suivantes dans l'entête de la PREP1 avant de la diffuser:

- L'adresse de sa meilleure destination MDj ;
- Son indice d'accessibilité, $IsDest(MV_N(V), MDj)$;
- Son indice de proximité, $IProx(MV_N(V), MDj)$;
- Sa distance à la destination MDj , $DIST(MV_N(V), MDj)$.

L'entête de la PREP1 diffusée par le nœud A contiendra par exemple les informations présentées dans l'Annexe H.5.

Le meilleur voisin $MV_N(E)$ ne diffusera pas de PREP1 pour la découverte initiée par N , car il est concave pour son unique meilleure destination, D . Le nœud B , quant à lui, n'insérera que ses attributs pour la destination D , car il est concave pour son autre meilleure destination, P . Les autres meilleurs voisins de N , $MV_N(A)$ et $MV_N(C)$, diffusent une PREP1 contenant leurs attributs pour leur meilleure destination.

Lorsque le délai *reply_delay* de la phase 2 de la découverte de proximité arrive à terme, tous les voisins du nœud initiateur suppriment l'entrée correspondante de leur cache de PREP1s et les nœuds ayant mémorisé une PREQ2 pour cette découverte font de même dans leur cache de PREQ2s.

❖ Étape 8 de la découverte: Réception des PREP1s par le nœud initiateur

Les PREP1s issues d'une découverte à destinations multiples permettent au nœud initiateur de mettre à jour sa table de meilleurs voisins pour chacune de ses destinations actives. Le nœud initiateur de la découverte de proximité, N , traite uniquement les PREP1s qui lui sont destinées, c'est-à-dire celles dont la valeur du champ *discv_id* correspond à son adresse.

A la réception d'une PREP1, le nœud N parcourt les informations sur les meilleures destinations du nœud émetteur dans l'entête de la PREP1 et, pour chacune d'entre elles, met à jour, ou insère, les informations du nœud émetteur dans sa table de meilleurs voisins. Le processus de classement est ici le même que pour une découverte de proximité à destination unique. Ainsi, à la fin de sa découverte de proximité, la table de meilleurs voisins du nœud N contient les données présentées dans l'Annexe H.6.

La Figure D.2 résume schématiquement le déroulement d'une découverte de proximité à destinations multiples.

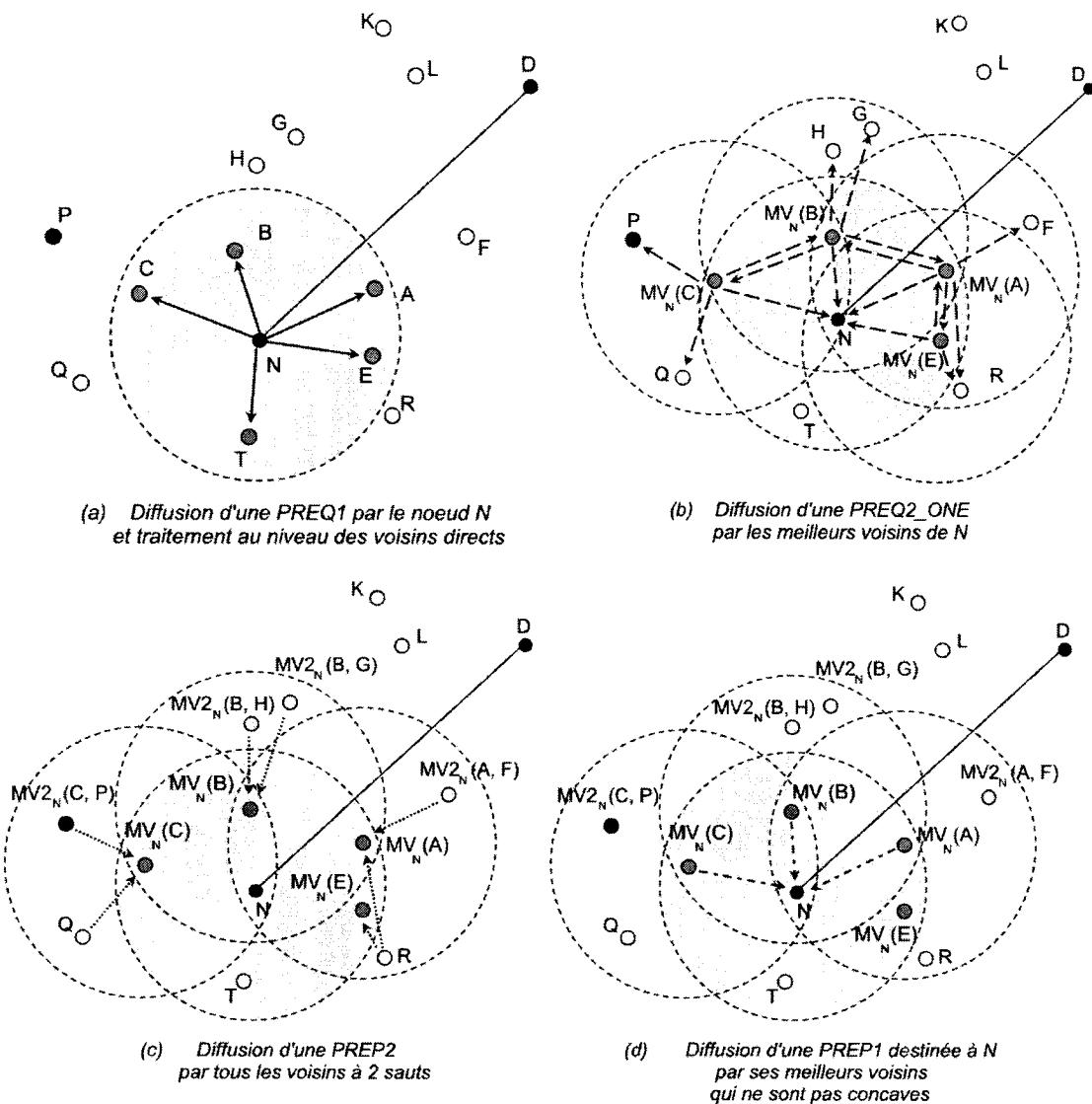


Figure D.1 : Échanges de paquets lors d'une découverte de proximité
à destinations multiples

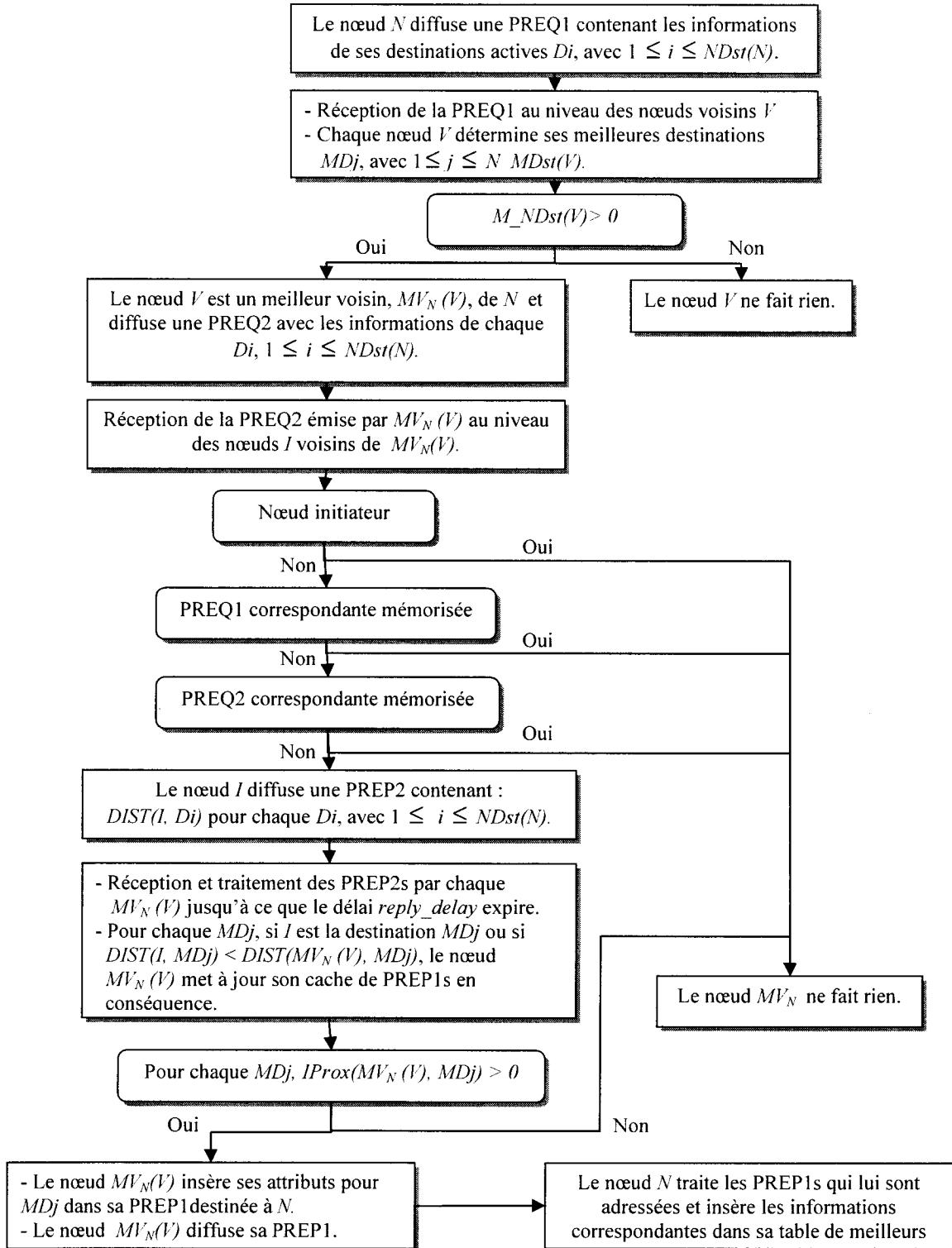


Figure D.2 : Déroulement d'une découverte de proximité à destinations multiples

ANNEXE E

Format du stockage des informations avec PARA

Différents types de stockage, tous temporaires, sont déployés par PARA. Les tableaux ci-dessous décrivent le format d'une entrée de chaque structure de stockage utilisée.

Tableau E.1 : Format d'une entrée du cache de PREQ1s au niveau d'un noeud N

Champ	Description
<i>Discv_id</i>	Adresse du nœud émetteur de la requête de proximité de niveau 1 et initiateur de la découverte
<i>Dst_init</i>	Nombre de destinations dont les informations sont incluses dans l'entête de la requête de proximité de niveau 1. Ce champ correspond aussi au nombre de destinations actives du nœud <i>Discv_id</i> .
<i>Dst_id</i>	- Si $Dst_init = 1$, adresse de l'unique destination dont les informations sont incluses dans l'entête de la requête de proximité de niveau 1. - Sinon, vaut -1.
<i>M_NDst</i>	Nombre de meilleures destinations du nœud <i>N</i> pour cette découverte. Si $nDst \geq 1$, alors pour chaque meilleur destination <i>MDi</i> du nœud <i>N</i> , avec $1 \leq i \leq M_NDst$:
<i>MDi</i>	Adresse de la meilleure destination <i>MDi</i> du nœud <i>N</i>
<i>x(MDi)</i>	Coordonnée en latitude de la destination <i>MDi</i>
<i>y(MDi)</i>	Coordonnée en longitude de la destination <i>MDi</i>
<i>z(MDi)</i>	Coordonnée en hauteur de la destination <i>MDi</i>
<i>IsDest(N, MDi)</i>	Indice d'accessibilité du nœud <i>N</i> pour sa meilleure destination <i>MDi</i>
<i>IProx(N, MDi)</i>	Indice de proximité du nœud <i>N</i> pour sa meilleure destination <i>MDi</i>
<i>DIST(N, MDi)</i>	Distance du noeud <i>N</i> à sa meilleure destination <i>MDi</i>
<i>t_ex</i>	Temps d'expiration de l'entrée, fixé à <i>reply_delay</i>

Tableau E.2 : Format d'une entrée de la table de meilleurs voisins

Champ	Description
Di	Adresse de la destination
Id	Adresse du meilleur voisin
$IsDest$	Indice d'accessibilité du meilleur voisin id pour Di
$IProx$	Indice de proximité du meilleur voisin id pour Di
$Dist$	Distance du meilleur voisin id à Di
t_{ex}	Temps d'expiration de l'entrée, fixé à $2\text{prequest_interval}$

Tableau E.3 : Format d'une entrée du cache de nœud concave

Champ	Description
Di	Adresse de la destination active concernée
$Concav_Id$	Adresse du nœud concave pour la destination Di
t_{ex}	Temps d'expiration de l'entrée, fixé à $2\text{prequest_interval}$

Tableau E.4 : Format d'une entrée du cache de PREQ2s

Champ	Description
$Discv_id$	Adresse du nœud initiateur de la découverte
Dst_init	Nombre de destinations dont les informations sont incluses dans l'entête de la requête de proximité de niveau 2. Ce champ correspond aussi au nombre de destinations actives du nœud $Discv_id$.
Dst_id	- Si $Dst_init = 1$, adresse de l'unique destination dont les informations sont incluses dans l'entête de la requête de proximité de niveau 1. - Sinon, vaut -1.
t_{ex}	Temps d'expiration de l'entrée, fixé à $reply_delay$

ANNEXE F

Format des entêtes PARA

Tableau F.1 : Format de l'entête PARA des paquets de données

Champ	Description
<i>Data_mode</i>	Mode d'acheminement du paquet : géographique, recouvrement ou topologique
<i>Iprox</i>	Indice de proximité du saut précédent du paquet
<i>Dx</i>	Coordonnées en latitude de la destination du paquet
<i>Dy</i>	Coordonnées en longitude de la destination du paquet
<i>Dz</i>	Coordonnées en hauteur de la destination du paquet
<i>Concav_id</i>	En mode recouvrement, adresse du nœud concave
<i>DistC</i>	En mode recouvrement, distance du nœud concave à la destination du paquet

Tableau F.2 : Format de l'entête d'une PREQ1

Champ	Description
<i>NDst</i>	Nombre de destinations actives du nœud émetteur de la PREQ1
Pour chaque destination active Di du nœud émetteur de la PREQ1, avec $1 \leq i \leq nDst$:	
Di	Adresse de la destination active Di
$x(Di)$	Coordonnée en latitude de la destination active Di
$y(Di)$	Coordonnée en longitude de la destination active Di
$z(Di)$	Coordonnée en hauteur de la destination active Di
$DIST(N, Di)$	Distance du nœud N , émetteur de la PREQ1, à la destination active Di

Tableau F.3 : Format de l'entête d'une PREQ2_ONE

Champ	Description
<i>Discv_id</i>	Adresse du nœud initiateur de la découverte de proximité
<i>Dst_id</i>	Adresse de la destination unique

Tableau F.4 : Format de l'entête d'une PREQ2

Champ	Description
<i>Discv_id</i>	Adresse du nœud initiateur de la découverte de proximité
<i>NDst</i>	Nombre de destinations actives du nœud initiateur de la découverte de proximité
Pour chaque destination active Di du nœud initiateur de la découverte, avec $1 \leq i \leq NDst$	
Di	Adresse de la destination active Di
$x(Di)$	Coordonnée en latitude de la destination active Di
$y(Di)$	Coordonnée en longitude de la destination active Di
$z(Di)$	Coordonnée en hauteur de la destination active Di

Tableau F.5 : Format de l'entête d'une PREP2

Champ	Description
<i>Discv_id</i>	Adresse du nœud initiateur de la découverte
<i>NDst</i>	Nombre de destinations actives du nœud initiateur de la découverte
Pour chaque destination Di , avec $1 < i < NDst$	
Di	Adresse de la destination Di
<i>Dist</i>	Distance du nœud émetteur de la PREP2 à la destination Di

Tableau F.6 : Format de l'entête d'une PREP1

Champ	Description
$Discv_id$	Adresse du nœud initiateur de la découverte
M_NDst	Nombre de meilleures destinations du nœud émetteur de la PREP1
Pour chaque destination Di , avec $1 < i < M_NDst$	
Di	Adresse de la destination Di
$IsDest(V, Di)$	Indice d'accessibilité du nœud émetteur de la PREP1 pour sa meilleure destination Di
$IProx(V, Di)$	Indice d'accessibilité du nœud émetteur de la PREP1 pour sa meilleure destination Di
$DIST(V, Di)$	Distance du nœud émetteur de la PREP1 à sa meilleure destination Di

ANNEXE G

Exemples de stockage de données dans le cadre de l'exemple de la Figure 3.4

G.1 Informations contenues dans les caches de PREP1s

Le tableau ci-dessous illustre les informations stockées dans le cache de PREP1s de chaque meilleur voisin $MV_{N,D}(V)$ de N à la fin de la phase 2. Les coordonnées de chaque meilleure destination sont aussi mémorisées.

Tableau G.1 : Contenu des caches de PREP1s pour la Figure 3.4

$MV_{N,D}(V)$	Discv_id	Dst_init	Dst_id	M_NDst	$IsDest(MV_{N,D}(V), D)$	$IProx(MV_{N,D}(V), D)$	$DIST(MV_{N,D}(V), D)$	t_{ex}
A	N	1	D	1	0	1	$DIST(A, D)$	T
B	N	1	D	1	0	2	$DIST(B, D)$	T
E	N	1	D	1	0	0	$DIST(E, D)$	T

T est équivalent au délai $reply_delay$ ajouté au temps d'insertion de l'entrée.

G.2 Contenu de la table de meilleurs voisins de N

Le tableau suivant illustre la table ordonnée des meilleurs voisins du nœud N pour la destination D à l'issu de sa découverte de proximité initiale avec $DIST(B, D) > DIST(A, D)$ et T est deux fois l'intervalle de découverte de proximité périodiques, $prequest_interval$, ajouté au temps d'insertion de l'entrée.

Tableau G.2 : Contenu de la table de meilleurs voisins de N pour la Figure 3.4

$MV_{N,D}(V)$	$IsDest$ $(MV_{N,D}(V), D)$	$IProx$ $(MV_{N,D}(V), D)$	$DIST$ $(MV_{N,D}(V), D)$	Tex
B	0	2	$DIST(B, D)$	T
A	0	1	$DIST(A, D)$	T

ANNEXE H

Exemples de stockage de données et d'entêtes PARA dans le cadre de l'exemple de la Figure annexée D.1

H.1 Initialisation des caches de PREP1s

Le Tableau H.1 présente le contenu des caches de PREP1s initialisés par les meilleurs voisins de N . Chaque meilleur voisin stocke aussi les coordonnées de chacune de ses meilleures destinations.

Tableau H.1 : Contenu des caches de PREP1s initialisés pour la Figure annexée D.1

$MV_N(V)$	Discv_id	Dst_Init	Dst_id	M_NDst	MDi	$IsDest(MV_N(V), MDi)$	$IProx(MV_N(V), MDi)$	$DIST(MV_N(V), Mdi)$	t_{ex}
A	N	2	-1	1	D	0	0	$DIST(A, D)$	T
E	N	2	-1	1	D	0	0	$DIST(B, D)$	T
C	N	2	-1	1	P	0	0	$DIST(C, P)$	T
B	N	2	-1	2	D	0	0	$DIST(B, D)$	
					P	0	0	$DIST(B, P)$	

H.2 Contenu de l'entête d'une PREQ2

Le Tableau H.2 présente l'entête de la PREQ2 diffusée par les meilleurs voisins de N .

Tableau H.2 : Contenu de l'entête d'une PREQ2 pour la Figure D.1

Discv_id	NDst	D1	X(D1)	y(D1)	z(D1)	D2	x(D2)	Y(D2)	Z(D2)
N	2	D	x(D)	Y(D)	z(D)	P	x(P)	y(P)	z(P)

H.3 Exemple d'entête d'une PREP2

L'entête de la PREP2 diffusée par le nœud G contiendra les valeurs contenues dans le Tableau H.3.

Tableau H.3 : Contenu de l'entête de la PREP2 diffusée par G pour la Figure D.1

Discv_id	NDst	D1	$DIST(I, D1)$	D2	$DIST(I, D2)$
N	2	D	$DIST(G, D)$	P	$DIST(G, P)$

H.4 Contenu des caches de PREP1s à la fin de la phase 2

Le Tableau H.4 présente le contenu du cache de PREP1s de chaque meilleur voisin de N à la fin de la phase 2 de la découverte de proximité. T est équivalent au délai *reply_delay* ajouté au temps d'insertion de l'entrée.

H.5 Exemple d'entête d'une PREP1

L'entête de la PREP1 diffusée par le nœud A contiendra par exemple les informations présentées dans le Tableau H.5.

Tableau H.4 : Contenu des caches de PREP1s pour la Figure D.1

$MV_N(V)$	Discv_id	Dst_init	Dst_id	M_NDSt	MDi	$IsDest(MV_N(V), MDi)$	$IProx(MV_N(V), MDi)$	$DIST(MV_N(V), MDi)$	t_{ex}
A	N	2	-I	1	D	0	1	$DIST(A, D)$	T
E	N	2	-I	1	D	0	0	$DIST(B, D)$	T
C	N	2	-I	1	P	1	1	$DIST(C, P)$	T
B	N	2	-I	2	D	0	2	$DIST(B, D)$	T
					P	0	0	$DIST(B, P)$	T

Tableau H.5 : Contenu de l'entête de la PREP1 diffusée par A pour la Figure D.1

Discv_id	Dst_init	Dst_id	M_NDSt	MDi	$IsDest(MV_N(V), MDi)$	$IProx(MV_N(V), MDi)$	$DIST(MV_N(V), MDi)$
N	2	-I	1	D	0	1	$DIST(A, D)$

H.6 Contenu de la table de meilleurs voisins du nœud N

A la fin de sa découverte de proximité, la table de meilleurs voisins du nœud N contient les données présentées dans le Tableau H.6.

Tableau H.6 : Contenu de la table de meilleurs voisins de N pour la Figure D.1

Dst_id	$MV_N(V)$	$IsDest$ ($MV_N(V), MDi$)	$IProx$ ($MV_N(V), MDi$)	$DIST(MV_N(V), MDi)$	t_{ex}
D	B	0	2	$DIST(B, D)$	T
	A	0	1	$DIST(A, D)$	T
P	C	1	1	$DIST(C, P)$	T

T est équivalent à trois fois l'intervalle de découverte de proximité périodiques, *prequest_interval*, ajouté au temps d'insertion de l'entrée.

ANNEXE I
Contenu de la table de meilleurs voisins de N
pour l'exemple de la Figure 3.6

Le contenu de la table de meilleurs voisins du nœud N pour la topologie illustrée par la Figure 3.6 est donné par le Tableau I.1. On suppose $DIST(MV_{N,D}(J), D) < DIST(MV_{N,D}(I), D) < DIST(MV_{N,D}(H), D)$.

Tableau I.1: Contenu de la table de meilleurs voisins de N pour la Figure 3.8

Dst_id	$MV_N(V)$	$IsDest$ $(MV_N(V), MDi)$	$IProx$ $(MV_N(V), MDi)$	$DIST(MV_N(V), MDi)$	t_{ex}
D	J	1	1	$DIST(J, D)$	T
	I	0	3	$DIST(I, D)$	T
	H	0	1	$DIST(H, D)$	T

ANNEXE J

Détails d'implémentation du simulateur NS

NS est basé sur une approche combinant deux langages de programmation. D'une part, le langage C++ est utilisé pour l'implantation des modules de base du simulateur, minimisant ainsi le temps d'exécution. D'autre part, une interface textuelle en OTcl est fournie à l'utilisateur. OTcl constitue un moyen flexible et puissant de contrôle de la simulation, comme le déclenchement d'événements, la configuration du réseau, etc.

Le modèle sans fil de NS a été ajouté par le groupe *CMU's Monarch* en tant qu'extension du simulateur de base pour la mobilité. Ce modèle repose essentiellement sur la définition d'un nœud mobile, *MobileNode*, permettant la simulation de réseaux ad hoc. La classe C++ *MobileNode* est dérivée de la classe parent *Node*, et constitue par conséquent un objet *Node* auquel sont ajoutées des fonctionnalités propres aux nœuds mobiles, telles que la capacité de se mouvoir, de recevoir et transmettre via un canal sans fil, etc. La Figure J.1 illustre les composants d'un noeud mobile.

L'agent est un composant essentiel d'un nœud mobile. Il modélise les constructeurs et les consommateurs de paquets IP. La classe *Agent* fournit des méthodes utiles au développement de la couche transport et à d'autres protocoles du plan de signalisation ou de gestion. Cette classe constitue la classe de base pour définir des nouveaux protocoles dans NS. Elle fournit notamment l'adresse locale et de destination et les fonctions pour générer les paquets.

La pile réseau d'un nœud mobile est quant à elle formée des éléments suivants :

- La couche liaison (LL), qui est similaire à celle d'un nœud classique excepté le fait qu'un module ARP lui est connecté. Ce module permet d'effectuer la conversion entre une adresse IP et une adresse physique MAC. Chaque paquet sortant (sur le canal) est transmis à la couche LL par l'agent de routage. La couche LL passe ensuite le paquet à la file d'attente de l'interface. De la même manière, tout paquet entrant (depuis le canal) est passé par la couche MAC à la couche LL, qui le transmet ensuite au point d'entrée du nœud, *node_entry_*.

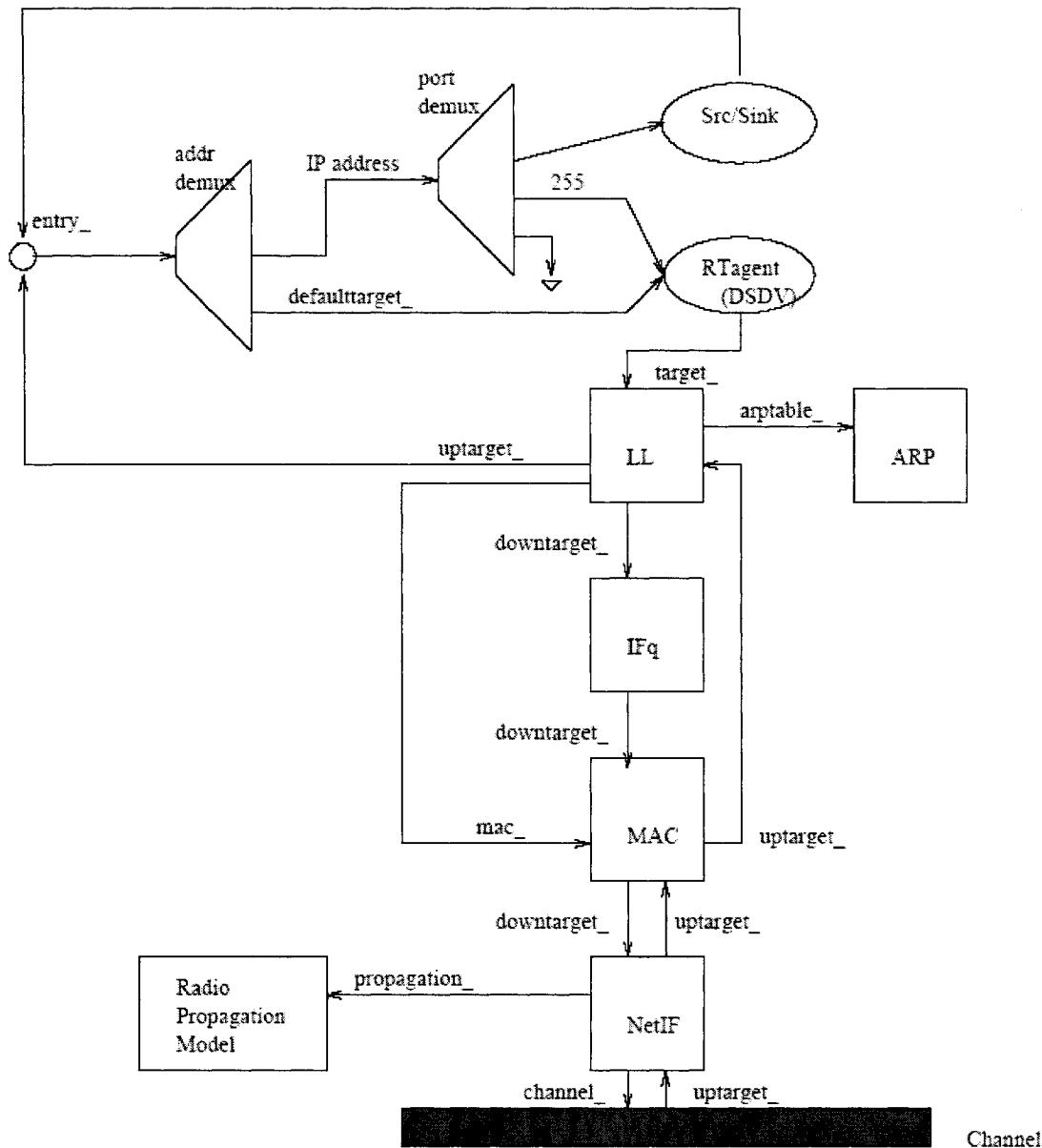


Figure J.1 : Composants d'un nœud mobile

- Le module ARP, implémentant le protocole de résolution d'adresse, reçoit les requêtes de la couche liaison. Si ARP possède l'adresse physique de la destination, il l'écrit dans l'en-tête MAC du paquet. Sinon, ARP diffuse une requête ARP et stocke

temporairement le paquet. Pour chaque adresse physique inconnue, il existe un tampon pour un seul paquet. Ainsi, si des paquets supplémentaires pour une même destination sont envoyés à ARP, le paquet stocké auparavant dans le tampon est abandonné. Une fois que l'adresse physique du prochain saut d'un paquet est connue, le paquet est inséré dans la file de l'interface.

- La file d'attente de l'interface est implémentée comme une file prioritaire, donnant priorité aux paquets de contrôle. Les paquets de contrôle sont en effet insérés en tête de file. Une fonctionnalité de ce composant permet notamment de parcourir tous les paquets présents dans la file et d'en retirer ceux destinés à une adresse spécifique.
- Le protocole Mac 802.11, utilisant la fonction distribuée de coordination DCF a été implémenté par le groupe *CMU*.
- L'interface réseau fait office d'interface physique qui est utilisée par le noeud mobile pour accéder au canal. Cette interface est sujette à des collisions, tandis que le modèle de propagation radio reçoit les paquets transmis par les interfaces des autres nœuds sur le canal. L'interface ajoute une étampe à chaque paquet transmis contenant les méta-données relatives à l'interface émettrice. Ces informations sont par la suite utilisées par le modèle de propagation au niveau de l'interface réceptrice pour déterminer si le paquet a la puissance minimale pour être reçu et / ou capté et / ou détecté par le nœud récepteur. Le modèle approxime l'interface radio DSSS (*LucentWaveLan direct-sequence spread-spectrum*).
- Le modèle de propagation radio utilise l'atténuation *Friss-space* pour les distances courtes et *Two-ray* pour les longues distances.
- Une antenne omnidirectionnelle dote d'un gain unitaire est utilisée par les nœuds mobiles.

ANNEXE K

Classes implémentées pour le routage PARA dans NS

Nous allons ici présenter brièvement les classes déployées pour PARA. La Figure K.1 illustre le diagramme de classes de PARA. Par souci de clarté, nous avons omis de ce diagramme les classes implémentant les temporiseurs.

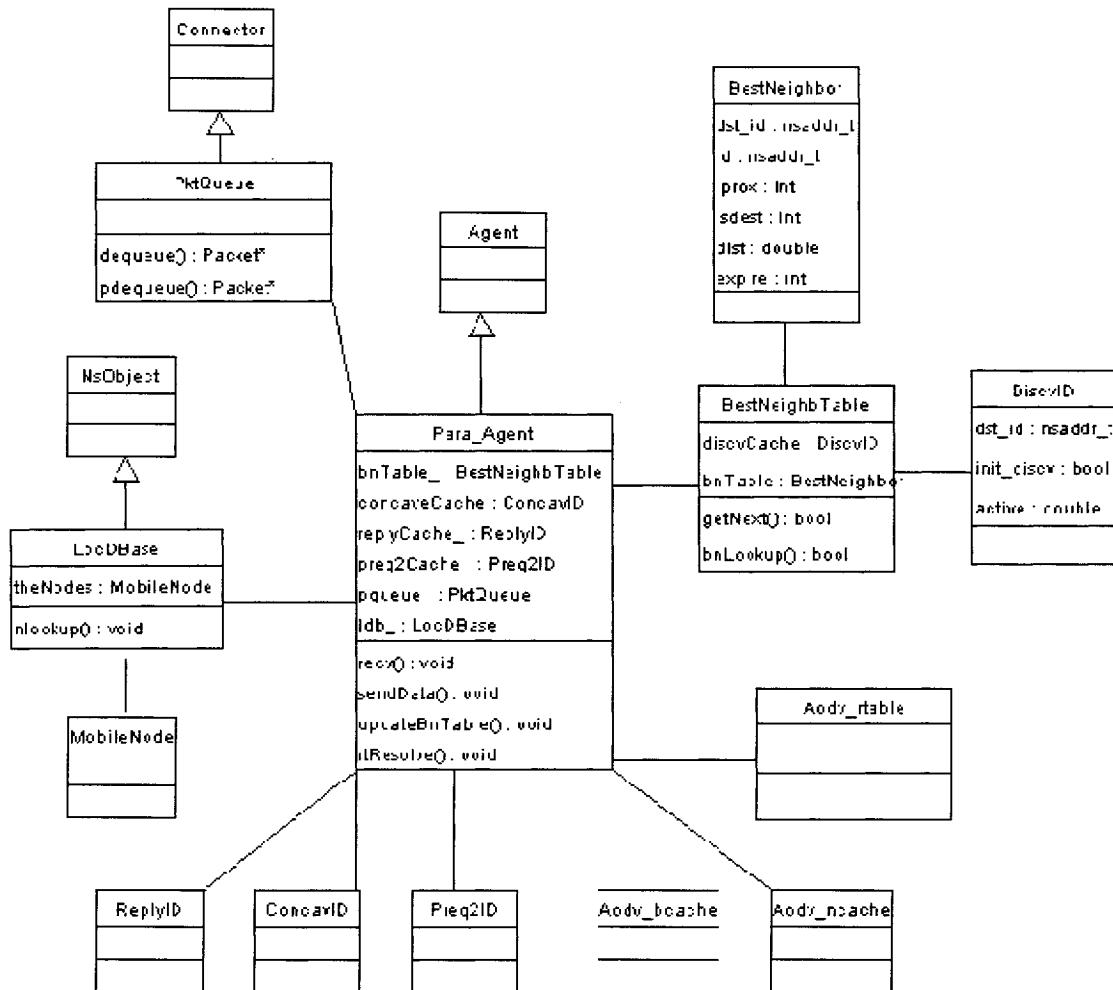


Figure K.1 : Diagramme de classes de PARA

Classe *ParaAgent*

Dans NS, les agents de routage sont des entités qui implémentent un algorithme de routage spécifique. Un nouvel algorithme est implémenté en dérivant une nouvelle classe d'agent de la super classe *Agent* de NS. Ainsi, la classe *ParaAgent* implémente le protocole PARA. Les fonctions les plus importantes de cette classe sont les suivantes : La fonction *recv()*, comme son nom l'indique, est responsable du traitement général de la réception des paquets, incluant les paquets de données générés. La fonction ajoute l'en-tête PARA aux paquets générés, après avoir récupéré les coordonnées de la destination via la fonction *LocDBase:mn_lookup()*. Les paquets de contrôle sont quant à eux transmis aux fonctions *recvPreq()* et *recvPrep()*, selon qu'il s'agit d'une requête ou d'une réponse de proximité. Pour chaque paquet de données reçu, la fonction met à jour l'état du noeud. Selon les données contenues dans la table de meilleurs voisins et son mode de transmission, un paquet peut être transmis via la fonction *sendData()* ou *rtResolve()*, ou encore inséré dans la file de paquets *pqueue* le temps de la découverte de proximité initiale. Le rôle de la fonction *recv()* est donc aussi d'initialiser le déclencheur pour les découvertes de proximité périodique et de lancer des découvertes de proximité initiale si besoin.

La fonction *updateBntable()*, appelée à la réception d'une PREP1, parcourt la table de meilleurs voisins du noeud, et met à jour les informations du meilleur voisin émetteur de la PREP1 pour chaque destination active insérée dans l'en-tête de la PREP1. L'insertion se fait de manière ordonnée, selon les caractéristiques du meilleur voisin, comme nous l'avons vu dans la Section 3.3.4.

La fonction *sendData()* implémente les changements de mode d'acheminement et le choix du prochain noeud transmetteur d'un paquet de données.

Enfin, la fonction *rtResolve()* est utilisée lors du routage topologique d'un paquet de données, d'une part, pour diffuser les RRQs si le noeud n'a pas de route pour la destination du paquet et d'autre part, pour transmettre le paquet au prochain saut selon la table de routage de AODV si celle-ci est peuplée.

Classe *LocDBase*

La classe *LocDBase* implémente le service de localisation du réseau ad hoc, et hérite de la super classe *Connector* de NS. Cette classe stocke en tout temps les adresses et positions de chacun des nœud du réseau ou instance de la classe *MobileNode* de NS. La fonction *mn_lookup()* parcourt ce tableau et retourne un pointeur sur le nœud mobile correspondant à l'adresse reçue en entrée. Celui-ci peut, par la suite, être utilisé pour récupérer des informations telles que la position et la vitesse du nœud.

Classe *BestNeighbor*

La classe *BestNeighbor* définit le format d'une entrée dans la table de meilleurs voisins et contient donc l'adresse de la destination concernée, l'adresse du meilleur voisin, son indice d'accessibilité, son indice de proximité, sa distance à cette destination, ainsi qu'un temps d'expiration.

Classe *BestNeighTable*

La classe *BestNeighTable* implémente la table de meilleurs voisins, le cache de découverte et toutes les manipulations inhérentes, telles que l'insertion, la suppression et la mise à jour d'un meilleur voisin dans la table ou d'une destination active dans le cache de découvertes. La fonction *bnLookup()* permet de s'assurer que l'on a des meilleurs voisins pour une destination spécifique. Par ailleurs, la fonction *getNext()* retourne l'adresse du meilleur voisin en tête de liste de la table de meilleurs voisins pour une destination donnée, ainsi que les informations du nœud appelant pour cette destination, telles que son indice de proximité et sa distance à la destination.

Classe *DiscvID*

La classe *DiscvID* décrit la structure d'une entrée dans le cache de découverte tel que nous l'avons évoqué dans le chapitre précédent.

Classe *PktQueue*

Le tampon permettant de stocker temporairement les paquets de données en attendant leur traitement est implémenté par la classe *PktQueue*. Il s'agit en fait d'une file, dotée notamment des fonctionnalités suivantes.

La fonction *dequeue()* est utilisée uniquement lorsque l'on est en mode topologique. Elle permet de retirer de la file les paquets de données dont le mode d'acheminement est topologique et destinés à un nœud donné.

La fonction *pdequeue()*, quant à elle, permet de récupérer, selon le paramètre en entrée, les paquets de données pour une destination précise ou encore tous les paquets de données contenus dans la file dont le mode d'acheminement est géographique ou recouvrement. Si aucune destination n'est spécifiée, la fonction vérifie qu'il n'y a pas de découverte initiale en cours pour les paquets retournés.

Classes *ConcavID*, *ReplyID* et *Preq2ID*

Le format des entrées des caches de nœuds concaves, de PREP1 et de PREQ2 est défini respectivement par les classes *ConcavID*, *ReplyCache* et *Preq2Cache*.

Classes *AodvRTable*, *AodvNCache* et *AodvBCache*

Les trois classes suivantes sont utilisées uniquement pour le routage topologique et sont tirées de l'implémentation d'AODV dans NS. La classe *AodvRTable* implémente la table de routage d'AODV et n'est peuplée qu'à l'issu d'une découverte de route lancée par un nœud initiant un passage en mode d'acheminement topologique pour un paquet de données. La classe *AodvNCache* implémente le cache permettant de stocker les informations sur la connectivité locale du nœud. Enfin, la classe *AodvBCache* implémente le cache permettant de mémoriser les RREQs reçues afin d'éviter qu'elles ne soient rediffusées plus d'une fois.

ANNEXE L

Structures décrivant les entêtes des paquets PARA

```

struct hdr_prox_id{
    nsaddr_t id;
    u_int8_t isdest;
    u_int8_t iprox;
    double dist;
};

struct hdr_para_preq1{
    u_int8_t prep_type;
    u_int8_t nDst; /*Nombre de meilleures destinations*/
    nsaddr_t discv_id;
    struct hdr_prox_id theDst [MAX_DST];

    inline void init (){
        nDst = 0;
    }

    inline void addDst (nsaddr_t id, u_int8_t isdest, u_int8_t iprox,
    double dist ){
        theDst[nDst].id      = id;
        theDst[nDst].isdest  = isdest;
        theDst[nDst].iprox   = iprox;
        theDst[nDst].dist    = dist;
        nDst++;
    }

    inline int size() {
        int sz = 0;
        sz = 2 * sizeof(u_int8_t) + sizeof(u_int32_t) + nDst * ( 2 *
            sizeof(u_int32_t) + 2 * sizeof(u_int8_t)) ;
        return sz;
    }
};

```

Figure L.1 : Structure *hdr_para_preq1* décrivant les champs de l'entête d'une PREQ1

```

Struct hdr_prox_id{
    nsaddr_t id;
    u_int8_t isdest;
    u_int8_t iprox;
    double dist;
};

struct hdr_para_prep1{
    u_int8_t prep_type;
    u_int8_t nDst; /*Nombre de meilleures
destinations*/
    nsaddr_t discv_id;
    struct hdr_prox_id theDst [MAX_DST];

    inline void init () {
        nDst = 0;
    }

    inline void addDst (nsaddr_t id, u_int8_t isdest, u_int8_t
iprox,
    double dist ){
        theDst[nDst].id      = id;
        theDst[nDst].isdest  = isdest;
        theDst[nDst].iprox   = iprox;
        theDst[nDst].dist    = dist;
        nDst++;
    }

    inline int size() {
        int sz = 0;
        sz = 2 * sizeof(u_int8_t) + sizeof(u_int32_t) + nDst * ( 2
*
            sizeof(u_int32_t) + 2 * sizeof(u_int8_t)) ;
        return sz;
    }
};

```

Figure L.2 : Structure *hdr_para_prep1* décrivant les champs de l'entête d'une PREP1

```
struct hdr_para_data{
    u_int8_t    data_type;
    u_int8_t    data_mode;
    u_int8_t    iprox;
    double      dx;
    double      dy;
    double      dz;
    double      distC;
    nsaddr_t    concav_id;

    inline int size() {
        int sz = 0;
        sz = 4 * sizeof(u_int32_t) + 3 * sizeof(u_int8_t);
        return sz;
    }
};
```

Figure L.3 : Structure *hdr_para_data* décrivant les champs de l'entête PARA d'un paquet de données

ANNEXE M

Evaluation de l'impact des paramètres *discv_delay* et *reply_delay* sur la performance de PARA

Afin d'évaluer l'impact des paramètres *discv_delay* et *reply_delay* sur la performance de PARA, nous avons réalisé pour chaque topologie de réseau, une série de 6 simulations d'une durée de 200s en faisant varier les valeurs de ces paramètres. Les tests ont été effectués dans des réseaux ad hoc de 50 nœuds statiques, avec 30 connexions établies et différentes connectivités. Les résultats présentés dans les Tableaux M.1, M.2 et M.3 sont donc les moyennes des valeurs obtenues pour chaque série de simulations. Par ailleurs, nous noterons *TxR* le taux de paquets reçus, *TxTopo*, la proportion de paquets reçus en mode topologique et *Hops* le nombre de sauts moyens effectués par les paquets.

Tableau M.1 : Résultats obtenus dans un réseau de connectivité 21 (1500x300m)

Reply_delay	Discv_delay	TxR.	TxTopo	Délai	Hops
0.04	0.06	0.798	0.296	0.8906	3.154
	0.07	0.788	0.305	0.9433	3.147
	0.08	0.790	0.295	0.9728	3.154
	0.09	0.784	0.295	0.9829	3.140
0.05	0.07	0.802	0.291	0.8913	3.160
	0.08	0.772	0.262	0.9955	3.103
	0.09	0.786	0.269	0.9905	3.119
	0.1	0.780	0.259	0.9964	3.103
0.06	0.08	0.7984	0.280	0.9712	3.178
	0.09	0.782	0.269	0.9970	3.139
	0.1	0.785	0.270	1.0605	3.146
	0.11	0.777	0.254	1.0284	3.098
0.07	0.09	0.800	0.279	1.0071	3.188
	0.1	0.786	0.267	1.0600	3.136
	0.11	0.778	0.2544	1.0159	3.1132
	0.12	0.772	0.2398	1.062	3.0944
0.08	0.1	0.801	0.255	0.9557	3.134
	0.11	0.779	0.239	1.0309	3.091
	0.12	0.792	0.259	1.013	3.137
	0.13	0.778	0.246	1.0418	3.122

Tableau M.2 : Résultats obtenus dans un réseau de connectivité 13 (500x500m)

Reply_delay	Discv_delay	TxR.	TxTopo	Délai	Hops
0.04	0.06	0.732	0.421	1.1743	4.137
	0.07	0.734	0.421	1.2234	4.129
	0.08	0.738	0.4021	1.242	4.119
	0.09	0.737	0.4180	1.1708	4.137
0.05	0.07	0.747	0.409	1.2685	4.148
	0.08	0.747	0.409	1.2751	4.122
	0.09	0.72994	0.415	1.2754	4.149
	0.1	0.7267	0.3747	1.2514	4.094
0.06	0.08	0.731	0.391	1.3801	4.123
	0.09	0.735	0.388	1.3221	4.121
	0.1	0.727	0.385	1.2944	4.1298
	0.11	0.731	0.372	1.2911	4.067
0.07	0.09	0.726	0.364	1.3246	4.089
	0.1	0.730	0.336	1.3333	4.110
	0.11	0.734	0.336	1.1998	4.013
	0.12	0.720	0.361	1.3659	4.081
0.08	0.1	0.732	0.366	1.2872	4.083
	0.11	0.739	0.370	1.3478	4.118
	0.12	0.748	0.354	1.2663	4.081
	0.13	0.7322	0.3723	1.3035	4.0873

Tableau M.3 : Résultats obtenus dans un réseau de connectivité 5 (1500x1300m)

Reply_delay	Discv_delay	TxR.	TxTopo	Délai	Hops
0.04	0.06	0.3812	0.342	0.1452	4.4522
	0.07	0.3815	0.360	0.1655	4.524
	0.08	0.3796	0.348	0.1652	4.508
	0.09	0.383	0.330	0.0977	4.437
0.05	0.07	0.379	0.371	0.1378	4.452
	0.08	0.3755	0.3274	0.1568	4.4699
	0.09	0.3812	0.349	0.1911	4.494
	0.1	0.377	0.334	0.1521	4.452
0.06	0.08	0.378	0.349	0.1566	4.477
	0.09	0.375	0.323	0.1664	4.410
	0.1	0.339	0.339	0.1954	4.461
	0.11	0.371	0.344	0.1456	4.464
0.07	0.09	0.379	0.337	0.1597	4.418
	0.1	0.381	0.345	0.1420	4.559
	0.11	0.380	0.631	0.0998	4.465
	0.12	0.3730	0.342	0.1756	4.5216
0.08	0.1	0.382	0.366	0.1387	4.556
	0.11	0.378	0.326	0.1173	4.443
	0.12	0.378	0.344	0.1602	4.537
	0.13	0.379	0.340	0.0991	4.422

ANNEXE N

Résultats de quelques simulations dans des réseaux ad hoc de 100 nœuds

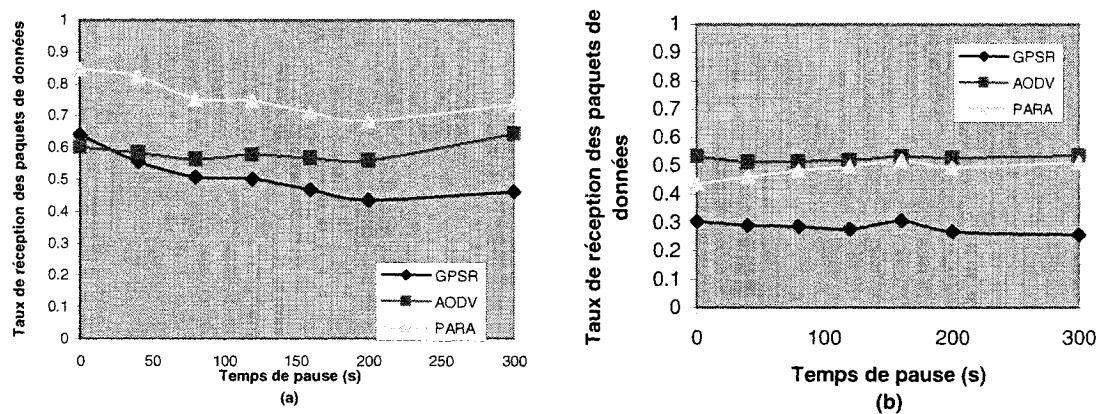


Figure N.1 : Taux de réception des paquets de données dans un réseau de 100 nœuds dispersés (5 nœuds en moyenne) pour (a) 10 et (b) 30 connexions CBR

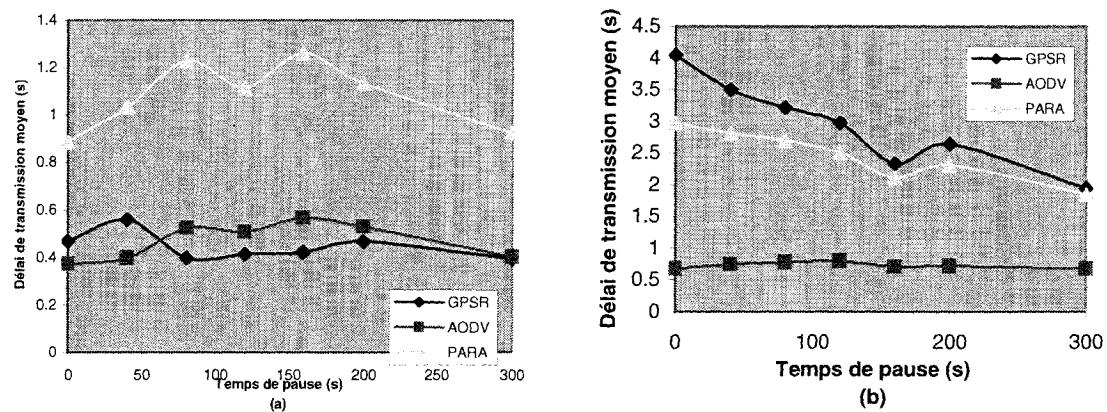


Figure N.2 : Délai de transmission moyen dans un réseau de 100 nœuds dispersés (5 nœuds en moyenne) pour (a) 10 et (b) 30 connexions CBR

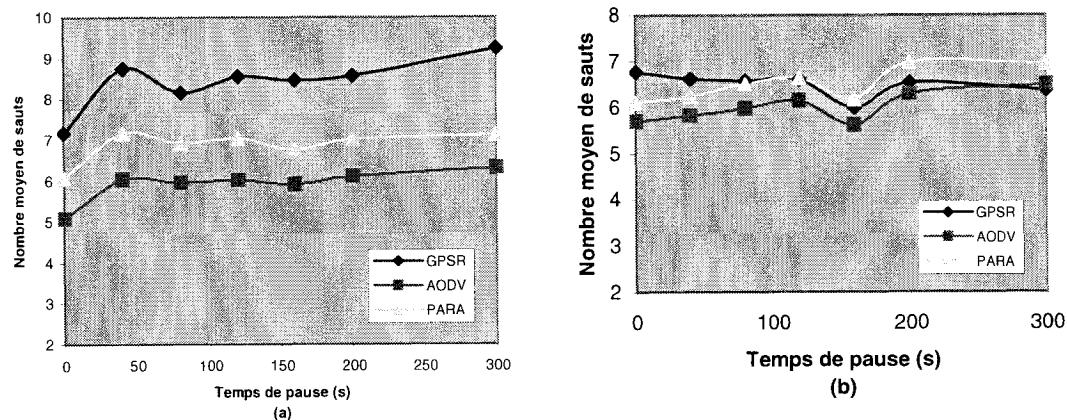


Figure N.3 : Longueur moyenne des routes dans un réseau de 100 nœuds dispersés (5 nœuds en moyenne) pour (a) 10 et (b) 30 connexions CBR